

Meta Analysis of Microarray Data Using Gene Regulation Pathways

Saira Ali Kazmi¹, Yoo-Ah Kim¹, Baikang Pei¹, Nori Ravi¹, David W. Rowe², Hsin-Wei Wang¹, Alan Wong¹, Dong-Guk Shin¹

¹*Department of Computer Science and Engineering, University of Connecticut, Storrs, CT 06269*

²*Department of Genetics and Developmental Biology, University of Connecticut Health Center, Farmington, CT 06030*

E-mail:saira@engr.uconn.edu

Abstract

Using microarray technology for genetic analysis in biological experiments requires computationally intensive tools to interpret results. The main objective here is to develop a “meta-analysis” tool that enables researchers to “spray” microarray data over a network of relevant gene regulation relationships, extracted from a database of published gene regulatory pathway models. The consistency of the data from a microarray experiment is evaluated to determine if it agrees or contradicts with previous findings. The database is limited to “activate” and “inhibit” gene regulatory relationships at this point and a heuristic graph based approach is developed for consistency checking. Predictions are made for the regulation of genes that were not a part of the microarray experiment, but are related to the experiment through regulatory relationships. This meta-analysis will not only highlight consistent findings but also pinpoint genes that were missed in earlier experiments and should be considered in subsequent analysis.

1. Introduction

Microarray tools enable scientists to determine expression levels of thousands of genes in a biological assay at once [1, 2]. This technology is utilized in a wide array of research areas including the study of gene function, gene classification, gene pathway modeling, disease management, and drug discovery [1-5]. Different statistical and quantitative analysis steps are required before any comparisons can be made using microarray data. Numerous methods have been proposed for the statistical and quantitative analyses and are discussed extensively in literature [6-8].

The main objective here is to develop a tool that performs the end stage of analysis, namely “meta-analysis,” normally taking place after the statistical analysis is complete. This meta-analysis step assumes that the data has already been quality assessed and normalized and the selection of regulated genes has been done. One group of popular meta-analysis methods is clustering. Clustering gathers genes together based on their expression profiles and other selected parameters. It has become a useful tool for hypothesizing which regulating genes might share common biological functions. There have also been many other quantitative data analysis techniques such as Principal Component Analysis, Self-Organizing Maps, and Bayesian Analysis [9-12], which are designed to derive biological meaning out of the statistically significant gene lists.

Recently, utilizing gene regulation pathways to analyze and interpret microarray gene profile data has received increased attention due to its ability to provide biologically interpretable results. The results are more intuitive for biologists because they can visualize the results given in terms of pathway models instead of long lists of genes and their associated expression scores. Several resources are already available that provide pathway-related information including KEGG [13], BioCarta [14], EcoCyc [15], and MetaCyc [16]. Mostly these resources are useful for viewing and displaying static pathways and do not allow microarray data to be integrated with the pathway data. A program called GenMAPP [17, 18] enables biologists to “spray” expression data onto pathways and allows creation and modification of pathways (called MAPPs) that can be shared with other researchers. This is a valuable visual tool, but it does not permit the user to fully integrate the information contained within a pathway with the microarray experiment data for prediction and hypothesis generation.

It is important to be able to visualize data in a biological context, like in GenMAPP, but what is missing in such systems is that they do not exploit the contextual information available from the pathways to “automatically” discover knowledge, that is, finding whether the microarray data either supports or refutes some segments of the pathway. It is safe to say that the current biological knowledge organized into pathways *may not* accurately describe what is really happening in the biological system being tested in the experiment. Therefore, we think that it is crucial to assess how much and what portion of the microarray data is either consistent or inconsistent with what is already known. In the system developed here, we assume that the gene regulation information is stored in a database. Data from microarray experiment is then “sprayed” over these gene regulatory network relationships, which are modeled as a directed graph, similar to pathway diagrams. An algorithm is developed that is capable of predicting expression values for genes that are not available from the microarray experiment. The algorithm will also identify sub-networks that are either highly consistent or highly inconsistent with the observed microarray expression data.

The rest of the paper is organized as follows: In Section 2, we define the network model that integrates pathway relationships with microarray data. Section 3 includes definitions of various consistency related terms. Section 4 discusses

heuristics used to predict missing microarray expression values. Section 5 contains the pseudo-code for the algorithm. The output from a hypothetical network is also shown in section 6 and we finally conclude the paper in section 7 and discuss future directions.

2. The Network Model

The gene regulatory pathways are modeled as a directed graph. A network $N = (V, E^+, E^-)$ represents known binary relationships between genes. Each vertex in the network $v \in V$ corresponds to a gene. Each positive edge $e \in E^+$ connecting vertex u to v represents the activation relationship between u and v . Each negative edge $e \in E^-$ connecting u to v indicates that the gene corresponding to u is known to inhibit the gene corresponding to v . These activation and inhibition relationships are the two most common gene regulation relationships and this model is limited to only these relationships between genes.

Given a pair of genes, u and v , multiple experiments may provide evidence for conflicting relationships. For example, one experiment may say “ u activates v ” whereas another experiment may say “ u inhibits v ”. The validity of the relationship depends on the context of the experiment and here we assume that the pathway information is relevant to the experiment and is not contradictory.

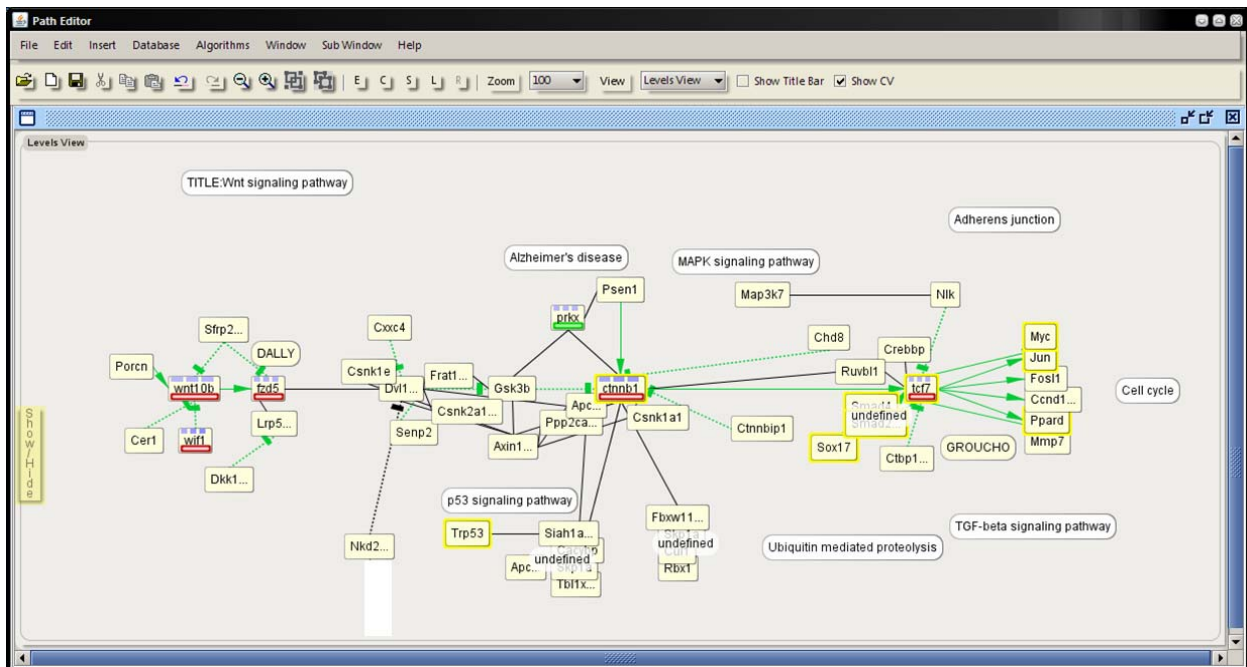


Figure 1. A Typical Graph with Microarray Data Sprayed Over a Pathway Network

Microarray data is represented as a set \mathbf{R} of numeric values corresponding to a subset of vertices in \mathbf{V} , denoted by \mathbf{V}_R . We limit ourselves to two discrete regulation values (+1 for upregulation and -1 for downregulation). The nodes in \mathbf{V} without any regulation information from the microarray experiment belong to a subset named \mathbf{V}_U . Figure 1 illustrates this network model. It shows three types of nodes (although we treat these types as indistinguishable): genes, protein complexes, and biological functions. A relationship is represented by an edge e that may be one of two types. A pointed arrowhead shows $e \in \mathbf{E}^+$, while a flat (or a rounded) arrowhead shows $e \in \mathbf{E}^-$.

3. Consistency Types

At a particular node, there may be more than one type of incoming edge, each denoting different kinds of relationships between the elements (activation or inhibition). The consistency of an edge or a relationship is determined by looking at the expression values of the corresponding nodes and the type of the relationship between them. An incoming edge at a node may be assigned red, green, yellow or orange based on the following consistency levels.

Consistent. A relationship is consistent if all gene regulation is consistent with the pathway data (Fig. 2 i, Fig. 2 viii). This is denoted by a green edge.

Inconsistent. A relationship is totally inconsistent if gene regulation is inconsistent with the pathway data and there is no other way of explaining the inconsistency (Fig. 2 iii, Fig. 2 vi). This is denoted by a red edge.

Inconsistent Explainable. A relationship is inconsistent explainable if gene regulation is inconsistent with the pathway data but there is at least one other incoming green edge with an up-regulated tail end that explains the regulation at this node (Fig. 2ii, Fig. 2v). This is denoted by a yellow edge as in Figure 2ii. The edge between B and C is yellow because even when $B=-1$ and $C=+1$, the edge between A and C is consistent and $C=+1$ is justified by $A=+1$.

Inconsistent Unexplainable. A relationship is inconsistent unexplainable if gene regulation is inconsistent with the pathway data and there is no other incoming green edge present at the node to explain the inconsistency. Presence of a gene is considered to be of more consequence than the

absence of a gene. If the relationship with the upregulated gene is inconsistent and the relationship with the downregulated gene is consistent, it is considered unexplainable (Fig. 2 iv, Fig. 2 vii). This is denoted by an orange edge.

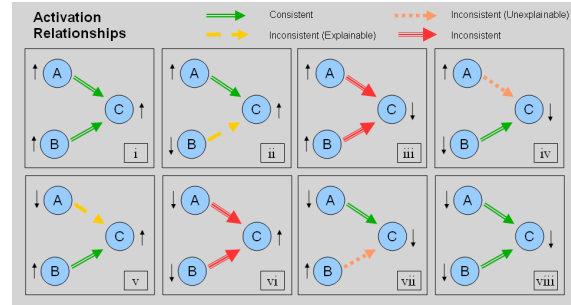


Figure 2. Consistency for Activation Relationships at Node C

It is assumed that green edges are most desirable and yellow edges are the next best. The algorithm described in Section 5 predicts the missing values for all nodes so that the resulting network forms a most consistent view with the experimental data.

4. Predicting Values at a Node

To obtain the best possible solution in terms of consistency, we first define the consistency level of the network, and our goal is to design an assignment algorithm. Consider the graph in Figure 3. There are three nodes in the network and it is assumed that the expression level for node A is known from the microarray experiment to be upregulated. The aim is to predict expression levels at the other nodes (node B and C) such that the data forms a highly consistent view with the pathway information. The four possible models based on potential assignments of the unknown nodes are shown in the Figure with the corresponding consistency assignments, generating four different solutions.

Assignment in Figure 3.1 forms the most consistent model having all green edges. Clearly, solutions 2, 3, and 4 are not preferable as each has only one green edge. The assignments in Figures 3.2 and 3.3 are regarded as superior to the one in Figure 3.4 because although there is one green relationship in each, the green edge is connecting a node with actual microarray data. We prefer green edges that involve microarray data over green edges that do not involve any microarray data. Also Figure 3.3 gives a better assignment than Figure 3.2 since there is a yellow edge between the nodes (inconsistent explainable relationships are considered better than totally inconsistent or unexplainable).

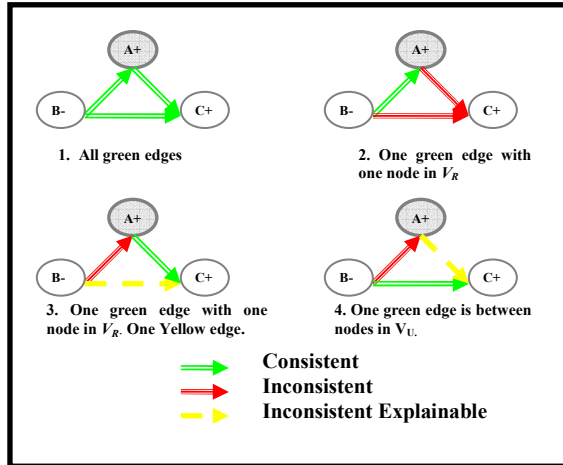


Figure 3. Possible Solutions

In general, we want to maximize green edges that involve microarray data at one end while yellow edges are preferred over orange and red edges. Our strategy is to search the solution space where each unknown node may be assigned either +1 or -1 while maximizing the consistency as previously defined. The total number of possible assignments is exponential in the size of the set V_U of unknown nodes, e.g. for a network with 10 unknown nodes, there are $2^{|V_U|} = 2^{10} = 1024$ possible assignments to search.

5. Algorithm

We now describe our algorithm. The goal is to assign positive or negative values to the unknown nodes while maximizing the consistency as defined.

To measure consistency, we define a color vector $C = [c_1, c_2, c_3, c_4]$ for each node where each element of the vector corresponds to the number of edges with consistency level green, yellow, orange, and red in order. Figure 4 shows color vectors for two different values in node B (the entire network is given in section 6). For the case where the value of B is positive, color vector C is given as $[1, 0, 1, 1]$, as there are one green, no yellow, one orange and one red. The colors represent the consistency types defined in Section 3. Color vector C can be decomposed into two vectors C_R and C_U (i.e., $C = C_R + C_U$). Vectors C_R and C_U correspond to color vectors for edges for which at least one endpoint is in V_R , and for edges connecting two nodes in V_U , respectively.

Given a network N and microarray data R , Algorithm *FindModel* (N, R) assigns values to nodes in V_U . Nodes are processed in order such that a node with more neighbors having known/estimated values is processed first.

Algorithm *FindModel* (N, R)

- (1) \forall edge $e' = (u, v)$ s.t. $u, v \in V_R$, assign consistency based on definitions in section 3.
- (2) **for** $i = 1$ to $|V_U|$
Sort nodes in V_U using the following rules:
 - Nodes with more edges to nodes in V_R come first.
 - A node with less number of edges to nodes in V_U is preferred.
 - With the same number of edges to V_R and V_U , a node with more estimated neighbor nodes gets priority.
Process the node, v , at the top of the list
run *NodeProcess*(v)
- (3) If \exists red edge $e' = (v, w)$, **run** *UpdateRed* (w)

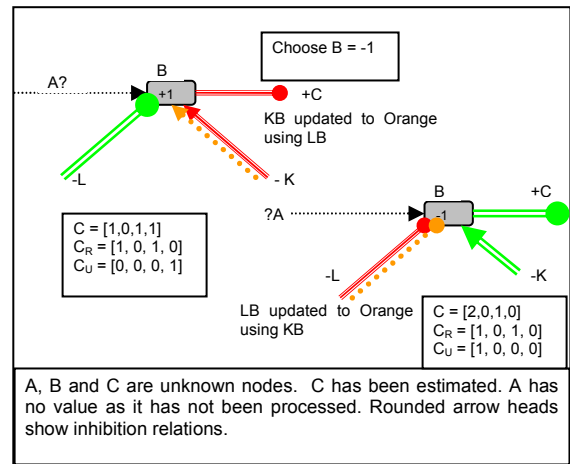


Figure 4. Computation at Node B

To process a node, we compute color vectors for each assignment and we choose a positive or negative value based on the following rules (*priority rules*):

- (1) Maximize Green and then Yellow in C_R
- (2) Maximize Green and then Yellow in C_U
- (3) Maximize Orange in C_R and then in C_U

A different order of priority may be used to best suit a given data and analysis requirements. Algorithm *NodeProcess*(v) provides the details of the procedure.

Algorithm *NodeProcess* (v)

- 1 Create two temporary copies, X and Y of node v and assign a positive and negative regulation at the nodes, respectively.
- 2 Evaluate the color of the edges for X and Y based on regulation at both ends for all edges.
- 3 **Run** *UpdateRed* (X), **run** *UpdateRed* (Y)
- 4 Calculate color vectors for X and Y .
- 5 Choose between X or Y using C_R and C_U and the priority rules (1)-(3).
- 6 \forall green edge $e' = (v, u)$ **run** *UpdateRed* (u)

To update the color of red edges to yellow or orange, Table 1 is used. While processing a node, colors are assigned after looking at incoming edges and the expression value at the other end. An incoming red edge at a node may be changed to yellow or orange if there is at least one incoming green edge to the node. In case there is no incoming green edge, the red edge may not be changed (Figure 4). If there is more than one incoming green edge/consistent relationship present at a node, then the one with a positive value at the tail of the edge is used first, as this assigns yellow (inconsistent explainable) rather than orange (inconsistent unexplainable).

Table 1. Reference Table to Update Red to Yellow or Orange

Consistent Incoming Edge Present at Node	Update all Incoming Red Edges
+ → -	no change
- → -	red → orange
+ → -	red → yellow
- → -	no change
+ → +	red → yellow
- → +	no change
+ → +	no change
- → +	red → orange

A red edge may also be updated at a node v if a neighboring node u was processed and there is a consistent edge from u to v . Node v is then updated so that any incoming red edge is converted to yellow if u is upregulated and orange otherwise. The following pseudo-code provides the details of algorithm *UpdateRed*(v).

Algorithm *UpdateRed* (Node v)
if ($v = +1$ and \exists green edge $e' = (u, v)$)
 if ($e' =$ activation and $u = +1$)
 update all incoming red at v to yellow
 if ($e' =$ inhibition and $u = -1$)
 update all incoming red at v to orange
else if ($v = -1$ and \exists green edge $e' = (u, v)$)
 if ($e' =$ activation and $u = -1$)
 update all incoming red at v to orange
 if ($e' =$ inhibition and $u = +1$)
 update all incoming red at v to yellow

6. Results

The algorithm was tested using a Java™ implementation with the graph shown in Figure 5 as input. The nodes A , B and C denote nodes that do not have any microarray expression value. Nodes H , I , J , K and L have regulation information encoded as +1 (for upregulation) and -1 (for downregulation). Nodes A , B and C are in the graph because inhibition

and activation links were found between these genes from the gene regulatory pathway database.

According to the processing order defined in *FindModel*(\mathbf{N} , \mathbf{R}), the nodes C , B , and A are processed in order. The solution is drawn in Figure 6 and shows one *inconsistent* and three *inconsistent unexplainable* links. The rest of the links are all *consistent*.

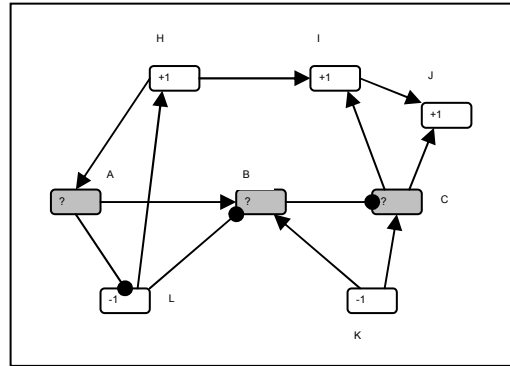


Figure 5. Example Network

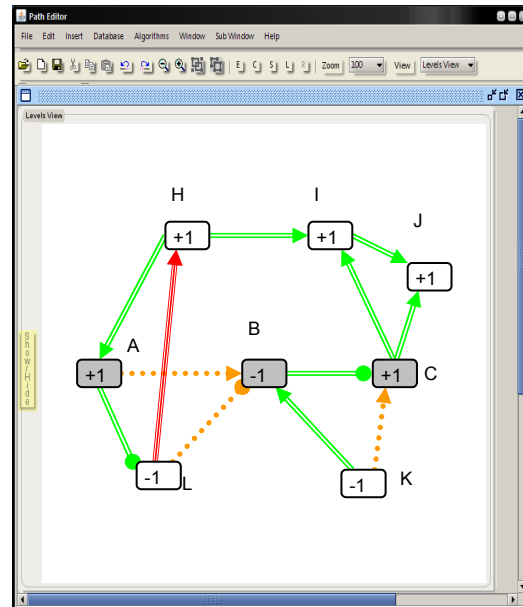


Figure 6. Output from Consistency Algorithm

The solution, shown in Figure 6, has a red edge between nodes H and L . This highlights either an error in the microarray experiment or suggests that the relationship found in the database is not verified and has to be investigated further.

Input graphs are constructed based on the relationships found in gene regulatory pathway models. Ultimately, when we apply our algorithm to a large number of gene regulation pathways, we will be able to automatically identify consistent or inconsistent parts within each pathway. By

examining the analysis outcome (consistency values) portrayed on multiple pathways one will be able to begin hypothesizing about pathways that are either actively involved or inhibited.

7. Conclusion and Future Work

Microarray data meta-analysis is a complex and evolving problem which we believe has not gained enough attention in the past. Particularly, this meta-analysis field lacks a framework capable of automating the derivation of biological insights from the microarray data. We envision that our proposed work can form a basis towards starting a new research direction with the focus on automating the discovery of biological interpretations out of the high-throughput microarray gene expression experiments.

In the future, our framework could be extended into multiple directions. The model presented here deals only with relationships that have atomic nodes at each end. In a more complicated model, each node could be non-atomic, meaning a complex node that is made up of multiple gene objects (e.g., protein complexes). The relationships themselves can be extended by including other types beyond activation and inhibition (e.g., phosphorylation, ubiquitination, etc.). It may also be useful to enable the software to deal with more than one set of expression data and spray it over the network model simultaneously for comparison. This approach may not be applicable to all domains where there is not enough literature available to establish a network structure. However, for organisms that have extensive pathway information available, this method can be of great benefit.

Acknowledgements

This work was supported in part by a grant from NIH/NIGMS, Grant No, P20 GM65764-04.

References

[1] A. Alizadeh et. al., "Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling," *Nature*, vol. 403, pp. 503 - 511, 2000.

[2] M. Arbeitman, et. al., "Gene expression during the life cycle of *Drosophila melanogaster*," *Science*, vol. 297, pp. 2270 - 2275, 2002.

[3] J. Welsh et. al., "Analysis of gene expression profiles in normal and neoplastic ovarian tissue samples identifies candidate molecular markers of epithelial ovarian cancer," *Proc Natl Acad Sci USA*, vol. 98, pp. 1176 - 1181, 2001.

[4] Y. Moreau et. al., "Functional bioinformatics of microarray data: from expression to regulation," *Proceedings of the IEEE*, vol. 90, pp. 1722-1743, 2002.

[5] D. Kostka and R. Spang, "Finding disease specific alterations in the co-expression of genes," *Bioinformatics*, vol. 20, pp. I194 - I199, 2004.

[6] F. Cordero, M. Botta, and R. A. Calogero, "Microarray data analysis and mining approaches," *Brief Funct Genomic Proteomic*, 2008.

[7] M. Reimers, "Statistical analysis of microarray data," *Addiction Biology*, Mar, pp. 23-35, 2005.

[8] P. Khatri and S. Draghici, "Ontological analysis of gene expression data: current tools, limitations, and open problems," *Bioinformatics*, vol. 21, pp. 3587 - 3595, 2005.

[9] Y. Tamada et. al., "Estimating gene networks from gene expression data by combining Bayesian network model with promoter element detection," *Bioinformatics*, vol. 19, pp. II227 - II236, 2003.

[10] M. Eisen, P. Spellman, P. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proc Natl Acad Sci USA*, vol. 95, pp. 14863 - 8, 1998.

[11] F. De Smet et. al., "Adaptive quality-based clustering of gene expression profiles," *Bioinformatics*, vol. 18, pp. 735 - 746, 2002.

[12] M. P. S. Brown et. al., "Knowledge-based analysis of microarray gene expression data by using support vector machines," *Proceedings of the National Academy of Sciences*, vol. 97, pp. 262-267, 2000.

[13] M. Kanehisa et. al., "From genomics to chemical genomics: new developments in KEGG," *Nucleic Acids Res*, vol. 34, pp. D354 - 7, 2006.

[14] "BIOCARTA" <http://www.biocarta.com>

[15] R. M. Karp PD et. al., "The EcoCyc Database," *Nucleic Acids Res*, vol. 30, pp. 56-8, 2002.

[16] P. D. Karp, M. Riley, S. M. Paley, and A. Pellegrini-Toole, "The MetaCyc Database," *Nucl. Acids Res.*, vol. 30, pp. 59-61, 2002.

[17] S. N. Dahlquist KD, Vranizan K, Lawlor SC, Conklin BR., "GenMAPP, a new tool for viewing and analyzing microarray data on biological pathways," *Nat Genet.*, vol. 31, pp. 19-20, 2002.

[18] S. Doniger et. al., "MAPPFinder: using Gene Ontology and GenMAPP to create a global gene-expression profile from microarray data," *Genome Biol*, vol. 4, pp. R7, 2003.