Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB 2007)

Associative Artificial Neural Network for Discovery of Highly Correlated Gene Groups Based on Gene Ontology and Gene Expression

Ji He, Xinbin Dai and Xuechun Zhao‡
Bioinformatics Group, Plant Biology Division, The Samuel Roberts Noble Foundation
2510 Sam Noble Parkway, Ardmore, OK 73401, USA
{jhe, xdai, pzhao‡}@noble.org

‡ Author of correspondence.

Abstract— **The advance of high-throughput experimental technologies poses continuous challenges to computational data analysis in functional and comparative genomics studies. Gene Ontology (GO) annotation and transcriptional profiling using gene expression array have been two of the major approaches for system-wide analysis of gene functions and gene interactions. In the literature, extensive studies have been reported in each aspect. Yet there is a lack of efficient algorithm that discover associative patterns across these two data domains. We proposed a mixture model associative artificial neural network to tackle this deficiency. The algorithm inherits the theoretical foundation of Adaptive Resonance Associative Map (ARAM), with essential redefinition of pattern similarity measures and learning functions. The proposed algorithm is capable of clustering data based on both GO semantic similarity and expressional correlation, for the purpose of systematically discovering genome-wide, highly correlated gene groups, which in turn suggest similar or closely related functions. We applied the proposed algorithm to the analysis of the *Saccharomyces cerevisiae* (yeast) dataset and obtained satisfactory results.**

I. Introduction

Gene functions and gene interactions are among the central topics of functional and comparative genomics studies. Recent advance of genome-wide experimental technologies has made it possible to investigate large number of genes in a systematic fashion. Practices that study multiple genes in similar functional families and/or multiple biological pathways in the same experimental design are common nowadays, whereas comprehensive studies of global interactome networks are emerging [31]. These high-throughput technologies often generate large-scale data, and pose constant challenges to computational data analysis research.

Given a library of unknown sequences, it has been a routine paradigm to predict their functions through computational approaches. Continuous efforts are being done to regularize the functional annotation using controlled vocabulary for the ease of comparison and categorization. The Gene Ontology (GO, at http://www.geneontology.org) [8] has been widely adopted for this purpose. GO provides a set of well defined annotation terms organized by means of a directed acyclic graph (DAG). Computational GO annotation of unknown sequences is essentially based on sequential homology to existing sequences with confirmed GO annotations. Studies have shown that GO annotation generally conforms with other sequence similarity based annotation paradigms such as TIGR's [17], [19]. Besides the notable advantage in controlled and formalized vocabulary, the hierarchical structure of the GO DAG facilitates functional annotation in different precision levels. The closer a term is to the ontology root, the rougher the annotation is. Methods for functional categorization based on GO have been extensively documented in the literature [17], [18], [19], [20], [24]. They usually involve an unsupervised learning approach to group genes according to a pre-defined similarity (or contrarily, dissimilarity/distance) function. Various similarity functions with different theoretical bases exist in the literature, one of which will be briefly reviewed in Section II-C.

While the semantic annotation and grouping of gene functions are generally based on computational sequence homology, transcriptional profiles are commonly adopted to investigate and verify gene functions in a more biological manner. The spread of genome-wide microarray technologies [7], [30] has made it possible to obtain large scale gene expression data in a short time frame. Experiments have shown that gene products with similar expression patterns may have similar or closely related functions (e.g. in the same biological pathway). As such, systematic discovery of gene expression patterns is of great value to biologists. A variety of clustering methods have been applied to this problem and have shown satisfactory performance [1], [9], [10], [13], [16], [25]. Commercial software including GeneSpring (http://www.agilent.com) and Spotfire (http://www.spotfire.com) have already gained a large user population.

In both GO and gene expression analysis, *grouping* plays an important role. By clustering highly-correlated genes into different groups, we greatly reduce the work on investigating individual genes and obtain a bird-eye-view of the whole genome, which is essential to functional and comparative genomics studies. Since we may group genes according to either GO annotations or expression patterns, one natural question would be: How high is the correlation between these two types of groupings? In other words, if a set of genes are found having similar expression patterns, would they really

1-4244-0710-9/07/$20.00 ©2007 IEEE

be annotated with closely related GO terms? Recent work of Sevilla et al. [23] partially answered this question by conforming the satisfactory correlations between expression and different GO similarity measures. There report in turn supports the validity of GO annotation based on computationally sequential homology.

Much to our surprise, while extensive research have been done on pattern analysis individually from either GO or gene expression data, few studies are reported to fully integrates knowledge from *both* fields. In reality, it has been a routine practice to investigate the major functional categories enriched by the genes of interest reflected in a microarray experiment. Yet there is a lack of intelligent and automatic paradigm to assist such studies. In our prior practices, we had to randomly pick up a group of genes with a certain expression pattern, and further investigate their functions individually; or, to limit our study to a set of genes with functions of interest, and further observe their expression patterns. Either way has proven to be human labor intensive, and rather critically, difficult to navigate genome-wide data in a systematic manner. Recent updates of some commercial software (such as GeneSpring and Spotfire) improve human analytical efficiency by displaying the pie-chart of GO term distribution, according to a list of selected genes and pre-defined GO categories. Yet, they are greatly dependent to human judgement and lack the functionality to intelligently discover system-wide, significant patterns.

To tackle this deficiency, we proposed a novel mixture model artificial neural network (ANN) for systematical discovery of gene patterns based on knowledge from both GO annotation and gene expression. The proposed algorithm incorporates these two types of data into a single infrastructure and system-widely identify significant gene grouping, with each group containing highly correlated genes in terms of *both* GO and expression similarities. We applied the proposed algorithm to the public *Saccharomyces cerevisiae* (yeast) genome dataset and obtained satisfactory results.

The rest of this paper is organized as follows. Section II introduces our proposed algorithm in detail. Section III reports our experiment on the yeast dataset. Section IV summarizes our conclusions and proposes future work.

II. Method

A. Computational Challenges to Grouping Genes based on GO and Expression Data

With reference to the term "grouping", end users are expecting an abstract view over the whole dataset – data within each group are similar and/or closely related to each other, whereas there are clear boundaries between different groups. With this understanding, an appropriate definition of the (dis)similarity measure is critical to the meaningfulness of the output clusters.

GO annotation and gene expression are provided in the nature that, GO is descriptive while gene expression is quantitative. Appropriate quantization of GO terms presents the first challenge to our study. This topic anyway has been extensively studied. Various prior studies are of valuable references to our studies. The integration of these two types of data however remains as the greatest challenge. Since the data are from different sources (knowledge domains), it is not theoretically valid to normalize and present them into a *single* vector format as most clustering algorithms require. Therefore our research starts with reported algorithms that are capable of handling inputs from multiple knowledge domains.

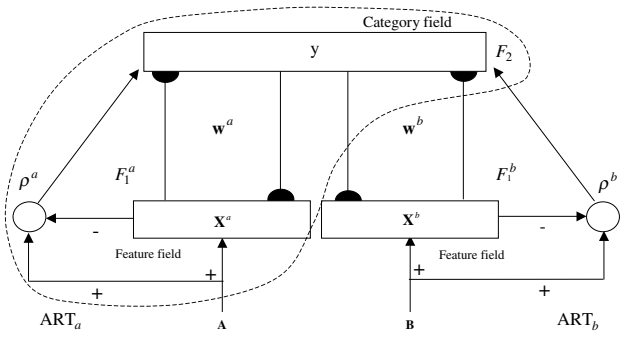


Fig. 1. The architecture of Adaptive Resonance Associative Map (ARAM) neural network.

B. Adaptive Resonance Associative Map (ARAM)

The Adaptive Resonance Associative Map (ARAM) [26] belongs to the family of Adaptive Resonance Theory (ART) self-organizing neural networks [3]. Like another member of the family, ART-MAP [4], ARAM is capable of incrementally learning recognition categories (pattern classes) and multidimensional maps of patterns. Yet compared to ART-MAP, ARAM contains a simplified pattern matching and learning process. The architecture of ARAM (Figure 1) can be understood as an overlap of two ART networks. An ARAM network has two individual short term memory (STM) layers F_1^a and F_1^b, responding to independent input signals **A** and **B** respectively, but an shared long term memory (LTM) layer F_2 that encodes the associated knowledge from these two feature fields. The learning of the network is guided by an orienting subsystem with two logical gates, defined with two vigilance parameters (ρ_a and ρ_b respectively). The logical gates conditionally switch and reset the network state according to predefined rules, and hence affect knowledge encoding in the LTM.

ARAM acquires its domain knowledge through an online, hard competitive learning process. In summary, the recognition neurons compete to each other in response to each incrementally presented (online) input stimulation, with only one neuron that wins the competition and gains knowledge from the input (hard learning). The ARAM learning paradigm has been comprehensively documented in the literature [26] and is summarized below for a better understanding of this paper.

1) Inputs and Recognition Categories: ARAM requires inputs **A** and **B** represented in vector format. There is a built-in normalization link between the input and the STM layer F_1, denoted as $\mathbf{I}_a = \Re\mathbf{A}$ and $\mathbf{I}_b = \Re\mathbf{B}$. The definition of the

normalization link varies depending on the application. Each LTM recognition category j in F_2 layer is associated with two adaptive weight templates, i.e. $\mathbf{w} = (\mathbf{w}_j^a|\mathbf{w}_j^b)$, $\mathbf{w}_j^a$ and $\mathbf{w}_j^b$ being same dimensional as $\mathbf{I}_a$ and $\mathbf{I}_b$ respectively. Initially, the F_2 recognition field contains a null set (zero category). Upon incremental presentation of input signals, it is adaptively expanded to encode new knowledge.

2) Category Competition: In response to a input signal $\mathbf{I} = (\mathbf{I}_a|\mathbf{I}_b)$, the similarity between the input and each LTM recognition category j is evaluated according to

$$T(\mathbf{I}, \mathbf{w}_j) = \gamma T_a(\mathbf{I}_a, \mathbf{w}_j^a) + (1-\gamma) T_b(\mathbf{I}_b, \mathbf{w}_j^b), \qquad (1)$$

where $\gamma \in [0, 1]$ is an *associative contribution* parameter, $T_a(.)$ (or $T_b(.)$) is a predefined function, referred to as the *choice function*, that measures the similarity in domain space a (or b). The linear combination $T(.)$ is referred to as the network's choice function. The category J that receives the highest choice score $T(\mathbf{I}, \mathbf{w}_J) = \max\{T(\mathbf{I}, \mathbf{w}_j)\}$ is marked as the *winner* of the competition.

3) Resonance or Reset: If the competition generates a winner category J, its similarity to the input $\mathbf{I}$ is further confirmed in domain spaces a and b individually, using another set of *match functions*, i.e. $M_a(\mathbf{I}_a, \mathbf{w}_J^a)$ and $M_b(\mathbf{I}_b, \mathbf{w}_J^b)$. The network is said to reach *resonance* if both match scores are over the corresponding *vigilance* threshold ρ, denoted as

$$\begin{cases} M_a(\mathbf{I}_a, \mathbf{w}_J^a) \geq \rho_a & \text{and} \\ M_b(\mathbf{I}_b, \mathbf{w}_J^b) \geq \rho_b, \end{cases} \qquad (2)$$

during which network learning ensures, as defined in the next step.

Mismatch reset happens when either of the match score does not reach the vigilance value. During mismatch reset, the network redo the winner selection and resonance check iterations with mismatched categories excluded, until a selected winner causes network resonance, or all LTM categories are reset.

4) Network Learning: Once the search ends and network resonance is achieved, the attentional subsystem updates the weight vector $\mathbf{w}_J$ by incorporating the input knowledge correspondingly from field a and b, according to two *learning functions*:

$$\begin{cases} \mathbf{w}_J^{'a} = L_a(\mathbf{I}_a, \mathbf{w}_J^a), & \text{and} \\ \mathbf{w}_J^{'b} = L_b(\mathbf{I}_b, \mathbf{w}_J^a). \end{cases} \qquad (3)$$

In case all LTM categories are reset but the network fails to reach a resonance state (or when F_2 is null upon the presentation of the first input), the network switches to *fast commitment* learning mode, which essentially expand the F_2 recognition field by creating a direct copy of the input as a new LTM category. That is, $\mathbf{w}_{new}^{'a} = \mathbf{I}_a$ and $\mathbf{w}_{new}^{'b} = \mathbf{I}_b$.

It deserves to review a few unique features of the ARAM architecture. Firstly, like ART, ARAM uses two functions (choice and match) to evaluate the similarity between the input and recognition category. These two functions may or may not have same definition, optionally providing a different view to conform the degree of pattern matching. Secondly, the use of vigilance thresholds ensures only significantly similar patterns may be grouped together. On the other hand, the vigilance parameters primarily affect the clustering process. Lower vigilance thresholds generally lead to fewer recognition categories, and hence rougher clustering result. Lastly while most importantly, ARAM provides an effective infrastructure for learning of associative knowledge from two different domains. Depending on the input signals, ARAM may be applied to different learning tasks. Examples include text and document classification [14], [27], personalized knowledge management [28], and associative rule mining [29].

Variations of ARAM models exist in the literature, according to the definition of normalization, choice, match and learning functions. For example, ARAM-2A consists of two ART-2A models [5] using second level normalization and cosine similarities, while fuzzy ARAM consists of two fuzzy ART models [6] using complemental normalization and similarity functions derived from fuzzy set theory. However, after close investigation of existing ARAM models, we find that there is not an "out of box" solution for the analysis of GO annotation and gene expression data. This is because most reported work used same sets of similarity measures with the same theoretical origin, as they assumed the inputs from pattern fields a and b are isogenous. This however is not true in our application.

Based on this understanding, we borrowed ARAM's architecture and learning process which have well established theoretical foundation, but redefined a set of similarity measures and learning functions that suite the nature of our heterogeneous data. We name our modified network *Mixture Model ARAM* to differ our practice to existing work, highlighting the fact that in our variation, fields a and b work on different data models. The details of the proposed network is given below.

C. Mixture Model ARAM for GO Annotation and Gene Expression Data

1) Inputs and Recognition Categories: Our application of the Mixture Model ARAM is straightforward: for each gene product, we use ARAM's pattern field a to encode its expression pattern and b to encode its GO annotation. Following common practices, the expression pattern is presented in vector format, while the GO annotation is presented as a set of descriptive GO terms, denoted as $\mathbf{I} = (\mathbf{I}_a|\mathbf{I}_b) = (\overrightarrow{\text{exp}}|\{\text{go terms}\})$. Understandably, each of the LTM recognition category encodes an associative pattern $\mathbf{w} = (\mathbf{w}^a|\mathbf{w}^b)$, where $\mathbf{w}^a$ and $\mathbf{w}^b$ respectively are the expression pattern and GO annotation term(s) representative to the inputs that form the corresponding category. Therefore, by investigating the major recognition patterns (in terms of category size), we are able to systematically review the significant gene functional groups, in terms of similar expression pattern and closely related functional annotations.

2) Pattern Field for Gene Expression: Since gene expressions are represented in vector format, it is relatively not difficult to handle them in the network. We understand that normalization of gene expression is still an arguable topic nowadays. Based on different natures of data, variations of

normalization techniques. Thus our mixture model ARAM network does not contain a fixed normalization link. Instead, we assume all input expressions are properly pre-normalized. Like the ART-2A [5] paradigm, we used symmetric choice and match functions to evaluate gene expression similarity. That is, both choice and match functions are defined with the *Pearson correlation coefficient* between two expressions, denoted as:

$$T_a(\mathbf{I}_a, \mathbf{w}^a) = M_a(\mathbf{I}_a, \mathbf{w}^a) = \frac{(\mathbf{I}_a - E(\mathbf{I}_a)) \cdot (\mathbf{w}^a - E(\mathbf{w}^a))}{||\mathbf{I}_a - E(\mathbf{I}_a)||||\mathbf{w}^a - E(\mathbf{w}^a)||} \quad (4)$$

where $E(.)$ and $||.||$ are the mean (expectation) and norm (length) of a vector respectively. Our use of Pearson correlation coefficient measure follows the majority of reported work. Particularly, if the expression is normalized with standard distribution (with 0.0 mean and 1.0 norm), our definition is equivalent to that of ART-2A, essentially being the cosine similarity of two vectors.

As to network learning, we adopted the common *adaptive learning rule*, given as:

$$\mathbf{w}^{'a} = L_a(\mathbf{I}_a, \mathbf{w}^a) = \mathbf{w}^a + \eta(\mathbf{I}_a - \mathbf{w}^a) \quad (5)$$

where the parameter $\eta \in [0, 1]$ is commonly referred to as the *learning rate*. With this learning process, the recognition pattern adaptively correct its weights to reduce the error between the recognition pattern and the input, so that when the network is stabilized, the recognition pattern will reflect the cluster centroid.

3) Pattern Field for GO Annotation: Given GO annotations in format of descriptive terms, it is not necessary to further normalize these terms. One of the focuses of our work is on the measurement of GO similarity. Since the establishment of GO generally follows the same paradigm on other lexical taxonomies such as the WordNet (http://wordnet.princeton.edu), a variety of similarity measurements in lexical taxonomy study have been applied to GO. Resnik [21] compared different semantic similarity measures against human judgements. He reported that in the controlled taxonomy, Information Content [22] based measurement outperformed two other measures, namely Edge Counting and Probability. Sevilla et al.'s study [23] further showed that Resnik's semantic similarity based on Information Contents produced relatively more consistent correlation to the gene expression similarity over two other authors'. Therefore, we adopted Resnik's Information Content based similarity measure in our studies. The measure is reviewed as below.

Information Content: Originated from probability studies, the concept of Information Content has existed for multiple decades [22]. Briefly, the information content of a lexical concept/class c is quantified as the negated log of its likelihood $p(c)$ in the corpus, formalized as

$$i(c) \equiv -log(p(c)) = -log(\frac{f(c)}{N}), \quad (6)$$

where $f(c)$ is the frequency of the instances of concept c and N is the corpus size.

In order to apply Information Content to GO, we treat each GO term as a conceptual class that subsumes the term itself as well as all its descendent (children) terms. Hence the likelihood on a GO term t is calculated according to

$$p(t) = \frac{\text{size\_of}\{C(t)\}}{\text{size\_of}\{C(root)\}}, \quad (7)$$

where $C(t)$ is the set of terms being subsumed by t, and $root$ is the most top level (root) term. The more specific a GO term t is, the lower the likelihood $p(t)$ is, and hence the higher information content $i(t)$ it has. Particularly, the information content of the root term has the lowest value 0.0.

Similarity between two GO Terms: Based on the definition above, Resnik [21] proposed the measurement of the similarity between two GO terms as the information content of their *minimal subsumer*. A so-called minimal subsumer of two terms t_i and t_j, denoted as $\text{ms}(t_i, t_j)$, is the subsumer that has the minimal likelihood (and hence maximal information content). To formalize:

$$\begin{aligned} \text{sim}(t_i, t_j) &\equiv i(\text{ms}(t_i, t_j)) \\ &= -log(\min\{p(t)|t \in S(t_i, t_j)\}), \end{aligned} \quad (8)$$

where $S(t_i, t_j)$ is the subsumer set of term t_i and t_j, essentially being their common ancestor terms.

Similarity between GO Annotations of Two Genes: While Equation 8 measures the semantic similarity between two GO terms, it is common that a gene product may be annotated with multiple GO terms, which will lead to multiple term-to-term similarities between two genes. We adopted a simple yet commonly applied approach [15], [24], to induce the maximal term-to-term similarity as the similarity between the GO annotations of two genes. To formalize, suppose the multiple GO annotations of two genes products g_i and g_j are denoted as $A(g_i) = \{t_{i1}, t_{i2}, \ldots, t_{iP}\}$ and $A(g_j) = \{t_{j1}, t_{j2}, \ldots, t_{jQ}\}$ respectively, their similarity is then calculated as:

$$\text{sim}(A(g_i), A(g_j)) = \max\{sim(t_{ix}, t_{jy})|x \in [1, P], y \in [1, Q]\}. \quad (9)$$

By applying the maximal term-to-term similarity as the similarity between to GO annotations, we essentially identify their subsumer that has the maximal information content, i.e. maximal common factor.

Choice, Match and Learning Functions on GO Annotations: While Equation 9 effectively evaluates the maximal common factor of two genes' GO annotations, this equation is not normalized, in the sense that the similarity value may range from zero to infinity. It is inappropriate to apply this definition directly to the mixture model ARAM, because the calculation of Equation 1 may be dominated by the score produced from Equation 9, given that Equation 4 outputs a score in $[-1, 1]$ range. Inspired by the work of Jiang and Conrath [15] as well as the fuzzy ART paradigm [6], we calculate the choice and match scores by applying different aspects of normalization to Equation 8. That is,

$$T_b(\mathbf{I}_b, \mathbf{w}^b) = \frac{\text{sim}(\mathbf{I}_b, \mathbf{w}^b)}{\alpha + i(\mathbf{w}^b)}, \quad (10)$$

and

$$M_b(\mathbf{I}_b, \mathbf{w}^b) = \frac{\text{sim}(\mathbf{I}_b, \mathbf{w}^b)}{\alpha + i(\mathbf{I}_b)}, \quad (11)$$

where sim(.) is given by Equation 9, $i(.)$ is given by Equation 6, and α is a small positive constant to prevent zero division. These definitions re-scale the choice and match scores to [0, 1] as the information content of a term's subsumer is always less than or equal to the term's information content.

With respect to the learning of GO annotation, we understand this process as the representation of the maximal common factor among all inputs being grouped into the same category. This idea harmonizes the definition of the minimal subsumer. Thus, we have a straight forward definition of the learning function:

$$\mathbf{w}^{'b} = L_b(\mathbf{I}_b, \mathbf{w}^b) = \text{ms}(\mathbf{I}_b, \mathbf{w}^b), \quad (12)$$

where the identification of the minimal subsumer ms(.) is given by Equation 8.

Equations 4 through 12 complete our construction of the mixture model ARAM network.

D. Summary of Network Parameters

This section briefly summarizes the parameters in the proposed algorithm. In general, the network's learning is controlled with the associative contribution parameter $\gamma \in [0, 1]$ (Equation 1), the vigilance thresholds $\rho_a \in [-1, 1]$ and $\rho_b \in [0, 1]$ (Equation 2), and the learning rate $\eta \in [0, 1]$ (Equation 5). As to the parameter α in Equations 10 and 11, it may be built in with a fixed small positive value (such as 1e-8).

γ decides the weights of the pattern fields during evaluation of overall pattern similarities. Particularly, $\gamma = 0.5$ gives equal weights to expression and GO annotation. As reviewed in Section II-B, ρ_a and ρ_b mainly decide the group size as well as the total number of groups over all inputs. Higher vigilance thresholds lead to a larger number of smaller groups. Readers should note that while $\rho_a \in [-1, 1]$ according to the range of the Pearson correlation coefficient (Equation 4), in practice, we use a positive ρ_a setting as we want our recognition categories contain positively correlated expression patterns only. The learning rate η controls how fast the recognition pattern adapts itself towards the new input knowledge. It should be noted that, as studied by Bottou et al.[2], [12], a constantly too high learning rate may cause network oscillation on densely distributed input data. It has been a common practice to initialize the learning with relatively low value (such as 0.1) and to gradually reduce it while the learning proceeds.

III. Experiment

We applied the proposed mixture model ARAM neural network to the genome-wide analysis of the budding yeast (*Saccharomyces cerevisiae*) data. The purpose of our experiment is to evaluate and validate the significant gene functional grouping generated by the proposed algorithm, through comparison with results from well documented studies. The details of our experiment are reported below.

A. Datasets and Pre-Processing

The yeast gene expressions provided by Eisen et al. [11] had been extensively studied in the literature. The so-called *Public Microarray Expression Data* (http://rana.lbl.gov/EisenData.htm) contains the expression profiles of 6221 genes labeled with the corresponding open frame reading (ORF) IDs. Each expression profile, maximally eighty-dimensional, consists of an aggregation of data from multiple experiments including time courses of the mitotic cell division cycle, sporulation, the diauxic shift, and responses to different shocks etc. [11]. The expressions had been normalized by Eisen et al. and hence were used in our experiment without alternation.

To facilitate our validation of gene functional groups, our experiment used the expressions of those ORFs which are annotated with known gene IDs. Approximately half of the 6221 ORF IDs are annotated with gene IDs and functional descriptions. We searched the *Saccharomyces* genome database (SGD, http://www.yeastgenome.org) with the list of gene IDs and downloaded their GO annotations in batch (http://db.yeastgenome.org/cgi-bin/batchDownload). This consolidated into a list of 3088 genes, with corresponding gene ID, functional description, expression, and GO annotation. Furthermore, in view of the three independent, non-intersecting categories of ontologies in the same GO infrastructure, namely *Biological Process*, *Cellular Component* and *Molecular Function*, and the understanding that the Biological Process ontology is mostly related to functional categorization, we limit our study within this category only. This further reduced the number of genes being tested in our experiment to 2974. In addition, noting that there are two major types of relations between GO terms, i.e. *is-a* and *part-of*, for the simplicity of analysis, we followed Lord et al.'s practice [20] to treat them equivalent to each other and consolidated GO into a uniform *is-a* taxonomy.

B. Results and Discussions

We applied the proposed algorithm on the 2974 data records. All inputs were randomly shuffled in presentation order and sent to the mixture model ARAM for batch training. In each learning iteration the input-category mapping was tracked and compared to the mapping of last iteration to calculate the prediction (i.e. category assignment) error rate. Learning of the network stopped when the prediction error rate was below 1%, or after 50 learning iterations, whichever was sooner. We adopted the default $\gamma = 0.5$ parameter for pattern association. The learning rate η was initialized with 0.1 and was linearly decreased by 10% in each new learning iteration once the prediction error rate was below 20%. By fine-tuning ρ_a and ρ_b thresholds, we were able to obtain different groupings over the 2974 inputs.

The network stabilized after 9 learning iterations with settings of $\rho_a = 0.3$ and $\rho_b = 0.2$, and generated 262 recognition categories. Among them, 120 categories were relatively small, in the sense that each of them contained less than 5 genes. This is however of no surprise to us, considering the diversity of

the GO annotations and gene expressions. On the other hand, the 27 largest categories, each containing 30 or more genes are of our major interest, as they grouped 1151, over 38% of all inputs and thus reflected the significant genome-wide patterns. In order to validate the discovered functional grouping, we compared the representative GO annotation on each group against SGD functional descriptions of its member genes – Readers shall note that the knowledge on SGD functional description was not used to train the network. In addition, we plotted the gene expressions of each group's member genes to validate their correlation. Our inspection discovered that these categories had given very satisfactory results: each recognition category had successfully grouped a number of significantly correlated genes, in terms of both expression profile and GO annotation. In addition, the GO annotation had shown close correlation to SGD functional descriptions. A few categories are depicted in Tables I - V for extended discussions. Each table illustrates the number of genes being clustered in the category, the representative GO annotation of the category, as well as an overview of the the expression profiles (each in different color), the IDs and the SGD functional descriptions of the member genes. Due to space constraint, the different series are not labeled on the X-axis and the expression profiles are not individually labeled.

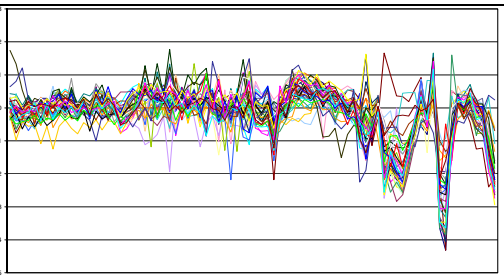
Category 10 and category 29 caught our first attention. It has been clearly shown that almost all genes grouped under these two categories are related to protein synthesis. Particularly, 76 of them are protein synthesis ribosomal proteins, covering around 44% of all (177) known protein synthesis ribosomal proteins over the full genome. We are amazed with the highly correlated expression patterns they had shown, which reflected the nature of their highly conserved functions and were successfully captured by our experiment. Interestingly, while genes from the two categories had nearly identical expression patterns and functions, two different GO terms, i.e. GO:0016043 (cell organization and biogenesis) and GO:0043170 (macromolecule metabolism) were found over the two categories. On the other hand, based on our understanding, both GO terms sound very appropriate on these genes, as they were based on different aspects of the protein synthesis process (cellular process and metabolism respectively). This reflects the inherent variety of GO annotations and the somewhat subjectivity over GO term definition.

On the other hand, category 11 provides a different view on the annotation power of GO terms. The 65 genes grouped in this category had shown high correlated expressions, suggesting closely related functions. Their SGD functional descriptions referred to a variety of sub-functions during different phases of cell cycle. If merely based on SGD functional descriptions, it is not easy for a computational program or even a human to group them together, without strong biological domain knowledge. However, the organized GO hierarchy well encoded the semantic relationship among their functions. Through adaptive learning, our experiment successfully discovered this significant functional group and annotated them with an relatively general, yet appropriate GO subsumer term, GO:0016043 (cell organization and biogenesis). This category particularly reflects the advantage of GO annotation over SGD's natural language type functional description, as well

TABLE I

CATEGORY 10 OF THE MIXTURE ARAM OUTPUT OVER THE YEAST DATASET, WHICH IDENTIFIES 40 PROTEIN SYNTHESIS RELATED GENES.

Category #10 (40 genes) **GO:0016043** "cell organization and biogenesis"

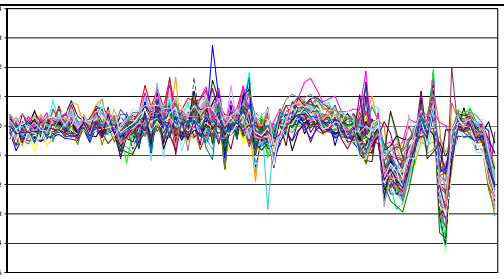


RPL1B, RPL3, RPL5, RPL6B, RPL10, RPL11A, RPL11B, RPL12A, RPL12B, RPL20B, RPL25, RPL34B, RPP0, RPP1A, RPS0A, RPS0B, RPS4A, RPS4B, RPS11A, RPS14A, RPS14B, RPS15, RPS16B, RPS19A, RPS19B, RPS21A, RPS22A, RPS23A, RPS23B, RPS27B, RPS28B, RPS31 ("protein synthesis ribosomal protein")
TIF1 ("protein synthesis translation initiation factor eIF4A")
NCA3 ("ATP synthesis regulates expression of F0F1 ATPase subunits")
GBP2, HMO1, ILV5, STE20, STM1, VMA2 (other related annotations, not listed here due to page constraint)

TABLE II

CATEGORY 29 OF THE MIXTURE ARAM OUTPUT OVER THE YEAST DATASET, WHICH IDENTIFIES 56 PROTEIN SYNTHESIS RELATED GENES THAT HAVE SIMILAR EXPRESSION PROFILES OF THOSE OF TABLE I BUT ARE ANNOTATED WITH A DIFFERENT GO TERM.

Category #29 (56 genes) **GO:0043170** "macromolecule metabolism"

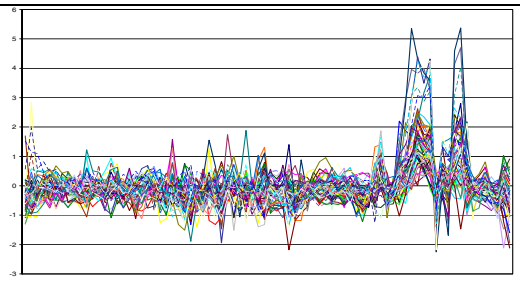


MRPL11, RPL1A, RPL2B, RPL4A, RPL8B, RPL15A, RPL16B, RPL18A, RPL18B, RPL20A, RPL23B, RPL27A, RPL27B, RPL30, RPL31A, RPL32, RPL33A, RPL35A, RPL35B, RPL37B, RPL40A, RPL40B, RPL42A, RPL42B, RPL43A, RPP1B, RPP2A, RPP2B, RPS5, RPS6A, RPS7A, RPS7B, RPS8A, RPS10A, RPS12, RPS13, RPS21B, RPS22B, RPS24A, RPS24B, RPS25B, RPS28A, RPS29A, RPS29B ("protein synthesis ribosomal protein")
ANB1, HYP2, PRT1, TIF2, TIF5 ("protein synthesis translation initiation factor")
EFB1, EFT1, EFT2 ("protein synthesis translation elongation factor")
FUN12, RKI1, SHR3, TIF4631 (other related)

TABLE III

CATEGORY 11 OF THE MIXTURE ARAM OUTPUT OVER THE YEAST DATASET, WHICH IDENTIFIES 65 CELL CYCLE RELATED GENES.

Category #11 (65 genes) **GO:0016043** "cell organization and biogenesis"

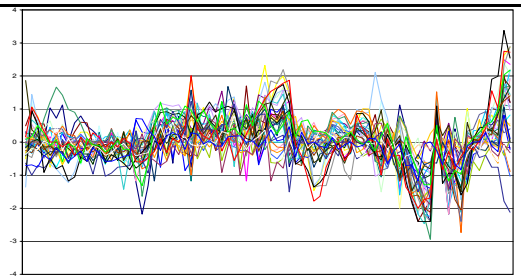


CNM67, NUF2, SPC19, SPC42, SPC98, STU2 ("cytoskeleton spindle pole body component")
ECM9, ECM11, GFA1, SMI1 ("cell wall biogenesis")
AXL1, BUD5 ("bud site selection")
NUP157, POM152 ("nuclear protein targeting nuclear pore protein")
FHL1, RAP1, RTT102, SET1, SRB5 ("transcription" related)
HST1, IRS4, RIF1, SAS3, SIR1, SIR4 ("silencing" related)
SEC72, TIP20, YKT6 ("secretion" related)
BBP1, BIM1, CDC3, DIG2, GAL83, GLE2, GTR2, ISA1, KAR3, MSB1, MSS4, NFI1, PAC11, PEA2, PEX2, PEX13, PKH2, PLC1, PMI40, PTC1, REF2, RNP1, SPA2, STN1, STU1, SUR4, TBF1, TID3, UBC9, UME6, VAM7, VIK1, VPS9, VPS30, WHI4, YRA1, YTA7 (other related)

TABLE IV

CATEGORY 12 OF THE MIXTURE ARAM OUTPUT OVER THE YEAST DATASET, WHICH IDENTIFIES 36 ENERGY TRANSPORTATION RELATED GENES.

Category #12 (36 genes) **GO:0006091** "generation of precursor metabolites and energy"

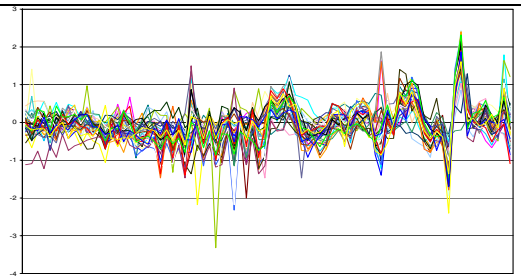


ADH1, ADH3, ADH4 ("glycolysis alcohol dehydrogenase")
COX4, COX8, COX13, PPA2, QCR2, QCR8, QCR9 ("oxidative phosphorylation" related)
ATP1, ATP4, ATP5, ATP14, ATP17 ("ATP synthesis" related)
GLG1, GSY2, GSY1 ("glycogen metabolism" related)
FUM1, IDH1, IDH2, MDH3, LSC1, LSC2 ("TCA cycle" related)
CBR1, COX14, CYB2, CYB5, GCR2, GLC3, JAC1, MCR1, MDL2, NCP1, TPS2, TSL1 (other related)

TABLE V

CATEGORY 35 OF THE MIXTURE ARAM OUTPUT OVER THE YEAST DATASET, WHICH IDENTIFIES 37 PROTEIN DEGRADATION AND FOLDING RELATED GENES.

Category #35 (37 genes) **GO:0043170** "macromolecule metabolism"



PRE1, PRE2, PRE4, PRE3, PRE5, PRE6, PRE7, PRE8, PRE9, PRE10, RPN10, RPT3, PUP1, PUP2, SCL1 ("protein degradation proteasome subunit")
RPN6, RPN7, RPN9, RPN12, RPT4, RPT5, RPT6 ("protein degradation proteasome regulatory subunit")
RPT1, UFD1, UMP1 ("protein degradation, ubiquitin")
CCT3, CCT8 ("protein folding cytoplasmic chaperonin complex")
BDF1, CDC48, HAP2, MPD2, PIB1, RBK1, RPN2, RPN11, SBA1, VPS45 (other related)

as the prediction power of the proposed algorithm.

Besides the observed generalization of sub-functions, the outputs of mixture model ARAM had also shown satisfactory specialization. For example, category 12 successfully identified 36 genes with energy transportation related functions in common, as well as strongly correlated expressions. The annotation with GO:0006091 (generation of precursor metabolites and energy) satisfactorily summarized the nature of their functionality. Moreover, category 35 contained 37 genes related to protein degradation and folding, with common GO annotation GO:0043170 (macromolecule metabolism) as well as highly correlated expressions.

Due to page constraint, we are not able to elaborate all results across the 27 categories. The full results are downloadable via http://bioinfo.noble.org/manuscript-support/he07cibcb.

IV. CONCLUDING REMARKS AND FUTURE WORK

In this paper, we discussed our observation on the high demand raising from functional and comparative genomics studies, in terms of both computational GO annotation and clustering of biological gene expressions. We targeted on the difficulty over human inspection of joint GO annotation and gene expression data for the purpose of identifying genome-wide functional groupings, and proposed a novel artificial neural network to tackle this deficiency. Associative learning of GO and gene expression knowledge still remains a challenge to computational studies, as most existing algorithms work on isogenous data only. Our proposed mixture model ARAM network inherits the solid theory foundation of the ARAM algorithm, with a full set of re-defined similarity measures and learning functions to handle these heterogenous input patterns. We believe our proposed algorithm is one of the few, if not

first, computational approaches that fully integrate system-wide GO and gene expression data in a single infrastructure.

We applied the mixture model ARAM network to the *Saccharomyces cerevisiae* (yeast) genome data. In general, within each recognition category generated by the algorithm, genes showed significantly high correlation in both GO annotation and gene expression aspects. This shows that the design of the mixture model architecture has delivered solid results that meet our expectation. Our design is based on the assumption that genes with high correlated GO annotations (which could be computational) and expressions (biological) will have similar or closely related functions (biological). To validate this assumption, we further investigated the independent SGD functional descriptions on the genes. We discovered that in each category, there was a satisfactory overlap over the SGD gene functional descriptions, which additionally harmonized the network-learnt GO annotation over the category. This reflects the prediction power of our proposed approach in systematically discovering significant functional gene groups.

While our proposed algorithms worked well on the extensively-studied and well-annotated yeast genome, it would be interesting to see its performance on relatively new genomes, whose genes' functions are not confirmed or only computationally predictable. Our future work is to apply the algorithm to the *Medicago truncatula* genome with first-hand Affymetrix GeneChip (http://www.affymetrix.com) data for systematical study of gene functions. Our discoveries will be presented in future publications.

ACKNOWLEDGEMENTS

We are grateful to the valuable comments and suggestions from our colleague Dr. Haiquan Li and anonymous reviewers. We would like to acknowledge the Samuel Roberts Noble Foundation for the financial support to our research work.

REFERENCES

[1] A. Ben-Dor, R. Shamir, and Z. Yakhini. Clustering gene expression patterns. *Journal of Computational Biology*, 6(3/4):281–297, 1999.

[2] L. Bottou and Y. Bengio. Convergence properties of the K-Means algorithms. In G. Tesauro, D. Touretzky, and T.K. Leen, editors, *Advances in Neural Information Processing System 7*. MIT Press, 1995.

[3] G.A. Carpenter and S. Grossberg. A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image processing*, 34:54–115, 1987.

[4] G.A. Carpenter, S. Grossberg, and J.H. Reynolds. ARTMAP: Supervised real-time learning and classification of nonstationary data by self-organizing neural network. *Neural Networks*, 4:565–588, 1991.

[5] G.A. Carpenter, S. Grossberg, and D.B. Rosen. ART 2-A: An adaptive resonance algorithm for rapid category learning and recognition. *Neural Networks*, 4:493–504, 1991.

[6] G.A. Carpenter, S. Grossberg, and D.B. Rosen. Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, 4:759–771, 1991.

[7] Mark Chee, Robert Yang, Earl Hubbell, Anthony Berno, Xiaohua C. Huang, David Stern, Jim Winkler, David J. Lockhart, Macdonald S. Morris, and Stephen P. A. Fodor. Accessing genetic information with high-density DNA arrays. *Science*, 274(5287):610–614, 1995.

[8] The Gene Ontology Consortium. Gene ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29, 2000.

[9] Sandrine Dudoit, Yee Hwa Yang, Matthew J. Callow, and Terence P. Speed. Statistical methods for identifying differentially expressed genes in replicated cdna microarray experiments. Technical Report 578, Department of Biochemistry, Stanford University School of Medicine, August 2000.

[10] M. Eisen, P.T. Spellman, D. Botstein, and P.O. Brown. Cluster analysis and display of genome-wide expression patterns. In *Proceedings of National Academy of Science USA*, volume 95, pages 14863–14867, 1998.

[11] Michael B. Eisen, Paul T. Spellman, Patrick O. Browndagger, and David Botstein. Cluster analysis and display of genome-wide expression patterns. In *Proceedings of the National Academy of Sciences of the United State of America*, volume 95, pages 14863–14868, 1998.

[12] S. Grossberg. *Studies of Mind and Brain*. D. Reidel Publishing, 1982.

[13] Erez Hartuv, Armin Schmitt, Jorg Lange, Sebastian Meier-Ewert, Hans Lehrachs, and Ron Shamir. An algorithm for clustering cDNAs for gene expression analysis. In *RECOMB*, pages 188–197, 1999.

[14] Ji He, Ah-Hwee Tan, and Chew-Lim Tan. On machine learning methods for chinese document classification. *Applied Intelligence*, 18:311–322, 2003.

[15] Jay J. Jiang and David W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of International Conference Research on Computational Linguistics (ROCLING)*, Taiwan, 1997.

[16] Ryung S. Kim, Hongkai Ji, and Wing H. Wong. An improved distance measure between the expression profiles linking co-expression and co-regulation in mouse. *BMC Bioinformatics*, 7(44), 2006.

[17] Anand Kumar, Barry Smith, and Christian Borgelt. Dependence relationships between gene ontology terms based on TIGR gene product annotations. In *3rd International Workshop on Computational Terminology*, pages 31–38, 2004.

[18] Sung Geun Lee, Jung Uk Hur, and Yang Seok Kim. A graph-theoretic modeling on GO space for biological interpretation of gene clusters. *Bioinformatics*, 20(3):381–388, 2004.

[19] P. W. Lord, R. D. Stevens, A. Brass, and C. A. Goble. Investigating semantic similarity measures across the gene ontology: the relationship between sequence and annotation. *Bioinformatics*, 19(10):1275 – 1283, July 2003.

[20] P.W. Lord, R.D. Stevens, A. Brass, and C.A. Goble. Semantic similarity measures as tools for exploring the gene ontology. In *Pacific Symposium on Biocomputing*, pages 601–12, 2003.

[21] Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 448–453, 1995.

[22] S. Ross. *A First Course in Probability*. Macmillan, 1976.

[23] Jose L. Sevilla, Victor Segura, Adam Podhorski, Elizabeth Guruceaga, Jose M. Mato, Luis A. Martinez-Cruz, Fernando J. Corrales, and Angel Rubio. Correlation between gene expression and GO semantic similarity. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2(4):330–338, 2005.

[24] Nora Speer, Holger Frohlich, Christian Spieth, and Andreas Zell. Functional grouping of genes using spectral clustering and gene ontology. In *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN)*, pages 298–303. IEEE Press, 2005.

[25] P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E.S. Lander, and T.R. Golub. Interpreting patterns of gene expression with self-organizing maps: Methods and application to homatopoietc differentiation. In *Proceedings of the National Academy of Science*, volume 96, pages 2907–2912, 1999.

[26] A.H. Tan. Adaptive Resonance Associative Map. *Neural Networks*, 8(3):437–446, 1995.

[27] Ah-Hwee Tan. Predictive self-organizing networks for text categorization. In *Proceedings of the Pacific-Aisa Conference on Knowledge Discovery and Data Mining (PAKDD)*, pages 66–77, 2001.

[28] Ah-Hwee Tan, Hwee-Leng Ong, Hong Pan, Jamie Ng, and Qiu-Xiang Li. Towards personalized web intelligence. *Knowledge and Information Systems*, 6(5):595–616, 2004.

[29] Ah-Hwee Tan and Hong Pan. Predictive neural networks for gene expression data analysis. *Neural Networks*, 18(3):297–306, 2005.

[30] V.E. Velculescu, L. Zhang, B. Vogelstein, and K.W. Kinzler. Serial analysis of gene expression. *Science*, 270(5235):484–487, 1995.

[31] Marc Vidal. Interactome networks. In *Proceedings of the Annual Meetings of the American Society of Plant Biologists (ASPB)*, 2006.