

# MIXTURE MODEL ADAPTIVE NEURAL NETWORK FOR MINING GENE FUNCTIONAL PATTERNS FROM HETEROGENEOUS KNOWLEDGE DOMAINS

Ji HE, Xinbin DAI, Patrick Xuechun ZHAO \*

*Bioinformatics Laboratory, Plant Biology Division, The Samuel  
Roberts Noble Foundation, 2510 Sam Noble Parkway, Ardmore, OK  
73401, USA*

E-mail: {jhe,xdai,pzhao}@noble.org

## Abstract

Gene Ontology (GO) annotation and gene expression profiling have been two major approaches for system-wide analysis of gene functions. Current high-throughput sequence alignment and microarray technologies produce large volumes of noisy data. In the literature, numerous clustering methods have been studied for discovery of gene functional grouping based on either approach. But there is a lack of algorithm that intelligently mines gene patterns across these two problem domains. This paper presents a mixture model associative artificial neural network that integrates these heterogeneous domain knowledge for discovery of genome-wide functional patterns. The algorithm inherits the theoretical foundation of Adaptive Resonance Associative Map (ARAM), with essential redefinition of pattern similarity measures and learning functions. The algorithm's efficacy was evaluated on the *Saccharomyces cerevisiae* (yeast) genome. Our controlled experiments showed that association of these domain knowledge reduces analytical noises and produces a more meaningful functional grouping.

**Keywords:** Associative clustering, Artificial neural network, Mixture Model Adaptive Resonance Associative Map, Gene Ontology, Gene expression.

---

\*Author of correspondence.

# 1 Introduction

Gene function is one of the central topics of functional genomics and comparative genomics. Understanding how genes of particular interest function and interact to each other is critical to numerous biomedical studies such as disease control and prevention. The advance of various high-throughput experimental technologies (such as retrotransposon-induced mutation, genome-wide sequencing and microarray transcription profiling) has lead biology into a post-genome era, in which researchers' vision is no longer limited to a few genes, but rather large volume of genes in a more systematic fashion. These high-throughput experiments usually generate large-scale, yet often noisy, heterogenesis data, allowing biologists to investigate the characteristics of numerous genes in different angles. Yet how to intelligently and effectively integrate these data for accurate gene functional analysis remains a challenge to computational science researchers.

Sequence data is the primary source, and essential information for understanding gene functions. Given a collection of sequences with unknown functions, biologists often predict their functions through various approaches based on sequence alignment. Programs such as BLAST [1] and HMMER [10] have been the *de facto* standards routinely used in numerous laboratories. It is a common practice to infer and annotate the function of an unknown sequence based on one or multiple significantly homolog sequences with known functions. However, in the past, gene functions are usually annotated with free text, which is often subjective, lacks formalization, and is difficult to understand and further process without expert knowledge in the particular domain. 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>) [36] is one of the protocols being widely adopted for this purpose. GO provides a set of well defined annotation terms organized by means of a directed acyclic graph (DAG). Studies have shown that GO annotation generally conforms with other sequence homology based annotation paradigms such as TIGR's [21, 23]. Annotating gene functions in the ontology space greatly facilitates data manipulation and makes it possible to analyze large gene set in a quantitative manner. Methods for functional categorization based on GO have been extensively documented in the literature [21, 22, 23, 24, 30]. 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 2.2.

The assumption behind homology-based sequence annotation is that, genes with common structure due to shared ancestry usually perform the same function. However, it is challenging to programmatically detect gene homology based on limited sequence information. In practice, two genes with high sequence-level similarity, as indicated by an alignment program, are assumed as homolog genes. This is however not necessarily true. As a matter of fact, in many cases, a single nucleotide mutation of a gene may silent (deactivate), or potentially alters its function. Therefore, algorithms based on sequence alignment, while practically providing a quick survey on the potential functions of a candidate gene, often introduces a high noise level to functional annotation.

Normally, a gene participates a particular biological process in form of the corresponding functional protein. Gene expression profiling, as a quantitative measurement on the intensity of the process of a gene's DNA sequences being converted into proteins under different conditions (in different organisms, at different time courses, or under different treatments, etc.) is widely adopted to investigate and verify gene functions in a more biological fashion. The spread of genome-wide microarray technologies [8, 38] has made it possible to obtain large scale gene expression data in a short time frame. A variety of clustering methods have been applied to analysis of large scale gene expression data [2, 9, 11, 13, 20, 31]. By grouping genes with similar expressions together, clustering-based software provide an overview of the whole genome's transcription profile, and reduce human labor in investigating individual genes' expressions.

Experiments have shown that genes with same or closely related functions may have highly correlated expression patterns. It is however not true in the reverse direction. In other words, genes with highly correlated expression patterns are not necessarily with similar functions. For example, genes that carry out different functions in a same pathway may show highly correlated expressions. In addition, transcription factors, a particular category of genes that regulate the transcription of other genes (their target genes), will have similar expression patterns as those of their target genes; but they are not necessarily performing the same function. Therefore, the practice of inferring functions from expression patterns always suffers from noises, and often requires human inspection and verification using other domain knowledge.

Understanding the noises in these data, researchers tend to combine the analytical results from both sequence alignment and microarray experiments for a better understanding of gene function. 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, and further investigate the

functional annotation of individual genes in each category; or to limit the study to a set of genes annotated with functions of interest, and study their expression patterns for further confirmation. However, either way has proven to be human labor intensive, and rather critically, difficult to navigate genome-wide data in a systematic manner. Therefore, an intelligent system that automatically identify gene functional patterns based on knowledge across these data domains is of high demand in real-life studies.

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. Hvidsten et al. [17] applied rough set theory to mine the association rules between microarray data and GO annotation on biology process. Association rules however are known to represent only very significant patterns and lacks the coverage of the full genome. The FUNC software by Prufer et al. [26] also detects significant associations between gene expressions and GO terms. Its implementation is based on statistical evaluation of multiple trial cluster membership assignments, which does not involve much machine intelligence and could be time consuming on large input set. Recent updates of some commercial software (such as GeneSpring (<http://www.agilent.com>) and Spotfire (<http://www.spotfire.com>)) 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 genome-wide gene patterns.

To fill this gap, this paper presents a novel mixture model artificial neural network (ANN) for discovery of gene patterns based on heterogenesis knowledge from both sequence alignment and gene expression profiling experiments. The proposed algorithm associatively clusters input genes in terms of similarities in *both* GO annotation and gene expression domains. We evaluated the algorithm's efficacy on the public *Saccharomyces cerevisiae* (yeast) genome and found the proposed algorithm's output clusters significantly represented gene functional groupings.

The rest of this paper is organized as follows. Section 2 introduces our proposed algorithm in detail. Section 3 reports our experiment on the yeast genome. Section 4 summarizes our conclusion and proposes future work.

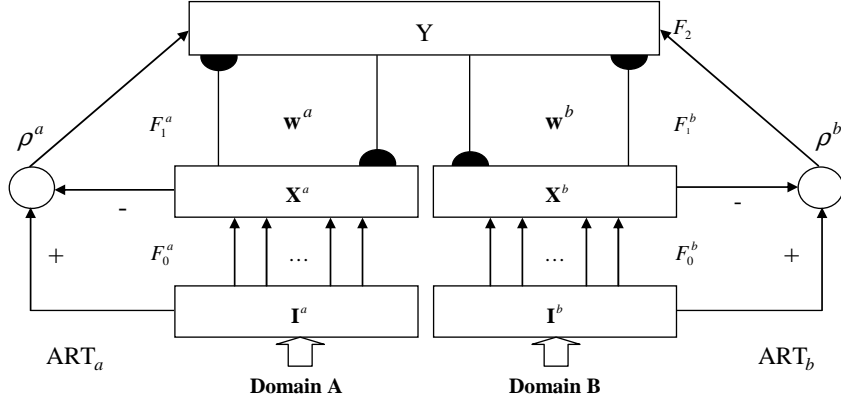
## 2 Method

Clustering refers to the task of partitioning unlabeled data into meaningful groups (clusters), in the sense that data samples in each group are more similar and/or more closely related to each other than to samples in other groups [18]. Numerous clustering algorithms represent the input samples in a vector format, and quantitatively measure the meaningfulness of the clustering process with a (dis)similarity function. With the vector representation, each input is mapped into a data point in a multi-dimensional, orthogonal space. That is, the features used to quantize input samples are from the same domain and are assumed to be independent to each other. However, GO annotation and gene expression data are provided in the nature that, they represent the measurements of the genes from *two* heterogenesis domains, the earlier being descriptive while the latter being quantitative. It is theoretically challenging to represent them into a *single* feature space as most clustering methods require. To handle these input data, our study focuses on an modular artificial neural network architecture that is capable of handling inputs from different knowledge domains, briefly reviewed below.

### 2.1 Adaptive Resonance Associative Map (ARAM)

The Adaptive Resonance Associative Map (ARAM) [32] belongs to the family of Adaptive Resonance Theory (ART) self-organizing neural networks [4]. Like another member of the family, ART-MAP [7], ARAM is capable of incrementally learning recognition categories (pattern classes) and multi-dimensional maps of patterns. 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 input signals  $\mathbf{I}^a$  and  $\mathbf{I}^b$  from its independent input layers  $F_0^a$  and  $F_0^b$  respectively, and a shared long term memory (LTM) layer  $F_2$  that encodes the associated knowledge from the two feature fields. The learning of the network is guided by an orienting subsystem with two logical gates, defined with two vigilance parameters ( $\rho^a$  and  $\rho^b$  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 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).



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

The ARAM learning paradigm has been comprehensively documented in the literature [32] and is summarized below for a better understanding of this paper.

*Inputs and Recognition Categories:* ARAM requires inputs  $\mathbf{I}^a$  and  $\mathbf{I}^b$  represented in vector format. There is a built-in normalization link between the input layer  $F_0$  and the STM layer  $F_1$ , denoted as  $\mathbf{x}^a = \mathfrak{R}(\mathbf{I}^a)$  and  $\mathbf{x}^b = \mathfrak{R}(\mathbf{I}^b)$ . 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.  $\mathbf{w}_j = (\mathbf{w}_j^a | \mathbf{w}_j^b)$ ,  $\mathbf{w}_j^a$  and  $\mathbf{w}_j^b$  being same dimensional as  $\mathbf{x}^a$  and  $\mathbf{x}^b$  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.

*Category Competition:* In response to the normalized input signal  $\mathbf{x} = (\mathbf{x}^a | \mathbf{x}^b)$ , the similarity between the input and each LTM recognition category  $j$  is evaluated according to

$$T(\mathbf{x}, \mathbf{w}_j) = \gamma \cdot T^a(\mathbf{x}^a, \mathbf{w}_j^a) + (1 - \gamma) \cdot T^b(\mathbf{x}^b, \mathbf{w}_j^b), \quad (1)$$

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

<sup>1</sup>With an assumption that a higher choice score indicates a higher pattern similarity. Otherwise the lowest if choice functions are dissimilarity/distance-based.

*Resonance or Reset:* If the competition generates a winner category  $J$ , its similarity to the input  $\mathbf{x}$  is further confirmed in domain spaces  $A$  and  $B$  individually, using another set of *match functions*, i.e.  $M^a(\mathbf{x}^a, \mathbf{w}_J^a)$  and  $M^b(\mathbf{x}^b, \mathbf{w}_J^b)$ . The network is said to reach *resonance* if both match scores are over<sup>2</sup> the corresponding *vigilance* thresholds  $\rho^a$  and  $\rho^b$ , denoted as

$$\begin{cases} M^a(\mathbf{x}^a, \mathbf{w}_J^a) \geq \rho^a & \text{and} \\ M^b(\mathbf{x}^b, \mathbf{w}_J^b) \geq \rho^b, \end{cases} \quad (2)$$

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.

*Network Learning:* Once the search ends and network resonance is achieved, the weight vector  $\mathbf{w}_J$  was updated to incorporate the input knowledge correspondingly from field  $A$  and  $B$ , according to two *learning functions*:

$$\begin{cases} \mathbf{w}_J^{\prime a} = L^a(\mathbf{x}^a, \mathbf{w}_J^a), & \text{and} \\ \mathbf{w}_J^{\prime b} = L^b(\mathbf{x}^b, \mathbf{w}_J^b). \end{cases} \quad (3)$$

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,  $\mathbf{w}_{new}^{\prime a} = \mathbf{x}^a$  and  $\mathbf{w}_{new}^{\prime b} = \mathbf{x}^b$ .

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, 33], clustering and personalized knowledge management [34], and association rule mining [35].

---

<sup>2</sup>With an assumption that a higher match score indicates a higher pattern similarity.

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 complementary 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 that is capable of handling 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. They however do not fit in our application.

Based on this understanding, we borrowed ARAM’s architecture and learning process which have well established theoretical foundation, and redefined the 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.

## 2.2 Mixture Model ARAM for GO Annotation and Gene Expression Data

Our application of the Mixture Model ARAM is straightforward: for each gene product, we use ARAM’s pattern field  $A$  to encode its expression profile and  $B$  to encode its GO annotation. Following common practices, the expression profile is presented in vector format, and the GO annotation is presented as a set of descriptive GO terms, denoted as  $\mathbf{x} = (\mathbf{x}^a | \mathbf{x}^b) = (\overline{\text{exp}} | \{\text{go terms}\})$ . 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. The Mixture Model ARAM’s learning activity in each pattern field is defined in the rest of this subsection.

### 2.2.1 Pattern Field for Gene Expression

Appropriate normalization of gene expression data prevents category proliferation without compromising their biological representation. Variations of normalization techniques are adopted in different experiments and are under continuous review. Thus the Mixture Model ARAM network does not contain a fixed normalization link to alter the original input gene expression. Instead,

we assume all input expressions are properly pre-normalized, and define the link between  $F_0^a$  and  $F_1^b$  as a simple feed forward copy operation, i.e.  $\mathbf{x}^a = \mathbf{I}^a$ .

Like the ART-2A [5] model, 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:

$$T^a(\mathbf{x}^a, \mathbf{w}^a) = M^a(\mathbf{x}^a, \mathbf{w}^a) = \frac{(\mathbf{x}^a - E(\mathbf{x}^a)) \cdot (\mathbf{w}^a - E(\mathbf{w}^a))}{\|\mathbf{x}^a - E(\mathbf{x}^a)\| \cdot \|\mathbf{w}^a - E(\mathbf{w}^a)\|} \quad (4)$$

where  $E(\cdot)$  and  $\|\cdot\|$  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:

$$\mathbf{w}'^a = L^a(\mathbf{x}^a, \mathbf{w}^a) = \mathbf{w}^a + \eta \cdot (\mathbf{x}^a - \mathbf{w}^a) \quad (5)$$

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.

## 2.2.2 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 [27] compared different semantic similarity measures against human judgements. He reported that in the controlled taxonomy, Information Content [28] based measurement outperformed two other measures, namely Edge Counting and Probability. Sevilla et al.'s study [29] 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 [28]. 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

$$\mathfrak{I}(c) \equiv -\log(p(c)) = -\log\left(\frac{f(c)}{N}\right), \quad (6)$$

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

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

where  $C(t)$  is the set of terms being subsumed by  $t$ , and  $\text{root}$  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 [27] 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  $\nabla(t_i, t_j)$ , is the subsumer that has the minimal likelihood (and hence maximal information content). To formalize:

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

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 [19, 30], 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  $A_i = \{t_{i1}, t_{i2}, \dots, t_{iP}\}$  and  $A_j = \{t_{j1}, t_{j2}, \dots, t_{jQ}\}$  respectively, their similarity is then calculated as:

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

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. the maximal common factor.

*Choice, Match and Learning Functions on GO Annotations:* Although 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 [19] 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,

$$T^b(\mathbf{x}^b, \mathbf{w}^b) = \frac{\text{sim}(\mathbf{x}^b, \mathbf{w}^b)}{\alpha + \mathfrak{I}(\bar{i}(\mathbf{w}^b))}, \quad (10)$$

and

$$M^b(\mathbf{x}^b, \mathbf{w}^b) = \frac{\text{sim}(\mathbf{x}^b, \mathbf{w}^b)}{\alpha + \mathfrak{I}(\bar{i}(\mathbf{x}^b))}, \quad (11)$$

where  $\text{sim}(\cdot)$  is given by Equation 9,  $\bar{i}(\mathbf{w}^b)$  and  $\bar{i}(\mathbf{x}^b)$  are the corresponding GO terms selected from  $\mathbf{w}^b$  and  $\mathbf{x}^b$  which have the highest term-to-term similarity for the calculation of  $\text{sim}(\cdot)$ ,  $\mathfrak{I}(\cdot)$  is given by Equation 6, and  $\alpha$  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.

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:

$$\mathbf{w}'^b = L^b(\mathbf{x}^b, \mathbf{w}^b) = \nabla(\mathbf{x}^b, \mathbf{w}^b), \quad (12)$$

where the identification of the minimal subsumer  $\text{ms}(\cdot)$  is given by Equation 8.

Equations 4 through 12 complete our definition of the Mixture Model ARAM network.

### 2.2.3 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  $\alpha$  in Equations 10 and 11, it may be built in with a fixed small positive value.

$\gamma$  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.  $\rho^a$  and  $\rho^b$  mainly affect 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  $\rho^a$  threshold as we want our recognition categories contain positively correlated expression patterns only. The learning rate  $\eta$  controls how fast the recognition pattern adapts itself towards the new input knowledge. It should be noted that, as studied by Bottou et al.[3, 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 small value (such as 0.1) and to gradually reduce it while the learning proceeds.

### 3 Experiment

We applied the proposed Mixture Model ARAM neural network to the analysis of the budding yeast (*Saccharomyces cerevisiae*) genome. The relative small size of the yeast genome enabled us to validate the algorithm’s efficacy through human inspection. The details of our experiment are reported below.

#### 3.1 The Yeast Genome Dataset

Our experiment adopted the yeast gene expression data provided by Eisen et al. (<http://rana.lbl.gov/EisenData.htm>) [11], which had been extensively studied in the literature. The public dataset contains genome-wide microarray 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., with a major purpose of studying yeast cell cycle [11].

To facilitate our validation of gene functional groups, our experiment used a subset of 3225 ORFs which are annotated with confirmed gene names and corresponding functional descriptions in free-text. To avoid category proliferation, we followed a common practice to normalize each gene’s expression

profile with the standard distribution normalization (also known as z-score normalization) [31], that is,

$$\mathbf{I}' = \frac{\mathbf{I} - \mathbf{E}(\mathbf{I})}{\sigma(\mathbf{I})}, \quad (13)$$

where  $\mathbf{E}(\cdot)$  and  $\sigma(\cdot)$  is the mean and standard deviation of the input. Readers should have noted this normalization is not part of the proposed Mixture Model ARAM network, but rather a data preprocessing before the microarray expressions were accepted by the network.

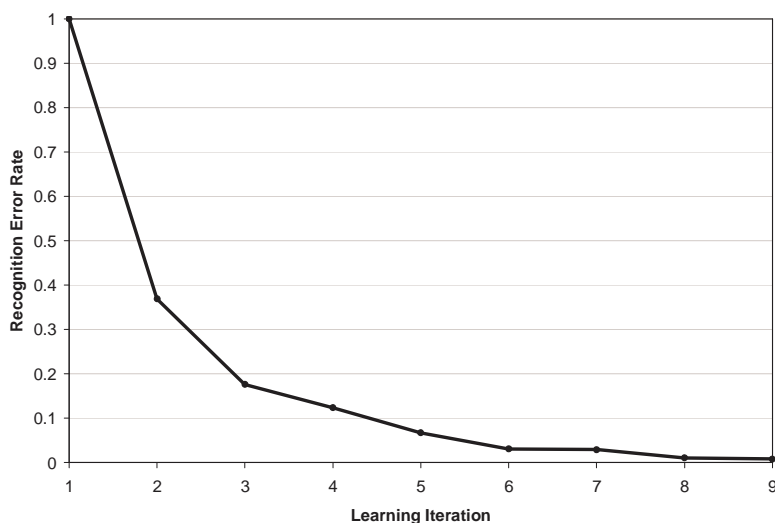
We downloaded these 3225 genes' GO annotations according to their gene names from the *Saccharomyces* genome database (SGD, <http://www.yeastgenome.org>). In view of the three independent, non-intersecting categories of ontologies in the same GO infrastructure, namely *Biological Process*, *Cellular Component* and *Molecular Function*, and the understanding that the Biological Process ontology is mostly related to functional categorization, we limited our study in this category only. 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 [24] to treat them equivalent to each other and consolidated GO into a uniform *is-a* taxonomy. Among the 3225 gene names, 3085 are annotated with at least one GO terms and 3080 are annotated with at least one Biological Process terms as of March 05, 2007.

This process generated a dataset of 3080 genes with corresponding gene name, functional description, microarray expression and GO annotation that was used in our experiment.

### 3.2 Parameter Settings

We applied the proposed algorithm on the above-mentioned dataset. 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 that 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  $\eta$  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 the  $\rho_a$  and  $\rho_b$  vigilance thresholds, we were able to obtain different gene groupings over the yeast genome.

The network converged at relatively fast speed regardless of the parameter tuning. The number of iterations ranged from 7 to 14 in our trials with different vigilance thresholds. This could be due to the inherited fast commitment



**Figure 2.** The recognition error rate of the Mixture Model ARAM after each learning iteration, with  $\rho^a = 0.3$  and  $\rho^b = 0.2$  on the yeast genome dataset. The error rate after the first iteration was considered as 1.0 because there was no previous cluster assignment information for comparison.

capacity of the ART-network which had been reported in prior studies [15]. Particularly, with settings of  $\rho^a = 0.3$  and  $\rho^b = 0.2$ , the network smoothly stabilized after 9 learning iterations (Figure 2), and generated 292 recognition categories. Analysis and evaluation of the results based on this set of parameter settings are reported below.

### 3.3 Results and Discussions

#### 3.3.1 Coverage of Genome-wide Functional Categories

*Small Number of Clusters Covered Majority of Full Genome:* Table 1 depicts the distribution of the output categories according to cluster size and the corresponding genome coverage. A first look at Mixture Model ARAM’s clustering result revealed that the size of the categories varied drastically. Among the 292 output categories, 190 were very small, in the sense that each of them grouped less than 10 genes. This is however not surprising to us, considering the facts that: 1) there are inherited noises in both the GO annotation and microarray expression data, 2) there is a high diversity of gene functions across the whole genome, and 3) some biological processes actually involves only few genes. Despite this, the remaining 102 categories had grouped a total of

2501 genes, which covered 81.20% of the whole genome dataset. We consider this a very good genome-wide representation, as it enables a biologist to review the majority of the genome data by looking into a manageable number of output categories (Figure 3).

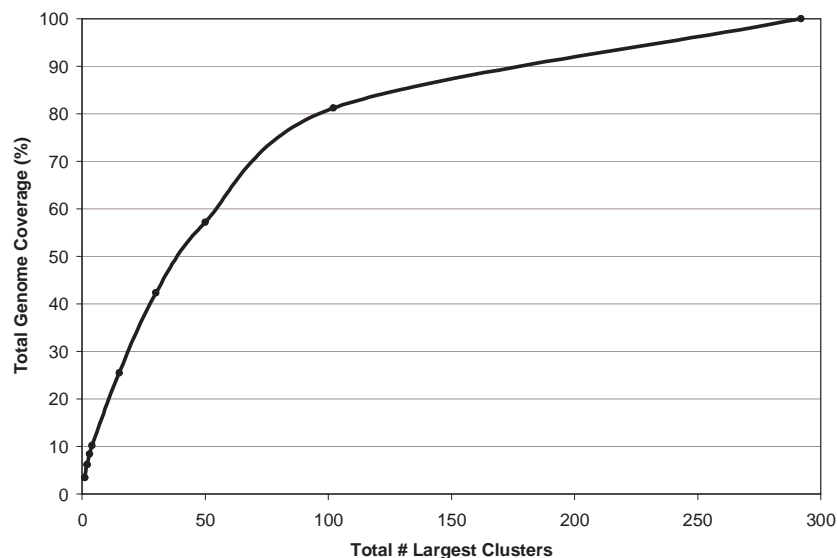
**Table 1.** The distribution of Mixture Model ARAM’s output clusters and corresponding genome coverage on the yeast genome.

Cluster Size Range	# Clusters	# Genes	Genome Coverage (%)
≥ 100	1	107	3.47
90 – 99	-	-	-
80 – 89	1	84	2.73
70 – 79	-	-	-
60 – 69	1	69	2.24
50 – 59	1	55	1.79
40 – 49	11	470	15.26
30 – 39	15	520	16.88
20 – 29	20	457	14.84
10 – 19	52	739	23.99
0 – 9	190	579	18.80

*Large Clusters Revealed Major Gene Functions:* For ease of analysis, we further focused our validation and evaluation on the 30 largest categories. These categories grouped no less than 30 genes in each, and 1305 genes in total, which covered 42.37% of the whole genome. Table 2 depicts the distribution of these clusters according to the subsumer GO term learnt and encoded in the category pattern. Interestingly, all of these GO annotations reflected the major biological processes involved in yeast cell cycle. This is due to the fact that the microarray experiments by Eisen et al. [11] were particularly designed to stress or shock cell cycle related genes, making them response in a systematically observable fashion. We further compared these GO terms with the major gene functions reported by Eisen et al. [11], and found these two sets of functional categories were highly correlated. This shows that the proposed Mixture Model ARAM network is capable of representing major gene functional categories based on the aggregation of GO annotation and gene expression data.

### 3.3.2 Quantitative Evaluation of Algorithm’s Efficacy for Gene Functional Categorization

Numerous cluster validity measures in the literature generally fall into three major categories, namely internal criteria, external criteria and relative criteria [37]. Internal criteria measure the capacity of a clustering algorithm



**Figure 3.** The coverage of the yeast genome according to different numbers of largest clusters to be reviewed. From here, one may see that the 30 largest clusters (10.27% of total number of clusters) covers 1035 genes (42.37% of the genome), and 102 (34.93%) covers 2501 (81.20%).

**Table 2.** The distribution of the 30 largest categories and their member genes according to GO annotation.

GO Term	GO Name	# Clusters	# Genes
GO:0051179	localization	2	74
GO:0044260	cellular macromolecule metabolic process	1	42
GO:0043283	biopolymer metabolic process	3	115
GO:0006139	nucleobase, nucleoside, nucleotide and nucleic acid metabolic process	5	212
GO:0043170	macromolecule metabolic process	5	179
GO:0016043	cell organization and biogenesis	10	470
GO:0009058	biosynthetic process	3	172
GO:0007049	cell cycle	1	41
<b>Sum:</b>		30	1305

in optimizing the predefined, internally used fitness function; external criteria measure the meaningfulness of an algorithm’s output by comparing them to some external standard; and relative criteria evaluate the validity of a clustering algorithm with different parameter settings, with reference to itself. A variety of cluster validity measures based on internal criteria have been widely adopted to evaluate the efficacy of gene expression clustering algorithms [16, 25]. Although these measures quantitatively evaluate how well the grouping of the input data fit the preset (dis)similarity criteria (such as low intra-cluster variance and high inter-cluster separation), they do not necessarily reflect the algorithm’s capacity of deducing gene functions from the clustering result. With this consideration, we adopted a *biological function blind test* (BFBT), which involve external expert knowledge on genes’ biological functions, to evaluate the efficacy of our proposed algorithm.

The BFBT test was proceeded as follows: For each output category, we first provided only the list of its membership genes to expert biologists, and asked them to investigate the functions of these genes. The experts may access any data, such as functional annotations in free-text, GO annotations, expressions in any experiments and any available literature for the understanding of each gene’s function (denoted as  $f_i$ ). The experts were then asked to annotate the “common factor” function of the category (denoted as  $f_c$ ), which was enriched by the majority of its membership genes. Furthermore, the experts were asked to estimate the correlation between each individual gene’s function  $f_i$  and the category function  $f_c$  with a *relevance score*  $s_i \in [-1, 1]$ , with a positive score indicating “related gene functions”, and negative score, “un-related gene functions”. Lastly, the GO annotation (denoted as  $g_c$ ) of the category which was learnt by the proposed algorithm was revealed to the experts. And the experts were asked if these category functional annotations (namely  $f_c$  decided by the experts and  $g_c$  learnt by the algorithm) agreed to each other, quantized as  $\beta \in \{-1, +1\}$ , +1 being “agreement” and  $-1$  being “disagreement”. The validity of the category, termed as *expert knowledge fitness score*, was then formalized as

$$s = \frac{\sum_{i=1}^K \beta \cdot s_i}{K}, \quad (14)$$

where  $K$  is the number of membership genes in the category. Understandably,  $s \in [-1, +1]$  represents how a category’s function discovered by the algorithm matches that deducted by the human with external expert knowledge, +1 indicating a perfect match and  $-1$  indicating a perfect mismatch.

To facilitate the experts’ evaluation, in our practice, we further break the evaluation of  $s_i$  into five levels, namely “certainly un-related”, “somewhat un-related”, “uncertain”, “somewhat related” and “certainly related”, correspond-

ing to scores of -1.0, -0.5, 0.0, 0.5 and 1.0 respectively.

**Table 3.** The cluster validity of the 30 largest categories, evaluated using *expert knowledge fitness score*. Categories are organized according to predicted GO functional annotation term. Refer to Table 2 for corresponding GO names.

Category ID	# of Genes	GO Term	Fitness Score	
3	42	GO:0051179	0.6667	
62	32		0.7968	
174	42	GO:0044260	<b>0.9524</b>	
82	44	GO:0043283	0.6250	
146	32		0.6563	
233	39		0.7564	
69	55		0.8727	
185	41	GO:0006139	0.8415	
200	40		0.8125	
202	38		<b>0.9079</b>	
266	38		0.8421	
23	33		GO:0043170	0.8333
43	37	0.7703		
78	38	0.7895		
87	33	0.7273		
117	38	0.8289		
7	84	GO:0016043		0.5417
15	45		0.7889	
21	69		0.8043	
26	44		0.7045	
30	45		0.7111	
31	43		0.7326	
38	35		0.8000	
54	30		0.7000	
64	43		0.6744	
156	32		0.7969	
2	107		GO:0009058	<b>0.9112</b>
74	34			<b>0.9118</b>
150	31	<b>0.9032</b>		
56	41	GO:0007049	<b>0.9268</b>	
<b>Total:</b>	1305	<b>Average:</b>	0.7862±0.0988	

Table 3 reports the expert knowledge fitness scores of the 30 largest categories evaluated with the above schema. Notably, the GO functional annotations of all these categories predicted by our proposed algorithm matched those deducted by the expert. In addition, no gene received a relevance score ( $s_i$ ) of -1.0 (certainly un-related) through the blind test. Interestingly, out of

the six categories that had the highest fitness scores, five categories significantly enriched three functions, namely GO:0044260 (cellular macromolecule metabolic process), GO:0009058 (biosynthetic process) and GO:0007049 (cell cycle) which harmonized with discoveries of Eisen et al. [11]. Compared to the majority, a few categories (e.g. C7 and C82) received relatively low fitness scores. However, a closer investigation into the individual genes' relevance scores revealed that this was mainly because that a relatively large number of member genes in these categories were ranked in the "uncertain" category (with  $s_i = 0$ ) due to the lack of external information the experts obtained for determining their function. Overall, we archived an average of 0.7862 fitness score in our controlled experiment, suggesting a significantly high correlation between the knowledge discovered by the proposed Mixture Model ARAM network and the external expert knowledge.

### 3.3.3 New Perspectives Raised from Inter-Cluster Comparative Analysis

As already shown in Table 2, the 30 largest categories enriched 6 major biological process functions, whose membership genes' actual functions were further validated in our blind test and reported in Table 3. An intuitive question raised to us was why some functions were represented with multiple categories instead of single ones. Since each output category of Mixture Model ARAM was expressed with both a representative gene expression pattern and a subsumer GO annotation term, we manually compared the 30 largest categories based on these information and had some interesting findings.

Sub-figures a-1 through a-3 in Figure 4 depict the expression profiles of the output clusters that were annotated with subsumer GO terms GO:0051179 (localization), GO:0043283 (biopolymer metabolic process) and GO:0009058 (biosynthetic process) respectively. It is interesting that although these gene categories in each sub-figure involve in the same biological process, they showed different expression patterns in the microarray experiments. Most representatively, in sub-figure a-1, 42 genes in category C3 were significantly down-regulated during cell division cycle (CDC) and with response to diauxic shift stresses (DIA), while 32 genes in category C62, involving in the same localization biological process, responded in a completely opposite manner, i.e., were significantly down-regulated under these conditions. This is however not surprising to us, considering these two groups of genes could participate in two or multiple competitive localization-related biological processes, in which the activation of a process suppresses the other. In general, our analysis of all categories subsumed under same GO terms showed that their expression patterns distinguished to each other in varying manners, which provided interesting in-

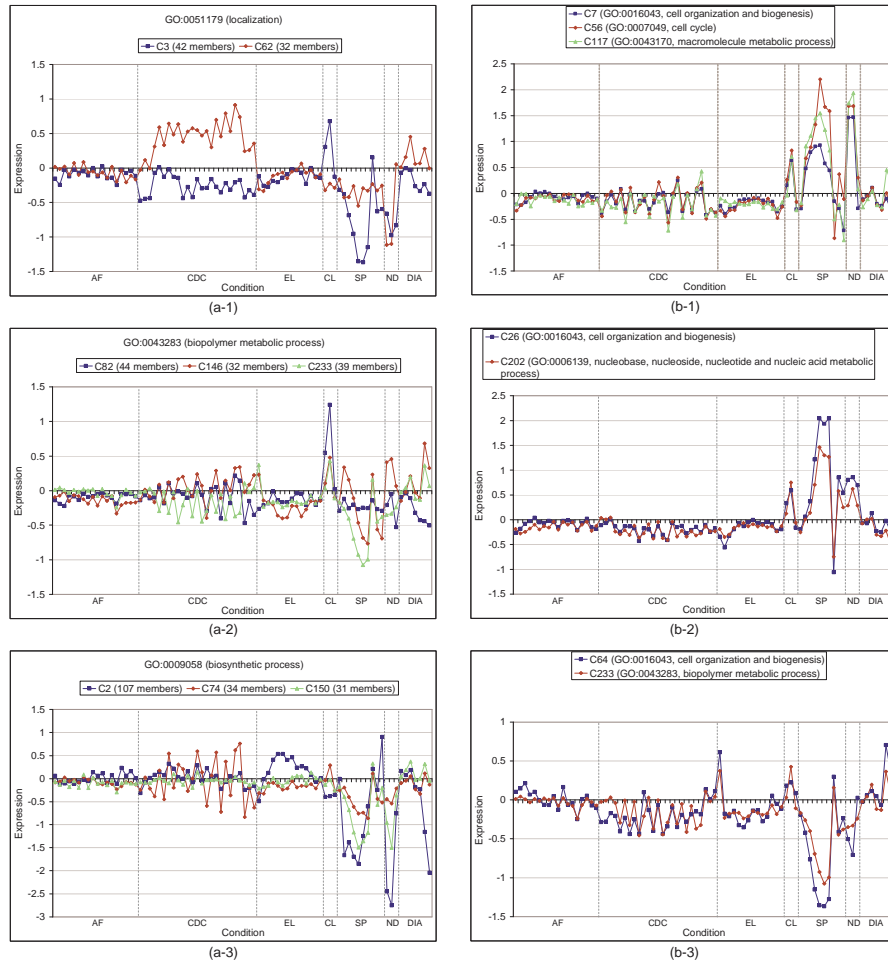
formation for further biological studies.

Similarly, as shown in sub-figures b-1 through b-3 in Figure 4, genes with highly correlated expression patterns may involve in different biological processes. Most notable, in sub-figure b-2, categories C26 and C202 grouped genes that were regulated in nearly the identical pattern under all experimental conditions but were subsumed under different GO terms, GO:0016043 (cell organization and biogenesis) and GO:0006139 (nucleobase, nucleoside, nucleotide and nucleic acid metabolic process) respectively. Our explanation to this is three-fold: Firstly, the GO terms used to annotate a gene's function are not exclusive, in the sense that many GO terms are inter-associated and the same function may be categorized under multiple GO terms, which lead to multiple Mixture Model ARAM output clusters with certain correlation to each other. Secondly, it is common that a gene may participate in multiple biological processes, which may not be fully annotated with GO terms, causing different Mixture Model ARAM output clusters referring to the same set of genes with similarly multiple functions. And thirdly, it is natural that multiple biological processes are functioned in parallel under different conditions. With such, some of Mixture Model ARAM's output clusters may actually reflect such parallel processes. While a deeper analysis of these data requires more intensive expert knowledge and could be time consuming, the output of the proposed algorithm could greatly facilitate such study.

Due to the nature of this paper in methodology research, extended biological analysis of the output clusters is not reported here. However, without losing generalization, Mixture Model ARAM presented an integrative means of investigating gene functional groups from both predicted GO functions and transcription profiles, which led to some new perspectives in genome-wide data analysis that were not raised by most other clustering algorithms based on either aspect.

## 4 Conclusion and Future Work

In this paper, we documented our understanding on the noises in sequence alignment based functional annotation data and gene expression data, and their challenges to gene functional studies based either aspect of knowledge. In order to systematically and intelligently discover gene groups that better represent common gene functions, we adopted an approach to integrate gene ontology (GO) annotation and gene expression profile for clustering genes. We proposed a novel artificial neural network named Mixture Model Adaptive Resonance Associative Map (ARAM) for this purpose. The proposed algorithm



**Figure 4.** Some representative gene categories that showed typical inconsistency between expression pattern correlation and functional correlation. Sub-figures a-1 through a-3 are categories that have similar functions but notably varying expression patterns. Sub-figures b-1 through b-3 are categories that have highly correlated gene expression patterns but different functions. Labels on X-axis identify multiple conditions in different experiments, including cell-cycle alpha-factor (AF), cell division cycle (CDC), cell-cycle elutriation (EL), cell-cycle CLN3 and CLB5 induction (CL), sporulation (SP), sporulation ndt80 (ND) and diauxic shift stress (DIA).

is based on the well-studied theoretical foundation of the ARAM architecture, with essential redefinition of pattern similarity measures and learning functions to particularly accommodate GO annotation and gene expression data. Our review of the literature revealed that our proposed algorithm is one of the few, if not the first, intelligent approaches for discovery of genome-wide functional categories through incorporating these two heterogenesis knowledge domains in a single infrastructure.

We studied the proposed algorithm's efficiency and efficacy on the budding yeast (*Saccharomyces cerevisiae*) genome through both human inspection and quantitative evaluation. Our controlled experiment showed that:

1. Mixture Model ARAM is efficient in analyzing whole genome data, in the sense that it converges in relatively few number of learning iterations using varying parameter settings.
2. The output clusters are representative to the major functional categories over the whole genome.
3. The membership genes in the same cluster significantly represent the same or highly correlated gene functions, which in turn satisfactorily match the results through human inspection with expert knowledge.
4. The inter-cluster comparison of the output categories provided useful information for the biological study of the associations between expression profile and gene function.

In summary, the Mixture Model ARAM provides an integrative infrastructure to associatively mine the GO annotation and gene expression data. This approach generally reduces analytical noises from either knowledge domain, in the sense that the output clusters significantly reflect major categories of gene functions. Most importantly, the integration of these heterogenesis knowledge domains during clustering generates interesting gene categories which are difficult to be achieved with other clustering methods based on knowledge from either domain, and provides new perspectives to genome-wide functional analysis. Although our reported work was focused on GO annotation and gene expression data, similar idea could be borrowed to integrate other heterogenesis domain knowledge for a deeper analysis of gene functions.

While the Mixture Model ARAM neural network has demonstrated satisfactory performance in our controlled experiment, a few points remain to be addressed in our future work. Firstly, like most other clustering algorithms, the proposed algorithm follows a typical unsupervised data mining paradigm to identify major functional patterns that are commonly enriched by significant

amount of gene samples, with reference to predefined similarity measures. In reality, biologists may be rather interested in some rare biological processes that involve only a few number of genes, or on the other hand, a few number of genes that only express in some particular biological conditions (such as under certain extreme stress). How to differ these genes of particular research interests from outlier/noisy data samples and discover such functional groups with minority of member genes poses challenge to data mining research based on unsupervised learning paradigms. Secondly, since the purpose of clustering genes is to deduct their functions, and the verification of these functions commonly involve intensively biological experiments, how to quantitative evaluate the efficacy of a clustering algorithm based on limited external knowledge remains an open question. In our studies, we adopted a *biological function blind test* (BFBT) which is essentially based on human inspection. This test however is labor intensive and could be subjective to the evaluators. With no doubt, there is a space for further improvement of our evaluation schema. Lastly but not least importantly, since Mixture Model ARAM generally falls into the category of multi-layer perception neural networks, it is widely understandable that gene expression pattern – gene function rules may be extracted based on the network’s recognition categories. The topic of automatic association rule extraction is yet to be covered in future studies.

## **Authors’ Contributions**

JH conducted the research, implemented the algorithm and drafted the manuscript. XD helped to evaluate the clustering results and conducted the biological function blind test. XZ coordinated and supervised the research. All authors read and approved the final manuscript.

## **Acknowledgments**

We are grateful to the valuable comments and suggestions from our colleagues Drs. Jiangqi Wen, Ewa Urbanczyk-Wochniak and Haiquan Li. Financial support to the research work was provided by the Samuel Roberts Noble Foundation.

## References

- [1] Altschul S. F., Gish W., Miller W., Meyers E. W., Lipman D. J., 1990, *Basic local alignment search tool*. Journal of Molecular Biology, Vol.215, No.3, pp.403–410.
- [2] Ben-Dor A., Shamir R., Yakhini Z., 1999, *Clustering gene expression patterns*. Journal of Computational Biology, Vol.6, No.3/4, pp.281–297.
- [3] Bottou L., Bengio Y., 1995, *Convergence properties of the K-Means algorithms*. In Tesauro G., Touretzky D., Leen T. (Eds), Advances in Neural Information Processing System, Vol.7. MIT Press.
- [4] Carpenter G., Grossberg S., 1987, *A massively parallel architecture for a self-organizing neural pattern recognition machine*. Computer Vision, Graphics, and Image processing, Vol.34, pp.54–115.
- [5] Carpenter G., Grossberg S., Rosen D., 1991, *ART 2-A: An adaptive resonance algorithm for rapid category learning and recognition*. Neural Networks, Vol.4, pp. 493–504.
- [6] Carpenter G., Grossberg S., Rosen D., 1991, *Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system*. Neural Networks, Vol.4, pp. 759–771.
- [7] Carpenter G. A., Grossberg S., Reynolds J., 1991, *ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network*. Neural Networks, Vol.4, pp. 565–588.
- [8] Chee M., Yang R., Hubbell E., Berno A., Huang X. C., Stern D., Winkler J., Lockhart D. J., Morris M. S., Fodor S. P. A., 1995, *Accessing genetic information with high-density DNA arrays*. Science, Vol.274, No.5287, pp. 610–614.
- [9] Dudoit S., Yang Y. H., Callow M. J., Speed T. P., 2000, *Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments*. Tech. Rep. 578, Department of Biochemistry, Stanford University School of Medicine.
- [10] Eddy S. R., 1998, *Profile hidden markov models*. Bioinformatics, Vol.14, 755–763.

- [11] Eisen M. B., Spellman P. T., Brownlager P. O., Botstein D., 1998, *Cluster analysis and display of genome-wide expression patterns*. Proceedings of the National Academy of Sciences, Vol.95, pp. 14863–14868.
- [12] Grossberg S., 1982, *Studies of Mind and Brain*. D. Reidel Publishing.
- [13] Hartuv E., Schmitt A., Lange J., Meier-Ewert S., Lehrachs H., Shamir R., 1999, *An algorithm for clustering cDNAs for gene expression analysis*. Proceedings of the International Conference on Computational Molecular Biology (RECOMB), pp. 188–197.
- [14] He J., Tan A.H., Tan C.L., 2003, *On machine learning methods for chinese document categorization*. Applied Intelligence, Vol.18, pp. 311–322.
- [15] He J., Tan A.H., Tan C.L., 2004, *Modified ART 2A growing network capable of generating a fixed number of nodes*. IEEE Transactions on Neural Networks, Vol.15, No.3, pp. 728–737.
- [16] He J., Tan A.H., Tan C.L., Sung S.Y., 2003, *On quantitative evaluation of clustering systems*. In: Wu W., Xiong H., Shekhar S. (Eds.), Clustering and Information Retrieval, Kluwer Academic Publishers, pp. 105–133.
- [17] Hvidsten T. R., Laegreid A., Komorowski J., 2003, *Learning rule-based models of biological process from gene expression time profiles using Gene Ontology*. Bioinformatics, Vol. 19, No.9, pp. 1116–1123.
- [18] Jain A., Murty M., Flynn P., 1999, *Data clustering: A review*. ACM Computing Surveys, Vol.31, No.3, pp. 264–323.
- [19] Jiang J. J., Conrath D. W., 1997, *Semantic similarity based on corpus statistics and lexical taxonomy*. Proceedings of International Conference Research on Computational Linguistics (ROCLING).
- [20] Kim R. S., Ji H., Wong W. H., 2006, *An improved distance measure between the expression profiles linking co-expression and co-regulation in mouse*. BMC Bioinformatics, Vol.7, pp. 44.
- [21] Kumar A., Smith B., Borgelt C., 2004, *Dependence relationships between gene ontology terms based on TIGR gene product annotations*. Proceedings of the International Workshop on Computational Terminology, pp. 31–38.

- [22] Lee S.G., Hur J.U., Kim Y.S., 2004, *A graph-theoretic modeling on GO space for biological interpretation of gene clusters*. *Bioinformatics*, Vol.20, No.3, pp. 381–388.
- [23] Lord P. W., Stevens R. D., Brass A., Goble C. A., 2003, *Investigating semantic similarity measures across the gene ontology: the relationship between sequence and annotation*. *Bioinformatics*, Vol.19, No.10, pp. 1275–1283.
- [24] Lord P. W., Stevens R. D., Brass A., Goble C. A., 2003, *Semantic similarity measures as tools for exploring the gene ontology*. *Proceedings of the Pacific Symposium on Biocomputing*, pp. 601 – 612.
- [25] Okada Y., Sahara T., Mitsubayashi H., Ohgiya S., Nagashima T., 2005, *Knowledge-assisted recognition of cluster boundaries in gene expression data*. *Artificial Intelligence in Medicine*, Vol.35, pp. 171–183.
- [26] Pruffer K., Muetzel B., Do H.-H., Weiss G., Khaitovich P., Rahm E., Paabo S., Lachmann M., Enard W., 2007, *FUNC: a package for detecting significant associations between gene sets and ontological annotations*. *BMC Bioinformatics*, Vol.8, pp. 41.
- [27] Resnik P., 1995, *Using information content to evaluate semantic similarity in a taxonomy*. *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 448–453.
- [28] Ross S., 1976, *A First Course in Probability*. Macmillan.
- [29] Sevilla J. L., Segura V., Podhorski A., Guruceaga E., Mato J. M., Martinez-Cruz L. A., Corrales F. J., Rubio A., 2005, *Correlation between gene expression and GO semantic similarity*. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, Vol.2, No.4, pp. 330–338.
- [30] Speer N., Frohlich H., Spieth C., Zell A., 2005, *Functional grouping of genes using spectral clustering and gene ontology*. *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN)*, pp. 298–303.
- [31] Tamayo P., Slonim D., Mesirov J., Zhu Q., Kitareewan S., Dmitrovsky E., Lander E., Golub T., 1999, *Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation*. *Proceedings of the National Academy of Science*, Vol.96, 2907–2912.

- [32] Tan A.H., 1995, *Adaptive Resonance Associative Map*. Neural Networks, Vol.8, No.3, pp. 437–446.
- [33] Tan A.H., 2001, *Predictive self-organizing networks for text categorization*. Proceedings of the Pacific-Aisa Conference on Knowledge Discovery and Data Mining (PAKDD), pp. 66–77.
- [34] Tan A.H., Ong H.L., Pan H., Ng J., Li Q.X., 2004 *Towards personalized web intelligence*. Knowledge and Information Systems, Vol.6, No.5, pp. 595–616.
- [35] Tan A.H., Pan H., 2005, *Predictive neural networks for gene expression data analysis*. Neural Networks, Vol.18. No.3, pp. 297–306.
- [36] The Gene Ontology Consortium, 2000, *Gene ontology: tool for the unification of biology*. Nature Genetics, Vol.25, No.1, pp. 25–29.
- [37] Theodoridis S., Koutroubas K., 1999, *Pattern Recognition*. Academic Press.
- [38] Velculescu V., Zhang L., Vogelstein B., Kinzler K., 1995, *Serial analysis of gene expression*. Science, Vol.270, No.5235, pp. 484–487.