Microarray Gene Cluster Identification and Annotation Through Cluster Ensemble and EM-Based Informative Textual Summarization

Xiaohua Hu, Member, IEEE, E. K. Park, and Xiaodan Zhang

Abstract—Generating high-quality gene clusters and identifying the underlying biological mechanism of the gene clusters are the important goals of clustering gene expression analysis. To get high-quality cluster results, most of the current approaches rely on choosing the best cluster algorithm, in which the design biases and assumptions meet the underlying distribution of the dataset. There are two issues for this approach: 1) usually, the underlying data distribution of the gene expression datasets is unknown and 2) there are so many clustering algorithms available and it is very challenging to choose the proper one. To provide a textual summary of the gene clusters, the most explored approach is the extractive approach that essentially builds upon techniques borrowed from the information retrieval, in which the objective is to provide terms to be used for query expansion, and not to act as a stand-alone summary for the entire document sets. Another drawback is that the clustering quality and cluster interpretation are treated as two isolated research problems and are studied separately. In this paper, we design and develop a unified system Gene Expression Miner to address these challenging issues in a principled and general manner by integrating cluster ensemble, text clustering, and multidocument summarization and provide an environment for comprehensive gene expression data analysis. We present a novel cluster ensemble approach to generate high-quality gene cluster. In our text summarization module, given a gene cluster, our expectation–maximization based algorithm can automatically identify subtopics and extract most probable terms for each topic. Then, the extracted top k topical terms from each subtopic are combined to form the biological explanation of each gene cluster. Experimental results demonstrate that our system can obtain high-quality clusters and provide informative key terms for the gene clusters.

Index Terms—Cluster ensemble, expectation–maximization (EM), microarray gene expression analysis, text mining.

I. INTRODUCTION

HUGE amounts of gene expression data have been generated as a result of the Human Genomic project, which creates a need and challenge for data mining. Clustering algorithms are used as essential tools to analyze gene expression datasets and provide valuable insight on various aspects of the genetic machinery such as identifying the functionality of genes, finding out what genes are coregulated, distinguishing the important genes between abnormal tissue and normal tissues, etc. [1], [2]. Generating high-quality gene clusters and identifying the underlying biological mechanism of the gene cluster are the ultimate goal of clustering gene expression analysis. Some efforts and progress have been made toward this goal [3]–[9].

But there are some drawbacks of these approaches. To get high-quality cluster results, these approaches rely on choosing the best cluster algorithm whose design biases and assumptions meet the underlying distribution of the dataset. Otherwise, the results will be poor if the assumptions are violated in a dataset. There are two issues for this approach: 1) usually, the underlying data distribution of the gene expression datasets is unknown and 2) there are so many clustering algorithms available. It is a challenging and daunting task for genomic researchers to choose the best one for a particular gene expression dataset because results of different clustering algorithms are inconsistent. K-means, self-organizing map (SOM), hierarchical approaches, fuzzy c-means, etc., are very different in some cases [10]. This is because clustering methods have their own biases and function criterion. It is well known that no single clustering algorithm performs best across various datasets.

On the other hand, clustering indeed reveals potential meaningful relationships among genes, but cannot explain the underlying biological mechanisms. To provide a textual summary of the gene clusters, the most explored approach currently is the extractive approach that essentially builds upon techniques borrowed from the information retrieval field such as term reweighting and relevance feedback [11], [12], in which the objective is to provide terms to be used for query expansion, and not to act as a stand-alone summary for the entire document sets. Another drawback is that the clustering quality and cluster interpretation are treated as two isolated research problems and are studied separately. But cluster quality and cluster interpretation are closely related and must be addressed in a coherent and unified way. It is essential to have relatively high-quality clusters first in order to get a correct, informative biological explanation of the gene cluster. Otherwise, the biological explanation will be incorrect or misleading, no matter how good or robust the text summarization technique is. Based on this consideration, this paper explores the first step toward dealing with these issues. We design and develop a unified system Gene Expression Miner...
(GE-Miner) to address these challenging issues in a principled and general manner by integrating cluster ensemble and text summarization and provide an environment for comprehensive gene expression data analysis. The task of establishing a unifying framework for comprehensive gene expression analysis is accomplished in three steps.

1) **Cluster ensemble**: This step involves building a cluster ensemble method to combine the clustering results from various clustering algorithms in order to obtain high-quality and robust results.

2) **Data integration server**: This step involves developing an extendable data integration server to gather related textual resource from various databases of the genes.

3) **Textual summarization**: This step involves integrating biomedical literature mining in gene expression analysis to provide informative biological explanation of the gene clusters.

The rest of the paper is organized as follows. In Section II, we give a brief overview of the architecture of GE-Miner, and then, we present the cluster ensemble method in Section III. We discuss the data integration server in Section IV and text summarization in Section V. The experimental results are presented in Section VI. We conclude with our future plan and discussion in Section VII.

### II. OVERVIEW OF GE-MINER ARCHITECTURE

Obtaining high-quality clustering results is very challenging because of the inconsistency of the results of different clustering algorithms and the information contained in microarray data is limited by the number of arrays, their quality, and noise and experimental errors. Another significant limitation of the current clustering approach is that most of these algorithms provide no biological interpretation of the cluster results; the users need to discover and interpret the biological similarities that may underlie the expression pattern by cross-referencing the experimental results in related literature or functional annotations in various genomic databases. Since gene cluster may include dozens or even hundreds of different genes, it is beyond the limits of biological researchers to detect and organize these data along multiple lines of conceptual similarity by inspection them manually. Thus, it is essential to develop a system capable of gathering biological information and extracting and summarizing relevant information in a well-organized and coherent manner for the gene cluster.

We develop a comprehensive gene expression mining system **GE-Miner**, as shown in Fig. 1, which is geared precisely for this task helping a biologist in cross-referencing experimental and analytical results obtained from microarray experiments, and provide concise and meaningful biological explanation of the gene clusters. **GE-Miner** aims to summarize the biomedical knowledge for genes on a genome-wide scale, generates relevant summaries from relevant biological literatures, and summarizes biological information about a group of genes in a concise and coherent manner. It has an open architecture and can easily add a wrapper if new data sources become available, without affecting the rest of the system. The components of GE-Miner are described in details in the following sections.

### III. CLUSTER ENSEMBLE

The purpose of cluster ensemble is to build a robust clustering portfolio that can perform as good as if not better than the single best clustering algorithm across a wide range of datasets. Different clustering algorithm may take a different approach. For example, $K$-means is to group the dataset so that the total mean square error to the centers of each cluster is minimum while graph-based partitioning clustering is to partition the graph into $K$ parts based on the minimum edge weight cuts. Thus, a cluster ensemble can be used to generate many cluster results using various clustering algorithms, and then, integrate them using a consensus function to yield stable results.

Classification ensemble approaches such as bagging and boosting have been proved very popular and effective in supervised learning to improve the learning accuracy [13], [14]. Even though many clustering algorithms have been developed [10], not much work is done in cluster ensemble in data mining and machine learning literature compared with classification ensemble method. Zeng et al. [15] proposed an adaptive metaclustering approach for combining different clustering results based on a hierarchical clustering strategy. Strehl and Ghosh [16] proposed a hypergraph-partitioned approach to combine different clustering results. Each cluster in an individual clustering algorithm is treated as a hyperedge. This crisp hypergraph lost much useful information, and it is not suitable for ambiguous and noisy environment such as micorarray experiments.

In this section, we discuss our novel cluster ensemble approach to combine the clustering results from various clustering algorithms, as shown in Fig. 2. We present a two-phase clustering combination strategy. At the first step, various clustering algorithms are run against the same datasets to generate clustering results. At the second step, these clustering results are

---

**Fig. 1.** Architecture of GE-Miner.

**Fig. 2.** Data flow of cluster ensemble.
combined by an autoassociative additive system based on the distance matrix of graph clustering. The diagram in Fig. 2 summarizes our approach.

In our approach, a distance matrix is first constructed based on the cluster results from each individual clustering algorithm; these similarity matrices are combined to form a master distance matrix. Then, a similarity graph is constructed from the master distance matrix, and a graph-based partitioning algorithm is applied to the graph for the final clustering results. Graph-based clustering uses various kinds of geometric structure or graphs for analyzing data. Different graphs reflect various local structure or inherent visual characteristic in the dataset.

Clustering divides the graph into connected components by identifying and deleting inconsistent edges, and each subgraph consisting of connected components refers to a cluster.

A. Clustering Ensemble Algorithm

Algorithm 1: Cluster Ensemble Based on Similarity Graph (CESG)

Input: (i) the dataset \( X = \{x_1, x_2, x_3, \ldots, x_n\} \), (ii) edge threshold value \( \delta \), (iii) a set of different clustering algorithms \( C^{(q)} \) 

Output: the final clustering result \( C^{(\text{opt})} \)

Method:

Step 1: Run the clustering algorithm \( C^{(q)} \) one at a time on the same dataset.

Step 2: Construct a distance matrix (DM\( ^{(q)} \)) for the clustering results for each clustering algorithm. (DM\( ^{(q)} \)) represents the similarity of two data \( x_i \) and \( x_j \) points under cluster algorithm \( C^{(q)} \).

Step 3: Combine the distance matrices by adding them into one master distance matrix (MDM).

Step 4: Construct a weighted graph based on the distance matrix. (There is an edge between data point \( x_i \) and \( x_j \) if the distance value \( MDM_{ij} \) of \( x_i \) and \( x_j \) is greater than some threshold value \( \delta \); \( MDM_{ij} \) is also the weight of the edge link \( x_i \) and \( x_j \).)

Step 5: Cluster the graph into a set of clusters according to the evaluation score.

In step 2, there are so many ways to construct the distance matrix based on cluster results from the individual clustering algorithm. We adapted a solution based on [15].

B. Final Stage Cluster-Based Distance Matrix DM\( ^{(q)} \) for the Clustering Result \( C^{(q)} \)

DM\( ^{(q)} \) is a pairwise distance matrix defined between two data points according to the clustering result. The matrix size is \( n \times n \). Since its size is independent of the clustering approach, it provides a way to align the different clusterings onto the same space even for some situations where the numbers of clusters are different for different clustering algorithms.

We assume that probability density function of \( s_j \) is given by \( p(x_i|s_j) \); the posterior probability of cluster \( s_j \) given \( x_i \) can be expressed as

\[
P(s_j|x_i) = \frac{p(x_i|s_j) \times P(s_j)}{\sum_{k=1}^{m} p(x_i|s_k) \times P(s_k)}
\]

where

\[
P(x_i|s_j) = \exp[-\frac{1}{2}(x_i - \mu_j)^T \sum_j (x_i - \mu_j)]
\]

\( m \) is the number of clusters, \( \sum_j \) is a matrix of covariances among attributes in cluster \( j \), and \( \mu_j \) is the mean vector of the data points in the cluster \( s_j \).

For each data point \( x_i \), we calculate the corresponding probability vector \( PX_i = \{P(s_1|x_i), P(s_2|x_i), \ldots, P(s_m|x_i)\} \), where \( \sum_{j=1}^{m} P(s_j|x_i) = 1 \); the probability vectors form a probability space of dimension of \( m \), with each dimension corresponding to one cluster. The probability space contains information from both the input data and the cluster results. So, we believe the similarity of any two points \( PX_i \) and \( PX_m \) in the probability space is a good measurement to reflect the distance of the corresponding points \( x_i \) and \( x_m \) in the original space.

Then, for any two points \( x_i \) and \( x_m \) in the dataset, their distance is defined as the distance between \( PX_i \) and \( PX_m \), namely \( DM^{(q)}(x_i, x_m) \). Many different distance measures such as Euclidean distance, Mahalanobis distance, or correlation distance can be used to calculate \( DL(PX_i, PX_m) \).

We define the distance of two points \( (x_i, x_j) \) in the dataset under algorithm \( C^{(q)} \) as

\[
DM^{(q)}_{ij} = 1 - \frac{\sum PX_i \cdot PX_j - (\sum PX_i \cdot \sum PX_j/N)}{\sqrt{(\sum PX_i^2 - (\sum PX_i)^2/N) \times (\sum PX_j^2 - (\sum PX_j)^2/N)}}
\]

In step 5, a graph-based clustering algorithm is applied to the weighted graph for the final clustering result. Many graph-based partitioning algorithms can be used for this purpose. We select METIS [17] for the graph partitioning because of its scalability and efficiency.

C. Clustering Result Evaluation

To evaluate the quality of cluster is a nontrivial and often ill-posed task. Generally speaking, there are internal criteria and external criteria. Internal criteria formulate quality as a function of the given data and/or similarities. External criteria on the other hand impose quality by additional, external information not given to the clusters, such as category labels. This is sometimes more appropriate since groupings are ultimately evaluated externally by humans. For example, when objects have already been categorized by an external source, i.e., when class labels are available, we can use information theoretical measure to quantify the match between the categorization and the clustering.

In our cluster ensemble, external criteria fit very well with our architecture. We use the Minkowski score (MS) [18] as our cluster quality indicator. Our formula for the clustering quality evaluation is given later.

A clustering solution for a set of \( n \) elements can be represented by an \( n \times n \) matrix \( C \) where \( C_{ij} = 1 \) iff \( x_i \) and \( x_j \) are in the same cluster according to the solution and \( C_{ij} = 0 \) otherwise. A measure of MS between the clustering results \( C^{(b)} \) from a particular clustering algorithm \( CA_h \) with a reference clustering
TABLE I

SOME OF YEAST GENE FUNCTION FAMILY

<table>
<thead>
<tr>
<th>Function Families</th>
<th># of genes</th>
<th>Cluster Sets</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATP synthesis</td>
<td>19</td>
<td>C3</td>
</tr>
<tr>
<td>mitosis</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>vacuolar protein targeting</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>silencing</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>fatty acid metabolism</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>meiosis</td>
<td>21</td>
<td>C5</td>
</tr>
<tr>
<td>phospholipid metabolism</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>TCA cycle</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>protein processing</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>DNA repair</td>
<td>29</td>
<td>C4</td>
</tr>
<tr>
<td>protein folding</td>
<td>30</td>
<td>C6</td>
</tr>
<tr>
<td>nuclear protein targeting</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>signaling</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>major facilitator superfamily</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>mRNA splicing</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>chromatin structure</td>
<td>42</td>
<td></td>
</tr>
<tr>
<td>DNA replication</td>
<td>42</td>
<td>C2</td>
</tr>
</tbody>
</table>

$T$ (or alternatively, the true clusters if the cluster information in the dataset is known in advance) is defined as

$$MS(T, C^{(h)}) = \frac{||T - C^{(h)}||}{||T||}$$

where $||T|| = \sqrt{\sum_i \sum_j T_{ij}}$.

We abbreviate the set of cluster groupings from $r$ different clustering algorithms as $\Psi = \{C^{(q)}| q \in \{1, \ldots , r\}\}$. The average MS score of combined clustering result $C$ with $\Psi$ is defined as

$$MS^{(ANMI)}(C, \Psi) = \frac{1}{r} \sum_{q=1}^{r} MS(C, C^{(q)}).$$

D. Yeast Gene Dataset

There are 6221 genes in the datasets but not every gene is classified into a certain function family. In our experiment, we considered the genes in a function family as one cluster and created six datasets (clusters 2, 3, 4, 5, 6, and 7). Table I shows six function families of yeast gene and how we construct the six datasets (C2, C3, C4, C5, C6, and C7) for our cluster ensemble comparison. For example, “C3” means the cluster set has three clusters (ATP synthesis, mitosis, and vacuolar protein targeting here).

Table II shows the clustering results including cluster ensemble in MSs for each cluster set. As clearly indicated by the MS values of the clusters, the cluster ensemble method made significant improvement of quality of the clustering results over the individual clustering algorithm on all the six gene datasets. For example, the best individual clustering algorithm for C3 is K-means (MS = 0.890), while the cluster ensemble has MS = 0.728. For C5, the best individual clustering algorithm is SOM (MS = 1.241) and the cluster ensemble reduced them to MS = 1.059.

IV. DATA INTEGRATION SERVER: COMPILATION OF THE KNOWLEDGE/INFORMATION OF THE GENE CLUSTERS

There are a variety of biological databases that can be mined to find possible functional relationships between genes in a cluster. For example, biomedical literature databases such as PubMed, which are a rich source of information, can be used to discover and analyze significant biological information on a genome-wide scale. We collect and compile several sources for textual annotations of the gene clusters. First, we retrieve the gene descriptions from UniGene and the corresponding genome databases of some species through Ensemble (for example, mouse, human, etc.). Second, proteins that are the products of the given genes are also of interest. Hence, it is important to know the proteins made by genes. We use SWISS-PROT, which is a curate protein sequence database. It serves as an extended textual resource for the genes. But this information is often insufficient and bibliographic information must be consulted by following the links to select a PubMed abstract provided in some sequence databases. Since only a small fraction of these pointers provide direct information about gene function, further references are usually collected by querying PubMed directly (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi) with gene names and their synonyms. Given the complexity and growth of biomedical literature, we need a system such as our GE-Miner capable of filtering the literature database and extracting and summarizing relevant information in a well-organized and coherent manner. In GE-Miner, we developed many built-in wrappers for data retrieval for various data sources. The GE-Miner can support any source for alias specification through a customized wrapper. Each query retrieves documents for a single gene that are associated with one or more of the aliases for the gene.

The key terms in the gene expression stop word list are removed from the abstracts. According to [19], in the enriched gene expression datasets, each gene will have words and expressions in common including: 1) standard English words, such as “the” and “experiment”; 2) words with general biological meaning, such as “gene,” “high-throughput technology,” “hybridization,” and “base sequence”; and 3) specific words and phrases such as cell cycle, glucose, kinase, and DNA replication. It is the last set that may be considered specific to the group of genes. The textual information is used to extract those words
that have significant frequency and specificity for each group. We need to create a gene expression stop list to remove those common English words and general biological terms from the text associated with the gene clusters. Following the method proposed in [6], we create the list by querying PubMed with a large random collection of genes so that an equal number of genes were chosen from different classes of organisms like eukaryotes, prokaryotes, chordates, etc. Then, we extract the keyphrase [20] for this list and rank them in descending order of their total frequency. The distribution of the list followed Zipf’s law, which is commonly seen in word-frequency distribution [21]. We consider the first region with high frequency of Zipf’s curve. The remaining terms from this region constituted the key phrase stop word list.

V. EXPECTATION–MAXIMIZATION BASED INFORMATIVE
TEXTUAL SUMMARYIZATION OF GENE CLUSTERS

A common approach to information retrieval from the biomedical literature is the use of “keywords” representing the essential concepts contained within a text. A variety of approaches to provide a biological explanation of gene clusters have been developed. TextQuest [22] is geared to summarize documents retrieved in response to a keyword(s) based search on PubMed. It does not retain the association between the genes (keywords) and the retrieved documents. MedMiner [9] can provide summarized literature information on genes but it is limited to finding relations between two genes only and it also returns a few hundreds sentences. Shatkey et al. [8] suggested a system, which attempts to find functional relations among genes on genome-wide scale, but this requires user to specify a representative document for each gene, which describe the gene very well. Looking for the representative document may need a lot of time, effort, and knowledge on part of user. Also, as genes have multiple biological functions, it is very rare to find a document that covers all aspect of gene across various biological domains. GEISHA [19] is based on a comparison of the frequency of abstracts linked to different gene clusters and containing a given term. Interpretation by the end user of the biological meaning of the terms is facilitated by embedding them in the corresponding significant sentences and abstracts and by establishing relations with other, equally significant terms.

Standard text summarization algorithms [6], [7] are geared to summarizing all the documents retrieved based on keyword searches. In multikeyword searches, the association between a document and the keywords(s) is not used. However, in our context, interest lies in capturing significant biological properties that are most relevant to the cluster as a whole.

Our approach takes each gene cluster as a major topic. Each major topic is summarized as several subtopics. Each subtopic contains certain number of documents and terms. Therefore, the related document set of a gene cluster is clustered into a few document term clusters; each document term cluster corresponds to a subtopic of the gene cluster. Our rational of summarization is that the document set of a gene cluster contains various biological subtopics. An expectation–maximization (EM) based multidocument summarization algorithm is applied to summarize each gene cluster separately. The EM-based algorithm automatically extracts most probable terms and documents for each subtopics. Finally, the extracted top $k$ terms from each subtopic are combined to form the biological explanation of each gene cluster.

A. Text Classification and Summarization

It should be noted that our EM model follows the framework of [23], especially, the naive Bayes generative probability model. However, our study has several major differences from [23]. The first major difference is that we assume different text section of a document plays different role for estimating the probability of a term belonging to a subtopic when it is known that some documents belong to this subtopic. A two-component mixture model is accordingly developed to estimate term probability belonging to a subtopic. Thus, it improves the classification and summarization results of EM when treating terms from different section of text differently. Second, while [23] focuses on document clustering, we focus on text summarization. They aim to predict a document class label, while our task is to find the subtopic within a class. In our case, document’s class label is already known, i.e., the gene cluster name. Third, for term probability smoothing, we import a mixture weights parameter $\alpha$ to control the portion of how much a term probability is distributed according to subtopic and collection frequency, which serves as lower ranking terms that appear in most of documents. Fourth, the presentation of a document is also different. While they use unigram to represent a document, we use medical concepts to represent a document, which help catch more meaningful phrases. The medical concepts are extracted by part of speech tagging using unified medical language system (UMLS) ontology.

B. Generative Probability Model

A generative probability model is presented embodying three assumptions: the data are produced by a subtopic, and there is one to one correspondence between subtopics and data, and a document is composed of a list of independent terms. Thus, every document is generated by a probability distribution denoted by $\theta$. The probability distribution consists of a mixture of subtopics $c_j \in C = \{c_1, c_2, \ldots, c_C\}$. A document $d_i$ is created by first selecting a subtopic according to subtopic prior probabilities $P(c_j | \theta)$, then having this selected subtopic generate a document according to its own parameters, with distribution denoted by $\theta$. The probability distribution consists of a mixture of subtopics $c_j \in C = \{c_1, c_2, \ldots, c_C\}$. A document $d_i$ is created by first selecting a subtopic according to subtopic prior probabilities $P(c_j | \theta)$, then having this selected subtopic generate a document according to its own parameters, with distribution $P(d_i | c_j, \theta)$. Thus, we can characterize the likelihood of document $d_i$ with a sum of total probability over all subtopics

$$P(d_i | \theta) = \sum_{j=1}^{C} P(c_j | \theta)P(d_i | c_j; \theta).$$ (1)
A document $d_i$ is considered to be an independent list of terms $(t_{d_i,1}, t_{d_i,2}, \ldots, t_{d_i,|d_i|})$; therefore, a naïve Bayes probability of a document is thus given its subtopic

$$P(d_i | c_j; \theta) = P(|d_i|) \prod_{k=1}^{|d_i|} P(t_{d_i,k} | c_j; \theta). \quad (2)$$

The complete collection of model parameters $\theta$ is a set of multinominals and prior probabilities over these multinomials: $\theta = \{ \theta_{c_m | c_j} : t_m \in T; c_j \in C; \theta_{c_j} : c_j \in C \}$. The term probability estimates is

$$\hat{\theta}_{t_m | c_j} = P(t_m | c_j; \hat{\theta}) = \frac{\sum_{n=1}^{|d|} N(t_m, d_i) P(c_j | d_i)}{\sum_{n=1}^{|d|} \sum_{j=1}^{|C|} N(t_m, d_i) P(c_j | d_i)} + (1 - \alpha) \frac{\sum_{n=1}^{|d|} N(t_m, d_i)}{\sum_{n=1}^{|d|} \sum_{j=1}^{|C|} N(t_m, d_i)} \quad (3)$$

where $N(t_m, d_i)$ is the number of the times term $t_m$ occurs in document $d_i$, where $\alpha \in (0, 1]$ is mixture weight to control the how much term probability is distributed according to subtopic frequency versus collection frequency; the collection frequency is mostly used here to remove background noise and prevent zero probability. $P(c_j | d_i) \in \{ 0, 1 \}$ is the document probability belonging to the subtopic label. For term probability estimation, a translation probability model is presented as in (3). We further assume that different section of text of a document plays different role on assigning term subtopic probability. For PubMed data, we have two text sections for a document: title and abstract. So, we integrate the language translation model to term probability estimation. Accordingly, we have the term probability equation as follows:

$$\hat{\theta}_{t_m | c_j} = P(t_m | c_j; \hat{\theta}) = \lambda P(t_m | \text{Title}; C_j; \hat{\theta}) + (1 - \lambda) P(t_m | \text{Abstract}; C_j; \hat{\theta}) \quad (4)$$

where $P(t_m | \text{Title}; C_j; \hat{\theta})$ is the conditional probability of $t_m$ given term from title and document from $C_j$, which is called title model; similarly $P(t_m | \text{Abstract}; C_j; \hat{\theta})$ is called abstract model; $\lambda \in [0, 1]$ is the interpolation factor, which can be viewed as mixture weight if (4) is considered as a two-component mixture model; $P(t_m | \text{Title}; C_j; \hat{\theta})$ can be easily calculated through (3), as well as $P(t_m | \text{Abstract}; C_j; \hat{\theta})$, in which $P(c_j | d_i)$ needs to be updated to $P(c_j | \text{Title}_i)$ and $P(c_j | \text{Abstract}_i)$ for title and abstract term probabilities, respectively. The mixing coefficient $\lambda$ is simply the fractional membership of all data points in the first title model

$$\lambda^{(n+1)} = \frac{1}{|T|} \frac{1}{K} \sum_{t_m \in T} \frac{\lambda^{(n)} P(t_m | \text{Title}; C_j; \hat{\theta})}{\lambda^{(n)} P(t_m | \text{Title}; C_j; \hat{\theta}) + (1 - \lambda^{(n)}) P(t_m | \text{Abstract}; C_j; \hat{\theta})} \quad (5)$$

where $K$ indicates the number of subtopics. The subtopic prior probabilities $\hat{\theta}_{c_j}$ are estimated in the same manner, and also involve a ratio of counts with smoothing. Pseudocounts referred to as Laplace smoothing is used to prevent zero probabilities

$$\hat{\theta}_{c_j} = P(c_j | \hat{\theta}) = \frac{1 + \sum_{d_i \in \mathcal{D}} P(c_j | d_i)}{|C| + |D|}. \quad (6)$$

Through the Bayes’ rule and (2), it is easy to derive the document–subtopic probability equation

$$P(c_j | d_i; \hat{\theta}) = \frac{P(c_j | \hat{\theta}) P(d_i | c_j; \hat{\theta})}{P(d_i | \hat{\theta})} = \frac{P(c_j | \hat{\theta}) \prod_{k=1}^{|d_i|} P(t_{d_i,k} | c_j; \hat{\theta})}{\sum_{c_j} P(c_j | \hat{\theta}) \prod_{k=1}^{|d_i|} P(t_{d_i,k} | c_j; \hat{\theta})} \quad (7)$$

with title as well as abstract is considered to be a list of independent terms. Finding $\theta$ that maximizes $P(\theta | D)$ is accomplished by first breaking this expression into two terms by Bayes’ rule: $P(\theta | D) \propto P(D | \theta) P(\theta)$. The first term is calculated by the product of all the documents likelihoods. The second term, the prior distribution over parameters, is represented by a Dirichlet distribution

$$P(\theta) \propto \prod_{c_j \in C} \left( \left( \theta_{c_j} \right)^{\beta - 1} \left( \lambda \prod_{t \in |T|} \theta_{t|c_j} \right)^{\beta - 1} \right)$$

where $\beta$ is parameter that affects the strength of the prior, and is some constant greater than zero. It is set as 2 in this paper, which is equivalent to Laplace smoothing. Let $l(\theta | D) \equiv \log(P(\theta) P(D | \theta))$. Then, the complete log likelihood is expressed as follows:

$$l_c (\theta | D, z) = \log(P(\theta) + \sum_{d_i \in D} \sum_{j=1}^{|C|} z_{ij} \log(P(c_j | \theta) P(d_i | c_j))) \quad (8)$$

where $z_{ij} = 1$ if document belongs to subtopic $c_j$, otherwise $z_{ij} = 0$. Thus, a locally maximum $\hat{\theta}$ by a hill-climbing procedure can be found, which was formalized as the EM algorithm by Dempster et al. [24].

C. Practical Algorithm

We will outline here our approach first as a series of steps, and then, give more detail in how to begin the computation and how to control it.

Inputs: The inputs are relative document set of a gene cluster $\alpha$ and $\lambda$, and seed document number $M$.

1) Set the number of mixture components (subtopics) per class (gene cluster) by cross-validation.

2) Pick $M$ documents randomly as seed documents to assign $P(c_j | d_i, \theta)$ for subtopics to initialize each subtopic.

3) Build an initial naïve Bayes classifier $\hat{\theta}$ from the assigned documents only. Use maximum $a posteriori$ parameter estimation to find $\theta = \arg \max_{\theta} P(D | \theta) P(\theta)$ [see (3), (4), and (6)].

4) Loop while classifier parameters improve—the change in complete log probability of the data, and the prior as measured by the change in $l_c (\theta | D, z)$ [see (8)]:

   a) (E-step) Use the current classifier $\hat{\theta}$ to estimate the subtopic membership of each document, i.e., the
probability that each subtopic generated each document \( P(c_j | d_i, \theta) \) [see (7)].

b) (M-step) Reestimate the classifier \( \hat{\theta} \), given the estimated subtopic membership of each document. Reestimate parameter \( \lambda \) [see (5)]. Use maximum \textit{a posteriori} parameter estimation to find \( \theta = \arg \max_\theta P(D | \theta) P(\theta) \).

**Outputs:** A classifier \( \hat{\theta} \) that predicts a subtopic label for a given document and term.

As for initialization of the algorithm, in practice, we set \( \alpha \) to 0.7, \( \lambda \) to 0.5, and \( M \) to 20. It is reasonable to set smoothing parameter \( \alpha \) to 0.7 because subtopic frequency should play more important role when clustering terms to subtopics. Setting \( \lambda \) initially to 0.5 indicates term from title plays as important role as term from abstract. \( \lambda \) is then iteratively learned through each step of probability distribution by (5). \( M \) is the number of documents assigned to each subtopic. \( M \) is set to a small number relative to document set because randomly assigned documents might not be the subtopic center, and it can be more robust to let unlabeled data to shape each subtopic. It is evident that if we can obtain the number of subtopics in step 1 and \( P(c_j | d_i, \theta) \) sampled from a uniform distribution over subtopics in step 2, the remaining steps are relatively straightforward to perform. As one to one correspondence between document and subtopic is assumed, the value of \( P(c_j | d_i, \theta) \) is restricted to 1 and 0. The algorithm begins by assigning the values \( P(c_j | d_i, \theta) \) to 0 or 1 depending on the preliminary random sampling process in steps 1 and 2. Then, in step 3, through (3), (4), and (6), term condition probability given subtopic can be calculated. Accordingly, document and term subtopic probability can be computed easily through E-step and M-step, respectively. In E-step, we set \( P(c_j | d_i, \theta) \) with the biggest value to 1 and others to 0. In step 4, convergence is tested by observing the change in complete log probability of the data, and the prior as measured by the change in \( l_c(\theta | D, z) \). When the algorithm converges, there is certain number of most probable terms for each subtopic. At last, top 20 most probable terms from each subtopic are picked as terms for summarization. Our experiments show that the algorithm usually converges within ten iterations and less than 30 s.

**D. Experiment Result**

The input dataset for gene cluster summarization is relevant document sets of each gene cluster. In Data Integration Sever, the input document sets of each gene cluster are generated in the following procedures.

1) For each gene, the synonyms are searched. If found, the synonym(s) is/are added to gene list.

2) Relevant documents are fetched from PubMed. The search keyword has this format: Gene name AND (Model organism names).

For example, for the yeast genes used in our experiment described in Section III-D, the number of genes (also synonym) and the related documents are shown in Table III.

We conduct some experiment study on yeast gene dataset (http://rana.lbl.gov/EisenData.htm). The reason we choose the yeast DNA microarray is because the validity of our methods is

| TABLE III  |
|---|---|---|
| Gene cluster # | # of genes in the cluster (including synonyms) | # of relevant PubMed documents | # of subtopics for each gene cluster |
| 1 | 19 (25) | 157 | 3 |
| 2 | 19 (35) | 820 | 4 |
| 3 | 19 (69) | 417 | 5 |
| 4 | 20 (30) | 1043 | 9 |
| 5 | 20 (34) | 641 | 5 |
| 6 | 21 (35) | 644 | 3 |
| 7 | 21 (31) | 374 | 4 |
| 8 | 22 (30) | 321 | 2 |
| 9 | 42 (67) | 1180 | 6 |
| 10 | 42 (75) | 2769 | 11 |

| TABLE IV  |
|---|---|
| Gene cluster # | Function Families |
| 1 | ATP synthesis |
| 2 | mitosis |
| 3 | vacuolar protein targeting |
| 4 | silencing |
| 5 | fatty acid metabolism |
| 6 | meiosis |
| 7 | phospholipid metabolism |
| 8 | TCA cycle |
| 9 | chromatin structure |
| 10 | DNA replication |
In this paper, we present a novel system GE-Miner for comprehensive gene expression analysis. Our system integrates cluster ensemble and text summarization, and the experiment results on yeast gene expression data indicate that the GE-Miner can generate better quality and robustness clusters and provide informative term summary for the gene clusters. Moreover, we provide a novel EM-based multidocument summarization method. We utilize a mixture model of terms from different text sections of documents to estimate topic probability and apply collection frequency smoothing method to remove background noise. Experiment results show that the trained coefficient $\lambda$ of the mixture model constantly get a stable value about 0.1, which indicates that title terms contribute more than abstract terms to topics. In practice, the algorithm can automatically identify subtopics and assign most probable terms to each subtopic, which are combined to form the biological explanation of each gene cluster. Clustering ensemble is a new and very promising research area. There are a lot of open problems for future research. We plan to expand our ensemble approach to integrate feature selection for clustering very high-dimensional dataset and add some inference mechanism to automatically infer valid information from the clustering results. Text summarization for gene literature has attracted a lot of attention recently, and one of the challenging issue is how to get relevant literature from the huge and diversified literature because a gene tends to have many alias name and there is no standard naming convention. In our future research, we plan to apply data mining techniques to automatically find gene synonyms to enhance the precise rate of the retrieved literature in order to get more relevant textual information. We hope to report our findings in the near future.

**REFERENCES**


Xiaohan Hu (M’01) received the B.Sc. degree (software) from Wuhan University, Wuhan, China, in 1985, the M.Eng. degree in computer engineering from the Institute of Computing Technology, Chinese Academy of Science, Beijing, China, in 1988, the M.Sc. degree in computer science from Simon Fraser University, Burnaby, BC, Canada, in 1992, and the Ph.D. degree in computer science from the University of Regina, Regina, SK, Canada, in 1995.

He is a Yellow-River Scholar at Henan University, Kaifeng, China. In 2002, he joined Drexel University, Philadelphia, PA, where he is currently an Associate Professor and the Founding Director of the Data Mining and Bioinformatics Laboratory, College of Information Science and Technology. He founded the International Journal of Data Mining and Bioinformatics in 2006 and the International Journal of Granular Computing,

Rough Sets and Intelligent Systems in 2008. He was a Research Scientist in the world-leading R&D centers such as Nortel Research Center, GTE Laboratories, and HP Laboratories. In 2001, he founded the DMW Software in Silicon Valley, CA. His current research interests include biomedical literature, data mining, bioinformatics, text mining, semantic Web mining and reasoning, rough set theory and application, information extraction, and information retrieval. He has authored or coauthored more than 160 peer-reviewed research papers in various journals, conferences, and books, including various IEEE/Association for Computing Machinery (ACM) Transactions, and coedited nine books/proceedings.

Mr. Hu is currently the Chair of the IEEE Computer Society Bioinformatics and Biomedicine Steering Committee and the IEEE Computational Intelligence Society Granular Computing Technical Committee (2007–2009). He has received a few prestigious awards including the 2005 National Science Foundation (NSF) Career Award (the most prestigious award from NSF to young faculty in USA), the Best Paper Award at the 2007 International Conference on Artificial Intelligence, the Best Paper Award at the 2004 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, the 2007 IEEE Bioinformatics and Bioengineering Outstanding Contribution Award, the 2006 IEEE Granular Computing Outstanding Service Award, and the 2001 IEEE Data Mining Outstanding Service Award. His research projects are funded by the National Science Foundation (NSF), U.S. Department of Education, and the PA Department of Health. He has obtained more than US$3.8 million research grants in the past four years as Principal Investigator (PI) or Co-PI.

E. K. Park received the Ph.D. degree in computer science from the Northwestern University, Evanston, IL.

He is currently a Professor of computer science at the University of Missouri at Kansas City, Kansas City. His current research interests include software engineering, software architectures, distributed systems, object-oriented methodology, computer communications and networks, optical networks, data mining and bioinformatics, optimizations, and information and knowledge management.

Xiaodan Zhang received the Bachelor’s degree in library and information science from the Northeast Normal University, Changchun, China, in 1997, and the Master’s degree in computer science from Jinan University, Guangzhou, China, in 2003. He is currently working toward the Ph.D. degree at the College of Information Science and Technology, Drexel University, Philadelphia, PA.

His current research interests include graph-based, model-based, and semantic-based text data mining.