Information Retrieval meets Gene Analysis

Hagit Shatkay[*]
Celera Genomics
45 West Gude Drive
Rockville, MD

Stephen Edwards
Rosetta Inpharmatics
12040-115th Ave. NE
Kirkland,WA

Mark Boguski
Visiting Investigator
Fred Hutchinson Cancer Research Center
Seattle, WA

Abstract

Current genomic research is characterized by immense volume of data, accompanied by a tremendous increase in the number of gene-related publications. This wealth of information presents a major data-analysis challenge. An ultimate goal is to understand the complex biological interrelationships among all discovered genes and proteins. Scanning the abundant literature available about each gene, and plenty of human expertise are currently required as a step towards meeting this goal. As has been recently noted by several research groups, automated systems for extracting relevant information from the literature can complement existing techniques, speed-up the analysis process, and greatly enhance our understanding of genetic processes. We present a new approach, based on probabilistic information retrieval, which uses the literature to establish functional relationships among genes on *a genome-wide scale*. Experiments applied to documents discussing yeast genes, and a comparison of the results to well-established gene functions, demonstrate the effectiveness of our approach.

1 Introduction

Advances in computational and biological methods during the last decade have remarkably changed the scale of genomic research. Sequencing machines and assembly algorithms enable sequencing complete genomes within months and even weeks. Automated gene finding methods [1, 4] expedite the identification of tens of thousands of genes within the sequenced DNA. Modern techniques such as DNA microarrays allow simultaneous measurements for all genes/proteins expressed in a living system. These methods, in turn, produce large quantities of *data*. When processed, it can provide actual *information* about gene expression patterns; for instance, which genes are expressed in various tissues, which ones are over/under expressed at the onset of a disease or during a specific phase of the cell development.

Still, the ultimate goal of conducting large-scale-biology is to translate these large amounts of *information* into *knowledge* of the complex biological processes governing the human body. Specifically we would like to understand the biological *function* of genes and proteins and the interrelationships between them. The hope is that once we *know* the biological roles of genes, proteins and their various interdependencies, we could understand and prevent, at the genomic level, undesirable processes, such as infection or tumor development, while encouraging desirable ones, such as normal growth and development.

The preceding era was characterized by the isolated research of only a few genes or proteins at a time. Such studies produced relatively little data that could be manually analyzed and turned into *knowledge* through careful and slow investigation. Obviously, this approach does not scale up to meet the current interpretation needs of abundant, newly produced data. Without suitable automated interpretation methods the full potential of the advanced technology, as a means to understand gene and protein function on a genomic scale, cannot be realized.

The ability to rapidly survey the published literature constitutes a necessary step in this interpretation process. It is also important for *designing* further large-scale experiments while generating hypotheses about plausible relationships among genes. Conducting a literature search about each gene separately is a tedious task, especially given that the genomic and proteomic literature is expanding in an unprecedented rate. Several techniques have recently been developed to address the need for expediting this search, as discussed in Section 2. Such methods are typically based on strong assumptions regarding the use of natural language in the literature, and on the use of common gene and protein nomenclature.

In contrast to other literature-based tools, the work presented here supports literature analysis on a *genome-wide scale*, *without* strong assumptions about explicit terminology and language usage. The hypothesis underlying our approach is that the function of many genes is (separately) described in the literature; by relating documents talking about well understood genes to documents discussing other genes, we can predict, detect and explain the functional relationships among the many genes that are involved in experiments. We do not attempt here to draw any information from the genomic data itself. Instead, we use a large database of abstracts, (a subset of *PubMed*[1]), as our information search space. Each gene is mapped to a respective document, roughly discussing the gene's biological func-

[*] Corresponding author.email:`hagit.shatkay@celera.com` This work was done while all the authors were at the National Center for Biotechnology Information, NLM, NIH. An earlier version appeared in the ISMB2000 conference.

[1] *PubMed* is maintained by NCBI/NLM and is accessible through *http://www.ncbi.nlm.nih.gov/PubMed.*

tion. The literature database is then searched for documents similar to the gene's document, using a probabilistic search method [13]. The resulting set of documents typically discusses the gene's function. As an integral part the algorithm produces an "executive summary" – a list of characteristic content bearing terms in the set of documents – found for each gene.

Next, we look for relationships among the genes. This we do through search for similarities among the resulting *sets* of documents. Since each set corresponds to a gene, we can map the similar document sets back to their originating genes and establish functional relationships among these genes. Thus we simultaneously address three goals:

- Finding functional relationships among genes, on a genome-wide scale.
- Obtaining the literature specifically relevant to the function of these genes.
- Producing a short term list characterizing the document set. This list suggests why the genes are considered relevant to each other, and what their biological function is.

The rest of the paper is organized as follows: Section 2 touches on current methods in large-scale expression analysis and surveys some of the developments in literature use for gene analysis; Section 3 describes our approach of searching the literature and its use for finding functional relationships among genes; Section 4 presents experiments and results over the set of well-studied yeast genes discussed by Spellman *et. al.* [14]. Our results demonstrate that automated information retrieval from the literature is a powerful tool for determining relationships among genes, and for assisting in both the design and the result-analysis of further large-scale experiments.

2 Related Work

Most analysis efforts, applied to large amounts of genomic data to date, concentrate on clustering genes, according to their expression patterns. The idea of such methods is to detect correlated expression patterns that may suggest regulatory and possible functional relationships. Traditional methods based on hierarchical clustering [14] or self-organizing maps [15], as well as more advanced stochastic clustering techniques [2, 12], have been shown to effectively group genes by the observed expression patterns. (See, for instance, work by Spellman *et al.* [14], on functional relationships in yeast genes.)

While clusters of simultaneously expressed genes often correlate with common function, this well-grounded approach has the following limitations as a stand-alone analysis tool:

- Functionally-related genes may demonstrate strong anti-correlation in their expression levels, (a gene may be *suppressed* to allow another to be expressed), thus clustered separately, blurring the existing relationship.
- Genes sharing similar expression profiles do not always share a function; they may be involved in distinct biological processes. as demonstrated below.
- Genes may play multiple roles in complex, interrelated biological processes. The stringent assignment of genes to single clusters by most clustering methods, potentially prevents the exposure of complex interrelationships among genes.
- Even when similar expression levels indeed correspond to similar functions, the functional relationships among genes in a cluster can not be determined from the cluster data alone. Explaining the formed clusters requires a lot of additional effort.

For example, careful analysis of the expression-based cluster CLB2 described by Spellman *et. al.* [14] reveals genes involved in several distinct cellular functions; CHS2, BUD8, and IQG1 are all involved in maintenance of the cell wall, while ACE2, ALK1, and HST3 are involved in nuclear events. Moreover, members of a common signaling pathway may play antagonistic roles, demonstrating anti-correlated expression levels. Thus, clusters based on expression profiles must be further analyzed, with respect to biological roles, before reliable conclusions about their biological function can be drawn.

In many cases, the information needed for such analysis can be found in the published literature. The conventional method for finding it, has been for individuals to search through the literature, gene by gene, or rely on their own knowledge of the biological processes. While this procedure can be effective on a very small scale, it does not scale up well to accommodate thousands of genes. Moreover, the advancement of genome sequencing techniques is accompanied by an overwhelming increase in the amount of literature discussing the discovered genes. This combined abundance of genes and literature, produces a major bottleneck for interpreting and planning genome-wide experiments.

To expedite the analysis process we propose a new automated method for exposing biological relationships among genes based on the biomedical literature. While our method can be used as a stand-alone tool for mining the literature, it complements the above approaches by providing literature-based explanations to the clusters and the relationships discovered directly from the expression data. We next survey current research aimed at automating literature mining in the context of gene analysis.

The prevailing on-line source for biomedical abstracts is the *PubMed* database. A typical search for relevant literature starts with a *boolean* query; The user provides a term (e.g. OLE1), or a boolean term-combination (e.g. OLE1 *and* lipid). The result is the set of *all* the abstracts in *PubMed* satisfying the query constraints. This form of query suffers several limitations:

1. The number of abstracts typically retrieved is *prohibitively large*.
2. A substantial part of the retrieved abstracts are *irrelevant* to the user's information need.
3. Many relevant abstracts *may not be retrieved*. For instance, abstracts discussing OLE1, using one of its aliases (e.g. DNA *repair protein* or *fatty-acid desaturase 1*) will not be retrieved.

Problem 2 above stems mostly from the well-known *polysemy* phenomenon; a word may have multiple meanings in different contexts. For instance when looking for the term "*CD*" we may retrieve all abstracts referring to "Cytosine Deaminase" in which we are interested but also all those discussing "Crohn's Disease" which are completely unrelated. On the other hand Problem 3, stems from *synonymy*, where a single concept may be discussed in various abstracts under different names.

The lack of uniformity in nomenclature used by authors further aggravates the problem. For instance, a search for abstracts about the gene AGP1 may not retrieve abstracts discussing this same gene under another name (e.g. YCC5). To improve the effectiveness, efficiency and accuracy of the navigation through the literature, several methods have been recently suggested, partly-automating the literature scanning process.

Most existing work focuses on automated *information extraction*, using curated lexica or natural language processing for identifying relevant phrases and facts in text, to assist in finding abstracts about a given gene or the relationships between specific genes. Leek [8], whose work is the earliest we are aware of in this domain, suggests using hidden Markov models (HMMs) for extracting sentences discussing gene localization on chromosomes. Craven *et. al.* [5, 10] have continued this line of work, presenting systems for extracting sentences discussing sub-cellular protein localization, training classifiers and an HMM to identify such sentences. Their methods require that a list of protein names and location descriptors are provided. Rindflesch *et. al.* [11], and more recently Friedman *et. al.* [6], propose methods based on parsing and thesauri use to extract facts about genes and proteins from documents. Blaschke *et. al.* [3] use a similar method, for extracting information about protein interaction from scientific text.

These methods have typically been applied to small and limited sample sets of documents/terms. They all require the user to specify a very accurate query in order to provide high-quality results. Most importantly, they all rely on strong assumptions about the use of natural language, such as terms typically used to indicate relationships, the typical sentence structure, gene/proteins names and their format, and the way these names are used within sentences. Such assumptions are not readily met throughout the abundant biological literature, (see [9] for an extensive discussion), thus limiting the scope within which these methods are effective.

A major step towards large-scale analysis was recently taken by Jenssen *et. al.* [7]. Using a *predefined* list of gene names and symbols, the authors executed a boolean search over *PubMed*, finding all abstracts in *PubMed* mentioning these genes. They then built a graph with the genes as nodes, and edges connecting genes that are mentioned in the same abstract. Weights on the edges represent the number of co-occurrences. The result is a very large network of genes related through the literature, and abstracts justifying each edge.

While the above is the most recent and extensive effort towards using the literature on a genome-wide scale, providing an unprecedented tool for researchers, it still suffers several limitations. First, as pointed out by the authors, the method relies on having a *complete list of gene names and synonyms*, it can only reveal relationships that are *already* reported in the literature, and does not attempt to detect new relations. Moreover, even while 60-70% of the found relationships (based on the authors' sample of 1000 analyzed pairs of genes) are correct in some respect, only a few of them (less than 10%) correspond to actual functional relationship. Another important point, pertaining to microarray experiments, is that over 30% of the relations detected by the system are *co-expression* relationships. These relationships may stem from papers reporting large-scale expression experiments, which are rich in co-occurring gene names. Researchers trying to biologically *explain* co-expression results in their own experiments, would typically look for biological relations among genes that are reported in the literature *independently* of the mere co-expression fact. Thus, in such scenarios, a drawback of the above method is that it finds relations among co-expressed genes based on their co-expression as formerly reported in the literature, without providing an independent way to *explain* this co-expression. The above is an artifact of the strong reliance of the method on co-occurrence of gene names.

As an alternative to using explicit gene names/synonyms while searching for "relationship sentences" or co-occurrences (known as *information extraction*), we shift our search focus from words and sentences to complete relevant abstracts. This kind of search is part of the field known as *information retrieval*. Moreover, we concentrate on the *similarity*-based query paradigm [16]. The user provides a sample relevant document and obtains other documents discussing the same subject matter. This mechanism does not depend on the choice of explicit query terms, but rather on the contents and quality of the example document. We use a recently developed probabilistic algorithm that, given an example document, finds a set of documents most relevant to it (*a theme*) and produces a *set of terms* summarizing the contents of this document set [13]. Other similarity-based methods for finding relevant documents do exist (see [16] and references within). However, these methods do not provide a list of summarizing terms which make the retrieved documents similar. The algorithm, as outlined in the next section, forms the basis to our approach.

3 Detecting Gene Relations and Functions through the Literature

Our approach is based on the hypothesis that many individual genes and their function are already discussed in the literature; A thorough analysis of the literature is a primary step for both design of experiments and results analysis following them. Thus, we shift our attention from experimental data to documents.

The actual search is conducted within a large[2], collection of *PubMed* abstracts, covering the literature relevant to the domain of discourse (*e.g. all* the abstracts in *PubMed* discussing yeast genes). We map each gene to a single abstract within the collection, discussing the gene's biological function. This abstract is treated as the gene's *representative*, and we call it the *kernel abstract* for that gene.

Applying the theme-finding algorithm, as described in Section 3.1 to each kernel, produces for each gene a body of related literature (20-50 abstracts bearing a common *theme*) based on the kernel abstract representing it, along with a list of terms that characterize the relevant literature. It is important to note that, in contrast to other literature-based methods, the retrieved abstracts are considered relevant *not* because they contain the *same gene name* as the one associated with the kernel abstract, but rather because they discuss the same *issues* (typically related to functionality) as those discussed in the kernel abstract. Once a set of abstracts is retrieved for each gene, we use an automated method to compare the abstract sets, and derive functional relationships among genes, as described in Section 3.2.

To use the theme-finding algorithm we first have to map the set of genes $\langle G_1, \ldots, G_N \rangle$ to a set of kernel abstracts $\langle K_1, \ldots, K_N \rangle$ (see top of Figure 3). Currently, kernel abstracts are obtained from the available curated literature about yeast genes (as explained in Section 4). The quality of the kernel abstracts strongly effects the quality of the results. Abstracts discussing experimental methods, rather than biological function, tend to draw other abstracts describing the same experimental methods, resulting in an abstract set not representative of the gene's function. In contrast, kernels discussing gene biology typically lead to high quality information about the function of related genes. The kernel selection process may be improved using machine-learning methods, so that each kernel abstract indeed represents the biology of its associated gene.

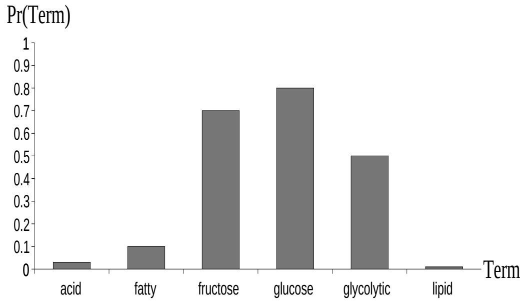

Figure 1: Typical term distribution for the *Nutrition* theme.

We next present the theme-finding algorithm for finding relevant abstracts (see [13] for a complete discussion), followed by a description of the second phase, in which relations are detected among the obtained *abstract collections*.

3.1 Finding Themes

The idea underlying our algorithm is that a set of documents sharing a coherent *theme* can be characterized by a set of probability distributions. For example, documents discussing genes responsible for *nutrition* during the cell-cycle, are likely to contain terms such as *fructose* or *glucose* and unlikely to contain the term *lipid*, as illustrated in Figure 1. More explicitly, our database, *DB*, is a set of documents represented as M-dimensional binary vectors, where M is the number of distinct terms[3] $\{t_1, \ \ldots, \ t_M\}$ in the database. The vector representation is commonly used in information retrieval systems. A document d is a vector $\langle d_1, \ d_2, \ \ldots, \ d_M \rangle$, where:

$$d_i = \delta_{di} \overset{\text{def}}{=} \begin{cases} 1 & \text{if } t_i \in d \ , \\ 0 & \text{otherwise} \ . \end{cases} \tag{1}$$

Presence/absence of terms in document d is viewed as a result of M independent Bernoulli events.

A *theme*, T, within the database DB, is a set of documents with a common subject. Documents sharing a common theme can be modeled as though they were generated through sampling from a common set of independent *Bernoulli* distributions representing the theme. Thus, a theme T, is modeled as a set of the following Bernoulli distributions. These distributions govern the occurrence of terms in the theme's documents:

- p_i^T — the probability that the term t_i occurs in a document d, given that d is a *theme* document: $p_i^T \overset{\text{def}}{=} \Pr(t_i \in d | d \in T)$.
- q_i^T — the probability that the term t_i occurs in a document d, given that d is an *off-theme* document: $q_i^T \overset{\text{def}}{=} \Pr(t_i \in d | d \notin T)$.

[2] On the order of several *tens of thousands* of abstracts.

[3] *Terms* consist of one or two words, excluding stop words.

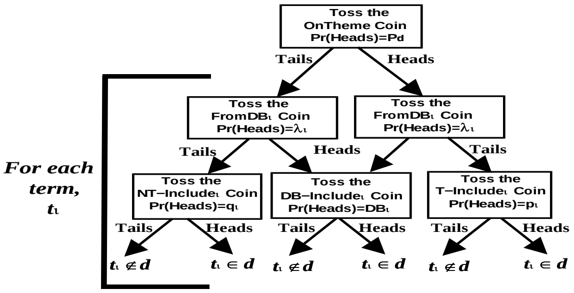
Toss the OnTheme Coin Pr(Heads)=Pd
Tails
Heads
Toss the FromDBι Coin Pr(Heads)=λι
Toss the FromDBι Coin Pr(Heads)=λι
Tails
Heads
Tails
For each term, tι
Toss the NT−Includeι Coin Pr(Heads)=qι
Toss the DB−Includeι Coin Pr(Heads)=DBι
Toss the T−Includeι Coin Pr(Heads)=pι
Tails
Heads
Tails
Heads
Tails
Heads
tι ∉ d
tι ∈ d
tι ∉ d
tι ∈ d
tι ∉ d
tι ∈ d

Figure 2: Stochastic Model for Generating Document d.

- DB_i — the probability that the term t_i occurs in a document d, given that d is a document in the database, regardless of its being an on-theme or an off-theme document: $DB_i \overset{\text{def}}{=} \Pr(t_i \in d | d \in DB)$.

The distribution DB_i models the possible arbitrary usage of terms in the language, without being strongly indicative of the main topic discussed. (e.g. the sentence *"I missed my flight"* is not particularly relevant to the topic *aviation*, despite the occurrence of the term *flight*).

Given a theme T, each document d has some a-priori probability, regardless of its content, to be a theme document. This probability is denoted by P_d where: $P_d \overset{\text{def}}{=} Pr(d \in T)$. Throughout this paper, we assume this parameter to be known and fixed for all documents, and do not attempt to estimate it here. (In the reported experiments $P_d = 0.01$ for all $d \in DB$.)

The last component of our model is the Bernoulli event representing the choice made for each term t_i, in each document d, whether it is to be generated according to the database probability, DB_i or according to the specific on/off-theme distribution. We denote this probability, for each term t_i, as λ_i.

Combining the above components, for a given theme T, we obtain the following generative model for each document $d \in DB$, as depicted in Figure 2:

- Decide, tossing a biased coin (*OnTheme* in the figure) with $Pr(H) = P_d$, whether d is in theme T.
- For each term t_i decide if t_i is distributed according to the general database distribution DB_i, by tossing a biased coin (*FromDB$_i$* in the figure) with $Pr(H) = \lambda_i$.
- For each term t_i decide if $t_i \in d$ by tossing one of the following biased coins:
    - ◇ The database coin for t_i, (*DB-Include$_i$*), if t_i is generated according to the *DB* distribution.
    - ◇ The on-theme coin for term t_i, (*T-Include$_i$*), whose $Pr(H) = p_i^T$, if $d \in T$ *and* t_i is generated according to p_i^T (*FromDB$_i$* came up *tails*).
    - ◇ The off-theme coin for t_i, (*NT-Include$_i$*), whose $Pr(H) = q_i^T$, if $d \notin T$ *and* t_i is generated according to q_i^T, (*FromDB$_i$* came up *tails*).

Note that for each document $d \in DB$, we *know* the terms it contains. The *missing information* is which documents are *theme* documents and which terms are generated from the general distribution, DB_i, as opposed to the theme-specific ones, p_i^T and q_i^T. Using a generative model allows us to explicitly represent and address such missing information. To support calculations within this model, we assume *conditional independence* between pairs of terms given the document containment in the theme, as well as independence among the hidden variables (representing the *missing information* above).

Under this framework, given a kernel document representing a gene, our task is to find a set of parameters $R = \{\{p_i^T\}, \{q_i^T\}, \{\lambda_i\}\}$[4], over all terms t_i in the database. Using a probabilistic Bayesian framework, we look for the parameters that maximize the likelihood of the documents in the database, $Pr(DB|R)$. These parameters are used to find the documents that are most likely to have been generated by sampling from these distributions. The latter documents are the ones focused on the theme represented by these distributions. In addition, we produce a set of terms characterizing this theme. These are the terms that have a high probability to occur in theme documents (high p_i^T) *and* a much lower probability to occur in documents outside the theme (high ratio p_i^T/q_i^T).

To estimate the Bernoulli parameters under missing information as described above, we use an Expectation Maximization algorithm (EM). This algorithm aims to maximize the likelihood of the database partition into theme/off-theme documents, given the Bernoulli parameters, based on the kernel document. The complete algorithm is described elsewhere [13], and we provide its outline here.

An EM algorithm starts by initializing the model parameters, (p^T, q^T, λ^T), based on some prior knowledge; We initially roughly approximate the Bernoulli parameters based on the kernel document and its comparison to the rest of the database[5]. The algorithm then alternates between:

- the *E-step* of computing the *expected likelihood*, of the documents to be in the theme, given the current parameter estimates, and
- the *M-step* of finding new model parameters that *maximize* the likelihood of the database partition into theme/off-theme documents given the parameters.

This iterative process is guaranteed, under mild conditions, to provide monotonically increasing convergence of the likelihood $\Pr(DB|R)$. We have proved that our algorithm

[4] Note that estimating DB_i is straightforward since all the required information is present in the database.

[5] Obviously, having multiple kernels to start from for a single theme would lead to a better initial estimate. Since obtaining informative kernels is currently hard, we make do with a single kernel.

is an instance of this family of algorithms, and follows this same pattern.

The algorithm is executed for each of the kernel documents, $\langle K_1, \ldots, K_N\rangle$, representing each of the genes, $\langle G_1, \ldots, G_N\rangle$, as illustrated in the top part of Figure 3. The result for each gene consists of:

- a list of the top 50 documents discussing the same theme as the kernel document, ordered by their degree of relevance to the theme, and
- a list of terms (keywords) constituting the theme, ordered by their degree of relevance to the theme.

Note that the keywords in the list are *not* merely the terms *most probable* to occur in the set of documents discussing the theme, but rather those that are much more probable to occur within this set than throughout the rest of the database. As shown in Section 4, this output in and of itself provides valuable support for gene analysis. Still, we further extend it in the next phase, to assist in finding biological relations among the genes.

3.2 Finding Functional Relations among Genes

Our primary assumption is that common relevant literature is a strong indicator of common functionality among genes; Genes which have similar lists of top ranking documents associated with them, share some common biological function described in the common literature. Our task is thus reduced to finding similarities among the *sets* of documents retrieved in the previous phase of the algorithm, and associating with each gene all other genes that have a similar document set.

To do this we use the *PubMed identifiers* associated with the abstracts, without examining the abstracts' contents. For each kernel we construct a *vector* characterizing it, based on the *abstracts* deemed relevant to it by the first phase of the algorithm (as described in Section 3.1). Note that this vector is different from the term-vector described in Section 3.1, as its entries represent *associated abstract identifiers* rather than *terms*. This vector representation can be used to rank for each kernel K_i, all the other kernels by their proximity to K_i in the kernel-vector space. Since each kernel corresponds to a gene, we can map the inter-related kernels back to their respective genes, and obtain a set of genes that are closely related. The method is illustrated at the bottom part of Figure 3 and is further described in the following paragraphs.

First, we construct the set of *PubMed* Identifiers of relevant abstracts, S_r, as follows:
Let N be the number of *kernel abstracts* used for representing genes[6]. We denote each kernel abstract by K_i where $1 \leq i \leq N$.
For each kernel, K_i, let L_i be the set of *PubMed* identifiers for the 50 top ranking abstracts associated with K_i. Formally: $L_i \overset{\text{def}}{=} \{ID_1^i \ldots ID_{50}^i\}$, where ID_j^i is the *PubMed* identifier of the j^{th} abstract ranked as relevant for kernel K_i.

Intuitively, if two distinct genes, G_i and G_j, represented by kernels K_i and K_j, have similar sets of relevant *PubMed* identifiers, L_i and L_j, then the literature relevant to these two genes has a lot in common. This in turn suggests that some roles and functions (typically discussed in the literature) are shared by these two genes.

The number of *PubMed* identifiers used for comparing abstract lists can be reduced by noting that identifiers occurring only within a single list L_i, do not contribute to the evaluation of any other list, L_j, as similar to L_i. Let *ID* denote a single *PubMed* identifier and $|ID|$ denote the total number of identifier lists, L_i, in which *ID* occurs. Our calculations need only take into account those identifiers for which $|ID| > 1$. Thus, S_r is defined to be the set of *PubMed* identifiers of all abstracts that are in the relevance list of at least two kernels. Formally:

$$S_r \overset{\text{def}}{=} \bigcup_{i=1}^{N} L_i - \{ID \mid |ID| \leq 1\} \quad . \tag{2}$$

We denote the number of *PubMed* identifiers in S_r, $|S_r|$, by M_r, and denote each *PubMed* identifier in S_r as ID^j where $1 \leq j \leq M_r$.

We can now represent each kernel abstract K_i, as an M_r-dimensional vector, $V_i \overset{\text{def}}{=} \langle v_i^1 \ldots v_i^{M_r}\rangle$ over S_r where v_i^j are defined as follows:

$$v_i^j = \delta_{ij} \overset{\text{def}}{=} \begin{cases} 1 & \text{if } ID^j \in L_i \\ 0 & \text{otherwise} \end{cases} \quad . \tag{3}$$

Each such kernel vector is then normalized.

To measure similarity between each pair of kernels, we calculate the *cosine coefficient* between their respective vectors. The cosine coefficient is often used in information retrieval to assess similarity between documents, where documents are viewed as term-vectors (see [16] and earlier references within). We use it in a *new, non-traditional way*, as our vector represents the kernels based on other *abstracts* rather than *terms*. Formally, the cosine coefficient between two vectors, V_i, V_k, whose respective lengths are $||V_i||, ||V_k||$ is defined as:

$$\cos(V_i, V_k) \overset{\text{def}}{=} \frac{\sum_{j=1}^{M_r} v_i^j \cdot v_k^j}{||V_i|| \cdot ||V_k||} \quad .$$

Since the vectors are normalized, their length is 1 and only the numerator needs to be calculated.

The closer V_i and V_j are to each other, the closer the coefficient is to 1. Hence, by calculating for each kernel vector, V_i, the cosine coefficients with respect to all other kernel vectors, V_j, we obtain for each kernel a ranking of how related it is to each of the other kernels, K_j. Recalling that

[6] The number of analyzed *genes* may *exceed* N since the same kernel abstract might discuss and represent more than a single gene.

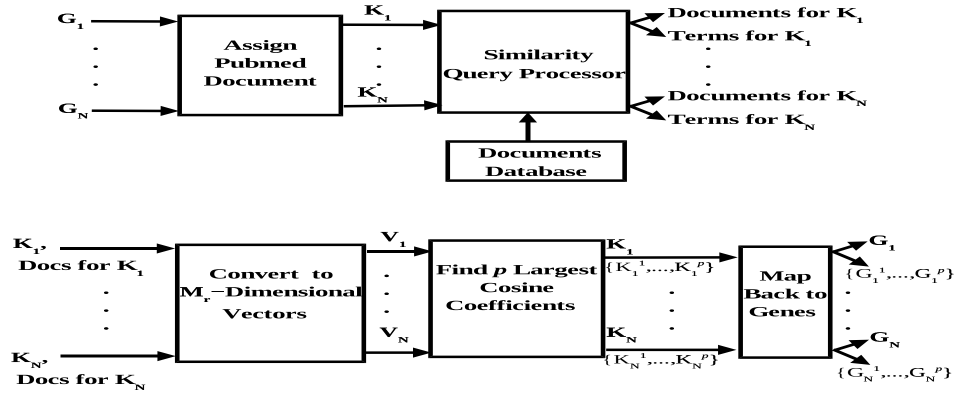

Figure 3: Finding Documents and Terms related to Genes (top), and Sets of Related Genes (bottom).

each kernel K_i corresponds in turn to a gene G_i, we obtain relationships among the respective genes. The reasoning for the assumed relationship is given by the lists of terms associated with the themes generated from the kernel abstracts, and thus the reasoning behind the suggested relationships can be easily checked.

The experiments and the results reported in the next section demonstrate the value of our methods for retrieving relevant abstracts and terms and for obtaining meaningful relationships among genes.

4 Experiments and Results

We apply the algorithms to yeast genes, and show how our methods indeed find relevant abstracts and provide useful summary terms. Moreover, we also discover meaningful relationships among the genes. We use the yeast DNA microarray testbed since the validity of our methods can only be assessed by comparison of the results with existing summaries of biological information; The SGD[7] and the YPD[8] databases as well as the functional analysis given by Spellman *et. al.* [14], are critical for rapid, objective evaluation of our results.

The rest of this section describes the experimental setting and reports the results obtained by applying our algorithm to the data. The quality of the results was verified through comparison to the functional groups of genes according to Spellman *et. al.* [14]. The portion of Spellman's table relevant to the results discussed here is shown in Table 1. The table categorizes the yeast genes according to their functionality (rows) and the phase in the cell-cycle in which they are expressed (columns).

4.1 Experimental Setting

The algorithms are applied to yeast genome data, in an attempt to find relevant literature and gene relations for the genes analyzed by Spellman *et. al.* [14]. The names of all the genes used by Spellman[9] were compared against the Saccharomyces Genome Database (SGD). Out of about 800 genes found by Spellman *et. al.* to be cell-cycle regulated, only 408 genes had curated *PubMed* references in the SGD, and our experiments concentrate on these 408 genes.

For each of the genes, the oldest reference cited in SGD is used as the *kernel abstract* corresponding to the gene. Since some of the closely related genes share the same reference, we obtain 344 distinct kernel abstracts. The database used in our experiments is a subset of *PubMed*, consisting of 33,700 abstracts discussing yeast genes. It includes about 2,250 abstracts deemed relevant for our 408 target genes by the SGD curators (about 86% of the total curated abstracts as of August, 1999). From all abstracts, we eliminated standard stop words, the Mesh term taggings typically associated with *PubMed* entries, as well as very common or extremely rare terms (those that occur in over 10% of the abstracts in the database or in 2 or fewer abstracts).

We applied the theme finding program, described in Section 3.1, to the 344 kernels, searching over the database of 33,700 abstracts. For each kernel, the program outputs a list of the top 50 related abstracts and a list of key words describing the contents of this relevant set.

The next phase, consists of looking for *relationships* among genes. For each of the kernels, the previous phase produced a list of 50 relevant abstracts. The first step of the current phase is to construct the set of relevant abstracts retrieved for *all* the kernels, eliminating duplicates. That is, even if an abstract is relevant to more than one kernel, it is still included in the set of relevant abstracts only once. We then eliminate all abstracts that are relevant to a single kernel only, as explained in Section 3.2. We are left with a set of 3063 abstracts that are relevant to 2 or more kernel abstracts, (this is the set S_r, defined in Eq. 2).

Each kernel is represented as a 3063-dimensional vector (Eq. 3), and the cosine coefficient is used to measure the similarity of each kernel to all the others. Each kernel is

[7] The Saccharomyces Genome Database available at *http://genome-www.stanford.edu/Saccharomyces/*.

[8] The Yeast Proteome Database, at *http://www.proteome.com/databases/index.html*.

[9] Available through the genome web site at Stanford, http://genome-www.stanford.edu/cellcycle/ .

Biological Function	**G1**	**S**	**G2**	**M**	**M/G1**
Replication Initiation	CDC45		ORC1	CDC47 CDC54 MCM2 MCM6	CDC6 CDC46 MCM3
Fatty Acids/ Lipids/ Sterols/ Membranes	EPT1 LPP1 PSD1 SUR1 SUR2 SUR4		AUR1 ERG3 LCB3	ERG2 ERG5 PMA1 PMA2 PMP1	ELO1 FAA1 FA A3 FAA4 FAS1
Nutrition	BAT2 PHO8		AGP1 BAT1 GAP1	DIP5 FET3 FTR1 MEP3 PFK1 PHO3 PHO5 PHO11 PHO12 PHO84 RGT2 SUC2 SUT1 VAP1 VCX1 ZRT1	AUA1 GLK1 HXT1 HXT2 HXT4 HXT7

Table 1: Yeast Genes: expression during cell-cycle and functionality. (Adapted from Spellman *et. al.* (1998))

then converted back to the gene(s) for which it was curated. The genes that are grouped as similar according to our method are compared with those grouped by functionality in Spellman's table (parts of which are shown in Table 1).

To quantitatively measure the validity of the keyword list assigned to each kernel, we compare each keyword to its associated function using a mini-thesaurus obtained from a panel of four independent yeast experts. Each functionality description listed in Spellman's table (such as *Secretion* or *Chromatin*) is associated with the terms judged most closely related to it according to the experts. Each expert received a list of the 22 function descriptions listed by Spellman *et al*, and a separate list of 330 alphabetically-sorted summary terms resulting from our program. The experts assigned to each term in the latter list, the functionality descriptors that they judged to be most related to it; non-specific terms were left unassigned. An example of two entries in the resulting thesaurus is shown in Table 2.

Function	**Associated Terms**
Chromatin	*chromatids, chromatin, chromosome, sister chromatids, telomere, telomeric*
Secretion	*acid phosphatase, coatomer, endoplasmic endoplasmic reticulum, er, golgi apparatus golgi complex, golgi transport, golgi, v snare*

Table 2: Example of thesaurus entries associating gene function with related terms.

For each gene, we compare its functionality according to Spellman with the functionality assigned by the panel to each of its key terms, counting how many of the key terms indeed correspond to the gene's functionality according to Spellman and how many do not. The results are described throughout the rest of this section.

4.2 Results

As described in Section 3.1, for each gene represented by a kernel abstract we obtain through the similarity query mechanism applied to the whole database:

1. A set of related abstracts.
2. A set of summarizing key terms.

In addition, from the set of related abstracts we obtain, for each kernel, through the vector representation and the cosine coefficient calculation, (described in Section 3.2), a set of related kernels. The latter kernels are mapped back to form a set of related genes.

To assess the value of the results obtained in the first phase we examine the set of summarizing keywords. We also examine the lists of related genes obtained in the second phase. The quality of the results is checked through a comparison with the functionality assigned to genes by Spellman *et. al.*[10], shown in Table 1. Since many of the genes in the experiment are not assigned any functionality by Spellman (120 out of the 344 kernels used) , we can only verify in this manner results for the ones whose functionality was determined by Spellman *et. al*. However, point-wise manual checking of the abstracts and genes associated with these 120 kernels not discussed by Spellman, shows that for many kernels the results do agree with the known biology and gene relationships.

An example of a typical successful search is shown in Table 3. The left column lists the *PubMed* identifiers for two kernel abstracts together with the genes they stand for and their respective functionality according to Spellman *et. al.* The second column lists, for each of the two kernels, the top 10 keywords associated with the retrieved set of abstracts, as determined by our algorithm. The third column lists the top genes associated with each kernel [11], based on the cosine coefficient. The fourth column lists the function of each gene according to Spellman *et. al*, as a validity check for our results. Since our experiment included more genes than listed in Spellman's table, some of the genes in the third column are not assigned functionality by Spellman. For these genes, (marked by '*' in the table), we found the functionality in YPD.

[10] The gene functionality assigned by Spellman *et. al.* is based on human judgment and expertise, rather than on an automated process.

[11] ELO1 has only 9 genes associated with it, since there were 9 non-zero cosine coefficients associated with its kernel.

Kernel (PMID, Gene,Function)	**Keywords**	**Assoc. Genes**	**Function**
8702485 **ELO1** **Fatty Acid/ Lipids/ Sterols/ Membranes**	fatty acid,	OLE1	(Fatty Acid, Sterol. Met.)*
	fatty,	FAA4	Fatty Acid/Lipids/Sterols/Membranes
	lipids,	FAA3	Fatty Acid/Lipids/Sterols/Membranes
	acid,	SUR2	Fatty Acid/Lipids/Sterols/Membranes
	grown,	FAA1	Fatty Acid/Lipids/Sterols/Membranes
	medium,	ERG2	Fatty Acid/Lipids/Sterols/Membranes
	carbon,	PSD1	Fatty Acid/Lipids/Sterols/Membranes
	synthase,	CYB5	(Fatty Acid, Sterol. Met.)*
	strains,	PGM1	(Carbohydrates Met.)*
	deficient		
7651133 **HXT7** **Nutrition**	hexose,	HXT1	Nutrition
	glucose uptake,	RGT2	Nutrition
	glucose conc.,	HXT4	Nutrition
	fructose,	HXT2	Nutrition
	glycolytic,	GLK1	Nutrition
	glucose,	SEO1	(Small Molecules Transport)*
	sugars,	PRB1	(Protein Degradation)*
	uptake,	AGP1	Nutrition
	aerobic,	ZRT1	Nutrition
	utilization	MIG2	(Carbohydrates Met.)*

Table 3: Example of a result obtained from two different kernel/gene using our algorithm, compared with functionality according to Spellman or YPD (YPD functionality denoted by *).

The table shows that except for two genes (PGM1 and PRB1) all of the genes found for these two kernels have a strong functional relationship to the genes represented by the kernels, and the keywords provide a strong indication of this biological function. (Note that the keywords are associated as a *set* with the whole kernel entry and not separated as one keyword per associated gene.) We note that PGM1 is involved in carbohydrates metabolism which is still functionally related to fatty acids metabolism. PRB1 is responsible for protein degradation, which is not related to nutrition. It is included in this set, since the abstract chosen for its kernel abstract discusses regulation of the enzyme PRB*1p* by glucose, rather than the biological function of PRB*1p*.

The results for about 100 out of the 220 kernels for which we had the Spellman-assigned functionality, closely resemble the ones demonstrated in Table 3 in the strong agreement with Spellman's cluster assignment and in the accurate description as given by the keywords learned by the similarity query algorithm.

As a *quantitative* measure, we calculated the average number of *correct* and *incorrect* keywords among the 5 top-ranking keywords associated with each of these kernels. A keyword occurring in a list for a specific gene (kernel), is considered *correct* if it appears in our thesaurus entry labeled by the same function as the one assigned to the gene by Spellman. If its thesaurus entry is labeled by a *different* function, it is considered *wrong*. If it was assigned no function by our panel of experts it is considered *non-descriptive*. An average of *3.27* out of the 5 top ranking keywords, were associated with the *correct* function, while only *1.12* out of the 5 were associated with the wrong function, and *0.61* out of the 5 were non-descriptive. The difference between the high rate of correct keyword assignment relative to the wrong and the non-descriptive assignment is highly statistically significant ($p \ll 0.005$, using the two-sample t-test).

For other kernels the groups of related genes contain many genes not assigned functionality by Spellman, which make the results harder to validate. Another set of cases, in which our results deviate from Spellman's functionality grouping of genes, are those for which the kernel abstract was not discussing the biology of the gene but rather the experimental method used to discover it. An example of such a result is given in Table 4.

In this case, the kernel abstract discusses the biology of the *technique* used for studying the MCM genes, involving autonomously replicating plasmids. The kernels considered similar to it also discuss such techniques. Thus, the commonality unifying the resulting set of genes, is that their curated abstracts all discuss manipulations within chromosomes, rather than gene biology. The keyword list (which highly ranks the terms *autonomous replication* and contains *leu2* and *ura3*), indicates that the theme underlying this set of abstracts and genes is not based on the biological function of the genes.

We are considering approaches for automating the kernel abstract selection, and expect them to lead to consistently good results. The excellent experience with the 100 high-quality kernel abstracts demonstrates that once a single informative abstract is given for a gene, many other quality abstracts about the related genes are automatically found, accompanied by a succinct characterization of their common functionality.

Kernel (PMID, Gene,Function)	**Keywords**	**Assoc. Genes**	**Function**
6323245 **MCM2,MCM3,MCM6** **Replication Init.**	ars,	CDC10	Site Selection/Morphogenesis
	autonom. replicating,	PHO3	Nutrition
	replicating sequence,	EST1	DNA Syn
	autonomously,	MIF2	Chromatin
	minichromosomes,	PHO12	Nutrition
	replicating,	POL2	DNA Syn.
	centrometric,	DHS1	DNA repair
	leu2,	SNQ2	*
	plasmids,	SMC3	Chromat. Cohes.
	ura3	EXG2	Cell Wall Synt.

Table 4: Example of a result obtained from an uninformative kernel, compared with functionality according to Spellman.

5 Conclusions and Ongoing Work

The information-retrieval approach presented here has four clear advantages:

1. It is an effective way for detecting putative relationships among genes. These can then be verified through well-targeted experiments.
2. It provides the relevant literature for analyzing the experimental results.
3. It generates summarizing terms explaining the discovered relationships. This summary can help explain and evaluate the relationships found by directly clustering the expression levels.
4. It is independent of natural language usage and nomenclature issues, as it does not search for explicit gene names or statements about their relationships.

We also note that our method does not use any pre-clustering of the genes among which it is looking to find relationships.

Thus, our method can be used both for generating hypotheses *prior* to the experiments, and for *post-experimental interpretation* of the results. Given a *functionally descriptive* kernel abstract, our program can provide insight into gene functional groupings, similar to that currently obtained through laborious, manual literature surveys relying on human expertise. Obviously, our method can not ascribe function to genes which have not yet been studied. However, by pointing out commonalities between abstracts discussing distinct genes, it can uncover functional relationships among known genes which heretofore have gone unnoticed.

The main current limitation of our technique is that of obtaining functionally descriptive kernel abstracts. We are studying machine-learning methods that can assist in automating the kernel selection process. Given a good source of kernels, we expect that utilizing *multiple kernels* for each gene, rather than a single kernel, would provide a better initialization to the EM algorithm and further improve the results. Another promising direction is that of extending the vectors representation of abstracts to include gene expression values, simultaneously searching for related abstracts and similarly expressed genes.

The methods described here complement the analysis techniques currently applied to microarray data. Combining our approach with other emerging analysis methods, can greatly expedite the tedious task of analyzing the vast amounts of data generated from genome-wide experiments.

Acknowledgments

We are grateful to Jan Fassler, Ken Katz, Steven Sullivan and Tyra Wolfsberg for the time and effort they have put into assigning functional tags to terms.

References

[1] V. Bafna and D. H. Huson. The conserved exon method for gene finding. *Proc. of the ISMB Conference*, pp. 3–12, 2000.

[2] A. Ben-Dor et al. Clustering gene expression patterns. *Journal of Computational Biology*, 6(3/4):281–297, 1999.

[3] C. Blaschke et al. Automatic extraction of biological information from scientific text: Protein-protein interactions. *Proc. of the ISMB Conference*, pp. 60–67, 1999.

[4] C. B. Burge and S. Karlin. Finding the genes in a genomic DNA. *Current Opinion in Structural Biology*, 8:346–354, 1998.

[5] M. Craven and J. Kumlien. Constructing biological knowledge bases by extracting information from text sources. *Proc. of the ISMB Conference*, pp. 77–86, 1999.

[6] C. Friedman et al. Genies: a natural-language processing system for the extraction of molecular pathways from journal articles. *Proc. of the ISMB Conference*, pp. S74–S82, 2001.

[7] T.-K. Jenssen et al. A literature network of human genes for high-throughput analysis of gene expression. *Nature Genetics*, 28:21–28, May 2001.

[8] T. R. Leek. Information extraction using hidden Markov models. M.Sc. thesis, Dept. of Computer Science, University of Califonia, San Diego, 1997.

[9] H. Pearson Biology's Name Game. *Nature*, 411:631-632, June 2001.

[10] S. Ray and M. Craven. Representing sentence structure in hidden Markov models for information extraction. *Proc. of the International Joint Conference on Artificial Intelligence*, 2001.

[11] T. C. Rindflesch et al. Edgar: Extraction of drugs, genes and relations from the biomedical literature. *Proc. of the Pacific Symposium on Biocomputing*, 2000.

[12] R. Sharan and R. Shamir. Click: A clustering algorithm with applications to gene expression analysis. *Proc. of the ISMB Conference*, pp. 307–316, 2000.

[13] H. Shatkay and W. J. Wilbur. Finding themes in Medline documents: Probabilistic similarity search. *Proc. of the IEEE Conference on Advances in Digital Libraries*, pp. 183–192, 2000.

[14] P. T. Spellman et al. Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces Cerevisiae by microarray hybridization. *Molecular Biology of the Cell*, 9:3273–3297, 1998.

[15] P. Tamayo et al. Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *Proc. of the National Academy of Science*, 96:2907–2912, 1999.

[16] I. H. Witten et al. *Managing Gigabytes, Compressing and Indexing Documents and Images*. Morgan-Kaufmann, 2^{nd} edition, 1999.