Information Retrieval Meets Gene Analysis

Hagit Shatkay, *Celera Genomics*
Stephen Edwards, *Rosetta Inpharmatics*
Mark Boguski, *Fred Hutchinson Cancer Research Center*

This method uses the scientific literature to establish functional relationships among genes on a genome-wide scale. Experiments on documents discussing yeast genes demonstrate its potential.

Advances in computational and biological methods during the last decade have remarkably changed the scale of genome research. Sequencing machines and assembly algorithms enable sequencing complete genomes within months and even weeks. Automated gene-finding methods[1,2] expedite the identification of tens of thousands of genes in the sequenced DNA. Modern techniques such as DNA microarrays allow simultaneous measurements for all genes and proteins expressed in a living system. These methods, in turn, produce large quantities of data. When processed, this data can provide information about gene expression patterns—for instance, which genes are expressed in various tissues or which ones are overexpressed or underexpressed at a disease's onset or during a specific phase of 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 their interrelationships. The hope is that once we know these things, we can 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 isolated research, focused on only a few genes or proteins at a time. Such studies produced relatively little data, which researchers manually analyzed and turned into knowledge through slow, careful 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 tedious, especially given the unprecedented rate at which the genomic and proteomic literature is expanding. Several techniques have recently been developed for expediting this search. Such methods typically depend on strong assumptions regarding the use of natural language and the availability of a common gene-related nomenclature. (For more on other research for finding gene relationships, see the sidebar "Gene clustering." For more on other techniques to expedite literature search, see the sidebar "Mining the Literature.")

Unlike other literature-based tools, the work we present here supports literature analysis on a genome-wide scale, without making strong assumptions about explicit terminology and language use. 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 involved in experiments.

Detecting gene relations and functions

Our search space is a large collection (several tens of thousands) of PubMed abstracts covering literature relevant to the domain of interest. For instance,

1094-7167/02/$17.00 © 2002 IEEE

Gene Clustering

Most genome-scale analysis efforts to date concentrate on clustering genes according to their expression patterns. Such methods try to detect correlated expression patterns that may suggest regulatory and possible functional relationships. Traditional methods based on hierarchical clustering[1] or self-organizing maps,[2] as well as more advanced stochastic clustering techniques,[3,4] can effectively group genes by the observed expression patterns. (See, for instance, work by P.T. Spellman and his colleagues on functional relationships in yeast genes.[1])

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

- Functionally related genes may demonstrate strong anticorrelation in their expression levels (a gene may be suppressed to allow another to be expressed), thus being 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 cannot be determined from the cluster data alone. Explaining the formed clusters requires much additional effort.

For example, careful analysis of the expression-based cluster CLB2 described by Spellman and his colleagues[1] reveals genes involved in several distinct cellular functions. CHS2, BUD8, and IQG1 are all involved in cell wall maintenance, while ACE2, ALK1, and HST3 are involved in nuclear events. Moreover, members of a common signaling pathway may play antagonistic roles, demonstrating anticorrelated expression levels. Thus, clusters based on expression profiles require further analysis, with respect to biological roles, before reliable conclusions about their biological function can be drawn.

In many cases, the published literature can provide the information for such analysis. The conventional method for finding it has been for individuals to search the literature, gene by gene, or rely on their own knowledge of the biological processes. Although 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 abundance of both genes and literature produces a major bottleneck for interpreting and planning genome-wide experiments.

To expedite analysis, we propose a new automated method for exposing biological relationships among genes based on the biomedical literature. Our method, which is described in the main article, can serve as a stand-alone tool for mining the literature. However, it also complements the previous methods by providing literature-based explanations for the clusters and the relationships discovered directly from the expression data.

References

1. 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*, vol. 9, no. 12, Dec. 1998, pp. 3273–3297.
2. P. Tamayo et al., "Interpreting Patterns of Gene Expression with Self-Organizing Maps: Methods and Application to Hematopoietic Differentiation," *Proc. Nat'l Academy of Science*, vol. 96, no. 6, 16 Mar. 1999, pp. 2907–2912.
3. A. Ben-Dor et al., "Clustering Gene Expression Patterns," *J. Computational Biology*, vol. 6, nos. 3–4, 1999, pp. 281–297.
4. R. Sharan and R. Shamir, "Click: A Clustering Algorithm with Applications to Gene Expression Analysis," *Proc. 8th Int'l Conf. Intelligent Systems for Molecular Biology*, AAAI Press, Menlo Park, Calif., 2000, pp. 307–316.

the collection may consist of all the abstracts in PubMed discussing yeast genes. (For more on PubMed, see the "Mining the Literature" sidebar.) We map each gene to a single abstract within the collection, discussing the gene's biological function. We treat this abstract as the gene's representative and call it the *kernel abstract* for the gene.

Applying the theme-finding algorithm (described below) to each kernel produces, for each gene, a body of related literature (20 to 50 abstracts bearing a common *theme*) based on its kernel abstract, along with a list of terms that characterize the relevant literature. In contrast to other literature-based methods, ours considers the retrieved abstracts 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. 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.

In order to apply the theme-finding algorithm, we first have to map the set of genes, $\{G_1, \ldots, G_N\}$, to a set of kernel abstracts $\{K_1, \ldots, K_N\}$ (see Figure 3a). For the experiments described later, kernel abstracts are obtained from the available curated literature about yeast genes. The quality of the kernel abstract strongly affects the quality of the results. Abstracts discussing experimental methods, rather than biological function, tend to draw other abstracts describing the same experimental methods. This results in an abstract set not representative of the gene's function. In contrast, kernel abstracts 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.

Finding themes and keywords

The idea underlying our theme-finding algorithm is that a set of documents (abstracts, in the case of PubMed) sharing a coherent theme can be characterized by a set of probability distributions. For example, documents discussing genes responsible for nutrition in yeast, 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, $\{t_1, t_2, \ldots, t_M\}$, in the database.

(Terms comprise one or two words, excluding stop words such as prepositions and pronouns.) The vector representation is extensively used in information retrieval systems. A document d is a vector $\langle d_1, d_2, \ldots, d_M \rangle$, where

$$d_i = \delta_{di} \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if term } t_i \in d, \\ 0 & \text{otherwise} \end{cases} . \qquad (1)$$

We view the presence or absence of terms in document d 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. These distributions govern the occurrence of terms in the theme's documents:

- p_i^T —the probability that term t_i occurs in document d, given that d is a theme document:
$$p_i^T \stackrel{\text{def}}{=} \Pr(t_i \in d \mid d \in T)$$
- q_i^T —the probability that t_i occurs in d, given that d is an *off-theme* document:
$$q_i^T \stackrel{\text{def}}{=} \Pr(t_i \in d \mid d \notin T)$$
- DB_i —the probability that t_i occurs in d, given that d is a document in the database, regardless of its being on-theme or off-theme:
$$DB_i \stackrel{\text{def}}{=} \Pr(t_i \in d \mid d \in DB)$$

The distribution DB_i models the possible arbitrary use of terms in the language, without strongly indicating the main topic discussed. (For example, 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. We denote this probability as P_d, where

$$P_d \stackrel{\text{def}}{=} \Pr(d \in T) .$$

Throughout this article, we assume this parameter is known and fixed for all documents and do not attempt to estimate it. (In the experiments we report here, $P_d = 0.01$ for all $d \in DB$.)

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

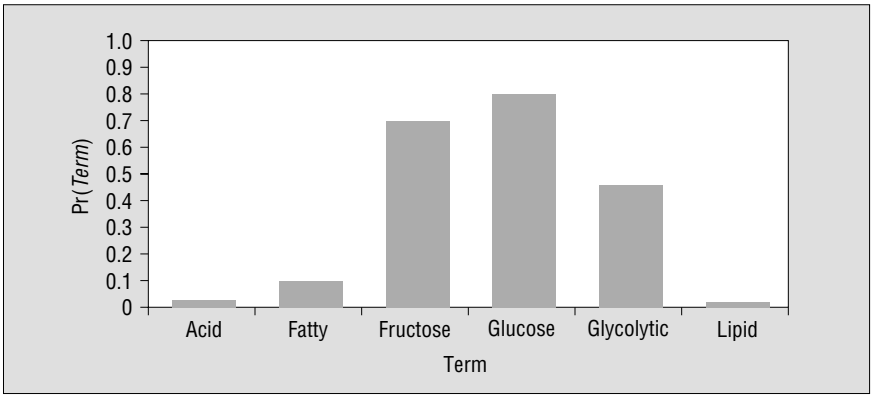

Figure 1. A typical term distribution for the "Nutrition" theme.

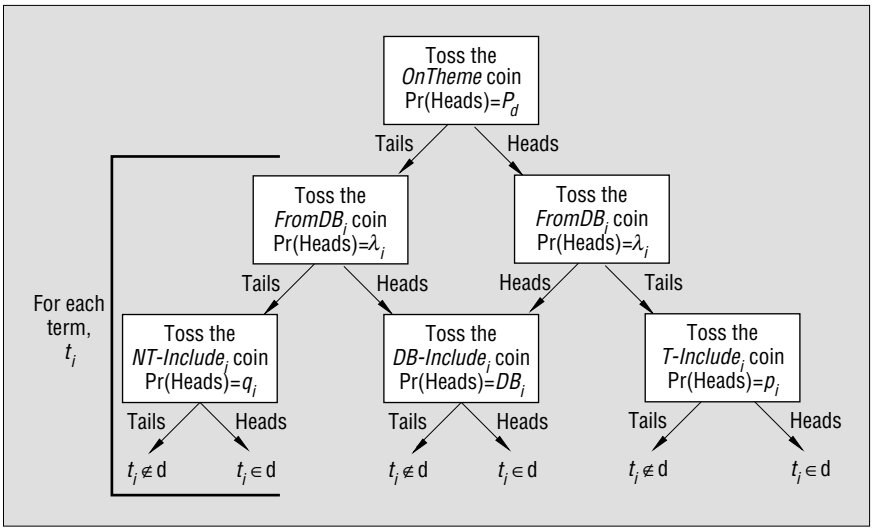

Figure 2. A stochastic model for generating a document *d*.

Combining these components, for a given theme T, we obtain the following generative model for each document $d \in DB$, as depicted in Figure 2. First, decide, tossing a biased coin (*OnTheme* in Figure 2) with $\Pr(H) = P_d$, whether d is in theme T. Second, 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 $\Pr(H) = \lambda_i$. Finally, for each term t_i, decide if $t_i \in d$ by tossing one of three biased coins:

- If t_i is generated according to the DB distribution, toss the database coin for t_i, (*DB-Include*$_i$).
- If $d \in T$ and t_i is generated according to p_i^T (*OnTheme* came up heads; *FromDB*$_i$ came up tails), toss the on-theme coin for term t_i, (*T-Include*$_i$), with $\Pr(H) = p_i^T$.
- If $d \notin T$ and t_i is generated according to q_i^T (both *OnTheme* and *FromDB*$_i$ came up tails), toss the off-theme coin for t_i, (*NT-Include*$_i$), with $\Pr(H) = q_i^T$.

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. The use of a generative model lets us explicitly represent and address such missing information. To support calculations within this model, we assume both conditional independence between pairs of terms given the document containment in the theme, and independence among the hidden variables (representing the missing information above).

Under this framework, given a kernel representing a gene, we must find a set of para-

Mining the Literature

A tremendous amount of knowledge about genes and proteins, such as their biological function and their role in development and disease, has been acquired and published in the literature. Obtaining this already existing knowledge from the literature is crucial for advancing our understanding of biological processes and further planning and interpreting experiments.

The prevailing online source for biomedical abstracts is the PubMed database (www.ncbi.nlm.nih.gov/PubMed), maintained by the National Center for Biotechnology Information in the National Library of Medicine. A typical search for relevant literature in PubMed starts with a Boolean query; the user provides a term (for example, "OLE1") or a Boolean term combination (for example, "OLE1 and lipid"). The result is a set of all the PubMed abstracts satisfying the query constraints.

This form of query suffers several limitations:

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

The second limitation 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, the third limitation stems from *synonymy*, where a single concept is discussed in several abstracts under different names.

The lack of uniformity in the 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 (for example, YCC5).

To improve the effectiveness, efficiency, and accuracy of navigation through the literature, researchers have recently suggested several methods that partly automate literature mining.

Most existing work focuses on automated *information extraction*. Such methods use curated lexica or natural language processing for identifying relevant phrases and facts in text, to help find abstracts about a gene or the relationships between specific genes. T.R. Leek, whose work is the earliest we are aware of in this domain, suggests using hidden Markov models (HMMs) to extract sentences discussing gene localization on chromosomes.[1] M. Craven and J. Kumlien have continued this line of work, presenting systems for extracting sentences discussing subcellular protein localization, by training classifiers and an HMM to identify such sentences.[2,3] Their methods require a list of protein names and location descriptors. T.C. Rindflesch and his colleagues,[4] and more recently C. Friedman and her colleagues,[5] propose methods based on parsing and thesauri use to extract facts about genes and proteins from documents. C. Blaschke and his colleagues use a similar method to extract information about protein interaction.[6]

These methods have typically been applied to small, limited sample sets of documents or terms. To obtain high-quality results, users must specify a very accurate query. Most importantly, these methods rely on strong assumptions about the use of natural language, such as terms typically used to indicate relationships, the typical sentence structure, gene or protein names and their format, and the way these names are used within sentences. Such assumptions do not readily apply throughout the abundant biological literature,[7] thus limiting these methods' effectiveness.

A major step towards large-scale analysis was recently taken by T.-K. Jenssen and his colleagues.[8] Still relying on nomenclature, the authors used a predefined list of gene names and symbols and executed a Boolean search over PubMed, finding all abstracts 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.

Although this method offers an unprecedented tool for researchers, it suffers several limitations. As the authors pointed out, it requires a complete list of gene names and synonyms, it reveals only relationships already reported in the literature, and it does not attempt to detect new relations. Moreover, although 60 to 70 percent of the found relationships (based on the authors' sample of 1,000 analyzed pairs of genes) are correct in some respect, only a few of them (less than 10 percent) correspond to an actual functional relationship. Another important point, pertaining to microarray experiments, is that over 30 percent of the detected relations are coexpression relationships. These relations may stem from papers reporting large-scale expression experiments, which are rich in co-occurring gene names. Researchers trying to biologically explain coexpression results in their own experiments would typically look for biological relations among genes that are reported in the literature independently of the mere coexpression fact. Hence, in such scenarios, this method's drawback is that it finds relations among coexpressed genes on the basis of their coexpression as reported in the literature, without providing an independent way to explain this co-expression. This drawback is an artifact of the method's strong reliance on co-occurrence of gene names.

As an alternative to information extraction methods that use explicit gene names and synonyms while searching for "relationship sentences" or co-occurrences, we shift our search focus from words and sentences to complete relevant abstracts. Such a search falls in the well-studied field of *information retrieval*.[9] Moreover, we concentrate on the similarity-based query paradigm.[10] 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[11] that, given an example document, finds a set of documents most relevant to it and produces a set of terms summarizing the contents of the document set.

Other similarity-based methods for finding relevant documents do exist

(see *Managing Gigabytes: Compressing and Indexing Documents and Images*[10] and the references therein); however, these methods do not provide a list of summarizing terms accounting for the similarity among the retrieved documents. (For more on the algorithm, see the section "Finding themes and keywords" in the main article.)

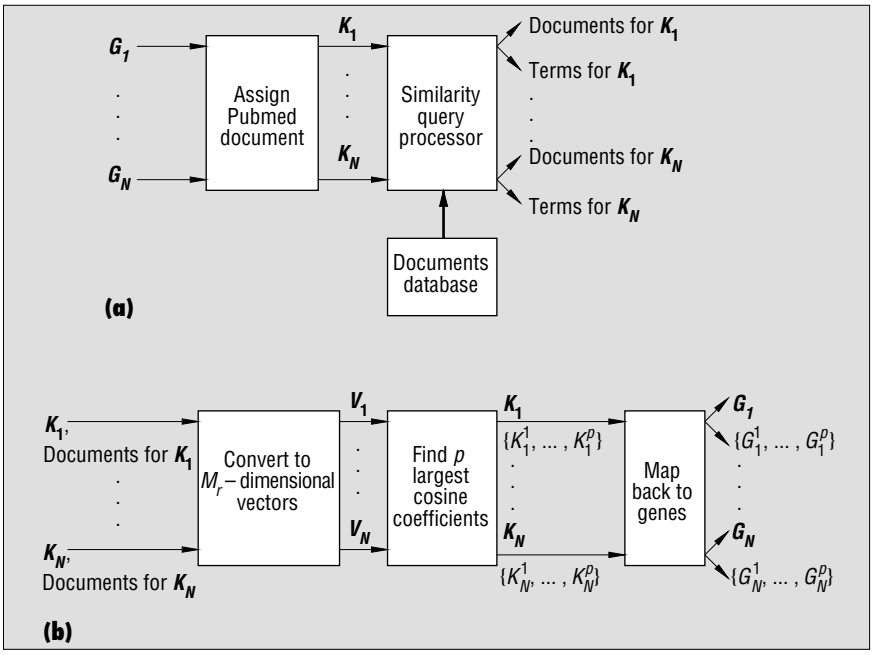

Figure 3. Finding (a) documents and terms related to genes and (b) sets of related genes.

meters

$$R = \left\{\left\{p_i^T\right\}, \left\{q_i^T\right\}, \{\lambda_i\}\right\}$$

over all terms t_i in the database. (Estimating DB_i is straightforward because the database contains the required information.) 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 the theme distributions. The latter documents are the ones focused on the kernel's theme as represented by the distributions.

Additionally, we produce the list of *keywords*—terms characterizing the theme. These are the terms that have a high probability of occurring in theme documents (high p_i^T) and a much lower probability of occurring in off-theme documents (a high ratio p_i^T / q_i^T).

To estimate the Bernoulli parameters under missing information, we use an Expectation Maximization algorithm (EM).[3] This algorithm aims to maximize the likelihood of the database partition into theme and off-theme documents, given the Bernoulli parameters, based on the kernel document. The complete algorithm is described elsewhere,[4] and we provide its outline here.

An EM algorithm starts by initializing the model parameters, (p^T, q^T, λ^T), on the basis of some prior knowledge. We initially roughly approximate the Bernoulli parameters on the basis of the kernel document and its comparison to the rest of the database. (Obviously, having multiple kernels 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). The algorithm then alternates between two steps:

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

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

The algorithm executes for each kernel document, $\{K_1, \ldots, K_N\}$, representing each gene, $\{G_1, \ldots, G_N\}$, as Figure 3a illustrates. The result for each gene consists of two lists:

- The top 50 documents discussing the same theme as the kernel document, ordered by their degree of relevance to the theme
- The keywords constituting the theme, ordered by their degree of relevance to the theme

References

1. T.R. Leek, *Information Extraction Using Hidden Markov Models*, master's thesis, Dept. of Computer Science, Univ. of California, San Diego, 1997.
2. M. Craven and J. Kumlien, "Constructing Biological Knowledge Bases by Extracting Information from Text Sources," *Proc. 7th Int'l Conf. Intelligent Systems for Molecular Biology*, AAAI Press, Menlo Park, Calif., 1999, pp. 77–86.
3. S. Ray and M. Craven, "Representing Sentence Structure in Hidden Markov Models for Information Extraction," *Proc. 17th Int'l Joint Conf. Artificial Intelligence*, Morgan Kaufmann, San Francisco, 2001, pp. 1273–1279.
4. T.C. Rindflesch et al., "EDGAR: Extraction of Drugs, Genes and Relations from the Biomedical Literature," *Proc. Pacific Symp. Biocomputing*, 2000, pp. 514–525; www-smi.stanford.edu/projects/helix/psb00/rindflesch.pdf.
5. C. Friedman et al., "GENIES: A Natural-Language Processing System for the Extraction of Molecular Pathways from Journal Articles," *Bioinformatics*, vol. 17, no. 90,001, June 2001, pp. S74–S82.
6. C. Blaschke et al., "Automatic Extraction of Biological Information from Scientific Text: Protein–Protein Interactions," *Proc. 7th Int'l Conf. Intelligent Systems for Molecular Biology*, AAAI Press, Menlo Park, Calif., 1999, pp. 60–67.
7. H. Pearson, "Biology's Name Game," *Nature*, vol. 411, no. 6,838, June 2001, pp. 631–632.

References continued on page 53

Table 1. Yeast genes: Functionality and expression during the cell cycle (adapted from Spellman and his colleagues[3]).

	Cell cycle phase				
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 FAA3 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

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 the results demonstrate, this output, in and of itself, is valuable for gene analysis. Still, we further extend it in the next phase, to assist in finding biological relationships among the genes.

Finding functional relations among genes

Our primary assumption is that common relevant literature strongly indicates common functionality among genes. That is, genes that have similar lists of top-ranking documents associated with them share some common biological function described in the common literature. Therefore, our task reduces to finding similarities among the sets of documents retrieved in the previous phase of the algorithm, and to associating with each gene all other genes that have a similar document set.

To do this, we use the *PubMed identifiers* (PMIDs) associated with the abstracts, without examining the abstracts' contents. For each kernel, we construct a characterizing vector, based on the abstracts deemed relevant to it. This abstract-based vector is fundamentally different from the term-based vector we described in the previous section, as its entries represent associated abstract identifiers rather than terms. We can use this vector representation to rank, for each kernel K_i, all the other kernels by their proximity to K_i in the abstract-based vector space. Because each kernel corresponds to a gene, we can map the interrelated kernels back to their respective genes and obtain a set of closely related genes. Figure 3b illustrates the method, and we now examine this phase in more detail.

First, we construct the set of PMIDs of relevant abstracts, S_r, as follows:

Let N be the number of kernels. (The number of analyzed genes may exceed N because the same kernel may discuss and represent more than one gene.) We denote each kernel by K_i, where $1 \le i \le N$.

For each kernel, K_i, let L_i be the set of PMIDs for the 50 top-ranking abstracts associated with K_i. Formally,

$$L_i \stackrel{\text{def}}{=} \left\{ ID_1^i \ \ldots \ ID_{50}^i \right\},$$

where ID_j^i is the PMID of the jth 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 PMIDs, L_i and L_j, then the literature relevant to these genes has much in common. This suggests that these two genes share some roles and functions (typically discussed in the literature).

Because identifiers occurring in only a single list, L_i, do not contribute to the evaluation of any other list, L_j, as similar to L_i, we can reduce the number of PMIDs we use. Let ID denote a single PMID and $|ID|$ denote the total number of identifier lists, L_i, in which ID occurs. Our calculations need take into account only those PMIDs for which $|ID| > 1$. Hence, we define S_r as the set of PMIDs of all abstracts that are in the relevance list of at least two kernels. Formally,

$$S_r \stackrel{\text{def}}{=} \bigcup_{i=1}^{N} L_i - \left\{ ID \,\middle|\, |ID| \le 1 \right\}. \qquad (2)$$

We denote the number of PMIDs in S_r, $|S_r|$, by M_r, and denote each PMID in S_r as ID^j, where $1 \le j \le M_r$.

We can now represent each kernel K_i as an M_r-dimensional vector, $V_i \stackrel{\text{def}}{=} \left\langle v_i^1 \ldots v_i^{M_r} \right\rangle$ over S_r, where v_i^j is

$$v_i^j = \delta_{ij} \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } ID^j \in L_i \\ 0 & \text{otherwise} \end{cases}. \qquad (3)$$

We then normalize each such vector.

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 term vectors represent the documents.[5,6] We use it in a new, nontraditional way—our vectors represent 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

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

Because the vectors are normalized, their length is 1, and we need to calculate only the numerator.

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 each kernel K_i corresponds to a gene G_i, we obtain suggested relationships among the respective genes. The keywords associated with the themes generated from the kernels provide the reasoning for these relationships, which we can easily check.

Experiments, validation, and results

We apply our method to yeast genes, to show that it indeed finds relevant abstracts, useful keywords, and meaningful gene relationships. We use the yeast DNA microarray testbed because the validity of our method can only be assessed by comparing its results with existing summaries of biological information. The Saccharomyces Genome Database (SGD, http://genome-www.stanford.edu/Saccharomyces), the Yeast Proteome Database (YPD, www.proteome.com/databases/index.html), and the functional analysis of yeast genes expressed during the cell cycle given by P.T. Spellman and his colleagues[7] are critical for rapid, objective evaluation of our results. 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).

The *cell cycle* is the sequence of steps a cell undergoes between two mitotic events. It comprises the DNA *synthesis* and *mitosis* phases (denoted S and M, respectively), a preparatory *gap phase* preceding each of these phases (G1 and G2, respectively), and a transitional phase following mitosis (M/G1).

Experimental setting

Our algorithms are applied to yeast genome data to find relevant literature and gene relations for the genes analyzed by Spellman.[7] We compared the names of all the genes used by Spellman (available at http://genome-www.stanford.edu/cellcycle) against the SGD. Of the approximately 800 genes found by Spellman and his colleagues to be cell-cycle regulated, only 408 genes had curated PubMed references in the SGD. Our experiments concentrate on these 408 genes.

For each gene, we use the abstract of its oldest reference cited in SGD as its kernel. Because some of the closely related genes share the same reference, we obtain 344 distinct kernels. 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 (approximately 86 percent of the total curated abstracts as of Aug. 1999). From all abstracts, we eliminated standard stop words, the Mesh tags (Medical subject headings) typically associated with PubMed entries, as well as very common or extremely rare terms (those occurring in more than 10 percent of the abstracts or in two or fewer abstracts in the database). We applied the theme-finding algorithm to the 344 kernels. For each kernel, the program outputs the list of the top 50 related abstracts and the keywords describing this set's contents.

To discover relationships among the genes, based on the results of the previous phase, we first construct the set of relevant abstracts retrieved for all the kernels, eliminating duplicates. That is, even if an abstract was relevant to more than one kernel, it is still included only once in the set. We then eliminate all the abstracts that were relevant to only a single kernel. This leaves a set of 3,063 abstracts that are relevant to two or more kernels (this is the set S_r, defined in Equation 2).

We represent each kernel as a 3,063-dimensional vector (see Equation 3) and use the cosine coefficient to measure the similarity of each kernel to all the others. We then convert each kernel back to the gene or genes for which it was curated. The result is clusters of genes, grouped as similar based on the similarity of their respective abstract sets.

As a qualitative assessment, we compare the genes grouped as similar according to our method, to those grouped by biological function in Spellman's table (parts of which are shown in Table 1). Since the functions shown do not involve any genes that are expressed during the S phase, the respective column in the table is blank.

To quantitatively measure the validity of the keyword list assigned to each kernel, we compare each keyword to its associated function using a minithesaurus obtained from a panel of four independent yeast experts. Each function 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.

To construct this thesaurus, each expert received a list of the 22 function descriptions listed by Spellman, and a separate list of 330 alphabetically sorted keywords our program produced. The experts assigned to each keyword the functionality descriptors that they judged to be most related to it; keywords that did not relate to any specific functionality were left unassigned. An example of two entries in the resulting thesaurus is shown in Table 2.

Table 2. Two thesaurus entries associating a gene function with its related keywords.

Function	Associated keywords
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

We compare the functionality of each gene according to Spellman with the functionality assigned by the panel to each of its keywords. That is, we count how many keywords that are associated with the gene according to our method indeed corresponded to the gene's functionality according to Spellman, and how many do not.

Results

The quality of our results is checked by comparing the sets of related genes produced by our method along with the associated keywords to the functionality assigned to genes (based on human judgment and expertise rather than on an automated process) by Spellman and his colleagues[7] as shown in Table 1.

Because Spellman's report does not assign functionality to many genes in the experiment (120 of 344 kernels used), we can only verify in this manner the results for the ones whose functionality Spellman determined. However, pointwise manual checking of the abstracts and genes associated with those 120 kernels not discussed by Spellman shows that for many kernels the results do agree with the known biology and gene relationships.

Table 3 illustrates a typical successful search. The left column lists the PMIDs for two kernels, the genes they stand for, and their respective functionality according to Spellman. The second column lists, for each of the two kernels, the top 10 keywords our algorithm associates with the set of abstracts it retrieved. The third column lists the top genes associated with each kernel (ELO1 has only nine genes associated with it because nine nonzero cosine coefficients were associated with its kernel). The fourth column lists each gene's function according to Spellman as a validity check for our results. Since the experiment includes genes that are not listed in Spellman's table, some of the genes in the third column are not assigned functionality by Spellman. For these genes (listed in parentheses in the table), we found the functionality in YPD.

Table 3. The results of using our algorithm for two different kernels, compared with functionality according to either Spellman or the Yeast Proteome Database (items in parentheses indicate YPD functionality).

Kernel (PMID, gene, function)	Keywords	Associated genes	Function
8702485	fatty acid,	OLE1	(Fatty acids, sterolic metabolism)
ELO1	fatty,	FAA4	Fatty acids, lipids, sterols, membranes
Fatty acids,	lipids,	FAA3	Fatty acids, lipids, sterols, membranes
lipids,	acid,	SUR2	Fatty acids, lipids, sterols, membranes
sterols,	grown,	FAA1	Fatty acids, lipids, sterols, membranes
membranes	medium,	ERG2	Fatty acids, lipids, sterols, membranes
	carbon,	PSD1	Fatty acids, lipids, sterols, membranes
	synthase,	CYB5	(Fatty acids, sterolic metabolism)
	strains,	PGM1	(Carbohydrate metabolism)
	deficient		
7651133	hexose,	HXT1	Nutrition
HXT7	glucose uptake,	RGT2	Nutrition
Nutrition	glucose	HXT4	Nutrition
	concentration,	HXT2	Nutrition
	fructose,	HXT2	Nutrition
	glycolytic,	GLK1	Nutrition
	glucose,	SEO1	(Small-molecule transport)
	sugars,	PRB1	(Protein degradation)
	uptake,	AGP1	Nutrition
	aerobic,	ZRT1	Nutrition
	utilization	MIG2	(Carbohydrate metabolism)

Table 4. The results for an uninformative kernel, compared with functionality according to Spellman.

Kernel (PMID, gene, function)	Keywords	Associated genes	Function
6323245	ARS,	CDC10	Site selection, morphogenesis
MCM2, MCM3, MCM6	autonomously replicating,	PHO3	Nutrition
Replication initiation	replicating sequence,	EST1	DNA synthesis
	autonomously,	MIF2	Chromatin
	minichromosomes,	PHO12	Nutrition
	replicating, centrometric,	POL2	DNA synthesis
	leu2, plasmids, ura3	DHS1	DNA repair
		SNQ2	Not available
		SMC3	Chromatin cohesion
		EXG2	Cell wall synthesis

The table shows that all of the genes found for these two kernels, except for PGM1 and PRB1, have a strong functional relationship to the genes represented by the kernels. Also, the keywords strongly indicate this biological function. (The keywords are associated as a set with the whole kernel entry and are not separated as one keyword per associated gene.) PGM1 is involved in carbohydrate metabolism, which is still functionally related to fatty acid metabolism. PRB1 is responsible for protein degradation, which is not related to nutrition. It is included in this set because its kernel abstract discusses regulation of the enzyme PRB1p by glucose rather than the biological function of PRB1p.

The results for approximately 100 of the 220 kernels with Spellman-assigned functionality closely resemble the results in Table 3 in two ways. The first is the strong agreement with Spellman's cluster assignment. The second is the accurate description given by the keywords that our method generated.

As a quantitative measure, we calculated the average number of correct and incorrect keywords among the five top-ranking keywords associated with each of these 100 kernels. We considered a keyword occurring in a list for a specific gene (kernel) as *correct* if it appeared in our thesaurus entry labeled with the same function that Spellman assigned to the gene. If its thesaurus entry was labeled with a different function, we considered it *wrong*. If the panel of experts did not assign it any function, we considered it to be nondescriptive. Out of the five top-ranking keywords, an average of 3.27 were associated with the correct function, only 1.12 were associated with the wrong function, and 0.61 were nondescriptive. The difference between the high rate of correct keyword assignment relative to the wrong and the nondescriptive assignment is highly statistically significant ($p << 0.005$, using the two-sample t-test).

For other kernels, the groups of related genes contain many genes to which Spellman did not assign functionality, which makes the results harder to validate. Our results also deviate from Spellman's functionality grouping when the kernel does not discuss the gene's biology but rather the experimental method for discovering it. Table 4 shows the results for one such uninformative kernel.

In this case, the kernel 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 unifying factor for the resulting set of genes is that their curated abstracts all discuss heterogeneous gene expression via plasmids, rather than gene function. The keyword list (which highly ranks the term "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.

Our information retrieval approach has four clear advantages:

- It effectively detects putative relationships among genes. These relationships can be verified through well-targeted experiments.
- It provides the relevant literature for analyzing the experimental results.
- It generates keywords (summary terms) explaining the discovered relationships. These keywords can help explain and evaluate the relationships found by direct clustering of expression data.
- It is independent of natural language usage

and nomenclature issues 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 preclustering of the genes among which it is looking to find relationships.

Thus, our method can be used both to generate hypotheses prior to an experiment and to interpret an experiment's results. Given a functionally descriptive kernel, 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 cannot ascribe function to genes that have not yet been studied. However, by pointing out commonalities between abstracts discussing distinct genes, it can uncover functional relationships among known genes that heretofore have gone unnoticed.

The main current limitation of our technique is the need to obtain functionally descriptive kernels. We are studying machine-learning methods that help automate kernel selection. Given a good source of kernels, we expect that using multiple kernels for each gene, rather than a single kernel, will provide a better initialization for the EM algorithm and further improve the results. Another promising direction is to extend the vector representation of abstracts to include gene expression values, simultaneously searching for related abstracts and similarly expressed genes.

Our method complements the analysis techniques currently applied to microarray data. Combining it 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 their time and effort in assigning functional tags to terms. This work was done while we were at the National Center for Biotechnology Information, NLM, NIH. An earlier version of it appeared in the ISMB 2000 conference.

References

1. V. Bafna and D.H. Huson, "The Conserved Exon Method for Gene Finding," *Proc. 8th Int'l Conf. Intelligent Systems for Molecular Biology*, AAAI Press, Menlo Park, Calif., 2000, pp. 3–12.
2. C.B. Burge and S. Karlin, "Finding the Genes in a Genomic DNA," *Current Opinion in Structural Biology*, vol. 8, 1998, pp. 346–354.
3. A.P. Dempster et al., "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. Royal Statistical Soc.*, vol. 39, no. 1, 1977, pp. 1–38.
4. H. Shatkay and W.J. Wilbur, "Finding Themes in Medline Documents: Probabilistic Similarity Search," *Proc. IEEE Conf. Advances in Digital Libraries*, IEEE CS Press, Los Alamitos, Calif., 2000, pp. 183–192.
5. G. Salton, *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computers*, Addison-Wesley, Reading, Mass., 1989.
6. I.H. Witten et al., *Managing Gigabytes: Compressing and Indexing Documents and Images*, 2nd ed., Morgan-Kaufmann, San Francisco, 1999.
7. 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*, vol. 9, no. 12, Dec. 1998, pp. 3273–3297.

The Authors

Hagit Shatkay is a research scientist in the Informatics Research group at Celera Genomics. Her main research interest is the application of machine-learning methods and probabilistic models to data-intensive problems. Her research and publications span a broad range of areas such as similarity queries, hidden Markov models for robot navigation, and using the literature for biological data analysis. She received her BSc and MSc in computer science with honors from the Hebrew University of Jerusalem and her PhD in computer science from Brown University. She is a member of the AAAI and Sigma Xi. Contact her at Celera Genomics, 45 W. Gude Dr., Rockville, MD 20850; hagit.shatkay@celera.com.

Stephen Edwards is a senior research scientist with Rosetta Inpharmatics, where he works on functional genomics using inkjet microarray technology. He previously completed a Pharmacology Research Associate Training Fellowship at the National Center for Biotechnology Information. While there, he worked on incorporating gene function information into automated DNA microarray analysis. He also helped create a drug-target-and-metabolizing-enzyme database as a repository of current genomic information. He received a BS in Chemistry with honors from the University of North Carolina at Chapel Hill and received his PhD from Vanderbilt University, where he studied the structural features of the alpha 2A-adrenergic receptor responsible for stabilizing this protein on the plasma membrane. Contact him at Rosetta Inpharmatics, 12040-115th Ave. NE, Kirkland, WA98034; stephen_edwards@merck.com.

Mark Boguski is a visiting scholar at the Fred Hutchinson Cancer Research Center in Seattle. While at the National Center for Biotechnology Information, he contributed to research on data mining, special-purpose databases, comparative genomics, and molecular evolution. As senior vice president of R&D at Rosetta Inpharmatics, he explored the interface between computational and high-throughput experimental biology. He received his BA from Johns Hopkins University and his MD and PhD from the Medical Scientist Training Program at Washington University in St. Louis. He is a member of the Institute of Medicine of the National Academy of Sciences and a fellow of the American College of Medical Informatics.

References from page 49

8. T.-K. Jenssen et al., "A Literature Network of Human Genes for High-Throughput Analysis of Gene Expression," *Nature Genetics*, vol. 28, no. 1, May 2001, pp. 21–28.
9. G. Salton, *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computers*, Addison-Wesley, Reading, Mass., 1989.
10. I.H. Witten et al., *Managing Gigabytes: Compressing and Indexing Documents and Images*, 2nd ed., Morgan Kaufmann, San Francisco, 1999.
11. H. Shatkay and W.J. Wilbur, "Finding Themes in Medline Documents: Probabilistic Similarity Search," *Proc. IEEE Conf. Advances in Digital Libraries*, IEEE CS Press, Los Alamitos, Calif., 2000, pp. 183–192.