

# Discovering Gene Functional Relationships Using FAUN (Feature Annotation Using Nonnegative matrix factorization)

Elina Tjioe<sup>1</sup>, Michael W. Berry<sup>\*1</sup>, Ramin Homayouni<sup>2</sup>

<sup>1</sup>Department of Electrical Engineering and Computer Science and Graduate School of Genome Science and Technology, University of Tennessee, Knoxville, TN 37996

<sup>2</sup>Department of Biology, Bioinformatics Program, University of Memphis, Memphis, TN 38152

Email: Elina Tjioe - tjioe@eecs.utk.edu; Michael W. Berry\* - berry@eecs.utk.edu; Ramin Homayouni - r.homayouni@memphis.edu;

\*Corresponding author

## Abstract

---

**Background:** Searching the enormous amount of information available in biomedical literature to extract novel functional relationships among genes remains a challenge in the field of bioinformatics. While numerous (software) tools have been developed to extract and identify gene relationships from biological databases, intuitive user-interfaces that would allow experts to immediately determine functional relationships among genes are lacking.

**Results:** In this study, we develop a Web-based bioinformatics software environment called FAUN or Feature Annotation Using Nonnegative matrix factorization (NMF) to facilitate both the discovery and classification of functional relationships among genes. Both the computational complexity and parameterization of NMF for processing gene sets are discussed. FAUN is tested on three manually constructed gene document collections. Its utility and performance as a knowledge discovery tool is demonstrated using a set of genes associated with Autism.

**Conclusions:** FAUN not only assists researchers to use biomedical literature efficiently, but also provides utilities for knowledge discovery. This Web-based software environment is particularly useful for the validation and analysis of gene associations suggested by microarray experimentation.

---

## **Background**

The MEDLINE 2008 literature database at NIH contains over 18 million records and is growing at an exponential rate [1]. With such rapid growth of the biomedical literature and the breakdown of disciplinary boundaries, it can be overwhelming to manually track of all new relevant discoveries, even on specialized topics. Moreover, the recent advances in genomic and proteomic technologies have added an abundance of genomic information into biomedical knowledge, which makes the situation even more complicated. One main difficulty in understanding high-throughput genomic data is to determine the functional relationships between genes. Even though DNA sequences have been completed in a growing number of organisms, gene sequence analysis unfortunately does not necessarily imply function.

Because research in this area is both time consuming and costly, it is very important to benefit from existing literature as much as possible. Some benefits include the discovery of hidden/implicit functional information of genes, and automated literature-based classification of a particular subset of genes. Consequently, a great deal of effort has been put into developing effective data mining tools to assist researchers in utilizing existing biomedical literature and genomic information.

## **Previous Work**

Numerous data mining tools have been proposed in the bioinformatics field (see reviews in [2–6]). One of the major steps in text mining is information retrieval (IR) [3] which consists of three basic types of models: set-theoretic (Boolean), probabilistic, and algebraic (vector space). Documents in each case are retrieved based on Boolean logic, probability of relevance to the query, and the degree of similarity to the query, respectively.

Some of the current software tools utilize functional gene annotations provided in public databases, such as Gene Ontology (GO) [7], Medical Subject Heading (MeSH) index [8], and KEGG [9]. For example, GoPubMed [10], a thesaurus-driven system, classifies abstracts based on GO, HAPI [11] identifies gene relationships based on co-occurrence of MeSH index terms in representative MEDLINE citations, and EASE [12] identifies gene relationships using the gene function classifications in GO. These co-occurrence based methods can be highly error prone. Ontology definitions help provide insights into biological

processes, molecular functions and cellular compartments of individual genes. However, they are often incomplete and lack information related to associated phenotypes [13]. In addition, Kostoff et al. [14] found a significant amount of conceptual information present in MEDLINE abstracts missing from the manually indexed MeSH terms. Moreover, indexing in MEDLINE can be inconsistent because of assignments by different human-indexers [15].

Several alternative approaches that use Medline-derived relationships to functionally group related genes have been reported [16]. Alako et al. [17] have developed CoPub Mapper which identifies shared terms that co-occurred with gene names in MEDLINE abstracts. PubGene [18] developed by Jenssen et al. constructs gene relationship networks based on co-occurrence of gene symbols in MEDLINE abstracts. Because of the inconsistency issues in gene symbol usage in MEDLINE, PubGene has low recall (ratio of relevant documents retrieved to the total number of relevant documents). It identifies only 50% of the known gene relationships on average. In addition to the official gene symbol, each gene typically has several names or aliases. In IR, these problems are referred to as synonymy (multiple words having the same meaning) and polysemy (words having multiple meanings). Several methods have been proposed to solve these ambiguity issues in gene or protein names [19–21].

To make discoveries based on literature, several tools have been developed using indirect or implicit relationships [22,23]. It is currently not feasible to use them for high-throughput studies [24]. Among many biomedical text mining tools that have been built, Chilibot [25], Textpresso [26], and PreBIND [27] are examples of tools that have high usage rates. These tools are thought to be successful due to their use in extremely domain-specific tasks [28]. Chilibot is a system with a special focus on the extraction of relationships between genes, proteins and other information. Textpresso is an information-retrieval and extraction tool developed for (WormBase) literature. Finally, PreBIND provides utilities in the extraction of protein-protein interactions.

One of the most promising vector space models to extract and identify functional relationships among genes is the Semantic Gene Organizer (SGO) software tool which allows researchers to view groups of genes in a global context [29]. It uses latent semantic indexing (LSI) which performs truncated singular value decomposition (SVD). Homayouni et al. [24] demonstrated the use of LSI for genomic studies. They extracted both explicit (direct) and implicit (indirect) gene relationships based on keyword queries, as well as gene-abstract queries, from the biomedical literature with better accuracy than term co-occurrence methods. The underlying SVD factorization technique decomposes the original term-by-document nonnegative matrix into a new set of factor matrices containing positive and negative values. These matrix

factors can be used to represent both terms and documents in a low-dimensional subspace. Unfortunately, the interpretation of the LSI factors is non-intuitive and difficult due to the negative factor components. The main limitation of LSI is that while it is robust in identifying *what* genes are related, it has difficulty in answering *why* they are related.

To address this difficulty, a new method, nonnegative matrix factorization (NMF), has been proposed. This factorization method, unlike SVD, produces decompositions which can be readily interpreted.

### **NMF Background**

Lee and Seung [30] were among the first researchers to introduce the nonnegative matrix factorization (NMF) problem. They demonstrated the application of NMF in text mining and image analysis. NMF decomposes and preserves the nonnegativity of the original data matrix. The low-rank factors produced by NMF can be interpreted as parts of the data. Recently, NMF has been widely used in the bioinformatics field, including the analysis of gene expression data, sequence analysis, gene tree labeling, functional characterization of gene lists and text mining [31–37]. Chagoyen et al. have shown the usefulness of NMF methods in extracting the semantic features in biomedical literature [32]. Pascual-Montano et al. [36] developed an analytical tool called bio-NMF for simultaneous clustering of genes and samples. It requires (on input) a data matrix (e.g., term-by-doc matrix) and outputs the corresponding matrix factors. Even though the tool is robust and flexible, its use by biologists might not be obvious. Therefore, an intuitive interface that allows the biologist to use literature-based NMF methods for determining functional relationships among genes is still needed.

In this study, we develop a Web-based bioinformatics software environment called Feature Annotation Using Nonnegative matrix factorization (FAUN) to facilitate both knowledge discovery and classification of functional relationships among genes. The ability to facilitate knowledge discovery makes FAUN very attractive to genomic scientists. Thus, one of the main design goals of FAUN is to be biologically user-friendly. Providing a list of genes with gene identifiers such as gene IDs or gene names, FAUN constructs a gene-list-specific document collection from the biomedical literature. NMF can be used to exploit the nonnegativity of term-by-gene document data, and can extract the interpretable features of text which might represent usage patterns of words that are common across vastly different gene documents. NMF methods are iterative in nature so that the problem involves computational issues such as: proper initialization, rank estimation (i.e., subspace dimension), stopping criteria, and convergence. To address these issues, many variations of NMF with different parameter choices have been proposed [34, 38, 39].

While developing FAUN, we try to understand how the NMF model can be adapted or improved for gene datasets that will not only yield good mathematical approximations, but also provide valuable biological information.

## Results

A preliminary assessment of FAUN feature classification has been conducted on three manually constructed gene document datasets with known functional gene relationships [40].

### Gene Datasets

The first dataset (50TG collection) is a gene document collection of 50 manually selected genes related to development, Alzheimer’s disease, and cancer biology [24]. The second dataset (BGM collection) is composed of three non-overlapping gene lists from the Biocarta, Gene Ontology and MeSH databases [41]. The third dataset (NatRev collection) is composed of five gene lists selected from Nature Reviews papers [42–46]. Categories used in all of these datasets are shown in Table 1.

### Parameterization

Initialization of the NMF factors  $W$  and  $H$  was primarily based on NNDSVD (described earlier). NNDSVD basically starts with the truncated SVD of the gene-by-document (sparse) matrix  $A$ . Although NNDSVD produces a static starting point, different methods (see [39, 47]) can be applied to remove zeros from the initial approximation and thereby prevent them from being fixed throughout the (multiplicative) update process.

The values of factorization ranks considered for the three datasets were 5, 10, 15, 20, 25, 30, 40, and 50. We restricted the maximum number of iterations to 1000 and 2000 and stopped iterations if the consecutive iterates of  $W$  and  $H$  (generated by the multiplicative update algorithm defined in Equation (4)) if the consecutive iterates of  $W$  and  $H$  were closer than  $\tau_W = 0.01$  and  $\tau_H = 0.001$ , respectively, in Frobenius norm. That is,  $\|W_{old} - W_{new}\|_F < \tau_W$  and  $\|H_{old} - H_{new}\|_F < \tau_H$ . The effect of constraints on smoothing and sparsity for the  $W$  and  $H$  iterates has been studied and we refer the reader to [39, 47] for more details on these effects.

## Classification Accuracy

For each dataset, six NMF models were generated with rank  $k$  set to 10, 15, 20, 30, 40 and 50. Features in the NMF models were annotated using the automated annotation process described in the Methods section. Using these annotated features, each gene in the dataset was assigned to the same category with its strongest annotated feature (indicated by the largest corresponding column entry of  $H$ ). For example, if gene X has the strongest association with feature Y, then gene X will be classified as being in the category with which feature Y is labeled. The FAUN classification using the strongest feature (per gene) with rank  $k = 30$  yielded 98%, 88.2% and 86.4% accuracy for datasets 1, 2, and 3, respectively (Figure 5).

The total gene recall per category was also investigated. For each feature, the corresponding maximum row entry of  $H$  ( $\max H$ ) is found and all genes in the feature with their  $H$  values  $< \max H \times f_\tau$ , where  $f_\tau$  is a chosen feature threshold, are skipped. For each gene, all features (above  $f_\tau$ ) associated with this gene are taken and then categories are assigned to the gene based on feature labeling.

The classification accuracy is evaluated in such a way that if the correct class is not among the classes assigned, the correctness is defined to be 0. If the correct class is among the classes assigned, the correctness is defined to be  $1/(\text{number of classes assigned to the gene})$ . The total correctness is the sum of correctness assigned to every gene expressed as a percentage (0-100%). Using a feature weight threshold  $f_\tau = 1.0$ , gene recall ranges of 78%-100%, 71.6%-97.1%, and 42.7%-80.9% for the 50TG, BGM, and NatRev datasets, respectively, have been reported [40].

Low classification accuracy equates to the misclassification of a human-curated category in the dataset.

However, this misclassification does not necessarily imply that FAUN cannot be used to infer new (previously unknown) functional properties. A few examples of such discovery are mentioned in the next section.

## Knowledge Discovery

One of the most important capabilities of FAUN is to discover novel gene relationships from the biomedical literature, leading to generation of experimental hypotheses by the end user (biologist). This is essentially accomplished by clustering genes according to word usage patterns from the literature. By altering parameters in FAUN, a user can control the granularity by which genes and terms/features are associated. In this section we evaluate the effect of two parameters (rank- $k$  and  $H$ -matrix threshold) on the discovery process using the NatRev dataset that contains 26 Autism associated genes from a total of 110 genes in the dataset.

Role of rank- $k$  on discovery process. First, we examined gene clustering at various rank- $k$ , using a constant global  $H$ -threshold. In general, the smaller the rank- $k$  the larger the number of genes associated with each feature. For example, the feature containing autism as a highly ranked term (hereafter called *autism feature*) included 21 genes at  $k = 10$  and only 5 genes at  $k = 30$  (Figure 6). Interestingly, all 5 genes in the  $k = 30$  autism feature are clearly linked to autism (100% precision) [42]. Using the gene list in the review article by Abrams and Geschwind as ground truth, precision, recall and F1 score (harmonic mean of precision and recall) values were calculated for the various features in this collection. Figure 6 shows that precision increases with  $k$  at the cost of recall and that rank-20 feature shows the best performance (highest F1 score). By examining the top ranked terms associated with this feature, we can see that the genes are clustered together because they share highly weighted terms such as neurons, brain, neuronal, mutation, cortex, receptor, syndrome, transmission, autism, hippocampus, synaptic and so on (Table 3). Although useful, these terms appear to be somewhat general and do not adequately classify the genes into functional groups. For more specific annotation, a larger rank- $k$  is required. For instance, at  $k = 30$ , the genes associated with autism by Abrams and Geschwind into 4 different features. These features are associated with more specific terms that better represent the gene functions, such as methylation, channels and transporters (Figure 6).

It is important to point out that NMF clusters genes together even if they do not share every top weighted term for the feature. For instance, the autism feature for the  $k = 20$  model included **DISC1**, **SHANK3** and **GATA3** although the term autism did not appear in the abstracts used to build the NMF model (Table 3, red highlighted genes and term). Indeed, the abstracts used in our collection were limited to 2006 and earlier and the discovery of **SHANK3** and **DISC1** as autism genes occurred only after 2007 [48, 49]. This association is due to the overlap of other terms that are highly weighted in this feature, demonstrating the utility of NMF for discovering new gene associations based on word pattern usage.

Role of  $H$ -matrix threshold on discovery process. Another way to view more genes associated with a specific feature is to loosen the  $H$ -matrix threshold for that feature (set at a global median value of 1.0 by default). The threshold can be modified based on a local value. Lowering the local  $H$ -matrix threshold expands the number of genes associated with the feature. We applied this strategy to expand the number of genes associated with the autism and methylation features produced by the rank-30 NMF model in Figure 7 to a total of 16 genes each. Calculation of the F1 score using the Abrams and Geschwind dataset on autism, revealed that both sets achieved reasonable precision and recall (Figure 8). Interesting, the union of these two sets achieved a better F1 score than that achieved in  $k = 20$  autism feature described

above. Therefore, this strategy is capable of identifying additional target genes. Importantly, both genesets contained **GATA3**, which has not been definitively linked to autism to date.

Taken together, we presented two possible strategies here that can be used within FAUN to explore relationships between genes and to make predictions that can later be tested experimentally. Related genes may be identified by lowering the rank- $k$  and using general terms to cluster genes together. Alternatively, related genes can be identified by lowering the  $H$ -matrix threshold on a higher rank- $k$  model, which uses more specific terms to cluster genes together. Each strategy has its own merit and would likely produce different results with different datasets. It is important to point out that both strategies produced reasonably high precision and recall.

### **Computational Complexity**

The cost per each NMF iteration typically ranges from 0.004 seconds to 0.011 seconds, depending on the choice of rank  $k$ , on a PC with 2 Intel Core CPU T5600/1.83 GHz processors, 1.5GB memory, running 32 bit Linux. The elapsed CPU time and operations per iteration of the NMF multiplicative update algorithm per dataset are listed in Table 3. Theoretically, one can estimate the complexity (floating-point operation count) of one iteration of Equation (4) to be  $k \times m \times n$ . This computational cost is compared with other competing NMF-update approaches in [47]. Overall, for the three datasets considered in this study, the runtimes are modest for a desktop/laptop computing environment.

### **Conclusions**

Given a list of genes, FAUN allows researchers to not only hypothesize *why* genes might be related but also classify them functionally with promising accuracy. More details on the effect of rank, initialization methods, stopping criteria, smoothing and sparsity constraints on the NMF model are provided in [47]. FAUN not only assists researchers to use biomedical literature efficiently, but also provides utilities for knowledge discovery. Furthermore, FAUN is designed to incorporate updates to the literature. FAUN is particularly useful for the validation and analysis of gene associations suggested by microarray experimentation.

## Methods

A simple demonstration of the FAUN bioinformatics software environment is available to the public<sup>1</sup>. Currently, NMF models are generated separately. We expect to have them automatically generated (on the server) from a user-supplied gene list in the very near future. The main functionalities of FAUN (along with illustrative screenshots) are described below.

The screenshots were taken from the third dataset (NatRev collection) which contains 110 manually selected genes related to autism, diabetes, translation, mammary gland development, and fanconi anemia. [47].

### Extracting Concept Features

Due to the nonnegativity constraints of the NMF model, interpretable features can be extracted from the columns in the term-by-feature ( $W$ ) matrix. Since the matrix is sparse, each column (feature) is represented by a small subset of terms, which forms a certain term usage pattern. This pattern will help FAUN users in determining the concept or meaning of the feature. For example, a feature containing the terms *mosquito*, *Plasmodium*, *blood*, and *quinine* might describe the disease malaria. Once the user recognizes a specific feature, based on its dominant terms, the feature can be annotated into something more meaningful (e.g., breast cancer) than the default label (e.g., Feature 1) for future reference. The screenshot of some annotated features and their top associated terms for the NatRev gene document collection (dataset 3) are shown in Figure 2. The entropy filter slider bar allows the user to get information on how the term is used consistently throughout the whole collection. If a term occurs in all documents the same way, it will have a low entropy weight and is probably not a very good concept discriminator. High entropy words tend to have specific usage patterns and are hopefully more meaningful. This entropy filter option might help the user to focus on more important features.

A typical usage scenario for FAUN concept features is shown in Figure 2. Once a document collection is built, three NMF models are generated with NMF ranks  $k = 10, 15,$  and  $20$ , for low, medium, and high resolutions by default. Even higher resolutions are certainly possible, and the screenshot in Figure 2 is taken from a NMF model with 30 features (only the first 5 features are shown). The user can then look through the top terms in each feature and supply (if possible) an appropriate label. For example, Feature 2 in Figure 2 could readily be labeled (or annotated) as a descriptor of insulin signaling.

If the user is interested in further exploring the *Insulin Signaling* feature, he/she can then click on the

---

<sup>1</sup>See <http://grits.eecs.utk.edu/faun>

feature to show all the genes in the collection that FAUN suggested to be highly associated with the feature. A description of how FAUN identifies the genes associated with each feature is provided in the next section.

### Gene Identification

For each feature, genes that are highly associated with it can be extracted from the feature-by-gene ( $H$ ) matrix. A gene can be described by more than one feature. The association strength (feature weight) between gene and feature is determined by the appropriate element in the matrix. Genes that share one or more of the same features might be functionally related to one another.

The genes in the gene document collection can also be color-coded based on their expression change in the microarray experiments: red is up-regulated and green is down-regulated. The user can see not only which genes belong to what features, but also see if a particular feature contains predominantly up- or down-regulated genes in the user-specific experiments. The feature is color-coded based on the predominant genes in that feature which are up- or down-regulated. If gene expression information is not provided, the gene and its feature are color-coded with yellow as shown in Figures 2 and 3. The screenshots of *genes* that are highly associated with Feature 6 and their top associated *terms* for the NatRev collection, along with some options, are shown in Figure 3.

Genes are listed from left-to-right by strength of association with the selected feature. The log-entropy weight of the terms in each gene is color-coded for visual analysis, with more red for a higher weight. The number of genes to be displayed is set to 15 and can be changed using the display-genes drop-down menu. All genes, above the set feature weight threshold, with their terms and term weights can be downloaded in csv format for further analysis. There might not be a single optimal threshold value that works the best for every case. FAUN provides global and local gene filter options to let users try different thresholds. The gene filter option allows users to filter the genes associated with each feature globally, across all gene documents above a certain threshold, or locally, within each gene document above the 70th local percentile. The *medium* global gene filter setting (i.e., feature weight = 1.0) is the default.

To see how the feature terms and/or gene symbols are used in the original gene document article, sentences using the terms and/or the gene symbols can be viewed. The sentences are ranked based on term frequency. The ranked sentences are displayed in the popup window by clicking on the gene symbol at the head of the column. This popup window also serves as the quick summary page for the gene and provides a link to the Entrez Gene page for more information about that gene.

At this point, the FAUN user might have some ideas about what kinds of features are present in the gene document collection, and some familiarity with the genes that are associated with certain features. Genes belonging to the same feature might suggest that they are functionally related based on the literature. Such hypotheses may well lead to new discoveries in gene characterization. Namely, genes represented by the same feature may function in the same pathway.

To explore even further why certain subsets of genes are related, and how strongly they are related, the user can click on the *gene vs gene correlation* link shown at the bottom of the screenshot in Figure 3.

### Gene Exploration

FAUN’s capability to identify subsets of genes which are related is described above. To see how strongly genes in the user-selected feature ( $i$ ) cross-correlate, the correlation of gene  $x$  and gene  $y$  for  $n$  selected features is estimated using the Pearson correlation coefficient ( $r$ ) which is defined by

$$r_{xy} = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n \sum x_i^2 - (\sum x_i)^2} \sqrt{n \sum y_i^2 - (\sum y_i)^2}}. \quad (1)$$

The Pearson correlation matrix for all the genes is then generated. The correlation is color-coded for visual analysis, with more red for a stronger correlation. An example of the correlation matrix for all the genes in Feature 20 (*Methylation*) of the NatRev collection is shown in Figure 4. Users can view the correlation of any pair of genes and with respect to any combinations of features with a minimum of 3 features selected. By default, the user-selected feature along with its left and right neighboring features are used to compute the Pearson correlation.

Another important capability of FAUN is to potentially explain why a subset of genes might be related. Two pairs of genes which associate with different contributions of features might have an overall similar degree of correlation. In FAUN, the users can view exactly which contributions of (annotated and/or un-annotated features) are involved by clicking on the correlation cell shown in Figure 4. The selected cell in the figure shows a strong correlation, indicated by the red cell color, between gene **MECP2** and gene **UBE3A**, mainly due to Features 1 (*Breast Cancer*) and 20 (*Methylation*), suggested by the feature strengths at the left side.

### Gene Classification

A new gene document added to a gene document collection can be analyzed (for the presence of annotated features) without having to update the NMF model for the collection. The FAUN classifier can accept a

stream of new documents and determine their features based on the presence of terms in the previously-annotated features. It is possible to automatically retrieve newly published articles and run the FAUN classifier to determine if they are related to any of the interest features in the studied gene collection without having to continually update the NMF model. Of course, periodic updating of the NMF model to reflect changes in literature may be needed.

## **Software Design**

FAUN consists of a computational core and Web-based user interface. The computational core consists of programs that build the gene document collection, parse the collection, build an NMF model from the collection, and classify new documents based on the NMF model. These programs will be described in more detail in the following sections. The primary design goal of the user interface was to make the analysis of NMF accessible to biologists.

FAUN users can have one or more separate accounts to perform independent analysis of gene datasets. The sessions persist between logins so users can easily resume their work after interruption. The users can take advantage of various interactive components such as the gene-to-gene correlation matrix, sentence display, filters and dynamic generation of results. FAUN utilizes a combination of technologies: PHP, Javascript, Flash, and C++. PHP is used to communicate between the computational core and the Web user interface. It is also used for generating HTML pages. Javascript is used for client-side scripting and creating graphical user interface (GUI) elements. The gene-to-gene correlation matrix is generated using a PHP/SWF chart tool [50]. This tool is used for simple and flexible chart generation, and quality of Flash graphics. PHP scripts are used to generate chart data, and then the data is passed to the PHP/SWF chart tool to generate Flash charts and graphics. C++ was used to write the computational core.

## **Gene Document Collection**

All genes in a given gene list are used to compile titles and abstracts in Entrez Gene [51]. Currently, to avoid polysemy and synonymy issues, there are still human interventions in the document compilation process, such that abstracts are not specific to a particular gene name or alias. Titles and abstracts for a specific gene are concatenated to prepare a gene document.

The collection of gene documents is parsed into terms using the current C++ version of General Text Parser (GTP) [52]. Terms in the document collection that are common and non-distinguishing are

discarded using a stoplist<sup>2</sup>. In addition, terms that occur less than twice locally in the gene document or globally in the entire document collection are ignored and not considered as dictionary terms. Hyphens and underscores are considered as valid characters. All other punctuation and capitalization are ignored. A term-by-gene document matrix is then constructed where the entries of the matrix are the nonnegative weighted frequencies for each term. These term weights, computed using a log-entropy weighting scheme [53], are used to describe the relative importance of term  $i$  for the corresponding (gene) document  $j$ . That is, the term-by-gene document matrix is defined as

$$A = [w_{ij}], \text{ where } w_{ij} = l_{ij} \times g_i .$$

The local component  $l_{ij}$  and the global component  $g_i$  can be computed as

$$\begin{aligned} l_{ij} &= \log_2(1 + f_{ij}), \\ g_i &= 1 + \left( \sum_j [p_{ij} \log_2(p_{ij})] \right) / \log_2 n, \\ p_{ij} &= f_{ij} / \sum_j f_{ij}, \end{aligned} \tag{2}$$

where  $f_{ij}$  is the frequency of term  $i$  in document  $j$ ,  $p_{ij}$  is the probability of the term  $i$  occurring in document  $j$  and  $n$  is the number of gene documents in the collection. This log-entropy weighting pair, which has performed well in several LSI-based retrieval experiments, decreases the effect of term spamming while giving distinguishing terms higher weight.

To summarize, a document collection can be expressed as an  $m \times n$  matrix  $A$ , where  $m$  is the number of terms in the dictionary and  $n$  is the number of documents in the collection. Once, the nonnegative matrix  $A$  has been created, nonnegative matrix factorization (NMF) is performed.

### Nonnegative Matrix Factorization

NMF is a matrix factorization algorithm to best approximate the matrix  $A$  by finding reduced-rank nonnegative factors  $W$  and  $H$  such that  $A \approx WH$ . The sparse matrix  $W$  is commonly referred to as the *feature matrix* containing *feature (column) vectors* representing certain usage patterns of prominent weighted terms, while  $H$  is referred to as the *coefficient matrix* since its columns describe how each document spans each feature and to what degree.

In general, the nonnegative matrix factorization (NMF) problem can be stated as follows:

---

<sup>2</sup>A sample stoplist is provided at <ftp://ftp.cs.cornell.edu/pub/smart/English.stop>

Given a nonnegative matrix  $A \in \mathcal{R}^{m \times n}$ , and an integer  $k$  such that  $0 < k \leq \min(m, n)$ , find two nonnegative matrices  $W \in \mathcal{R}^{m \times k}$  and  $H \in \mathcal{R}^{k \times n}$  that minimize the cost function

$$f(W, H) = \frac{1}{2} \|A - WH\|_F^2 = \frac{1}{2} \sum_{ij} (A_{ij} - (WH)_{ij})^2 . \quad (3)$$

This cost function, half of the squared Frobenius norm of the residual error, equals 0 if and only if  $A = WH$ . The minimization of  $f(W, H)$  can be challenging due to the existence of local minima owing to the fact that  $f(W, H)$  is non-convex in both  $W$  and  $H$ . As noted before, due to its iterative nature, the NMF algorithm may not necessarily converge to a unique solution on every run. For a particular NMF solution of  $W$  and  $H$ ,  $WDD^{-1}H$  is also a solution for any nonnegative invertible matrix  $D$  [38]. The NMF solution depends on the initial conditions for  $W$  and  $H$ . To address this issue, we use the Nonnegative Double Singular Value Decomposition (NNDSVD) initialization strategy proposed by Boutsidis and Gallopoulos [54]. This NNDSVD algorithm does not rely upon randomization and is based on approximations in the positive components of the truncated SVD factors of the original data matrix. Essentially, this provides NMF a fixed starting point, and the iteration to generate  $W$  and  $H$  will converge to the same minima. As noted by Chagoyen et al. in [32], having multiple NMF solutions does not necessarily mean that any of the solutions must be erroneous.

We use the multiplicative update algorithm proposed by Lee and Seung [55] to compute consecutive iterates of  $W$  and  $H$ :

$$\begin{aligned} H_{cj} &\leftarrow H_{cj} \frac{(W^T A)_{cj}}{(W^T W H)_{cj} + 10^{-9}} , \\ W_{ic} &\leftarrow W_{ic} \frac{(A H^T)_{ic}}{(W H H^T)_{ic} + 10^{-9}} . \end{aligned} \quad (4)$$

To avoid division by zero, the small constant  $10^{-9}$  is added to the denominator of each update rule above. In each iteration, both  $W$  and  $H$  are updated, which generally gives faster convergence than updating each matrix factor independently of the other. The computational complexity of the multiplicative update algorithm is easily shown to be  $\mathcal{O}(kmn)$  floating-point operations per iteration.

The choice of factorization rank  $k$  (selected number of features) is often problem-dependent, and it is a difficult problem to find the optimum value of  $k$ . In general,  $k$  is chosen such that it is less than the minimum dimension of  $A$  ( $m$  or  $n$ ). We investigate the effect of rank  $k$  for classifying gene documents in the Results section. As discussed in [38, 47], one can compensate for uncertainties in the data or to enforce desired structural properties in the factor matrices. Additional application-dependent constraints can be

added by modifying the cost function of Equation (3) to be

$$f(W, H) = \frac{1}{2}\|A - WH\|_F^2 + \alpha J_1(W) + \beta J_2(H),$$

where  $\alpha$  and  $\beta$  are relatively small regularization (control) parameters and  $J_1(W)$  and  $J_2(H)$  are functions defining additional constraints (e.g., smoothness or sparsity) on  $W$  and  $H$ , respectively. As explained in [38, 47], the rationale for enforcing smoothness (e.g.,  $J_1(W) = \|W\|_F^2$ ) or sparsity constraints on the  $W$  factor is to potentially improve the interpretability of its feature (column) vectors. Applying such constraints to the columns of the  $H$  (coefficient) matrix factor can control the span (or use) of features to explain documents in the collection.

### FAUN Workflow

As shown in workflow design depicted in Figure 1, for a given gene list, FAUN creates a *raw* term-by-gene document (sparse) matrix upon which an NMF model is built. The matrix for FAUN, created in the same manner as for SGO [24, 29], is then decomposed using NMF methods. The raw matrix is factored using rank  $k$  to produce a  $k$ -feature-NMF model for the gene document collection. Currently, FAUN uses a rank  $k$  of 10, 15, and 20 for low, medium and high *resolution*<sup>3</sup> of the NMF model, respectively. The NMF model containing  $W$  and  $H$  matrices is used to extract dominant terms and dominant genes for all  $k$  features. Dominant genes are then correlated for selected feature(s). FAUN users can then annotate some or all of the features and then deploy the FAUN classifier to determine the (concept-based) features of a new stream of gene documents.

The FAUN classifier accepts a new document to be classified, the entropy weights of terms in Equation (2) used in the NMF model, the term-by-feature matrix factor ( $W$ ), stop words, and thresholds for entropy weight and term frequency.

The module then computes the weight for each feature based on the weight of its terms whose entropy is larger than the entropy threshold and frequency is larger than the term frequency threshold. It then outputs the features sorted by weight from the highest to the lowest. The process of mapping features to gene classes will be described below.

Preliminary testing indicated that the classifier accuracy was around 80%. The test was conducted based on the first dataset (described below) that contained 50 genes. NMF models were first built with ranks of 10, 20, 30 and 40 using 40 genes randomly selected from the 50-gene dataset. The classifier was then

---

<sup>3</sup>The proper range of  $k$  could be determined based on user feedback in the future.

trained using the matrix in newly built NMF models. The accuracy was tested using the remaining 20% of the gene documents.

### **Automated Feature Annotation**

The FAUN classifier described above classifies genes based on annotated features in the NMF models. The process of annotating the features is typically done manually with the FAUN interface while exploring the gene dataset. Features in the NMF model can be annotated manually by the domain-expert using dominant feature terms. To automate the process for the other two datasets, features are annotated or mapped to gene classes using the FAUN annotation script (see [47]). In order to assign classification categories (classes) to the genes, the script requires the matrix from the NMF model, the (known) classification categories, the NMF rank, and a feature weight threshold.

### **Authors contributions**

Professors Berry and Homayouni have a long standing collaboration on the use of text mining algorithms and software in bioinformatics. This work is based on the PhD dissertation of Dr. Elina Tjioe, a former graduate student in the Graduate School of Genome Science and Technology at the University of Tennessee. Under Professor Berry's supervision, Dr. Tjioe designed and implemented the FAUN core modeling and interface software. Professor Homayouni provided gene lists and conducted extensive testing with the NMF models (i.e., features) generated by Dr. Tjioe. Professor Berry and Dr. Tjioe were primarily responsible for the Background and Methods and the screenshots and computational complexity components of the Results section. Professor Homayouni provided the knowledge discovery components of the Results section related to the NatRev (Autism) collection.

### **Acknowledgements**

This work is supported by NIH-subcontract (HD052472) involving the University of Tennessee, University of Memphis, Oak Ridge National Laboratory, and the University of British Columbia.

### **References**

1. [[http://www.nlm.nih.gov/bsd/revup/revup\\_pub.html#med\\_update](http://www.nlm.nih.gov/bsd/revup/revup_pub.html#med_update)].
2. Ananiadou S, DB DK, Tsujii J: **Text mining and its potential applications in systems biology**. *Trends Biotechnol* 2006, **24**(12):571–579.
3. Baeza-Yates R, Ribeiro-Neto B: *Modern Information Retrieval*. New York: ACM Press 1999.
4. Cohen A, Hersh W: **A survey of current work in biomedical text mining**. *Brief Bioinform* 2005, **6**:57–71.

5. Jensen L, Saric J, Bork P: **Literature mining for the biologist: from information retrieval to biological discovery.** *Nat Rev Genet* 2006, **7**(2):119–129.
6. Krallinger M, A AV: **Text-mining and information-retrieval services for molecular biology.** *Genome Biol* 2005, **6**(7):224.
7. Ashburner M, Ball C, Blake J, Botstein D, Butler H, Cherry J, Davis A, Dolinski K, Dwight S, Eppis J, et al.: **Gene ontology: tool for the unification of biology.** *Nat Genet* 2000, **25**:25–29. [The Gene Ontology Consortium].
8. Golbeck J: **The National Cancer Institute’s thesaurus and ontology.** *J Web Semantics* 2003, **1**:75–80.
9. Kanehisa M, Goto S: **KEGG: kyoto encyclopedia of genes and genomes.** *Nucleic Acids Res* 2000, **28**:27–30.
10. Doms A, Schroeder M: **GoPubMed: exploring PubMed with the Gene Ontology.** *Nucleic Acids Res* 2005, **33** (Web Server issue):W783–786.
11. Masys D, Welsh J, Fink JL, Gribskov M, Klacansky I, Corbeil J: **Use of keyword hierarchies to interpret gene expression patterns.** *Bioinformatics* 2001, **17**(4):319–326.
12. Hosack D, Jr GD, Sherman B, Lane H, Lempicki R: **Identifying biological themes within lists of genes with EASE.** *Genome Biol* 2003, **4**(10):R70.
13. Khatri P, Draghici S: **Ontological analysis of gene expression data: current tools, limitations, and open problems.** *Bioinformatics* 2005, **21**(18):3587–3595.
14. Kostoff R, Block J, Stump J, Pfeil K: **Information content in Medline record fields.** *Int J Med Inform* 2004, **73**(6):515–527.
15. Funk M: **Indexing consistency in MEDLINE.** *Bull Med Libr Assoc* 1983, **71**:176–183.
16. Shatkay H, Feldman R: **Mining the biomedical literature in the genomic era: an overview.** *J Comput Biol* 2003, **10**(6):821–855.
17. Alako B, Veldhoven A, van Baal S, Jelier R, Verhoeven S, Rullmann T, Polman J, Jenster G: **CoPub Mapper: mining MEDLINE based on search term copublication.** *BMC Bioinformatics* 2005, **6**:51.
18. Jenssen T, Laegreid A, Komorowski J, Hovig E: **A literature network of human genes for high-throughput analysis of gene expression.** *Nat Genet* 2001, **28**:21–28.
19. Gaudan S, Kirsch H, Rebholz-Schuhmann D: **Resolving abbreviations to their senses in Medline.** *Bioinformatics* 2005, **21**(18):3658–3664.
20. Hanisch D, Fundel K, Mevissen H, Zimmer R, Fluck J: **ProMiner: rule-based protein and gene entity recognition.** *BMC Bioinformatics* 2005, **6** Suppl:S14.
21. Schijvenaars B, Mons B, Weeber M, Schuemie M, van Mulligen E, Wain H, Kors J: **Thesaurus-based disambiguation of gene symbols.** *BMC Bioinformatics* 2005, **6**:149.
22. Becker K, Hosack D, Jr GD, Lempicki R, Bright T, Cheadle C, Engel J: **PubMatrix: a tool for multiplex literature mining.** *BMC Bioinformatics* 2003, **4**:61.
23. Swanson D: **Fish oil, raynaud’s syndrome, and undiscovered public knowledge.** *Perspect Biol Med* 1986, **30**:7–18.
24. Homayouni R, Heinrich K, Wei L, Berry M: **Gene clustering by latent semantic indexing of MEDLINE abstracts.** *Bioinformatics* 2005, **21**:104–115.
25. H HC, B BS: **Content-rich biological network constructed by mining PubMed abstracts.** *BMC Bioinformatics* 2004, **5**:147.
26. Muller H, Kenny E, Sternberg P: **Textpresso: an ontology-based information retrieval and extraction system for biological literature.** *PLoS Biol* 2004, **2**(11):e309.
27. Donaldson I, Martin J, de Bruijn B, Wolting C, Lay V, Tuekam B, Zhang S, Baskin B, Bader G, Michalickova K, et al.: **PreBIND and Textomy—mining the biomedical literature for protein-protein interactions using a support vector machine.** *BMC Bioinformatics* 2003, **4**:11.
28. Cohen K, Hunter L: **Getting started in text mining.** *PLoS Comput Biol* 2008, **4**:e20.

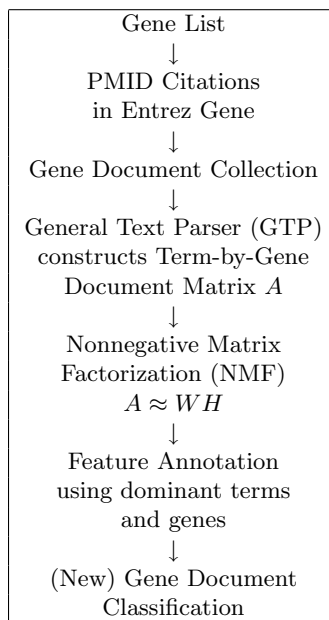
29. Heinrich K: **Finding functional gene relationships using the Semantic Gene Organizer**. *Master's thesis*, Department of Computer Science, University of Tennessee, Knoxville, TN 2004.
30. Lee D, Seung H: **Learning the parts of objects by non-negative matrix factorization**. *Nature* 1999, **401**(6755):788–791.
31. Carmona-Saez P, Pascual-Marqui R, Tirado F, Carazo J, Pascual-Montano A: **A: Biclustering of gene expression data by Non-smooth Non-negative Matrix Factorization**. *BMC Bioinformatics* 2006, **7**:78.
32. M MC, Carmona-Saez P, Shatkay H, Carazo J, Pascual-Montano A: **Discovering semantic features in the literature: a foundation for building functional associations**. *BMC Bioinformatics* 2006, **7**:41.
33. Gao Y, Church G: **Improving molecular cancer class discovery through sparse non-negative matrix factorization**. *Bioinformatics* 2005, **21**(21):39703975.60.
34. Heinrich K, Berry M, Homayouni R: **Gene tree labeling using nonnegative matrix factorization on biomedical literature**. *Comput Intell Neurosci* 2008:276535.
35. HKim, Park H, Drake B: **Extracting unrecognized gene relationships from the biomedical literature via matrix factorizations**. *BMC Bioinformatics* 2007, **8 Suppl**(9):S6.
36. Pascual-Montano A, Carmona-Saez P, Chagoyen M, Tirado F, Carazo J: **RD: bioNMF: a versatile tool for non-negative matrix factorization in biology**. *BMC Bioinformatics* 2006, **7**:366.
37. Pehkonen P, Wong G, Toronen P: **Theme discovery from gene lists for identification and viewing of multiple functional groups**. *BMC Bioinformatics* 2005, **6**:162.
38. Berry M, Browne M, Langville A, Pauca VP, Plemmons R: **Algorithms and applications for approximate nonnegative matrix factorization**. *Computational Statistics & Data Analysis* 2006, **52**:155–173.
39. Heinrich K: **Automated gene classification using nonnegative matrix factorization on biomedical literature**. *PhD thesis*, Department of Computer Science, University of Tennessee, Knoxville, TN 2007.
40. Tjioe E, Berry M, Homayouni R: **Using a literature-based NMF model for discovering gene functional relationships**. In *Proceedings of the 2008 International Bioinformatics and Biomedicine Conference, Workshop on Data Mining in Functional Genomics*, Los Alamitos, CA: IEEE Computer Society 2008:185–192.
41. Burkart M, Wren J, Herschkowitz J, Perou C, Garner H: **Clustering microarray-derived gene lists through implicit literature relationships**. *Bioinformatics* 2007, **23**(15):1995–2003.
42. Abrahams B, Geschwind D: **Advances in autism genetics: on the threshold of a new neurobiology**. *Nat Rev Genet* 2008, **9**(5):341–355.
43. Frayling T: **Genome-wide association studies provide new insights into type 2 diabetes aetiology**. *Nat Rev Genet* 2007, **8**(9):657–662.
44. Robinson G: **Cooperation of signalling pathways in embryonic mammary gland development**. *Nat Rev Genet* 2007, **8**(12):963–972.
45. Scheper G, van der Knaap M, Proud C: **Translation matters: protein synthesis defects in inherited disease**. *Nat Rev Genet* 2007, **8**(9):711–723.
46. Wang W: **Emergence of a DNA-damage response network consisting of Fanconi anaemia and BRCA proteins**. *Nat Rev Genet* 2007, **8**(10):735–748.
47. Tjioe E: **Discovering gene functional relationships using a literature-based NMF model**. *PhD thesis*, Genome and Science Technology Program, Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville, TN 2009.
48. Kilpinen H, Ylisaukko-Oja T, Hennah W, Palo O, Varilo T, Vanhala R, von Wendt TN, von Wendt L, Paunio T, Peltonen L: **Association of DISC1 with autism and Asperger syndrome**. *Mol Psychiatry* 2008, **13**(2):187–196. [Epub 2007 Jun 19].
49. Durand C, Betancur C, Boeckers T, Bockmann J, Chaste P, Fauchereau F, Nygren G, Rastam M, Gillberg I, Anckarsär H, Sponheim E, Goubran-Botros H, Delorme R, Chabane N, Mouren-Simeoni M, de Mas P, Bieth E, Rogé B, Héron D, Burglen L, Gillberg C, Leboyer M, Bourgeron T: **Mutations in the gene encoding the synaptic scaffolding protein SHANK3 are associated with autism spectrum disorders**. *Nat Gene* 2007, **39**:25–27.

50. [<http://www.maani.us/charts4/index.php>].
51. [<http://www.ncbi.nlm.nih.gov/sites/entrez?db=gene>].
52. Giles J, Wo L, Berry M: **GTP (General Text Parser) software for text mining**. In *Software for Text Mining, in Statistical Data Mining and Knowledge Discovery*. Edited by Bozdogan H, Boca Raton, FL: CRC Press 2003:455–471.
53. Berry M, Browne M: *Understanding Search Engines: Mathematical Modeling and Text Retrieval*. Philadelphia, PA: SIAM, second edition 2005.
54. Boutsidis C, Gallopoulos E: **On SVD-based initialization for nonnegative matrix factorization**. Tech. Rep. HPCLAB-SCG-6/08-05, University of Patras, Patras, Greece 2005.
55. Lee D, Seung H: **Algorithms for nonnegative matrix factorization**. *Advances in Neural Information Processing Systems* 2001, **13**:556–562.

## Figures

**Figure 1 - FAUN workflow.**

All genes in the gene list are used to construct a gene document collection from which a term-by-gene document matrix is constructed using GTP [52]. The matrix is then factored using rank  $k$  to produce a  $k$ -feature-NMF model. The resulting  $W$  and  $H$  matrix factors are used to extract dominant/significant terms and dominant genes for all  $k$  features. Dominant genes are then correlated for each feature. The FAUN user can annotate any feature and the resulting *annotated* NMF model can be used by the FAUN classifier to classify new gene documents.



**Figure 2 - Five FAUN-generated features for the NatRev collection (110 genes) along with their top (highest intensity) terms.**

Available options such as *show-terms* and *term and entropy filtering* are shown as pull-down menus or slider bars.

**FAUN**

(Feature Annotation Using Nonnegative matrix factorization)  
© 2008-2009, Dr. Michael Berry, Dr. Ramin Homayouni, Dr. Kevin Heinrich, Dr. Elina Tjioe, and Kevin C. Biss

**NMF classification for LL\_NatureRev\_Gene\_July08 collection**

Show Top 10 Terms

**Term Filter**

Selected Term Range: 0 - 10

Highlight

**Entropy Filter**

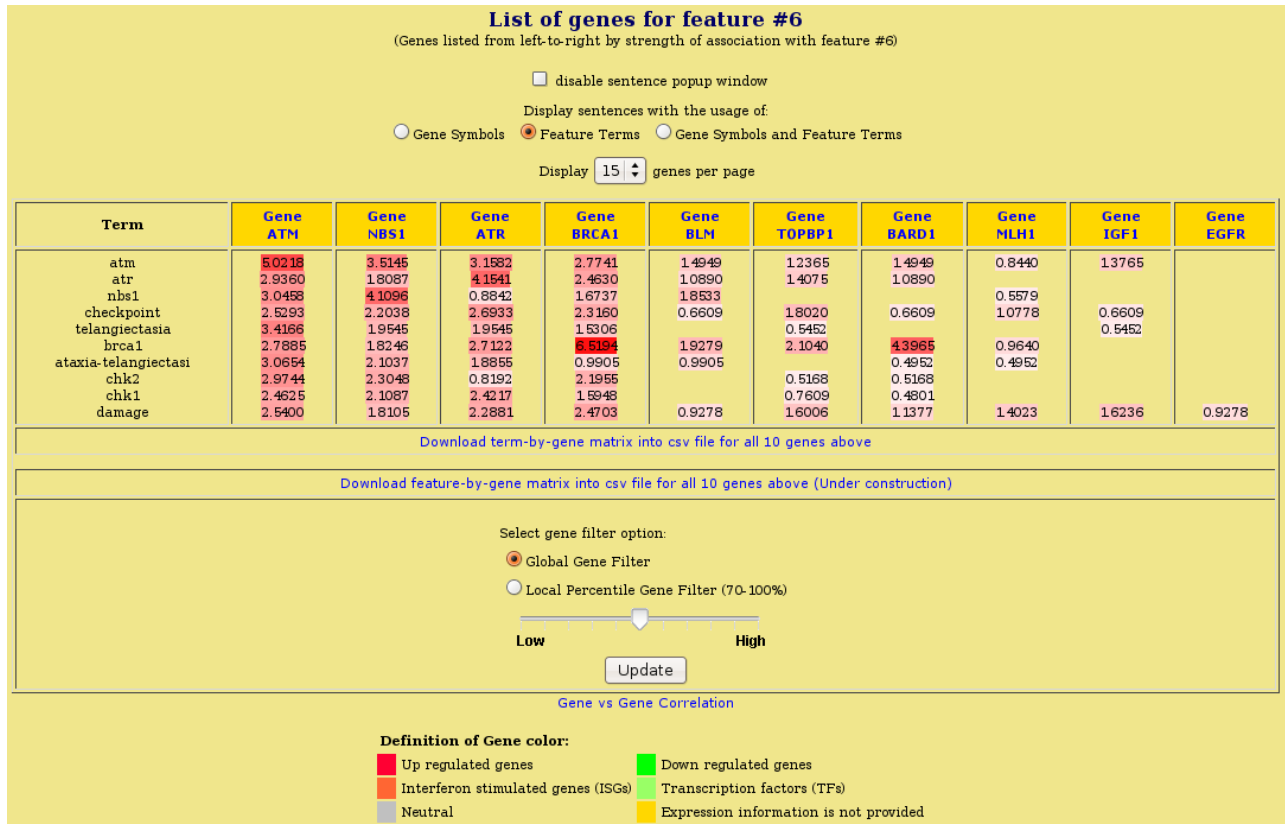
No Low Med High

Highlight

Breast Cancer	Insulin Signaling	Cancer / egf	p par gamma	Feature #5
default -- Feature #1 Gene Count: 93	default -- Feature #2 Gene Count: 5	default -- Feature #3 Gene Count: 2	default -- Feature #4 Gene Count: 2	default -- Feature #5 Gene Count: 5
<input type="text"/> Enter New Label	<input type="text"/> Enter New Label	<input type="text"/> Enter New Label	<input type="text"/> Enter New Label	<input type="text"/> Enter New Label
growth cancer breast tumor factor kinase activation levels patients tumors	igf-1 igf-1 igfbp-3 gh igfbp-1 factor-i insulin-like igf igf-ii protein-3	egfr egf egf-r egf-induced tgf-alpha a431 her-2 epidermal erbb-1 her2	ppargamma ppar peroxisome proliferator-activat pro12ala gamma receptor-gamma troglitazone ciglitazone ppars	p15 ink4a arf p16ink4a cdkn2a p14 p14arf p15 hypermethylation methylation

**Figure 3 - Display of the dominant terms of Feature 6 (DNA damage/ATM) from the higher resolution (rank-30) NMF model of the NatRev collection as they occur in genes highly associated with that feature.**

The *gene filtering* option uses thresholding on components of the *H* matrix factor in the NMF to vary the number of genes displayed. The *display sentences* option allows the user to view a ranked list of sentences (based on term frequency) from any particular gene (document).



**Figure 4 - An illustration of the gene-to-gene correlation FAUN option based on the Pearson correlation of gene features.**

The rightmost window shows the correlation between genes highly associated with the user-selected features 10, 20, and 27; the leftmost window shows the feature strength (and manually annotated labels) for the genes from the user-selected correlation cell.

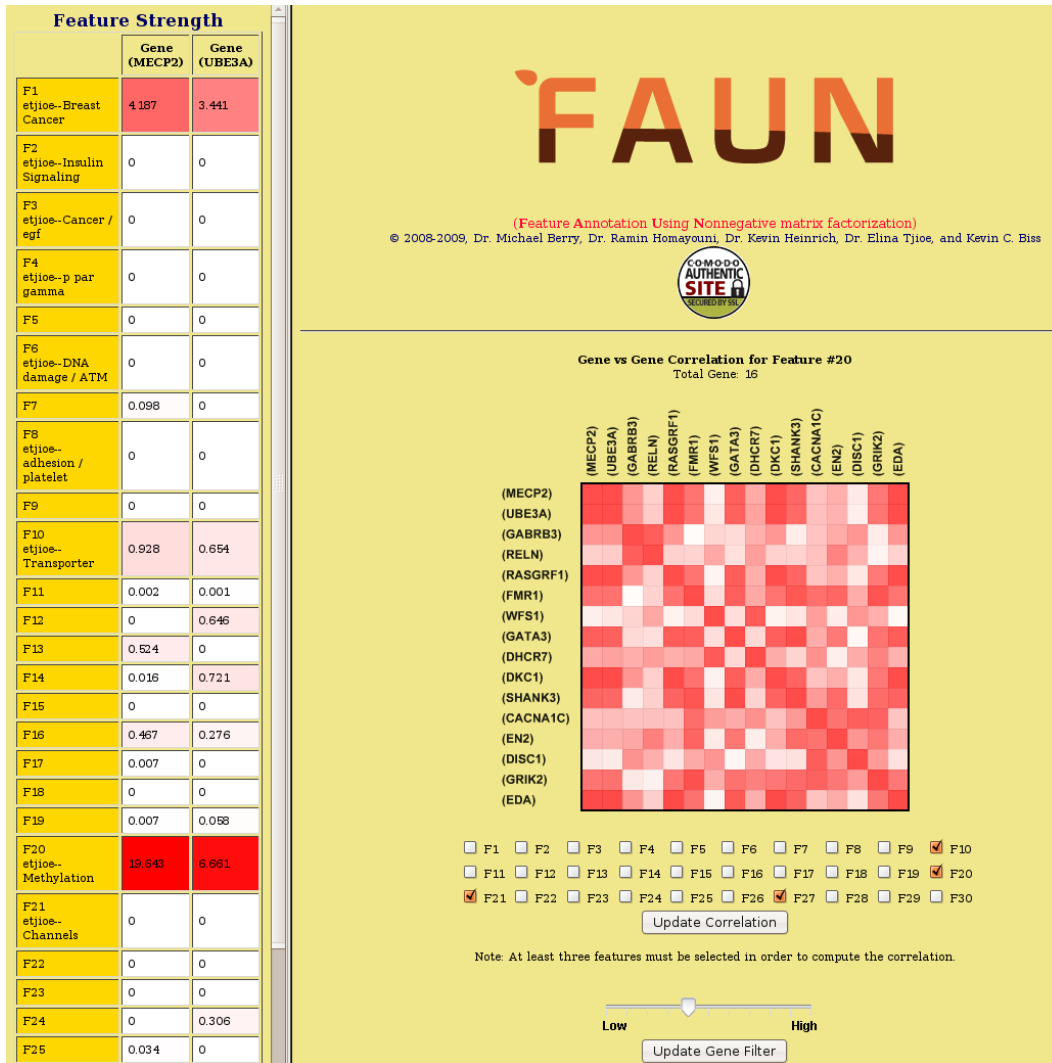


Figure 5 - FAUN classification accuracy based on the strongest feature associated with each gene. [40]

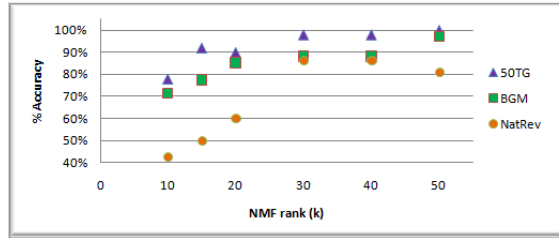


Figure 6 - Venn diagram of genes from different NMF (rank- $k$ ) models generated from Autism gene documents in the NatRev collection.

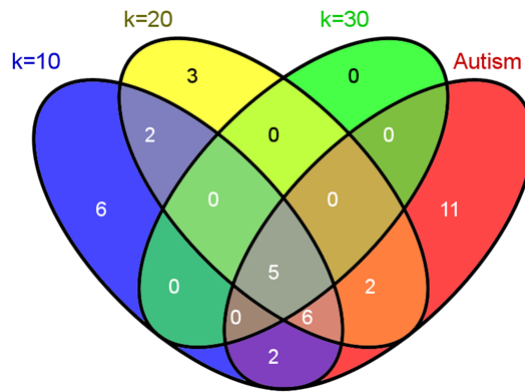
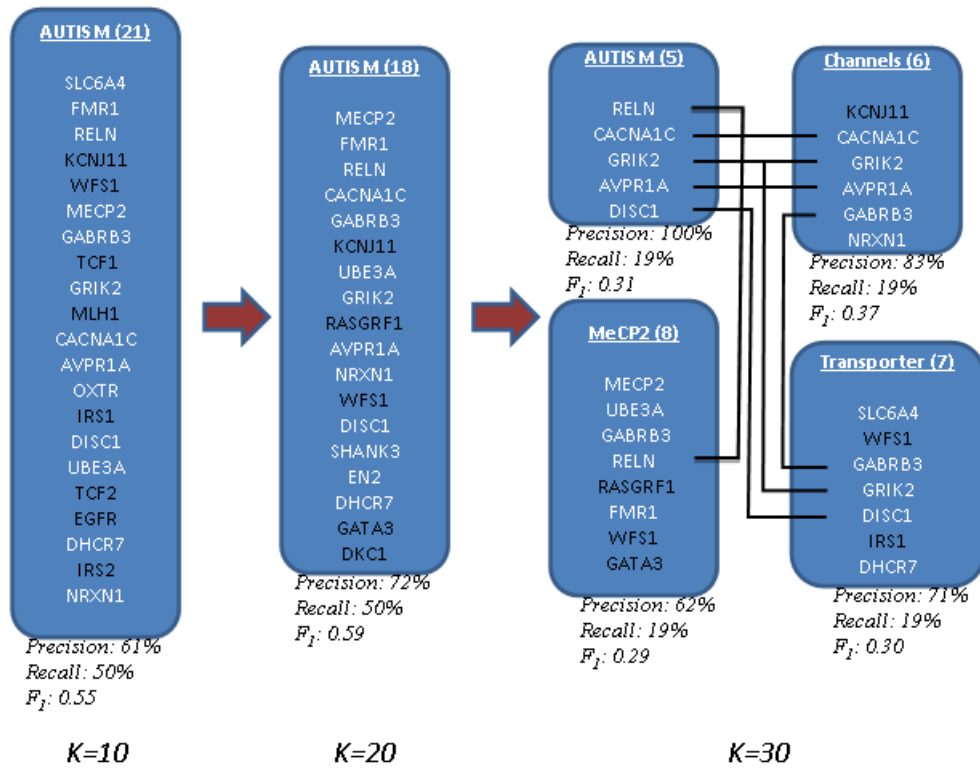
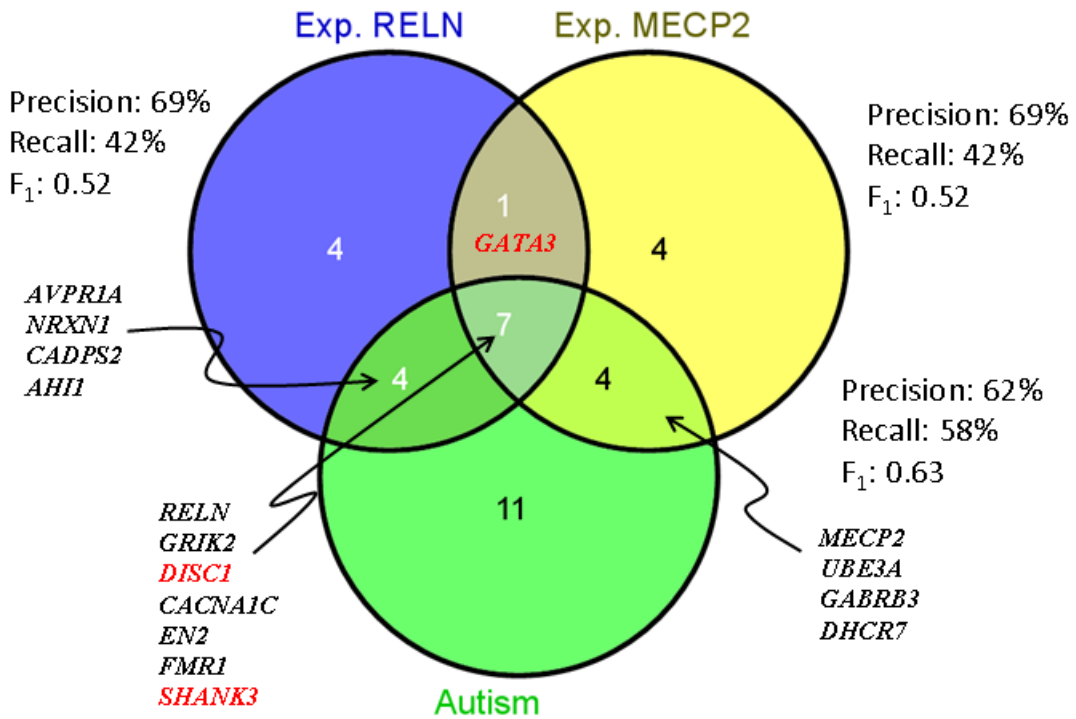


Figure 7 - Gene distributions across different features from NMF (rank- $k$ ) models.



**Figure 8 - Venn diagram of genes associated with Autism (RELN-related, blue) and methylation (MECP2-related, yellow) features using a lower threshold for  $H$ -matrix in the rank-30 NMF model of the NatRev collection.**

The resulting genes sets were compared to the 26 autism associated genes reported by Abrams and Geschwind (green). Red highlighted genes were new discoveries identified by adjusting the rank- $k$  on the same dataset (Figure 7).



## Tables

**Table 1 - List of categories for each dataset used to evaluate FAUN classification performance. GC is the gene count per category.**

### **Dataset 1 (50TG)**

<b>GC</b>	<b>Category</b>
-----------	-----------------

---

15	Cancer
11	Alzheimer
5	Development
16	Cancer & Development
3	Alzheimer & Development

### **Dataset 2 (BGM)**

<b>GC</b>	<b>Category</b>
-----------	-----------------

---

21	Biocarta: Caspase cascade in apoptosis
8	Biocarta: Sonic hedgehog pathway
10	Biocarta: Adhesion and diapedesis of lymphocytes
10	GO: Biological process: telomere maintenance
7	GO: Cellular constituent: cornified cell envelope
20	GO: Molecular function: DNA helicase
8	MeSH: Disease: retinitis pigmentosa
8	MeSH: Disease: retinitis pancreatitis
10	MeSH: Disease: nephroblastoma (Wilm's tumor)

### **Dataset 3 (NatRev)**

<b>GC</b>	<b>Category</b>
-----------	-----------------

---

26	Autism
10	Diabetes
25	Translation
37	Mammary Gland Development
12	Fanconi Anemia

**Table 2 - Computational cost of the NMF multiplicative-update algorithm.**

The number of terms ( $m$ ) for the 50TG, BGM, and NatRev datasets are 8,750, 12,590, and 13,038, respectively. The number of gene documents ( $n$ ) for the 50TG, BGM, and NatRev datasets are 50, 102, and 110, respectively.

NMF Rank $k$	Number of iterations			CPU Time (seconds)			Millions of ops. per iteration		
	50TG	BGM	NatRev	50TG	BGM	NatRev	50TG	BGM	NatRev
10	86	130	92	3.57	14.40	8.20	4	13	14
20	114	133	106	11.55	28.96	15.88	8	26	29
30	162	154	119	28.34	46.57	23.91	13	39	43
40	166	165	147	36.27	70.40	40.59	18	51	57
50	634	171	180	197.96	94.70	61.33	22	64	72

**Table 3 - Matrix of genes by feature terms for the rank-20 NMF model of the NatRev collection.**

Feature Term	neurons	brain	neuronal	mutation	cortex	receptor	mutations	syndrome	transmission	autism	hippocampus	synaptic	subunit	cortical
MCP2	1.4234	0.8412	0.9212	1.0928	1.1452	0.6899	1.3692	1.2904	0.3243	0.7844	0.7315	0.7069	0	0.6969
FMR1	1.0126	0.7729	0.8631	0.904	0.6377	0.6899	0.5615	1.2672	0.753	0.5354	0.826	1.0714	0.6378	0.9514
RELN	1.733	0.9361	1.2408	0.739	1.3854	1.3578	0.7171	0	0.8384	1.6736	0.9642	0.7069	0.2747	1.5863
CACNA1C	1.1558	0.7153	0.7642	0.5424	0.2746	0.8866	0	0.3975	0.6486	0.6756	0.7315	0.9487	1.7651	0
GABRB3	0.6018	0.6095	0.5129	0.4674	0.771	1.6968	0	0.7766	0.753	0.7844	0.413	0.8227	1.6836	0.3001
KCNJ11	1.0126	0.4945	0.3501	0.8783	0.2746	1.3223	1.1457	0.1988	0	0	0.2606	0	1.4201	0.3001
UBES1	0.67	0.594	0.2209	0.7941	0	0	1.042	0.9848	0.514	0.3378	0.413	0.2742	0.2747	0.3001
GRIK2	0.861	0.6627	0.2209	0.1808	0.5493	1.6873	0	0.1988	1.487	1.0134	0.5211	0.8693	1.2425	0
RASGRF1	0.6018	0.6627	0.6202	0	0.2746	0.6899	0.5615	0	0.514	0	0	0.6367	0.2747	0
AVPR1A	0.8216	0.312	0	0	0.9123	1.8916	0	0	0.9105	0.8732	0	0	0	0.8426
NRXN1	0.861	0.3622	0.571	0	0	0.7493	0.317	0	0	0.3378	0	0.7699	0.2747	0
WFS1	0.4108	0.468	0.4418	0.7232	0	0	1.1717	1.1835	0.3243	0	0.413	0	0	0
DISC1	0.2592	0.5397	0.571	0.1808	0.5493	0	0.317	0	0.6486	0	0.826	0	0	0.6002
SHANK3	0.6018	0.4945	0.3501	0	0.5493	0.8007	0	0.7355	0	0	0	0.8693	0	0.3001
EN2	0.2592	0.4033	0	0.4198	0.4353	0	0.317	0	0.3243	0.7844	0.2606	0	0	0
DHCR7	0.4108	0.4033	0	0.768	0	0.423	0.9289	0.9848	0	0.5354	0	0	0	0
GATA3	0.5184	0.2473	0.2209	0.5731	0.2746	1.0427	0.7402	0.5963	0	0	0	0	0	0.3001
DKC1	0.2592	0.156	0	0.5424	0	0	0.8497	0.4615	0	0	0	0	0.5494	0