To the Graduate Council:

I am submitting herewith a dissertation written by Elina Tjioe entitled “Discovering Gene Functional Relationships Using a Literature-based NMF Model.” I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Life Sciences.

Michael W. Berry

Major Professor

We have read this dissertation
and recommend its acceptance:

Michael A. Langston

Igor B. Jouline

Robert C. Ward

Accepted for the Council:

Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Discovering Gene Functional Relationships Using a Literature-based NMF Model

A Dissertation

Presented for the

Doctor of Philosophy Degree

The University of Tennessee, Knoxville

Elina Tjioe

May 2009

Acknowledgements

First and foremost, thanks go to my advisor, Dr. Michael W. Berry. Without him, this dissertation would not have been possible. I deeply thank him for his encouragement that carried me on through difficult times, and for his insights and suggestions that helped to shape my research skills. His valuable feedback contributed greatly to this dissertation. I would also like to express my deep gratitude to Dr. Ramin Homayouni at the University of Memphis for his ideas and valuable guidance through the research process.

I would like to thank the other members of my doctoral committee: Dr. Michael A. Langston, Dr. Igor B. Jouline, and Dr. Robert C. Ward for their helpful suggestions, support and patience.

I also thank other professors at the University of Tennessee, especially Dr. Cynthia B. Peterson, Dr. Nitin U. Jain, Dr. Dean A. A. Myles, Dr. Engin Serpersu, Dr. Hong Guo, Dr. Chris Dealwis, Dr. Elizabeth Howell, Dr. Albrecht von Arnim, and Dr. Daniel M. Roberts. They greatly influenced the way I think about science.

Finally, I would like to express my love and appreciation for my family, my friends and my significant other, Marek, for their continuous support and encouragement during my years as a doctoral student at UTK.

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

Abstract

The rapid growth of the biomedical literature and genomic information presents a major challenge for determining the functional relationships among genes. Several bioinformatics tools have been developed to extract and identify gene relationships from various biological databases. However, an intuitive user-interface tool that allows the biologist to determine functional relationships among genes is still not available. 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. We tested FAUN on three manually constructed gene document collections, and then used it to analyze several microarray-derived gene sets obtained from studies of the developing cerebellum in normal and mutant mice. FAUN provides utilities for collaborative knowledge discovery and identification of new gene relationships from text streams and repositories (e.g., MEDLINE). It is particularly useful for the validation and analysis of gene associations suggested by microarray experimentation. The FAUN site is publicly available at http://grits.eecs.utk.edu/faun.

Table of Contents

List of Tables

List of Figures

Chapter 1.

Introduction

1.1 Research Problems

The MEDLINE 2008 literature database at NIH contains over 17 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 keep track manually 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 subset of genes that interest a researcher.

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.

1.2 Overview of 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) [7] 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) [8], Medical Subject Heading (MeSH) index [9], and KEGG [10]. For example, GoPubMed [11], a thesaurus-driven system, classifies abstracts based on GO, HAPI [12] identifies gene relationships based on co-occurrence of MeSH index terms in representative MEDLINE citations, and EASE [13] 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 [14]. In addition, Kostoff *et al.* [15] 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 [16].

Several alternative approaches that use Medline-derived relationships to functionally group related genes have been reported [17]. Alako *et al.* [18] have developed CoPub Mapper which identifies shared terms that co-occurred with gene names in MEDLINE abstracts. PubGene [19], developed by Jenssen *et al.* constructs gene relationship networks based on co-occurence 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 [20-22].

To make discoveries based on literature, several tools have been developed using indirect or implicit relationships [23, 24]. It is currently not feasible to use them for high-throughput studies [25]. Among many biomedical text mining tools that have been built, Chilibot [26], Textpresso [27], and PreBIND [28] 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 [6]. 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 $Caenorhabditis\ elegans$

(WormBase) literature. Finally, PreBIND provides utilities in the extraction of protein-protein interactions.

One of the most promising vector space model tools 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.* [25] 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 decomposition which can be readily interpreted. NMF is introduced briefly in Section 1.3, and in detail in Section 2.3.

1.3 Brief Introduction to NMF

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. This interpretability property has been utilized in many areas, including image processing and facial pattern recognition [30, 31], protein fold recognition [32], analysis of NMR spectra [33], sparse coding [34], speech recognition [35], video summarization [36], and internet research [37]. 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 [38-44]. Chagoyen *et al.* have shown the usefulness of NMF methods in extracting the semantic features in biomedical literature [40]. Pascual-Montano *et al.* [43] developed an analytical tool called bio-NMF for simultaneous clustering of genes and samples. It requires input of a data matrix (e.g. term-by-doc matrix) and gives output of W and H matrices. Even though the tool is robust and flexible, however, its use by biologists might not be trivial. Therefore, an intuitive user 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 the knowledge discovery and classification of functional relationships among

genes. The ability to facilitate knowledge discovery makes FAUN very attractive. A few examples of knowledge discovery using FAUN are given in Chapter 4. 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 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 different gene documents.

NMF methods are iterative in nature so that the problem involves computational issues such as: proper initialization, rank estimation, stopping criteria, and convergence. To address these issues, many variations of NMF with different parameter choices have been proposed [30, 39, 45, 46]. 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.

Chapter 2 describes the NMF model and the gene document construction process utilized by FAUN. Chapter 3 describes FAUN capabilities and usability. The results of using FAUN as an automated gene (function) classifier are given in Chapter 4. A summary and brief discussion of future work is provided in Chapter 5.

Chapter 2.

Methods

2.1 FAUN Software Architecture

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 the 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 [47]. 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.

2.2 Gene Document Collection

All genes in a given gene list are used to compile titles and abstracts in Entrez Gene [48]. 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) [49]. Terms in the document collection that are common and non-distinguishing are discarded using a stoplist (see Cornell's SMART project repository, ftp://ftp.cs.cornell.edu/pub/smart/English.stop). 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. Hyphen and underscore 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 [50], are used to describe the relative importance of term i for the corresponding document j. That is, the term-by-gene document matrix is defined as

$$A = [w_{ij}] \ ,$$

$$w_{ij} = l_{ij} \times g_i \ .$$

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

$$l_{ij} = \log_2(1 + f_{ij}) \ ,$$

$$g_i = 1 + \left(\frac{\sum_j [p_{ij} \log_2(p_{ij})]}{\log_2 n} \right) \ , \qquad (2.1)$$

$$p_{ij} = \frac{f_{ij}}{\sum_j f_{ij}} \ ,$$

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 best in most LSI-based retrieval experiments reported, 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.

2.3 Nonnegative Matrix Factorization (NMF)

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:

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

$$f(W,H) = \frac{1}{2}\|A - WH\|_{\mathrm{F}}^2 = \frac{1}{2}\sum_{\mathrm{ij}}\left(A_{\mathrm{ij}} - (WH)_{\mathrm{ij}}\right)^2 . \qquad (2.2)$$

This cost function, half of the squared Frobenius norm of the residual error, equals 0 if and only if $A = WH$. This cost minimization problem remains challenging with 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 [45].

The NMF solution depends on the initial conditions for W and H. To address this issue, we are using the Nonnegative Double Singular Value Decomposition (NNDSVD) proposed by Boutsidis and Gallopoulos [51]. This NNDSVD algorithm does not rely upon randomization and is based on approximations of 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 [40], 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 [52] to compute consecutive iterates of W and H:

$$H_{cj} \leftarrow H_{cj} \frac{(W^T A)_{cj}}{(W^T W H)_{cj} + 10^{-9}},$$

$$W_{ic} \leftarrow W_{ic} \frac{(AH^T)_{ic}}{(WHH^T)_{ic} + 10^{-9}}. \qquad (2.3)$$

To avoid division by zero, the 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 can be shown to be $O(k \times m \times n)$ floating-point operations per iteration [53].

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 several gene datasets in Chapter 4.

To compensate for uncertainties in the data or to enforce desired structural properties in the factor matrices, additional application-dependent constraints can be added, modifying the cost function of Equation (2.2) as follows:

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

where α and β are relatively small regularization (control) parameters and $J_1(\mathrm{W})$ and $J_2(\mathrm{H})$ are functions defining additional constraints on W and H, respectively. To apply *smoothness* to the W matrix, which essentially bounds the relative magnitudes of the factor's components, set:

$$J_1(W) = \|W\|_F^2\,.$$

Hence, this constraint penalizes W solutions of large Frobenius norm. This type of constraint can certainly be applied to H as well. As shown in [54], if smoothing is enforced on both W and H, the multiplicative update rules become:

$$H_{cj} \leftarrow H_{cj}\,\frac{(W^T A)_{cj} - \beta H_{cj}}{(W^T W H)_{cj} + 10^{-9}},$$

$$W_{ic} \leftarrow W_{ic} \frac{(AH^T)_{ic} - \alpha W_{ic}}{(WHH^T)_{ic} + 10^{-9}}.$$

Sparsity constraints on either W or H can similarly be applied. Such constraints were introduced by Hoyer [34] to enforce a statistical sparsity in the resulting matrix factors. Increasing the sparsity of the H matrix, for example, might enhance the visibility of features in W [34, 55]. To apply sparsity to the H matrix, set:

$$J_2(H) = (\omega \|vec(H)\|_2 - \|vec(H)\|_1)^2 ,$$

where $\omega = \sqrt{kn} - \left(\sqrt{kn} - 1\right)\gamma$ and $vec(H)$ is the vec operator that transforms a matrix into a vector by stacking the columns of H. The sparsity parameter (s) is defined by γ with a value between 0 and 1. The multiplicative update rule for H can then be written as:

$$H_{cj} = H_{cj} \frac{(W^T A)_{cj} - \beta(c_1 H_{cj} + c_2)}{(W^T WH)_{cj} + 10^{-9}},$$

where

$$c_1 = \omega^2 - \omega \frac{\|vec(H)\|_1}{2\|vec(H)\|_2},$$

$$c_2 = \|vec(H)\|_1 - \omega \|vec(H)\|_2 .$$

We investigate the effect of additional smoothing and sparsity constraints in the corresponding objective function. The usefulness of applying smoothness and sparsity constraints has been shown in numerous NMF applications [41, 45, 46, 50, 56, 57]. Gao

et al. [41] observed improved clustering results when applying sparsity constraints on the H matrix on gene-expression data. Berry et al. [45] demonstrated that the addition of smoothness constraints on the W matrix produced a higher quality of features in the NMF application of the extraction and detection of concepts or topics from the Enron electronic mail collection. High-entropy terms were shown to be conserved, which could be used for a critical task such as tracking specific topics or discussions. It is plausible that the conservation of high-entropy terms could yield a control vocabulary for the classification function of FAUN.

In light of recent advancements in computing sparse NMF solutions, we compare our NMF algorithm with the sparse nonnegative matrix factorization (SNMF) algorithm proposed by Kim and Park [46]. This algorithm attempts to solve the following optimization problem:

$$\min_{W,H} \frac{1}{2} \left\{ \|A - WH\|_F^2 + \eta \|W\|_F^2 + \beta \sum_{j=1}^{n} \|H(:,j)\|_1^2 \right\}, \quad s.t.\ W, H \geq 0,$$

where $H(:,j)$ is the j-th column vector of H, β is a sparseness parameter for H and η is used to suppress $\|W\|_F^2$. Each iteration involves solving two nonnegativity constrained least squares problems:

$$\min_{H} \left\| \begin{pmatrix} W \\ \sqrt{\beta}\, e_{1\times k} \end{pmatrix} H - \begin{pmatrix} A \\ 0_{1\times n} \end{pmatrix} \right\|_F^2, \qquad s.t.\ H \geq 0,$$

$$\min_{W} \left\| \begin{pmatrix} H^T \\ \sqrt{\eta} I_k \end{pmatrix} W^T - \begin{pmatrix} A^T \\ 0_{k \times m} \end{pmatrix} \right\|_F^2, \qquad s.t.\ W \geq 0,$$

where $e_{1\times k}$ is a row vector with k components equal to one, $0_{1\times n}$ is a zero vector with n components, I_k is an k-dimensional identity matrix and $0_{k\times m}$ is a zero matrix of dimension $k \times m$. The least squares sub-problems are solved using a fast nonnegativity constrained least squares algorithm [58].

For demonstration purposes, an NMF application to a small document collection in [38] is reproduced in Appendix A. The sample collection of 9 documents (Table A. 1) was parsed using GTP [49] to produce a term-by-document matrix A (Table A. 2). Using the restriction that a dictionary term must occur more than once in the document and in the collection, matrix A consists of 24 dictionary terms and has size of 24×9. Table A**. 3** and Table A. 4 represent W and H matrices as one possible factorization solution of matrix A using factorization rank $k = 4$. The approximation of matrix A given by the multiplication of $W \times H$ is given in Table A. 5. The top 5 weighted terms for each feature are shown in Table A. 6. Table A. 7 shows a reorganized matrix A by assigning each document and term to its most dominant feature. The ability to reveal the meaning of each NMF feature (column of W) using its dominant (largest magnitude) components drives the annotation process in FAUN.

2.4 FAUN Workflow

The workflow design for FAUN is shown in Figure 1, as presented in [59]. A gene list provided by a FAUN user is used to create a raw term-by-gene document (sparse) matrix upon which an NMF model is built. The matrix for FAUN is created the same way as in the SGO [25] and is described in Section 1.2. The term-by-gene document matrix is then decomposed using the NMF methods described in Section 2.3 [60]. 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 of the NMF model, respectively. The proper range of k could be determined based on user feedback in the future. 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 each feature. FAUN users can then annotate some or all of the features. The annotated NMF model is later used by the FAUN classifier to determine the (concept-based) features of a new gene document.

2.5 FAUN Classifier

The FAUN classifier accepts a new document to be classified, the entropy weights of terms in Equation (2.1) used in the NMF model, the term-by-feature W matrix factor, 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

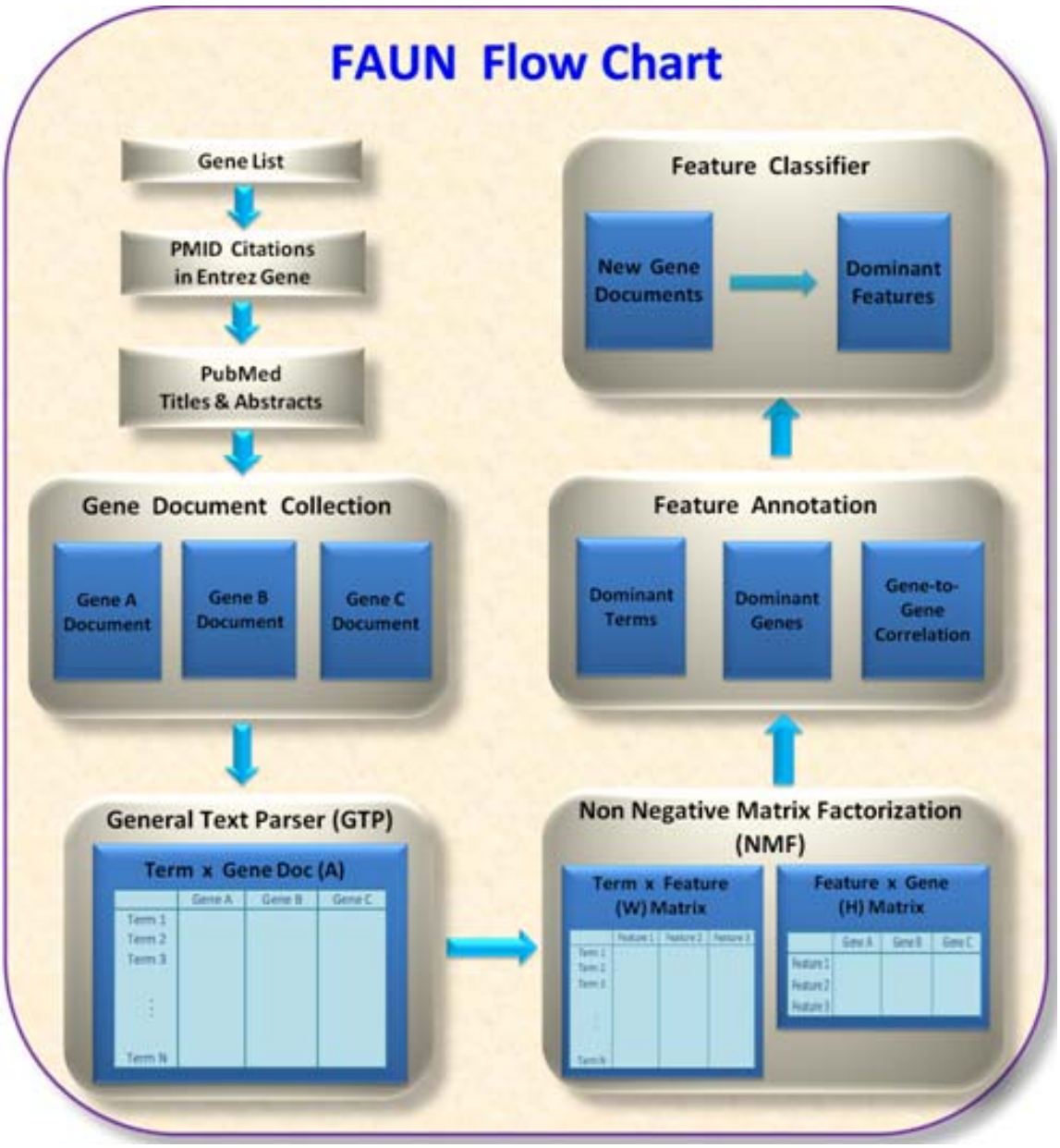


Figure 1. FAUN Flow Chart

All genes in the gene list are used to construct a gene document collection which is used to build a term-by-gene document matrix using GTP. The matrix is then factored using rank k to produce a k-feature-NMF model for the gene document collection. 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 each feature. FAUN users can then annotate some or all of the features. The annotated NMF model is later used by the FAUN classifier to classify the new gene documents.

the term frequency threshold. It then outputs the features sorted by weight from the highest to the lowest. Pseudocode for the FAUN classifier is provided in Figure 2A. The process of mapping features to gene classes is described in Section 2.6.

Preliminary testing indicated that the classifier accuracy is around 80%. The test was conducted based on the first dataset (see below for details) that contains 50 genes. NMF models were first built with ranks k = 10, 20, 30 and 40 using 40 genes selected randomly from the 50 gene dataset. The classifier was then trained using the W matrix in newly built NMF models. The accuracy was tested using the remaining 20% of the gene documents.

2.6 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 at the FAUN site by FAUN users while exploring the gene dataset. Features in the NMF models for the 50TG dataset, for example, have been annotated manually by a 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 (shown below). This automated annotation process for the 50TG dataset gave the same results as manual annotation. The script accepts the H matrix from the NMF model, known classification categories for genes (class), the NMF rank k, and a feature weight threshold. The script then performs the algorithm shown in Figure 2B for assigning feature F with accepted classes (F).

A. Pseudocode for FAUN Classifier:

1. Read terms T with their entropies from the key file: entropy (T)
2. Read term weights for each feature F: term_weight (T, F)
3. Read stopwords
4. Compute the frequencies of all terms in the document: frequency(T)
5. Total_terms = total number of terms in the document
6. Compute feature weights:

```
for each feature F:
    weight(F) = 0
        for each term T in F:
        if entropy(T) >= entropy_thres and frequency(T) >
doc_freq_thres:
        w = term_weight(T, F) * entropy(T) * log(1 +
frequency(T)/total_terms)
        weight(F) = weight(F) + w
```

7. Sort all features by weight in decreasing order
8. The document is then classified based on its top feature (i.e., with the largest weight)

B. Pseudocode for automated FAUN Annotation:

1 Let weight (f, g) be the weight of feature f with respect to gene g
2 Let class (g) be the class assigned to gene g
3 For each feature F:

Let G = the set of genes for which F is the top feature (the feature with max weight)
Let L = { class(g) for all g in G }
For each label l in L:
Let max_weight (F, l) = max { weight (F, g) for each gene g in G such that class $(g) = l$ }
Let sum_weight (F, l) = sum { weight (F, g) for each gene g in G such that class $(g) = l$ }
Let max_sum (F) = max { sum_weight (F, l) for l in L }
Let accepted_classes (F) = (l for each l in L
such that sum_weight $(F, l) \geq$ max_sum$(F) \times$ threshold
sorted by max_weight (F, l) in descending order)

Figure 2. Pseudocodes for FAUN Classifier and Automated FAUN Annotation

Chapter 3.

FAUN Capabilities and Usability

The infrastructure of the FAUN bioinformatics software environment has been built and is available online with restricted access at http://grits.eecs.utk.edu/faun. Once the FAUN user provides a gene list, the document collection and NMF models will be generated on the server. The user can later retrieve information about the gene list from the FAUN site. The main functionalities of FAUN (along with illustrative screenshots) are described below. The screenshots were taken from the first dataset (50TG collection) which contains 50 manually selected genes related to development, Alzheimer's disease, and cancer biology [25].

3.1 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 #5) for future reference. The screenshot of some annotated *features* and their top associated *terms* for the 50TG gene document collection (dataset 1) are shown in Figure 3. The entropy filter slider bar allows the user to get some 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 indicator for the model. 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 3. 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. The screenshot in Figure 3 is taken from a high resolution NMF model with 20 features. It only shows the first 10 features. 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 3 could readily be labeled (or annotated) as a descriptor of Alzheimer's disease. Note that Alzheimer's disease is characterized by the formation of ***beta-amyloid*** protein plaques that build up in the brain; the ***APP*** gene has been linked to Alzheimer's disease; accumulation of ***amyloid*** beta peptide (***Abeta***) in the brain has been linked to the development of Alzheimer's disease [61]; Apoliproprotein E (***apoE***) has been shown to exhibit an isoform-specific

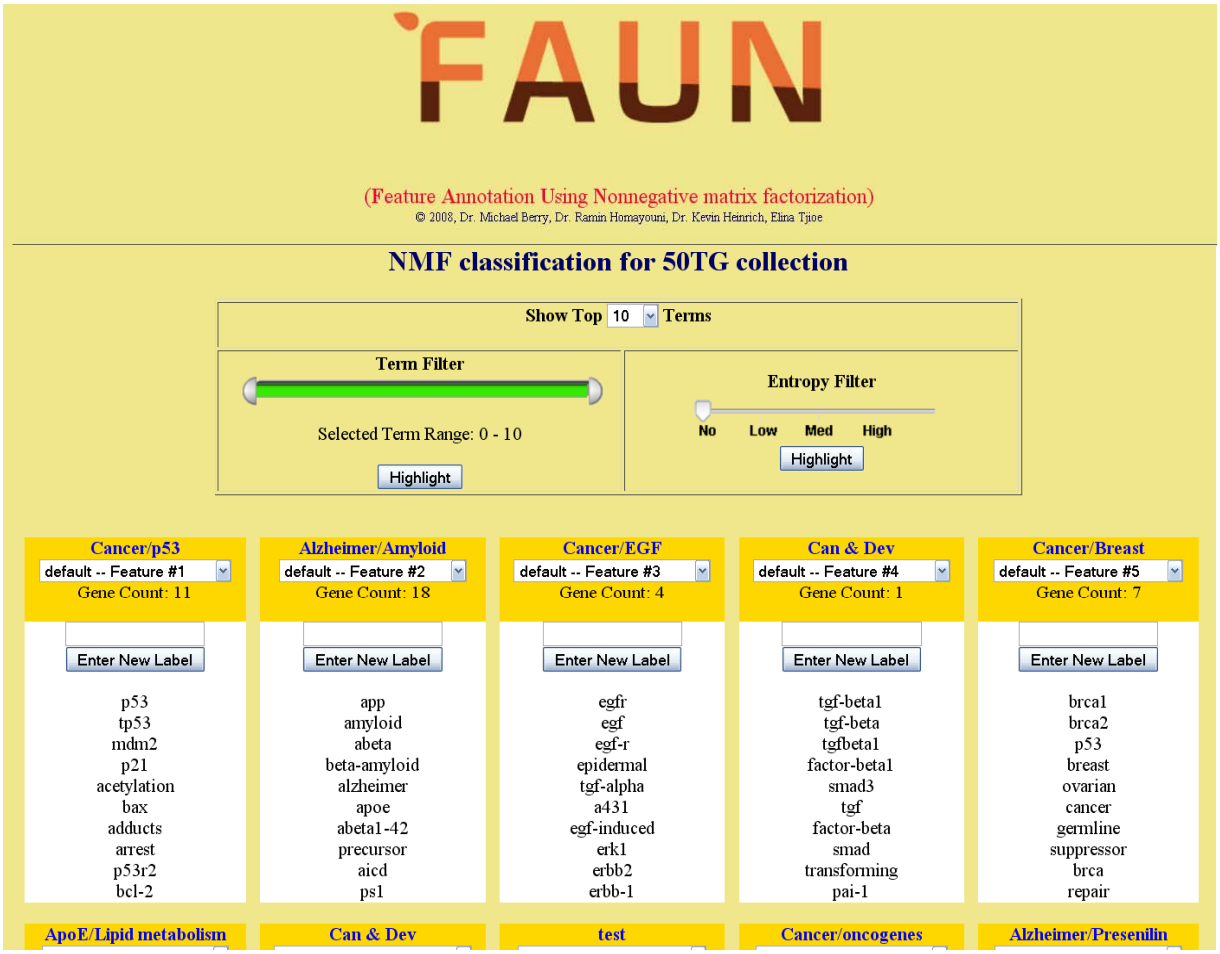


Figure 3. FAUN Concept Features

Some features for 50TG collection and their top associated terms are shown. The transparent box at the upper center of the page contains the show-terms option, term and entropy filter slider bars.

association with Alzheimer's disease [62]; and mutations in the ***PS1*** gene could cause early onset of Alzheimer's disease [63]. Note that feature #2 contains all the bold and italic terms mentioned.

If the user is interested in exploring further the "Alzheimer" feature, the user can then click on the 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.

3.2 Identifying genes in a feature

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 H 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's 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 Figure 3

and Figure 4. The screenshot of *genes* that are highly associated with feature #2 and their top associated *terms* for the 50TG collection, along with some options, are shown in Figure 4. These options are the popup window option, display sentences option, display-genes option, and gene filter option.

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 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 which works the best for every case. To address thresholding problems, analysis of several thresholding algorithms has been conducted [64]. 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 70^{th} local percentile. The global gene filter is set to medium (feature weight = 1.0) by 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.

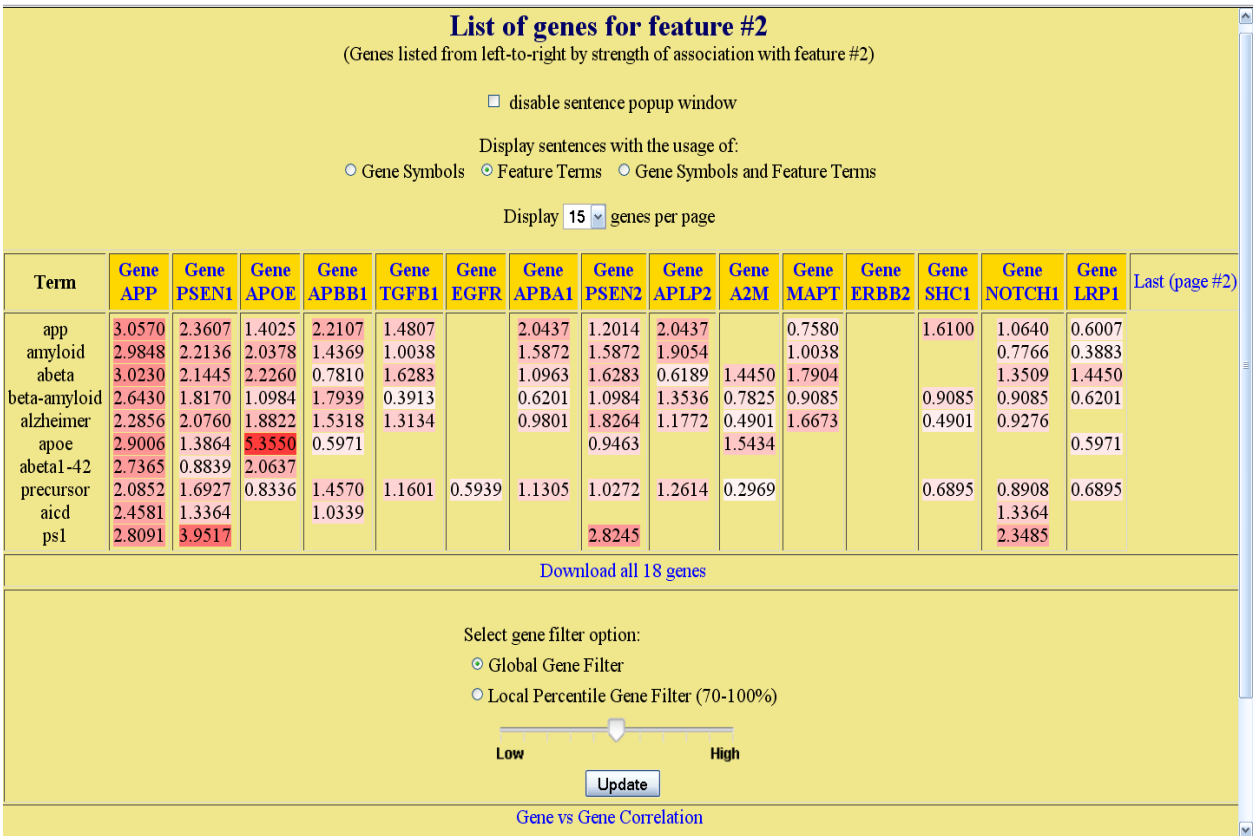


Figure 4. List of Genes for Selected Feature

Use of dominant terms across genes highly associated with the user-selected feature (e.g. feature #2). Options shown are the popup window option, display sentences option, display-genes option, and gene filter option.

Furthermore, terms and/or annotation for a feature can be used as keyword queries in SGO (mentioned in Section 1.2) or other SGO-like software to retrieve genes which might be missed by FAUN.

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 what genes are associated with some particular 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 4.

3.3 Exploring gene relationships

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 total selected n features is estimated using the Pearson correlation coefficient (r) which can be written:

$$r_{xy} = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\sqrt{n\sum y_i^2 - (\sum y_i)^2}}. \quad (0.1)$$

The Pearson correlation matrix for all the genes is then generated. The correlation is color-coded for visual analysis, with more red for stronger correlation. An example of the correlation matrix for all the genes in feature #2 of the 50TG collection is shown in Figure 5. Users can view the correlation of any particular genes x and y with respect to any combinations of features with a minimum of 3 features selected. By default, the user-selected feature (feature #2) and its left and right neighboring features (feature #1 and #3) are selected.

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 features are involved by clicking on the correlation cell shown in Figure 5. The selected cell in the figure shows a strong correlation, indicated by the red cell color, between gene ERBB2 and gene EGFR, mainly due to feature #3, suggested by the feature strength at the left side. Feature #3 has been labeled as a cancer feature. **It is interesting to show that FAUN is able to show that two genes involved in Alzheimer's disease are related strongly due to the cancer feature.**

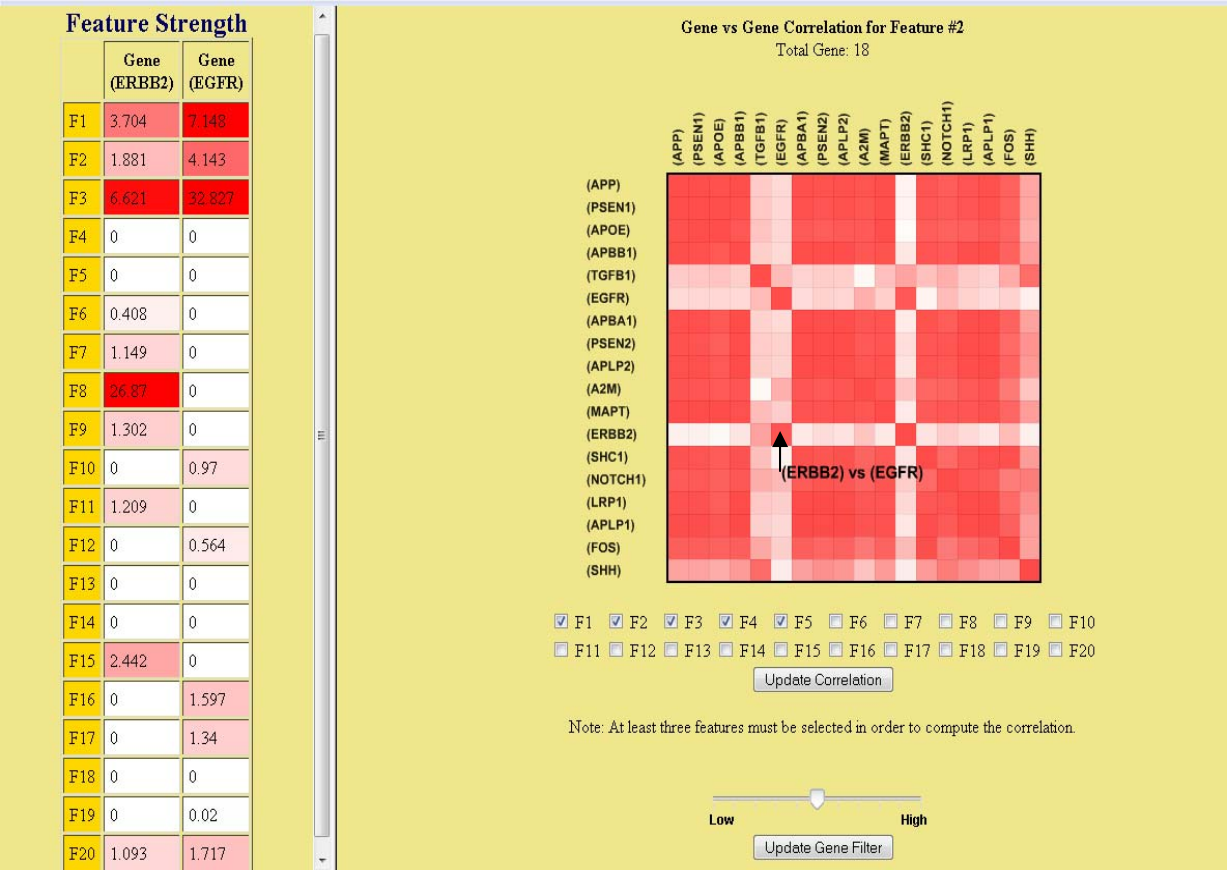


Figure 5. Gene-to-gene Correlation and Feature Strength

The right window shows the correlation between genes highly associated with the user-selected feature; the left window shows the feature strength for the genes from the user-selected correlation cell (pointed to by an arrow).

3.4 Classifying new gene documents

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. It is an important feature to have due to the fact that scientific papers are published daily. 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 will be 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 changing literature is advisable.

3.5 Discovering novel gene functional relationships

One of the most important capabilities in FAUN is to discover novel gene functional relationships. Several knowledge discovery examples have been made using FAUN. As illustrated in Section 3.3, FAUN discovered two cancer genes involved in Alzheimer's disease. This fact was presented in [65] and was not included in the dataset analyzed by FAUN. Other knowledge discovery examples will be given in Section 4.

Chapter 4.

Results

4.1 Gene Datasets

For a preliminary assessment of FAUN feature classification, FAUN is tested on three manually constructed gene document datasets with known functional gene relationships.

The first dataset (50TG collection) is a gene document collection of 50 manually selected genes related to development, Alzheimer's disease, and cancer biology [25]. The second dataset (BGM collection) is composed of three non-overlapping gene lists from the Biocarta, Gene Ontology and MeSH databases [66]. The third dataset (NatRev collection) is composed of five gene lists selected from Nature Reviews papers [67-71]. Categories used in all of these datasets are shown in Table 1.

The genes comprising each dataset along with other associated information is presented in Table B. 1, Table B. 2, and Table B. 3 in Appendix B for the 50TG, BGM, and NatRev datasets, respectively.

Table 1.

List of categories for each dataset used
to evaluate FAUN classification performance.

	Dataset 1 (50TG)	
	Categories	**# of genes**
1	Cancer	15
2	Alzheimer	11
3	Development	5
4	Cancer & Development	16
5	Alzheimer & Development	3
	Dataset 2 (BGM)	
	Categories	**# of genes**
1	Biocarta: Caspase cascade in apoptosis	21
2	Biocarta: Sonic hedgehog pathway	8
3	Biocarta: Adhesion and diapedesis of lymphocytes	10
4	GO: Biological process: telomere maintenance	10
5	GO: Cellular constituent: cornified cell envelope	7
6	GO: Molecular function: DNA helicase	20
7	MeSH: Disease: retinitis pigmentosa	8
8	MeSH: Disease: chronic pancreatitis	8
9	MeSH: Disease: nephroblastoma (Wilm's tumor)	10
	Dataset 3 (NatRev)	
	Categories	**# of genes**
1	Autism	26
2	Diabetes	10
3	Translation	25
4	Mammary Gland Development	37
5	Fanconi Anemia	12

4.2 Input Parameters

Initialization is based on [38]: NNDSVD and randomization. NNDSVD starts with the truncated SVD of starting matrix A. Although NNDSVD produces a static starting point, different methods can be applied to remove zeros from the initial approximation and thereby prevent them from being fixed throughout the update process. NNDSVDz keeps zero elements at the original positions, while NNDSVDa, NNDSVDe, and NNDSVDme replace the zero elements with the average of all elements, ϵ (10^{-9}) , or $\epsilon_{machine}$, respectively.

The values of factorization ranks are 5, 10, 15, 20, 25, 30, 40, and 50. For datasets 2 and 3, we didn't consider ranks 5 and 25. We restricted the maximum number of iterations to 1000 and 2000 and stopped iteration if the consecutive values of W and H are closer than $\tau = 0.01$ and $\tau = 0.001$, respectively, in Frobenius norm,

$$\|W_{old} - W_{new}\| < \tau \quad \text{and} \quad \|H_{old} - H_{new}\| < \tau. \tag{4.1}$$

We use the multiplicative update algorithm to compute consecutive values of W and H, as defined in Equations (2.3).

We also determine the effect of smoothing and sparsity. Control parameters α (for smoothing W) and β (for smoothing H) are chosen as 0.1, 0.01, and 0.001, and sparsity parameter s is chosen as 0.1, 0.5, and 0.9.

4.3 Evaluation Approaches

Classification accuracy based on the strongest feature

For each dataset, six NMF models are generated with rank k = 10, 15, 20, 30, 40 and 50. Features in the NMF models are annotated using the automated annotation process which is described in Section 2.6. Then, using these annotated features, each gene in the dataset is assigned to the same category with its strongest annotated feature (indicated by the largest corresponding column entry of H). For example, if gene A has the strongest association with feature A, then gene A will be classified as being in the category with which feature A 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 6). This is done using NNDSVDz initialization with no additional smoothing or sparsity constraints.

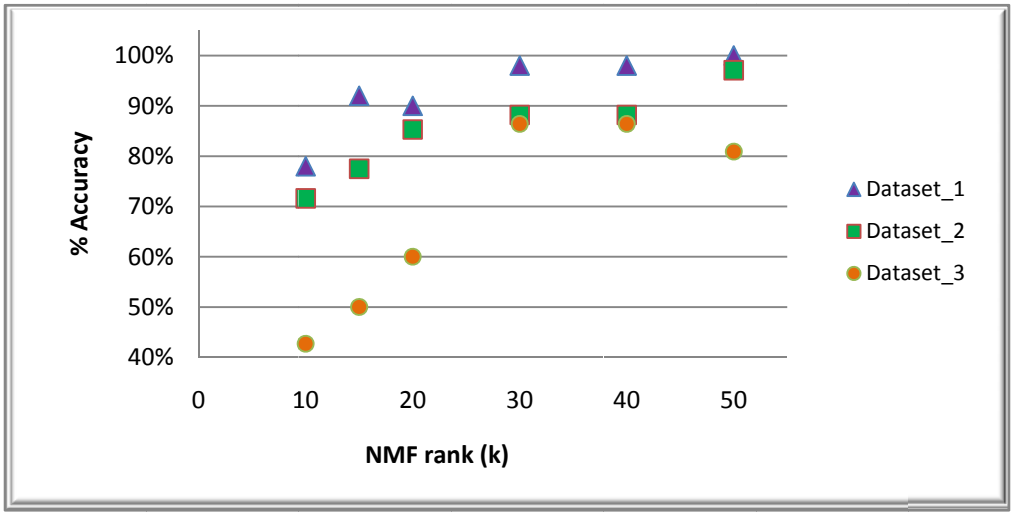


Figure 6. FAUN Classification Accuracy Based on the Strongest Feature

Classification accuracy based on the total gene recall

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

The classification accuracy is evaluated in a fuzzy way such 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 1/(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%).

The FAUN gene recall results with several thresholds are shown in Figure 7. The accuracy with feature weight threshold = 1.0 ranges from 78%-100%, 71.6%-97.1%, and 42.7%-80.9% for datasets 1, 2, and 3, respectively.

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. In the next Subsection, a few examples of such discovery will be presented.

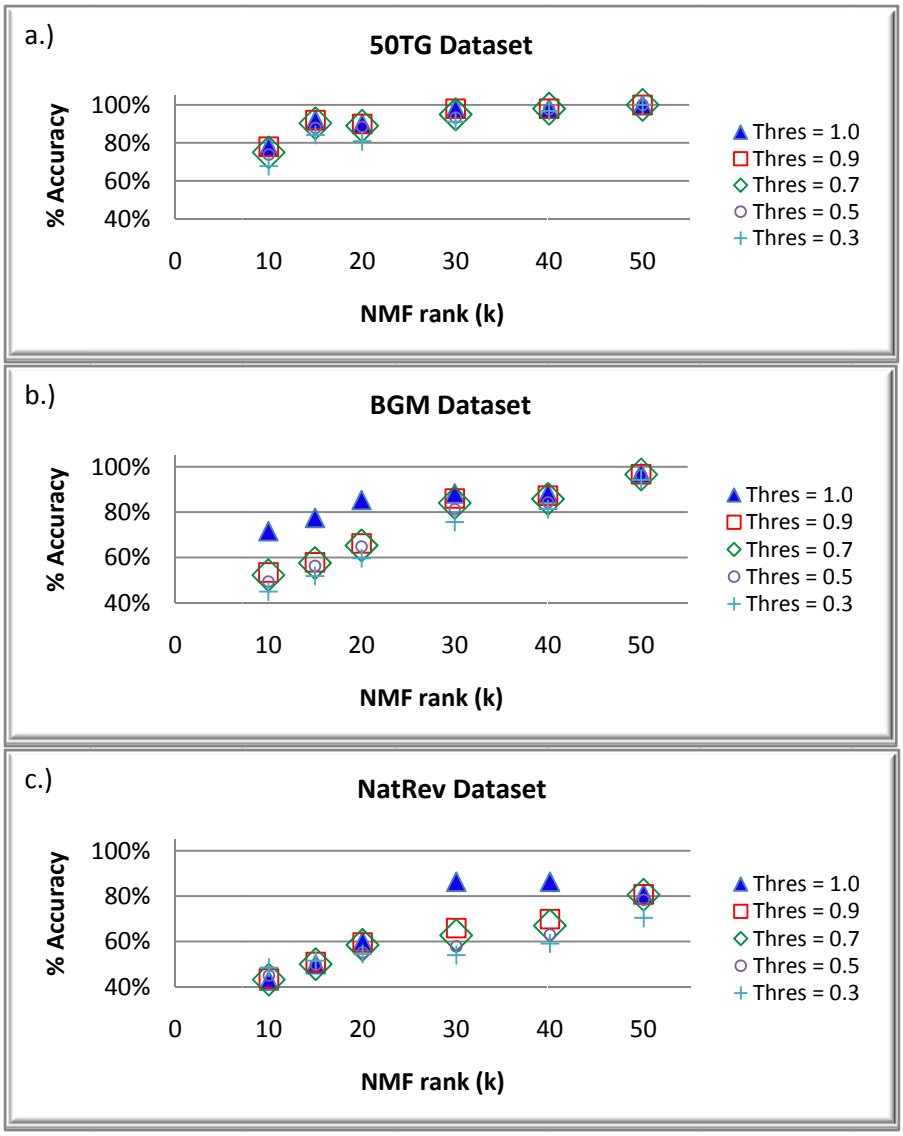


Figure 7. FAUN Classification Accuracy Based on the Total Gene Recall

FAUN classification accuracy based on the total gene recall for each category represented by one or more features. For each feature, genes above the set feature weight threshold will be assigned at least to that feature. The accuracy for datasets 1, 2, 3 with feature weight threshold ranging from 0.3 to 1.0 is shown in a, b, and c, respectively. NMF models are generated using NNDSVDz initialization and no smoothing or sparsity constraint.

Examples of knowledge discovery using FAUN

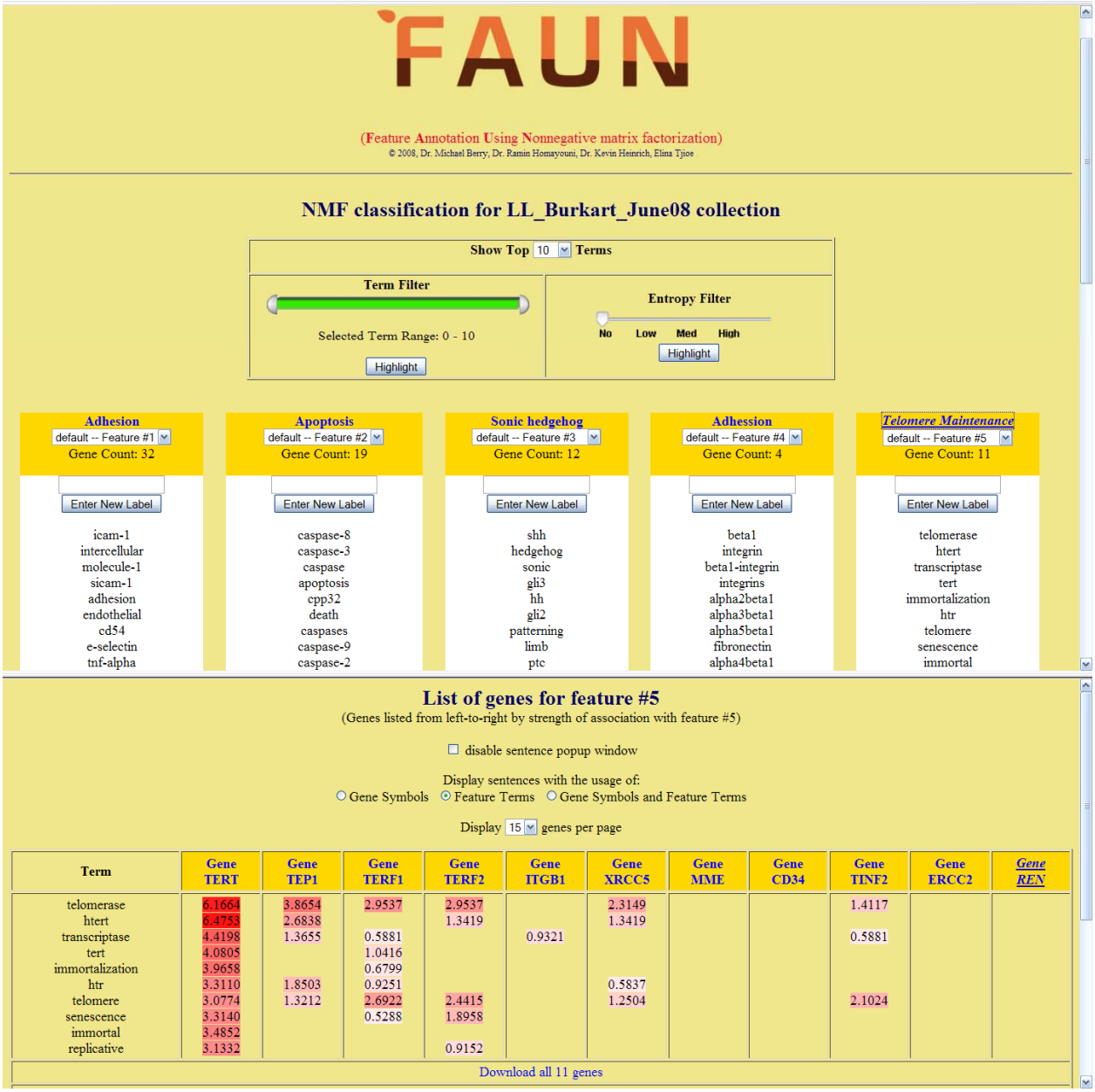
FAUN was able to discover new knowledge in the 50TG and BGM gene datasets. For the 50TG dataset, FAUN associated two genes, ERBB2 and EGFR, with cancer and Alzheimer features. This is considered new knowledge because these genes were originally classified only as cancer genes.

For the BGM gene dataset, FAUN revealed that the gene REN (renin), involved in nephroblastoma, might also be involved in telomere maintenance. FAUN associated the renin gene in both the nephroblastoma and telomere maintenance features. The screenshots of a high resolution NMF model for the BGM dataset showing the renin gene associated with both features are shown in Figure 8 and Figure 9. This association was recently realized and confirmed by Vasan R. S. *et al.* [72].

FAUN has also been applied to a new cerebellum gene set. Interestingly, a FAUN-based analysis of the gene set has revealed new knowledge – **the gene set contains a large component of transcription factors**. It was only when the domain expert used FAUN (as opposed to other similar bioinformatics tools) that this knowledge became apparent. Copyright restrictions prevent further elaboration at this time.

Comparison with the sparse NMF (SNMF) algorithm

We compared our NMF algorithm with the sparse NMF (SNMF) algorithm of Kim and Park [46]. The accuracy of SNMF was quite comparable to that of our multiplicative update NMF algorithm. We tested Kim and Park's sparse NMF using their Matlab implementation with random initialization and the sparseness control parameter (β)



Term	Gene TERT	Gene TEP1	Gene TERF1	Gene TERF2	Gene ITGB1	Gene XRCC5	Gene MME	Gene CD34	Gene TINF2	Gene ERCC2	Gene REN
telomerase	6.1664	3.8654	2.9537	2.9537		2.3149			1.4117		
htert	6.4753	2.6838		1.3419		1.3419					
transcriptase	4.4198	1.3655	0.5881		0.9321				0.5881		
tert	4.0805		1.0416								
immortalization	3.9658		0.6799								
htr	3.3110	1.8503	0.9251			0.5837					
telomere	3.0774	1.3212	2.6922	2.4415		1.2504			2.1024		
senescence	3.3140		0.5288	1.8958							
immortal	3.4852										
replicative	3.1332			0.9152							

Figure 8. FAUN screenshot of the BGM dataset showing genes associated with telomere maintenance feature

The top window shows some features for the BGM collection and their top associated terms; the bottom window shows genes highly associated with feature #5 which has a label of telomere maintenance.

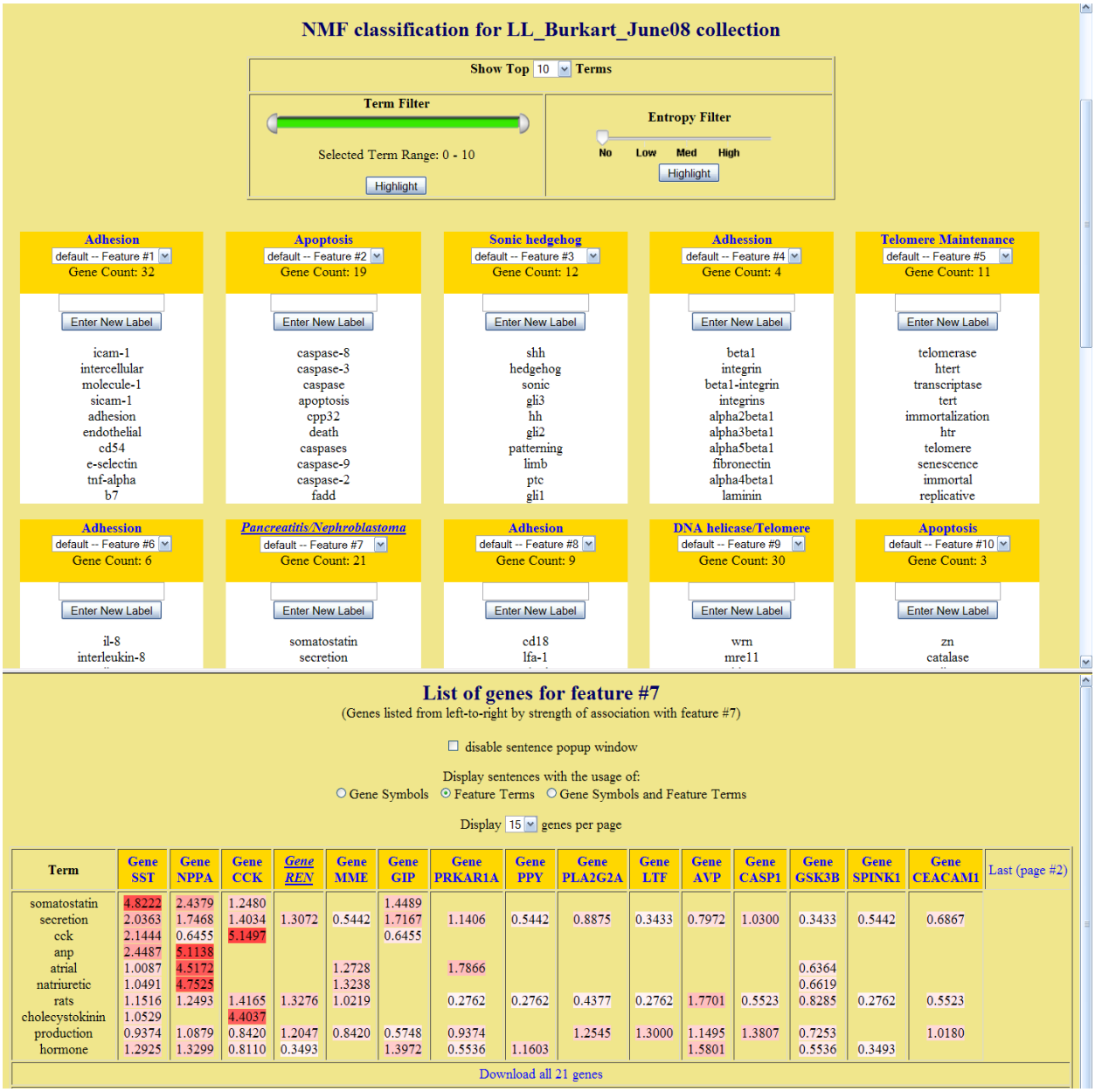


Figure 9. FAUN screenshot of the BGM dataset showing genes associated with nephroblastoma feature

The top window shows some features for the BGM collection and their top associated terms; the bottom window shows genes highly associated with feature #7 which has a label of pancreatitis/nephroblastoma.

equal to 0.01. The comparison in classification accuracy with our default NMF algorithm using random and NNDSVDz initializations (with no additional constraints) is shown in Figure 10.

For our NMF algorithm, the cost per each NMF iteration ranged from 0.004 seconds to 0.011 seconds per k. The test was conducted on a PC with Intel Core 2 CPU T5600 @ 1.83 GHz processor, 1.5GB memory, running 32 bit Linux. In the case of SNMF, there can be great variance in the time it takes to perform one iteration. In our tests, this time varied from as low as 0.061 seconds to 2.157 seconds per iteration per k (Table 2). This variation depends on the convergence of the embedded iterative nonnegativity constrained least squares algorithm used in SNMF. The number of outer iterations in SNMF is significantly lower than that of our NMF algorithm with random initialization, but quite comparable to our NMF algorithm with NNDSVDz initialization. Theoretically, one can estimate the complexity of one iteration of SNMF to be $k^4 \times \langle m+n \rangle$ compared to $k \times m \times n$ for one iteration of the simple multiplicative update algorithm in Equations (2.3).

NMF rank effect

As shown in Figure 10, the classification accuracy in general increases with the increase of NMF rank (k). Higher values of k tend to reveal more specific or localized features in the literature. It is a difficult problem to determine the optimum value of NMF rank k. Brunet $et.al.$ [73] proposed the use of cophenetic correlation coefficient to find the optimal k that gives stable solution. However, this method appears to be very costly, as it requires multiple runs for every k.

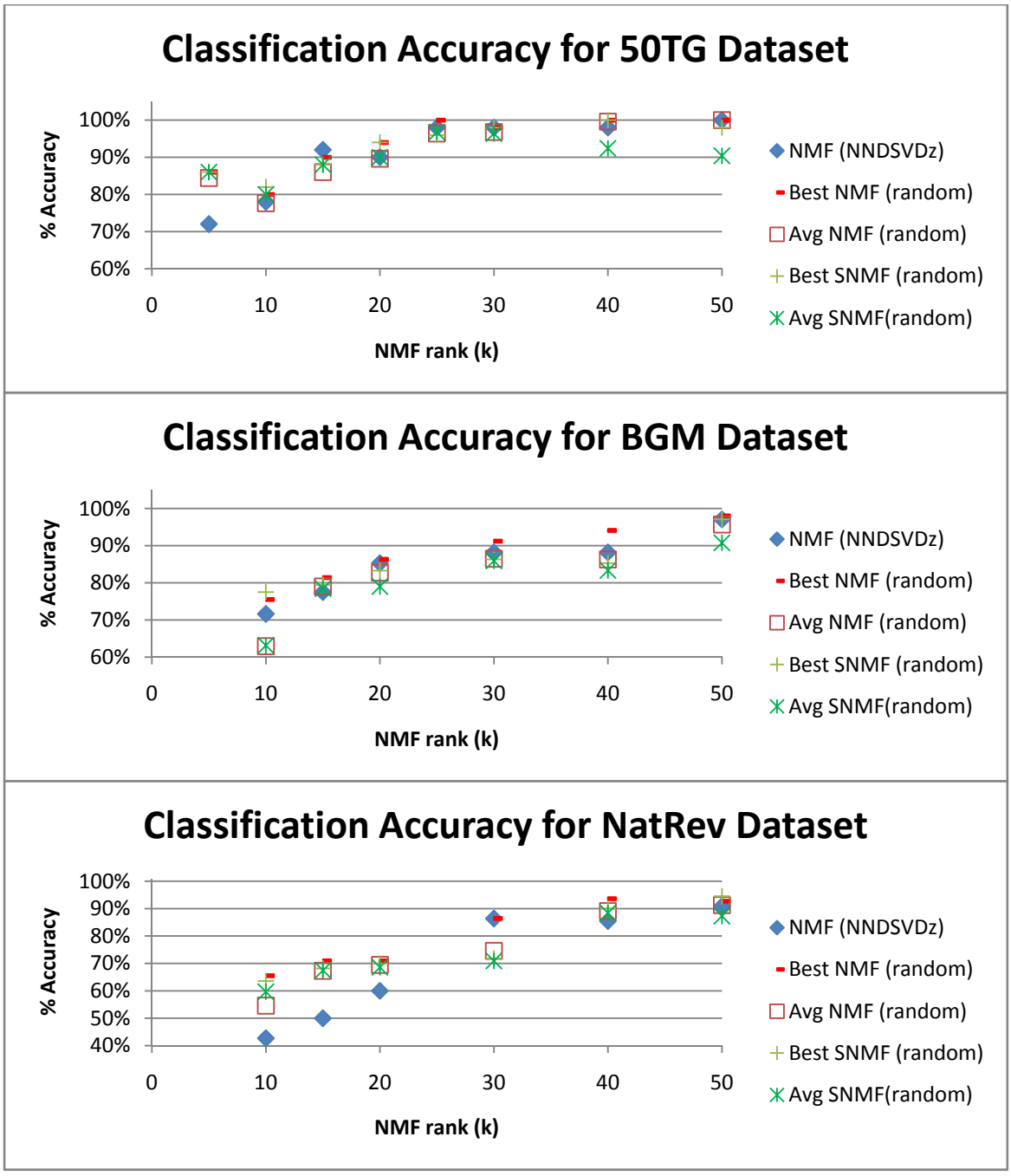


Figure 10. Comparison of Classification Accuracy with sparse NMF algorithm

Classification accuracy using our NMF algorithm and Kim and Park's sparse NMF algorithm is shown for datasets 1, 2, and 3. For results using random initialization, the best and average results are shown.

Table 2.

Performance Comparison between SNMF and Default NMF

SNMF (Matlab)									
NMF Rank *k*	Number of Iterations			CPU Time (s)			Number of operations per iteration: $O(k^4(m+n))$		
	50TG	BGM	NatRev	50TG	BGM	NatRev	50TG	BGM	NatRev
10	237	202	281	145	251	335	8.8E+07	1.3E+08	1.3E+08
20	282	270	234	829	2,175	2,952	1.4E+09	2.0E+09	2.1E+09
30	217	451	249	3,408	13,533	7,452	7.1E+09	1.0E+10	1.1E+10
40	30	397	231	1,544	23,112	12,318	2.3E+10	3.2E+10	3.4E+10
50	26	249	165	1,763	16,848	17,837	5.5E+10	7.9E+10	8.2E+10
Default NMF (C++)									
NMF Rank *k*	Number of Iterations			CPU Time (s)			Number of operations per iteration: $O(kmn)$		
	50TG	BGM	NatRev	50TG	BGM	NatRev	50TG	BGM	NatRev
10	86	130	92	3.57	14.4	8.2	4.4E+06	1.3E+07	1.4E+07
20	114	133	106	11.55	28.96	15.88	8.8E+06	2.6E+07	2.9E+07
30	162	154	119	28.34	46.57	23.91	1.3E+07	3.9E+07	4.3E+07
40	166	165	147	36.27	70.4	40.59	1.8E+07	5.1E+07	5.7E+07
50	634	171	180	197.96	94.7	61.33	2.2E+07	6.4E+07	7.2E+07

	50TG	BGM	NatRev
Number of terms (m)	8,750	12,590	13,038
Number of gene docs (n)	50	102	110

Chagoyen $et\ al.$ [40] applied the cophenetic correlation method to the 50TG dataset for k ranging from 2 to 16 with 100 independent runs for each k, and obtained k=7 as the rank that gives the stable solution. However, our experiments indicated that a rank of 7 does not necessary give the highest accuracy. The classification accuracy of k = 5, 7, 10, and 15 is 72%, 72%, 78%, and 92%, respectively.

The cophenetic correlation method hasn't been implemented in FAUN yet. Currently, FAUN uses a default NMF rank (k) of 10, 15, and 20 for low, medium, and high resolution of the models. A low resolution NMF model for the 50TG dataset, for example, consists of five features describing a specific concept about *cancer*, two features about *alzheimer*, two features about *cancer* and *development*, and one very general feature describing *cancer*, *development* and *alzheimer.* Increasing NMF rank (k), for example to 20, enables FAUN to isolate features that specifically covers *development*.

Initialization effect

We investigated the effect on classification accuracy using different initialization methods described in Section 4.2, namely, initializations based on randomization and non-randomization methods. For random initialization, we performed 5 runs and showed the average of the runs in Figure 11. For non-randomization methods, we used several types of NNDSVD that preserve zero elements of the singular vectors differently. NNDSVDz keeps zero elements at the original positions, while NNDSVDa, NNDSVDe, and NNDSVDme replace the zero elements with the average of all elements, ϵ, or $\epsilon_{machine}$, respectively.

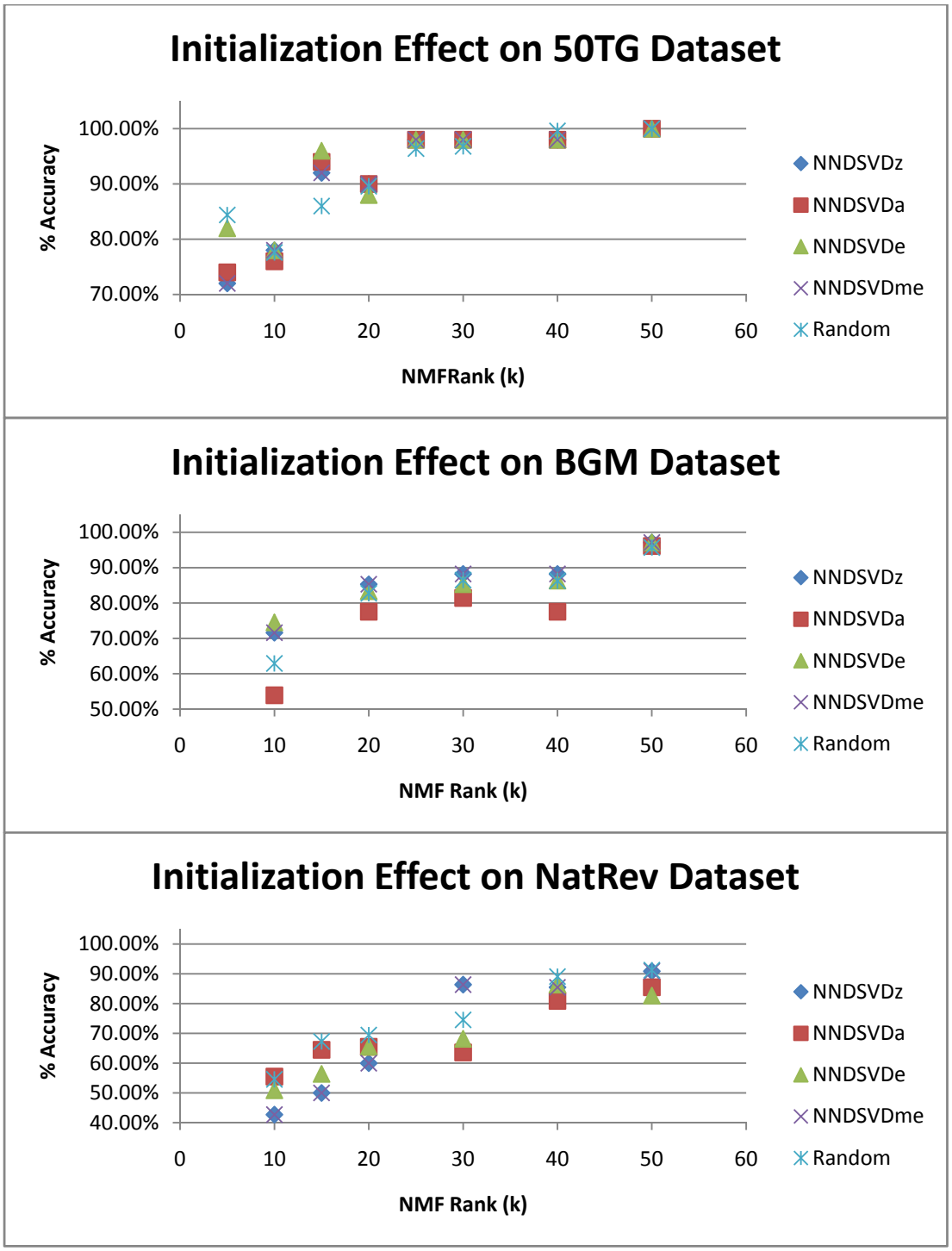


Figure 11. Initialization Effect

Effect of using different methods of initialization on classification accuracy on all three datasets.

In general, all initializations show very similar classification accuracy trends except for a few cases (Figure 11). NNDSVDz and NNDSVDme initializations significantly outperformed other initializations on the NatRev dataset with rank k=30. However, they underperformed on the same dataset with lower ranks. NNDSVDa noticeably underperformed on the BGM dataset.

We also investigated the initialization effects on feature terms and genes for each feature from different NMF models. Features produced from two different NMF models are compared by computing the *mutual information* $I(F,G)$ of each pair of features (F,G) where F is a feature from the first NMF model and G is a feature from the second model. Then, for every feature F, the algorithm extracts all features G such that $I(F,G) \geq \tau$, where $\tau > 0$ is a given threshold. Such features are considered to be strongly related. In addition, the underlying script displays the sets $T(F), T(G)$ of the top terms of features F and G, respectively, as well as $T(F) \setminus T(G)$ and $T(G) \setminus T(F)$. The mutual information $I(F,G)$ is computed by treating F and G as random variables such that $F(t) = 1$ if a term t is a top term of F, and 0 otherwise. Then:

$$I(F,G) = \sum_{t\in\{0,1\}} \sum_{u\in\{0,1\}} p(F=t, G=u) \log_2 \left(\frac{p(F=t, G=u)}{p(F=t)p(G=u)} \right).$$

On the NatRev dataset, 72% to 90% of the features from the NNDSVDz model are captured in the NNDSVDa model. Those features have at least 7 identical top terms, and are associated with the same genes for rank k=10, 15, and 20. For higher ranks, a few of those features in the NNDSVDa models are associated with significantly fewer numbers of genes.

Since the multiplicative update rules in NMF (deployed in FAUN) only guarantee (at best) a local minimum for the objective function, it is usually recommended to perform a simulation multiple times and select the best factorization for analysis [73, 74]. FAUN classification performance using random initialization is comparable to that when using NNDSVD methods. FAUN currently is using NNDSVDz as the default initialization method simply because the fixed starting point implies more predictability, and allows FAUN users to easily map features in different resolutions (ranks) of NMF models.

Stopping criteria effect

We restricted the number of maximum iterations to 1000 and 2000 and stopped the iteration if the consecutive values of W and H are closer than 0.01 and 0.001, respectively, in Frobenius norm. Increasing the maximum number of iterations beyond 1000 with the tolerance 0.01 does not appear to increase the classification accuracy.

Smoothing effect

Smoothing could be used to reduce noise, which might increase the quality of feature or make features more localized. In the case of smoothing the H matrix, this additional constraint filters out genes that are weakly associated with the features. In some cases, it increases classification accuracy up to 5.5%, shown for the 50TG dataset in Figure 12 with rank k=5, the BGM dataset with rank k=10, 15, 20, 30, and the NatRev dataset with rank k=10, 15, 20, 50. In a few cases, smoothing on H matrix factors decreases the accuracy up to 5.8% shown on the BGM dataset with rank k=40 and NatRev dataset with rank k=30, 40.

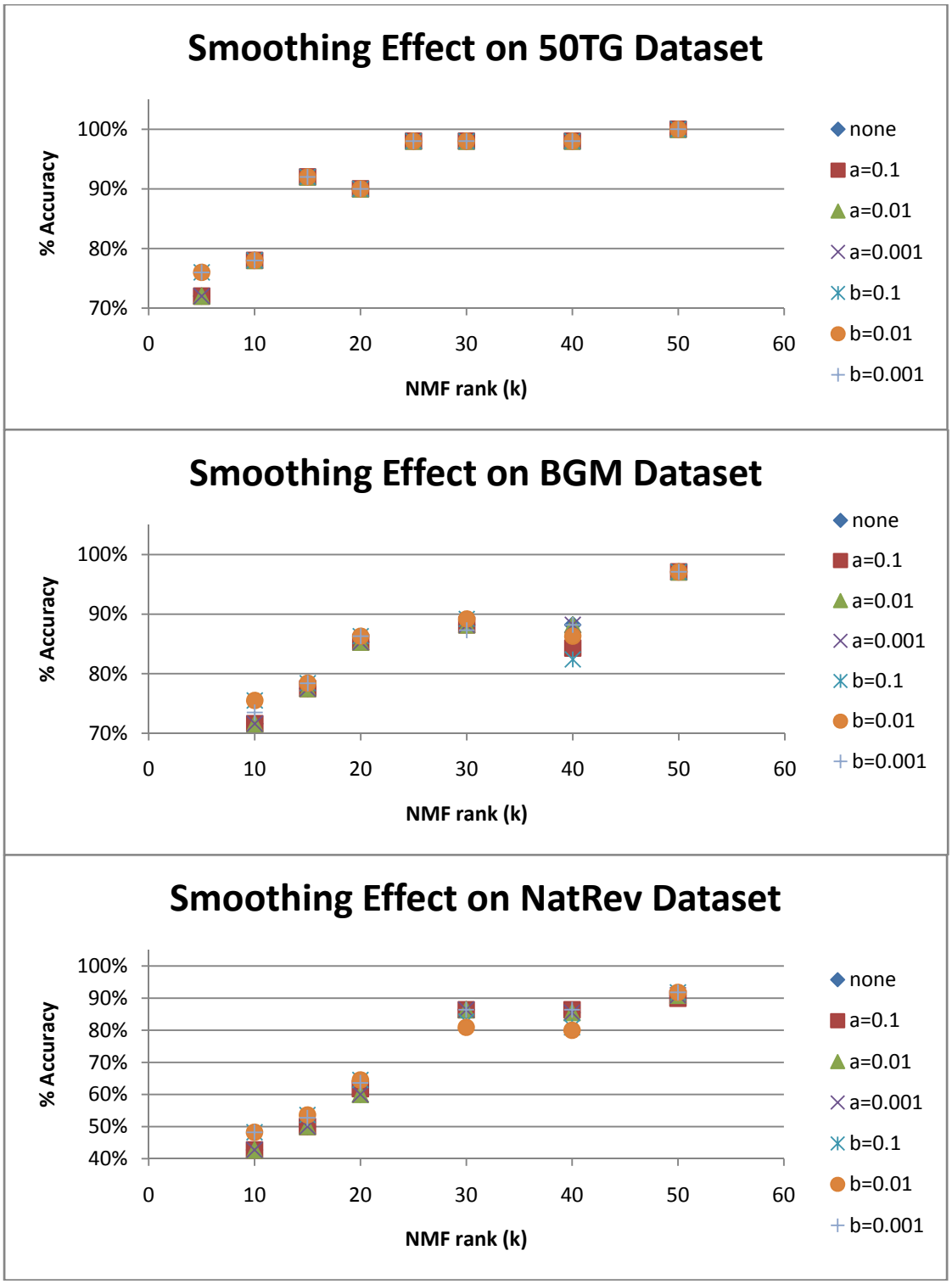


Figure 12. Smoothing Effect on Classification Accuracy

Control parameters for smoothing of 0.1, 0.01, and 0.001 are indicated with alpha (a) and beta (b) for W and H, respectively.

The smoothing constraint on W matrices observed with NNDSVDz (for all 3 datasets) has little or no effect on the accuracy, the feature terms and genes. Control parameters for smoothing are indicated with $\alpha(a)$ and $\beta(b)$ for W and H matrices respectively. This smoothness constraint is applied to either the W or H matrix (not both).

Sparsity effect

Sparsity constraints on the W or H matrix factors observed with NNDSVDz, in general, reduce the classification accuracy (Figure 13). Applying the sparsity constraint can reduce the accuracy up to 51%. This reduction, in most cases, is due to early termination of the NMF iterations (a symptom of over constraining the objective function). In a few cases, applying sparsity constraint increases the accuracy, for example on the NatRev dataset for low rank ($k = 10,\ 20$), the accuracy increases by 4 to 22%. Sparsity parameters are indicated with Ws and Hs for W and H matrices respectively.

Sparsity and smoothing effect on CPU time

Smoothing with the control parameter, alpha or beta, set to 0.1 results in a longer run time (Figure 14). In cases where early termination does not occur, time to convergence is considerably longer when applying sparsity constraints Figure 15. The associated errors are shown in Figure 16. Error is the distance from A to WH in Frobenius norm, $\|A - WH\|_F$.

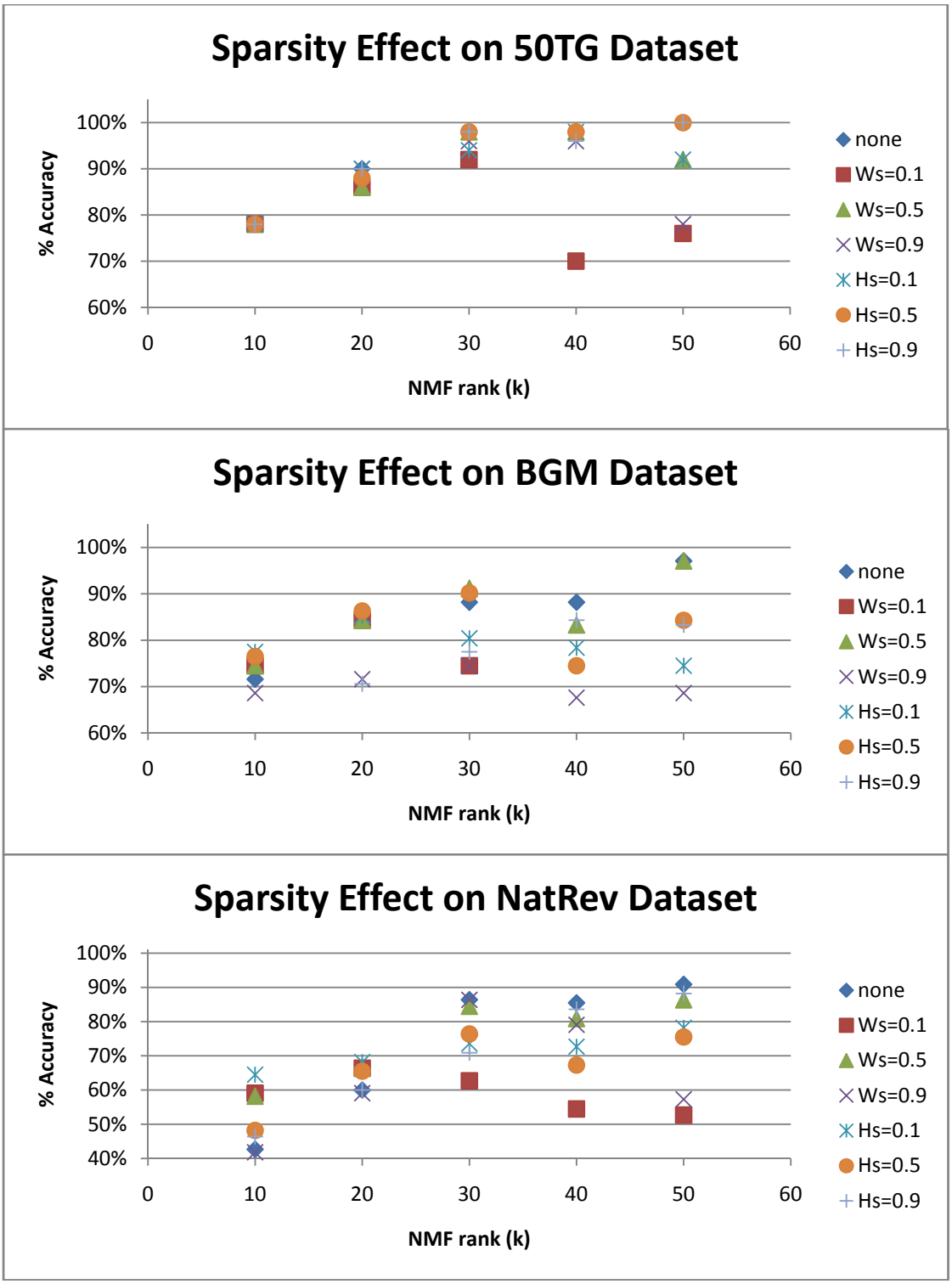


Figure 13. Sparsity Effect on Classification Accuracy

Control parameter for sparsity, for W (alpha) and H (beta), of 0.001 is used. Sparsity parameters of 0.1, 0.5, and 0.9 are indicated with Ws and Hs for W and H, respectively.

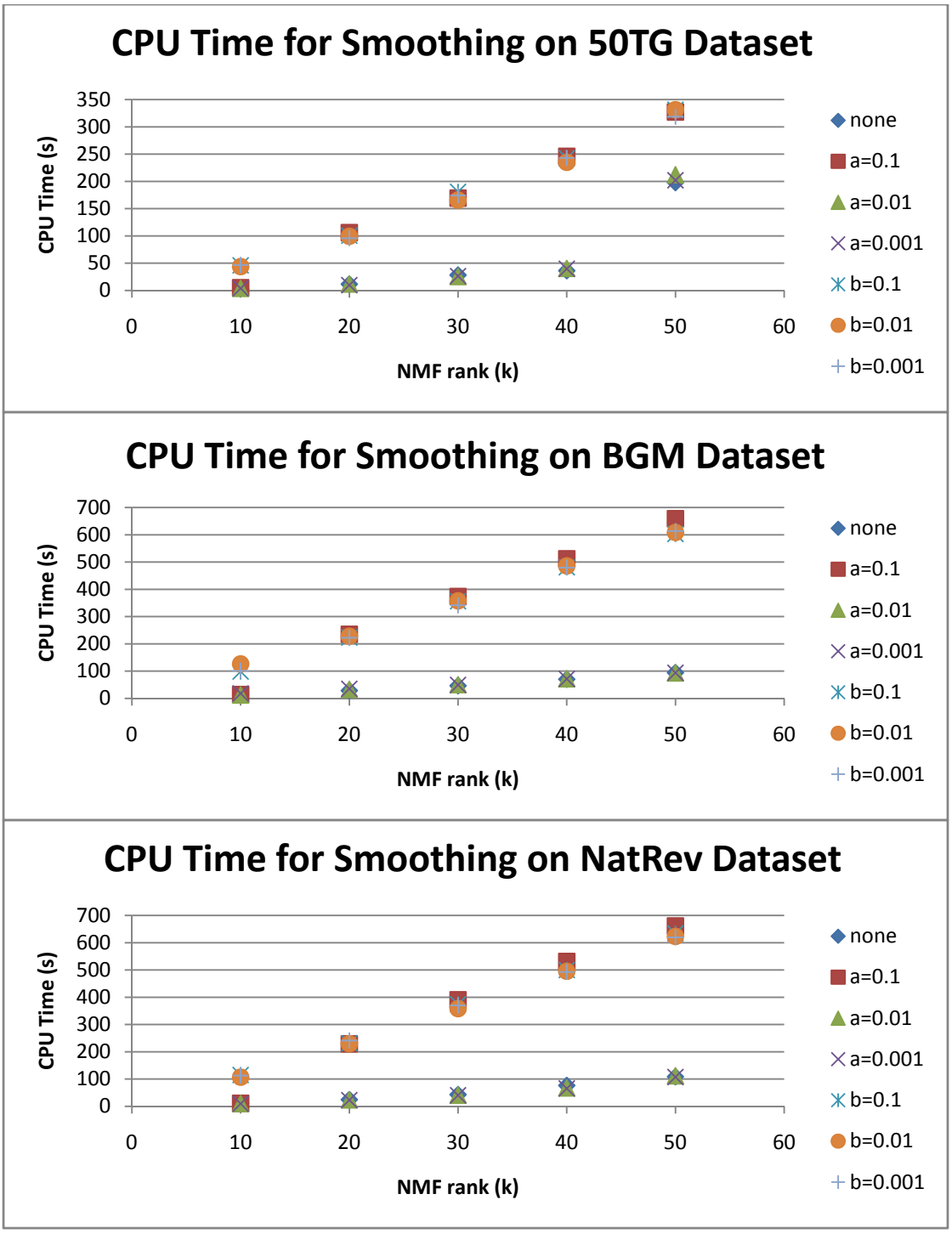


Figure 14. Smoothing Effect on CPU Time

Effect of using smoothing constraint on CPU Time on the 50TG, BGM, and NatRev datasets with NNDSVDz initialization.

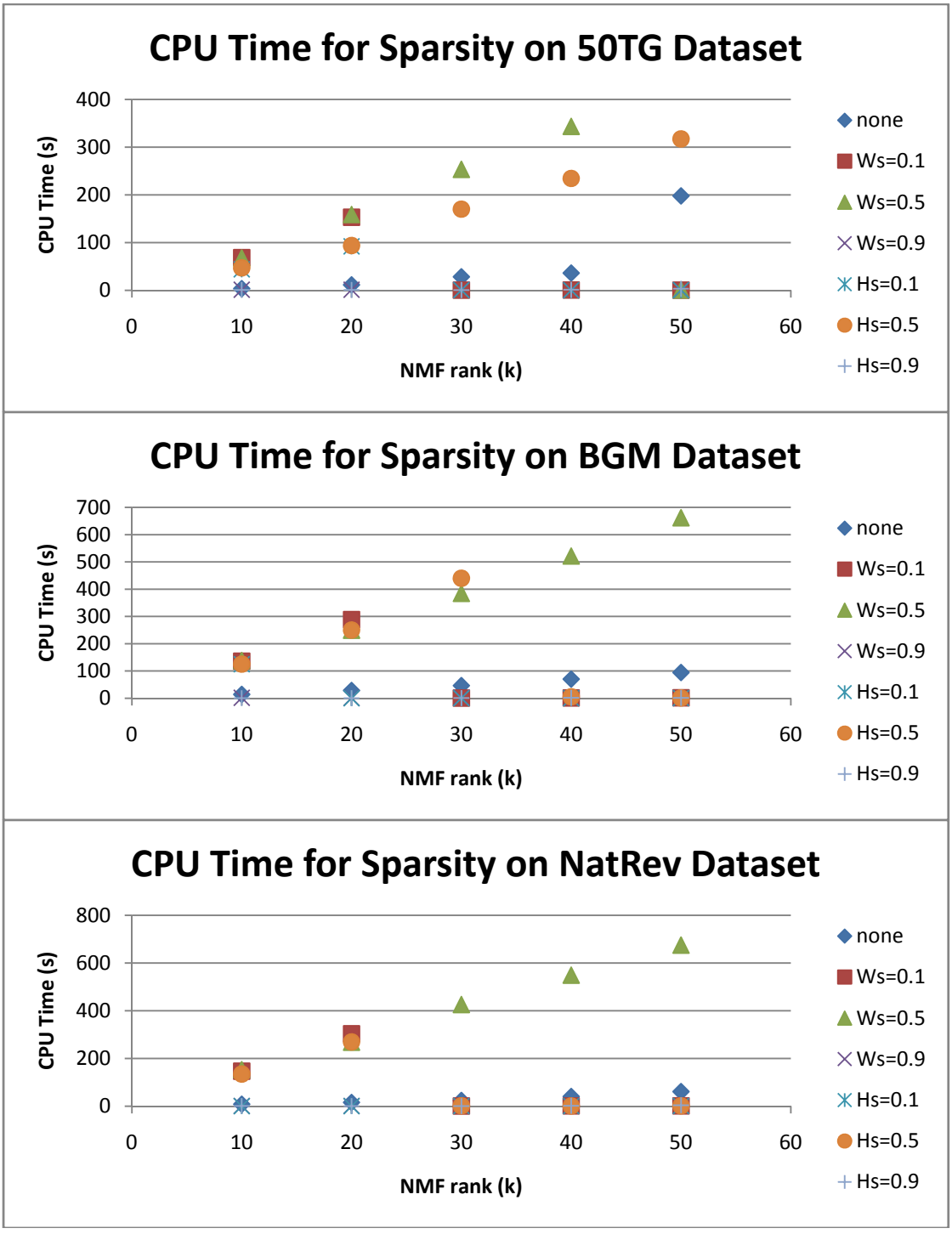


Figure 15. Sparsity Effect on CPU Time

Effect of using sparsity constraint on CPU Time on the 50TG, BGM, and NatRev datasets with NNDSVDz initialization.

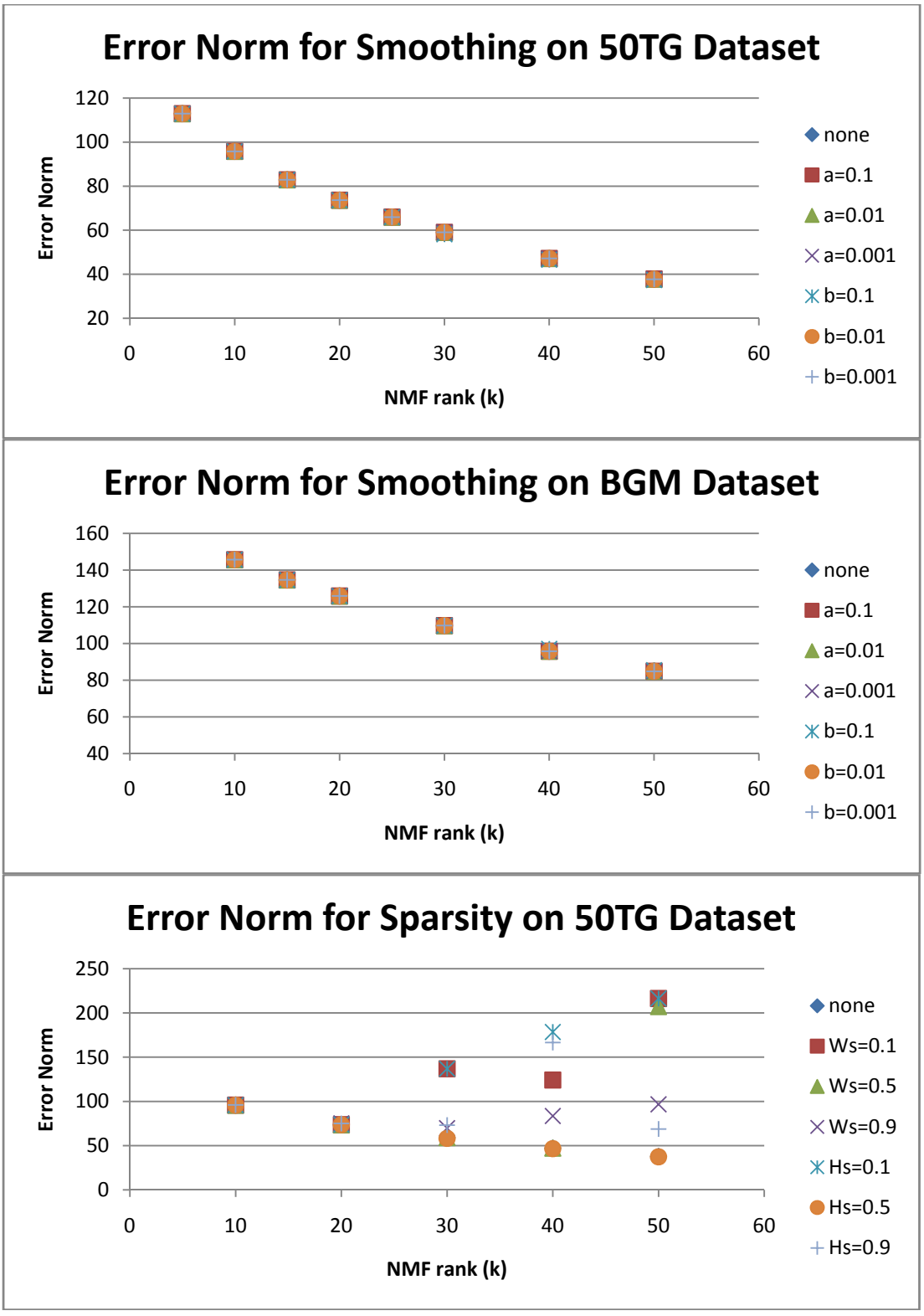


Figure 16. Error Estimation

Summary of the effects of NMF constraints

In general, one can increase rank k to increase the specificity of the features, impose smoothing constraints to reduce noise, or apply sparsity constraints to enhance the visibility of the features. For our test datasets, applying smoothing constraints does not have a significant influence on the classification accuracy. On the other hand, applying sparsity constraints, in general, reduces the accuracy in all but a few cases. Using different NMF initialization methods produces similar accuracy trends.

Discussion

Nonnegative matrix factorization has gained high popularity in many areas in science, engineering, and medicine [45]. It has also been attractive in the Bioinformatics field owing to its potential to provide new insights about the complex relationships in large data sets. However, an intuitive user-interface tool that allows the biologist to exploit the use of nonnegative matrix factorization on biomedical literature is still not available. In this study, we develop a Web-based bioinformatics software environment called FAUN to facilitate both the discovery and classification of functional relationships among genes.

Unlike most gene classification tools, FAUN not only identifies functionally related genes, but also provides a utility to explore why certain subsets of genes are related. In addition, FAUN is not based on co-occurrence methods which would only allow extracting explicit relationships. Genes identified in experiments oftentimes have not been studied together or published in the same article. Thus, the ability to extract the implicit relationships is crucial in the knowledge discovery process.

With recent advances in genomic technologies, it is possible to rapidly identify groups of genes that are coordinately regulated in particular experimental conditions. However, it remains challenging to derive the functional relationships between the genes. The process involves manually extracting and assembling gene information from literature and various biological databases. Given a list of the genes, FAUN assists researchers to narrow down relevant literature, provide the essence of the document collection, and present it in an intuitive, biologically user-friendly form.

Chapter 5.

Summary and Future Work

Given a list of genes, FAUN allows researchers to identify what genes are related and classify them functionally with promising accuracy. We have done some analysis on the effect of rank, initialization methods, stopping criteria, smoothing and sparsity constraints on the NMF model. In addition, we are also comparing our NMF algorithm with the Kim, Park, and Drake algorithm [42]. Interestingly, the comparison results are very similar. 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.

Enhancing features such as dragging and selecting multiple cells on the gene-to-gene correlation matrix are being implemented. These features would help researchers in analyzing the gene dataset. Another improvement that might be of interest is a gene query system which would make it possible to scan all features at once. This would be particularly useful for a large gene collection.

BIBLIOGRAPHY

Bibliography

1. [http://www.nlm.nih.gov/bsd/revup/revup_pub.html#med_update]
2. Cohen AM, Hersh WR: **A survey of current work in biomedical text mining**. *Brief Bioinform* 2005, **6**(1):57-71.
3. Jensen LJ, Saric J, Bork P: **Literature mining for the biologist: from information retrieval to biological discovery**. *Nat Rev Genet* 2006, **7**(2):119-129.
4. Krallinger M, Valencia A: **Text-mining and information-retrieval services for molecular biology**. *Genome Biol* 2005, **6**(7):224.
5. Ananiadou S, Kell DB, Tsujii J: **Text mining and its potential applications in systems biology**. *Trends Biotechnol* 2006, **24**(12):571-579.
6. Cohen KB, Hunter L: **Getting started in text mining**. *PLoS Comput Biol* 2008, **4**(1):e20.
7. Baeza-Yates R, Ribeiro-Neto B. In: *Modern Information Retrieval.* New York: ACM Press; 1999.
8. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT *et al*: **Gene ontology: tool for the unification of biology. The Gene Ontology Consortium**. *Nat Genet* 2000, **25**(1):25-29.
9. Golbeck J: **The National Cancer Institute's thesaurus and ontology**. *J Web Semantics* 2003, **1**:75-80.

10. Kanehisa M, Goto S: **KEGG: kyoto encyclopedia of genes and genomes**. *Nucleic Acids Res* 2000, **28**(1):27-30.

11. Doms A, Schroeder M: **GoPubMed: exploring PubMed with the Gene Ontology**. *Nucleic Acids Res* 2005, **33**(Web Server issue):W783-786.

12. Masys DR, Welsh JB, Lynn Fink J, Gribskov M, Klacansky I, Corbeil J: **Use of keyword hierarchies to interpret gene expression patterns**. *Bioinformatics* 2001, **17**(4):319-326.

13. Hosack DA, Dennis G, Jr., Sherman BT, Lane HC, Lempicki RA: **Identifying biological themes within lists of genes with EASE**. *Genome Biol* 2003, **4**(10):R70.

14. Khatri P, Draghici S: **Ontological analysis of gene expression data: current tools, limitations, and open problems**. *Bioinformatics* 2005, **21**(18):3587-3595.

15. Kostoff RN, Block JA, Stump JA, Pfeil KM: **Information content in Medline record fields**. *Int J Med Inform* 2004, **73**(6):515-527.

16. Funk ME, Reid CA: **Indexing consistency in MEDLINE**. *Bull Med Libr Assoc* 1983, **71**:176-183.

17. Shatkay H, Feldman R: **Mining the biomedical literature in the genomic era: an overview**. *J Comput Biol* 2003, **10**(6):821-855.

18. Alako BT, Veldhoven A, van Baal S, Jelier R, Verhoeven S, Rullmann T, Polman J, Jenster G: **CoPub Mapper: mining MEDLINE based on search term co-publication**. *BMC Bioinformatics* 2005, **6**:51.

19. Jenssen TK, Laegreid A, Komorowski J, Hovig E: **A literature network of human genes for high-throughput analysis of gene expression**. *Nat Genet* 2001, **28**(1):21-28.

20. Hanisch D, Fundel K, Mevissen HT, Zimmer R, Fluck J: **ProMiner: rule-based protein and gene entity recognition**. *BMC Bioinformatics* 2005, **6 Suppl 1**:S14.

21. Gaudan S, Kirsch H, Rebholz-Schuhmann D: **Resolving abbreviations to their senses in Medline**. *Bioinformatics* 2005, **21**(18):3658-3664.

22. Schijvenaars BJ, Mons B, Weeber M, Schuemie MJ, van Mulligen EM, Wain HM, Kors JA: **Thesaurus-based disambiguation of gene symbols**. *BMC Bioinformatics* 2005, **6**:149.

23. Swanson DR: **Fish oil, raynaud's syndrome, and undiscovered public knowledge**. *Perspect Biol Med* 1986, **30**:7-18.

24. Becker KG, Hosack DA, Dennis G, Jr., Lempicki RA, Bright TJ, Cheadle C, Engel J: **PubMatrix: a tool for multiplex literature mining**. *BMC Bioinformatics* 2003, **4**:61.

25. Homayouni R, Heinrich K, Wei L, Berry MW: **Gene clustering by latent semantic indexing of MEDLINE abstracts**. *Bioinformatics* 2005, **21**(1):104-115.

26. Che H, Sharp B: **Content-rich biological network constructed by mining PubMed abstracts**. *BMC Bioinformatics* 2004, **5**:147.

27. Muller HM, Kenny EE, Sternberg PW: **Textpresso: an ontology-based information retrieval and extraction system for biological literature**. *PLoS Biol* 2004, **2**(11):e309.

28. Donaldson I, Martin J, de Bruijn B, Wolting C, Lay V, Tuekam B, Zhang S, Baskin B, Bader GD, 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.

29. Heinrich KE: **Finding functional gene relationships using the Semantic Gene Organizer (SGO)**. M.S. thesis, Department of Computer Science, Knoxville, TN, USA: University of Tennessee; 2004.

30. Lee DD, Seung HS: **Learning the parts of objects by non-negative matrix factorization**. *Nature* 1999, **401**(6755):788-791.

31. Buciu I, Pitas I: **Application of non-negative and local non negative matrix factorization to facial expression recognition**. In: *Proceedings of the 17th International Conference on Pattern Recognition.* Cambridge, United Kingdom; 2004.

32. Jung I, Lee J, Lee SY, Kim D: **Application of nonnegative matrix factorization to improve profile-profile alignment features for fold recognition and remote homolog detection**. *BMC Bioinformatics* 2008, **9**:298.

33. Snyder DA, Zhang F, Robinette SL, Bruschweiler-Li L, Bruschweiler R: **Non-negative matrix factorization of two-dimensional NMR spectra: application to complex mixture analysis**. *J Chem Phys* 2008, **128**(5):052313.

34. Hoyer P: **Non-negative sparse coding**. In: *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing: 2002; Martigny, Switzerland*; 2002.

35. Behnke S: **Discovering hierarchical speech features using convolutional non-negative matrix factorization**. In: *Proceedings of the International Joint Conference on Neural Networks: 2003; Portland, Oregon*; 2003.

36. Cooper M, Foote J: **Summarizing video using nonnegative similarity matrix factorization**. In: *Proceedings of the IEEE Workshop on Multimedia Signal Processing: 2002; St. Thomas, U.S. Virgin Islands*; 2002.

37. Lu J, Xu B, Yang H: **Matrix dimensionality reduction for mining Web logs**. In: *Proceedings of the IEEE/WIC International Conference on Web Intelligence: 2003; Nova Scotia, Canada*; 2003.

38. Heinrich KE, Berry MW, Homayouni R: **Gene tree labeling using nonnegative matrix factorization on biomedical literature**. *Comput Intell Neurosci* 2008:276535.

39. Carmona-Saez P, Pascual-Marqui RD, Tirado F, Carazo JM, Pascual-Montano A: **Biclustering of gene expression data by Non-smooth Non-negative Matrix Factorization**. *BMC Bioinformatics* 2006, **7**:78.

40. Chagoyen M, Carmona-Saez P, Shatkay H, Carazo JM, Pascual-Montano A: **Discovering semantic features in the literature: a foundation for building functional associations**. *BMC Bioinformatics* 2006, **7**:41.

41. Gao Y, Church G: **Improving molecular cancer class discovery through sparse non-negative matrix factorization**. *Bioinformatics* 2005, **21**(21):3970-3975.

42. Kim H, Park H, Drake BL: **Extracting unrecognized gene relationships from the biomedical literature via matrix factorizations**. *BMC Bioinformatics* 2007, **8 Suppl 9**:S6.

43. Pascual-Montano A, Carmona-Saez P, Chagoyen M, Tirado F, Carazo JM, Pascual-Marqui RD: **bioNMF: a versatile tool for non-negative matrix factorization in biology**. *BMC Bioinformatics* 2006, **7**:366.

44. Pehkonen P, Wong G, Toronen P: **Theme discovery from gene lists for identification and viewing of multiple functional groups**. *BMC Bioinformatics* 2005, **6**:162.

45. Berry MW, Browne, M., Langville, A.N., Paul Pauca, V., and Plemmons, R.J.: **Algorithms and applications for approximate nonnegative matrix factorization**. *Computational Statistics & Data Analysis* 2006, **52**:155-173.

46. Kim H, Park H: **Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis**. *Bioinformatics* 2007, **23**(12):1495-1502.

47. **PHP/SWF Charts** [http://www.maani.us/charts4/index.php]

48. **Entrez Gene** [http://www.ncbi.nlm.nih.gov/sites/entrez?db=gene]

49. Giles JT, Wo, L., and Berry, M.W.: **GTP (General Text Parser) software for text mining**. *Statistical Data Mining and Knowledge Discovery Boca Raton, FL: CRC Press;* 2003.

50. Berry MW, Browne, M.: **Email surveillance using non-negative matrix factorization**. *Computational & Mathematical Organization Theory* 2005, **11**:249-264.

51. 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.

52. Lee DD, Seung HS: **Algorithms for nonnegative matrix factorization**. *Advances in Neural Information Processing Systems* 2001, **13**:556-562.

53. Pauca V, Shahnaz F, Berry M, Plemmons R: **Text mining using nonnegative matrix factorizations**. In: *Proceedings of the Fourth SIAM International Conference on Data Mining: 2004; Lake Buena Vista, FL*: SIAM; 2004.

54. Piper J, Pauca VP, Plemmons RJ, Giffin M: **Object characterization from spectral data using nonnegative factorization and information theory**. In: *Proceedings of the AMOS Technical Conference: 2004; Maui, Hawaii*; 2004.

55. Hoyer P: **Non-negative matrix factorization with sparseness constraints**. *J Mach Learning Res* 2004, **5**:1457-1469.

56. Pascual-Montano A, Carazo JM, Kochi K, Lehmann D, Pascual-Marqui RD: **Nonsmooth nonnegative matrix factorization (nsNMF)**. *IEEE Trans Pattern Anal Mach Intell* 2006, **28**(3):403-415.

57. Shahnaz F, Berry, M.W., Pau Pauca, V., and Plemmons, R.J.: **Document clustering using nonnegative matrix factorization**. *Information processing and management* 2006, **42**:373-386.

58. Benthem MHv, Keenan MR: **Fast algorithm for the solution of large-scale non-negativity-constrained least squares problems.** *J Chemometrics* 2004, **18**:441-450.

59. Tjioe E, Berry MW, Homayouni R: **Using a Literature-based NMF Model for Discovering Gene Functional Relationships**. In: *Proceedings of the 2008 IEEE International Bioinformatics and Biomedicine Conference on Data Mining in Functional Genomics Workshop.* Philadelphia, PA: IEEE Computer Society; 2008: 185-192.

60. Heinrich KE: **Automated gene classification using nonnegative matrix factorization on biomedical literature**. Ph.D. dissertation, Department of Computer Science, Knoxville, TN, USA: University of Tennessee; 2007.

61. Shen J, Kelleher RJ, 3rd: **The presenilin hypothesis of Alzheimer's disease: evidence for a loss-of-function pathogenic mechanism**. *Proc Natl Acad Sci U S A* 2007, **104**(2):403-409.

62. Deane R, Sagare A, Hamm K, Parisi M, Lane S, Finn MB, Holtzman DM, Zlokovic BV: **apoE isoform-specific disruption of amyloid beta peptide clearance from mouse brain**. *J Clin Invest* 2008, **118**(12):4002-4013.

63. Duyckaerts C, Potier MC, Delatour B: **Alzheimer disease models and human neuropathology: similarities and differences**. *Acta Neuropathol* 2008, **115**(1):5-38.

64. Borate BR: **Comparative analysis of thresholding algorithms for microarray-derived gene correlation matrices**. Master thesis, Department of Life Sciences, Knoxville, TN, USA: University of Tennessee; 2008.

65. Chaudhury AR, Gerecke KM, Wyss JM, Morgan DG, Gordon MN, Carroll SL: **Neuregulin-1 and erbB4 immunoreactivity is associated with neuritic**

plaques in Alzheimer disease brain and in a transgenic model of Alzheimer disease. *J Neuropathol Exp Neurol* 2003, **62**(1):42-54.

66. Burkart MF, Wren JD, Herschkowitz JI, Perou CM, Garner HR: **Clustering microarray-derived gene lists through implicit literature relationships**. *Bioinformatics* 2007, **23**(15):1995-2003.

67. Abrahams BS, Geschwind DH: **Advances in autism genetics: on the threshold of a new neurobiology**. *Nat Rev Genet* 2008, **9**(5):341-355.

68. Frayling TM: **Genome-wide association studies provide new insights into type 2 diabetes aetiology**. *Nat Rev Genet* 2007, **8**(9):657-662.

69. Robinson GW: **Cooperation of signalling pathways in embryonic mammary gland development**. *Nat Rev Genet* 2007, **8**(12):963-972.

70. Scheper GC, van der Knaap MS, Proud CG: **Translation matters: protein synthesis defects in inherited disease**. *Nat Rev Genet* 2007, **8**(9):711-723.

71. Wang W: **Emergence of a DNA-damage response network consisting of Fanconi anaemia and BRCA proteins**. *Nat Rev Genet* 2007, **8**(10):735-748.

72. Vasan RS, Demissie S, Kimura M, Cupples LA, Rifai N, While C, Wang TJ, Gardner JP, Cao X, Benjamin EJ *et al*: **Association of leukocyte telomere length with circulating biomarkers of the Renin-Angiotensin-Aldosterone system: the Framingham Heart study**. *Circulation* 2008, **117**:1138-1144.

73. Brunet JP, Tamayo P, Golub TR, Mesirov JP: **Metagenes and molecular pattern discovery using matrix factorization**. *Proc Natl Acad Sci U S A* 2004, **101**(12):4164-4169.

74. Kim PM, Tidor B: **Subsystem identification through dimensionality reduction of large-scale gene expression data**. *Genome Res* 2003, **13**(7):1706-1718.

APPENDICES

Appendix A

Table A. 1

Sample collection with dictionary terms displayed in bold

Document	Text
d1	Work-related ***stress*** can be considered a factor contributing to ***anxiety.***
d2	***Liver cancer*** is most commonly associated with ***alcoholism*** and ***cirrhosis.*** It is well-known that ***alcoholism*** can cause ***cirrhosis*** and increase the risk of ***kidney failure***.
d3	***Bone marrow*** transplants are often needed for patients with ***leukemia*** and other types of ***cancer*** that ***damage bone marrow***. Exposure to toxic chemicals is a risk factor for leukemia.
d4	Different types of ***blood cells*** exist in ***bone marrow.*** Bone marrow procedures can detect ***tuberculosis.***
d5	Abnormal ***stress*** or ***pressure*** can cause an ***anxiety attack***. Continued ***stress*** can elevate ***blood pressure.***
d6	***Alcoholism*** can cause high ***blood pressure*** (hypertension) and increase the risk of ***birth defects*** and ***kidney failure***.
d7	The presence of ***speech defects*** in ***children*** is a sign of ***autism.*** As of yet, there is no consensus on what causes ***autism.***
d8	***Alcoholism,*** often triggered at an early age by factors such as environment and genetic predisposition, can lead to ***cirrhosis.*** Cirrhosis is the ***scarring*** of the ***liver.***
d9	***Autism*** affects approximately 0.5% of children in the US. The link between ***alcoholism*** and ***birth defects*** is well-known; researchers are currently studying the link between ***alcoholism*** and ***autism.***

Table A. 2

Term-document matrix for the sample collection in Table A.1

	d1	d2	d3	d4	d5	d6	d7	d8	d9
Alcoholism	—	0.4338	—	—	—	0.2737	—	0.2737	0.4338
Anxiety	0.4745	—	—	—	0.4745	—	—	—	—
Attack	—	—	—	—	0.6931	—	—	—	—
Autism	—	—	—	—	—	—	0.752	—	0.752
Birth	—	—	—	—	—	0.4745	—	—	0.4745
Blood	—	—	—	0.3466	0.3466	0.3466	—	—	—
Bone	—	—	0.752	0.752	—	—	—	—	—
Cancer	—	0.4745	0.4745	—	—	—	—	—	—
Cells	—	—	—	0.6931	—	—	—	—	—
Children	—	—	—	—	—	—	0.4745	—	0.4745
Cirrhosis	—	0.752	—	—	—	—	—	0.752	—
Damage	—	—	0.6931	—	—	—	—	—	—
Defects	—	—	—	—	—	0.3466	0.3466	—	0.3466
Failure	—	0.4745	—	—	—	0.4745	—	—	—
Hypertension	—	—	—	—	—	0.6931	—	—	—
Kidney	—	0.4745	—	—	—	0.4745	—	—	—
Leukemia	—	—	1.0986	—	—	—	—	—	—
Liver	—	0.4745	—	—	—	—	—	0.4745	—
Marrow	—	—	0.752	0.752	—	—	—	—	—
Pressure	—	—	—	—	0.7804	0.4923	—	—	—
Scarring	—	—	—	—	—	—	—	0.6931	—
Speech	—	—	—	—	—	—	0.6931	—	—
Stress	0.4923	—	—	—	0.7804	—	—	—	—
Tuberculosis	—	—	—	0.6931	—	—	—	—	—

Table A. 3

Feature matrix W for the sample collection

	f1	f2	f3	f4
Alcoholism	0.0006	0.3503	—	—
Anxiety	—	—	0.4454	—
Attack	—	—	0.4913	—
Autism	—	0.003	—	0.8563
Birth	—	0.1111	0.0651	0.273
Blood	0.0917	0.0538	0.3143	—
Bone	0.522	—	0.0064	—
Cancer	0.1974	0.1906	—	—
Cells	0.1962	—	0.0188	—
Children	—	0.0019	—	0.5409
Cirrhosis	0.0015	0.5328	—	—
Damage	0.2846	—	—	—
Defects	—	0.0662	—	0.4161
Failure	0.0013	0.2988	—	—
Hypertension	—	0.1454	0.1106	—
Kidney	0.0013	0.2988	—	—
Leukemia	0.4513	—	—	—
Liver	0.0009	0.3366	—	—
Marrow	0.522	—	0.0064	—
Pressure	—	0.066	0.6376	—
Scarring	—	0.208	—	—
Speech	—	—	—	0.4238
Stress	—	—	0.6655	—
Tuberculosis	0.1962	—	0.0188	—

Table A. 4

Coefficient matrix H for the sample collection

	d1	d2	d3	d4	d5	d6	d7	d8	d9
f1	—	0.0409	1.6477	1.1382	0.0001	0.0007	—	—	—
f2	—	1.3183	—	—	0.0049	0.6955	0.0003	0.9728	0.2219
f3	0.3836	—	—	0.0681	1.1933	0.3327	—	—	—
f4	—	—	—	—	—	0.1532	0.9214	—	0.799

Table A. 5

Approximation to sample term-document matrix given in Table A. 2

	d1	d2	d3	d4	d5	d6	d7	d8	d9
Alcoholism	—	0.4618	0.001	0.0007	0.0017	0.2436	0.0001	0.3408	0.0777
Anxiety	0.1708	—	—	0.0303	0.5315	0.1482	—	—	—
Attack	0.1884	—	—	0.0334	0.5863	0.1635	—	—	—
Autism	—	0.004	—	—	—	0.1333	0.789	0.0029	0.6848
Birth	0.025	0.1464	—	0.0044	0.0783	0.1407	0.2516	0.108	0.2428
Blood	0.1206	0.0746	0.1511	0.1258	0.3754	0.142	—	0.0523	0.0119
Bone	0.0025	0.0214	0.8602	0.5946	0.0077	0.0025	—	—	—
Cancer	—	0.2593	0.3252	0.2247	0.001	0.1327	0.0001	0.1854	0.0423
Cells	0.0072	0.008	0.3233	0.2246	0.0224	0.0064	—	—	—
Children	—	0.0025	—	—	—	0.0842	0.4984	0.0019	0.4326
Cirrhosis	—	0.7025	0.0024	0.0017	0.0026	0.3705	0.0002	0.5183	0.1183
Damage	—	0.0116	0.4689	0.3239	—	0.0002	—	—	—
Defects	—	0.0873	—	—	0.0003	0.1098	0.3834	0.0644	0.3472
Failure	—	0.3939	0.0022	0.0015	0.0015	0.2078	0.0001	0.2906	0.0663
Hypertension	0.0424	0.1916	—	0.0075	0.1327	0.1379	—	0.1414	0.0323
Kidney	—	0.3939	0.0022	0.0015	0.0015	0.2078	0.0001	0.2906	0.0663
Leukemia	—	0.0185	0.7437	0.5137	—	0.0003	—	—	—
Liver	—	0.4437	0.0015	0.0011	0.0017	0.2341	0.0001	0.3274	0.0747
Marrow	0.0025	0.0214	0.8602	0.5946	0.0077	0.0025	—	—	—
Pressure	0.2445	0.087	—	0.0434	0.7612	0.258	—	0.0642	0.0147
Scarring	—	0.2742	—	—	0.001	0.1446	0.0001	0.2023	0.0462
Speech	—	—	—	—	—	0.0649	0.3905	—	0.3386
Stress	0.2553	—	—	0.0453	0.7942	0.2214	—	—	—
Tuberculosis	0.0072	0.008	0.3233	0.2246	0.0224	0.0064	—	—	—

Table A. 6

Top 5 weighted terms for each feature from the sample collection

f1	f2	f3	f4
Bone	Cirrhosis	Stress	Autism
Marrow	Alcoholism	Pressure	Children
Leukemia	Liver	Attack	Speech
Damage	Kidney	Anxiety	Defects
Cancer	Failure	Blood	Birth

Table A. 7

Rearranged term-document matrix for the sample collection

	d3	d4	d2	d6	d8	d1	d5	d7	d9
Bone	0.752	0.752	—	—	—	—	—	—	—
Cancer	0.4745	—	0.4745	—	—	—	—	—	—
Cells	—	0.6931	—	—	—	—	—	—	—
Damage	0.6931	—	—	—	—	—	—	—	—
Leukemia	1.0986	—	—	—	—	—	—	—	—
Marrow	0.752	0.752	—	—	—	—	—	—	—
Tuberculosis	—	0.6931	—	—	—	—	—	—	—
Alcoholism	—	—	0.4338	0.2737	0.2737	—	—	—	0.4338
Cirrhosis	—	—	0.752	—	0.752	—	—	—	—
Failure	—	—	0.4745	0.4745	—	—	—	—	0.4745
Hypertension	—	—	—	0.6931	—	—	—	—	—
Kidney	—	—	0.4745	0.4745	—	—	—	—	0.4745
Liver	—	—	0.4745	—	0.4745	—	—	—	—
Scarring	—	—	—	—	0.6931	—	—	—	—
Anxiety	—	—	—	—	—	0.4745	0.4745	—	—
Attack	—	—	—	—	—	—	0.6931	—	—
Blood	—	0.3466	—	0.3466	—	—	0.3466	—	—
Pressure	—	—	—	0.4923	—	—	0.7804	—	—
Stress	—	—	—	—	—	0.4923	0.7804	—	—
Autism	—	—	—	—	—	—	—	0.752	0.752
Birth	—	—	—	0.4745	—	—	—	—	0.4745
Children	—	—	—	—	—	—	—	0.4745	0.4745
Defects	—	—	—	0.3466	—	—	—	0.3466	0.3466
Speech	—	—	—	—	—	—	—	0.6931	—

Appendix B

Table B. 1

The 50 genes in the 50TG dataset

Category	Gene Symbol	Entrez Gene ID	Official Gene Name
Alzheimer	A2M	232345	alpha-2-macroglobulin
	APBA1	108119	amyloid beta (A4) precursor protein-binding, family A, member 1
	APBB1	11785	amyloid beta (A4) precursor protein-binding, family B, member 1
	APLP1	11803	amyloid beta (A4) precursor-like protein 1
	APLP2	11804	amyloid beta (A4) precursor-like protein 2
	APOE	11816	apolipoprotein E
	APP	11820	amyloid beta (A4) precursor protein
	LRP1	16971	low density lipoprotein receptor-related protein 1
	MAPT	17762	microtubule-associated protein tau
	PSEN1	19164	presenilin 1
	PSEN2	19165	presenilin 2
Alzheimer & Development	CDK5	12568	cyclin-dependent kinase 5
	CDK5R	12569	cyclin-dependent kinase 5, regulatory subunit (p35) 1
	CDK5R2	12570	cyclin-dependent kinase 5, regulatory subunit 2 (p39)
Cancer	ABL1	11350	v-abl Abelson murine leukemia oncogene 1
	BRCA1	12189	breast cancer 1
	BRCA2	12190	breast cancer 2
	DNMT1	13433	DNA methyltransferase (cytosine-5) 1
	EGFR	13649	epidermal growth factor receptor
	ERBB2	13866	v-erb-b2 erythroblastic leukemia viral oncogene homolog 2, neuro/glioblastoma derived oncogene homolog (avian)
	ETS1	24356	v-ets erythroblastosis virus E26 oncogene homolog 1 (avian)
	FOS	14281	FBJ osteosarcoma oncogene
	FYN	14360	Fyn proto-oncogene
	KIT	16590	kit oncogene
	MYC	17869	myelocytomatosis oncogene
	NRAS	18176	neuroblastoma ras oncogene
	SHC1	20416	src homology 2 domain-containing transforming protein C1
	SRC	20779	Rous sarcoma oncogene
	TRP53	22059	transformation related protein 53

Genes in *50TG* dataset (Table B.1 continued).

Category	Gene Symbol	Entrez Gene ID	Official Gene Name
Cancer & Development	DLL1	13388	delta-like 1 (Drosophila)
	GLI	14632	GLI-Kruppel family member GLI1
	GLI2	14633	GLI-Kruppel family member GLI2
	GLI3	14634	GLI-Kruppel family member GLI3
	JAG1	16449	jagged 1
	NOTCH1	18128	Notch gene homolog 1 (Drosophila)
	PAX2	18504	paired box gene 2
	PAX3	18505	paired box gene 3
	PTCH	19206	patched homolog 1
	ROBO1	19876	roundabout homolog 1 (Drosophila)
	SHH	20423	sonic hedgehog
	SMO	20596	smoothened homolog (Drosophila)
	TGFB1	21803	transforming growth factor, beta 1
	WNT1	22408	wingless-related MMTV integration site 1
	WNT2	22413	wingless-related MMTV integration site 2
	WNT3	22415	wingless-related MMTV integration site 3
Development	ATOH1	11921	atonal homolog 1 (Drosophila)
	DAB1	13131	disabled homolog 1 (Drosophila)
	LRP8	16975	low density lipoprotein receptor-related protein 8, apolipoprotein E receptor
	RELN	19699	reelin
	VLDLR	22359	very low density lipoprotein receptor

Table B. 2

The 102 genes in the BGM dataset

Category	Gene Symbol	Entrez Gene ID	Official Gene Name
Adhesion and diapedesis of lymphocytes	CD34	947	CD34 molecule
	ICAM1	3383	intercellular adhesion molecule 1 (CD54), human rhinovirus receptor
	ICAM2	3384	intercellular adhesion molecule 2
	IL8	3576	interleukin 8
	ITGA4	3676	integrin, alpha 4 (antigen CD49D, alpha 4 subunit of VLA-4 receptor)
	ITGAL	3683	integrin, alpha L (antigen CD11A (p180), lymphocyte function-associated antigen 1; alpha polypeptide)
	ITGB1	3688	integrin, beta 1 (fibronectin receptor, beta polypeptide, antigen CD29 includes MDF2, MSK12)
	ITGB2	3689	integrin, beta 2 (complement component 3 receptor 3 and 4 subunit)
	PECAM1	5175	platelet/endothelial cell adhesion molecule (CD31 antigen)
	SELL	6402	selectin L (lymphocyte adhesion molecule 1)
Caspase cascade in apoptosis	APAF1	317	apoptotic peptidase activating factor 1
	ARHGDIB	397	Rho GDP dissociation inhibitor (GDI) beta
	BIRC2	329	baculoviral IAP repeat-containing 2
	BIRC3	330	baculoviral IAP repeat-containing 3
	BIRC4	331	baculoviral IAP repeat-containing 4
	CASP1	834	caspase 1, apoptosis-related cysteine peptidase (interleukin 1, beta, convertase)
	CASP10	843	caspase 10, apoptosis-related cysteine peptidase
	CASP2	835	caspase 2, apoptosis-related cysteine peptidase (neural precursor cell expressed, developmentally down-regulated 2)
	CASP3	836	caspase 3, apoptosis-related cysteine peptidase
	CASP4	837	caspase 4, apoptosis-related cysteine peptidase
	CASP6	839	caspase 6, apoptosis-related cysteine peptidase
	CASP7	840	caspase 7, apoptosis-related cysteine peptidase
	CASP8	841	caspase 8, apoptosis-related cysteine peptidase
	CASP9	842	caspase 9, apoptosis-related cysteine peptidase
	DFFA	1676	DNA fragmentation factor, 45kDa, alpha polypeptide
	DFFB	1677	DNA fragmentation factor, 40kDa, beta polypeptide (caspase-activated DNase)
	GZMB	3002	granzyme B (granzyme 2, cytotoxic T-lymphocyte-associated serine esterase 1)
	LAMA1	284217	laminin, alpha 1
	LMNB1	4001	lamin B1
	LMNB2	84823	lamin B2
	PRF1	5551	perforin 1 (pore forming protein)

Genes in *BGM* dataset (Table B.2 continued).

Category	Gene Symbol	Entrez Gene ID	Official Gene Name
Chronic pancreatitis	CCK	885	cholecystokinin
	CEACAM1	634	carcinoembryonic antigen-related cell adhesion molecule 1 (biliary glycoprotein)
	GIP	2695	gastric inhibitory polypeptide
	LTF	4057	lactotransferrin
	PLA2G2A	5320	phospholipase A2, group IIA (platelets, synovial fluid)
	PPY	5539	pancreatic polypeptide
	SPINK1	6690	serine peptidase inhibitor, Kazal type 1
	SST	6750	somatostatin
Cornified cell envelope	CDSN	1041	corneodesmosin
	CNFN	84518	cornifelin
	SCEL	8796	sciellin
	SPRR2A	6700	small proline-rich protein 2A
	SPRR2B	6701	small proline-rich protein 2B
	TGM1	7051	transglutaminase 1 (K polypeptide epidermal type I, protein-glutamine-gamma-glutamyltransferase)
	TGM3	7053	transglutaminase 3 (E polypeptide, protein-glutamine-gamma-glutamyltransferase)
DNA helicase	ATRX	546	alpha thalassemia/mental retardation syndrome X-linked (RAD54 homolog, S. cerevisiae)
	BLM	641	Bloom syndrome
	BRIP1	83990	BRCA1 interacting protein C-terminal helicase 1
	CHD1	1105	chromodomain helicase DNA binding protein 1
	CHD2	1106	chromodomain helicase DNA binding protein 2
	CHD3	1107	chromodomain helicase DNA binding protein 3
	CHD4	1108	chromodomain helicase DNA binding protein 4
	DHX9	1660	DEAH (Asp-Glu-Ala-His) box polypeptide 9
	ERCC2	2068	excision repair cross-complementing rodent repair deficiency, complementation group 2 (xeroderma pigmentosum D)
	ERCC3	2071	excision repair cross-complementing rodent repair deficiency, complementation group 3 (xeroderma pigmentosum group B complementing)
	ERCC6	2074	excision repair cross-complementing rodent repair deficiency, complementation group 6
	G3BP	10146	GTPase activating protein (SH3 domain) binding protein 1
	RAD54B	25788	RAD54 homolog B (S. cerevisiae)
	RECQL	5965	RecQ protein-like (DNA helicase Q1-like)
	RECQL4	9401	RecQ protein-like 4
	RECQL5	9400	RecQ protein-like 5
	RUVBL1	8607	RuvB-like 1 (E. coli)
	RUVBL2	10856	RuvB-like 2 (E. coli)
	WRN	7486	Werner syndrome
	XRCC5	7520	X-ray repair complementing defective repair in Chinese hamster cells 5 (double-strand-break rejoining; Ku autoantigen, 80kDa)

Genes in *BGM* dataset (Table B.2 continued).

Category	Gene Symbol	Entrez Gene ID	Official Gene Name
Nephroblastoma	AVP	551	arginine vasopressin (neurophysin II, antidiuretic hormone, diabetes insipidus, neurohypophyseal)
	CTGF	1490	connective tissue growth factor
	MME	4311	membrane metallo-endopeptidase (neutral endopeptidase, enkephalinase)
	NOV	4856	nephroblastoma overexpressed gene
	NPPA	4878	natriuretic peptide precursor A
	PRCC	5546	papillary renal cell carcinoma (translocation-associated)
	REN	5972	renin
	SOX9	6662	SRY (sex determining region Y)-box 9 (campomelic dysplasia, autosomal sex-reversal)
	SYNPO	11346	synaptopodin
	WISP3	8838	WNT1 inducible signaling pathway protein 3
Retinitis pigmentosa	ABCA4	24	ATP-binding cassette, sub-family A (ABC1), member 4
	CHM	1121	choroideremia (Rab escort protein 1)
	CRB1	23418	crumbs homolog 1 (Drosophila)
	MYO7A	4647	myosin VIIA
	PDE6B	5158	phosphodiesterase 6B, cGMP-specific, rod, beta (congenital stationary night blindness 3, autosomal dominant)
	PRPF31	26121	PRP31 pre-mRNA processing factor 31 homolog (S. cerevisiae)
	RHO	6010	rhodopsin (opsin 2, rod pigment) (retinitis pigmentosa 4, autosomal dominant)
	RPGR	6103	retinitis pigmentosa GTPase regulator
Sonic hedgehog pathway	DYRK1B	9149	dual-specificity tyrosine-(Y)-phosphorylation regulated kinase 1B
	GLI2	2736	GLI-Kruppel family member GLI2
	GLI3	2737	GLI-Kruppel family member GLI3 (Greig cephalopolysyndactyly syndrome)
	GSK3B	2932	glycogen synthase kinase 3 beta
	PRKAR1A	5573	protein kinase, cAMP-dependent, regulatory, type I, alpha (tissue specific extinguisher 1)
	PTCH	5727	patched homolog 1 (Drosophila)
	SHH	6469	sonic hedgehog homolog (Drosophila)
	SUFU	51684	suppressor of fused homolog (Drosophila)
Telomere maintenance	MRE11A	4361	MRE11 meiotic recombination 11 homolog A (S. cerevisiae)
	RAD50	10111	RAD50 homolog (S. cerevisiae)
	RFC1	5981	replication factor C (activator 1) 1, 145kDa
	TEP1	7011	telomerase-associated protein 1
	TERF1	7013	telomeric repeat binding factor (NIMA-interacting) 1
	TERF2	7014	telomeric repeat binding factor 2
	TERT	7015	telomerase reverse transcriptase
	TINF2	26277	TERF1 (TRF1)-interacting nuclear factor 2
	TNKS	8658	tankyrase, TRF1-interacting ankyrin-related ADP-ribose polymerase
	TNKS1BP1	85456	tankyrase 1 binding protein 1, 182kDa

Table B. 3

The 110 genes in the NatRev dataset

Category	Gene Symbol	Entrez Gene ID	Official Gene Name
Autism	AHI1	54806	Abelson helper integration site 1
	AVPR1A	552	arginine vasopressin receptor 1A
	CACNA1C	775	calcium channel, voltage-dependent, L type, alpha 1C subunit
	CADPS2	93664	Ca2+-dependent activator protein for secretion 2
	CNTNAP2	26047	contactin associated protein-like 2
	DHCR7	1717	7-dehydrocholesterol reductase
	DISC1	27185	disrupted in schizophrenia 1
	EN2	2020	engrailed homolog 2
	FMR1	2332	fragile X mental retardation 1
	GABRB3	2562	gamma-aminobutyric acid (GABA) A receptor, beta 3
	GRIK2	2898	glutamate receptor, ionotropic, kainate 2
	ITGB3	3690	integrin, beta 3 (platelet glycoprotein IIIa, antigen CD61)
	MECP2	4204	methyl CpG binding protein 2 (Rett syndrome)
	MET	4233	met proto-oncogene (hepatocyte growth factor receptor)
	NLGN3	54413	neuroligin 3
	NLGN4X	57502	neuroligin 4, X-linked
	NRXN1	9378	neurexin 1
	OXTR	5021	oxytocin receptor
	PTEN	5728	phosphatase and tensin homolog (mutated in multiple advanced cancers 1)
	RELN	5649	reelin
	SHANK3	85358	SH3 and multiple ankyrin repeat domains 3
	SLC25A12	8604	solute carrier family 25 (mitochondrial carrier, Aralar), member 12
	SLC6A4	6532	solute carrier family 6 (neurotransmitter transporter, serotonin), member 4
	TSC1	7248	tuberous sclerosis 1
	TSC2	7249	tuberous sclerosis 2
	UBE3A	7337	ubiquitin protein ligase E3A (human papilloma virus E6-associated protein, Angelman syndrome)
Diabetes	CDKN2A	1029	cyclin-dependent kinase inhibitor 2A (melanoma, p16, inhibits CDK4)
	CDKN2B	1030	cyclin-dependent kinase inhibitor 2B (p15, inhibits CDK4)
	FTO	79068	fat mass and obesity associated
	HHEX	3087	homeobox, hematopoietically expressed
	KCNJ11	3767	potassium inwardly-rectifying channel, subfamily J, member 11
	PPARG	5468	peroxisome proliferator-activated receptor gamma
	SLC30A8	169026	solute carrier family 30 (zinc transporter), member 8
	TCF2	6928	transcription factor 2, hepatic; LF-B3; variant hepatic nuclear factor
	TCF7L2	6934	transcription factor 7-like 2 (T-cell specific, HMG-box)
	WFS1	7466	Wolfram syndrome 1 (wolframin)
Fanconi Anemia	ATM	472	ataxia telangiectasia mutated (includes complementation groups A, C and D)
	ATR	545	ataxia telangiectasia and Rad3 related
	BARD1	580	BRCA1 associated RING domain 1
	BLM	641	Bloom syndrome
	BRCA1	672	breast cancer 1, early onset
	MLH1	4292	mutL homolog 1, colon cancer, nonpolyposis type 2 (E. coli)
	NBS1	4683	nibrin
	PMS2	5395	PMS2 postmeiotic segregation increased 2 (S. cerevisiae)
	RAD51	5888	RAD51 homolog (RecA homolog, E. coli) (S. cerevisiae)
	TOP3A	7156	topoisomerase (DNA) III alpha
	TOPBP1	11073	topoisomerase (DNA) II binding protein 1
	USP1	7398	ubiquitin specific peptidase 1

Genes in *NatRev* dataset (Table B.3 continued).

Category	Gene Symbol	Entrez Gene ID	Official Gene Name
Mammary Gland Development	BMP4	652	bone morphogenetic protein 4
	BMPR1A	657	bone morphogenetic protein receptor, type IA
	DKK1	22943	dickkopf homolog 1 (Xenopus laevis)
	EDA	1896	ectodysplasin A
	EGFR	1956	epidermal growth factor receptor (erythroblastic leukemia viral (v-erb-b) oncogene homolog, avian)
	ERBB2	2064	v-erb-b2 erythroblastic leukemia viral oncogene homolog 2, neuro/glioblastoma derived oncogene homolog (avian)
	FGF10	2255	fibroblast growth factor 10
	FGF17	8822	fibroblast growth factor 17
	FGF4	2249	fibroblast growth factor 4 (heparin secretory transforming protein 1, Kaposi sarcoma oncogene)
	FGF8	2253	fibroblast growth factor 8 (androgen-induced)
	FGF9	2254	fibroblast growth factor 9 (glia-activating factor)
	FGFR2	2263	fibroblast growth factor receptor 2 (bacteria-expressed kinase, keratinocyte growth factor receptor, craniofacial dysostosis 1, Crouzon syndrome, Pfeiffer syndrome, Jackson-Weiss syndrome)
	FOXA1	3169	forkhead box A1
	GATA3	2625	GATA binding protein 3
	GLI1	2735	glioma-associated oncogene homolog 1 (zinc finger protein)
	GLI2	2736	GLI-Kruppel family member GLI2
	GLI3	2737	GLI-Kruppel family member GLI3 (Greig cephalopolysyndactyly syndrome)
	IGF1	3479	insulin-like growth factor 1 (somatomedin C)
	IGF1R	3480	insulin-like growth factor 1 receptor
	IRS1	3667	insulin receptor substrate 1
	IRS2	8660	insulin receptor substrate 2
	LEF1	51176	lymphoid enhancer-binding factor 1
	MSX2	4488	msh homeobox homolog 2 (Drosophila)
	PTHLH	5744	parathyroid hormone-like hormone
	PTHR1	5745	parathyroid hormone receptor 1
	RASGRF1	5923	Ras protein-specific guanine nucleotide-releasing factor 1
	SMO	6608	smoothened homolog (Drosophila)
	TBX15	6913	T-box 15
	TBX2	6909	T-box 2
	TBX3	6926	T-box 3 (ulnar mammary syndrome)
	TCF1	6927	transcription factor 1, hepatic; LF-B1, hepatic nuclear factor (HNF1), albumin proximal factor
	TCF3	6929	transcription factor 3 (E2A immunoglobulin enhancer binding factors E12/E47)
	TCF4	6925	transcription factor 4
	TNC	3371	tenascin C (hexabrachion)
	WNT10A	80326	wingless-type MMTV integration site family, member 10A
	WNT10B	7480	wingless-type MMTV integration site family, member 10B
	WNT6	7475	wingless-type MMTV integration site family, member 6

Genes in *NatRev* dataset (Table B.3 continued).

Category	Gene Symbol	Entrez Gene ID	Official Gene Name
Translation	AARS	16	alanyl-tRNA synthetase
	DKC1	1736	dyskeratosis congenita 1, dyskerin
	EEF1A2	1917	eukaryotic translation elongation factor 1 alpha 2
	EIF2AK1	27102	eukaryotic translation initiation factor 2-alpha kinase 1
	EIF2AK3	9451	eukaryotic translation initiation factor 2-alpha kinase 3
	EIF2B1	1967	eukaryotic translation initiation factor 2B, subunit 1 alpha, 26kDa
	EIF2B2	8892	eukaryotic translation initiation factor 2B, subunit 2 beta, 39kDa
	EIF2B3	8891	eukaryotic translation initiation factor 2B, subunit 3 gamma, 58kDa
	EIF2B4	8890	eukaryotic translation initiation factor 2B, subunit 4 delta, 67kDa
	EIF2B5	8893	eukaryotic translation initiation factor 2B, subunit 5 epsilon, 82kDa
	EIF4EBP1	1978	eukaryotic translation initiation factor 4E binding protein 1
	EIF4EBP2	1979	eukaryotic translation initiation factor 4E binding protein 2
	GARS	2617	glycyl-tRNA synthetase
	GFM1	85476	G elongation factor, mitochondrial 1
	GSPT1	2935	G1 to S phase transition 1
	LARS2	23395	leucyl-tRNA synthetase 2, mitochondrial
	PUS1	80324	pseudouridylate synthase 1
	RMRP	6023	RNA component of mitochondrial RNA processing endoribonuclease
	RPS19	6223	ribosomal protein S19
	RPS24	6229	ribosomal protein S24
	SBDS	51119	Shwachman-Bodian-Diamond syndrome
	SPG7	6687	spastic paraplegia 7, paraplegin (pure and complicated autosomal recessive)
	TSFM	10102	Ts translation elongation factor, mitochondrial
	TUFM	7284	Tu translation elongation factor, mitochondrial
	YARS	8565	tyrosyl-tRNA synthetase

Vita

Elina Tjioe was born in Medan, North Sumatra, Indonesia, on November 5, 1980. She graduated from Sutomo 1 High School in Medan in June, 1999 with a concentration in Natural Science. The following August, she entered Pellissippi State Technical Community College in Knoxville, Tennessee. In the fall of 2001, she enrolled at the University of Tennessee in Knoxville, graduating in 2003 Summa Cum Laude with a Bachelor of Science degree with a double major in Computer Science and Mathematics. She was then employed in a research laboratory at the University of Tennessee in the Biochemistry, Cellular and Molecular Biology Department as a Research Specialist III. In the fall of 2004, Elina enrolled as a doctoral student at the University of Tennessee in Life Sciences, with a concentration in Genome Science and Technology.