Krylov Subspace Algorithms for Computing GeneRank for the Analysis of Microarray Data Mining

GANG WU,1 YING ZHANG,2 and YIMIN WEI3

ABSTRACT

GeneRank is a new engine technology for the analysis of microarray experiments. It combines gene expression information with a network structure derived from gene notations or expression profile correlations. Using matrix decomposition techniques, we first give a matrix analysis of the GeneRank model. We reformulate the GeneRank vector as a linear combination of three parts in the general case when the matrix in question is non-diagonalizable. We then propose two Krylov subspace methods for computing GeneRank. Numerical experiments show that, when the GeneRank problem is very large, the new algorithms are appropriate choices.

Key words: Arnoldi method, gene network, GeneRank, Krylov subspace, microarray, PageRank.

1. INTRODUCTION

Ranking is an important task in the research field of gene networks, and a prioritized gene list is crucial in the analysis of gene expression data sets in molecular biology (Stekel, 2003; Devauckelle et al., 2001; Jaklič et al., 2006). Every genetic disease is caused by sequence variance of many genes; however, among these genes, only few of them play an important rule in causing disease. Therefore, one of the main goals for microarray data mining is to mine the few most important genes from thousands of genes in the experiments.

A microarray is a sequence of dots of DNA, protein, or tissue arranged on an array for easy simultaneous experiments. The most famous one is the DNA microarray, which plans an important role in gene expression profiling. Interpretation of simple microarray experiments is usually based on the fold-change of gene expression between a reference and a “treated” sample where the treatment can be of many types from drug exposure to genetic variation. Interpretation of the results usually combines lists of differentially expressed genes with previous knowledge about their biological function. However, some existing methods are heuristic and empirical, and are not efficient enough in many cases (Morrison et al., 2005; Yue et al., 2007).

A new mathematical model, called GeneRank, was proposed in 2005 by Morrison et al. (2005). Gene Rank is an intuitive modification of Google’s PageRank (Page et al., 1998; Langville and Meyer, 2005;
that maintains many of its mathematical properties. It combines gene expression information with a network structure derived from gene annotations (gene ontologies) or expression profile correlations.

The computation of GeneRank resorts to solving a large nonsymmetric linear system problem. Morrison et al. (2005) proposed a GeneRank algorithm for the computation of GeneRank. However, when the problem in question is very large, the GeneRank algorithm may not be efficient. Yue et al. (2007) reformulate the GeneRank model as a linear combination of three parts. Unfortunately, their formula requires that the (nonsymmetric) matrix involved is diagonalizable, which may not be satisfied in practice. Due to a great amount of gene expression data required, efficient computational methods are essential (Hai et al., 2008; Lin, 2008; Osmani and Rahman, 2007). It is now one of the hot topics in molecular biology and opens new challenges for seeking efficient algorithms for computing GeneRank, especially for very large gene networks.

Large-scale matrix computations is a fruitful area since there exists many excellent algorithms, such as the Krylov subspace method (Bai et al., 2000; Saad, 2003). Indeed, this type of method ranks among “The Top 10 Algorithms” of the 20th century (Cipra, 2000). In this article, we focus on the Krylov subspace methods and propose two Arnoldi-type algorithms for computing GeneRank. The idea stems from the facts that the GeneRank problem can be rewritten as a large-scale eigenvalue problem (Yue et al., 2007), and the largest eigenvalue of the GeneRank matrix is 1, as well as the refined projection strategies (Jia, 1997; Jia and Elsner, 2000; Golub and Greif, 2006).

This article is organized as follows. In Section 2, we briefly introduce the GeneRank model. In Section 3, we give a matrix analysis of the GeneRank model based on matrix decomposition techniques; first, we derive the eigenvalues of the GeneRank matrix, and then we reformulate the GeneRank vector as a linear combination of three parts, in the general case when the matrix in question is non-diagonalizable. In Section 4, we propose a restarted Arnoldi algorithm for computing GeneRank. In Section 5, we consider how to accelerate convergence of the Arnoldi algorithm using the \((m + 1)\)-th Arnoldi basis vector \(v_{m+1}\), and propose a modified Arnoldi algorithm for the computation of GeneRank. The distance between the optimal update vector and \(v_{m+1}\) is also discussed. Theoretical analysis indicates that the modified algorithm is possible to obtain dramatically faster convergence and improve results over the Arnoldi algorithm if the residual of the Arnoldi algorithm is close to \(A v_{m+1} - v_{m+1}\), where \(A\) is the GeneRank matrix in question. In Section 6, numerical experiments illustrate the numerical behaviors of the new algorithms.

Our notation is given as follows. Given a matrix \(A\) and a unit norm vector \(v\), we denote by

\[ \mathcal{K}_m(A,v) = \text{span} \{ v, Av, \cdots, A^{m-1}v \} \]

the Krylov subspace of dimension \(m\) (Bai et al., 2000; Saad, 2003). We denote by \(I\) the identity matrix with the order clear from context, by \(\sigma_{\text{max}}(X)\), \(\sigma_{\text{min}}(X)\) the largest and the smallest singular value of the matrix \(X\), respectively, and by “\(^T\)” and “\(^H\)” the transpose and conjugate transpose of a vector or matrix. Given two vectors \(x\), \(y\) of the same dimension, let

\[ \cos \angle(x,y) = \frac{|x^H y|}{\|x\|_2 \|y\|_2}, \]

be the cosine of the (acute) angle between the two vectors. Let \(\| . \|_1\) and \(\| . \|_2\) be the 1-norm and 2-norm of a vector, as well as the induced matrix norms, respectively (Golub and Van Loan, 1996), and let \(e = [1, 1, \cdots, 1]^T\) be the vector of all ones. MATLAB notation is used throughout this article.

2. THE GENERANK MODEL

The genetic information contained on each gene is not sufficient for understanding insight into biological processes (Hai et al., 2008). Therefore, understanding the mechanism of a biological system requires knowledge about regulatory network between genes, the so-called gene regulatory network (Akutsu et al., 1999).

In this section, we briefly introduce how to transform the GeneRank problem into a large-scale matrix computation problem (Morrison et al., 2005). A gene network can be constructed naturally via the Gene Ontology (GO) database (http://geneontology.org). Let the set \(G = \{ g_1, g_2, \cdots, g_N \} \) be the \(N\) genes on a
microarray. Similar to the idea of PageRank (Page et al., 1998), if a gene is connected with many high 
ranked genes, it should be ranked high, even if it may be ranked low by the experimental data (Morrison 
et al., 2005). If two genes share at least one annotation in GO, they are defined to be connected. From this 
idea, one can build a gene network whose adjacent matrix is $W$, with elements

$$w_{ij} = \begin{cases} 
1, & \text{if } g_i, g_j \text{ share one annotation in GO}, \\
0, & \text{else}.
\end{cases}$$

If the $i$-th row (column) of $W$ is zero, we reset it to be $e^T/N$ ($e/N$). Indeed, $W$ is symmetric as the network is 
undirected, i.e., $W^T = W$. This is unlike the PageRank problem, where a directed network is constructed. If 
we define

$$deg_i = \sum_{j=1}^{N} w_{ij} = \sum_{j=1}^{N} w_{ji},$$

(1)

to be the out-degree of the $i$-th gene, and $D = \text{diag}(deg_1, deg_2, \cdots, deg_N)$ to be a diagonal matrix. Then the 
GeneRank problem resorts to the following large-scale unsymmetric linear system

$$(I - d \cdot W^T D^{-1}) r^* = (1 - d) \cdot ex,$$

(2)

where $0 < d < 1$ is the damping factor, and

$$ex = [ex_1, ex_2, \cdots, ex_N]^T$$

with $ex_i \geq 0, i = 1, 2, \cdots, N$, is the absolute value of the expression change for $g_i$. The solution $r^*$ is the 
GeneRank vector, whose elements reflect the importance of the corresponding genes (Morrison et al., 2005).

In the GeneRank model, the damping factor $d$ plays an important role. If $d = 0$, then $r^*$ is nothing but $ex$. 
On the other hand, when $d = 1$, we have $W^T D^{-1} r^* = r^*$, and the solution is pure connectivity dependent. We 
point out in this case $r^*$ may not be unique, since $W^T D^{-1}$ may be reducible (Berman and Plemmons, 1994). 
Morrison et al. (2005) suggested using $d = 0.5$. However, the optimal choice of $d$ is still an interesting topic, 
and deserves further study.

### 3. A MATRIX ANALYSIS OF THE GENERANK MODEL

As was pointed out by Yue et al. (2007), the GeneRank problem (2) can also be reformulated as a large 
eigenvalue problem

$$[(1 - d) \cdot ex \cdot e^T + d \cdot W^T D^{-1}] r^* = r^*,$$

(3)

where we used $e^T r^* = \| r^* \|_1 = 1$. Thus, the GeneRank vector is nothing, but the eigenvector corresponding 
to the eigenvalue 1 of the GeneRank matrix defined as follows

$$A = d \cdot W^T D^{-1} + (1 - d) \cdot ex \cdot e^T.$$

(4)

It is easy to see from (1) and the definition of $D$ that $e^T (W^T D^{-1}) = e^T$. As a result

$$e^T [d \cdot W^T D^{-1} + (1 - d) \cdot ex \cdot e^T] = d \cdot e^T + (1 - d) \cdot e^T = e^T,$$

where we used $e^T \cdot ex = \| ex \|_1 = 1$. Therefore, both $A$ and $W^T D^{-1}$ are column stochastic matrices\(^1\) and 1 is an 
eigenvalue of both $W^T D^{-1}$ and $A$, with $e$ be the corresponding left eigenvector. As $W^T D^{-1}$ is a 
nonnegative matrix, we have $\| W^T D^{-1} \|_1 = 1$, and $\rho(W^T D^{-1}) \leq \| W^T D^{-1} \|_1 = 1$, where $\rho(B)$ denotes the 
spectral radius of a matrix $B$. As a result, 1 is the largest eigenvalue of $W^T D^{-1}$. Notice that a matrix and its 
transpose share the same eigenvalues, so 1 is also an eigenvalue of $D^{-1} W$ with $e$ being its right eigenvector.

---

\(^1\)A stochastic matrix is a nonnegative matrix with elements between 0 and 1, whose rows or columns are sum to 1 
(Berman and Plemmons, 1994).
Using matrix decomposition techniques, in this section, we consider the GeneRank problem from a matrix analysis point of view. Given the Schur decomposition of $D^{-1}W$ (which is the transpose of $W^TD^{-1}$), the following theorem gives the Schur decomposition of $A^T = d \cdot D^{-1}W + (1 - d) \cdot e \cdot e^T$.

**Theorem 1.** Let $L = ZH(D^{-1}W)Z$ be the Schur decomposition of $D^{-1}W$, where $Z = [e/\|e\|_2, Z_\perp] \in \mathbb{C}^{N \times N}$ is unitary, and $L \in \mathbb{C}^{N \times N}$ is upper triangular with diagonal elements $\{1, \lambda_2, \cdots, \lambda_N\}$ which are eigenvalues of $D^{-1}W$. Then

$$Z^H A^T Z = d \cdot L + (1 - d) \cdot \begin{bmatrix} 1 & \|e\|_2 \cdot e^T Z_\perp \\ 0 & O \end{bmatrix} = \tilde{L}$$

is the Schur decomposition of $A^T$, here $\tilde{L}$ is upper triangular with diagonal elements $\{1, d \cdot \lambda_2, \cdots, d \cdot \lambda_N\}$.

**Proof.** Note that

$$Z^H A^T Z = d \cdot L + (1 - d) \cdot Z^H (e \cdot e^T) Z.$$

Furthermore, we have

$$Z^H (e \cdot e^T) Z = \begin{bmatrix} e^T/Z_\perp \\ Z_\perp^H \end{bmatrix} \cdot (e \cdot e^T) \cdot \begin{bmatrix} e/Z_\perp \\ \|e\|_2 \cdot e^T Z_\perp \end{bmatrix} = \begin{bmatrix} \|e\|_2 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} 1/\|e\|_2 \cdot e^T Z_\perp \\ 0 \end{bmatrix} = \begin{bmatrix} 1/\|e\|_2 \cdot e^T Z_\perp \\ 0 \end{bmatrix},$$

where we used $Z^H e = O$ and $(e^T) e = \|e\|_1 = 1$. So we complete the proof by combining (5) with (6). □

**Remark 1.** Theorem 1 indicates that $A^T$ and $D^{-1}W$ share the same Schur vectors, and

$$Z^H A Z = L^T$$

is a lower triangular matrix with diagonal elements $\{1, d \cdot \lambda_2, \cdots, d \cdot \lambda_N\}$. Hence, given the eigenvalues $\{1, \lambda_2, \cdots, \lambda_N\}$ of $W^TD^{-1}$, the eigenvalues of $A$ are $\{1, d \cdot \lambda_2, \cdots, d \cdot \lambda_N\}$. We conclude that 1 is the largest eigenvalue of $A$, since $|d \cdot \lambda_i| < 1, \quad i = 2, 3, \cdots, N$. Consequently, the corresponding eigenvector $r^*$ of $A$, which is the GeneRank vector, is unique, even if $A$ is reducible.

Now we give insight into approximate solutions of the GeneRank vector. Assume that $A = \{1, \lambda_2, \cdots, \lambda_N\}$ are eigenvalues of the matrix $W^TD^{-1}$, and $X = \{x_1, x_2, \cdots, x_N\}$ are the corresponding eigenvectors. For any $d \epsilon [0, 1]$, it was proved in Yue et al. (2007) that $r^*$ can be formulated as

$$r^* = (1 - d) \cdot ex + d \cdot f_1 \frac{\text{deg}}{\|\text{deg}\|_1} + (1 - d) \cdot \sum_{i=2}^{N} \frac{d \cdot \lambda_i}{1 - d \cdot \lambda_i} f_i \cdot x_i,$$

where $f_i = x_i^H \cdot ex/\|x_i\|_2^2$. Moreover, they suggested using

$$\tilde{r}^* = (1 - d) \cdot ex + d \cdot f_1 \frac{\text{deg}}{\|\text{deg}\|_1}$$

as an approximate solution for the GeneRank vector $r^*$ (Yue et al., 2007).

However, it is seen that the derivation of (7) requires that $W^TD^{-1}$ has a complete set of eigenvectors, or in other words, the matrix is diagonalizable. Unfortunately, as $W^TD^{-1}$ is a nonsymmetric matrix, it may not be diagonalizable even if $W$ is symmetric. Therefore, it is necessary to consider the general case in which $W^TD^{-1}$ is non-diagonalizable. We have from (3) and (2) that

$$r^* = (1 - d) \cdot ex + d(1 - d) \cdot (W^TD^{-1})(I - d \cdot W^TD^{-1})^{-1}ex.$$
Recall that $\text{deg}/\|\text{deg}\|_2$ is a right eigenvector of $W^TD^{-1}$ corresponding to the largest eigenvalue 1 (Morisonet al., 2005), where $\text{deg}=[\text{deg}_1, \text{deg}_2, \cdots, \text{deg}_N]^T$. Let $W^TD^{-1} = URH$ be the Schur decomposition of $W^TD^{-1}$, where $R \in \mathbb{C}^{N \times N}$ is an upper triangular matrix. Without loss of generality, let the diagonal elements of $R$ be $\{1, \lambda_2, \cdots, \lambda_N\}$ which are eigenvalues of $W^TD^{-1}$, and let

$$U = \begin{bmatrix} \text{deg}/\|\text{deg}\|_2, u_2, \cdots, u_N \end{bmatrix} \in \mathbb{C}^{N \times N}$$

be unitary, whose columns are Schur vectors of $W^TD^{-1}$. It is easy to verify that

$$(W^TD^{-1})(I - d \cdot W^TD^{-1})^{-1} = UR(I - d \cdot R)^{-1}U^H,$$

and

$$R(I - d \cdot R)^{-1} = \begin{bmatrix} (1 - d)^{-1} \tilde{r}_{12} & \cdots & \tilde{r}_{1N} \\ \frac{\lambda_2}{1 - d \lambda_2} \tilde{r}_{22} & \cdots & \tilde{r}_{2N} \\ \vdots & \ddots & \vdots \\ \frac{\lambda_N}{1 - d \lambda_N} \tilde{r}_{NN} \end{bmatrix}.$$

Therefore,

$$(W^TD^{-1})(I - d \cdot W^TD^{-1})^{-1} \text{ex} = U \begin{bmatrix} (1 - d)^{-1} \tilde{r}_{12} & \cdots & \tilde{r}_{1N} \\ \frac{\lambda_2}{1 - d \lambda_2} \frac{\lambda_2}{1 - d \lambda_2} & \cdots & \tilde{r}_{2N} \\ \vdots & \ddots & \vdots \\ \frac{\lambda_N}{1 - d \lambda_N} \frac{\lambda_N}{1 - d \lambda_N} \end{bmatrix} U^H \text{ex}$$

$$= \begin{bmatrix} (1 - d)^{-\frac{\text{deg}^H}{\|\text{deg}\|_2}} \tilde{r}_{12} \frac{\text{deg}}{\|\text{deg}\|_2} + \frac{\lambda_2}{1 - d \lambda_2} u_2, \cdots, \frac{\lambda_N}{1 - d \lambda_N} u_N \end{bmatrix} U^H \text{ex}$$

$$= (1 - d)^{-\frac{\text{deg}^H}{\|\text{deg}\|_2}} \text{ex} + \frac{\lambda_2}{1 - d \lambda_2} u_2, \cdots, \frac{\lambda_N}{1 - d \lambda_N} u_N,$$

where $\tilde{u}_2 = \tilde{r}_{12} \frac{\text{deg}}{\|\text{deg}\|_2} + \frac{\lambda_2}{1 - d \lambda_2} u_2, \cdots, \tilde{u}_N = \tilde{r}_{1N} \frac{\text{deg}}{\|\text{deg}\|_2} + \frac{\lambda_N}{1 - d \lambda_N} u_N$. So we obtain from (9) and (10) that

**Theorem 2.** Under the above notation, there holds

$$r^* = (1 - d) \cdot \text{ex} + d \cdot \frac{\|\text{ex}\|_2}{\|\text{deg}\|_2} \cos \langle \text{ex}, \text{deg} \rangle \cdot \text{deg} + d(1 - d) \cdot \frac{\|\text{ex}\|_2}{\|\text{ex}\|_2} \tilde{u}_2 + \cdots + \frac{\lambda_N}{1 - d \lambda_N} \tilde{u}_N. \quad (11)$$

**Remark 2.** Note that $\|U^H\text{ex}\|_2/\|\text{ex}\|_2 = 1$, so we have $\cos^2 \langle \text{ex}, \text{deg} \rangle + \sum_{j=2}^{n} \cos^2 \langle u_j, \text{ex} \rangle = 1$. Therefore, Theorem 2 implies that if $\cos \langle \text{ex}, \text{deg} \rangle$ is close to 1, then

$$\tilde{r}^* = (1 - d) \cdot \text{ex} + d \cdot \frac{\text{deg}^H}{\|\text{deg}\|_2} \text{deg} \quad (12)$$

will be a good approximation to the GeneRank vector $r^*$. 
4. AN ARNOLDI ALGORITHM FOR THE GENERANK PROBLEM

Modern numerical linear algebra makes use of the Krylov subspace methods in different advanced iterative procedures (Bai et al., 2000; Saad, 2003). Indeed, the Krylov subspace methods rank among “The Top 10 Algorithms” of the 20th century (Cipra, 2000). The Arnoldi method (Bai et al., 2000) is one of the most popular Krylov subspace methods for computing a few dominant eigenpairs of a large matrix. In Sections 4 and 5, we focus on how to solve the large eigenvalue problem (3) using Arnoldi-type algorithms.

Given an initial vector $v_1$ of unit norm, if computations are performed in exact arithmetic, then the $m$-step Arnoldi process generates successively an orthonormal basis $V_{m+1} = [v_1, v_2, \ldots, v_m, v_{m+1}]$ for the Krylov subspace $K_{m+1}(A, v_1) = \text{span}\{v_1, Av_1, \ldots, A^m v_1\}$. In the subspace $K_m(A, v_1)$, the restriction of $A$ is represented by an $m \times m$ upper Hessenberg matrix $H_m$ with the entries $h_{ij}$. Furthermore, the following relations hold (Bai et al., 2000; Saad, 2003)

$$AV_m = V_m H_m + h_{m+1, m} v_m + e_m^T = V_{m+1} \tilde{H}_m,$$  

(13)

where $e_m$ is the $m$-th coordinate vector of dimension $m$, and $\tilde{H}_m$ is an $(m+1) \times m$ upper Hessenberg matrix, which is the same as $H_m$ except for an additional row whose unique nonzero entry is $h_{m+1, m}$. One variant of the Arnoldi process is described as follows (Bai et al., 2000; Saad, 2003).

**Algorithm 1** The $m$-step Arnoldi process

1. **Start:** Given the initial vector $v_1$ of unit norm, and the steps $m$ of the Arnoldi process;
2. **Iterate:**
   
   for $j = 1, 2, \ldots, m$
   
   $q = Av_j$

   for $i = 1, 2, \ldots, j$
   
   $h_{i,j} = v_i^T q$

   $q = q - h_{i,j} v_i$

   end for

   $h_{j+1,j} = \|q\|_2$;

   if $h_{j+1,j} == 0$

   break;

   end if

   $v_{j+1} = q/h_{j+1,j}$;

   end for

Inspired by the fact that the largest eigenvalue of the GeneRank matrix is 1, and a strategy due to Golub and Grief for PageRank (Golub and Greif, 2006), we propose an Arnoldi algorithm for the GeneRank problem. In the new algorithm, one seeks a unit norm vector $x^G \in K_m(A, v_1)$ that satisfies

$$\| (A - I) x^G \|_2 = \min_{\|u\|_2 = 1} \| (A - I) u \|_2.$$  

(14)

The difference between the new method and the refined Arnoldi method (Jia, 1997) is that the former uses the fact that the largest eigenvalue of the GeneRank matrix is 1, while the latter uses Ritz values as shifts, which may be complex in practice.

We consider how to solve (14) efficiently. At the first glance, one has to solve a large optimal problem for (14). Indeed, we can solve this problem by means of a small singular value decomposition, in $O(m^3)$ flops. Denote by $[I; O]$ an $(m+1) \times m$ matrix which is the same as the $m \times m$ identity matrix, except an additional zero row in the bottom. It follows from (13) that
while

\[(A - I)x^G\] 

\[\min_{\|u\|_2 = 1} \| (A - I)u \|_2 \]

\[= \min_{\|u\|_2 = 1} \| (A - I)V_m z \|_2 \]

\[= \min_{\|u\|_2 = 1} \| V_m + 1 (H_m - [I; O]) \|_2 \]

\[= \min_{\|u\|_2 = 1} \| (H_m - [I; O])z \|_2, \]

(15)

where in the last step we use the fact that \(V_m + 1\) is unitary. Let \(H_m - [I; O] = UTS^T\) be the singular vector decomposition (SVD) of the matrix, then it is easy to see from (15) that

\[x^G = V_mz^G,\]

(16)

where \(z^G = S(:, m)\) is the right singular vector corresponding to the smallest singular value \(\sigma_{\min}(H_m - [I; O])\), and

\[r^G = (A - I)x^G = \sigma_{\min}(H_m - [I; O]) \cdot V_m + 1 U(:, m),\]

(17)

is the residual vector with \(\|r^G\|_2 = \sigma_{\min}(H_m - [I; O])\), where \(U(:, m)\) is the \(m\)-th column of \(U\), which is the left singular vector associated with \(\sigma_{\min}(H_m - [I; O])\).

In practical calculations, the standard implementations of Arnoldi’s method are limited by the high storage and computational requirements as \(m\) increases. One way to circumvent this difficulty is to restart the algorithm (Bai et al., 2000; Saad, 2003). That is, after a run with \(m\) Arnoldi vectors, we compute the approximate eigenvector and use it as an initial vector for the next run with Arnoldi’s method. The frame of a restarted Arnoldi algorithm for GeneRank is outlined as follows.

---

**Algorithm 2** A restarted Arnoldi algorithm for computing GeneRank

1. **Start**: Given an initial guess \(v_1\) of unit norm, the Arnoldi steps number \(m\), as well as a prescribed tolerance \(tol\); Set \(\rho = 1\);

2. **Iterate**
   
   **while** \(\rho > tol\)
   
   Run Algorithm 1 for the computation of \(V_m + 1\) and \(H_m\);
   
   Compute small SVD: \(H_m - [I; O] = UTS^T\);
   
   Approximate eigenvector: \(v_1 = V_m S(:, m)\); % \(x^G\), the initial vector for the next iteration
   
   Compute the residual norm \(\rho\) of the approximate eigenvector;
   
   **end**

One can use the 2-norm of residual \(\|r^G\|_2\) as a cheap convergence criterion, which is a by-product of the Arnoldi algorithm. However, as the GeneRank vector is the stationary distribution of a Markov chain, it is often normalized so that its 1-norm is one, thus it is preferable to use the 1-norm for residual as the stopping criterion. To do this, one option is to evaluate the 1-norm of residual directly: \(\rho = \|Ax^G - x^G\|_1/\|x^G\|_1\).

However, we have to pay a superfluous matrix-vector product for this. Indeed, the matrix-vector product can be realized implicitly: By (13), we have

\[\rho = \|Ax^G - x^G\|_1/\|x^G\|_1 = \|V_m + 1[H_m S(:, m)] - V_m S(:, m)\|_1/\|V_m S(:, m)\|_1,\]

(18)

which can be calculated in \(O(N)\) flops.

---

**5. A MODIFIED ARNOLDI ALGORITHM FOR COMPUTING GENERANK**

In this section, we propose a modified Arnoldi algorithm for computing GeneRank. The key idea is to exploit \(v_m + 1\) which is the \((m + 1)\)-th basis vector of \(K_m + 1(A, v_1)\), to accelerate the convergence of the Arnoldi algorithm proposed in the Section 4.
5.1. A new algorithm for computing GeneRank

Recall that the approximation \( x^G \) obtained from the Arnoldi algorithm is in \( \mathcal{K}_m(A, v_1) = \text{span}\{v_1, v_2, \cdots, v_m\} \), while the \( m \)-step Arnoldi process generates an orthonormal basis for \( \mathcal{K}_{m+1}(A, v_1) = \text{span}\{v_1, v_2, \cdots, v_m, v_{m+1}\} \). Motivated by the strategies proposed in Jia and Elsner (2000) and Wu (2007), and the fact that the largest eigenvalue of the Google matrix is 1, we seek a vector \( x^G + \hat{\beta} v_{m+1} \in \mathcal{K}_{m+1}(A, v_1) \) that satisfies the following optimal property

\[
\begin{align*}
\| (A - I)(x^G + \hat{\beta} v_{m+1}) \|_2 &= \min_{\beta \in \mathbb{C}} \| (A - I)(x^G + \beta v_{m+1}) \|_2 \\
&= \min_{\beta \in \mathbb{C}} \| r^G + \hat{\beta}(Av_{m+1} - v_{m+1}) \|_2.
\end{align*}
\]

Let \( \hat{\beta} \) be the minimizer of (19), then the modified Arnoldi-type algorithm uses

\[
x^M = \frac{x^G + \hat{\beta} v_{m+1}}{\| x^G + \hat{\beta} v_{m+1} \|_2},
\]

to approximate the GeneRank vector \( r^* \). The following theorem shows how to compute \( \hat{\beta} \) efficiently in practice.

**Theorem 3.** If we denote by \( w_{m+1} = Av_{m+1} - v_{m+1} \), then

\[
\hat{\beta} = -\frac{w_{m+1}^T r^G}{\| w_{m+1} \|_2},
\]

and the new approximation \( x^M \) satisfies

\[
\begin{cases}
  x^M \in \text{span}\{x^G, v_{m+1}\} \\
  (A - I)x^M \bot \text{span}\{w_{m+1}\}.
\end{cases}
\]

**Proof.** It is well known that the minimizer \( \hat{\beta} \) of (19) satisfies (Golub and Van Loan, 1996)

\[
(w_{m+1}^T w_{m+1}) \hat{\beta} = -w_{m+1}^T r^G,
\]

from which we get (20); and (21) follows from \( (A - I)x^M = r^G + \hat{\beta}w_{m+1} \) and (20).

If we denote by \( r^M = (A - I)x^M \) the residual with respect to \( x^M \), the following theorem reveals the relationship between \( \| r^G \|_2 \) and \( \| r^M \|_2 \), which can be easily monitored in practice.

**Theorem 4.** Under the above notation, we have

\[
\| r^M \|_2 = \sqrt{\frac{\sigma_{\min}(H_m - I; O) - \hat{\beta}^2 \| w_{m+1} \|_2^2}{1 + \hat{\beta}^2}} \| r^G \|_2 \sin \angle(r^G, w_{m+1})
\]

\[
= \sqrt{1 + \frac{\| r^M \|_2^2}{\| w_{m+1} \|_2^2}} \cos^2 \angle(r^G, w_{m+1})
\]

and

\[
\frac{\| r^M - r^G \|_2}{\| r^G \|_2} = \cos \angle(r^G, w_{m+1}).
\]

**Proof.** It is seen that

\[
\| r^M \|_2 = \frac{\| r^G + \hat{\beta} w_{m+1} \|_2}{\| x^G + \hat{\beta} v_{m+1} \|_2}.
\]
On the one hand, the squared numerator is

\[ \left\| r^G + \beta w_{m+1} \right\|_2^2 = \left\| r^G \right\|_2^2 + 2 \beta \left( r^G, w_{m+1} \right) + \beta^2 \left\| w_{m+1} \right\|_2^2 = \left\| r^G \right\|_2^2 - \beta^2 \left\| w_{m+1} \right\|_2^2 = \sigma_{\text{min}}^2 (H_m - [I; O]) - \beta^2 \left\| w_{m+1} \right\|_2^2, \]

where we used (20). On the other hand, the squared denominator is

\[ \left\| x^G + \beta v_{m+1} \right\|_2^2 = \left\| x^G \right\|_2^2 + 2 \beta \left( x^G, v_{m+1} \right) + \beta^2 \left\| v_{m+1} \right\|_2^2 = 1 + \beta^2, \]

where we used the facts \( \left\| x^G \right\|_2 = 1, \left\| v_{m+1} \right\|_2 = 1, \) and \( v_{m+1}^T x^G = 0. \) For (22), we have

\[ \left\| r^M \right\|_2^2 = \frac{\left\| r^G \right\|_2^2 - \frac{\left\| w_{m+1} + r^G \right\|_2^2}{\left\| w_{m+1} \right\|_2^2}}{1 + \frac{\left\| w_{m+1} + r^G \right\|_2^2}{\left\| w_{m+1} \right\|_2^2}} = \frac{\left\| r^G \right\|_2^2 - \frac{\left\| w_{m+1} + r^G \right\|_2^2}{\left\| w_{m+1} \right\|_2^2}}{1 + \frac{\left\| w_{m+1} \right\|_2^2}{\left\| w_{m+1} \right\|_2^2}} \cdot \frac{\left\| r^G \right\|_2^2}{\left\| r^G \right\|_2^2} = \frac{\left\| r^G \right\|_2^2 - \frac{\left\| w_{m+1} + r^G \right\|_2^2}{\left\| w_{m+1} \right\|_2^2}}{1 + \frac{\left\| w_{m+1} \right\|_2^2}{\left\| w_{m+1} \right\|_2^2}} = \frac{\left\| r^G \right\|_2^2 \cdot \sin^2 \angle (r^G, w_{m+1})}{1 + \frac{\left\| r^G \right\|_2^2}{\left\| w_{m+1} \right\|_2^2 \cdot \cos^2 \angle (r^G, w_{m+1})}}. \]

For (23), we have

\[ \frac{\left\| r^M - r^G \right\|_2}{\left\| r^G \right\|_2} = \frac{\left\| \hat{\beta} \right\| \left\| w_{m+1} \right\|_2}{\left\| w_{m+1} \right\|_2} = \frac{\left\| w_{m+1} \cdot r^G \right\|}{\left\| w_{m+1} \right\|_2 \left\| r^G \right\|_2} = \cos \angle (r^G, w_{m+1}). \]

### Remark 3.

Theorem 4 indicates that \( \left\| r^M \right\|_2 < \left\| r^G \right\|_2 \) provided \( \sin \angle (r^G, w_{m+1}) < 1. \) Moreover, the smaller the distance between \( r^G \) and \( w_{m+1} \) is, the larger the difference between \( r^G \) and \( r^M \) will be, or in other words, the better the new approximation \( x^M \) will be.

In summary, we present the main algorithm of this article.

### Algorithm 3 A restarted modified Arnoldi algorithm for computing GeneRank

1. **Start:** Given an initial guess \( v_1 \) of unit norm, the Arnoldi steps number \( m \), as well as a prescribed tolerance \( tol \); Set \( \rho = 1; \)
2. **Iterate**
   while \( \rho > tol \)
   
   (2.1) Run Algorithm 1 for the computation of \( V_{m+1} = [V_m, v_{m+1}] \) and \( H_{m+1} \);
   
   (2.2) Compute small SVD: \( H_{m+1} - [I; O] = U T S^T \), and set \( \rho = \sigma_{\text{min}} (H_{m+1} - [I; O]) \);
   
   (2.3) Compute the new approximation \( x^M \):

   \[
   \begin{align*}
   w &= A_{V_m+1} + v_{m+1};\gamma = \left\| w \right\|_2^2; \\
   \hat{\beta} &= -\rho \cdot w^T [V_{m+1} \cdot U (:, m)] / \gamma; \% r^G = \sigma_{\text{min}} (H_{m+1} - [I; O]) \cdot V_{m+1} \cdot U (:, m) \\
   v_1 &= v_{m+1} \cdot (IS (:, m); \hat{\beta}); \% x^M, \text{ the new approximation} \\
   w &= w + v_{m+1}; \% AV_{m+1} \\
   w &= v_{m+1} [H_{m+1} S (:, m)] + \hat{\beta} w; \% Ax^M \\
   \rho &= \left\| w - v_1 \right\|_2 / \left\| v_1 \right\|_2; \% r = \left\| Ax^M - x^M \right\|_1 / \left\| x^M \right\|_1, \text{ the residual with 1-norm} \\
   v_1 &= w / \left\| w \right\|_2, \% Ax^M / \left\| x^M \right\|_2, \text{ the initial vector for the next Arnoldi iteration}
   \end{align*}
   \]
end
Remark 4. We make use of \( Ax^M = V_{m+1}[\bar{H}_mS(:, m)] + \bar{B}Av_{m+1} \) as the initial vector for the next iteration, which is available from the current iteration. Recall that the restated Arnoldi algorithm uses \( x^G = V_mS(:, m) \) as the initial guess for the next iteration.

In Algorithm 3, we evaluate the 1-norm of residual as follows:

\[
\|Ax^M - x^M\|_1 \approx \frac{A[V_mS(:, m) + \hat{B}v_{m+1}] - [V_mS(:, m) + \hat{B}v_{m+1}]}{\|V_mS(:, m) + \hat{B}v_{m+1}\|_1} = \frac{[V_{m+1}\bar{H}_mS(:, m) + \bar{B}Av_{m+1}] - [V_mS(:, m) + \hat{B}v_{m+1}]}{\|V_mS(:, m) + \hat{B}v_{m+1}\|_1}. \tag{24}
\]

Special care should be taken on the tradeoff between speedup and overhead. Compared with the \( m \)-step standard Arnoldi algorithm, the main overhead for computing \( x^M \) involves one matrix-vector multiplication (depending on the number of non-zeros of the matrix), the computation of one inner product, as well as the storage of an \( N \)-dimensional vector. We point out that the \( m \)-step modified Arnoldi algorithm is cheaper than the \((m+1)\)-step standard Arnoldi algorithm. Compared with the \( m \)-step standard Arnoldi algorithm, the \((m+1)\)-step standard one requires not only an additional matrix-vector product, but also (at least) \( m+1 \) additional inner products, as well as the storage of an \( N \)-dimensional vector.

Table 1 lists the computational complexities required by the two algorithms per iteration, where Arnoldi \((m+1)\) and Modi-Arn \((m)\) stand for the \((m+1)\)-step standard Arnoldi algorithm and the \( m \)-step modified Arnoldi algorithm, respectively, and “Mat-Vec Products” stands for the number of matrix-vector products required.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Arnoldi ((m+1))</th>
<th>Modi-Arn ((m))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matrix-vector products</td>
<td>(m+1)</td>
<td>(m+1)</td>
</tr>
<tr>
<td>Storage</td>
<td>(m+2)</td>
<td>(m+2)</td>
</tr>
<tr>
<td>Inner products</td>
<td>((m^2 + 3m + 2)/2)</td>
<td>((m^2 + m + 2)/2)</td>
</tr>
</tbody>
</table>

5.2. When \( v_{m+1} \) is an appropriate choice?

Generally speaking, \( v_{m+1} \) is not the optimal choice of \( u \) in the update \( x^M = x^G + \bar{B}u \). In this section, we consider how to seek the optimal choice from a theoretical point of view. Our aim is to shed light on when \( v_{m+1} \) can be close to the optimal update, or in other words, \( v_{m+1} \) is an appropriate choice for updating \( x^G \).

In terms of Theorem 4, it is reasonable to find a vector \( v_{opt} \in span\{V_m\}^\perp \), such that

\[
\sin \angle(r^G, (A-I)v_{opt}) = \min_{v \in span\{V_m\}^\perp : \|v\|_2 = 1} \sin \angle(r^G, (A-I)v),
\]

where \( span\{V_m\}^\perp \) denotes the orthogonal complement space of \( K_m(A, v_1) = span\{V_m\} \). The following theorem indicates how to choose the optimal vector theoretically.

Theorem 5. Let \( V_m^\perp \) be an orthonormal basis of \( span\{V_m\}^\perp \) such that \( V_m, V_m^\perp \) is unitary. Let \((A-I)V_m^\perp = P\Sigma Q^T\) be the singular value decomposition, where \( P \in \mathbb{R}^{n \times l}, Q \in \mathbb{R}^{m \times l} \) are orthonormal, and \( \Sigma = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_l) \in \mathbb{R}^{l \times l} \) is nonsingular. Then

\[
\min_{v \in span\{V_m^\perp\}^\perp : \|v\|_2 = 1} \sin \angle(r^G, (A-I)v)
\]

is attained by

\[
v_{opt} = \eta \cdot V_m^\perp (Q\Sigma^{-1}P^T r^G),
\]

where \( \eta \) is the normalizing factor.
Proof. For any \( v \in \text{span}\{V_m\}^\perp \), to minimize \( \sin \angle (r^G, (A - I)v) \), we just need to maximize \( \cos \angle (r^G, (A - I)v) \). Let \((A - I)V_m^\perp = P\Sigma Q^T\) be the singular value decomposition of \((A - I)V_m^\perp\), then

\[
\cos \angle (r^G, (A - I)v) = \frac{\|[(A - I)v]^T r^G\|}{\|r^G\|_2 \cdot \|(A - I)v\|_2} = \frac{\|[(A - I)V_m^\perp v]^T r^G\|}{\|r^G\|_2 \cdot \|(A - I)V_m^\perp v\|_2} = \frac{\|P\Sigma Q^T r^G\|_2}{\|r^G\|_2 \cdot \|\Sigma y\|_2}, \%
\]

which is maximized when

\[
P(\Sigma \tilde{y}) = \delta \cdot r^G,
\]

with \( \delta \in \mathbb{C} \) is a scalar. Denote by \( \tilde{y} = Q\tilde{y} \), we obtain

\[
\tilde{y} = \delta (Q\Sigma^{-1}P^T)r^G,
\]

so we complete the proof by setting \( v_{opt} = \eta \cdot V_m^\perp \tilde{y} \), where \( \eta = 1/\|\tilde{y}\|_2 \) is the normalizing factor.

It is interesting to evaluate how much \( v_{m+1} \) differs from \( v_{opt} \). The following theorem gives an upper bound for the difference between the two vectors.

**Theorem 6.** Let \( \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_t > 0 \) be the non-zero singular values of \((A - I)V_m^\perp\), where \( t \leq n - m \). Denote by

\[
\phi^T \equiv e_1^T Q = [q_1, q_2, \cdots, q_t], \quad \psi \equiv P^T r^G = [\hat{p}_1, \hat{p}_2, \cdots, \hat{p}_t]^T,
\]

respectively, where \( e_1 \) denotes the first coordinate vector. If \( v_{m+1} = V_m^\perp e_1 \) and

\[
\left| \sum_{i=1}^t \frac{\hat{p}_i q_i}{\sigma_i} \right| \geq \frac{1}{\sigma_1} \left| \sum_{i=1}^t \hat{p}_i q_i \right|, \tag{26}
\]

then

\[
\sin \angle (v_{opt}, v_{m+1}) \leq \sqrt{1 - \left( \frac{\|\phi\|_2 \cos \angle (\phi, \psi)}{\kappa((A - I)V_m^\perp)} \right)^2}, \tag{27}
\]

here \( \kappa(B) = \sigma_{max}(B)/\sigma_{min}(B) \) is the condition number of a matrix \( B \) with the 2-norm.

**Proof.** It follows from Theorem 5 that

\[
\cos \angle (v_{opt}, v_{m+1}) = \cos \angle (V_m^\perp \tilde{y}/\|V_m^\perp \tilde{y}\|_2, V_m^\perp e_1) = \cos \angle (e_1, \tilde{y}/\|\tilde{y}\|_2) = \frac{\|e_1^T Q\Sigma^{-1}(P^T r^G)\|_2}{\|\Sigma^{-1}(P^T r^G)\|_2}.
\]

On the one hand, it follows from (26) that

\[
\left| (e_1^T Q)\Sigma^{-1}(P^T r^G) \right| = \left| \sum_{i=1}^t \frac{\hat{p}_i q_i}{\sigma_i} \right| \geq \frac{1}{\sigma_1} \left| \sum_{i=1}^t \hat{p}_i q_i \right|.
\]

On the other hand, we have

\[
\|\Sigma^{-1}(P^T r^G)\|_2 \leq \frac{1}{\sigma_1} \|\phi\|_2.
\]

Therefore,
\[ \cos \angle(v_{opt}, v_{m+1}) \geq \frac{\sigma_1}{\sigma_1} \frac{|\psi^T \varphi|}{\|\varphi\|_2} = \frac{\|\varphi\|_2 \cos \angle(\varphi, \psi)}{\kappa[(A-I)V_m^\perp]} , \]

where we used \( \kappa[(A-I)V_m^\perp] = \sigma_1 / \sigma_j. \)

Remark 5. Theorem 6 shows that if the angle between \( \varphi \) and \( \psi \) is not large, and \( (A-I)V_m^\perp \) is not too ill-conditioned, then \( v_{m+1} \) is close to \( v_{opt} \), or in other words, \( v_{m+1} \) will be an appropriate choice for updating \( x^T \).

6. NUMERICAL EXAMPLES

In this section, we present some numerical examples to illustrate the numerical behavior of the new algorithms. All the numerical experiments were run on a dual core Intel(R) Pentium(R) processor with CPU 1.86 GHz and RAM 1GB under the Windows XP operating system. We report on experimental results obtained with a MATLAB 7.0 implementation with machine precision \( \varepsilon = 2.22 \times 10^{-16} \). The GeneRank data files are provided by Morrison et al. (2005) (available at www.biomedcentral.com/content_supplementary/1471-2105-6-233-S2.mat). We run the GeneRank algorithm (whose MATLAB implementation is available at www.biomedcentral.com/content_supplementary/1471-2105-6-233-S1.m), the restarted Arnoldi algorithm (Arnoldi), as well as the restarted modified Arnoldi algorithm (Modi-Arn) on all the test matrices.

The choice of \( d \) is crucial to the GeneRank model. It was pointed out in Morrison et al. (2005) that the optimal choice of \( d \) is data-dependent. Therefore, in all the numerical experiments below, we choose \( d = 0.5, 0.6, 0.7, 0.8 \) and \( 0.85 \), respectively. We note that the GeneRank algorithm exploits a direct method (Golub and Van Loan, 1996) for the linear system (2). For the sake of justice, the stopping criterion for the Arnoldi-type algorithms is

\[ \|A\vec{x} - \vec{x}\|_1 / \|\vec{x}\|_1 \leq 10^{-15}, \]

where \( \vec{x} \) are approximations to the GeneRank vector. In practical computations, the Arnoldi and the modified Arnoldi algorithms will become impractical when \( m \), the steps of the Arnoldi process, is large because of the growth of memory and computational requirements as \( m \) increases. Therefore, it is desirable to pick \( m \) as small as possible in the two algorithms. In all the numerical experiments, we set \( m = 3 \) for the Arnoldi and the modified Arnoldi algorithms, and use \( v_1 = ex / \|ex\|_2 \) as the initial vector, where \( ex \) is the experimental data.

We would like to remind the reader that the GeneRank matrix \( A \) defined in Section 3 is a full matrix in general, and it is impracticable to be formed explicitly. Given an vector \( x = [x_1, x_2, \ldots, x_N]^T \), in the Arnoldi and the modified Arnoldi algorithms, we compute the matrix-vector products as follows (refer to (4)):

\[ Ax = d \cdot W(x./\text{deg}) + (1-d) \cdot (e^T x) \cdot ex, \]

where \( \text{deg} = [\text{deg}_1, \text{deg}_2, \ldots, \text{deg}_N]^T \), and "/" is the right array divide command in MATLAB.

Example 1. In the 1st example, we show that new algorithms are superior to the GeneRank algorithm for computing GeneRank. The test matrix is the \( w_{\text{All}} \) matrix, it is of size 4047 \times 4047, with 339596 nonzero elements, and the experimental data is \( ex = \text{extr data} \). Table 2 lists the numerical results. We observe that both the new algorithms are better than the GeneRank algorithm in terms of CPU time in seconds, while the modified Arnoldi algorithm performs the best.

For the Arnoldi algorithm and its modified version, we see that the larger the damping factor is, the more the CPU time are needed for convergence. Indeed, it follows from Theorem 1 that the second largest eigenvalue of the GeneRank matrix is \( d \cdot \lambda_2 \), whose absolute value will approach 1 as \( d \) is close to 1. As a result, the large eigenproblem (3) will become a clustered eigenvalue problem that is difficult to be solved in practice (Bai et al., 2000). So as to compare the Arnoldi algorithm with the modified one, in Figure 1 we depict the convergence curves of the two algorithms with \( d = 0.5 \) and \( 0.85 \), respectively. It is obvious to see that the modified Arnoldi algorithm converges much faster than the Arnoldi algorithm.
Example 2. This example aims to show that (12) can be used as an approximation to the Gene-Rank vector. The test matrix is the $w_{\text{Down}}$ matrix, which is of size $1625 \times 1625$, with 67252 nonzero elements. The experimental data is $ex = \text{extr\_data\_Down}$. Table 3 lists the numerical results obtained. Similarly, it is seen that both the new algorithms work better than the GeneRank algorithm, and the modified Arnoldi algorithm performs the best.

In order to illustrate that (12) can be used as an approximation to the GeneRank vector, we plot the bar graphs of the “exact” GeneRank vector $r^*$ (obtained from the GeneRank algorithm), and that of the approximation (12) when $d = 0.5$ (Fig. 2). We see that the two figures are similar to each other.

Example 3. In the 3rd example, we try to show that (12) is a better approximation to the GeneRank vector than (8). The test matrix is the $w_{\text{Up}}$ matrix, it is of size $2392 \times 2392$, with 128468 nonzero elements. The experimental data is $ex = \text{extr\_data\_Up}$. Table 4 presents the numerical results. As before, the Arnoldi algorithm outperforms the GeneRank algorithm, while the modified Arnoldi algorithm is superior to the other two.

Figure 3 depicts GeneRank values (sorted in descending order) of the top 100 genes when $d = 0.5$. It is seen that our approximation (12) is more accurate than (8) provided in Yue et al. (2007). Again, we observe that our approximation can be used as a good approximation to the exact solution.

![Figure 1](image_url)

**FIG. 1.** Example 1: Convergence curves of the restarted Arnoldi algorithm and the restarted modified Arnoldi algorithm on the $w_{\text{All}}$ matrix, $tol = 10^{-15}$, $N = 4047$.  

---

**Table 2. Three Algorithms on the $w_{\text{All}}$ Matrix**

<table>
<thead>
<tr>
<th>$d$</th>
<th>GeneRank</th>
<th>Arnoldi</th>
<th>Modi-Arn</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>30.9</td>
<td>0.47</td>
<td>0.30</td>
</tr>
<tr>
<td>0.6</td>
<td>30.9</td>
<td>0.61</td>
<td>0.36</td>
</tr>
<tr>
<td>0.7</td>
<td>30.9</td>
<td>0.77</td>
<td>0.44</td>
</tr>
<tr>
<td>0.8</td>
<td>30.9</td>
<td>1.03</td>
<td>0.56</td>
</tr>
<tr>
<td>0.85</td>
<td>30.9</td>
<td>1.25</td>
<td>0.64</td>
</tr>
</tbody>
</table>

Example 1: CPU time (in seconds) of the GeneRank algorithm, the Arnoldi algorithm ($m = 3$), and the modified Arnoldi-type algorithm ($m = 3$) on the $4047 \times 4047$ $w_{\text{All}}$ matrix.
FIG. 2. Example 2: Bar graphs of the exact solution and the approximation (12), the w_Down matrix, $d = 0.5$, $N = 1625$.

Table 3. Three Algorithms on the w_Down Matrix

<table>
<thead>
<tr>
<th>$d$</th>
<th>GeneRank</th>
<th>Arnoldi</th>
<th>Modi-Arn</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>2.20</td>
<td>0.11</td>
<td>0.08</td>
</tr>
<tr>
<td>0.6</td>
<td>2.20</td>
<td>0.14</td>
<td>0.08</td>
</tr>
<tr>
<td>0.7</td>
<td>2.19</td>
<td>0.17</td>
<td>0.09</td>
</tr>
<tr>
<td>0.8</td>
<td>2.22</td>
<td>0.25</td>
<td>0.14</td>
</tr>
<tr>
<td>0.85</td>
<td>2.22</td>
<td>0.30</td>
<td>0.14</td>
</tr>
</tbody>
</table>

Example 2: CPU time (in seconds) of the GeneRank algorithm, the Arnoldi algorithm ($m = 3$), and the modified Arnoldi-type algorithm ($m = 3$) on the $1625 \times 1625$ w_Down matrix.

Table 4. Three Algorithms on the w_Up Matrix

<table>
<thead>
<tr>
<th>$d$</th>
<th>GeneRank</th>
<th>Arnoldi</th>
<th>Modi-Arn</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>6.55</td>
<td>0.17</td>
<td>0.13</td>
</tr>
<tr>
<td>0.6</td>
<td>6.67</td>
<td>0.22</td>
<td>0.14</td>
</tr>
<tr>
<td>0.7</td>
<td>6.67</td>
<td>0.28</td>
<td>0.17</td>
</tr>
<tr>
<td>0.8</td>
<td>6.67</td>
<td>0.38</td>
<td>0.22</td>
</tr>
<tr>
<td>0.85</td>
<td>6.67</td>
<td>0.45</td>
<td>0.22</td>
</tr>
</tbody>
</table>

Example 3: CPU time (in seconds) of the GeneRank algorithm, the Arnoldi algorithm ($m = 3$), and the modified Arnoldi-type algorithm ($m = 3$) on the $2392 \times 2392$ w_Down matrix.
7. CONCLUSION

The GeneRank model provides an alternative method of evaluating microarray experimental results which combines prior knowledge about the underlying network. In this article, first we give a matrix analysis of the GeneRank model, and provide an approximation which is available to the GeneRank vector. Next we propose two Krylov subspace methods for computing GeneRank. The idea stems from the facts that the GeneRank problem can be reformulated as a large-scale eigenvalue problem, the largest eigenvalue of the GeneRank matrix is 1; and the strategies derived from Golub and Greif (2006) and Jia and Elsner (2000). Numerical experiments illustrate that, when the GeneRank problem in question is very large, the new algorithms are appropriate choices.

However, there is still much research to be done. For instance, how should one choose the "optimal" parameter \( d \) based on the nature of the GeneRank problem in question? How can one make use of the special structure of the symmetric matrix \( W \), so that we can propose more efficient algorithms? Can we propose better mathematical models based on the GeneRank model for the analysis of microarray experiments? These are interesting topics and deserve further study.

ACKNOWLEDGMENTS

G.W. is supported by the National Science Foundation of China (grant 10901132), the Natural Science Foundation for Colleges and Universities in Jiangsu Province (grant 08KJB110012), the Qing-Lan Project of Jiangsu Province (grant QL200612), and the Natural Science Foundation of Xuzhou Normal University (grant 08XLY01).

Y.W. is supported by the National Natural Science Foundation of China (grant 10871051), Shanghai Education Committee (grant 08SG01), Shanghai Science & Technology Committee (grant 09DZ2272900), and Doctoral Program of the Ministry of Education under grant 20090071110003.

DISCLOSURE STATEMENT

No competing financial interests exist.
REFERENCES


Address correspondence to:
Dr. Yimin Wei
Department of Mathematics
Fudan University
Han-Dan Road, No. 220
Shanghai 200433, China

E-mail: yimin.wei@gmail.com