Electronic Transactions on Numerical Analysis.
Volume 40, pp. 311-320, 2013.

ISSN 1068-9613.

ETNA
Kent State University
http://etna.math.kent.edu

CHEBYSHEV ACCELERATION OF THE GENERANK ALGORITHM*

MICHELE BENZI† AND VERENA KUHLEMANN†

Abstract. The ranking of genes plays an important role in biomedical research. The GeneRank method of Morrison et al. [BMC Bioinformatics, 6:233 (2005)] ranks genes based on the results of microarray experiments combined with gene expression information, for example from gene annotations. The algorithm is a variant of the well known PageRank iteration, and can be formulated as the solution of a large, sparse linear system. Here we show that classical Chebyshev semi-iteration can considerably speed up the convergence of GeneRank, outperforming other acceleration schemes such as conjugate gradients.

Key words. GeneRank, computational genomics, Chebyshev semi-iteration, polynomials of best uniform approximation, conjugate gradients

AMS subject classifications. 65F10, 65F50; 9208, 92D20

1. Introduction.

Advances in biotechnology make it possible to collect a vast amount of genomic data. For example, using gene microarrays, it is now possible to probe a person's gene expression profile over the more than 30,000 genes of the human genome. Biomedical researchers try to link signals extracted from these gene microarray experiments to genetic factors underlying disease. One of the most important problems is to identify key genes that play a role in a particular disease.

A gene microarray consists of a large number of known DNA probe sequences that are put in distinct locations on a slide. Gene microarrays can be used for gene expression profiling. The DNA in a cell does not change, but certain derivatives of the DNA, the mRNA and tRNA, are produced as stimulation occurs through environmental conditions, and in response to treatments. For further details, see [1, 3].

A promising approach is to use bioinformatics methods that can analyze a variety of gene-related biological data and rank genes based on potential relevance to a disease; such methods can be invaluable in helping to prioritize genes for further biological study. We refer the reader to [12] and [18] for discussions of the many challenges arising in this important research area.

In 2005 Morrison et al. proposed a new model called GeneRank [12]. It is a modification of Google's PageRank algorithm [13, 11]. The model combines gene expression information with a network structure derived from gene annotations (gene ontologies) or expression profile correlations [12]. This makes errors in the microarray experiments less likely to influence the results than in methods which are based on expression levels alone. The resulting gene ranking algorithm shares many of the mathematical properties of PageRank; in particular, the ranking of genes can be reduced to the solution of a large linear system [18].

In two papers ([18] and [17]), Wu and coworkers have carried out a systematic matrix analysis of the GeneRank algorithm, together with a comparison of different iterative solvers for the resulting large sparse linear systems. In particular, they showed that the problem can be formulated as the solution of a symmetric positive definite linear system, and they obtained bounds on the extreme eigenvalues of the coefficient matrix when diagonal scaling (Jacobi preconditioning) is used. These bounds are independent of the size of the matrix, and they only depend on the value of a parameter used in the GeneRank model (analogous to the parameter used in PageRank). The numerical experiments in [17] indicate that the conjugate

* Received December 3, 2012. Accepted June 5, 2013. Published online on August 15, 2013. Recommended by R. Nabben. Work supported in part by National Science Foundation Grant DMS 1115692.

† Department of Mathematics and Computer Science, Emory University, Atlanta, GA 30322, USA ({benzi,vkuhlem}@mathcs.emory.edu).

ETNA
Kent State University
http://etna.math.kent.edu

gradient (CG) algorithm with diagonal scaling is the most effective solver, among all those tested, in terms of solution times.

In this note we further investigate properties of the GeneRank system and we consider a few alternative solution methods. In particular, we show that in conjunction with diagonal scaling, Chebyshev acceleration can significantly outperform CG in terms of solution times.

2. Definitions and auxiliary results. Connections between genes can be constructed via the Gene Ontology (GO) database.[1] Let the set $G = \{g_1, g_2, \ldots, g_n\}$ consist of n genes in a microarray. Two genes g_i and g_j are connected if they share an annotation in GO. Similar to PageRank, the idea of GeneRank is that a gene is significant if it is connected to many highly ranked genes. In contrast to PakeRank, the connections are not directed. Thus, instead of a nonsymmetric hyperlink matrix, GeneRank considers the symmetric adjacency matrix W of the gene network. W is given by

$$w_{ij} = \begin{cases} 1 & \text{if } g_i \text{ and } g_j \ (i \neq j) \text{ share an annotation in GO,} \\ 0 & \text{otherwise.} \end{cases}$$

Note that W is unweighted, while the hyperlink matrix in PageRank is weighted so that each row sums up to one. A diagonal matrix D is constructed to provide such a scaling. Since a gene might not be connected to any of the other genes, W may have zero rows. We let $\deg_i$ denote the degree (number of immediate neighbors) of gene i in the underlying graph; this is just the sum of the entries in the ith row (or column) of W, that is,

$$\deg_i = \sum_{j=1}^{n} w_{ij} = \sum_{j=1}^{n} w_{ji}.$$

The diagonal matrix D is defined by $D = \operatorname{diag}(d_1, \ldots, d_n)$, where

$$d_i = \begin{cases} \deg_i & \text{if } \deg_i > 0, \\ 1 & \text{otherwise.} \end{cases}$$

Note that D is nonsingular and nonnegative by construction. Now, $(D^{-1}W)^T$ corresponds to the weighted hyperlink matrix in PageRank. In the case of GeneRank we do not need to modify the matrix further, since irreducibility is not needed.

So far only the connections between genes are considered, but not the results from the gene microarray experiment. Let $\boldsymbol{ex} = [ex_1, ex_2, \ldots, ex_n]^T$ be a vector that is obtained from such an experiment. The entry $ex_i \geq 0$ is the absolute value of the expression change of gene g_i. In addition, a *damping factor* α with $0 \leq \alpha \leq 1$ that controls the relative weighting of expression change vs. connectivity is introduced. Then, GeneRank can be written as a large linear system:

$$(I - \alpha W D^{-1})\boldsymbol{x} = (1 - \alpha)\boldsymbol{ex}, \tag{2.1}$$

where I denotes the $n \times n$ identity matrix. The solution vector $\boldsymbol{x}$ is called *GeneRank vector*, and its entries provide information about the significance of a gene. Note that for $\alpha = 0$ genes are ranked solely on the basis of the microarray experiment. If $\alpha = 1$, then a ranking is constructed using only the connectivity information, and the results from the microarray experiments are ignored. The problem of how to choose the value of α will not be discussed here, but typically values in the interval $[0.5, 1)$ are used.

[1] http://geneontology.org

ETNA
Kent State University
http://etna.math.kent.edu

3. Symmetric formulation of GeneRank. The matrix W is symmetric, but $I-\alpha WD^{-1}$ is not. Thus, for the solution of the linear system (2.1), nonsymmetric methods have to be used. The symmetry of W cannot be exploited. In [17], however, Wu et al. recognized that the GeneRank problem can be rewritten as a symmetric linear system. The main idea is simply to write the linear system (2.1) as

$$(D-\alpha W)D^{-1}\boldsymbol{x} = (1-\alpha)\boldsymbol{e}\boldsymbol{x}, \tag{3.1}$$

or equivalently, as

$$(D-\alpha W)\hat{\boldsymbol{x}} = (1-\alpha)\boldsymbol{e}\boldsymbol{x}, \tag{3.2}$$

with $\hat{\boldsymbol{x}} = D^{-1}\boldsymbol{x}$. The matrix $D-\alpha W$ is symmetric. With this modification, methods that are suitable for symmetric systems can be used for the GeneRank problem. In the next section we will see that the symmetric GeneRank matrix enjoys additional useful properties.

4. Properties of the symmetric GeneRank matrix. In [17], Wu et al. also showed that the matrix $D-\alpha W$ has some nice properties besides symmetry. First of all, they showed that $D-\alpha W$ is positive definite for $0 \le \alpha < 1$. Thus, the conjugate gradient (CG) method [8] can be used to solve the linear system (3.2). We note in passing that for $\alpha = 1$ this matrix reduces to the (combinatorial) graph Laplacian $L = D - W$ of the network, and is positive semidefinite (singular). If the network is connected, the null space of L is one-dimensional and is spanned by the constant vector with all entries equal to 1.

Wu et al. investigated the effectiveness of the Jacobi preconditioner (symmetric diagonal scaling) on $D-\alpha W$. In that case, the preconditioned matrix is given by

$$D^{-1/2}(D-\alpha W)D^{-1/2} = I - \alpha D^{-1/2}WD^{-1/2}.$$

Thus, the preconditioned linear system reads

$$(I-\alpha D^{-1/2}WD^{-1/2})\bar{\boldsymbol{x}} = (1-\alpha)D^{-1/2}\boldsymbol{e}\boldsymbol{x}, \tag{4.1}$$

with $\bar{\boldsymbol{x}} = D^{1/2}\hat{\boldsymbol{x}} = D^{1/2}(D^{-1}\boldsymbol{x}) = D^{-1/2}\boldsymbol{x}$.

Since $D^{-1/2}WD^{-1/2}$ is doubly stochastic, the eigenvalues of the preconditioned matrix satisfy:

$$\begin{aligned} \lambda_{\max}(I-\alpha D^{-1/2}WD^{-1/2}) &\le 1+\alpha\,, \\ \lambda_{\min}(I-\alpha D^{-1/2}WD^{-1/2}) &= 1-\alpha\,. \end{aligned} \tag{4.2}$$

The range of eigenvalues increases as α increases from 0 to 1. Moreover, the matrix becomes increasingly ill-conditioned. Thus, the rate of convergence of CG can be expected to decrease as α increases. Using Gershgorin's Circle Theorem [16], we can also bound the eigenvalues of $D-\alpha W$ as follows:

$$\begin{aligned} \lambda_{\max}(D-\alpha W) &\le (1+\alpha)\max_{1\le i\le n}\{d_i\}\,, \\ \lambda_{\min}(D-\alpha W) &\ge (1-\alpha)\min_{1\le i\le n}\{d_i\}\,. \end{aligned} \tag{4.3}$$

Next, we show that both $D-\alpha W$ and $I-\alpha D^{-1/2}WD^{-1/2}$ are Stieltjes matrices (that is, symmetric nonsingular M-matrices).

PROPOSITION 4.1.

1. *$D-\alpha W$ is a nonsingular M-matrix, for $0 \le \alpha < 1$.*

ETNA
Kent State University
http://etna.math.kent.edu

2. *$I - \alpha D^{-1/2} W D^{-1/2}$ is a nonsingular M-matrix, for $0 \leq \alpha < 1$.*

Proof. We can rewrite the first matrix as $D - \alpha W = \Delta I - (\tilde{D} + \alpha W)$, where Δ is the maximum degree of the underlying graph of W. That is, $\Delta = \max\{d_i \mid i = 1, \ldots, n\}$. The matrix $\tilde{D}$ is diagonal with $\tilde{d}_i = \Delta - d_i$ on the diagonal. Thus, $\tilde{D}$ is nonnegative, and from $\alpha > 0$ and $W \geq 0$ it follows that $\tilde{D} + \alpha W \geq 0$. Note that $\Delta I - (\tilde{D} + \alpha W)$ is a nonsingular M-matrix if $\rho(\tilde{D} + \alpha W) < \Delta$. The spectral radius is bounded above by the 1-norm. Thus, $\rho(\tilde{D} + \alpha W) \leq ||\tilde{D} + \alpha W||_1 = \max_{d_i}\{\Delta - d_i + \alpha d_i\} < \Delta$, for $0 < \alpha < 1$.

With regard to the second matrix, the conclusion follows from the fact that $D^{-1/2} A D^{-1/2}$ is a nonsingular M-matrix if A is a nonsingular M-matrix and D is a positive diagonal matrix. □

There are several important consequences of the property just shown. First of all, the right-hand side vector $e\boldsymbol{x}$ is nonnegative; hence, the GeneRank vector $\boldsymbol{x}$ is also guaranteed to be nonnegative, since the inverse of a nonsingular M-matrix is nonnegative. Moreover, whenever the underlying graph is connected the GeneRank matrices $D - \alpha W$ or $I - \alpha D^{-1/2} W D^{-1/2}$ are irreducible; therefore, they have a positive inverse, thus making the ranking vector $\boldsymbol{x}$ strictly positive, as it should be if the vector is to be used for ranking purposes.

Additionally, the M-matrix property ensures that various classical iterations based on matrix splittings are guaranteed to converge, including the Jacobi and Gauss–Seidel methods and their block variants [16], as well as Schwarz-type methods. Moreover, the existence and stability of various preconditioners, like incomplete factorizations, is guaranteed.

5. Methods tested. In [17], Wu and coworkers successfully employed the Jacobi preconditioner together with CG for the solution of the linear system (3.2). They compared the method with the original GeneRank scheme (essentially a power iteration), Jacobi's method, and a (modified) Arnoldi algorithm. CG preconditioned with Jacobi was faster for every example tested. The rate of convergence was found to be essentially independent of the problem size n, consistent with the uniform bounds on the eigenvalues of the preconditioned matrices. The number of iterations, on the other hand, increases as α approaches 1.

In an attempt to improve on the results of Wu et al., we tested a number of other methods. First of all, we tried the sparse direct solver in Matlab ("backslash"). This is a sparse implementation of the Cholesky algorithm which uses an approximate minimum degree ordering; see [4]. An obvious advantage of the direct approach is that its cost is independent of α. However, we found this approach to be extremely time-consuming due to the enormous fill-in in the factors; see Section 7 below. Preconditioners based on incomplete Cholesky factorization or SSOR (symmetric SOR) were also found to be inefficient in comparison to CG with a simple diagonal preconditioner. Additional experiments were performed with additive Schwarz-type preconditioners with overlapping blocks [2, 10]. For several of the tested examples, we found that these preconditioners achieve fast convergence independent of the parameter α; unfortunately, however, the additional complexity of these preconditioners makes them not competitive with simple diagonal scaling [10].

A close look at the eigenvalues of the diagonally preconditioned GeneRank matrices reveal that they are more or less uniformly distributed in the spectral interval $[\lambda_{\min}, \lambda_{\max}]$, with no clustering within the spectra. As is well known, such a spectral distribution is essentially the worst possible for CG. This suggests investigating the performance of other methods, such as methods based on (shifted and scaled) Chebyshev polynomials [5, 16], or methods based on polynomials of best approximation [9].

An advantage of these techniques is that the cost per iteration is very low. Also, they do not require inner products, which makes them attractive on some parallel architectures. A potential disadvantage is that they require bounds on the eigenvalues. But, as we saw earlier,

ETNA
Kent State University
http://etna.math.kent.edu

Algorithm 1 Chebyshev iteration

$d = (l_{\max} + l_{\min})/2$, $c = (l_{\max} - l_{\min})/2$
$x = x_0$, $r = b - Ax$
for $i = 1, 2, \ldots, m$ **do**
  $z = M^{-1}r$
  if $i = 1$ **then**
    $p = z$
    $\alpha = 2/d$
  else
    $\beta = (c \cdot \alpha/2)^2$
    $\alpha = 1/(d - \beta)$
    $p = z + \beta \cdot p$
  end if
  $x = x + \alpha \cdot p$
  $r = b - Ax$
  if $norm(r) < tol$ **then**
    break;
  end if
end for

here we do have bounds on the largest and smallest eigenvalue of $I - \alpha D^{-1/2} W D^{-1/2}$.

The Chebyshev (semi-)iteration is a classical method for solving linear systems based on the properties of Chebyshev polynomials. It can be regarded as a polynomial scheme for accelerating the convergence of a standard linear stationary iteration

$$x^{(k+1)} = x^{(k)} + M^{-1}(b - Ax^{(k)}), \quad k = 0, 1, \ldots \tag{5.1}$$

for solving a linear system $Ax = b$, with M nonsingular. If $I - M^{-1}A$ is similar to a symmetric matrix with eigenvalues lying in an interval $[l_{\min}, l_{\max}]$, the Chebyshev acceleration of the stationary method (5.1) can be written as

$$y^{(k+1)} = \frac{\omega_{k+1}}{2 - (l_{\min} + l_{\max})} \left\{ 2M^{-1}(b - Ay^{(k)}) + [2 - (l_{\min} + l_{\max})](y^{(k)} - y^{(k-1)}) \right\} + y^{(k-1)}, \quad k = 1, 2, \ldots,$$

with $y^{(0)} = x^{(0)}$, $y^{(1)} = x^{(1)}$ and

$$\omega_{k+1} = \frac{1}{1 - \frac{\omega_k^2}{4w^2}}, \quad \omega_2 = \frac{2w^2}{2w^2 - 1}, \quad \omega_1 = 1, \quad w = \frac{2 - (l_{\min} + l_{\max})}{l_{\max} - l_{\min}};$$

see, e.g., [5, pages 514–516]. This acceleration can be employed, for instance, when A and M are symmetric positive definite (SPD). In particular, if A is symmetric positive definite and $M = D = \operatorname{diag}(A)$, the diagonally preconditioned Chebyshev iteration can be interpreted as a polynomial acceleration scheme applied to Jacobi's method.

An algorithmic description of the Chebyshev iteration is given in Algorithm 1 (using a slightly different noation). The input parameters $l_{\min}$ and $l_{\max}$ are bounds on the smallest and largest eigenvalue of the preconditioned matrix. In our case, $l_{\min} = 1 - \alpha$ and $l_{\max} = 1 + \alpha$. The algorithm is applied to $A = D - \alpha W$, with the right-hand side $b = \boldsymbol{ex}$. Also, M denotes the preconditioner; in our case, $M = D$.

It is known [14] that when $l_{\min} = \lambda_{\min}(A)$ and $l_{\max} = \lambda_{\max}(A)$, the shifted and scaled Chebyshev polynomial of degree $k-1$ minimizes the condition number of the preconditioned

ETNA
Kent State University
http://etna.math.kent.edu

Algorithm 2 Polynomial of best uniform approximation method

$\mu_0 = 1/l_{\max}$, $\mu_1 = 1/l_{\min}$
$a = (l_{\max} + l_{\min})/(l_{\max} - l_{\min})$
$\delta_1 = (a - \sqrt{a^2 - 1})^2$
$\delta_2 = \frac{4 \cdot \mu_0 \mu_1}{(\sqrt{\mu_0} + \sqrt{\mu_1})^2}$;
$x = x_0$, $r = b - Ax$
$v_1 = 0.5 \cdot (\mu_0 + \mu_1) \cdot r$
$v_2 = 0.5 \cdot (\sqrt{\mu_0} + \sqrt{\mu_1})^2 \cdot r - \mu_0 \mu_1 \cdot Ar$
for $i = 2, \ldots, m$ **do**
  $r = r - Av_2$
  if $norm(r) < tol$ **then**
    $x = x + v_2$
    break;
  end if
  $v_3 = v_2 + \delta_1 \cdot (v_2 - v_1) + \delta_2 \cdot r$
  $v_1 = v_2$, $v_2 = v_3$
end for

matrix $p_k(A)A$ over all polynomials $p_k(\lambda)$ of degree not exceeding $k-1$; when diagonal scaling is used, of course, the same holds with $D^{-1}A$ in place of A. In principle, however, there may be other polynomials that result in faster convergence. Similar to [9], we consider an alternative approach based on the use of polynomials of best uniform approximation to the function $1/\lambda$. These are the polynomials of prescribed degree k such that the approximation error, $\max |p_k(\lambda) - \frac{1}{\lambda}|$, is minimized over all polynomials of degree not exceeding k, the maximum being taken over an interval $[l_{\min}, l_{\max}]$ containing the spectrum of $D^{-1}A$. These polynomials, which were found by Chebyshev in 1887, can be generated by a three-term recurrence, as discussed in [9]. Here we use again $l_{\min} = 1-\alpha$ and $l_{\max} = 1+\alpha$ as endpoints, and the algorithm is applied to the preconditioned matrix $A = I - \alpha D^{-1/2} W D^{-1/2}$. The details can be seen in Algorithm 2. We mention that we tried using the "true" eigenvalue $\lambda_{\max}$, both with Chebyshev and with the polynomials of best approximation, but the results were essentially unchanged. Hence, little (if anything) is lost by using the freely available estimate $\lambda_{\max} \approx 1 + \alpha$.

6. Description of test problems. As in [17], we use two different types of test data (real and synthetic) for our experiment. The first matrix is a SNPa adjacency matrix (single-nucleotide polymorphism matrix). This matrix has $n = 152,520$ rows and columns, and is very sparse with only $639,248$ nonzeros. The sparsity pattern of the SNPa matrix can be seen in Figure 6.1. The degree distribution in the underlying graph ranges from 1 to 40, and is highly irregular.

The second type is a class of matrices from a range-dependent random graph model called RENGA. Two vertices i and j are connected with probability $\beta\lambda^{|j-i|-1}$, where $0 < \lambda < 1$ and $\beta > 0$ are given parameters. These networks capture the connectivity structure seen in proteome interaction data [7, 6]. MATLAB code for generating these and other networks is available as part of the CONTEST package [15]. In our experiments we set $\lambda = 0.9$ and $\beta = 1$, the default values in RENGA. Note that with $\beta = 1$ node i is connected to node $i+1$ for $i = 1, \ldots, n-1$.

7. Numerical experiments. The implementation was done in Matlab 7.8.0 on a 2.3 GHz Intel Core i7 Processor with 4GB main memory. We compare the Chebyshev method and the method based on polynomials of best approximation ("Poly") with the conjugate gradient method and a Jacobi-preconditioned conjugate gradient.

ETNA
Kent State University
http://etna.math.kent.edu

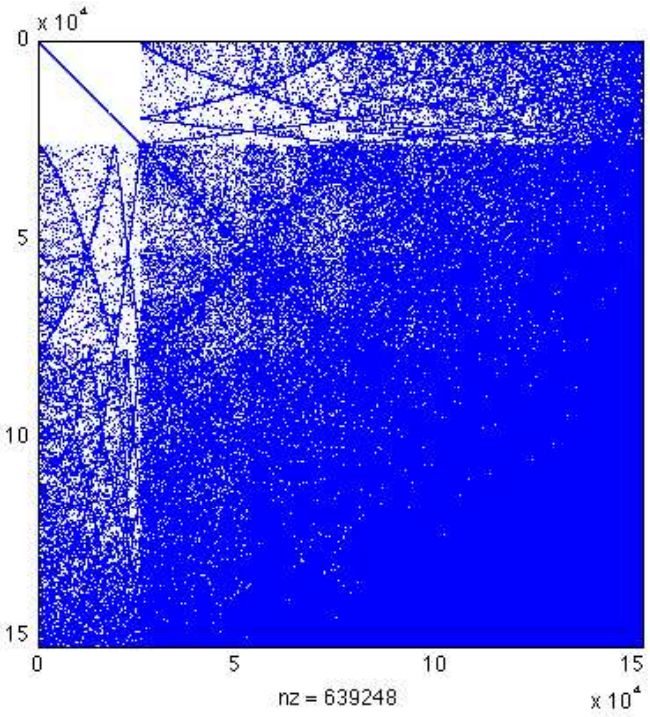

FIG. 6.1. *Nonzero pattern of the SNPa matrix.*

TABLE 7.1

Results for the SNPa matrix. The matrix has $n = 152,520$ rows and columns. The tolerance used is 10^{-10}. Here $\boldsymbol{ex} = (1/n)\boldsymbol{e}$, where $\boldsymbol{e}$ is the vector of all ones. The number of iterations and the CPU time in seconds (in brackets) are given.

α	0.50	0.75	0.80	0.99
CG	86 (1.06)	116 (1.33)	129 (1.46)	470 (6.11)
PCG	17 (0.26)	27 (0.36)	30 (0.42)	91 (1.28)
Poly	17 (0.11)	28 (0.17)	32 (0.20)	149 (0.89)
Chebyshev	17 (0.10)	28 (0.16)	31 (0.18)	130 (0.73)

We use the same stopping criteria for each of the methods tested, based on the 1-norm of the residual. That is, we stop iterating as soon as $\|r\|_1 < tol$. The initial guess is the zero vector. We use two different choices for $\boldsymbol{ex}$: $\boldsymbol{ex} = (1/n)\boldsymbol{e}$, where $\boldsymbol{e}$ is the vector of all ones, and $\boldsymbol{ex} = \boldsymbol{p}$, where p is a randomly chosen probability vector—that is, a random vector with entries in $(0, 1)$, normalized to have $\|\boldsymbol{p}\|_1 = 1$. For each adjacency matrix, we use four different values of α to form the corresponding GeneRank matrices $D - \alpha W$: $\alpha = 0.5\,, 0.75\,, 0.80\,, 0.99$.

The results for the SNPa matrix are given in Tables 7.1 and 7.2, and the results for the RENGA matrices (with $n = 100,000$ and $n = 500,000$) are given in Tables 7.3 and 7.4. Note that, as stated earlier, the rate of convergence depends only on α and not on n.

The results for the SNPa matrix show that diagonal scaling dramatically accelerates the convergence of CG; also, this is the fastest method among those tested in terms of number

ETNA
Kent State University
http://etna.math.kent.edu

TABLE 7.2
Results for the SNPa matrix. The matrix has $n = 152,520$ rows and columns. The tolerance used is 10^{-10}. Here $\boldsymbol{ex} = \boldsymbol{p}$, where $\boldsymbol{p}$ is a random probability vector. The number of iterations and the CPU time in seconds (in brackets) are given.

α	0.50	0.75	0.80	0.99
CG	87 (1.07)	118 (1.34)	130 (1.49)	484 (5.54)
PCG	17 (0.23)	27 (0.36)	30 (0.40)	90 (1.19)
Poly	17 (0.11)	28 (0.17)	32 (0.20)	152 (0.91)
Chebyshev	17 (0.10)	28 (0.16)	31 (0.18)	131 (0.73)

TABLE 7.3
Results for the RENGA matrices. The tolerance used is 10^{-10}. Here $\boldsymbol{ex} = (1/n)\boldsymbol{e}$, where $\boldsymbol{e}$ is the vector of all ones. The number of iterations and the CPU time in seconds (in brackets) are given.

α	0.50	0.75	0.80	0.99
$n = 100,000$				
CG	20 (0.21)	27 (0.28)	30 (0.30)	105 (1.07)
PCG	13 (0.15)	21 (0.24)	23 (0.27)	92 (1.06)
Poly	16 (0.12)	26 (0.19)	29 (0.20)	131 (0.86)
Chebyshev	17 (0.11)	27 (0.18)	30 (0.20)	125 (0.81)
$n = 500,000$				
CG	22 (1.38)	28 (1.77)	30 (1.91)	108 (6.87)
PCG	13 (0.92)	21 (1.48)	23 (1.62)	92 (6.56)
Poly	16 (0.67)	26 (1.04)	29 (1.15)	131 (4.92)
Chebyshev	17 (0.63)	27 (0.99)	30 (1.11)	125 (4.49)

of iterations. However, the method based on the polynomials of best approximation, while often requiring more iterations, is faster in terms of solution time, sometimes more than twice as fast, and Chebyshev iteration is even faster. Similar conclusions apply to the RENGA matrices, except that now the effect of diagonal scaling on CG is less pronounced, probably due to the fact that the distribution of nonzeros in these matrices is much more regular than for the SNPa example, thus leading to matrices $D - \alpha W$ which are better conditioned. Here again we find that Chebyshev outperforms the competition, albeit by a smaller margin than in the SNPa examples.

Also note that in all cases, the results are essentially independent of the right-hand side used.

Finally, use of a direct solver (sparse Cholesky in MATLAB) is not competitive for these problems, due to tremendous fill-in, even with the best available fill-reducing orderings. For example, the Cholesky factor for the SNPa matrix contains over $200,000,000$ non-zeros.

8. Conclusions. We investigated several methods for the solution of the linear system arising from the gene ranking problem. Good results were obtained with (diagonally scaled) Chebyshev iteration. While the number of iterations is higher than for CG with the same scaling, the cost per iteration is much lower and leads to faster convergence in terms of CPU time. We note that Chebyshev iteration is more desirable in a parallel setting, as it avoids computing dot products.

ETNA
Kent State University
http://etna.math.kent.edu

TABLE 7.4
Results for the RENGA matrices. The tolerance used is 10^{-10}. Here $\boldsymbol{ex} = \boldsymbol{p}$, where $\boldsymbol{p}$ is a random probability vector. The number of iterations and the CPU time in seconds (in brackets) are given.

α	0.50	0.75	0.80	0.99
		$n = 100,000$		
CG	23 (0.24)	28 (0.29)	31 (0.31)	116 (1.19)
PCG	14 (0.16)	22 (0.25)	25 (0.29)	98 (1.13)
Poly	17 (0.13)	26 (0.19)	30 (0.21)	141 (0.95)
Chebyshev	17 (0.11)	27 (0.18)	30 (0.20)	125 (0.81)
		$n = 500,000$		
CG	23 (1.45)	29 (1.86)	32 (2.03)	118 (7.50)
PCG	14 (0.99)	22 (1.55)	25 (1.77)	99 (7.03)
Poly	17 (0.72)	26 (1.03)	30 (1.18)	141 (5.23)
Chebyshev	17 (0.62)	27 (1.02)	30 (1.09)	125 (4.48)

The question remains open whether it is possible to develop preconditioners for the GeneRank problem which result in converge rates independent of α and are competitive with simple diagonal scaling.

Acknowledgement. We would like to thank Prof. Yimin Wei of Fudan University for providing the SNPa data to us.

REFERENCES

[1] D. E. BASSETT, M. B. EISEN, AND M. S. BOGUSKI, *Gene expression informatics—it's all in your mine*, Nat. Genet., 21 (1999), pp. 51–55.
[2] M. BENZI AND V. KUHLEMANN, *Restricted additive Schwarz methods for Markov chains*, Numer. Linear Algebra Appl., 18 (2011), pp. 1011–1029.
[3] P. O. BROWN AND D. BOTSTEIN, *Exploring the new world of the genome with DNA microarrays*, Nat. Genet., 21 (1999), pp. 33–37.
[4] T. A. DAVIS, *Direct Methods for Sparse Linear Systems*, SIAM, Philadelphia, 2006.
[5] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 3rd edition, Johns Hopkins University Press, Baltimore, 1996.
[6] P. GRINDROD, *Range-dependent random graphs and their application to modeling large small-world proteome datasets*, Phys. Rev. E (3), 66 (2002), 066702 (7 pages).
[7] ———, *Modeling proteome networks with range-dependent graphs*, Amer. J. Pharmacogenomics, 3 (2003), pp. 1–4.
[8] M. R. HESTENES AND E. STIEFEL, *Methods of conjugate gradients for solving linear systems*, J. Research Nat. Bur. Standards, 49 (1952), pp. 409–436.
[9] J. KRAUS, P. VASSILEVSKI, AND L. ZIKATANOV, *Polynomial of best uniform approximation to $1/x$ and smoothing in two-level methods*, Comput. Methods Appl. Math., 12 (2012), pp. 448–468.
[10] V. KUHLEMANN, *Iterative methods and partitioning techniques for large sparse problems in network analysis*, Ph.D. Thesis, Department of Mathematics and Computer Science, Emory University, 2012.
[11] A. N. LANGVILLE AND C. D. MEYER, *Google's PageRank and Beyond—The Science of Search Engine Rankings*, Princeton University Press, Princeton, 2006.
[12] J. L. MORRISON, R. BREITLING, D. J. HIGHAM, AND D. R. GILBERT, *GeneRank: Using search engine technology for the analysis of microarray experiments*, BMC Bioinformatics, 6:233 (2005).
[13] L. PAGE, S. BRIN, R. MOTWANI, AND T. WINOGRAD, *The PageRank citation ranking: bringing order to the Web*, Technical Report, Stanford InfoLab, Stanford University, 1998. Available at http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf.
[14] Y. SAAD, *Iterative Methods for Sparse Linear Systems*, 2nd edition, SIAM, Philadelphia, 2003.
[15] A. TAYLOR AND D. J. HIGHAM, *CONTEST: A controllable test matrix toolbox for MATLAB*, ACM Trans. Math. Software, 35 (2009), pp. 26:1–26:17.

ETNA
Kent State University
http://etna.math.kent.edu

[16] R. S. VARGA, *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs, 1962.
[17] G. WU, W. XU, Y. ZHANG, AND Y. WEI, *A preconditioned conjugate gradient algorithm for GeneRank with application to microarray data mining*, Data Min. Knowl. Discov., 26 (2013), pp. 27–56.
[18] G. WU, Y. ZHANG, AND Y. WEI, *Krylov subspace algorithms for computing GeneRank for the analysis of microarray data mining*, J. Comput. Biol., 17 (2010), pp. 631–646.