

Automated Gene Classification using Nonnegative Matrix Factorization on Biomedical Literature

Kevin Heinrich

PhD Dissertation Defense
Department of Computer Science, UTK

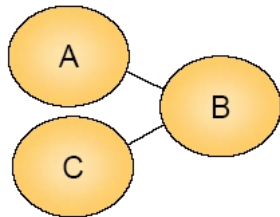
March 19, 2007

What Is The Problem?

- Understanding functional gene relationships requires expert knowledge.
- Gene sequence analysis does not necessarily imply function.
- Gene structure analysis is difficult.
- Issue of scale.
 - Biologists know a small subset of genes.
 - Thousands of genes.
- Time & Money.

Defining Functional Gene Relationships

- Direct Relationships.
 - Known gene relationships (e.g. A-B).
 - Based on term co-occurrence.¹
- Indirect Relationships.
 - Unknown gene relationships (e.g. A-C).
 - Based on *semantic structure*.



¹Jenssen et al., *Nature Genetics*, 28:21, 2001.

- Gene information is compiled in human-curated databases.
 - Medical Literature, Analysis, and Retrieval System Online (MEDLINE)
 - EntrezGene (LocusLink)
 - Medical Subject Heading (MeSH)
 - Gene Ontology (GO)
- *Gene documents* are formed by taking titles and abstracts from MEDLINE citations cross-referenced in the Mouse, Rat, and Human EntrezGene entries for that gene.
- Examines literature (phenotype) instead of genotype.
- Can be used as a guide for future gene exploration.

- Gene documents are parsed into *tokens*.
- Tokens are assigned a weight, w_{ij} , of i^{th} token in j^{th} document.
- An $m \times n$ term-by-document matrix, A , is created.

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

- Genes are m -dimensional vectors.
- Tokens are n -dimensional vectors.

Term-by-Document Matrix

	d_1	d_2	d_3	\dots	d_n
t_1	w_{11}	w_{12}	w_{13}		w_{1n}
t_2	w_{21}	w_{22}	w_{23}		w_{2n}
t_3	w_{31}	w_{32}	w_{33}		w_{3n}
t_4	w_{41}	w_{42}	w_{43}		w_{4n}
\vdots				\ddots	
t_m	w_{m1}	w_{m2}	w_{m3}		w_{mn}

Typically, a term-document matrix is sparse and unstructured.

- Term weights are the product of a local, global component, and document normalization factor.

$$w_{ij} = l_{ij}g_i d_j$$

- The log-entropy weighting scheme is used where

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

$$g_i = 1 + \left(\frac{\sum_j (p_{ij} \log_2 p_{ij})}{\log_2 n} \right), p_{ij} = \frac{f_{ij}}{\sum_j f_{ij}}$$

Latent Semantic Indexing (LSI)

LSI performs a truncated singular value decomposition (SVD) on M into three factor matrices

$$A = U\Sigma V^T$$

- U is the $m \times r$ matrix of eigenvectors of AA^T
- V^T is the $r \times n$ matrix of eigenvectors of $A^T A$
- Σ is the $r \times r$ diagonal matrix of the r nonnegative singular values of A
- r is the rank of A

- A rank- k approximation is generated by truncating the first k columns of each matrix, i.e., $A_k = U_k \Sigma_k V_k^T$
- A_k is the closest of all rank- k approximations, i.e., $\|A - A_k\|_F \leq \|A - B\|$ for any rank- k matrix B

- Document-to-Document Similarity

$$A_k^T A_k = (V_k \Sigma_k) (V_k \Sigma_k)^T$$

- Term-to-Term Similarity

$$A_k A_k^T = (U_k \Sigma_k) (U_k \Sigma_k)^T$$

- Document-to-Term Similarity

$$A_k = U_k \Sigma_k V_k^T$$

Advantages of LSI

- A is sparse, factor matrices are dense. This causes improved recall for concept-based matching.
- Scaled document vectors can be computed once and stored for quick retrieval.
- Components of factor matrices represent concepts.
- Decreasing number of dimensions compares documents in a broader sense and achieves better compression.
- Similar word usage patterns get mapped to same geometric space.
- Genes are compared at a concept level rather than a simple term co-occurrence level resulting in vocabulary independent comparisons.

Problem:

- Biologists are familiar with interpreting trees.
- LSI produces ranked lists of related terms/documents.

Solution:

- Generate pairwise distance data, i.e., $1 - \cos \theta_{ij}$
- Apply distance-based tree-building algorithm
 - Fitch - $O(n^4)$
 - NJ - $O(n^3)$
 - FastME - $O(n^2)$

Defining Functional Gene Relationships on Test Data

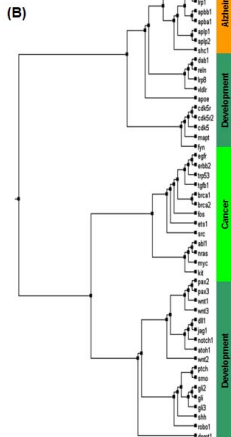
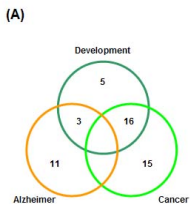


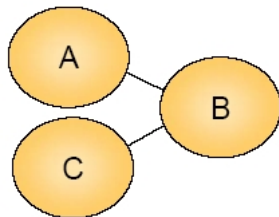
Figure 6. Gene neighbors deduced from the literature. (A) The overlap in genes involved in Development, Alzheimer's Disease and Cancer. (B) Hierarchical tree of genes using the PHYLIP algorithm and a distance matrix derived from the cosine of the vector angles between gene-documents. The genes cluster into 2 major groups: I) Development/Alzheimer, II) Development/Cancer.

“Problems” with LSI

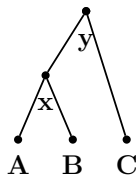
- Initial term weights are nonnegative; SVD introduces negative components.
- Dimensions of factored space do not have an immediate interpretation.
- Want advantages of factored/reduced dimension space, but want to interpret dimensions for clustering/labeling trees.
- Issue of scale—understand small collections better rather than huge collections.

Defining Functional Gene Relationships

- Direct Relationships.
 - Known gene relationships (e.g. A-B).
 - Based on term co-occurrence.²
- Indirect Relationships.
 - Unknown gene relationships (e.g. A-C).
 - Based on *semantic structure*.



- Label Relationships (e.g. x & y).



²Jenssen et al., *Nature Genetics*, 28:21, 2001.

NMF Problem Definition

Given nonnegative V , find W and H such that

$$V \approx WH$$

- $W, H \geq 0$
- W has size $m \times k$
- H has size $k \times n$

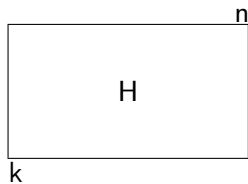
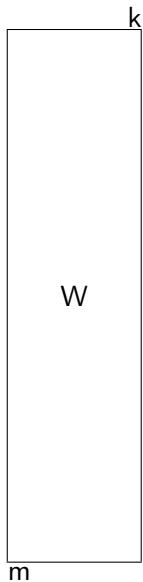
Given nonnegative V , find W and H such that

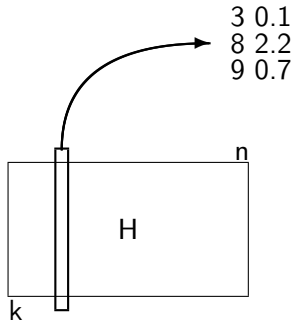
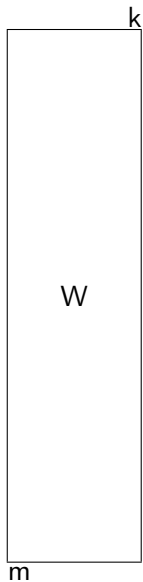
$$V \approx WH$$

- $W, H \geq 0$
- W has size $m \times k$
- H has size $k \times n$
- W and H are not unique.
i.e., $WDD^{-1}H$ for any invertible nonnegative D

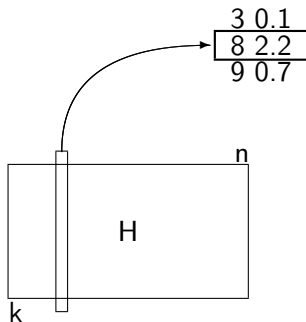
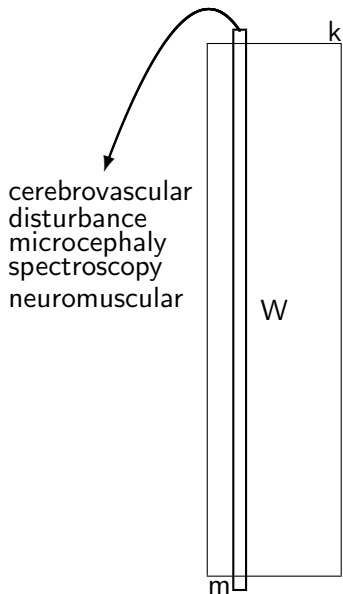
$$V \approx WH$$

- Columns of W are k “feature” or “basis” vectors; represent semantic concepts.
- Columns of H are linear combinations of feature vectors to approximate corresponding column in V .
- Choice of k determines accuracy and quality of basis vectors.
- Ultimately produces a “parts-based” representation of the original space.





3 0.1
8 2.2
9 0.7



Euclidean Distance (Cost Function)

$$E(W, H) = \|V - WH\|_F^2 = \sum_{i,j} \left(V_{ij} - (WH)_{ij} \right)^2$$

- Minimize $E(W, H)$ subject to $W, H \geq 0$.
- $E(W, H) \geq 0$.
- $E(W, H) = 0$ if and only if $V = WH$.
- $\|V - WH\|$ convex in W or H separately, not both simultaneously.
- No guarantee to find global minima.

Since NMF is an iterative algorithm, W and H must be initialized.

- Random positive entries.
- Structured initialization typically speeds convergence.
 - Run k -means on V .
 - Choose representative vector from each cluster to form W and H .

Most methods do not provide static starting point.

Non-Negative Double SVD

- NNDSVD is one way to provide a static starting point.³
- Observe $A_k = \sum_{j=1}^k \sigma_j u_j v_j^T$, i.e. sum of rank-1 matrices
- Foreach j
 - Compute $C = u_j v_j^T$
 - Set to 0 all negative elements of C
 - Compute maximum singular triplet of C , i.e., $[\hat{u}, \hat{\sigma}, \hat{v}]$
 - Set j th column of W to \hat{u} and j th row of H to $\sigma_j \hat{v}$
- Resulting W and H are influenced by SVD.

³Boutsidis & Gallopoulos, Tech Report, 2005

Zero elements remain “locked” during MM update.

- NNDSVDz keeps zero elements.
- NNDSVDe assigns $\epsilon = 10^{-9}$ to zero elements.
- NNDSVDa assigns average value of A to zero elements.

Update rules should

- decrease the approximation.
- maintain nonnegativity constraints.
- maintain other constraints imposed by the application (smoothness/sparsity).

Multiplicative Method (MM)

$$H_{cj} \leftarrow H_{cj} \frac{(W^T V)_{cj}}{(W^T W H)_{cj} + \epsilon}$$

$$W_{ic} \leftarrow W_{ic} \frac{(V H^T)_{ic}}{(W H H^T)_{ic} + \epsilon}$$

- ϵ ensures numerical stability.
- Lee and Seung proved MM non-increasing under Euclidean cost function.
- Most implementations update H and W “simultaneously.”

$$\|V - WH\|_F^2 + \alpha J_1(W) + \beta J_2(H)$$

α and β are parameters to control level of additional constraints.

Smoothing Update Rules

For example, set $J_2(H) = \|H\|_F^2$ to enforce smoothness on H to try to force uniqueness on W .⁴

$$H_{cj} \leftarrow H_{cj} \frac{(W^T V)_{cj} - \beta H_{cj}}{(W^T W H)_{cj} + \epsilon}$$
$$W_{ic} \leftarrow W_{ic} \frac{(V H^T)_{ic} - \alpha W_{ic}}{(W H H^T)_{ic} + \epsilon}$$

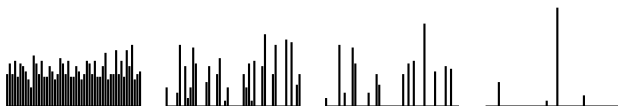
⁴Piper et. al., AMOS, 2004

Hoyer defined sparsity as

$$\text{sparseness}(\vec{x}) = \frac{\sqrt{n} - \frac{\sum |x_i|}{\sqrt{\sum x_i^2}}}{\sqrt{n} - 1}$$

- Zero if and only if all components have same magnitude.
- One if and only if x contains one nonzero component.

Visual Interpretation



- Sparsity constraints are explicitly set and built into update algorithm.
- Can be applied to W , H , or both.
- Dominant features are (hopefully) preserved.

Pauca et. al. implemented sparsity within MM as

$$H_{cj} \leftarrow H_{cj} \frac{(W^T V)_{cj} - \beta (c_1 H_{cj} + c_2 E_{cj})}{(W^T WH)_{cj} + \epsilon}$$

$$c_1 = \omega^2 - \omega \frac{\|\bar{H}\|_1}{2\|\bar{H}\|_2}$$

$$c_2 = \|\bar{H}\| - \omega \|\bar{H}\|_2$$

$$\omega = \sqrt{kn} - \left(\sqrt{kn} - 1 \right) \text{sparseness}(H)$$

- Automated cluster labeling (& clustering)
- Synonym generation (features)
- Possible automated ontology creation
- Labeling can be applied to *any* hierarchy

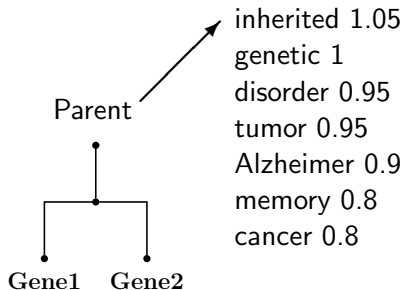
Comparison of SVD vs. NMF

	SVD	NMF
Solution Accuracy	A	B
Uniqueness	A	C
Convergence	A	C-
Querying	A	C+
Interpretability of Parameters	A	C
Interpretability of Elements	D	A
Sparseness	D-	B+
Storage	B-	A

Labeling Algorithm

Given a hierarchical tree and a weighted list of terms associated with each gene, assign labels to each internal node

- Mark each gene as labeled.
- For each pair of labeled sibling nodes
 - Add all terms to parent's list
 - Keep top t terms



Alzheimer 0.9
memory 0.8
disorder 0.6
genetic 0.5
inherited 0.45

tumor 0.95
cancer 0.8
inherited 0.6
genetic 0.5
disorder 0.35

Calculating Initial Term Weights

Three different methods to calculate initial term weights:

- Assign global weight associated with each term to each document.
- Calculate document-to-term similarity (LSI).
- For NMF, for each document j :
 - Determine top feature i (by examining H).
 - Assign dominant terms from feature vector i to document j , scaled by coefficient from H .
 - (Can be extended/thresholded to assign more features)

- Many studies validate via inspection.
- For automated validation, need to generate a “correct” tree labeling.
 - Take advantage of expert opinions, i.e., indexers from MeSH.
 - Create MeSH meta-document for each gene, i.e., list of MeSH headings.
 - Label hierarchy using global weights of meta-collection.

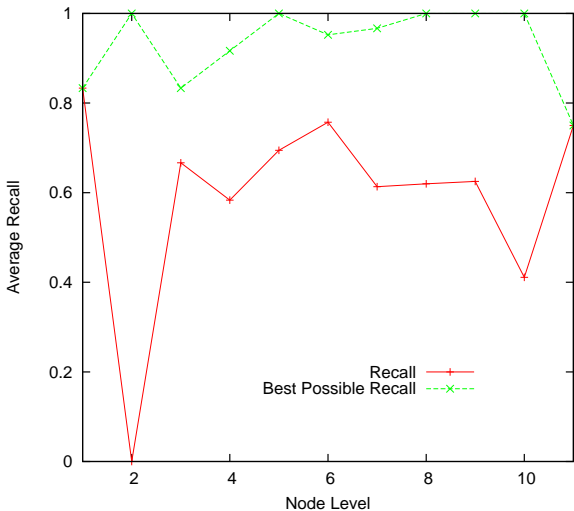
- From traditional IR, recall is ratio of relevant returned documents to all relevant documents.
- Extending this to trees, recall can be found at each node.
- Averaging across each tree depth level produces *average recall*.
- Averaging average recalls across all levels produces *mean average recall*.

Feature Vector Replacement

Unfortunately, MeSH vocabulary is too restrictive, so nearly all runs produced near 0% recall.

- Map NMF vocabulary to MeSH terminology.
- Foreach document i :
 - Determine j , the highest coefficient from i th column of H .
 - Choose top r MeSH headings from corresponding MeSH meta-document.
 - Split each MeSH heading into tokens.
 - Add each token to j th column of W' , where weight is global MeSH header weight \times coefficient from H .

Result: feature vector comprised solely of MeSH terms (MeSH feature vector).

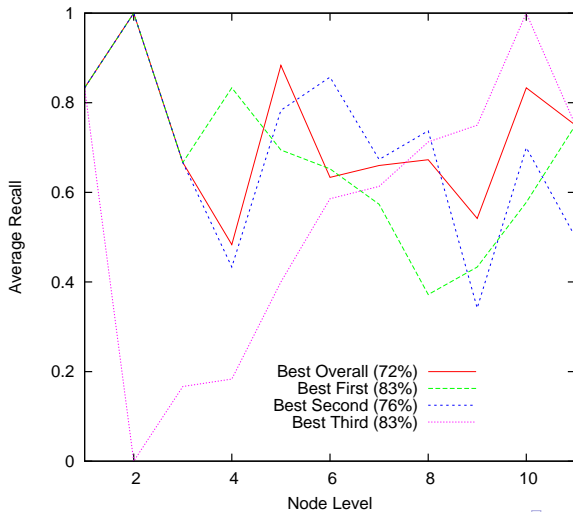


Five collections were available:

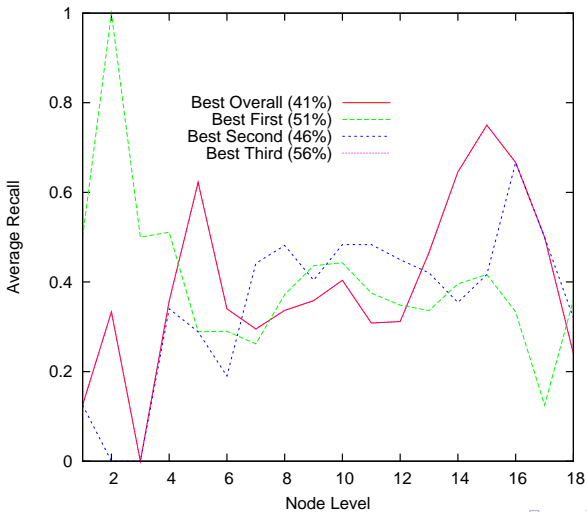
- 50TG, test set of 50 genes
- 115IFN, set of 115 interferon genes
- 3 cerebellar datasets, 40 genes of unknown relationship

Each set was run under the given constraints for $k = 2, 4, 6, 8, 10, 15, 20, 25, 30$:

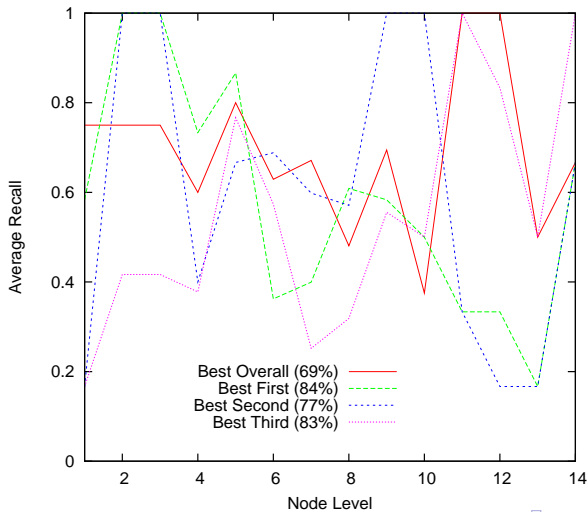
- no constraints
- smoothing W with $\alpha = 0.1, 0.01, 0.001$
- smoothing H with $\beta = 0.1, 0.01, 0.001$
- sparsifying W with $\alpha = 0.1, 0.01, 0.001$ and sparseness = 0.1, 0.25, 0.5, 0.75, 0.9
- sparsifying H with $\beta = 0.1, 0.01, 0.001$ and sparseness = 0.1, 0.25, 0.5, 0.75, 0.9

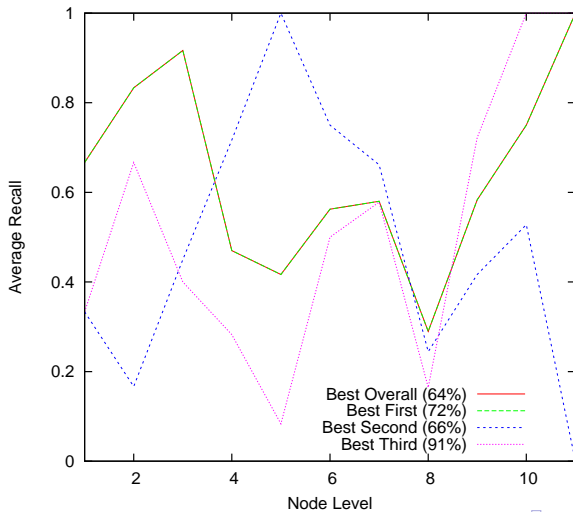


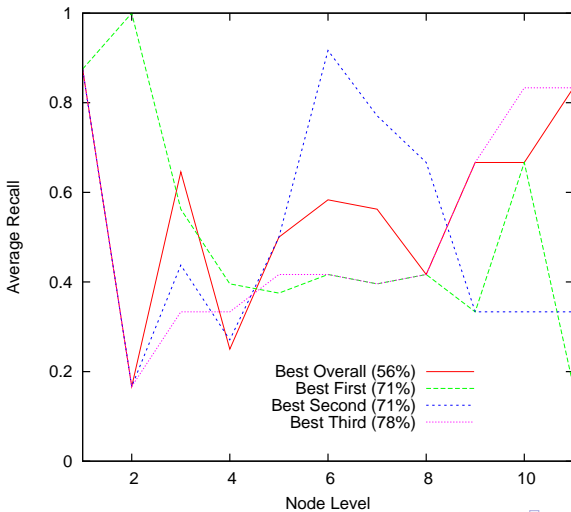
115IFN Recall



Math1 Recall







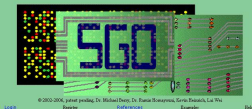
- Recall was more a function of initialization and choice of k than constraint.
- Often, the NMF run with the smallest approximation error did not produce the optimal labeling.
- Overall, sparsity did not perform well.
- Smoothing W achieved the best MAR in general.
- Small parameters choices and larger values of k performed better.

- Lots of NMF algorithms, each with different strengths/weaknesses.
- More complex labeling scheme.
- Different weighting schemes.
- Investigate hierarchical graphs.
- Visualization?
- Gold Standard?

This tool as implemented is called the Semantic Gene Organizer (SGO).

Much of the technology used in SGO are the basis for Computable Genomix, LLC.

SGO & CGx Screenshots



Description

Semantic Gene Organizer (SGO) is an automated method to cluster genes based on conceptual relationships derived from MEDLINE abstracts. It uses a variant of the vector-space model called Latent Semantic Indexing (LSI) to represent genes as vectors in lower-dimensional (concept) space. The relationship between genes is deduced from the cosine of the angle between gene-document vectors. A gene document is a concatenation of MEDLINE titles and abstracts identified in the LocalLink entry for each gene.

IMPORTANT NOTE: This is a test version of the program and is intended to demonstrate proof-of-concept for using LSI to functionally cluster genes. The current document collection contains only 50 hand-selected genes. In the near future, we will expand the collection to include all genes in LocalLink and OMIM databases.

Quick Start

1. Login Anonymously
2. Select the document collection (e.g. "SGO_Test_Genes") from the pull-down menu.
3. Assign a session name (e.g. "BWT") so that you can recall your searches at a later time.
4. Select accession number or keyword option from the pull-down menu.
5. Type in (or cut and paste a list of accession numbers or keywords) for your query. A keyword can be a string of words separated by a space. Each keyword query should be separated by a blank line.
6. Press Submit.
7. In the main window, your queries will appear in the upper right panel. Click any query to view the relevant genes in ranked order in the bottom panel.
8. Click on the gene symbols/boxes to view the titles and abstracts in the document associated with the gene in the upper left panel. Click on the LocalLink beside the gene symbol to go directly to the LocalLink page in a new window.

ComputableGenomix

Focusing Genomic Research

About Us

- ◆ Products
- ◆ Publications
- ◆ News
- ◆ People
- ◆ Contact Us

Welcome to Computable Genomix

Computable Genomix is the provider of bioinformatics tools including GeneIndeavor, a literature based gene function and pathway analysis tool. GeneIndeavor automatically mines empirical research documents to extract both explicit and implicit gene associations and to assist in making predictions about target genes for further research, that's just the beginning. Using the primary literature, GeneIndeavor can rapidly rank genes with respect to any user defined keyword query (e.g. disease names, molecular functions, phenotypes, etc.), identify gene neighbors and gene hubs in a dataset, and determine the overall functional cohesiveness of any group of genes.

Login

Username:

Password:



Login

Forgotten password?

Retrieve password

New user?

Get an account

 <p>About Us</p> <p>Computable Genomix helps researchers integrate high throughput genomic data within the context of freely and privately available research documents.</p> <p>Learn More</p>	 <p>Products</p> <p>Computable Genomix has the right product for you and can even deliver a tailored solution to fit your genomic research needs.</p> <p>Learn More</p>	 <p>News</p> <p>Computable Genomix is a fast moving and growing enterprise that is delivering on its mission. In the new and other media you can find out more.</p> <p>Learn more</p>
--	--	--

Computable Genomix, LLC © 2007 • Memphis, Tennessee • 901.313.4750 • About Us • Contact Us • Created by Sarah Pelobonok

SGO & CGx Screenshots

Queries

1	19699 (reln)
2	12568 (c08.5)
3	22359 (v18r)

19699 (reln)

Gene Rank	LocalRank	OMM
1	19699	
2	0.90476 v50z1	13131
3	0.872171 kv6	14975
4	0.750166 v1d4	22350
5	0.724115 kv6f3	11921
6	0.599143 kv6c2	22413
7	0.548956 kv6o1	19876
8	0.39928 kv6c3	22415
9	0.39928 kv6c1	22416

GeneIndexer

[Select a collection] (in genes in database with a factory available)

Ranked gene 5

Top weighted terms of Ranked gene 5

- GO term 1
- GO term 2
- GO term 3
- GO term 4
- GO term 5
- GO term 6
- GO term 7
- GO term 8
- GO term 9
- GO term 10
- GO term 11
- GO term 12
- GO term 13
- GO term 14

Ranked gene 5

Related genes (10 items):

- Gene term 1
- Gene term 2
- Gene term 3
- Gene term 4
- Gene term 5
- Gene term 6
- Gene term 7
- Gene term 8
- Gene term 9
- Gene term 10

Copyright © 2007. All rights reserved. For more information, please contact us at support@computablegenomics.com

Acknowledgments

SGO: shad.cs.utk.edu/sgo

CGx: computablegenomix.com

Committee:

- Michael W. Berry, chair
- Ramin Homayouni (Memphis)
- Jens Gregor
- Mike Thomason

Paúl Pauca, WFU