Pacific Symposium on Biocomputing 12:379-390(2007)

# RECOMMENDING PATHWAY GENES USING A COMPENDIUM OF CLUSTERING SOLUTIONS

DAVID M. NG$^+$, MARCOS H. WOEHRMANN$^+$, AND JOSHUA M. STUART$^*$

*Department of Biomolecular Engineering, University of California, Santa Cruz*
*Santa Cruz, CA 95064, USA*
$^+$*Equal coauthorship.* $^*$*Email: jstuart@soe.ucsc.edu*

A common approach for identifying pathways from gene expression data is to cluster the genes without using prior information about a pathway, which often identifies only the dominant coexpression groups. Recommender systems are well-suited for using the known genes of a pathway to identify the appropriate experiments for predicting new members. However, existing systems, such as the GeneRecommender, ignore how genes naturally group together within specific experiments. We present a collaborative filtering approach which uses the pattern of how genes cluster together in different experiments to recommend new genes in a pathway. Clusters are first identified within a single experiment series. Informative clusters, in which the user-supplied query genes appear together, are identified. New genes that cluster with the known genes, in a significant fraction of the informative clusters, are recommended. We implemented a prototype of our system and measured its performance on hundreds of pathways. We find that our method performs as well as an established approach while significantly increasing the speed and scalability of searching large datasets. [Supplemental material is available online at sysbio.soe.ucsc.edu/cluegene/psb07.]

## 1. Introduction

We developed an approach that efficiently searches the growing body of functional genomics data for new genes that act in a pathway of interest. For many pathways, the cell must coordinate the expression of the participating genes so that their products are present at the same time and place. The functional similarity of these genes may be detectable in gene expression data, if the context under which the pathway is activated has been assayed. As the results of DNA microarray studies continue to be contributed to public repositories such as the Gene Expression Omnibus[1], the chance that such a context exists in the database becomes more likely.

[*] corresponding author.

However, finding this context among the many irrelevant experiments can be as challenging as finding a needle in a haystack.

Existing recommendation systems for gene pathway discovery, such as the GeneRecommender[2] and the Signature Algorithm[3], have shown promise for finding genes of related function. However, these approaches do not take advantage of the natural clustering of genes in different experiment series. Rather than using pre-existing clusters, they build a cluster around the given query genes using microarray hybridizations under which the query genes are most strongly up- or down-regulated. Because of this, they can miss correlations present across multiple hybridizations where the absolute levels of the query genes are different but where their expression changes are still highly similar. In addition, these approaches can be computationally intensive since the algorithms must compute the correlation of every gene compared to the input query set. Therefore, we expect these approaches to scale poorly as the number of microarray hybridizations increases.

The task of identifying new genes that act in a pathway is analogous to the task of making product recommendations for customers of online stores. We have developed a *collaborative filtering*-based gene recommendation system[4], ClueGene, which uses pre-computed clusters of genes to recommend new genes for a query pathway.

In online shopping, recommendations for additional purchases are based on the contents of a customer's shopping cart and on the purchasing history of previous customers[5]. In gene pathway prediction, recommendations for additional genes in a pathway are based on the known genes of the pathway (the *query genes*) and on clusters of coexpressed genes computed from experimental data. The ClueGene system precomputes clustering solutions for multiple data sets and stores each identified cluster in a database referred to as the *Cluster Compendium* (see Figure 1). Storing clusters provides a more compact representation of gene regulation groups compared to storing the entire set of microarray results. Given a query, consisting of a set of genes, the ClueGene recommender algorithm scans the Cluster Compendium for clusters containing a significant proportion of the query genes. It scores each gene in the genome using a weighted vote across these clusters. ClueGene returns its recommendation as a list of all the genes in the genome ranked by score.

Using a method analogous to e-commerce recommender systems, ClueGene quickly and accurately predicts members for a large number of pathways. We conclude that collaborative filtering approaches may provide an efficient and accurate methodology for scanning large amounts of functional

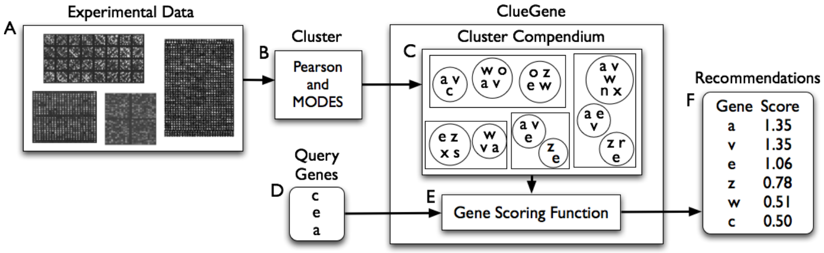

Figure 1. ClueGene method overview. A. Datasets from Stanford Microarray Database and Gene Expression Omnibus are collected. B. Clusters are derived from each dataset by forming a network from all significant pairwise Pearson correlations from which dense subnetworks are identified with MODES. C. The grouping of clusters by dataset is maintained in the Cluster Compendium. D. A list of genes is supplied as the input query. E. All genes are scored according to their degree of co-clustering with the query. F. The top-scoring genes are returned as the recommendations.

genomics data to predict gene function.

## 2. Construction of the Cluster Compendium

To test the ClueGene system, we clustered 44 different *Saccharomyces cerevisiae* experiment series collected from the Stanford Microarray Database[6] and the Gene Expression Omnibus. The datasets represent a diverse collection of experiments ranging from perturbations such as from various stresses and deletions to normative conditions such as cycling cells and regulation of general transcription. To increase the diversity of clusters in our search, we included two datasets in which the binding of specific transcription factors were assayed with genome-wide chromatin immunoprecipitation [7,8]. For a full list of the datasets and their references, please see Supplemental Table 1.

To find clusters of coregulated genes within each data series, we first constructed a coexpression network and then identified clusters as dense subnetworks in the network. A coexpression network was constructed by connecting any two genes whose Pearson correlation was equal to or greater than four standard deviations above what was expected by chance (based on randomly permuting gene vectors). The MODES (Mining Overlapping DEnse Subgraphs) algorithm[9] was used to identify highly-connected sets of genes in the resulting network. MODES clusters genes into overlapping subsets, allowing a gene to belong to multiple clusters.

In total, 6900 clusters from the 44 datasets were identified and loaded into a yeast Cluster Compendium from which recommendations could be computed. Clusters derived from the gene expression study represent sets of genes whose relative changes in expression across a single dataset are highly similar. Clusters derived from the chromatin-immunoprecipitation experiments represent sets of genes that are bound by a common set of transcription factors. Thus, clusters from both types of dataset group genes according to shared regulatory information. Note that the clustering step does not depend on a particular query and therefore was pre-computed.

## 3. Scoring Genes Based on Co-clustering with a Pathway

ClueGene is given a set of genes, called the *query*, $Q$, that are thought to be functionally related. It then scores each gene, $g$, in the genome, $G$, based on how often the gene appears in clusters with the genes in $Q$. We define a function that assigns higher scores to genes that appear in clusters containing a high proportion of query genes.

Let $D$ be a set of clustering solutions where each element of $D$ is a set of clusters. Define $N_{gd}$ to be the number of clusters in data set $d$ that contain $g$ and at least one gene from $Q$. The co-clustering score $C(g)$ of gene $g \in G$ is:

$$C(g) = \sum_{d \in D} \left[ \frac{1}{N_{gd}} \sum_{c \in d} \frac{|Q \cap c|}{|Q \cup c|} I(g \in c) \right]$$

where $I$ is the *indicator function* that returns 1 if its argument is true and 0 otherwise.[a]

The intuition underlying the choice of scoring function is to identify genes that occur in small and specific clusters with the query genes. If $g$ belongs to a large cluster that also happens to have several of the query genes, this observation is down-weighted because the co-occurrence of gene $g$ with the query may arise by chance if the cluster is large enough. On the other hand, if $g$ belongs to a small cluster that also contains several of the query genes, this observation receives a high weight because the co-occurrence is less likely to be serendipitous.

Dividing by $N_{gd}$ corrects for the number of clusters that a gene appears in. Without this correction, high scores could be assigned to genes that

[a] The time complexity of ClueGene is $O(|D|)$, where $|D|$ is the number of data sets. The time complexity of GeneRecommender is $O(|D| \times \overline{e})$, where $\overline{e}$ is the average number of experiments per data set. We expect $\overline{e}$ to grow over time as high-throughput techniques become less costly and more common. For details of the scoring algorithm and time complexity analysis see Supplemental Appendix A.

are "central" in the coexpression network simply because they appear in several clusters. Note that one could also consider including an additional normalization term to correct for missing data.[b]

## 4. Results on Positive Control Pathways

To estimate the accuracy of a search, either from ClueGene or the GeneRecommender, we used a leave-half-out strategy. Half of the original genes in the pathway were used as the query to search for the remaining half. We refer to the withheld members as the *expected* set of genes. We obtain a conservative estimate of the accuracy of the search by using only the ranks of the expected genes while ignoring the ranks of the query genes.

A single leave-half-out search results in a list of genes, sorted by their co-clustering scores, $C(g)$. At a given score cutoff $z$, the precision and the recall of the search are measured. Expected genes with scores of at least $z$ are considered to be recommended, while the rest are not. The precision is defined to be $p/n$ where $p$ is the number of expected genes with scores of at least $z$, and $n$ is the number of total genes with scores of at least $z$. The recall is defined to be $p/t$ where $t$ is the total number of expected genes, and $p$ is the same as before. Sweeping through a range of cutoff levels produces various precision levels as a function of recall.

For positive control testing, we selected four functionally-related groups of genes defined by KEGG[10]: the *Cell Cycle* category—containing genes involved in the actuation and regulation of the cell cycle, the *Oxidative Phosphorylation* category—containing genes involved in the final stage of cellular respiration, the *Proteasome* category—containing genes encoding subunits of the 26S or 19S components of the proteasome, and the *Ribosome* category—containing genes that encode subunits of the small and large cytosolic ribosome. These pathways were previously shown in Stuart *et al.* [11] to contain genes with highly correlated expression profiles conserved across multiple species.

As a negative control, we created four sets of genes selected at random from the entire yeast genome; these random sets contained 10, 25, 50, and 100 genes. ClueGene and GeneRecommender were both run on the posi-

[b] Dividing by $M_g$, the number of datasets in which $g$ appears, would allow genes with differing amounts of missing data to be directly compared. We found that dividing by $M_g$ had little effect on our results, presumably because the yeast data contains very little missing data. However, we suggest including a division by $M_g$ if applied to other species in which more missing data is expected.

tive control pathways and the randomly constructed sets of genes. Figure 2 shows the precision-recall curves for the *Cell Cycle*, *Oxidative Phosphorylation*, *Proteasome*, and *Ribosome* categories.

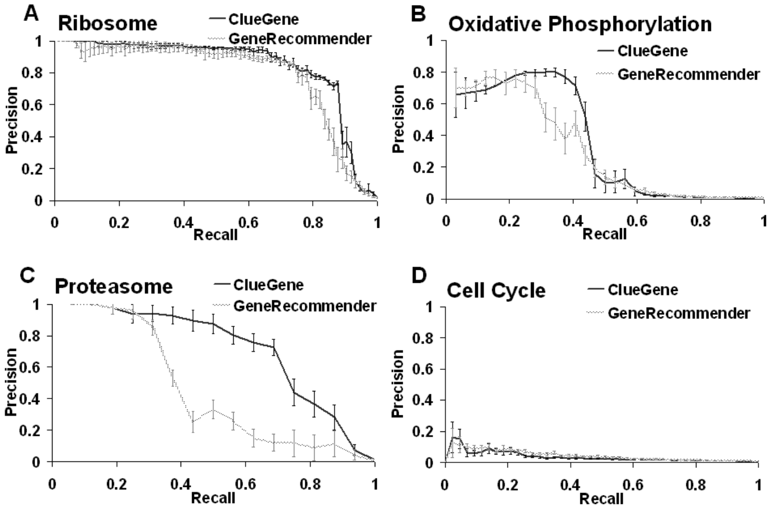

Figure 2. Estimates of the precision at various levels of recall for the four test pathways. Black lines, accuracies for ClueGene; gray lines, accuracies for GeneRecommender. Error bars show +/−1 standard deviations from 10 leave-half-out runs. A. Ribosomal subunits. B. Oxidative phosphorylation. C. Proteasomal subunits. D. Cell cycle related genes.

As expected, ClueGene and GeneRecommender perform equally well for the *Ribosome* and *Oxidative Phosphorylation* categories. ClueGene and GeneRecommender both cannot identify members of the *Cell Cycle* category. Since the KEGG *Cell Cycle* category contains genes that act at different stages of the cell cycle, we tested whether the performance could be improved by dividing up the category into gene groups that are known to act at the same phase. However, we found that all subsets of the *Cell Cycle* category, corresponding to different phases of the cell cycle, also performed poorly on the searches (see Supplemental Table 3). This suggests that the yeast Cluster Compendium does not contain informative clusters for identifying genes involved in this process.

ClueGene appears to perform better than the GeneRecommender on predicting subunits of the proteasome. To assess whether the difference between the performance of the two methods was significant, we measured the area under the curve (AUC) of the precision-recall plot to summarize the overall performance of a search method.[c] The average and standard deviation across ten leave-half-out tests was calculated. Table 1 summarizes the results of testing on four positive control pathways.

Table 1. Control Test Results.

| KEGG category | Category size | ClueGene AUC | GeneRecommender AUC | CG Random[a] AUC | $z$-score[b] |
|---|---|---|---|---|---|
| *Cell Cycle* | 87 | 0.0373 | 0.0444 | 0.0082 | -0.5139 |
| *Oxidative Phosphorylation* | 64 | 0.3941 | 0.3058 | 0.0057 | 0.4549 |
| *Proteasome* | 32 | 0.7149 | 0.5631 | 0.0028 | 0.2827 |
| *Ribosome* | 147 | 0.8579 | 0.7942 | 0.0147 | 0.2387 |

*Note*: [a] ClueGene run on random pathways of size 10, 25, 50 and 100; the AUCs of three runs were averaged for each size. Reported values derived by linear interpolation.
[b] Mann-Whitney $z$-score.

The AUCs matched our intuitive sense of the performance as observed in Figure 2. In addition, the results on the negative controls yielded AUCs expected from uniformly distributed ranks (Table 1 shows the negative control results for ClueGene only; the results on negative controls were nearly identical for GeneRecommender). The AUC can be used as a single measure to evaluate a large collection of pathways to identify those pathways associated with high ClueGene performance. To detect pathways with significant accuracy, one could perform a $t$-test between the AUCs obtained on the random controls compared to a specific pathway. Our focus, however, was to identify any pathways for which the two search algorithms produced significantly different precision levels.

To test whether the search results were statistically comparable or different for a particular pathway, a Mann-Whitney test was performed to compare the ranks of the expected genes returned by ClueGene to those returned by GeneRecommender.[d] We used the $z$-score returned by the

[c] The AUC was estimated using the trapezoid rule, commonly used in discrete integration. The final AUC was normalized by dividing by the theoretical maximum: $1 - \frac{1}{t}$.

[d] For a pathway of size $2t$, let the ranks assigned by ClueGene to the $t$ expected genes be $X_1, X_2, \ldots, X_t$, and the ranks assigned by GeneRecommender be $Y_1, Y_2, \ldots, Y_t$. $X$ and $Y$ were combined into a single vector $W$ and sorted. The Mann-Whitney statistic was computed as $t^2 + 0.5(t+1) - U$, where $U$ is the sum of the new ranks of X in $W$.

Mann-Whitney test as a measure of the difference in prediction accuracy between the two search engines. $z$-scores larger than 2 indicated ClueGene found a significantly more accurate result than GeneRecommender. Conversely, $z$-scores less than $-2$ indicated the GeneRecommender performed more accurately for a pathway than ClueGene. For each pathway, we calculated the Mann-Whitney $z$-score for each of the 10 different leave-half-out tests. We reported the median $z$-score from these 10 runs. Note that this is equivalent to calling a difference between the two methods significant if a majority of the leave-half-out tests yield significantly different rankings.

For the positive control pathways, we found that ClueGene and GeneRecommender returned statistically similar results. For example, even though the difference in AUC between the two methods is higher for ClueGene (0.71) compared to the GeneRecommender (0.56) for the *Proteasome*, the rankings assigned to the expected genes were not found to be significantly different (0.28 standard deviations) (Table 1). Thus, the ClueGene search engine was found to perform as accurately as the GeneRecommender using the Mann-Whitney test for the four positive control pathways. To gauge the general performance of ClueGene, we next set out to test it on a large set of pathways.

## 5. Results On Diverse Pathways

To compare the performance of ClueGene and GeneRecommender, the accuracy of each method was measured on 1441 functionally-related groups of genes defined by Gene Ontology[12], 80 defined by KEGG[10], and 180 defined by MIPS[13], for a total of 1701 pathways (see Supplemental Table 2 for the complete results). Figure 3A shows the distribution of AUCs computed for ClueGene and GeneRecommender.

The Mann-Whitney $z$-scores were centered on 1 (Figure 3A). This was surprising because it indicated that, in general, ClueGene had higher, although not significantly higher, performance across the pathways compared to the GeneRecommender. Few extreme $z$-scores were observed, indicating the two methods perform comparably on the set of pathways.

We collected five pathways with the highest and five with the lowest Mann-Whitney $z$-scores (see Table 2). The results indicate the ClueGene method performed better for pathways specific to energy generation. For example, ClueGene obtained significantly better rankings for GO categories

A $z$-score was computed as $z = (U - \frac{1}{2}t^2)/(t\sqrt{\frac{1}{12}(2t+1)})$.

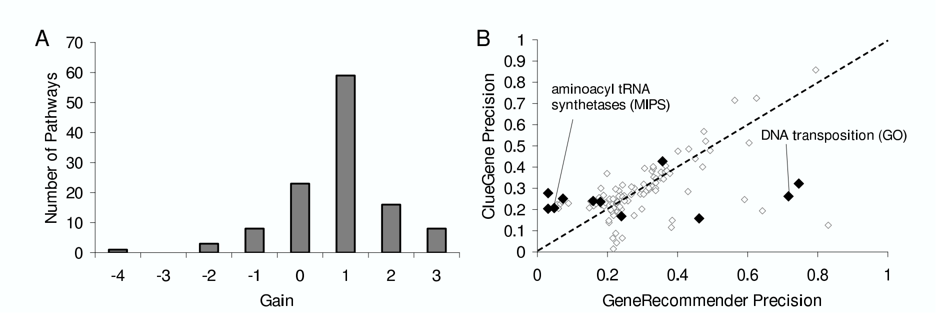

Figure 3. Performance comparison of ClueGene to GeneRecommender on diverse pathways. A. Distribution of AUCs for ClueGene and GeneRecommender on a non-redundant set of pathways for which at least one of the methods had an AUC of 0.20 or better. B. Each pathway's precision at the 50% recall rate is plotted for ClueGene against GeneRecommender. Open diamonds, pathways with absolute Mann-Whitney $z$-scores less than 2; black diamonds, pathways with absolute $z$-scores of at least 2.

Table 2. Selected Pathways with Extreme Mann-Whitney $z$-scores.

| Top ClueGene Categories | $z$-score[a] | Top GeneRecommender Categories | $z$-score |
|---|---|---|---|
| *membrane-bound organelle* (GO) | 2.8 | *protein binding* (GO) | -5.1 |
| *carboxylic acid metabolism* (GO) | 2.6 | *DNA recombination* (GO) | -4.2 |
| *respiratory chain complex III (GO)* | 2.6 | *DNA metabolism* (GO) | -3.7 |
| *oxidoreductase activity, acting on heme group of donors* (GO) | 2.6 | *DNA transposition* (GO) | -2.9 |
| *aminoacyl-tRNA-synthetases* (MIPS) | 2.2 | *nucleic acid binding* (GO) | -2.9 |

*Note*: [a] Mann-Whitney derived $z$-score. Higher $z$-scores indicate ClueGene ranked query genes toward the top compared to GeneRecommender.

*carboxylic acid metabolism* and *respiratory chain complex III.* The GeneRecommender outperformed ClueGene on pathways directly involved in the generation and manipulation of DNA (e.g. the GO categories *DNA recombination* and *DNA metabolism*). The ClueGene algorithm had a higher precision for several pathways, including the MIPS *aminoacyl tRNA synthetase* category (Figure 3B), but the significance compared to GeneRecommender was borderline. To identify datasets that contributed significantly to the high-ranking of the top-scoring genes, for the 25 highest scoring genes we summed the contributions from each dataset. This assigns each dataset a score relative to its contribution. In the case of the *aminoacyl tRNA synthetases*, ClueGene was able to find a significant coregulation pattern in the datasets of Brem *et al.* and Yvert *et al.* (see Supplemental Table 1 for the references). We plotted the expression levels of the query genes for a subset of the conditions (see Supplemental Figure 1). Visual inspection of the expression levels reveals that, while the shape of the expression levels of the aminoacyl tRNA synthetases change in a coordinate fashion, their absolute levels of expression are very low. The GeneRecommender therefore assigned these experiments low scores and therefore missed the coordinate expression changes of this group of functionally-related genes.

The GeneRecommender had high accuracy on the GO *DNA Transposition* category, and identified a significant coexpression pattern within the dataset published by Hughes *et al.*[14]. The transposons had extremely high levels of expression with very little variance across this dataset (data not shown). The shape of the expression levels relative to each other look dissimilar. Thus, clustering based on centered Pearson correlation fails to capture the pattern of coregulation of the transposons in this dataset.

## 6. Discussion

We have found that a collaborative-filtering-based strategy for predicting new members of a pathway from gene expression data gains speed and scalability without sacrificing search performance. The ClueGene search engine uses pre-computed clustering solutions to identify patterns of coregulation between novel and known genes of a pathway. In general, the ClueGene search engine performed comparably and, in some cases, better than the GeneRecommender on a diverse collection of categories from MIPS, Gene Ontology, and KEGG.

The current implementation of ClueGene has several limitations. For example, we only considered positive correlation in our search for related genes. In the future, we plan to test the hypothesis that including anticorrelation can improve pathway prediction. We will build a new Cluster Compendium using absolute Pearson correlation. ClueGene could use these new clusters either alone or together with the original clusters. By measuring AUCs associated with each Cluster Compendium, the search engine could identify which compendium is more predictive for a specific pathway. ClueGene could be used to predict the function of unknown proteins. A single gene of unknown function could be supplied as the query and its function inferred from the functions of known genes that sort to the top of the search result. In our study, we focused on assessing the performance of the search algorithm for identifying known genes of well-defined pathways. To facilitate these additional uses, we have made the source code available from our website.

ClueGene was designed to extend to a diversity of organisms and datasets. The advantage of using clusters rather than the primary data is that ClueGene avoids the problem of having to normalize across different microarray platforms. The approach can accommodate new datasets in an online fashion: when a new dataset is available, clusters can be identified and added to the compendium from which updated recommendations can be made.

Generalizing over species and data types will broaden the range of genetic processes that can be searched. The speed of ClueGene enables it to be applied to a large number of datasets for which the application of the GeneRecommender would be prohibitively slow, or have datasets too large to load into computer memory. Extending the Cluster Compendium to additional organisms will allow searches to be performed in organisms where predicted gene sequences (but possibly not the entire genome sequence) are available. ClueGene could be generalized by extending the cluster database to additional organisms, as well as by developing a method that identifies patterns of conservation in gene co-clustering.

Genes recommended from coexpression clusters on several organisms may correspond to either ancient members of the pathway or to more newly evolved participants. For example, a gene may co-cluster with a pathway in humans and mice, but not in non-mammals. The co-clustering pattern in this case suggests the gene was recruited into the pathway sometime after the mammals diverged from the other animals. Finding such a pattern indicates the gene's regulation program (*cis*- or *trans*-acting regulatory factors) may have undergone recent adaptations rather than a slower fine-tuning over a larger evolutionary period, which may provide clues about the gene's function. We envision computing a co-clustering score at each node in the phylogenetic tree that relates the set of organisms for which a Cluster Compendium is available. In this way, the ClueGene algorithm could make explicit use of the complementary information present in experimental data collected from a variety of organisms.

## References

1. Barrett T, Suzek TO, Troup DB, Wilhite SE, Ngau WC, Ledoux P, Rudnev D, Lash AE, Fujibuchi W, Edgar R. NCBI GEO: mining millions of expres-

sion profiles—database and tools. *Nucleic Acids Res.* 2005 Jan 1;33 Database Issue:D562–D566.

2. Owen AB, Stuart J, Mach K, Villenvue AM, Kim S. A Gene Recommender Algorithm to Identify Coexpressed Genes in *C. elegans*. *Genome Research* 2003;13:1828–1837.
3. Ihmels J, Friedlander G, Bergmann S, Sarig O, Ziv Y, Barkai N. Revealing modular organization in the yeast transcriptional network. *Nature Genetics* 2002 Aug;31(4):370–377.
4. Breese JS, Heckerman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence* 1998 July:43–52.
5. Linden G, Smith B, York J. Amazon.com Recommendations: Item-to-Item Collaborative Filtering. *IEEE Internet Computing* 2003 Jan/Feb;7(1):76–80.
6. Ball CA, Awad IA, Demeter J, Gollub J, Hebert JM, Hernandez-Boussard T, Jin H, Matese JC, Nitzberg M, Wymore F, Zachariah ZK, Brown PO, Sherlock G. The Stanford Microarray Database accommodates additional microarray platforms and data formats. *Nucleic Acids Res* 2005 Jan 1;33(1):D580–2.
7. Lieb JD, Liu X, Botstein D, Brown PO. Promoter-specific binding of Rap1 revealed by genome-wide maps of protein-DNA association. *Nat Genet* 2001 Aug;28(4):327–334.
8. Iyer VR, Horak CE, Scafe CS, Botstein D, Snyder M, Brown PO. Genomic binding sites of the yeast cell-cycle transcription factors SBF and MBF. *Nature* 2001;409(6819):533–8.
9. Hu H, Yan X, Huang Y, Han J, Zhou XJ. Mining coherent dense subgraphs across massive biological networks for functional discovery. *Bioinformatics* 2005;21 Suppl. 1 2005:i213–i221.
10. Kanehisa M, Goto S, Hattori M, Aoki-Kinoshita KF, Itoh M, Kawashima S, Katayama T, Araki M, Hirakawa M. From genomics to chemical genomics: new developments in KEGG. *Nucleic Acids Res.* 2006;34:D354–D357.
11. Stuart JM, Segal E, Koller D, Kim S. A gene-coexpression network for global discovery of conserved genetic modules. *Science* 2003;302:249–55.
12. The Gene Ontology Consortium. Gene Ontology: tool for the unification of biology. *Nature Genet.* 2000;25: 25–29.
13. Guldener U, Munsterkotter M, Kastenmuller G, Strack N, van Helden J, Lemer C, Richelles J, Wodak SJ, Garcia-Martinez J, Perez-Ortin JE, Michael H, Kaps A, Talla E, Dujon B, Andre B, Souciet JL, De Montigny J, Bon E, Gaillardin C, Mewes HW. CYGD: the Comprehensive Yeast Genome Database. *Nucleic Acids Res.* 2005 Jan 1;33(Database issue):D364–8.
14. Hughes TR, Marton MJ, Jones AR, Roberts CJ, Stoughton R, Armour CD, Bennett HA, Coffey E, Dai H, He YD, Kidd MJ, King AM, Meyer MR, Slade D, Lum PY, Stepaniants SB, Shoemaker DD, Gachotte D, Chakraburtty K, Simon J, Bard M, Friend SH. Functional discovery via a compendium of expression profiles. *Cell* 2000 Jul 7;102(1):109–26.