Significance Metrics for Clusters of Mixed Numerical and Categorical Yeast Data

Bill Andreopoulos

Aijun An

Xiaogang Wang

Technical Report CS-2003-12

October 2003

Department of Computer Science and Engineering

4700 Keele Street North York, Ontario M3J 1P3 Canada

Significance metrics for clusters of mixed numerical and categorical yeast data

Bill Andreopoulos[1], Aijun An[1], Xiaogang Wang[2]

[1]Department of Computer Science, York University, Toronto, Ontario, Canada, M3J 1P3
[2]Department of Mathematics and Statistics, York University, Toronto, Ontario, Canada, M3J 1P3
billa@cs.yorku.ca, (416)787-1651

October 2003

Abstract : We designed, implemented and tested a clustering tool for numerical data sets derived from gene expression DNA microarray studies. Our tool incorporates into the clustering process the existing knowledge as *categorical annotations (CAs)* on the genes, as well as the uncertainties concerning the correctness of the CAs as *confidence values (CVs)* on the CAs. CVs are a measure of the certainty of correctness of the existing knowledge and are derived from GeneOntology Evidence Codes. This allowed us to apply new significance metrics to the resulting clusters to extract the most prominent CAs in the clusters. We applied the extracted CAs to the other genes in the cluster to predict their function and we validated the predictions.

Keywords: clustering, categorical, numerical, function prediction, GeneOntology

Availability of software: http://www.cs.yorku.ca/~billa/SIG/clust.tar.gz

1 INTRODUCTION AND MOTIVATION

Clustering attempts to partition a set of data items into groups, so that items with similar characteristics are grouped together and different groups contain items with different characteristics [8-10,12,18]. There exist a plethora of data sets that are fully annotated with numerical data – such as gene expression data sets - but for which some uncertain knowledge exists. In many of these data sets it is easy to extract confidence values on the knowledge to indicate the certainty that the knowledge is correct [5,14].

Clustering a numerical gene expression data set derived from DNA microarray studies, by incorporating in the clustering process existing knowledge on the genes along with confidence values on the correctness of the knowledge, results in clusters on which it is simple to spot the most reliable classifications, as well as the suspected misclassified genes [6-8]. This provides a different type of insight for predicting the functional roles of genes by giving the 'full picture' of the data set. This enables new significance metrics to be applied to the resulting clusters, for predicting gene functions and validating the predictions.

We designed and tested a practical clustering tool for numerical data sets - particularly from gene expression DNA microarray studies - that allowed us to consider the uncertain knowledge and confidence both during and after the clustering when analyzing the data. One of the main advantages of our algorithm is that it offered us the opportunity to develop and apply novel significance metrics to the resulting clusters. These significance metrics are novel in that they allow CAs to be extracted from clusters based on their CVs, for functional prediction of other genes, as well as for validating the placement of genes in specific clusters. The extracted CAs can be applied to genes that were attached to clusters on the basis of either numerical similarity or categorical similarity, as described later.

The remainder of this paper is organized as follows. First, in Section 2, we describe previous work on clustering and discuss the differences and advantages that our algorithm offers. Then, in Section 3, we describe our algorithm, we discuss the biological data sets it is designed for and we explain how the design meets the goals of clustering and analysis. In Section 4 we describe how we applied the clustering algorithm to highly noisy simulated genomic data sets for which the correct result was known, thus showing that the algorithm was successful in reproducing approximately the correct cluster structure. In Section 5 we propose two significance metrics for the clusters and we discuss their utility for functional prediction and analysis. Implications of our study for biologists and functional prediction are discussed in Section 6. Finally, we conclude the paper in Section 7.

2 RELATED WORK

K-modes is a clustering algorithm that deals with categorical annotations (CAs) only [15]. It assumes all data items have CAs, an assumption our technique does not make. K-modes does not consider uncertainties on CAs, while our technique does. More details on K-modes are given in Section 3.2.1.

K-prototypes is a mixed categorical/numerical clustering algorithm. This algorithm takes both categorical and numerical annotations into consideration at the same level [16]. Our algorithm differs, as it deals with the categorical as serving as an underlying framework for the clustering of the numerical.

Various numerical clustering algorithms have been applied to biological data sets in the past. Some of these include *Self-Organizing Maps and Eisen's hierarchical clustering [6,7,18]*. These algorithms result in a classification of data items that is based on numerical annotations alone, and excludes any existing knowledge. It is hard to evaluate the correctness of the results to find results that may seem questionable. Our algorithm tackles this issue specifically.

Wu et al. used a new approach to analyze biological microarray data, by assigning likely cellular functions with confidence values to new yeast proteins [1]. Our approach differs in that we start from confidence values that already exist and go the opposite way, reaching numerical clustering eventually.

Various researchers have attempted to incorporate knowledge in the process of clustering biological data sets [3,11]. Brown et al. used *Support Vector Machines* to perform clustering of genomic data based on existing knowledge. However, this approach can only incorporate a limited amount of knowledge into the clustering process. Our technique incorporates a much larger amount of knowledge. Some methods have been proposed recently for ontology-based analysis of gene expression data; our algorithm integrates both the Gene Ontology annotations and GO evidence codes much more closely into the clustering process, allowing both a novel insight into the resulting clusters and the ability to determine the most reliable and most unreliable classifications in a cluster. Finally, Bayesian clustering such as the AutoClass algorithm has previously been applied to biological data sets [20].

Our approach has various advantageous differences from previous approaches:

- Our clustering algorithm may cluster data sets where all genes have numerical annotations but not all genes have CAs. Furthermore, the CAs are not definite, but probable; each CA has a CV (a real number between 0.0 and 1.0) associated with it, indicating how likely it is to be correct.
- During the clustering process, our method starts from CAs and CVs at the lower level and then moves to numerical clustering at a higher level. We use the CAs and CVs in the clustering process, not just to annotate the clusters afterwards. The method of Wu et al. as applied previously to high-throughput biological data, starts from the numerical clustering, then adds CAs at a higher level and finally CVs are calculated (P-values) [1].
- During the clustering process, genes for which a reasonable amount of knowledge exist, get clustered by putting more weight on the existing knowledge that supports the genes' similarity and less weight on the numerical similarity. On the other hand, genes for which no or little knowledge exists get clustered on the basis of numerical similarity. Thus, intra-cluster coherence and inter-cluster separation are maximized, in comparison to other methods used in this domain.
- For gene functional prediction purposes, our clustering algorithm allows us to define additional significance metrics indicating how reliable a gene's classification in a cluster is. Such metrics are calculated on the basis of how the CAs, CVs and numerical annotations support the gene's belonging in a particular cluster.
- For gene functional prediction purposes, CVs are assumed to be values that already exist (derived from GO evidence codes, as explained in Section 2.1). Our CVs point out the most reliable CAs to be used for function prediction. This is in contrast to previous methods, where CVs were calculated at the end to indicate the reliability of a CA belonging to a cluster [1].
- For validating the functional predictions, one can distinguish the CAs that are desirable to be predicted as the ones with high CVs and separate them from the rest that are of less interest because they have lower confidence.

3 DESIGN OF THE KNOWLEDGE-BASED CLUSTERING ALGORITHM

In this section we discuss the data sets that our method was designed for and then describe the algorithms.

3.1 Type of the data that is considered

Our algorithm is designed with the goal of applying it to biological numerical data sets for which some knowledge exists and the confidence that the knowledge is correct may vary. We dealt with numerical data derived from DNA microarray gene expression studies on the yeast *Saccharomyces cerevisiae*. These data sets were produced at Stanford to study the yeast cell cycle across time and under various experimental conditions. These data sets are available from the SGD database [9]. When clustering this data set, we consider each gene to be a 'data item'.

We represented *categorical annotations (CAs)* on a gene in terms of Gene Ontology (GO) and GOSlim annotations. GO is a dynamically controlled vocabulary that can be applied to many organisms, even as knowledge of gene and protein roles in cells is changing. GO is organized along the categories of molecular function, biological process and cellular location [9]. GOSlim are GO annotations that represent high level knowledge on genes and are also organized along the categories of function, process and location. Most of the GO and GOSlim annotations on the yeast genes [9] exist in the publicly accessible SGD database, along with GO evidence codes. We created six pools of CAs for each gene and each pool contained GO annotations of a specific type. Three pools contained GO annotations for molecular function, biological process and cellular location of a gene. The other three pools contained GOSlim annotations for each GO annotation.

We attached *confidence values (CVs)* to the CAs to represent our confidence that the corresponding CA is correct. CVs are real numbers (0.0 to 1.0) assigned to the CAs of a gene. Besides indicating the confidence that a CA is correct, the CVs on a gene also specify how strongly the gene's CAs should influence the clustering process. The CVs are also used in the significance metrics that we define later. We determined the CVs by using GO evidence codes. GO evidence codes symbolize the evidence that exists for any particular GO or GOSlim annotation [9].

GO evidence codes can be thought of in a loose hierarchy from strong evidence to weak evidence. For example, 'TAS' means 'Traceable Author Statement', while 'NAS' means 'Non-traceable Author Statement' [9]. We assigned a numerical CV to each of the GO evidence codes based on its location in the hierarchy, as shown in Figure 1. NR and ND are set to 0.0, because they are used for annotations to 'unknown', so the CAs should not have an effect on the clustering process. In certain DBs (Swiss-prot-human) only 3 of these evidence codes are commonly used and the most commonly used one is TAS, which is at the top of the hierarchy, meaning that strong evidence exists [17]. We combined the CAs, CVs and numerical gene expression data using a Perl script.

TAS	1.0
IDA	1.0
IMP	0.8
IGI	0.8
IPI	0.8
ISS	0.5
IEP	0.5
NAS	0.2
IEA	0.1
ND	0.0
NR	0.0

Fig. 1. GO evidence codes mapped to numerical confidence values

A CV depends primarily on whether the CA refers to something that has been *observed* to be true, as opposed to something that is just *believed* to be true. For example, a CA of a "cancerous tissue" refers to an observed phenomenon with a high CV, while a CA of a "non-cancerous tissue" refers to something that is just believed to be true, as the tissue might turn out later to be cancerous.

3.2 Algorithm details

In our clustering process, we give the existing incomplete knowledge (CAs) and uncertainty (CVs) a different weight from the numerical data; our method starts from CAs and CVs at the lower level and then moves to numerical clustering at a higher level, so that the categorical clustering serves as an underlying framework for the numerical clustering.

An overview of our two-level clustering algorithm is illustrated in Figure 2. The first clustering level involves only a subset of genes having CAs with high CVs. The results of the first level are given as input to the second clustering level. The second level involves all genes in the data set that have a numerical annotation. The output of the second clustering level is the final result of the whole process.

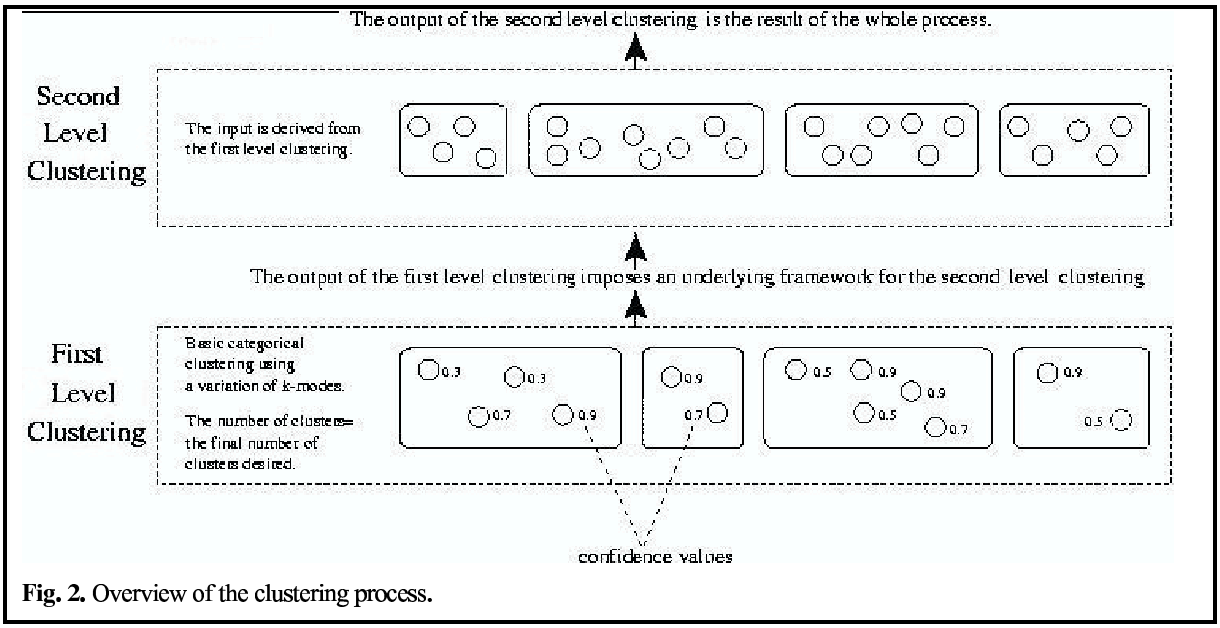

Fig. 2. Overview of the clustering process.

A desirable cluster resulting from our algorithm might have the structure shown in Figure 3. For predicting gene function, it is desirable for our clustering results to have the following format, in order of preference:

a. A cluster should have one or more CAs that appear frequently with high CVs.
b. A cluster should have one or more CAs that appear frequently with high CVs and one or more CAs that appear frequently with low CVs.
c. A cluster should have one or more CAs that appear frequently with low CVs.
d. A cluster should have one or more CAs that appear infrequently with high CVs.

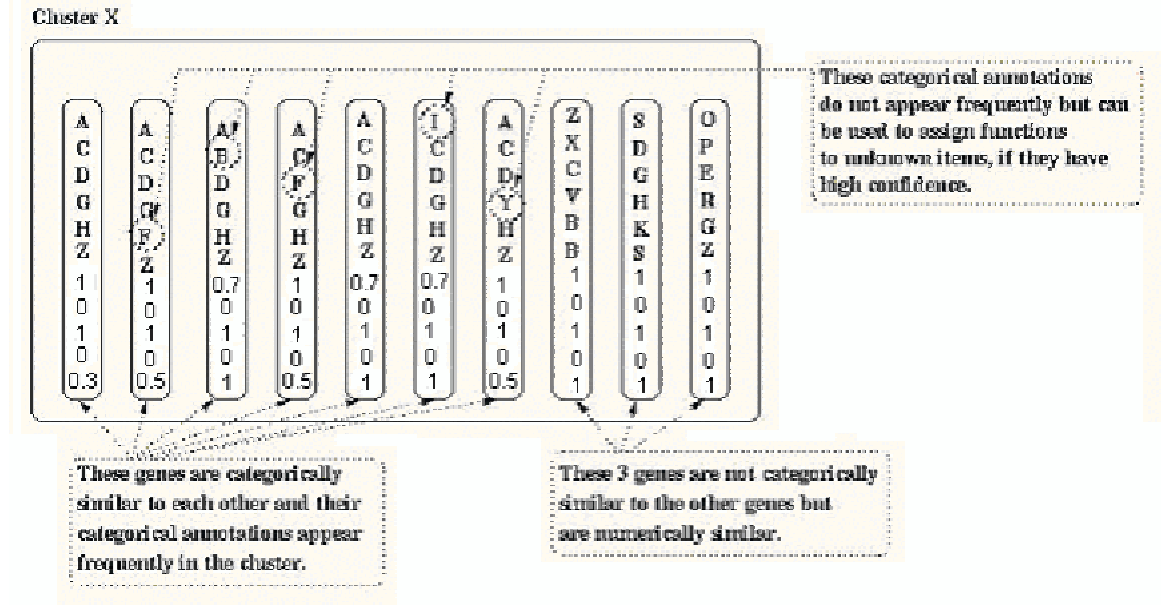

Fig. 3. The format of a desirable cluster resulting from our clustering process.

3.2.1 First level clustering

At the first level, clustering is performed using a variation of the k-Modes clustering algorithm for categorical data sets [15]. The number of clusters resulting from this level equals the final number of

clusters desired, as illustrated by Figure 2. During k-Modes clustering, the following generic loop is performed and most of its steps are redefined in our algorithm:

```
Select the initial K modes for K clusters
Repeat {
       For (all data items) {
              Calculate the similarity between the data item and the modes of all
clusters.
              Allocate the data item to the cluster C whose mode is nearest to
the data item.
              Recalculate C's mode from the data items allocated to it so that
the intra-cluster similarity is maximized.
                     }
} until (few data items change clusters)
```

Each cluster has a mode associated with it. Modes are used to choose the closest cluster to a gene by computing the similarity between the cluster's mode and the gene. In the loop above, the gene is then allocated to the closest cluster and the mode gets updated.

The reasons for using a modification of K-modes are not just that K-modes is a widely used categorical clustering algorithm. Using a mode for each cluster during the clustering process ensures that a cluster will contain many CAs of the same type that have high confidence. At the same time it allows many CAs of the same type that have low confidence to be included and some CAs that have high confidence to be included in the cluster a few times.

A similarity metric is needed to choose the closest cluster to a gene by computing the similarity between the cluster's mode and the gene. In our k-Modes version, we use a similarity metric that also takes into consideration the CVs associated with the CAs, as indicated below.

$$\text{similarity}(x,y) = \sum_{j=1}^{m} \frac{[5 - 4 * \text{confidence}(x_j)] + [5 - 4 * \text{confidence}(y_j)]}{[5 - 4 * \text{confidence}(x_j)] * [5 - 4 * \text{confidence}(y_j)]} * \delta(x_j, y_j)$$

$\delta(x_j,y_j)$ = *1 if* $x_j=y_j$, *0 otherwise.*

This similarity metric gives more importance to high CVs (1.0) than low CVs (0.0) at CAs the values of which are matched between the genes x and y.

Modes always have CVs 4.0 associated with them by default. A mode Q for a cluster is found as follows:

$\sum_{i=1}^{n} similarity(X_i, Q)$ is maximised if and only if frequency(X_j=q_j|X) >= frequency(X_j=c_j|X) for $q_j \neq c_j$ for all j=1 to m. Thus, frequency(X_j=q_j|X) must be maximal for all j=1 to m.

Finally, not all genes participate at the first level. Only a subset of genes with high confidence values – our tests selected 30% of the total number of genes - participate at the first level. The purpose is to maximize the chances that the genes in first level clusters have enough high confidence values, since these lay the framework for second level clustering.

Figure 4 shows that our similarity formula for comparing a mode to a gene assists for all of the goals a-c stated at the beginning of Section 3.2 to be satisfied in order of preference, because the similarity formula increases a gene's likelihood to be attached to a cluster as many CAs match the mode and as the CVs of those CAs increase.

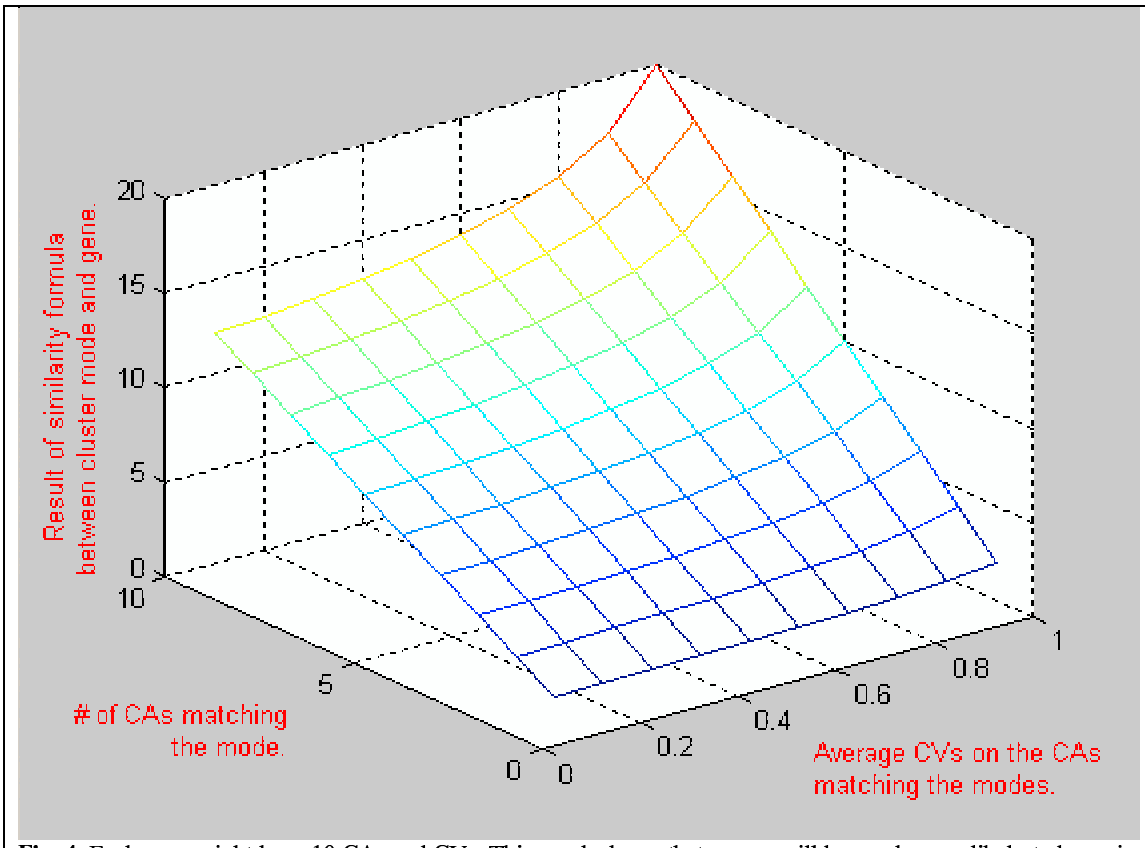

Fig. 4. Each gene might have 10 CAs and CVs. This graph shows that a gene will be much more likely to be assigned to a cluster if all CAs match the mode with high CVs (e.g. 1.0), than if all CAs match the mode with medium CVs (e.g. 0.5), than if all CAs match the mode with low CVs, than if 1 CA matches the mode with high CV, than if 1 CA matches the mode with low CV. This graph was produced with the numerical values for the CVs assigned to the GO evidence codes as described previously.

3.2.2 Second level clustering

The genes that did not participate at the first level participate at the second level. The second level involves a number of subclusters equal to the number of genes clustered at the first level. The subclusters are created on the basis of numerical similarity and categorical similarity between the genes clustered at the first level and all the rest of the genes. The subclusters are merged with one another to form clusters. A number of clusters equal to the final desired number of clusters emerges from the whole process.

Second level step 1

In this step one gene in each first level cluster is set as a *seed*, while all the rest of the genes in the cluster are set as *centers*. The gene to be set as seed is determined by finding the gene the CAs of which have the highest overall CVs. The rationale behind setting the seed of a cluster this way is that the most influential genes at the second level should be those that have the highest confidence values at the first level. Thus, each first level cluster consists of one seed and one or more centers.

Second level step 2

In this step each seed and center that participated at the first level clustering is inserted into a newly created second level subcluster.

Second level step 3

In this step each gene that did not participate at the first level is inserted into the second level subcluster containing the most similar seed or center. Similarity is determined in one of two ways: 1. If the CVs on the gene's CAs are high enough - e.g. the average over all CVs is greater than a specified parameter value such

as 0.6 - then insert into the subcluster with the most similar CAs. 2. If the CVs on the gene's CAs are lower then insert into the subcluster with the most similar numerical attributes, as determined by the numerical similarity metric described by Cherepinsky et al. [4]. Later, in step 5, the action taken for each second level subcluster will be determined based on how many genes were inserted into the subcluster on the basis of high confidence.

Second level step 4

In this step each second level subcluster containing a center is merged with the most numerically similar cluster containing a seed. The most similar seed-containing subcluster is determined by using our own version of the ROCK goodness measure [13] that is evaluated between the center-containing subcluster in question and all seed-containing subcluster:

$$G(C_i,C_j) = \frac{link[C_i,C_j]}{a\ normalization\ factor}$$

$link[C_i,C_j]$ stores the number of cross links between subcluster C_i and C_j, by evaluating $\Sigma[p_q \in C_i,\ p_r \in C_j)\ link(p_q,p_r)$. link($p_q$,$p_r$) is a boolean value specifying whether a link exists between points p_q,p_r . It is computed by determining if the points' numerical annotations are similar enough and if so, setting a link between them. The rationale for using a variation of ROCK's goodness metric for this step is that the link-based approach of ROCK adopts a global approach to the clustering problem, by capturing the global knowledge of neighboring data points between clusters. Thus, it has been shown to be more robust than methods that adopt a local approach to clustering and are susceptible to errors, such as hierarchical clustering [13].

Second level step 5

In this step the loop shown below is performed; this loop refines the second level subcluster, increasing the likelihood that a center-containing subcluster with a majority of genes that were attached to it on the basis of categorical similarity (a prerequisite for which is the gene to have high average confidence over its CAs, according to Step 3) will merge with the subcluster containing the seed with which the center was clustered together at the first level. On the other hand, a center-containing subcluster with a minority of genes that were assigned to it on the basis of categorical similarity should be likely to remain merged with its current numerically similar seed-containing subcluster (see Step 4).

```
Repeat {
  for each center in a Second Level Cluster {
      if the center's cluster at 2nd level contains >70% items merged on the basis of categorical similarity then
        { if the center's cluster at 2nd level is a little numerically similar (>=0.1) to its seed+centers cluster then merge }
      if the center's cluster at 2nd level contains >40% items merged on the basis of categorical similarity then
        { if the center's cluster at 2nd level is moderately numerically similar (>=0.4) to its seed+centers cluster then merge }
      if the center's cluster at 2nd level contains >10% items merged on the basis of categorical similarity then
        { if the center's cluster at 2nd level is very numerically similar (>=0.7) to its seed+centers cluster then merge }
  }
} until (number of center clusters at 2nd level merging with seed clusters < threshold)
```

The rationale for this loop is that if a "center" and a "seed" get clustered together at the first level this means that they have similar CAs and high CVs, but a metric is still needed to specify if this similarity is sufficient to merge their subclusters together at the second level. CVs serve this purpose precisely: the CVs on a cluster's genes' CAs have an impact on whether a center-containing subcluster is merged to its corresponding seed-containing subcluster at the second level. If *many* of the center subcluster's genes have *high* CVs over their CAs, this means that many genes were attached on the basis of categorical similarity (see step 3); then the likelihood of the center subcluster and the seed subcluster merging at the second level *increases*. If on the other hand *few* of the center subcluster's genes have *high* CVs over their CAs, then the likelihood of the subclusters merging at the second level *decreases*.

4 VALIDATION OF CLUSTERING ALGORITHM ON SIMULATED DATA SETS

We generated artificial data sets that simulate the results by Spellman et al, who showed that in each cluster there is a consistent pattern of NAs that appear frequently and that different CAs are characteristic of different clusters [19]. We used numerical gene expression data on the yeast cell cycle, provided by Stanford and analyzed by Spellman et al [6,7,19]. The purpose of our simulation was to assign annotations to each gene based on the numerical gene expression data, in such a manner that the assignment of annotations simulates knowledge about the role of genes in the yeast cell cycle. For example, the first annotation was set for each gene based on the experimental point at which the gene reaches its peak expression level, indicating what cell cycle phase it is likely to be involved in. Furthermore, we perturbed the annotations to simulate noise in the resulting data set. Our aim was to retrieve the known underlying cluster structure effectively. A significant outcome of our experiments was to show that given the genes whose CAs were not perturbed in the simulation (most of which are likely to have high CVs) a fair number of genes were assigned to the correct clusters to which they were categorically similar and were not assigned to the incorrect clusters to which they may be numerically similar. Another significant outcome of our experiments was to show that given the genes whose CAs were perturbed in the simulation (most of which are likely to have low CVs) a fair number of genes were assigned to the correct clusters to which they were likely to be numerically similar and were not assigned to the incorrect clusters to which they were categorically similar.

We generated a data set, in which many annotations were perturbed and the items in the data set had overall low confidence. For this purpose, for each annotation we generated a *limit* (in a range from *0.4* to 1.0) and then generated a random number (from 0.0 to 1.0). If the random number exceeded the limit, then we perturbed the CA by assigning it a value taken randomly from the set of possible values for that annotation. The confidence value for the CA was set equal to the *limit* (regardless of whether it was actually perturbed or not). This simulates the uncertainty that exists on current knowledge and that is expressed in SGD as GO evidence codes [9,17]. In the produced data set 2,024 data items had their original annotations modified, out of 6100. All annotations on all data items were assigned a CV between 0.4 and 1.0 and items whose annotations were modified were likely to have lower CVs than unmodified ones.

We clustered the simulated data sets using our algorithm, into 20 clusters. The results are given in Tables 1, where we ignore clusters whose size was too small.

What is most noteworthy in this table are clusters 8, 11, 15, because all of their data items had their annotations modified during our simulation (see column 4). As can be seen, a vast majority of the data items in these clusters had original annotations on their first 3 CAs consistent with the most representative annotations {A,B,C} for the cluster (see columns 5,6). Furthermore, all of the data items whose first 3 CAs were equal to the most representative annotations {A,B,C} for the cluster, were items whose annotations had been modified during the simulation to different values (see column 7).

Another interesting result is cluster 2, in which the most prominent genes are those with the CA sequence {MG1, E, N} in their first 3 CAs. 202/1102 items had their CAs modified to a totally different annotation, but were nevertheless assigned to the correct cluster because they had low CVs (see column 7). This shows that our algorithm can overcome a poor prior that is likely to be incorrect and can still produce correct results by using numerical clustering instead. In this cluster, from all items assigned to it that had their CAs modified (305/1186, as shown in column 4) 202/305 truly had an annotation of {MG1, E, N} or {MG1, E, O} (see column 5). The total cluster size was 1186 and consisted of 180 merged second level subclusters (see column 8). Four of the merged clusters contained a vast majority of items with modified annotations, and all of these clusters had a substantial portion - or a majority - of items with an original annotation of {MG1, E, N}.

Table 1 - Results for clustering the data set into 20 clusters (we ignore clusters whose size was too small).

Clus#	Number of items in the cluster	Most common annotations {A,B,C} on the items' first 3 CAs	Percentage X of all items that had its annotations modified during the simulation	Percentage of X that had an original annotation of {A,B,C} or very close to {A,B,C}	Percentage P of all items in the cluster that had an original annotation of {A,B,C} or very close to {A,B,C}	Percentage of P that had its annotations modified during the simulation	Number of merged second level subclusters
1	2032	{M,D,L}	616/ 2032	217/ 616	1047/ 2032	217/ 1047	537
2	1186	{MG1,E,N}	305/ 1186	202/ 305	1102/ 1186	202/ 1102	180
3	724	{G2,C,J}	177/ 724	111/ 177	672/ 724	111/ 672	121
4	317	{G1,A,F}	83/ 317	48/83	302/ 317	48/ 302	44
5	709	{S,B,H}	94/ 709	24/94	684/ 709	24/ 684	183
6	218	{M,D,M}	50/ 218	26/50	198/ 218	26/ 198	59
8	66	{MG1,E,N}	66/66	22/66	22/66	22/22	9
11	71	{MG1,E,N}	71/71	27/71	27/71	27/27	3
15	74	{MG1,E,N}	74/74	24/74	24/74	24/24	2
20	333	{S,B,H} and {S,B,I}	180/ 333	148/ 180	260/ 333	148/ 260	13

5 TWO METHODS FOR ANALYZING THE SIGNIFICANCE OF AN ANNOTATION'S CLASSIFICATION IN A CLUSTER

We used the following two methods for determining how significant a CA's classification in a cluster was and the results were used to support the functional prediction process and the validation:

5.1 M-values that take P1-values and confidence values into consideration assigned to each CA in a cluster

Given a resulting cluster, we assigned a P1-value to each CA in the cluster; the term 'P1-value' was derived from the statistical 'P-value'. A P1-value measures whether a cluster contains a CA of a particular type more frequently than would be expected by chance [1]. A P1-value close to 0.0 indicates a frequent occurrence of the CA in the cluster, while a P1-value close to 1.0 its seldom occurrence. We multiplied the resulting P1-value with the reciprocal of the average of all CVs assigned to the CA in the cluster (1/avg(CV)), thus resulting in what we call an *M-value*. M-values allow us to take into consideration the probability that a particular CA occurs in the cluster more frequently than expected by chance, in addition to our confidence that the CA is correct in the cluster. For some CAs that occur only once or twice in the cluster, a high P1-value results with a trivial avg(CV).

5.2 Analyzing the significance of a second level subcluster's classification in a larger cluster

This significance metric was inspired by the loop of step 5 that refines the subclusters composing a larger second level cluster (see Section 3.2.2). Specifically, each subcluster was assigned a significance number by evaluating a formula that considers both categorical (*CA&CVsimilarity)* and numerical (*NAsimilarity)* similarity of the subcluster to the larger second level cluster:

*(weight1*CA&CVsimilarity) + (weight2*NAsimilarity)*

The CA&CVsimilarity for a subcluster is computed by evaluating a categorical variation of ROCK's goodness measure [13] between the subcluster and its larger cluster and multiplying the result by the percentage of genes in the subcluster that were assigned to it on the basis of categorical similarity (see Step 3). The NAsimilarity for a subcluster is computed similarly, by evaluating a numerical variation of ROCK's goodness measure[13] between the subcluster and its larger cluster and multiplying the result by the percentage of genes in the subcluster that were assigned to it on the basis of numerical similarity (see Step 3). Usually, *weight2* in the above formula needs to be significantly higher than *weight1*, for reasons explained later.

The subclusters in an overall second level cluster for which the above metric yields the highest values are used for functional prediction by extracting the genes' CAs in the cluster. We are particularly interested in identifying the CAs with highest avg(CV)s in the cluster, because these are the most reliable ones for functional prediction purposes.

When a subcluster is placed in a larger second level cluster on the basis of high CA&CV similarity (0.5-1.0) - whether it was assigned there at the beginning of the clustering process or joined it later - this is a factor that increases the significance. The NAsimilarity on the subcluster might be:

- high (0.5-1.0) in which case the significance of its membership is increased, because both CAs/CVs and NAs support the gene's classification in the cluster.

- low (0.1-0.4) in which case the significance of its membership is decreased, because CAs/CVs support the gene's classification in the cluster but NAs do not.

When a subcluster is placed in a larger second level cluster on the basis of low CA&CV similarity (0.0-0.4) - whether it was assigned there at the beginning of the clustering process or joined it later - this is a factor that decreases the significance. The NAsimilarity on the subcluster is:

- always high (0.7-1.0), however since the CA/CV similarity was low the significance of its membership is decreased because NAs support the gene's classification in the cluster but CAs/CVs do not.

5.3 Functional prediction for uncharacterized genes

Both of the above significance metrics (SM) were used for functional prediction of genes.

The first SM was used for functional prediction by taking for each cluster the CAs with the lowest M-values for molecular function, biological process, cellular location and for the GOSlim terms. Then, we tried to apply these CAs to genes in the cluster having CAs labeled as 'Unknown'. Therefore, the CAs with the lowest M-values in a cluster were used to predict the cellular role of genes.

The second SM was used for functional prediction in a similar way. We identified the subcluster with the highest significance in a larger second level cluster and identified its genes' CAs with highest avg(CV)s in the cluster, as these were the most reliable ones. Then, we tried to apply the extracted CAs to other elements in the cluster having CAs labeled as 'Unknown'. Therefore, the CAs belonging to the subcluster that had the highest significance were used to predict the cellular roles of genes.

5.4 Validation of predictions using the *first* significance metric

Our strategy for validating the accuracy of the functional predictions is to reclassify certain genes' CAs as 'Unknown' before the clustering process and attempt to predict the correct genes' cellular roles using the cluster CAs pointed out by our metric. The CAs to be set to 'Unknown' were chosen to have a high average(CV) over all their occurrences in the cluster, because these are primarily the ones that we would like to be able to predict correctly. The process described next helps us to determine how likely genes are to be re-assigned their correct CAs.

We iterated over the genes in the cluster with CAs that were labeled as 'Unknown'. To assess the effectiveness of our technique, we verified that the original CAs of these genes correlated better to the cluster CAs with low M-values - that are pointed out by our SM - than those with high M-values. This correlation signified the likelihood that the genes' CAs labeled as 'Unknown' would be re-assigned their original annotations, by using CAs with low M-values that are likely to be pointed out during functional prediction. A relatively large number of genes' CAs labeled as 'Unknown' should be likely to be re-assigned their original annotations using CAs with low M-values in the cluster, because a low M-value indicated that a CA occurred frequently amongst the cluster's genes and also indicated that a CA was likely to be correct. We assumed that the CAs that we were trying to predict had high avg(CV).

We initially clustered the yeast data into 5 clusters. Table 2 gives a description of some CAs that were pointed out in all 5 clusters by our SM, after the CAs with the highest average(CV) in each cluster were set to 'Unknown' and the set was re-clustered.

Table 2 - CAs pointed out in 5 clusters as the most reliable ones.

Cluster	Some of the CAs pointed out in each cluster as having low M-values (meaning they occurred frequently and had high avg(CV)) after the CAs with the highest avg(CV) in each cluster were set to 'Unknown' and the set was re-clustered. All of these CAs were also found to exist in the original cluster with higher avg(CV) than other CAs.
1	vacuolar membrane, ubiquitin-specific protease, small nuclear ribonucleoprotein complex, glycolysis, 3'-5' exoribonuclease, cytosolic small ribosomal subunit, lipid particle, cytosolic large ribosomal subunit, tricarboxylic acid cycle
2	rRNA modification, ATP dependent RNA helicase, nuclear pore, structural molecule, small nucleolar ribonucleoprotein complex, snoRNA binding, mediator complex
3	cytosol, proteasome endopeptidase, non-selective vesicle fusion, translation initiation factor
4	transcription initiation from Pol II promoter, general RNA polymerase II transcription factor, nucleus
5	endoplasmic reticulum membrane, component:endoplasmic reticulum

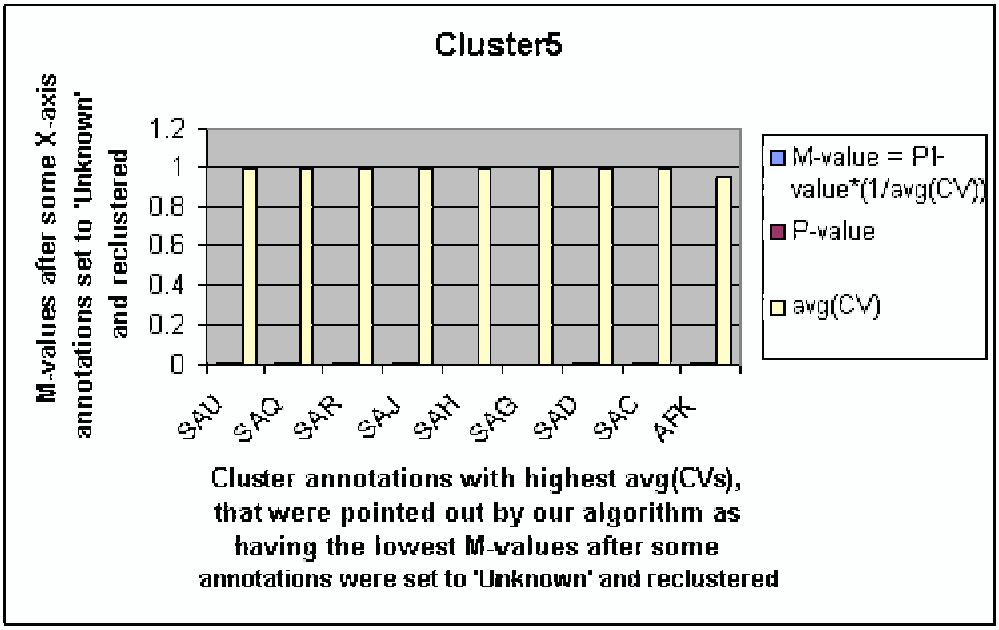

Fig. 5. X-axis: a selection *A* of CAs with highest-avg(CV) that were assigned before the clustering took place. These CAs overlap with a selection *B* of CAs with lowest M-values that were selected for function prediction, after some CAs in A were set to 'Unknown' and the set was re-clustered. Y-axis: indicates for each CA in A, with what M-value, P1-value and what avg(CV) it occurs in B. As shown, the M-values are quite low.

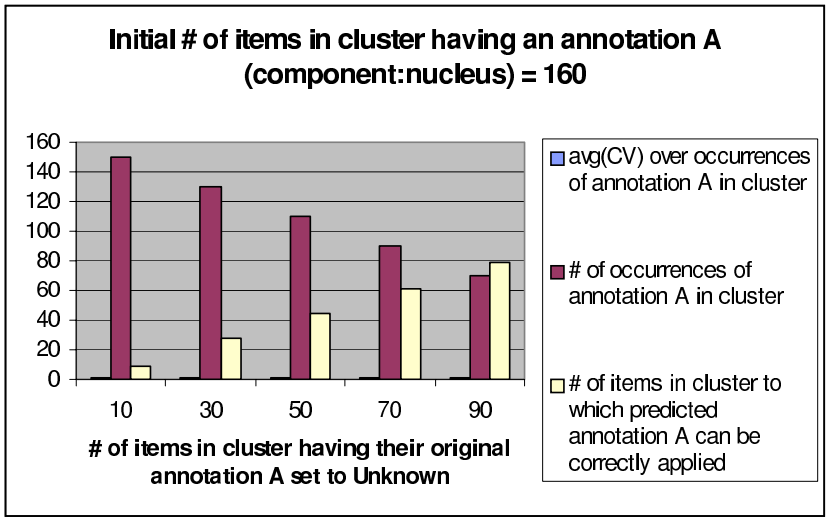

Fig. 6. As an increasing number of genes' CAs in the cluster were set to 'Unknown', the same annotation qualified as one of the most prominent annotations in the cluster and also remained applicable to a relatively large number of genes.

Figure 5 illustrates that the CAs pointed out in cluster 5 as having the lowest M-values - the most representative ones for the cluster - correlate with the original CAs that were set to 'Unknown'. We also performed tests on the yeast data producing 35 and 71 clusters; a summary of the results is given below.

We also performed these tests on the yeast data producing 35 clusters. We provide a concrete example of the utility of our technique for 35 clusters, by focusing on the second cluster having 224 genes. In the original clustering, the following CAs were pointed out as having the lowest M-values:

function:transcription regulator, component:nucleus, process:transport, process:cell growth and/or maintenance, process:metabolism, function:transporter, nucleus(a specific, granular annotation). We focus on the 2 most significant (representative) CAs for the cluster:

1) annotation component:nucleus occurred in 160 genes in this cluster and had an average(CV) of 1.0 across all genes. Some genes containing this annotation were YOR064C, YBR247C, YDR205W, YDR206W, YFR023W, YKL117W, YPR196W, YOR141C, YOL116W, YOR294W, YDR076W, YFR037C, YNL148C, YDR510W, YLR074C, YPL049C, YDL064W, YML109W, YNL016W.

2) annotation nucleus(a specific, granular annotation) occurred in 82 genes in this cluster and had an average(CV) of 0.904878 across all genes. Some genes containing this annotation were YBR247C, YDR205W, YDR206W, YFR023W, YKL117W, YPR196W, YOL116W, YOR294W, YDR076W, YFR037C, YNL148C, YLR074C.

We set the CAs of some genes in which these two annotations occurred originally to 'Unknown' and then re-clustered the data set into 35 clusters. Using the results, we were able to predict correctly that these genes should be annotated as either component:nucleus or nucleus(a granular annotation) by extracting the CAs with lowest M-values. This means that these genes were predicted to have their original correct CAs, after setting them to 'Unknown', re-clustering the data set and extracting the CAs with lowest M-values.

Figure 6 illustrates the detailed results obtained.

We have also performed these tests on the yeast data producing 71 clusters. We provide a concrete example of the utility of our technique for 71 clusters, by focusing on the second cluster having 196 genes. In the original clustering, the following CAs were pointed out as having the lowest M-values: function:transcription regulator, component:nucleus, process:transport, process:cell growth and/or maintenance, process:metabolism, function:transporter, nucleus(a specific, granular annotation). We focus on the 2 most significant CAs:

1) annotation component:nucleus occurred in 161 genes in this cluster and had an average(CV) of 1.0. Some genes containing this annotation were YOR064C, YBR247C, YDR205W, YDR206W, YFR023W, YKL117W, YPR196W, YOR141C, YOL116W, YOR294W, YDR076W, YFR037C, YNL148C, YDR510W, YLR074C, YPL049C, YDL064W, YML109W, YNL016W.

2) annotation nucleus(a specific, granular annotation) occurred in 95 genes in this cluster and had an average(CV) of 0.905263 . Some genes containing this annotation were YBR247C, YDR205W, YDR206W, YFR023W, YKL117W, YPR196W, YOL116W, YOR294W, YDR076W, YFR037C, YNL148C, YLR074C.

We set the CAs of each of the genes in which these two CAs occurred originally to 'Unknown' and then re-clustered the data set into 71 clusters. Using the results, we were able to predict correctly that these genes should be annotated as either component:nucleus or nucleus(a granular annotation) by extracting the CAs with lowest M-values. This means that these genes were predicted to have their original correct CAs, after setting them to 'Unknown', re-clustering the data set and extracting the CAs with lowest M-values.

5.5 Validation of predictions using the *second* significance metric

Our strategy for validating the accuracy of the functional predictions was to reclassify the CAs of certain genes as 'Unknown' before the clustering process and attempt to predict the correct genes' cellular roles using the cluster CAs pointed out by our metric. The CAs to be set to 'Unknown' were chosen to have a high average(CV) over all their occurrences in the cluster, because these were primarily the ones that we would like to be able to predict correctly. The process described next helped us to determine how likely genes were to be re-assigned their correct CAs.

We iterated over the CAs in the cluster that were labeled as 'Unknown' (ones with high avg(CVs)). To assess the effectiveness of our technique, we verified that the original CAs of these genes correlated better to the cluster annotations pointed out as having the highest significance. This correlation signified the likelihood that the genes' CAs labeled as 'Unknown' would be re-assigned their original annotations, by using CAs that were pointed out using our significance metric. A reasonable number of genes' CAs should be likely to be re-assigned their original annotations using the CAs pointed out by our SM. CAs pointed out as highly significant occurred frequently across the cluster's genes with high avg(CV).

We initially clustered the yeast data into 35 clusters, each of which contained a number of smaller subclusters. The second level subclusters pointed out by our significance metric (SM) as significant enough were those containing genes:

1) YHR053C (SM>1 ; 80% of genes not having CV high enough ; 23 genes total),
2) YDL179W (SM>1 ; 96% of genes not having CV high enough ; 104 genes total),
3) YKL182W (SM>0.58 ; 94% of genes not having CV high enough ; 210 genes total),
4) YKR075C (SM>0.44 ; 96% of genes not having CV high enough ; 27 genes total),
5) YLR342W (SM>0.10 ; 91% of genes not having CV high enough ; 22 genes total),
6) YMR246W (SM>0.88 ; 75% of genes not having CV high enough ; 61 genes total),
7) YJL079C (SM>0.06 ; 75% of genes not having CV high enough ; 4 genes total),
8) YCR005C (SM>0.58 ; 63% of genes not having CV high enough ; 470 genes total),
9) YMR186W (SM>0.06 ; 50% of genes not having CV high enough ; 8 genes total),
10) YBR029C (SM>1 ; 0% of genes not having CV high enough ; 1 gene total),

The reason all the other subclusters yielded low significance was because a majority of their participant genes had high average(CV) over their CAs, so most genes were assigned on the basis of categorical similarity - too few or no genes had been assigned on the basis of numerical similarity. Thus the dominant factor in the significance metric yielded zero and the overall result was low.

We next needed to identify the CAs in these clusters with the highest average(CV) throughout the entire cluster. We identified the following CAs, for each of the subclusters listed above:

1) copper binding, avg(CV) 0.5 ; cytosol, avg(CV) 1.0
2) cell cycle, avg(CV) 0.5
3) fatty-acid synthase complex, avg(CV) 1.0 ; fatty acid biosynthesis, avg(CV) 1.0 ; vacuole (sensu Fungi), avg(CV) 0.8 ; vacuole inheritance, avg(CV) 0.8 ; thiol-disulfide exchange intermediate, avg(CV) 0.5 ; plasma membrane, avg(CV) 1.0 ; tricarboxylic acid cycle, avg(CV) 1.0
4) cytoplasm, avg(CV) 1.0
5) 1,3-beta-glucan synthase, avg(CV) 0.55
6) long-chain-fatty-acid-CoA-ligase, avg(CV) 0.55 ; lipid metabolism, avg(CV) 0.75 ; lipid particle, avg(CV) 1.0
7) nuclear membrane, avg(CV) 1.0
8) glyoxylate cycle, avg(CV) 1.0 ; peroxisomal matrix, avg(CV) 0.95 ; folic acid and derivative biosynthesis, avg(CV) 0.95 ; pantothenate biosynthesis, avg(CV) 0.8 ; allantoin catabolism, avg(CV) 0.8 ; purine nucleotide biosynthesis, avg(CV) 0.95 ; helicase, avg(CV) 0.5 ; spore wall assembly, avg(CV) 0.8 ; RAB-protein geranylgeranyltransferase, avg(CV) 0.55 ; protein amino acid geranylgeranylation, avg(CV) 1.0 ; RAB-protein geranylgeranyltransferase complex, avg(CV) 1.0
9) response to stress, avg(CV) 0.75
10) phosphatidate cytidylyltransferase, avg(CV) 1.0 ; phosphatidylserine metabolism, avg(CV) 1.0 ; mitochondrion, avg(CV) 1.0

We set a number of genes containing these CAs to 'Unknown' in each of these clusters, and re-clustered the entire data set. The same annotations were still pointed out by our significance metric as highly significant in the corresponding clusters. This encouraged us to re-assign the original annotations to the genes whose CAs were set to 'Unknown', which we interpret as a success of our approach.

6 DISCUSSION: UTILIZING THE TWO SIGNIFICANCE METRICS FOR DERIVING POTENTIAL GENE FUNCTIONS FOR EXPERIMENTAL VALIDATION

Biologists will find this method useful in wet lab work, for deriving hints about the potential functions of genes and proteins. The hints that are derived as to a gene's function can later be validated experimentally. This will save time and money from the biological experimentalists' side. In our experiments with the yeast cell cycle data set, the utility of the significance metrics is especially evident from the fact that the vast majority of genes in each cluster or subcluster analyzed had all CAs set to 'Unknown' meaning that no knowledge exists. For example, when analyzing the subcluster containing YHR053C using the second significance metric, only 6 out of 20 genes had some kind of CA, while the other 14 genes had CAs set to 'Unknown'. Our significance metric could point out the most representative CAs that are likely to be applicable to the other 14 genes and these functional hints can be tested experimentally.

The *first* significance metric is useful for identifying the most representative CAs in clusters with a plethora of CAs that have *high CVs*. The first metric allows one to identify the CAs in this pool that appear frequently (with a low P1-value) so as to try to apply them to other genes. In our experiments with the yeast cell cycle data set we realised that although 1185 second level subclusters had been produced in total, most of those (1175 = 1185-10 subclusters) had a majority of genes with an average CV over their CAs that was considered high. These genes were assigned to the clusters on the basis of categorical rather than numerical similarity, according to step 3. The first significance metric could be utilized on these 1175 subclusters; or it could be utilized on the overall clusters produced as an end result by our algorithm (as explained in Section 3, the total number of clusters was 5, 35 and 71).

The *second* significance metric applies primarily for identifying the most representative CAs in clusters with a plethora of CAs that have *low CVs*. The second metric allows us to identify the few CAs that have high CVs in these clusters, so as to try to apply them to other genes. In our experiments, only 10 second level subclusters out of 1185 in total, had a majority of genes with an average CV across their CAs that was considered low enough. These genes were assigned to the clusters on the basis of numerical rather than categorical similarity, according to step 3. The second significance metric could be utilized on these 10 subclusters.

7 CONCLUSION

In analyzing genomic data, we need to develop our ability to integrate data from various sources, including proteomic data and genomic data. Furthermore, we need to be able to claim that what we see in an analysis is either reliable or partially reliable and therefore cannot form a strong basis to draw conclusions. In this paper we have described a novel clustering algorithm for DNA microarray data sets that incorporates both annotations on the genes and confidence values on the correctness of the annotations. This clustering algorithm inspired us to define two new significance metrics for extracting from each cluster the most reliable annotations that form a strong basis for deriving conclusions on the functions of genes. We showed that these significance metrics can be successfully used for finding out which annotations are the most promising ones in a cluster and try to apply them to other genes in the cluster. Furthermore, we experimented with our clustering tool on highly noisy simulated data sets for which the correct results were known; we showed that our clustering algorithm can reliably identify the cluster structure in such simulated data sets.

Future work will include applying our algorithm to more DNA microarray data sets for organisms on which uncertain knowledge exists. Furthermore, we will be developing more significance metrics for the combined clustering results. We will experiment with a variation of the algorithm that clusters all data items at the first level, but each cluster is separated into layers - higher layers are more coherent than lower layers. Then the higher layer items in each cluster would form a basis for the second level clustering.

8 REFERENCES

1. Altschuler S.J., Wu L.F., Hughes T.R., Davierwala A.P., Robinson M.D., Stoughton R. (2002). Large-scale Prediction of Saccharomyces Cerevisiae Gene Function Using Overlapping Transcriptional Clusters. *Nature Genetics* 31:255-265.

2. Ben-Dor A., Shamir R., Yakhini Z. (1999) Clustering Gene Expression Patterns. Journal of Computational Biology 6(3/4): 281-297.

3. Brown M.P.S. Grundy W.N., Lin D., Cristianini N., Sugnet C.W., Furey T.S., Manuel Ares, and Haussler D. (2000) Knowledge-based analysis of microarray gene expression data by using support vector machines. PNAS 97(1), 262-267.

4. Cherepinsky V., Feng J., Rejali M. and Mishra B. (2003) Shrinkage-Based Similarity Metric for Cluster Analysis of Microarray Data. PNAS 100(17): 9668-9673.

5. Dwight SS, Harris MA, Dolinski K, Ball CA, Binkley G, Christie KR, Fisk DG, Issel-Tarver L, Schroeder M, Sherlock G, Sethuraman A, Weng S, Botstein D, Cherry JM. (2002) Saccharomyces Genome Database provides secondary gene annotation using the Gene Ontology. *Nucleic Acids Research* 30: 69-72.

6. Eisen, M.B. & Brown, P.O. (1999) DNA arrays for analysis of gene expression. *Methods Enzymol.* 303, 179-205.

7. Eisen MB, Spellman PT, Brown PO, Botstein D. (1998) Cluster analysis and display of genome-wide expression patterns. *Proc Natl Acad Sci USA. 1998 Dec 8;*95(25):14863-8.

8. Fasulo D. (1999) An Analysis of Recent Work on Clustering Algorithms, Technical Report # 01-03-02, Department of Computer Science & Engineering, University of Washington.

9. The Gene Ontology Consortium (2001). Creating the gene ontology resource: design and implementation. *Genome Research* 11: 1425-1433. http://www.geneontology.org/GO.evidence.html

10. Goebel, M. & Gruenwald, Le (1999). A survey of data mining and knowledge discovery software tools. *ACM SIGKDD Explorations* 1, 20-33 .

11. Golub, T. R. et al. (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* 286, 531-537.

12. Grambeier J.,, Rudolph A. (2002) Techniques of Cluster Algorithms in Data Mining. *Data Mining and Knowledge Discovery* 6: 303-360.

13. Guha S., Rastogi R., Shim K. (2000). ROCK: A Robust Clustering Algorithm for Categorical Attributes. *Information Systems* 25(5): 345-366.

14. Hartigan, J. A. (1975) Clustering algorithms. (John Wiley and Sons, New York, 1975).

15. Huang Z. (1998) Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values. *Data Mining and Knowledge Discovery* 2(3): 283-304.

16. Huang, Z. (1997) Clustering Large Data Sets with Mixed Numeric and Categorical Values. *Knowledge discovery and data mining: techniques and applications*. World Scientific.

17. Lord P.W., Stevens R.D., Brass A. and Goble C.A. (2003). Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation. *Bioinformatics* 19: 1275-83.

18. Slonim D.K., Tamayo P., Mesirov J.P., Golub T.R., and Lander E.S.. (2000) Class prediction and discovery using gene expression data. *Proceedings of the Fourth Annual Conference on Computational Molecular Biology (RECOMB),* 263-272.

19. Spellman, P.T. et al. Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces Cerevisiae by microarray hybridization. Mol. Biol. Cell 9, 3273-3297 (1998).

20. Stutz J. and Cheeseman P. (1995) Bayesian Classification(AutoClass): Theory and results. Advances in Knowledge Discovery and Data Mining, 153-180, Menlo Park, CA, AAAI Press.