Clustering of Genes based on Vector Divergence of Expressions

Anca Ralescu and Waibhav Tembe[1]
ECECS Department, University of Cincinnati,
ML 0030, Cincinnati, OH 45221.
e-mail: Anca.Ralescu@uc.edu, wtembe@ececs.uc.edu

1 Introduction

Recently, the advent of the gene chip and micro-array analysis has brought about the need to integrate new computational approaches in the processing and interpretation of the data provided by the gene chip. As a first step in processing such data clustering algorithms have been developed and/or adopted [1], [2], [3], [4], [5]. High data dimensionality and volume call for efficient, yet biologically meaningful approaches so as to support and possibly guide further experimental work.

2 Vector Divergence for Gene Expression

2.1 Problem Definition

We view the problem of clustering as grouping of high dimensional vectors; each of these vectors corresponds to a gene expression data across classes of experiments (e.g. *Saccharomyces Cerevisiae*).

2.2 Assumptions

A cluster of vectors reflects some common property that these vectors share. This property may be expressed directly in terms of the original feature space (observed vector components) or in terms of new (higher level) features calculated from the observed ones. Another equivalent way of looking at a cluster is in terms of a metric, either on the initial feature space, or in a higher level feature space. In this view the vectors in the cluster will share a closeness property. In general, in the absence of a training data set little can be hypothesized about either one of these issues.

2.3 Vector divergence

Given two vectors a and b of dimension n the *divergence* of these vectors, denoted by $D(a,b)$ is defined as a weighted distance between the two vectors. More precisely,

$$D(a,b) = \sum_{i=1}^{n} w_a(i)[a(i) - b(i)]^2$$

and

$$D(b,a) = \sum_{i=1}^{n} w_b(i)[b(i) - a(i)]^2$$

where w_a and w_b are weights depending only on the vectors a and b respectively. $D(a,b)$ is a distance-like quantity which is not symmetric and may not satisfy the triangle inequality. Properties of this divergence are discussed in an extended version of this paper. Here we illustrate the effect of the weight on its calculations:

Let $a = (1,2,3,4,5)$ and $b = (1,1,1,1,1)$ and let us consider that $w_x(i) = x(i)$. Then D(a,b) = 130, while D(b,a)=30.

[1] The order of the authors' names is purely alphabetical. Both the authors have contributed equally to the work.

3 Using Vector Divergence to Define Similarity Measures

Consider a two dimensional array c, where $c(i,e)$ is the expression of i^{th} gene under the experiment e. We define a quantity

$$d(i,e) = \sum_{j=1}^{n} [c(i,e) - c(j,e)]^2$$

$d(i,e)$ can be interpreted as the cumulative discrepancy between the gene i and the remaining genes, along the experiment e. It is obvious that if $c(i,e) = c(j,e)$ for all j, then $d(i,e) = 0$ and more generally, the larger the variability of expression between gene i and the remaining genes in the e^{th} experiment, the larger the value of $d(i,e)$. $d(i,e)$ is next used to compute a measure of *relative difference*, $R(i,j;e)$, between the expressions of genes i and j over experiment e. More precisely,

$$R(i,j;e) = \frac{[c(i,e) - c(j,e)]^2}{d(i,e)}$$

The following example will make the calculation of *R(i,j;e)* more clear.

$$\text{Let } \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} exp & \rightarrow & & \\ genes & 1 & 2 & 3 \\ \downarrow & 1 & 1 & 1 \\ & 2 & 3 & 1 \end{bmatrix}$$

$$\begin{bmatrix} d(a,1) \\ d(b,1) \\ d(c,1) \end{bmatrix} = \begin{bmatrix} 0+0+1=1 \\ 0+0+1=1 \\ 1+1+0=2 \end{bmatrix} \begin{bmatrix} d(a,2) \\ d(b,2) \\ d(c,2) \end{bmatrix} = \begin{bmatrix} 0+1+1=2 \\ 1+0+4=5 \\ 1+4+0=5 \end{bmatrix} \begin{bmatrix} d(a,3) \\ d(b,3) \\ d(c,3) \end{bmatrix} = \begin{bmatrix} 0+4+4=8 \\ 4+0+0=4 \\ 4+0+0=4 \end{bmatrix}$$

$$\begin{bmatrix} R(a,a;1) & R(a,b;1) & R(a,c;1) \\ R(b,a;1) & R(b,b;1) & R(b,c;1) \\ R(c,a;1) & R(c,b;1) & R(c,c;1) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0.5 & 0.5 & 0 \end{bmatrix}$$

$$\begin{bmatrix} R(a,a;2) & R(a,b;2) & R(a,c;2) \\ R(b,a;2) & R(b,b;2) & R(b,c;2) \\ R(c,a;2) & R(c,b;2) & R(c,c;2) \end{bmatrix} = \begin{bmatrix} 0 & 0.5 & 0.5 \\ 0.2 & 0 & 0.8 \\ 0.2 & 0.8 & 0 \end{bmatrix}$$

$$\begin{bmatrix} R(a,a;3) & R(a,b;3) & R(a,c;3) \\ R(b,a;3) & R(b,b;3) & R(b,c;3) \\ R(c,a;3) & R(c,b;3) & R(c,c;3) \end{bmatrix} = \begin{bmatrix} 0 & 0.5 & 0.5 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

3.1 Gene Expression Divergence

The divergence between the vectors forming the rows of the array $c(i,e)$ is now defined as the *average relative difference* over the total number of experiments. More precisely,

$$D(i,j) = \frac{\sum_{e=1}^{E} R(i,j;e)}{E}$$

where E denotes the total number of experiments.
The divergence matrix D for the above example will be

$$D = \begin{bmatrix} & a & b & c \\ a & 0 & 0.33 & 0.67 \\ b & 0.4 & 0 & 0.6 \\ c & 0.57 & 0.43 & 0 \end{bmatrix}$$

3.2 Similarity

The divergence is used to define a non-symmetric *similarity* measure between the genes i and j:

$$S(i,j) = 1 - D(i,j)$$

The similarity matrix S for the above example will be

$$S = \begin{bmatrix} & a & b & c \\ a & 1 & 0.67 & 0.33 \\ b & 0.6 & 1 & 0.4 \\ c & 0.43 & 0.57 & 1 \end{bmatrix}$$

When the expression levels of the genes i and j coincide $S(i,j) = 1$. Based on this we consider that $S(i,j)$ represents the degree to which the gene i should be grouped with gene j.

3.3 Cluster structure

Each member gene in a cluster has a weight to indicate the importance of that gene within the cluster. The weights of all the genes add up to 1 indicating that at each stage, the current genes in the cluster fully describe it. For example, in a cluster with one gene that gene will have the weight 1; in a cluster with two genes the weights may be any two values which add up to 1, and they are calculated based on the quantities in S.

3.4 Cluster Integrity

The weights of genes and the similarity between them are further used to calculate a global value for a cluster, the *cluster integrity*. This value is used to differentiate between clusters not only based on their structure but also based on a qualitative measure of this structure in a way that captures the history of obtaining that particular cluster. Cluster integrity and weights are updated during computations of merging clusters as described below.

3.5 Updating the cluster integrity

When the cluster $C_x = \{(x, w_x)\}$ is merged with the cluster $C_y = \{(y, w_y)\}$ the resulting cluster C_{xy} has the integrity

$$I_{C_{xy}} = \frac{\sum_x \sum_y S(x,y)(w_x + w_y) + \sum_y \sum_x S(y,x)(w_x + w_y)}{2}$$

The algorithm seeks to group vectors under the constraint of optimizing two conditions:

1. maximum similarity
2. maximum cluster integrity

To implement these conditions, the algorithm uses a simple instance of fuzzy inference, encoding the rule of thumb *highly similar vectors belong to the same cluster*. The fuzzy set *highly similar* with membership function $\mu_{highsimilarity}$ is given by the equations below and as shown in the figure 1. By changing the threshold values we can generate different membership functions.

$$\begin{aligned} \mu(x) &= 0 && x < threshold1, \\ &= 1 && x > threshold2, \\ &= \frac{x - threshold1}{threshold2 - threshold1} && threshold1 <= x <= threshold2 \end{aligned}$$

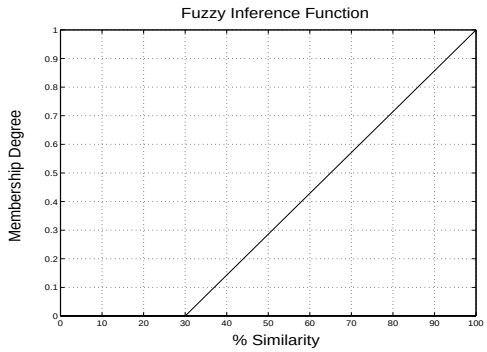

Figure 1: Fuzzy Inference Function (threshold1 = 30 and threshold2 = 100)

For a given pair of genes (i, j), the smallest of the quantities $S(i, j)$ and $S(j, i)$ is evaluated into this fuzzy set. The result, $0 \leq \mu_{highlysimilar}(S(i, j) \wedge S(j, i)) \leq 1$, is the degree to which these similarity measures satisfy the antecedent of the rule. This degree is then taken as the strength of the grouping of genes i and j.

If this degree is greater than a preassigned threshold α then the genes are grouped subject to the additional condition that the integrity of the cluster thus formed does not fall below another preassigned integrity threshold β. Both of these thresholds are subject to a tuning, or trial-and-error procedure. The values α and β control the extent to which the cluster merging will occur and the extent to which dissimilar genes are allowed to merge.

For our example the above procedure yields the following : We start with three clusters each having one gene with weight equal to 1 and the cluster integrity will be 1 (i.e. 100%). Since there are 3 clusters we have 3 possible pairs $(1, 2)$, $(2, 3)$, $(3, 1)$. Each pair is evaluated as shown below using the similarity matrix:

$$\begin{bmatrix}(1,2)\\(2,3)\\(1,3)\end{bmatrix} = \begin{bmatrix}\mu(min(\frac{(1+1)*S(1,2)}{2}, \frac{(1+1)*S(2,1)}{2})) = \mu(min(67, 60)) = \mu(60) = \frac{60-30}{100-30} = 0.4285\\ \mu(min(\frac{(1+1)*S(1,3)}{2}, \frac{(1+1)*S(3,1)}{2})) = \mu(min(33, 43)) = \mu(33) = \frac{33-30}{100-30} = 0.0428\\ \mu(min(\frac{(1+1)*S(2,3)}{2}, \frac{(1+1)*S(3,2)}{2})) = \mu(min(40, 57)) = \mu(40) = \frac{40-30}{100-30} = 0.1428\end{bmatrix}$$

Since initially the weight of the each gene is 1, the terms (1+1) appear in the above matrix. In general the "rank" of merging clusters C_1 and C_2 will be given by

$$\mu(min(\frac{\sum_x \sum_y S(x,y)(w_x+w_y)}{number\ of\ genes\ in\ C_1+number\ of\ genes\ in\ C_2}, \frac{\sum_y \sum_x S(y,x)(w_x+w_y)}{number\ of\ genes\ in\ C_1+number\ of\ genes\ in\ C_2}))$$

Let $\alpha = 0.4$ and β=60 for the above example. Then the algorithm will proceed as follows :

1. The pair $(1, 2)$ gets selected for merging because its rank is maximum $(0.4285 > \alpha)$
2. Only one pair of clusters can be merged in one iteration.
3. The resulting cluster's integrity will be $\frac{67+60}{2} = 63.5 > \beta$.
4. Both the conditions for merging have been satisfied so 1 and 2 are merged forming a new cluster with members = $gene_1$, $gene_2$ and integrity = 63.5.
5. The weights of the genes in the cluster are distributed in the proportion $S(1, 2) : S(2, 1)$. So $weight_{gene_1} = 0.473$ and $weight_{gene_2} = 0.527$.

6. Steps 1 to 5 are repeated for the cluster which is newly formed and the remaining clusters.

7. Merging will continue while the constraints (α and β) are satisfied.

4 Initial Results

- **Results on the synthetic data**: The results of the above algorithm on synthetic data are very promising and accurate. The synthetic data is generated as follows: Let us assume that we have m genes and n experiments. The genes belonging to the same cluster should have similar expression levels across the experiments. We group the genes randomly (so that we know the solution before running the algorithm and we can verify if the solution found by the algorithm is indeed correct). For each gene we assign a vector of n randomly generated numbers taking care that the genes in the same group will have similar expression levels across the experiments. A sample synthetic data is shown below. We assume m=10 and n=6 because of space constraints. The groups in the dataset are [0], [1 2 3], [4 5], [6 7 8 9]

expers →						
genes	−3.000000	−16.571428	−19.250000	12.409091	−6.000000	−16.611111
↓	9.540000	16.552631	8.428572	3.466667	−6.533333	−11.462963
	9.533334	16.541666	8.444445	3.461539	−6.532258	−10.500000
	9.541667	17.458334	7.558824	3.409091	−7.466667	−10.558824
	−18.590910	8.466666	−13.500000	−14.531250	−8.611111	19.357143
	−18.576923	7.500000	−13.333333	−15.428572	−9.447369	18.534483
	−4.558824	3.531250	8.447369	−13.590909	20.459999	3.500000
	−5.462963	3.533333	8.000000	−14.461538	19.555555	3.590909
	−5.423077	4.467742	7.590909	−13.538462	20.452381	4.467742
	−4.531250	3.540000	7.558824	−14.458333	19.540001	4.450000

  The algorithm correctly identifies the genes (row vectors) which have very high similarity and groups them.

- **Results on the real data**: When working on the real data set [3], we ignore all the vectors for which some expression values are not available. Further we analyze the data on experiment by experiment by basis. We identified following four categories of experiments : Alpha, Cdc15, Cdc28 and Elu. The names have been derived from the actual names of the experiments mentioned in the dataset. Since we discard the genes having empty data values, the actual number of genes considered is less than 6178, the actual number of genes in the dataset. For computational reasons we have split the genes into groups of 50 and have identified the genes having similar expressions. We note the following:

  1. The minimum value of cluster integrity β controls the extent of grouping. As we go on decreasing β more genes tend to get grouped in the cluster
  2. By changing the fuzzy inference rule the grouping of the genes can be controlled. The liberal approach will allow relatively more genes to cluster as compared to the conservative approach

- **Visualization of Clusters** : To examine the correctness and structure of the clusters obtained we have implemented a simple visualization strategy. Based on the maximum and minimum expression level in the dataset, we assign ten different letters to ranges of expression levels. So every gene in the cluster will be represented as a sequence of letters. Those genes which fall in the same cluster should have their alphabet representation similar to each other, differing at very few positions. Following are the examples of the clusters obtained when the algorithm was run on the dataset for Elu type experiment for a sample of 50 genes, when $\beta = 99.7$ and we use the linear fuzzy inference function in figure 1.

$$[UTSUUTUTUVTVTS]\ \begin{bmatrix} PRUUUUUVUVUUTU \\ QSTTVVUUUVUUTT \\ SRTTUTUVUUUTTU \\ SSSSUTTUVVTUUU \\ RTSTUUUVUUTUTU \\ SSTTTUVUUUUVTU \end{bmatrix}\ \begin{bmatrix} RRSUTUWVVUTURT \\ RRTSUWVVVVTUTT \\ RTRTUVVVVUSUTT \end{bmatrix}$$

$$\begin{bmatrix} USTTUWUUUTVSST \\ TTTTTUTTUUUUUS \\ USUTTVTUUTUTTT \\ TTTTTUUTUTVTUT \end{bmatrix}\ \begin{bmatrix} USUUVSTTUTSUTU \\ UTUUTTTTUUUUSU \\ VSTVUTTTUTTUTU \\ VUVUTUTSTTTTTT \\ VUUUUTSSTUSUSU \\ UTUUUUTTTUSTTU \end{bmatrix}\ [RSSTSRTUVWUWVV]$$

$$\begin{bmatrix} VUTUUSTTSUTTTU \\ UUUTUUTTRUTTTU \\ UTUUUUUUTTTTSU \\ VUUTTUUTTTUTSU \\ VUTUUTUTSTUTTU \\ TTTUTTUUTTUTTU \end{bmatrix}\ \begin{bmatrix} UTUSTTTTUTUUUU \\ TUSUSTUUUUTUTU \\ UTSTTUTUUTTVUT \\ UUTTSTTTUUSUUU \end{bmatrix}\ \begin{bmatrix} XUVSSSSTTTTUUU \\ WUUTSSSTTTTUTV \\ XUUTSTTTTTTUUT \\ YTTSTTTTTTTUUT \end{bmatrix}$$

$$\begin{bmatrix} SSUSSTTWUUTUUU \\ TUSSSTTUVUUUVV \\ TTTSSSTVVUTVVU \end{bmatrix}\ [TVSRRRTUWUUVVU]\ \begin{bmatrix} UTSSTSTTVVSWVU \\ VTSSSTSTVUUVTV \end{bmatrix}$$

$$[STSUUUUTTTUVSU]\ [QTXSSSTUUUTVUV]\ [VTTTSSSVUVSUUV]$$
$$[UTTUTTSUTSZTST]\ [QXUUUUUTSTUTST]\ [PQQSUWXWWVVVTR]$$
$$[PRURSVVXWVVVTS]\ [SYSSVTUUSTTTTT]\ [RRSSTUUVVVUVVU]$$

$$\begin{bmatrix} WXWXXXXWWWWWWW \\ WWXWXWWXXXWXWX \\ WWWWWXWXXXWXWX \\ XWXXWWXWWWWXWW \\ WWWXXXWXXWXXWW \\ VWWXXXWXXXXWXWX \\ XWWXXXXXXWWXWW \\ WWWXXWXXXXXXWX \\ WWXWXXXWWXWXWX \end{bmatrix}\ \begin{bmatrix} WWWXXXXXWXXWWXW \\ XWWXXXWXWXXXWWWX \\ WWWXXXXXWXXWWWW \\ WWWXXXXWWXXWWXW \end{bmatrix}\ \begin{bmatrix} VWWWWWWWXXXXWXXX \\ XXVXWWWWXXXXWXXX \end{bmatrix}$$

$$\begin{bmatrix} WWYWWWWXWXWXXX \\ WWWXWWWXXXXWXWX \\ WWWXWWWXXXXWXXX \\ WWWXWWWXXXXWXXX \end{bmatrix}\ \begin{bmatrix} WXXXXXXXXWWXWWW \\ WXXXXXXXXWWYWWW \\ WXWXXXXXXXXWXWWW \end{bmatrix}\ \begin{bmatrix} WWWXXXXXWXXXWXW \\ VWWXWXXXXXXWXXX \\ VWWXXXXXXWXXXWW \\ VWXXXXXXXXXXXWW \end{bmatrix}$$

$$\begin{bmatrix} XXXWXWWWXWWXWX \\ WXWWWWXXXXXWXXX \\ XXWWWWWXXXXWXXX \\ XWWWWWWXXXWWXXX \\ XWWWWWWXXXXWXXX \\ XWWWWXWXXXXWXXX \\ WWWXXWWXXXWWXXX \\ WWWXXWWWXXXWXXX \end{bmatrix}\ [WWWYWWWXXWWXWW]\ \begin{bmatrix} VWXWXXXXXXXWWWW \\ WWXWXXXWXXXXXWXW \\ WWWWXXWXXXXXXXW \end{bmatrix}$$

$$[WVWWWWXXXXXXXXX]\ [ZWXXXXXWXWWWWW]\ \begin{bmatrix} UWXXYXXXWXWWWW \\ WWWXXXXXXXXWWWW \end{bmatrix}$$

$$\begin{bmatrix} XXXXXWXWWWWWVX \\ XWXXXWXWWXXVVW \end{bmatrix}\ [VXPYXXXYYYXYXX]\ [XXXXWWWWWWWWWW]$$

$$\begin{bmatrix} XXYXWWWWWWWXWX \\ XXWXWWWWXXWXXX \end{bmatrix}\ [WWWYYXXWWXWVWW]\ [XWWXXWXXXWWXWX]$$

$$\begin{bmatrix} SRTSSSTTUUSTPT \\ RRTSSSUSTTTTRT \end{bmatrix} \; [UTTTRQRSTSQUVU] \; \begin{bmatrix} USTUSTSSTTSTRS \\ SRTTSTTSTUSTRT \\ TRTTSTTSTTSTQS \end{bmatrix}$$

$$[TSTSSTRTTTSTTS] \; \begin{bmatrix} RSSTTTTSTTSTST \\ SSSTTSRTTTSTTT \\ SSSSTSTTTTRTTT \\ SSTRTSSTTSSTTS \\ TSSSTSTSTTSTTT \end{bmatrix} \; \begin{bmatrix} VSTTTTUSRSSRRS \\ TTTTUTTTSSRSRS \end{bmatrix}$$

$$\begin{bmatrix} PSTTSSTTTTTTST \\ RSSTSSSTTTSTTT \end{bmatrix} \; [TRTUSSSTTTRSTR] \; [PSUTUTTTSSSSRT]$$

$$\begin{bmatrix} QRSUTSTTTTSTSS \\ QRSUUTTTTTSSSS \\ QSTTUSTTTTSTSS \end{bmatrix} \; [RTUTSTTSTTTTPT] \; \begin{bmatrix} TRSTSTSTTTTSST \\ STTTTSSSSTSTST \\ USSTTTSTSSSSTS \\ TRSTTSSTSSTTSS \\ TSSTTSTTSSSTSS \\ TSTSSSTTSSSTST \end{bmatrix}$$

$$[TSSSSRRRTTRTVU] \; \begin{bmatrix} PRSSUUUTTTTSST \\ RRSRTTUTTTTTST \end{bmatrix} \; [ZUTSRQRSSRSTTT]$$

$$\begin{bmatrix} RSRSSTTTUTSTUT \\ SRRSSUTTTTSTSS \\ TRSTSTTTTTRTSS \end{bmatrix} \; \begin{bmatrix} TQSSSSSSTTTTTT \\ TRRTSSTSTUSUST \end{bmatrix} \; \begin{bmatrix} RRRSSSSTUUSTUT \\ PRSSTTUTUTSTST \\ QRRTTSSTTTSTTT \end{bmatrix}$$

$$[WSSTSRSTSRRTST] \; [STRTSSSSTTSTST] \; [STSSSSTTTSSUSS]$$

$$[TSRTRSUSSSTTTS] \; [TSTSTUSSTSTSTR] \; [USSTRSSSUTRTTT]$$

$$[UTTTUSSSSSRTRT] \; [URSTTSSSTTSTTT] \; [USTSSRSTTSRTTT]$$

$$[PSUSSTUSTUSTST]$$

$$\begin{bmatrix} RTTVVUUSVVTUUU \\ PTRUUVVUVVTUUU \end{bmatrix} \; \begin{bmatrix} PPPRUVXXXWVWUS \\ SPPSTWWWWVUVTS \end{bmatrix} \; [UUVTUUUVUVTQTU]$$

$$\begin{bmatrix} VSTSSTSUVUUVVU \\ SSUUTTTVUVUVUV \\ VSTSTTUVUUTUVU \\ TTTSUVTUVVTVUT \\ SSTTTUUVUVUUUU \\ TTUTUSSUVVTVUT \\ TUTSTTUUUVUUTU \\ TUTUUTTUTUTUUV \\ USUTUUTUUUUUUU \\ USTUUSUUUTTVUU \\ TSTUUTUVVUSVUU \\ UTTTUTTTVVTVUT \\ TTUTTUTUVVTUTU \end{bmatrix} \; \begin{bmatrix} VTVUTUVTUUTUPU \\ TUVTTVVTTUTUST \end{bmatrix} \; \begin{bmatrix} XUTUSRRTUUSVWW \\ VTTUSSTTUTSVVW \end{bmatrix}$$

$$\begin{bmatrix} RSUTUTVWUUSUTT \\ RTUUVVUVUUTTTS \\ STTUUUUUUUUTU \\ STTTVUUUUVUUTT \\ SSTUVTUUUVUUTT \end{bmatrix} \; \begin{bmatrix} VVUTTUUTTTSTUU \\ UUVTURTTUUSUTU \\ VUTTTSTUUUUUUV \\ VTUTUTTTUTTUUU \\ VUUUUSTUTTSUUU \\ XTTTUSTUUTTUTU \end{bmatrix} \; \begin{bmatrix} UTTSUUWTUTVTTT \\ USUTUTUSVUUUVT \\ SUUUUUVTUUUUTS \\ UTTTUUUTUTUTUT \\ TTVTTUUTUUUTUT \\ UTTTUUUUUTTUTT \end{bmatrix}$$

$$\begin{bmatrix} WTVUVTTSTTRTTU \\ TTUVUTUTUUSUSU \\ UUTVUSTTUUTUTU \end{bmatrix} \; [TUXUTUTUUURTST] \; \begin{bmatrix} QSSUVVUVUWUTUT \\ SRSTUVUVVVTVTT \end{bmatrix}$$

$$\begin{bmatrix} SUSTSSSVVVTVVV \\ USTSSTRUWUTWVV \\ USTSTTUUVVSWVU \\ STSTTTUUVUTVUU \end{bmatrix} \; [QPQPQSSVXYYZXW]$$

5 Future Work

We are currently working on analyzing the clusters obtained by summarization and description.

- Summarization : The weights of the genes in the cluster should give an insight into the cluster structure. One way so summarize a cluster would be to represent it using the most important genes which may be termed as the *core* members.
- Cluster Validation : The clusters obtained thus need to be validated by making sure that the genes found in the same clusters are indeed similar.
- Singletons Separation : It is interesting that many genes need not be similar to others at all. In that cases these genes remain unclustered. We refer to these genes as singletons.
- Dimensionality Reduction : Identifying those experiments whose expression levels do not change the clustering can help in reducing the dimensionality of the problem. One way to identify such experiments is by using the values for $d(i, e)$.
  For example if $\sum_{i=1}^{n} d(i, e)$ or $max_{i=1,n} d(i, e)$ is small then it is likely that the experiment e does not affect the clustering by a large extent.

References

[1] M. B. Eisen, P. T. Spellman, P. O. Brown and D. Botstein. **Cluster Analysis and Display of Genome-wide Expression Patterns**. PNAS Vol. 95, 14863-14868, December 1998.

[2] E. Hartuv, A. O. Schmitt, J. Lange, S. Meyer-Ewert, H. Lehrach, R. Shamir. **An Algorithm for Clustering cDNA Fingerprints**. Genomics **66** 249-256 (2000).

[3] P. T. Spellman *et al.*. **Comprehensive Identification of Cell Cycle-regulated genes of the yeast *Saccharomyces cervisiae* by Micro-array Hybridization**. Molecular Biology of the Cell, Vol. 9, 3273-3297, December 1998.

[4] Y. Cheng and G.M. Church, **Biclustering of expression data**, ISMB2000.

[5] Y. Cheng, **Clustering with competing self-organizing maps**, International Joint Conference on Neural Networks, Baltimore, pp. IV-785-790, 1992.