Paper SA07_05

Clustering Analysis of Micro Array Data

John Schwarz, University of North Carolina, Chapel Hill, North Carolina

ABSTRACT

One such area of genetics research that has developed a great deal in the past several years has been micro arrays. Advancements in genetic research have often led to increased amounts of data but without efficient analyses techniques. The purpose of this project is to provide some examples of statistical methodology that can be used to analyze data from micro array experimentation. The data gathered from the micro array experimentation is contained in a 12,489 by 42 sized table. There are several different clustering techniques available to analyze the data. Kernel density estimation was used to compare the densities of different genes. Clustering methods used include hierarchical and least squares methods. With the hierarchical models, different distance parameters are determined and the results compared. The least squares method attempts to reduce the time it takes to cluster the data. The results from each method will be compared and different advantages and disadvantages will be discussed.

INTRODUCTION

Every living organism is made up of a single cell or many groups of cells. The identity of these different cells and the function of these cells are determined by genes. Genes are segments of DNA providing the code for producing proteins. Different organisms contain different numbers of genes. For example, a human being contains 30,000 genes as estimated and the fruit fly contains only about 13,000 genes. The identification of genes depends upon the knowledge of the DNA sequence made up of different alleles. Alleles are the different forms of the gene and cause different physical properties in different individuals. The human genome has recently been fully mapped. However, not every gene has been fully expressed in the DNA alleles. Genes can often be hidden within compacted DNA strands and are not easily identifiable.

Genes also interact in different ways; some have similar properties and responsibilities while other genes may be totally different. Genes that are different, however, may be involved in the same reactions inside a cell and, equivalently, genes that do not have similar properties may not be involved in the same reactions. Micro arrays offer an efficient method of comparing multiple genes quickly and easily. The analysis of micro arrays offers substantial evidence of genes that may or may not be related in a cell. The different cycles of cell replication and division express certain genes. These cycles are important in the understanding of cell behavior. The technologies can be used to identify the onset of certain genetic related conditions.

BACKGROUND

Gene expression is important in cellular identification and gene function. With new technologies and research, gene expression and identification have become an ever growing area in biotechnologies with the opportunity for new, more efficient analyses. The field of cellular genetics has shown that changing pH and temperature causes certain genes to be expressed while others are not expressed. It is possible to alter these settings in a lab so that the expressed genes can be identified. Most genes are known by the proteins they produce and by the function of these proteins. Common human cells have similar structure to that of figure 1. Nerve cells and other similar structural specified cells have different structures than those of the majority of cells in the human body. The main structures concerned with DNA replication and protein production process are the nucleus, ribosome and rough endoplasmic reticulum. The nucleus contains the DNA and is where the unpacking and copying is completed. Any copies of the DNA, which is the mRNA, are transported out of the nucleus and to the ribosome. The ribosome is where proteins are produced in the cell from the mRNA. Protein production can also occur in the rough endoplasmic reticulam.

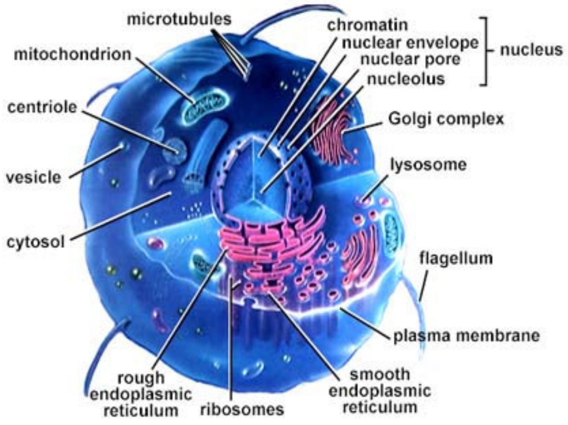


Figure 1- Cell Structure (Brazma, p. 1)

Gene expression can be broken down into several different stages and identified at any time during one of the specific stages. The stages range from gene to functional protein activity. The first stage in expression is the unpacking of chromatin DNA. Chromatin is the complex of DNA and proteins that makes up a chromosome (Figure 1). In chromatin form, DNA is closely packed and regulatory proteins used in the transcription process are often unable to access certain portions of the DNA; and consequently, certain genes are not expressed. The differences in chromatin

packing vary in each different cell type and thus inhibit or allow the expression of certain genes.
Even though entire DNA strands exist in chromatin form at times, they can be unpacked to allow access to certain strands for protein production and replication. Methylation occurs, meaning the process when the DNA is unpacked and methyl groups (CH_3) groups are placed on the ends of the DNA strand for the gene being used, and identifies the start and finish of the gene to replication compounds used for protein construction. Transcription occurs and the DNA is copied into RNA to change certain alleles, forms of genes on different location, and is then copied to mRNA for protein production outside the nucleolus. The mRNA is moved to the cytoplasm where protein translation is accomplished. Polypeptide (polymer chain in which amino acids are linked together with the peptide bonds) groups are created from the mRNA code and, after cleaving and modification, are known as proteins. From this stage, the proteins are sent to certain areas of the cell for purposes identified by the type of protein. This whole process is known as the gene expression, and any stage of this process can be used to identify the gene expression (Campbell 2002).

Different cells have different roles in the body and in turn, produce a number of different proteins. By learning which conditions activate certain genes and deactivate other genes, more can be understood about certain cellular identities (Campbell 2002). Cells generally have the ability to regenerate and reproduce themselves. Cell functions are marked with the expression of certain genes. The understanding of when and how these processes take place, and more importantly, why some cells do not go through this process can better be understood through gene expression micro array analysis.

The main objective in micro array analysis is to identify gene expression and the comparison of several genes at once. The process of identifying and comparing the genes expressed in a cell or culture is a complex process resulting in large amounts of data. The process can be simplified into six main steps (Figure 2):

1. Selecting the cell culture for analysis
2. Identifying the specific DNA gene sequence
3. Radioactively tagging the DNA sequences
4. Hybridization of the array (Hybridization is the process in which the fluorescently tagged cDNA is applied to the array)
5. Laser intensity readings from the plate
6. Interpreting the results of the hybridized array

Of course, only fully sequenced genes can be used in this experimentation since known DNA sequences are hybridized to an array of thousands of different genes at one time. The idea behind creating this array is to identify genes at certain points in an expression and to isolate conditions for the certain genes to be expressed.
There are several different types of micro array experimentations. The first type is gene expressions using DNA, which uses the DNA sequences of specific genes that are applied to an arrayed, cultured plate with the goal of identifying genes in the cells and comparing their interactions. Initially, the mRNA is used in this experiment. The mRNA, messenger RNA (synthesized DNA), is used in the production of proteins. Recall mRNA is copied directly from DNA in the cell and when the mRNA is copied, different gene alleles are used that differ from normal DNA. The mRNA sequences are known to be unstable and to deteriorate after a short amount of time, making them useless for hybridization reactions. For this reason, cDNA (DNA copied from mRNA with enzyme reverse transcription) is copied from the mRNA for its ease of use and compatibility (Fortina 2000).

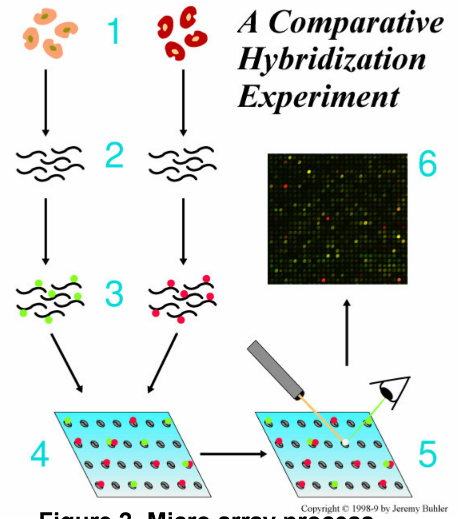

A Comparative Hybridization Experiment
1
2
3
4
5
6
Copyright © 1998-9 by Jeremy Buhler

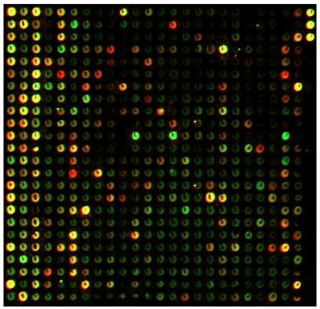
Figure 2- Micro array process (Buhler 2003, p. 1)

The cDNA, complementary DNA, is copied from the mRNA and uses the original alleles as DNA. The cDNA is then tagged with fluorescent markers and applied to an array of cellular cultures. Fluorescent identification can be used to understand evolutionary traits across species.

The creation of an array plate generates thousands of closely contained DNA segments. The DNA segments are placed within a small region on a glass slide and glued to prevent washing away during the gene reaction process (Campbell 2002). The DNA segments are placed in a grid pattern for identification once the hybridization reaction has taken place. The use of certain genes depends upon what is known about the organism. Some organisms have fully sequenced genes so that all the genes can be used in the expression. Organisms that have a great number of genes may require the use of several array plates since not all the genes can fit on a single array.
Continuing with the other steps involved in micro array, the analysis will be described using the techniques of gene expression micro arrays. Hybridization is the next step to micro array.

“In addition, realization of such developments will facilitate early diagnosis and evaluation of treatment strategies. The array is template-dependent and involves primer extension by a radioactive- or dye-labeled dideoxynucleotide terminator (ddNTP) with the tag of the incorporated base revealing the identity of the template complementary nucleotide imme-diately 3' to the primer. Solution-phase extensions followed by electrophoresis on gels as well as microtiter plate-based assays have been described (Fortina 2000, p. 884)." Orphan line

Each spot on the array contains more than one sample strand of DNA. The ability for more than one interaction at each spot is possible. Thus, the intensity of the fluorescent marking can differ from spot to spot and cannot be used to determine how accurately the strength of the binding is to the sequence. The different sequences are marked with different colored fluorescent tags. Using different colors allows for multiple detection of gene expression (Figure 3). The different colors can determine similar genes and gene interactions. Following the hybridization process, a washing stage is carried out. The purpose of the washing is to eliminate any fluorescently labeled sequences that did not hybridize to gene spots. That is why it is important to secure the spots before the hybridization process. Securing hybridization spots allows for a clearer interpretation of which gene spots hybridize with the applied marked gene sequences.



plate(Leming 2003, p. 1)

The fluorescent tags can be identified by computer analysis. Since hybridization can occur on only a single sequence of DNA in each spot, visual detection cannot be accurate. Furthermore, the fluorescent tags used in the hybridization process cannot be visually identified unaided. The tags used require certain frequencies provided by laser treatment after hybridization and washing has occurred. Consequently, there are thousands and thousands of data results from each micro array hybridization experiment. Simple visual analysis in most cases is time consuming and inefficient. Analysis by computer is the most efficient and accurate option. For this reason, different statistical methods must be employed to understand the relation between different genes. Several different programs exist currently designed to analyze the large sets of data produced by the micro array process (DeRisi et al. 1997). Some of these programs include the simple frequency counts from each of the different experiments and genes. Other methods include simple statistical tests to understand small portions of the large array.

METHODS

Kernel Density

The first step in the analysis was to examine the properties of each gene that reacted with the micro array plate. The method used for modeling the data was kernel density estimation (PROC KDE). Two-dimensional graphical representations of kernel density estimators resemble a smoothed histogram. The kernel density estimator approximates a probability density function allowing for specific values to be accentuated. The kernel density estimator uses all the values in the data. The kernel density method differs from the histogram because data are separated into certain sections that allow for the maximum value to be emphasized, not just the section in which this maximum occurs. When dealing with such large data sets for visual analysis, certain portions of the data are shortened by boundary adjustments. The adjustments allow for the most prominent portions of the data to be observed clearly. The equation used for kernel density estimation is as follows:

$$\hat{f}(x) = \frac{1}{n a_n} \sum_{j-1}^{n} k\left(\frac{x - X_j}{a_n}\right)$$

The value n is the sample size for the dataset, a_n is a constant based on the sample size and the known population density. Inside the summation, k is a known density function, and x is any value within the domain of the density function f.

The kernel density results can be analyzed visually to allow for preliminary relationships to be determined. Since this process is visual, optimal graphical representations must be achieved through bandwidth adjustment. Several different methods to adjust the bandwidth are available through SAS, and each method produces somewhat different results. For determining the best bandwidth adjustment, the different methods are tested individually at different percentage levels.

In SAS, kernel density estimation is performed using the KDE procedure, which returns output in table form. The table lists 401 rows of the density function, data position, and count. From the table, the kernel density model can be plotted and then visually examined. The specified SAS code is as follows:

```
Proc kde data=_proj_.John gridl=0 gridu=954 method=SROT
bwm=1.00 out=outkde1;
Var CJ_BaP1;
Run;
```

The first section of the code designates procedure. Next, the lower and upper bounds are declared for the data and the method of bandwidth adjustments is specified. The second line begins with the percentage for the bandwidth adjustment and finishes with the location of the output file where the analysis will be stored. The third line defines the variable that is analyzed.

For the micro array data set, the method of bandwidth was Silverman's Rule of Thumb Method. This method is modeled by the following equation:

$$h = 0.9 \min[\hat{\sigma}, (Q_1 - Q_3)/1.34] n^{-1/5}$$

In this equation, n is the sample size and $\hat{\sigma}$ is the sample standard deviation. The Q values represent the first and third quartiles. This method proved visually optimal for the data set at 100%, which gives a smooth representation of the data. Multiple graphs allow for a simple, distinct comparison. The lower bound was set at zero and the upper bound was determined based on the analysis of each individual variable. Each upper bound had the same density endpoint value.

The resulting kernel density analysis produced a series of tables with three columns and four hundred rows plus one. The first two columns, value and density, were used to plot solutions for graphical analysis. The graphs from each different variable could then be combined, and simple visual interpretations made. Each single variable plot was similar in pattern, but maximum regions, a frequency from 2000 to 3000, often differed from variable to variable. Figure 4 is an example of a plotted graph using the variable of CJ_BaP1.

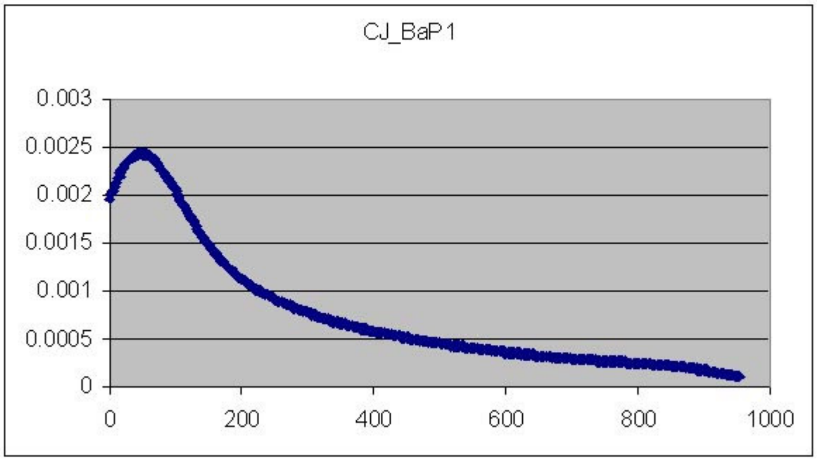


Figure 4-Gene CJ_BaP1 Kernel Density Plot

There were several different methods involving the comparison of the separate graphs. The first was to plot all of the variables on one graph. The resulting plot is illustrated in figure 5. The y-axis represents the probability density and the x-axis defines the frequency of the light recorded. The boxes indicate the different concentrations of genes.

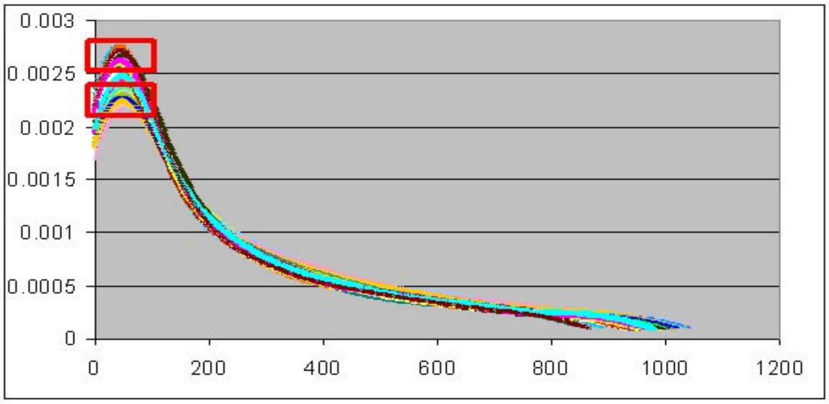


Figure 5-Comparison of all genes using kernel density

Upon visual analysis, it can be determined that there are two major concentrations of variables that represent different genes run through the experiment, and several outliers. The upper concentration is made up of 26 gene variables. The lower concentration is made up of 14 variables and the remaining 2 variables look to be outliers. From this point, several different comparisons were made. The concentrations represent possible groups of related genes. First, each concentration was examined more thoroughly by removing the other variables from the plot. This was repeated several times, and smaller concentrations throughout the group could be examined individually. An example of a smaller concentration is illustrated in figure 6, and the clusters within are identified by the boxes on the plot.

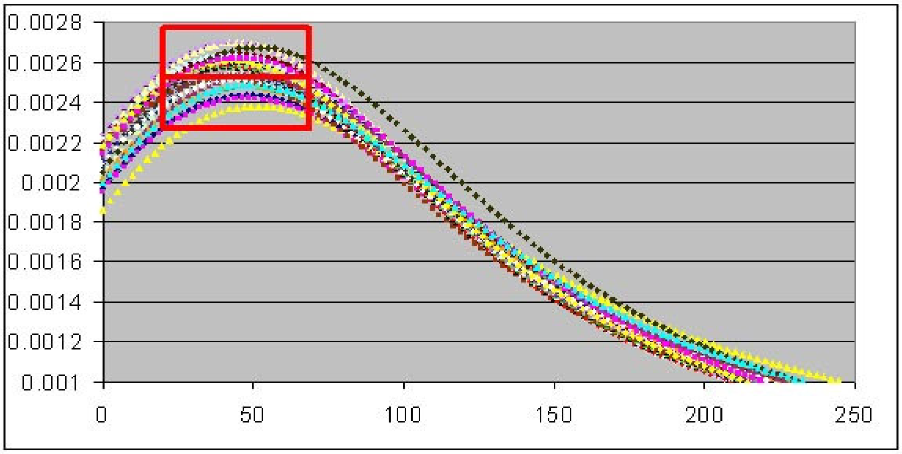


Figure 6-Upper concentration of genes analyzed

Similarly, the analysis was repeated for the lower concentration, and again, smaller clusters could be identified and interpreted. An example from the lower concentration is illustrated here (figure 7), and the smaller clusters within the lower cluster is identified by the boxes.

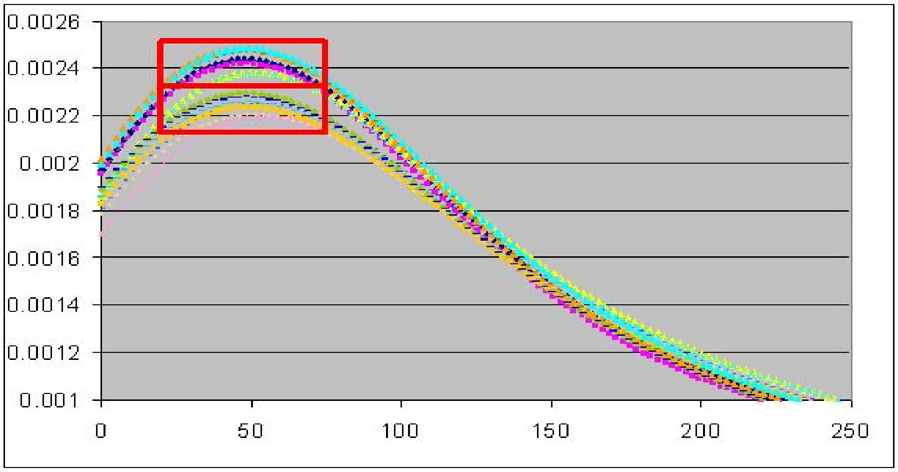


Figure 7-Lower concentration of genes analyzed

Next, an examination of the difference in the concentrations and outliers was undertaken. Several variables were selected from each cluster nearly at random, and plotted to observed differences in position and behavior. Plotting variables against one another helped to distinguish outliers and cluster groups. Figure 8 is an example of this analysis.

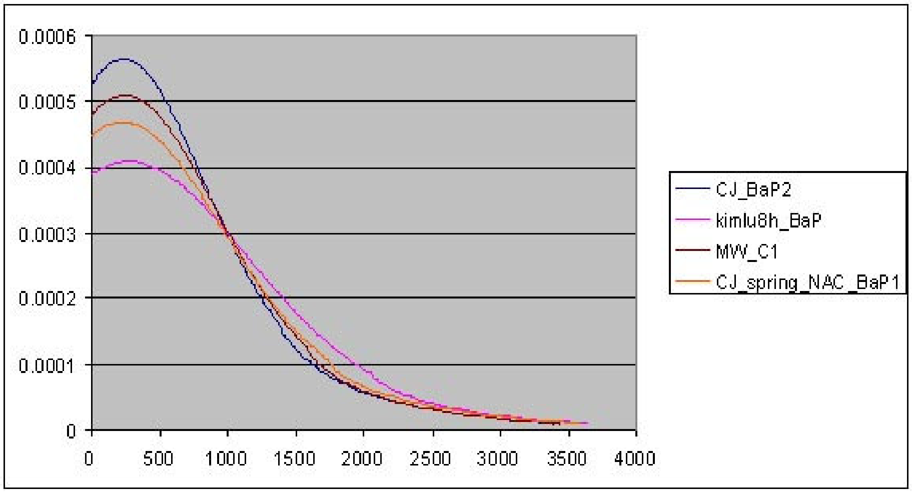


Figure 8- Example of genes that differ in expression

The difference in concentrations was the initial step to understand gene relationships, and which genes could possibly be associated with one another. Using this information, the individual clusters could help identify more exactly which genes had a higher probability of being related to one another. This is demonstrated in figure 9.

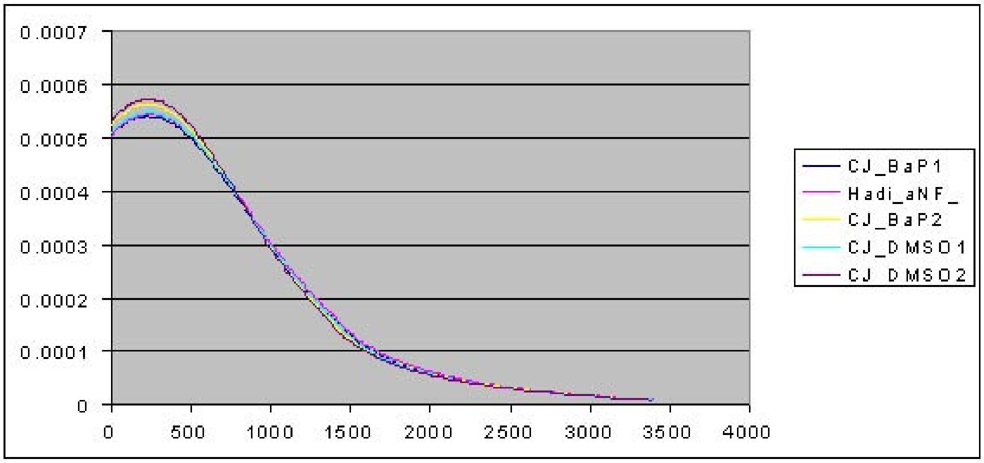


Figure 9-Example of genes that are similar in expression

Proc Cluster

Additional clustering methods were examined to compare to the kernel density clustering process. The first method used for comparison was proc cluster done through SAS. Originally, the data had the form where the genes were the columns and the array sites were the rows. For PROC CLUSTER, the data were first transposed so that the rows were now the genes and the columns were the array spots. This formatting was required for the cluster procedure in SAS to identify clusters based on the results from the micro array experimentation.

The SAS cluster procedure allows for selection of several different models. The choice of the model often depends upon what form the data are in, and upon what type of clustering is to be used. For the micro array data, the most appropriate model choice was the maximum-likelihood hierarchical clustering model. Hierarchical modeling uses pair wise similarities to identify variable relationships (Hastie et al. 2001). Maximum-likelihood hierarchical clustering follows a tree diagram with successive branches and leaves producing more and more clusters. The number of generations performed by the algorithm determines the format of the tree and where the branch divisions occur. From this tree, the number of clusters can be chosen. The fewer the clusters, the lower the square of the correlation coefficient will be. As the cluster choice approaches more clusters with fewer elements in each cluster, the square of the correlation coefficient approaches 1. Hierarchical clustering has been shown to be effective in several different micro array analysis experiments (Alon et al. 1999, Eisen et al. 1998, Gasch et al. 2000, Wen et al. 1998). In recent studies, clustering based on different hierarchical approaches has been one of the most popular methods utilized (Eisen et al. 1998, Michaels et al. 1998, Ramoni et al. 2002).

The size of the micro array data was so immense that a random sample had to be obtained to reduce the size of the data set. The way micro arrays chips are designed, the first several columns contain highly expressive and frequently occurring genes. Therefore, the chance of interaction is increased in the first several columns. The last several columns, however, contain rare genes and less frequently occurring strands of DNA that do not have as high a chance of reaction during experimentation. These last columns are used to exclude genes that may have the same reactions on the high frequency genes, but may not be related. The stratified sample was defined differently for each part of the micro array chip. From the first 2000 genes, 400 were chosen by means of a random sample selection. From the remaining 10,000 genes, 800 were chosen by the random sample procedure. The resulting 1200 micro array positions were able to be analyzed more efficiently and in less time.

The SAS code for clustering allows the user to determine the length of duration and several other factors to fit the model exactly to the data. For the micro array data, the first step was a standardization of the data to a mean of 0 and a standard deviation of 1. The purpose was to eliminate the variation across different gene spots and allow for the experimentation results to be compared to one another in the same range. The SAS code is as follows:

```
proc cluster data=sasuser.TRANSPOSED_TOTALSAMPLE outtree=sasuser.Tree method=EML
    standard
    ccc pseudo print=15;
    var a104058_at a101207_at a100900_at …a103846_;
    id Label;
run;
```

The first line of the code designates the procedure and the data source. Next, the output tree is defined along with the clustering method and the standardization procedure. The ccc statement defines a table for the last generations in the clustering algorithm to be displayed. The variables for the clustering are then defined along with the variable containing the label for the data.

The table produced of the last 15 generations can allow for determination of the optimal number of clusters. The table used the cubic clustering criterion and the f and t^2 statistics. By comparing these values and their fluctuation points, the optimal number of clusters can be determined.

Table 1
Cluster procedure iterations

Cluster History											
NCL	**Clusters Joined**		**FREQ**	**SPRSQ**	**RSQ**	**ERSQ**	**CCC**	**PSF**	**PST2**	**LNLR**	**LNLIKE**
15	**CL21**	**MW_T6_4**	3	0.0130	.787	.	.	6.6	1.2	3022	-101E3
14	**CL26**	**CL34**	4	0.0135	.774	.	.	6.8	1.7	2952	-104E3
13	**MW_T1_5_2**	**CL22**	4	0.0145	.759	.	.	7.1	2.4	2984	-107E3
12	**CL18**	**CL31**	7	0.0150	.744	.	.	7.4	2.2	2895	-11E4
11	**CL19**	**HADI_BAP_A**	6	0.0152	.729	.	.	7.8	1.7	2769	-113E3
10	**KIMLU8H_BA**	**KIMLU8H_DM**	2	0.0158	.713	.	.	8.3	.	2712	-116E3
9	**CL20**	**CL13**	10	0.0188	.694	.	.	8.8	2.2	3041	-119E3
8	**CL15**	**CL12**	10	0.0200	.674	.603	3.23	9.5	2.2	3034	-122E3
7	**HADI_FALL_**	**HADI_FALL1**	2	0.0235	.651	.569	3.70	10.3	.	3341	-125E3
6	**CL9**	**CL8**	20	0.0283	.623	.530	4.18	11.2	2.8	3716	-129E3
5	**CL16**	**CL10**	8	0.0467	.576	.486	4.09	11.9	4.4	5594	-134E3
4	**CL14**	**CL5**	12	0.0668	.509	.432	3.08	12.4	4.7	7005	-141E3
3	**CL7**	**CL11**	8	0.0691	.440	.361	2.81	14.5	5.6	6313	-148E3
2	**CL4**	**CL3**	20	0.1561	.284	.249	1.11	15.1	8.0	11777	-16E4
1	**CL2**	**CL6**	40	0.2839	.000	.000	0.00	.	15.1	15972	-176E3

The table represents the last 15 generations of the proc cluster algorithm. The frequency defines the number of clusters identified in a specific generation. Statistics for each generation determine the strength of the correlation coefficients for each generation including the partial f test, PSF, and the partial t^2 test, PST2. The LNLR and LNLIKE describe the linear distances of the clusters. Fluctuation points between generations determine the optimal number of clusters to be used. A number of different correlation constants are also included in the table. The first correlation constant, SPRSQ, compares the r squared values between two clusters compared to the overall r squared. The RSQ and ERSQ are the overall r square of the populations for a particular generation and an estimation of specific r squared values respectively. The table output only gives indication about the last 15 generations and what the possible optimal cluster values are. In some cases, a single variable is listed in the space for the cluster identity. The problem with interpretation of the iterations is in the clustering procedure, multiple comparisons on the same gene are made during each clustering step leading to incorrect verification data (Levenstein et al. 2003). The tree diagram produced by the hierarchical clustering is more effective to determine the number of optimal clusters.

The tree diagram from the hierarchical clustering procedure can be obtained from SAS code. Using the results from the table output, a specific number of clusters can be determined in the code when extracting the tree diagram. For the micro array data, four clusters were defined to compare against the kernel density results.

Determining the optimal number of clusters can be tedious using different procedures and techniques. Using hierarchical clustering makes it difficult to determine the true number of clusters in the data (Dudoit and Fridlyand 2002, Horimoto and Toh 2001). The actual number of optimal clusters is specifically defined by the data present(Levenstein). The SAS code is as follows:

```
goptions vsize=8in htext=1pct htitle=2.5pct;
axis1 order=(0 to 1 by 0.2);
proc tree data=sasuser.Tree out=New nclusters=4
graphics haxis=axis1 horizontal; height _rsq_;
copy a104058_at a101207_at … a103846_;
id _name_;
run;
```

The first lines define the graphical output in SAS. The tree procedure line is where the number of clusters is defined, and the source of the data. The other pieces of code define more graphical options as well as the axis options. The axis was defined to be the r- squared values of the clusters.

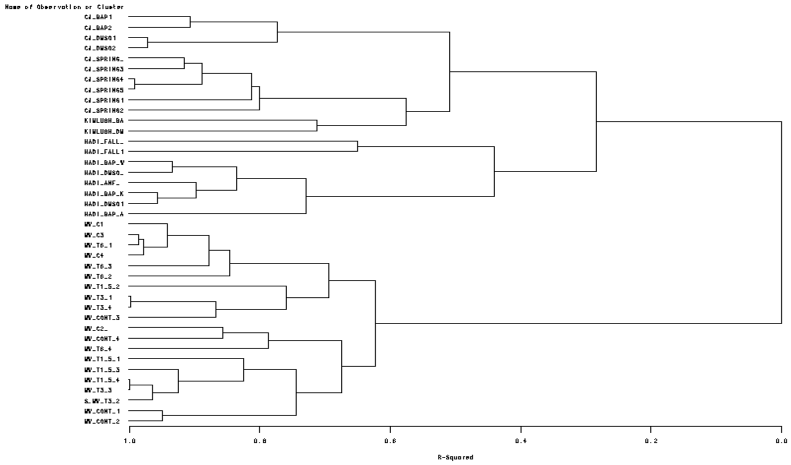
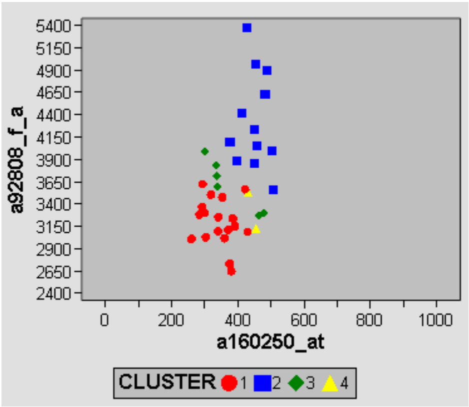


Figure 10 - Cluster procedure tree output

The tree diagram shows the gene locations in each cluster as well as where the cluster designations are made with respect to their r-squared value. By defining the number of clusters in the code, plots of the genes on array spots can be generated to observe the definition of the cluster. Each plot is unique, so the generation of every plot is inefficient and time consuming. By producing a few plots, a general idea of the clusters can be identified. The SAS code for the cluster plots is as follows:

```
legend1 frame cframe=ligr cborder=black position=center value=(justify=center);
        axis1 label=(angle=90 rotate=0) minor=none order=(2400 to 5400 by 250);
        axis2 minor=none order=(0 to 1000 by 100);
proc gplot data=New;
        plot a92808_f_a*a160250_at=cluster/frame cframe=ligr
        legend=legend1 vaxis=axis1 haxis=axis2;
run;
```



The micro array positions being plotted must be defined as well as the data source. The plots show two positions on the array and how the genes reacted. Observe in Figure 11 the plot of two array reactions for the spot a92808_f_a on the vertical axis and the spot a160250_at on the horizontal axis. The figure shows the clear separation of each cluster. This separation indicates a clear designation of the clusters. Other plots with other array spot comparisons may not have such desired results. The cluster procedure produced four clusters to be compared to the kernel density clusters. The procedure also produced a great deal of information on the clusters to identify optimal placement. There were also methods to determine the validity of the clusters and ways to visually observe their behavior according to the data. Clustering only indicates what possible expression relations may exits, but does not prove what relationships are present (Gibbons and Roth 2002).

Figure 11 – Cluster rsults for a92808_f_a and a160250_at

Enterprise Miner Clustering

The third method of clustering was the procedure provided in SAS Enterprise Miner. The Enterprise Miner system is a graphical interface system to perform data mining. The analysis in Enterprise Miner is constructed from a system of connected nodes. Each node designates a different procedure or process in SAS. The nodes are linked and are run successively in a linear progression. Each node has different options to specify the process or model to the data. Adjusting the model allows the results to be more accurate according to the properties of the data.

The cluster node contained the process to cluster the data. However, several preprocessing nodes had to be employed before the clustering node could be used. The first node used was the data input node. This node designates the source of the data, and whether variables are to be used in the calculations. The data input node is required in almost every Enterprise Miner procedure. The second node was the data transformation node. Again, data random sample of the data was used. Recall that the first 400 array spots in the sample were randomly obtained from the first 2000 highly occurring array spots, and the remaining 800 were randomly sampled from the other 10,000 less frequently occurring genes.

The next node used in the calculations was the data transformation node, which allowed for new variables to be defined from the original dataset. Again, each variable can be designated whether or not to be used in the calculations. However, the main reason for the data transformation node was for the data standardization option. The variables were standardized with mean 0 and standard deviation of 1 to allow for the comparison of the genes in the same numerical region. The model for standardization is a best fit process based on the values in each variable.

The cluster node was the third and final node in the process. In the cluster node, the variables are defined to be a label or a variable included in the calculations. Next, a method of clustering must be chosen. For the micro array data, the method chosen was least squares (fast). The fast least squares method only performs one iteration and then takes the minimum distances squared of the variables. Thus, the method takes relatively little time. Following the model selection, the number of clusters can be chosen as a specific number or can be optimized by the model. For the micro array data, the number of clusters chosen was 5. The processes were then run in succession with the cluster node being the final procedure.

The clustering node produces several different results. When using the least squares method, the minimum distance is used to determine cluster bounds. The results depict the differences in the distances in a map to indicate how well separation of clusters occurs. Figure 13 shows the different distances in the clusters and their sizes. The cluster point labeled 4 shows a single point, marking it as an outlier. Cluster 4 is the reason 5 clusters were designated in the cluster node (figure 13). When only 4 clusters were designated, three clusters were formed overlapping and there remained an outlier. The diagram shows a clear separation of clusters with just a slight overlap of clusters 5 and 3. Two of the clusters were of similar size and were smaller than the larger two. The clustering results initially appear to be similar to the original kernel density clustering. A more detailed profile of the clusters can be obtained. A clustering profile includes size, clustering information as well as several statistical measures.

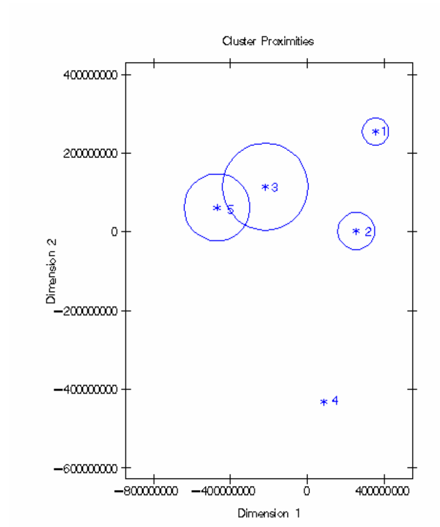


Figure 13- **Basic cluster structures**

Table 2 Enterprise Miner clustering properties

CLUSTER	Frequency of Cluster	Root-Mean-Square Standard Deviation	Maximum Distance from Cluster Seed	Nearest Cluster	Distance to Nearest Cluster
1	12	1311365.748	68915057.95	2	285910508.6
2	7	1960485.047	95382803.37	1	285910508.6
3	16	2738717.385	222260470.8	5	256821568.6
4	1	-	0	2	471591265.2
5	4	3986273.403	170980823.6	3	256821568.6

From Table 2, the overall structure and layout of the clusters can be determined. Properties about the clusters can be used to determine how well the clusters are formed and how unique each cluster profile is from one another. Using the clustering statistics, the clusters can be compared to kernel density clusters. The additional information from Enterprise Miner clustering can be used to verify the results from the kernel density with statistical significance. These cluster statistics can be compared to the cluster profiles and the results from the cluster procedure. Comparison to the other methods can confirm the results from each process and how well each cluster is defined.

Results:

The clusters identified by the kernel density process were then compared with the other clusters identified by the two other methods. The first method of comparison, PROC Cluster, identified four clusters as well. The sizes of the clusters, however, differed slightly from the clusters identified by the kernel density process. The kernel density procedure produced clusters of size 14, 12, 7 and 7 whereas the PROC Cluster results were of size 12, 8, 10 and 10. The Enterprise Miner solutions which produced 4 clusters with a single outlier, differed from the kernel density results as well. The sizes of the clusters produced from the Enterprise Miner system were 12, 7, 16 and 4. The composition of the clusters must be identified to determine how well each cluster matched up to each other. The largest size of a cluster from one method does not always indicate that it is the same cluster as the largest cluster from another method. To determine which clusters were the same from each method, the location of the genes had to be identified. By comparing each cluster from each method, the similarities could be identified and in turn, the clusters could be matched up between methods.

Table 3
Contents of Cluster 1

KDE	proc Cluster	EM cluster
CJ_BaP1	**CJ_BaP1**	**CJ_BaP1**
CJ_BaP2	**CJ_BaP2**	**CJ_BaP2**
CJ_spring_BaP	CJ_DMSO1	CJ_DMSO1
CJ_spring_DMSO	CJ_DMSO2	CJ_DMSO2
CJ_spring_NAC+BaP1	**CJ_spring_BaP**	**CJ_spring_BaP**
CJ_spring_NAC+BaP2	**CJ_spring_DMSO**	**CJ_spring_DMSO**
CJ_spring_NAC1	**CJ_spring_NAC+BaP1**	**CJ_spring_NAC+BaP1**
CJ_spring_NAC2	**CJ_spring_NAC+BaP2**	**CJ_spring_NAC+BaP2**
kimlu8h DMSO	**CJ_spring_NAC1**	**CJ_spring_NAC1**
MW_C3	**CJ_spring_NAC2**	**CJ_spring_NAC2**
MW_CONT_1	kimlu8h BaP	kimlu8h BaP
MW_CONT_2	**kimlu8h DMSO**	**Kimlu8h DMSO**
MW_T1_5_1		
MW_T3_1		

Table 3 shows genes in bold type that are located in cluster one for each method. The clusters produced from the proc cluster method and Enterprise Miner method are identical and all but three genes in these clusters are contained in the kernel density cluster. These results show a high overlap of genes and indicate 9 genes that could be identified within cluster one. The remaining genes that occur in each method that do not agree with one another would be subject to more analysis and tests to indicate whether or not they belong in a particular cluster. It is concluded that each method created a very similar cluster designated cluster 1.

Table 4
Contents of Cluster 2

KDE	Proc Cluster	EM cluster
Hadi_fall_BaP	**Hadi_fall_BaP**	**Hadi_fall_BaP**
Hadi_fall_DMSO	**Hadi_fall_DMSO**	**Hadi_BaP_WT**
Hadi_BaP_WT	**Hadi_BaP_WT**	**Hadi_DMSO_WT**
Hadi_DMSO_WT	**Hadi_DMSO_WT**	**Hadi_BaP+aNF**
Hadi_BaP+aNF	**Hadi_BaP+aNF**	**Hadi_aNF**
Hadi_aNF	Hadi_aNF	**Hadi_BaP_KO**
Hadi_BaP_KO	**Hadi_BaP_KO**	Hadi_DMSO_KO
	Hadi_DMSO_KO	

Table 4 displays the results from the clustering methods designated cluster 2. Again, there most genes (6) are contained in the cluster by all three methods. Notice that PROC cluster and Enterprise Miner clustering, have begun to diverge from one another. In cluster 1, the results were identical to one another; however in cluster 2, there are a couple of differences between the clustering methods. The results still indicate the majority of genes to be designated cluster 2. Again, further analysis would determine the proper location of the remaining genes.

Table 5
Contents of Cluster 3

KDE	proc Cluster	EM cluster	
CJ_DMSO1	**MW_C1**	**MW_C1**	s-MW_T3_2
CJ_DMSO2	MW_C3	MW_C2	MW_T3_3
MW_C1	MW_T6_1	MW_C3	**MW_T3_4**
MW_T3_3	MW_C4	MW_C4	MW_Cont_1
MW_T3_4	MW_T6_3	MW_T6_1	MW_Cont_2
MW_T6_2	**MW_T6_2**	**MW_T6_2**	MW_Cont_3
S_MW_T3_2	MW_T1#5_2	MW_T6_4	
	MW_T3_1	MW_T1#5_3	
	MW_T3_4	MW_T1#5_4	
	MW_Cont_3	MW_T3_1	

The results from the different methods show a great deal of variance for cluster 3 as seen in table 5. Not only do the sizes vary, but the composition of each of the clusters also differs considerably from one method to the next. Only 3 genes are the same with each method, which is not the majority in any of the clusters identified in each method. Therefore, different methods are needed to further identify the genes located in cluster 3.

Table 6
<u>Contents of Cluster 4</u>

KDE	proc Cluster	EM cluster
Hadi_DMSO_KO	MW_T6_4	MW_T6_3
Kimlu8h BaP	MW_T1#5_1	MW_T1#5_1
MW_C2_	MW_T1#5_3	MW_T1#5_2
MW_CONT_3	MW_T1#5_4	**MW_Cont_4**
MW_T1_5_2	s-MW_T3_2	
MW_T1_5_3	MW_T3_3	
MW_T1_5_4	MW_Cont_1	
MW_T6_1	MW_Cont_2	
MW_T6_3	**MW_Cont_4**	
MW_T6_4	MW_C2	
MW_C4		
MW_CONT_4		

The results from cluster 4 are similar to those of cluster 3. There is very little overlap; in fact, only one gene exists in all three clusters. With only one gene overlap between each of the three methods, it is difficult to determine the size or composition of cluster 4. The cluster 4 identified by kernel density and cluster 4 identified by PROC cluster were more closely related to one another than that of Enterprise Miner. More testing and analysis would need to be performed to determine the actual composition of cluster 4.

CONCLUSION

The results show many similarities between the clusters designated cluster 1 and cluster 2 by all three methods. The third and fourth clusters were not as clearly defined as were clusters 1 and 2. These differences indicate that the decision to extract 4 clusters may not be the optimal solution. Recall the micro array data were sampled for the clustering methods, PROC Cluster and Enterprise Miner clustering, but the kernel density method utilized all of the micro array data. Additional clustering trials and adjustment of the methods could lead to stronger support of the kernel density results. Another method of verification used to identify the precise composition of the clusters would be biological interpretation. Certain gene relationships were identified and can be used to verify the clustering methods. The next step of the analysis would be to identify what gene relationships are known and whether the clusters agree with this information. This identification could determine the exact composition of the third and fourth cluster where the models did not agree on the structure.

REFERENCES

1. Alon U, Barkai N, Notterman DA, Gish K, Ybarra S, Mack D, Levine AJ. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Science* 96, 6745-50.
2. Brazma A, Parkinson H, Schlitt T, Shojatalab M. (2003). *Quick introduction to elements of biology* (on-line). Available: http://www.ebi.ac.uk/microarray/biology_ intro.html.
3. Buhler, J. (2003). Anatomy of a comparative gene expression study (on-line). Available: http://www.cs.wustl.edu/~jbuhler/research/array/.

4. Campbell NA, Reece Jane B. (2002). *Biology*. San Francisco, Ca: Pearson Education.
5. DeRisi JL, Iyer VR, Brown PO. (1997). Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science* 278, 680–6.
6. Dudoit S, Fridlyand J. (2002). A prediction-based resampling method for estimating the number of clusters in a dataset. *Genome Biology* 3(7), 31-6.
7. Eisen MB, Spellman PT, Brown PO, Botstein D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Science* 95, 14863-8.
8. Fortina P, Delgrosso K, Sakazume T, Santacroce R, Moutereau, S, Su HJ, Graves D, McKenzie S, Surrey S. (2000). Simple two-color array-based approach for mutation detection. *European Journal of Human Genetics* 8(11), 884-95.
9. Gasch AP, Spellman PT, Kao CM, Carmel-Harel O, Eisen MB, Storz G, Botstein D, Brown PO. (2000). Genomic expression programs in the response of yeast cells to environmental changes. *Molecular and Cellular Biology* 2000 11, 4241-57.
10. Gibbons FD, Roth FP. (2002). Judging the quality of gene expression-based clustering methods using gene annotation. *Genome Research* 12, 1574–81.
11. Hastie T, Tibshirani R, Friedman J. (2001). *The elements of statistical learning. Data mining, inference, and prediction*. New York: Springer.
12. Horimoto K, Toh H. (2001). Statistical estimation of cluster boundaries in gene expression profile data. *Bioinformatics* 17, 1143-51.
13. Levenstien M, Yang Y, Ott J. (2003). Statistical significance for hierarchical clustering in genetic association and microarray expression studies. *BMC Bioinformatics* 4, 62.
14. Leming S. (2004). *DNA Microarray* (on-line). http://www.gene-chips.com/.
15. Michaels GS, Carr DB, Askenazi M, Furman S, Wen X, Somogyi R. (1998). Cluster analysis and data visualization of large-scale gene expression data. Pacific *Symposium in Biocomputing* 42–53.
16. Ramoni MF, Sebastiani P, Kohane IS. (2002). Cluster analysis of gene expression dynamics. *Proceedings of the National Academy of Science* 99(14), 9121–6.
17. Reimer RR, Fraumeni JF Jr., Ozols RF, Bender R. (1977). Re: Pancreatic cancer in father and son. *Lancet* 1, 911-912.
18. SAS Institute Inc. (2003). *SAS Online Doc, Version 8.2*. Cary, North Carolina: SAS Institute Inc.
19. Wen X, Fuhrman S, Michaels G,Carr D, Smith S,Barker J, Somogyi R. (1998). Large-scale temporal gene expression mapping of central nervous system development. *Proceedings of the National Academy of Science* 95, 334–9.

ACKNOWLEDGMENTS

I would like to acknowledge the never ending support and help from my advisor at the University of Louisville, Dr. Cerrito.

CONTACT INFORMATION

(In case a reader wants to get in touch with you, please put your contact information at the end of the paper.)
Your comments and questions are valued and encouraged. Contact the author at:

John Schwarz
University of North Carolina, Chapel Hill
502-235-5230
jschwarz@email.unc.edu