Applying Wave Processing Techniques to Clustering of Gene Expressions

Paul D. O'Neill,[1] George D. Magoulas[2] and Xiaohui Liu[1]

[1]*School of Information Systems, Computing and Maths, Brunel University, Uxbridge, Middlesex, UB8 3PH. U.K.*

[2]*School of Computer Science, and Information Systems, Birkbeck College, University of London, Malet Street, London WC1E 7HX, U.K.*

ABSTRACT

This paper examines the current process of clustering gene-expression time-series data and proposes a novel application of filtering techniques with the intention of reducing the noise that is commonly found in this type of data. Currently, most noise reduction that is performed on gene-expression data is restricted to just individual points of expression, such as the removal of background noise. This paper proposes that multiple samples of each gene can be treated as a waveform and therefore, such standard wave-smoothing techniques as a moving average or Fourier transform filtering can improve the quality of the data. This hypothesis has been tested on a synthetic, human herpesvirus 8 and yeast-cell-cycle gene-expression experiments. The paper illustrates that the use of these techniques generally improves the results of clustering the dataset, illustrated by contrasting the quality of the clusters generated by *k*-means, partitioning around medoids, and hierarchical-clustering algorithms. These improvements are demonstrated using various techniques, including *homogeneity*, *separation*, and a *weighted-kappa* based metric. The clustering results are also verified biologically by contrasting the effect of filtering on common proximity metrics used by clustering algorithms and then verified against domain knowledge.

Reprint requests to: Paul D. O'Neill, School of Information Systems, Computing and Maths, Brunel University, Uxbridge, Middlesex, UB8 3PH. U.K.; e-mail: paul.oneill@brunel.ac.uk

KEYWORDS

gene expression, clustering, digital filtering, pre-processing, time series

1. INTRODUCTION

Microarray technology (Moore, 2001) is in its infancy and is still very expensive and time consuming, leading to experimental results that are often limited in the number of sample points and lack the rigorous error checking and re-testing that is more common in other areas of experimental research. This drawback means that before these results are analyzed, every step should be taken to 'clean' the data, using all the available tools and information. Recently, more interest has been shown in the processing of the Microarray images, from image processing methods (Kooperberg et al., 2002) to the development of new clustering algorithms, such as Cast (Ben-Dor at al., 1999), and the use of support vector machines (Brown et al., 2000). This paper looks at using wave-processing techniques on gene-expression time-series data to reduce the impact of erroneous sample points and in doing so, give us more confidence in the results obtained from the analysis of such experiments.

This paper proposes treating the expression profiles as waveforms rather than the current approach of correcting for errors individually on each expression point in the gene's profile. This approach allows for the application of other error reduction techniques, such as those more commonly used in filtering and reconstruction of digitally stored analogue signals. The paper takes a preliminary look at two such methods, a moving average and a Fourier transform filter (FTF) in order to explore the effect that these have on a selection of clustering algorithms used to process the resultant data.

The paper is organized as follows; Sec. 2 looks at the clustering of gene-expression data, including an overview of microarray technology, the metrics and different algorithms used for clustering, and a brief discussion of the methods that can be used to compare the quality of the generated clusters. Section 3 gives a detailed explanation of the proposed filtering process, from an introduction of simple filtering algorithms to an example demonstrating

how they can be incorporated as part of the existing clustering process. In Sec. 4, the paper presents the results from three sets of experiments performed on datasets of different dimensionality, ranging from 106 to over 2000 variables. These findings show the improvements this method has had on the quality of the clusters and demonstrate that our approach can improve the biological significance of the resultant groupings. Finally, Sec. 5 summarizes these findings and discusses future work.

2. CLUSTERING EXPRESSION DATA

Microarray technology allows biologists to design experiments in which the investigator can contrast the expression levels of genes from two different cell cultures, such as comparing the genes expressed in an infected cell against those in a normal cell. An example of this approach can be seen in the HHV8 dataset (Jenner et al., 2001) used in the present paper, in which the expression levels of 106 genes were monitored and recorded at 8 time points.

This paper will focus on experiments of a sequential or time series nature, w here at each time point a microarray with all relevant genes is hybridized and then scanned. The expression levels of each gene are then recorded; simple error correction such as subtracting the local background noise has been performed. At this point, however, a large amount of noise can remain in the measurement of each point or gene on the array. There can be many reasons for this noise, including bad alignment or recognition of the expression point, background noise on the chip, or the over expression of a gene contaminating surrounding genes. The noise is reduced to an extent by the current pre-processing techniques (Smyth et al., 2002). Nevertheless, if we treat the data as a whole, we can then reduce any errors further.

One way to do this is to treat each gene expression profile as a waveform. The reason for treating the data as waveforms is inherent in the nature of the genes. They cannot jump from one level of expression to another in any one instant and therefore have to exhibit a gradual change in expression; this can be over a matter of minutes or last several hours. The main problem that will be faced with this type of data is the question—are there enough sample points to reconstruct each waveform successfully?

2.1 Similarity Metrics

In the case of gene-expression profiles, two main comparison metrics when clustering are Euclidian distance and Pearson's uncentered correlation coefficient, as used in the Cluster package provided by 'Eisen Lab' (http://rana.lbl.gov). Given the expression levels observed at k time points of two genes p: $(p_1, p_2, \ldots, p_k)$ and q: $(q_1, q_2, \ldots, q_k)$, the formula for calculating the Euclidean distance, d, between the two gene vectors is shown in Eq. (1).

$$d = \sqrt{\sum_{i=1}^{k}(p_i - q_i)^2} \tag{1}$$

Pearson's correlation coefficient, r, a well-established method for the comparison of objects, has been used extensively in the clustering of gene-expression data. This method measures the linear relationships between two variables, x_1 and x_2, which can be either discrete or continuous. We use the uncentered version, defined in Eq. (2), where x_{1i} and x_{2i} are the ith component of the two variables, respectively, given a total of k observations. This version is used, as it is more suitable for gene expression data.

$$r = \frac{\sum_{i=1}^{k} x_{1i}x_{2i}}{\sqrt{\sum_{i=1}^{k} x_{1i}^{\,2} \cdot \sum_{i=1}^{k} x_{2i}^{\,2}}} \tag{2}$$

The limits of this coefficient are [–1, 1], where a value greater than zero indicates a positive linear relationship and a value less than zero indicates a negative linear relationship.

2.2 Clustering Methods

Currently one of the main uses of gene expression data is the application of clustering algorithms in an attempt to classify unknown genes (Moore, 2001; D'Haeseleer et al., 2000). For example, if gene A is clustered with gene B and one knows that gene A has function F, one can hypothesize that gene B may also have function F. As this paper is studying the effect of pre-

processing the data before clustering, we will be using three standard clustering algorithms—K-means, Partitioning Around Medoids (PAM), and Hierarchical. All are used regularly in this field and for this paper, the implementation from the statistical package 'R' (http://www.r-project.org) is used. Implementations of K-means and Hierarchical clustering can also be found in Eisen's Lab, 'Cluster' package.

K-means clustering (McQueen, 1967) partitions data by maintaining k cluster centers that define the boundaries of each partition. These centers are initially random points in the hypervolume containing the dataset. Each data point is assigned the nearest cluster center, and then the centers are recomputed using their current members. The entire procedure is repeated until a certain convergence criterion is met, such as no reassignment of data points or a minimal decrease in squared error. This method, however, is heavily influenced to initial conditions and often becomes stuck in local minima.

Partitioning Around Medoids is described in Kaufman and Rousseeuw (1990). This approach is based around a search for k medoids that are representative of the data. Once these are found, the clusters are created by assigning each profile to the nearest medoid. Partitioning Around Medoids is often compared to K-means and generally is not so vulnerable to initial starting conditions; PAM differs from K-means in that it can use a dissimilarity matrix as its initial input and in that it minimizes a sum of dissimilarities instead of a sum of squared Euclidean distances. In most cases, PAM is accepted to produce better clustering results than K-means.

Hierarchical clustering produces a hierarchical (binary) tree or dendrogram representing a nested set of data partitions. Sectioning a tree at a particular level leads to a partition with a number of disjoint groups, therefore yielding different clusters within the data. Hierarchical clustering has extensively been applied to many gene-expression datasets, such as Moore (2001, Eisen et al. (1998) and Gasch et al. (2000). In this paper, the 'cutree' algorithm in 'R' was then used to divide the dendrogram into k clusters.

2.3 Analyzing the Quality of Clusters

One main problem with gene expression data is the verification of the resulting clusters, as often very little domain knowledge is available about the

dataset being used. This paper attempts to alleviate this problem in three main ways. First of all real data for which we have a limited amount of domain knowledge will be used. Second, a synthetic gene expression dataset that has been heavily distorted with noise will be tested and verified against its original grouping. Finally, the quality of the clusters that are generated will be evaluated based on two metrics: *separation* and *homogeneity*. These metrics are useful as they give us an indication of the clusters quality without the need for knowledge of a true solution. Next, we briefly present how these metrics have been tailored for our experiments; the reader can find detailed explanations of these metrics and of how they are applied to clustering in Sharan and Shamir (2000).

Homogeneity is a measure of how close each gene within a cluster matched the clusters *fingerprint*. In this paper, we took the clusters fingerprint to be the expression profile with the highest average correlation to all the other members within the cluster. To calculate homogeneity, we took the average Pearson's correlation between the fingerprint and all members of the cluster. This calculation results in a correlation value between –1 and 1, with 1 being perfect homogeneity.

Separation (sometimes referred to as independence) is a measure of the dissimilarity between clusters. Separation is calculated from the average correlation between each clusters fingerprint and that of all the other clusters. Once again, Pearson's correlation coefficient was used as the distance metric, with a value of –1 indicating perfect separation. To allow for an easier comparison with homogeneity, however, we multiplied this value by –1 so that a value of 1 indicates maximum separation, meaning that when contrasting the clusters generated from the various experiments, we will be looking to maximize both homogeneity and separation to show an improvement in the quality of the clusters produced.

3. FILTERING EXPRESSION DATA

In this paper, we will be using two waveform-processing techniques, a moving average and a FTF. Both are well-established techniques that are well documented in Percival and Walden (2000) and therefore this paper will

focus instead on the aspects important to their implementation with gene expression data. Generally, these aspects are used in waveform reconstruction, such as the reconstruction of under-sampled digital sounds signals to their original waves. A Weighted Moving Average (WMA) filter is used smooth signals and is defined in Eq. (3), where D is the data series, n is the item of data, and f defines the weight of the filter. The greater the value of f the less of an effect each new point will have on the waveform.

$$D_n = \frac{D_{n-1}}{f} + \frac{D_n \cdot (f-1)}{f}, 1 < n \leq |D_{Size}|, \quad f \in Z^+ \tag{3}$$

Fourier Series Approximation uses a mathematical technique known as Fourier analysis. This technique can be used to show that any time-varying signal is made up of a possibly infinite number of single frequency sinusoidal signals, as described in Eq. (4), where n is the number of terms in the series, a_0, a_1,..., a_n are amplitude coefficients, θ is the phase angle in radians, T is the period of the signal, and ω is the fundamental frequency of the wave.

$$\begin{aligned} &x(t) = a_0 + a_1 \cos(\omega t + \theta_1) + \ldots a_n \cos(n\omega t + \theta_n), \; 1 \leq n \leq |a|, \\ &0 \leq \theta < 2\pi, \omega = 2\pi/T \end{aligned} \tag{4}$$

By taking the low frequency components of these sinusoidal frequencies, a signal can be reconstructed mathematically. Using these principles, the FTF is constructed to give a good approximation of these components, and the standard version of the filter that is often used in wave reconstruction in digital to analogue converters. Figure 1 shows the effect of two filters on a noisy gene profile. Shown here is a simulated upregulated gene with heavy noise. This figure contrasts the difference between the way in which these two methods work—the moving average filter just smoothes the wave to an extent as opposed to the Fourier transform filter that attempts to approximate the original waveform.

The Filtering Process

Figure 2 shows both approaches to clustering. The classic method of treating gene expression data is shown with a dashed arrow and the proposed

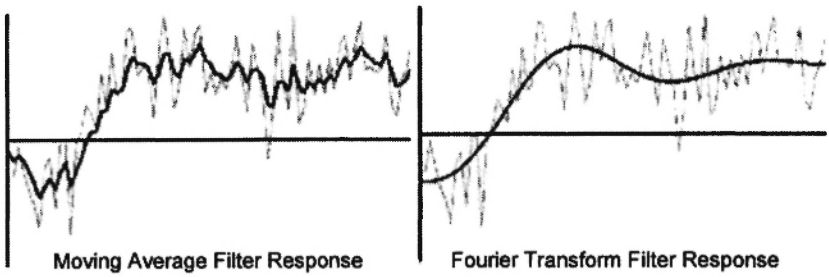


Fig. 1: Example of both Moving average and Fourier filters

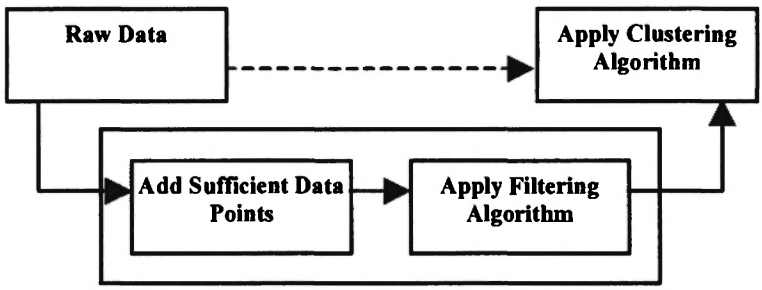


Fig. 2: Processing stages of gene expression data

filter based pre-processing stages are highlighted in the shaded area. The first stage of proposed pre-processing deals with a problem commonly found in gene expression data in which there are only a very limited number of time points. Although we are clustering hundreds of genes, it is quite common to have only as few as eight measurements in the series. This is normally usable by most clustering algorithms. Nevertheless, applying filter methods to such restricted observation space can lead to the data becoming distorted or corrupted. Nevertheless, in this situation filters are also commonly used, an example being the reconstruction of under-sampled digital sound signals, where a digital sound signal is converted back to its original waveform.

To accomplish this effect with the gene expression data we have simply added an extra three expression points between every two measurements. These points were calculated using the formula shown in Equation (5).

$$X_n = x_1 + \frac{n}{N+1}(x_2 - x_1), \quad n \in [1, 2, \ldots N] \tag{5}$$

Here the new points X_n to be inserted between x_1 and x_2 are calculated with N being the number of points to insert. Noteworthy is that by increasing the number of data points in this way leads to no alteration in the genes profile, and as it is applied globally across the dataset there is only a scaled change to the results of the distance metrics.

The next stage is the application of the filtering algorithms to the gene expression profiles. Any waveform-filtering algorithm can be used for this stage. This paper focuses on using a moving average and a FTT with particular interest in their effect on popular clustering algorithms. The main benefit of this simple process is that it can easily be applied as part of the normal clustering process, allowing the existing clustering tools to be used without any modification.

3.2 An Example of the Proposed Approach

This section will demonstrate visually how this process can be applied to eight sample genes to show the effect of the process on the metric that is used when clustering. In this case, Pearson's correlation coefficient has been selected because (a) it is commonly used when processing gene expression data and (b) it can be easily interpreted. We should note that this example is presented only as a simple demonstration and as such, only a few genes known to cluster with each other have been used.

In Fig. 3, one can see the expression profiles of the eight sample genes, which can be easily clustered into four pairs—one pair that oscillates, one that is downregulated, one that is upregulated, and one pair that just shows a constant level of expression. If this small dataset is clustered, then the majority of the clustering algorithms will correctly cluster the genes.

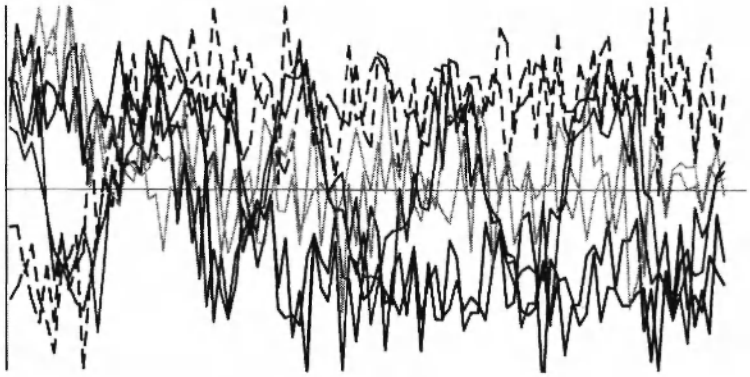


Fig. 3: Eight original gene profiles

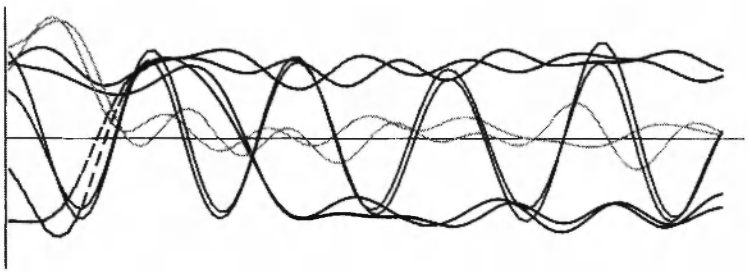


Fig. 4: Original gene profiles after applying a Fourier transform

It would take at least a few minutes for most people to pick out the correct clusters from Fig. 3. If we apply a simple FTT to the data, however, one can see from Fig. 4 that this task is almost effortless, with all the clusters easily definable. This example shows how the FTT has approximated each of the expression profiles, removing a lot of the noise that previously made clustering these genes difficult.

Another filter that we are testing in this paper is the moving average. Instead of approximating the expression profile, this filter works by smoothing the wave to a degree. The results of applying this filter can be seen in Fig. 5.

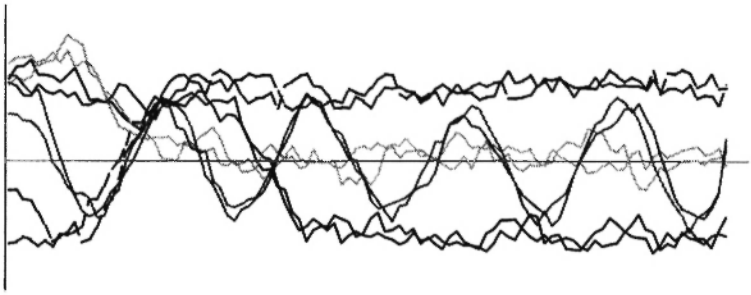


Fig. 5: Original gene profiles after applying a moving average filter

TABLE 1

Summary of the correlation between the original and filtered data

Change in Homogeneity		
Gene pairs	FTF	MA
1 & 2	+11.3%	+10.9%
3 & 4	+15.7%	+14.7%
5 & 6	+35.7%	+38.0%
7 & 8	+18.7%	+14.6%
Avg.	+20.3%	+19.5%

Change in Separation		
Gene pairs	FTF	MA
1 & 2	+5.1%	+7.4%
3 & 4	+0.5%	–3.5%
5 & 6	–0.2%	–7.6%
7 & 8	+0.3%	V5.3%
Avg.	+1.4%	–2.3%

Contrasting this to the original data one can see that although not quite as clear as the Fourier filter, the moving average has still made the data far easier to cluster. This effect can also be seen in the metrics used to cluster the different gene profiles, such as Pearson's uncentered correlation coefficient. Table 1 shows a summary of the differences shown between correlation matrices for the original data compared against the Fourier and moving average filtered data.

On the left side is a table showing the change in correlation between the four pairs that we know should cluster. On average there is roughly a 20% increase shown in correlation after applying both the Fourier and the moving average filter. On the right side is a table showing the change in separation

between each pair of genes and the other 6 genes that we know should not be clustered with them. For example, we know that genes 1 and 2 should not cluster with genes 3 to 8, and they showed a 5% increase in separation. Overall, there was approximately a 2% change, although we should note that in the case of the moving average filter, this caused a slight reduction in the separation of the clusters. The results give an indication of the effect of filtering on real datasets and this will be presented in the next section. On the whole, filtering seems to improve cluster quality, especially homogeneity. In certain cases, however, the cluster quality is reduced for whatever reason. For example, genes 5 and 6 showed only an average 8% reduction in separation when the moving average filter was used but they did show an average 37% improvement in their homogeneity. The next section highlights these results and looks at possible explanations for their occurrences.

4. EXPERIMENTAL RESULTS

This section presents the results from three main experiments. Each experiment and its significance are explained in detail along with a summary of any improvements brought about by the use of filtering and highlights results that show filtering has had a negative effect. All the results here can be assumed the average of 10 separate tests unless the algorithm is deterministic in nature. Additionally, wherever the three clustering algorithms are being compared, we can assume that Euclidean distance was used as the proximity metric.

Several factors, such as the number of clusters and various heuristics for the filters affect the quality of the results. The experiments reported here provide a preliminary evaluation of the effect of filtering. Therefore, these heuristics such as the weighted coefficient of the WMA filter, order of the Fourier filter, and cluster size may not be optimal for the given dataset but remain constant throughout the experiments unless stated otherwise.

4.1 Description of the Datasets

During this paper, three datasets have been used to validate the effect of filtering. These datasets have been picked to offer some diversity in the data

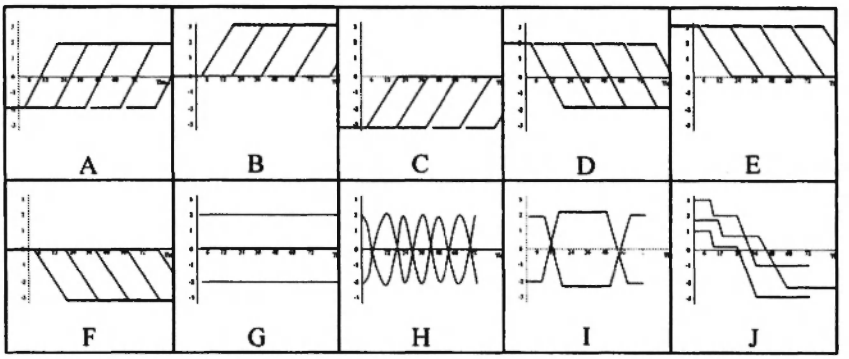


Fig. 6: Synthetic gene expression profiles

being used. The first dataset is synthetic and is constructed to resemble real gene-expression data, based on previous biological knowledge of expression profiles. The dataset consists of 2217 genes expressed over 100 time points. Each variable belongs to one of a family of functions that determines the shape of the expression profile and these were designed to mimic the typical gene expression patterns shown in Fig. 6. There are 40 potential clusters in total, each of which varies in size from 30 to 70 members. Patterns of gene expression were produced with time-dependent variations in a given gene expression profile ($\log_2$ ratio). The data reflect observations from two color microarray experiments. For example, patterns of continuous time lagged increase in gene-expression ratios depicted in (A) and cyclical time-dependent patterns of gene expression, depicted in (H). A zero-centered normal noise process was added with standard deviation of 0.6. In addition, operators have been applied to distort certain variables to produce effects like skewing some gradients.

A second experimentally derived dataset of 106 human herpesvirus 8 (HHV8) genes expressed over 8 time points of the viruses lytic replication cycle (Jenner et al., 2001; Kellam et al., 2001) was also used to evaluate the clustering results. This dataset includes various control genes and genes of known functions so that any results can be verified against domain knowledge.

The Yeast Data is taken from experiments conducted by Spellman et al. (1998) on the yeast's cell cycle. For the purpose of these tests, only the yeast

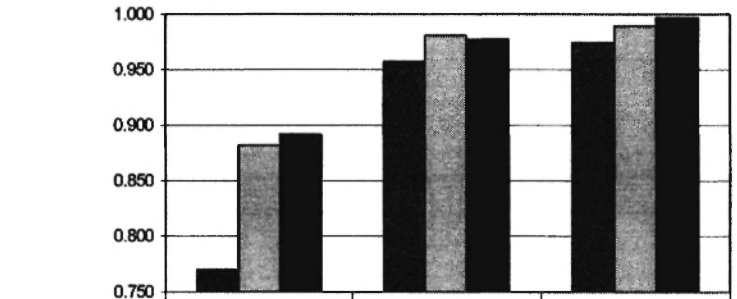
cultures that were synchronized by the arrest of a CDC15 temperature-sensitive mutant were used. The dataset consists of samples taken every 7 minutes for 119 minutes in total. From this, only the gene profiles that contained no missing values across the 17 time points were used, leaving 623 gene profiles in total.

4.2 Weighted-Kappa Based Comparison

In this experiment, a weighted-kappa based metric is used to evaluate each of the final clusters produced from the synthetic gene expression profiles, as described in Fig. 6. One of the biggest advantages for using synthetic expression data is that all of the expected clusters are already defined and therefore we can compare how well each method clusters the data. This metric was chosen as it is often used in assessing agreement in medical statistics to rate the agreement between two or more observers by creating a contingency table between the classifications of each. In the case of the clustering algorithm, the first observer can be seen as the correct clustering solution, and the second observer is the clusters produced by the algorithm in question. A good explanation of how to calculate this metric can be found in Altman (1997), and as a general guide, values of 0.6 and above are considered *good*, with 0.8 and above being *very good*.

Figure 7 shows the average weighted-kappa values for each of the three algorithms when run to find the 40 original clusters on both the original (raw) data and the two filtered versions of the data. This dataset is designed to be particularly noisy to test different clustering algorithms to their limits and therefore should give a good indication of the effect of filtering.

In the *k*-means test, both filtered versions showed a significant increase in the quality of the clusters with the moving average filter performing best. In this particular test, the use of filtering brought the *k*-means algorithm up to a weighted-kappa value of 0.89, which is an increase of 12%. The results for both the PAM and hierarchical algorithms are more interesting as these already have very high weighted-kappa values. In both cases, the moving average filtered data once again shows an increase in the quality of the clusters although this time to a much lesser extent, with a 2% improvement shown in the hierarchical tests. The results for PAM do not follow this



	K-Means	PAM	Hierarchical
Raw	0.770	0.958	0.975
Fourier	0.882	0.981	0.990
Moving Average	0.892	0.978	0.997

Fig. 7: Comparison of weighted-kappa results on the synthetic dataset

trend in that the Fourier filter performed better than the MA filter. This difference could be due to bias' within the algorithm but obviously highlights the need for further research into the exact effect filtering has on different algorithms.

The Fourier filter does not perform as well as the moving average. One possible explanation for this could be in the way the algorithm works, instead of modifying the actual data to remove noise, the algorithm attempts to approximate the profile data using a series of sinusoidal waves, which may have resulted in the loss of valuable characteristics of the expression profile.

4.3 Biological Effects of Filtering

When applied to the synthetic data, the algorithms seemed to show that they could improve the results of clustering; what effect would this have on real data and would any improvement be biologically significant? In an attempt to answer this question, this experiment looks at the effects filtering has had on the HHV8 dataset. The experiment compares the results of both filtered and unfiltered data against 6 biological features consisting of 16 known correlations between 29 genes. These include five sets of genes that we know have similar functions and nine genes that were put into each experiment twice, as a control, and should,

TABLE 2

Change in Euclidean distance for the 16 correlations

	Fourier Filter		Moving Average	
	Order = 5	Order = 10	Weight = 4	Weight = 8
Overall decrease in distance	0.911	0.108	4.867	8.765
Gene pairs showing decreased distance	16/16	16/16	15/16	14/16
Gene pairs showing increased distance	0/16	0/16	1/16	2/16

TABLE 3

Change in Pearson's coefficient for 16 correlations

	Fourier Filter		Moving average	
	Order = 5	Order = 10	Weight = 4	Weight = 8
Overall increase in correlation	0.019	0.001	0.016	0.086
Gene pairs showing improved correlation	6/16	1/16	8/16	7/16
Gene pairs showing decreased correlation	0/16	1/16	1/16	2/16

therefore cluster in pairs. The presence of only six features by which we can judge the quality of any changes to the clustering metrics and the limited numbers of time points (only eight observations) introduce additional difficulties; such problems are common to gene-expression data clustering.

Table 2 shows the results of this test when using the Euclidean distance metric and Table 3 shows the results for Pearson's uncentered correlation coefficient. One should remember that with Euclidean distance we are looking for a decrease in distance between gene profiles, whereas with the Pearson correlation coefficient we are looking for an increase in correlation.

The results for the Euclidean distance seem to show that all the expected correlations between genes show an improvement, with the moving average set at a weight of 8 showing the greatest improvement. Note, however, that with this higher value, two gene pairs showed reduced correlation. The Fourier filter set to an order of 10 showed the lowest overall improvement,

emphasizing that further study is needed into the best way of setting the heuristic parameters of each filter.

Overall, Pearson's correlation coefficient showed a lot less improvement with the Fourier filter, showing at best a 0.02 increase. When using the moving average, approximately half the gene pairs showed improvement in their correlation, with at most two gene pairs showing a decrease. The Fourier filter set at an order of 5 showed a similar trend, albeit when set at an order of 10, the correlation between virtually all the gene pairs appeared to have remained the same. This result could be attributed to the order of the filter being set too high, resulting in too much of the genes expression profile being distorted. This conjecture, however, can be supported only by more in-depth experiments. We can therefore say that overall, filtering shows a positive improvement in the correlation between these biological features with only 2 or fewer of the 16 correlations showing any sort of significant decrease.

4.4 Cluster Quality Comparison

In this final experiment, all three datasets were used to produce an overview of the filtering process on each. Two scoring metrics were used—homogeneity and separation, as explained in Sec. 2.3. Additionally, note that for these tests the synthetic dataset was clustered into 40 groups, whereas the CDC15 and HHV8 datasets were grouped into 10 clusters. Forty is the optimal number of clusters for the synthetic dataset but ten clusters may not be the optimal number for the other two. This just means that it may be possible to improve the clustering results slightly for the CDC15 and HHV8 datasets by fine-tuning the clustering algorithms. For all tests, the Fourier filter was used with an order of five and the moving average filter with a weight of four. These values were chosen as they showed significant change to the waveforms without distorting them too much.

Figure 8 illustrates the overall change in both the separation and the homogeneity for each algorithm. As both of these metrics have the same scale, we were able to sum the improvement shown to give an overview of the effect filtering had. Any positive values in the graph represent tests for which an improvement was shown. Overall, it seems that either filtering method had a positive effect on the results of k-means although as with the other

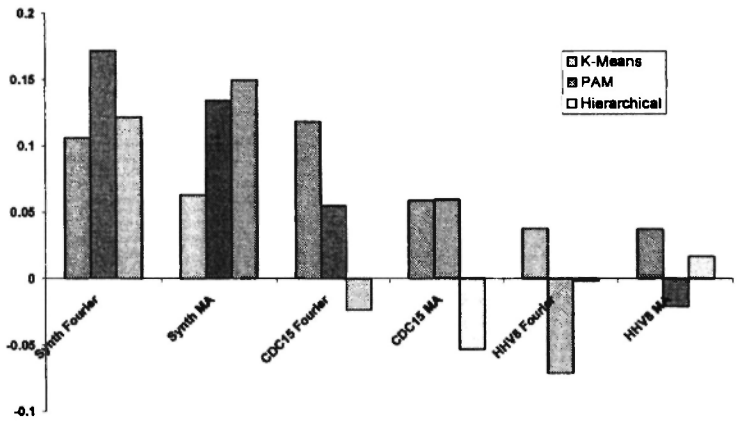


Fig. 8: Improvement in cluster quality over all methods and datasets

clustering algorithms this was most beneficial to the synthetic dataset. The PAM and hierarchical did show quite as much improvement, with hierarchical doing poorly on all the real microarray datasets and PAM doing poorly on the HHV8 data. One interesting trend, however, is the falloff in performance as the number of data points decreases. The synthetic data had the most with 100 time points, whereas the Yeast (CDC15) data had only 17 time points, and the HHV8 even less with only 8. It seems reasonable to expect this sort of drop in performance with so few time points as generally filtering algorithms are designed to work on much longer series.

Generally speaking, filtering shows an improvement in the clusters produced because, as expected, filtering removes a lot of the noise and other artefacts that normally reduce cluster quality. This means that filtering is improving on the performance over just using a standard clustering routine on the original data. Rather than concentrating on the improvements, we instead decided to take a more detailed look at the results that showed reduced correlation after filtering. For example, when hierarchical clustering was used in the CDC15 dataset, both the Fourier and the moving average filtered data exhibited a drop in homogeneity that resulted in a drop in quality (see Fig. 8). In an attempt to investigate further, we re-ran the CDC15 homogeneity test

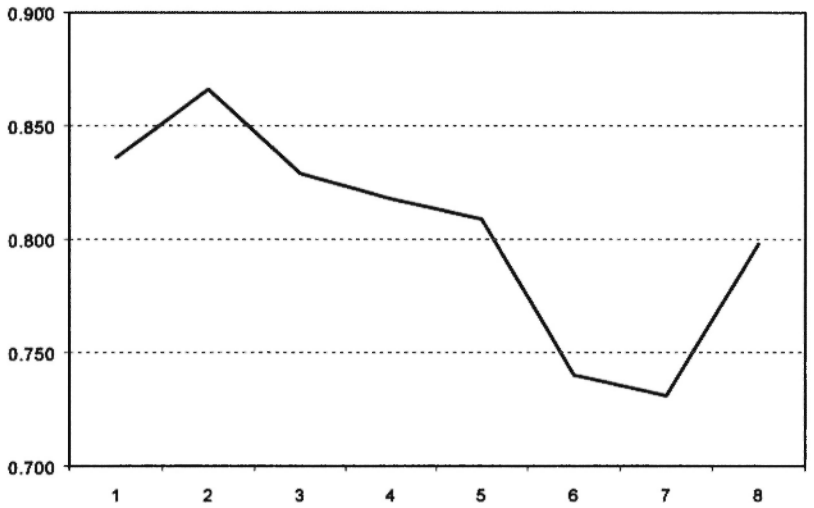


Fig 9: Change in homogeneity as filter weight is increased.

using hierarchical clustering with the moving average filter weight set between 1 and 8. The results for this are shown in Fig. 9.

Figure 9 appears to suggest that the weight of the filter plays a very important role and that by changing the filter to a setting of two, we were able to improve on the previously negative results. Additionally, it does not follow a simple trend because the homogeneity seems to show an increase again when the filter was set to eight. This tends to indicate that other factors may be influencing the results. A more detailed study into the effect of these and other factors will have to be conducted.

5. CONCLUSION AND FUTURE WORK

As microarray technology increases in power by the day, we are faced with very high dimensional data that often only has a limited number of time points. This paper has highlighted that it may be possible to get more information from gene expression data than we do at present. Rather than

looking at each time point on a microarray in isolation, we suggest that the genes expression profile can be considered as a whole. In a world in which we are still in our infancy in the exploration of gene-expression data, this paper proposes that we should make use of ideas from the field of signal processing in an attempt to make the best use of what is—often due to limits in technology and finance—less than ideal experimental results.

This paper has shown from three different experiments that filtering can increase our confidence in existing clusters. Additionally, it seems that using the filtered data not only helps to improve the quality of the clusters but also improves them in a way that is biologically significant. We do not claim that these filtering methods are perfect or that they are the most suitable choice for this type of data; what this paper shows is that there is potential for filtering technologies to be used in this way, and that further research into this area is definitely needed. For example, gaining a deeper insight into the effect of applying these as well as other filtering techniques to gene-expression data, along with an analysis of the effects shown by varying the values of various heuristics (especially filter order and weights). One possible way to extend the research in this area would be to look at more sophisticated filters such as adaptive neural filtering (Principe et al., 2000) or the use of Wavelet filtering methods (Percival & Walden 2000). These techniques have already been applied to other areas such as medical informatics, for example the pre-processing of anomalous Computer Tomography (CT) images (Frangakis et al., 2001) and show significant potential. One thing is clear though, while gene expression datasets remain limited to small numbers of sparse time points, the form of signal analysis will be difficult at best.

REFERENCES

Altman, D.G. 1997. *Practical statistics for medical research*, London, UK, Chapman and Hall.

Ben-Dor, A., Shamir, R. and Yakhini, Z. 1999. Clustering gene expression patterns, *Journal of Computational Biology*, **6**, 281–297.

Brown, M., Grundy, W., Lin, D., Cristianini, N., Sugnet, C., Furey, T., Ares, M. and Haussler, D. 2000. Knowledge-based analysis of microarray gene expression data using support vector machines, *Proceedings of the*

National Academy of Science USA, **97**, 262–267.

D'haeseleer, Liang, S. and Somogyi, R. 2000. Genetic network inference: from co-expression clustering to reverse engineering, *Bioinformatics*, **16**, 707–726.

Eisen, M.B., Spellman, P.T., Brown, P.O. and Botstein, D. 1998. Cluster analysis and display of genome-wide expression patterns, *Proceedings of the National Academy of Science USA*, **95**, 14863–14868.

Frangakis, A.S., Stoschek, A. and Hegerl, R. 2001. Wavelet transform filtering and nonlinear anisotropic diffusion assessed for signal reconstruction performance on multidimensional biomedical data, *IEEE Transactions on Biomedical Engineering*, **48**, 213–222.

Gasch, A.P., Spellman, P.T., Kao, C.M., Botstein, D. and Brown, P.O. 2000. Genomic expression programs in the response of yeast cells to environmental changes, *Molecular Biology of the Cell*, **11**, 4241–4257.

Jenner, R.G., Alba, M.M., Boshoff, C., and Kellam, P. 2001. Kaposi's sarcoma-associated herpesvirus latent and lytic gene expression as revealed by DNA arrays, *Journal of Virology*, **75**, 891–902.

Kaufman, L. and Rousseeuw, P.J. 1990. *Finding groups in data: an introduction to cluster analysis* , New York, NY, USA, Wiley.

Kellam, P., Liu, X., Martin, N., Orengo, C., Swift, S. and Tucker, A. 2001. Comparing, contrasting and combining clusters in viral gene expression data, London, UK, *Proceedings of the International Workshop on Intelligent Data Analysis in Medicine and Pharmacology*, 56–62.

Kooperberg, C., Fazzio, T.G., Delrow, J.J. and Tsukiyama, T. 2002. Improved background correction for spotted DNA microarrays, *Journal of Computational Biology*, 9, 55–66.

McQueen, J. 1967. Some methods for classification and analysis of multivariate observations, *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, 281–297.

Moore, S.K. 2001. Making chips to probe genes. *Biotechnology, IEEE Spectrum*, March, 54–60.

Percival, D.B. and Walden, A.T. 2000. *Wavelet methods for time series analysis*, Cambridge, Massachusetts, USA, Cambridge University Press, 20–28.

Principe, J.C., Euliano, N.R. and Lefebvre, C.W. 2000. *Neural and adaptive systems: fundamentals through simulations*, John Wiley and Sons, 429–524.

Sharan, R. and Shamir, R. 2000. CLICK: A clustering algorithm with applications to gene expression analysis, *Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology*, San Diego, California, USA, 307–316.

Smyth, G.K., Yang, Y.H. and Speed, T. 2002. Statistical issues in cDNA microarray analysis, in: *Functional genomics: methods and protocols*, Methods in molecular biology, Volume 224, edited by Brownstein, M.J.

and Khodursky, A.B., Totowa, New Jersey, USA, Humana Press, 111–136.

Spellman, P.T., Sherlock, G., Zhang, M.Q., Iyer, V.R., Anders, K., Eisen, M.B., Brown, P.O., Botstein, D. and Futcher, B. 1998. Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization, *Molecular Biology of the Cell*, **9**, 3273–3297.