BIOINFORMATICS

Vol. 21 Suppl. 1 2005, pages i159–i168
doi:10.1093/bioinformatics/bti1022

Clustering short time series gene expression data

Jason Ernst[1,*], Gerard J. Nau[2] and Ziv Bar-Joseph[1]

[1]Center for Automated Learning and Discovery, School of Computer Science, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA and [2]Department of Molecular Genetics and Biochemistry, University of Pittsburgh School of Medicine, 200 Lothrop Street, Pittsburgh, PA 15261, USA

Received on January 15, 2005; accepted on March 27, 2005

ABSTRACT

Motivation: Time series expression experiments are used to study a wide range of biological systems. More than 80% of all time series expression datasets are short (8 time points or fewer). These datasets present unique challenges. On account of the large number of genes profiled (often tens of thousands) and the small number of time points many patterns are expected to arise at random. Most clustering algorithms are unable to distinguish between real and random patterns.

Results: We present an algorithm specifically designed for clustering short time series expression data. Our algorithm works by assigning genes to a predefined set of model profiles that capture the potential distinct patterns that can be expected from the experiment. We discuss how to obtain such a set of profiles and how to determine the significance of each of these profiles. Significant profiles are retained for further analysis and can be combined to form clusters. We tested our method on both simulated and real biological data. Using immune response data we show that our algorithm can correctly detect the temporal profile of relevant functional categories. Using Gene Ontology analysis we show that our algorithm outperforms both general clustering algorithms and algorithms designed specifically for clustering time series gene expression data.

Availability: Information on obtaining a Java implementation with a graphical user interface (GUI) is available from http://www.cs.cmu.edu/~jernst/st/

Contact: jernst@cs.cmu.edu

Supplementary information : Available at http://www.cs.cmu.edu/~jernst/st/

1 INTRODUCTION

Time series gene expression experiments is an increasingly popular method for studying a wide range of biological processes. Examples include response to temperature changes and other stress conditions (Gasch *et al*., 2000), immune response (Guillemin *et al*., 2002), developmental studies (Arbeitman *et al*., 2002), and various systems in the cell (Storch *et al*., 2002).

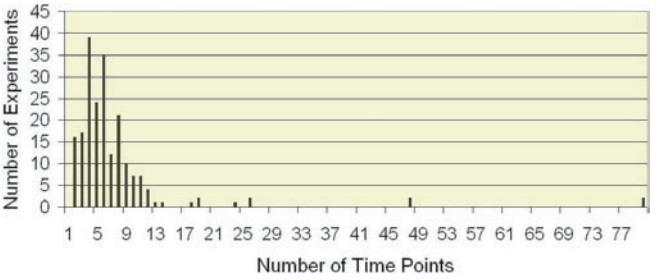

Fig. 1. Distribution of lengths of times series in the SMD as of June 2004.

Although there have been time series experiments with as many as 80 time points (Arbeitman *et al*., 2002), almost all time series are much shorter. To investigate the frequency of time series data and the distribution of their length we have examined the Stanford Microarray Database (SMD) (Gollub *et al*., 2003). Although SMD contains only experiments carried out in Stanford (a small part of the total published microarray datasets), we believe that it is representative of the distribution of microarray datasets. As of June 2004, SMD contained microarray data related to ~170 published papers. Approximately 30% of these papers contained original microarray time series data.[1] Many of these papers contain multiple time series datasets. In total, SMD contained data from 20 distinct time series experiments. In Figure 1 we present the distribution of the number of time points in time series data in SMD. As the figure indicates, while some of the time series are long most are very short. In fact, >80% of all time series datasets contain ≤8 points.

There are a number of reasons why short time series datasets are so common. Time series experiments require multiple arrays (and in many cases each point is repeated at least once) making them very expensive. Although microarray

*To whom correspondence should be addressed.

[1]Some of the papers in SMD do not present original data and so the percentage of time series among new expression dataset is probably larger.

technology have greatly improved over the last five years, its cost is still high at around $300–1000 per microarray which is a limiting factor for many researchers. Even if prices go down short time series experiments would remain prevalent since in many studies it is prohibitive to obtain large quantities of biological material. As an example, consider a clinical study in which blood needs to be drawn from patients at various points in time.

Owing to the large number of genes that are being profiled, most papers presenting short time series datasets use one of several clustering methods to analyze their data. Hierarchical clustering (Eisen *et al*., 1998) along with other standard clustering methods (such as *k*-means and self-organizing maps, Tamayo *et al*., 1999) are often used for this task. Although these clustering algorithms yielded many biological insights, they are not designed for time series data. Specifically, all these methods assume that data at each time point is collected independent of each other, ignoring the sequential nature of time series data. More recently, a number of clustering algorithms specifically designed for time series expression data were suggested. These algorithms include clustering based on the dynamics of the expression patterns (Ramoni *et al*., 2002), clustering using the continuous representation of the profile (Bar-Joseph *et al*., 2003) and clustering using a hidden Markov model (HMM) (Schliep *et al*., 2003). While these algorithms work well for relatively long time series dataset (10 points or more) they are not appropriate for shorter time series. As we discuss in the next section (see also Section 3), these algorithms will overfit the data when the number of time points is small. In addition, when analyzing short time series datasets with thousands (or tens of thousands) of genes, many patterns can be expected to appear at random. On account of the noise and the small number of points for each gene, some of these patterns will be shared by many genes. Most clustering algorithms cannot distinguish between patterns that occur because of random chance and clusters that represent a real response to the biological experiment.

In this paper, we present an algorithm designed specifically for short time series datasets. Our algorithm starts by selecting a set of potential expression profiles. These set of profiles cover the entire space of possible expression profiles that can be generated by the genes in the experiment and each represents a unique temporal expression pattern. Since we are dealing with a time series experiment, and because it does not contain many points, a relatively small set of profiles can be defined for such data. Next, each gene is assigned to one of the profiles, and the enrichment of genes in each of the profiles is computed to determine profile significance. Significant profiles can either be analyzed independently or they can be grouped into larger clusters (based on noise estimates from the data). The resulting profiles or clusters of profiles are then analyzed using Gene Ontology (GO) annotations to determine their biological function.

1.1 Related work

As mentioned above, there are many general clustering algorithms that have been applied to gene expression data (Shamir *et al*., 2002). However, these algorithms do not take into account the sequential nature of time series expression data and thus are less appropriate for such data.

This observation has led a number of researchers to investigate methods of analysis specifically designed for time series data. For instance, Ramoni *et al*. (2002) suggest clustering genes based on their dynamics. This method relies on regression and groups together genes whose dynamics can be expressed with roughly the same auto-regressive equation. While this method works well for long time series, it is not appropriate for short ones. Even when using only two regression parameters (the minimum required to distinguish between up and down trend) a 5 time points expression experiment can only use the last 3 time points (the first two cannot be regressed). This may lead to overfitting, and also results in poor cluster separation as we show in Section 3. Bar-Joseph *et al*. (2003) presented a clustering algorithm that uses splines to cluster the continuous representation of time series expression data. Again, this algorithm is not appropriate for short time series. Even when only two spline segments are used, this algorithm requires the estimation of five parameters for each gene (and a few other class related parameters). This will clearly overfit if the dataset contains only a small number of points. Schliep *et al*. (2003) suggest clustering genes based on a mixture of HMM. In an EM style algorithm genes are associated with the HMM most likely to have generated their time courses, then the parameters of the HMMs are estimated based on the genes associated with them. This algorithm requires that the number of time points be much larger than the number of states (or nodes in each Markov chain). Thus, while this algorithm works well for long time series datasets it is not appropriate for short ones.

Predefined profiles have been used in the past to fit expression profiles. For example, Zhao *et al*. (2001) and Lu *et al*. (2004) have used sinusoids to identify cycling yeast genes. However, unlike our method these profiles require the a priori knowledge of the shape of the curve they wish to fit. In most cases, such knowledge is not available. Möller-Levet *et al*. (2003) present a method in which a comprehensive set of profiles is defined. Using these profiles, genes are clustered by assigning them to matching profiles. Unlike our method, their algorithm does not select a subset of the potential profiles and so the number of profiles grows exponentially in the number of time points. Thus, their algorithm is only appropriate for very short time series. In addition, their method cannot differentiate between patterns arising from random noise and patterns representing biological response. Thus, many of the resulting profiles do not represent true biological response. Peddada *et al*. (2003) suggest a method to specify expression profiles based on inequality constraints. Genes are assigned

to the profile for which they best match as determined by a statistical procedure. Unlike our method, their algorithm requires the user to specify the set of profiles in which she is interested. In addition, their method requires several repeats. Such large number of repeats are usually not obtained in time series experiments.

The method presented by De Hoon *et al*. (2002) fits linear splines when few time points and several repeats are available. The focus of their paper is different from ours, as they are interested in leveraging the statistical power of having several repeats to estimate the profile of a specific gene better and to determine if the gene is differentially expressed. In contrast, we focus on clustering, and our method can work even if no repeats are available by leveraging the statistical power obtained from the large number of genes being profiled simultaneously.

2 IDENTIFYING SIGNIFICANT EXPRESSION PATTERNS

Our algorithm starts by selecting a set of potential profiles. Next, genes are assigned to the profile that best represents them among the preselected profiles. We first discuss how to choose a representative set of profiles. Next, we discuss how we assign genes to profiles and how we determine the significance of each of the profiles, and then finally how to group them.

2.1 Selecting model profiles

As discussed in the introduction, we are interested in selecting a set of model expression profiles all of which are distinct from one another, but representative of any expression profile we would probably see. Here we assume that the raw expression values are converted into log ratios where the ratios are with respect to the expression of the first time point. The first value of the series after transformation will thus always be 0. To define a set of model profiles the user defines a parameter c that controls the amount of change a gene can exhibit between successive time points. For example, if $c = 2$ then between successive time points a gene can go up either one or two units, stay the same, or go down one or two units. Note that because our method relies on correlation (see below) 'one unit' may be defined differently for different genes. For n time points, this strategy results in $(2c+1)^{n-1}$ distinct profiles. Note that this method takes advantage of both facts, the fact that we are dealing with a time series (resulting in a limited set of values at the beginning compared to the end) and the fact that they are short (and so n is small). For example, 5 time points and $c = 1$ would result in $3^4 = 81$ model profiles. Since we are dealing with thousands of genes, many genes will be assigned to each of the 81 profiles allowing us to identify the significant profiles in this experiment.

While the above method generates a manageable number of profiles for short time series when $c = 1$, the number of profiles grows as a high order polynomial in c. For example, for 6 time points and $c = 2$ this method results in $5^5 = 3125$ model profiles which are too much for any user to view and are also likely to be very sparsely populated. In such cases we are interested in selecting a (manageable) subset of the profiles. Assume that we are interested in m representative profiles. There are a number of ways to select such a subset. Since the major purpose of the expression experiment is to identify distinct patterns of gene expression (which are likely to correspond to different functional categories), we would like to retain a distinct set of profiles. Computationally speaking, let P represent the $(2c+1)^{n-1}$ set of possible profiles. We would like to select a set $R \subset P$ with m profiles ($|R| = m$) such that the minimum distance between any two profiles in R is maximized. Such a set will guarantee that the profiles retained from P are as distinct as possible from each other. This requirement can be formalized by selecting a subset R which maximizes the following function:

$$\max_{R \subset P, |R|=m} \min_{p_1, p_2 \in R} d(p_1, p_2), \tag{1}$$

where d is a distance metric.

For a set R let $b(R) = \min_{p_1, p_2 \in R} d(p_1, p_2)$, i.e. $b(R)$ is the minimal distance between profiles in R, which is the quantity we wish to maximize. Let R' be the set of profiles that maximizes Equation (1). Thus, $b(R')$ is the optimal value we can achieve. Unfortunately, as we prove on the supporting website, finding such a set R' that maximizes this function is NP-Hard. Moreover, an approximation that guarantees a solution that is better than $b(R')/2$ is also NP-Hard. Both proofs rely on a reduction from the maximal independence set problem.

Thus, unless $P = NP$, the best polynomial algorithm for this problem can only guarantee a set R which achieves a value of $b(R')/2$. We now present a greedy algorithm that is guaranteed to find such a set, i.e. our algorithm finds a set of profiles R, with $b(R) \geq b(R')/2$. Our algorithm (Fig. 2) starts with one of the two types of extreme profiles (going down at each time point). Let R be the set of profiles selected so far. The next profile added to R is the profile p that maximizes the following equation:

$$\max_{p \in (P \setminus R)} \min_{p_1 \in R} d(p, p_1), \tag{2}$$

i.e. in each iteration we select the profile that is farthest from all the profiles selected so far (R). This selection is repeated until m profiles have been selected and the resulting set is returned. The top image of Figure 4 illustrates the profiles selected when $m = 50$ and $c = 2$.

The following theorem proves our claim about the optimality of this algorithm:

THEOREM 1. *Let d be a distance metric. Let $R' \subset P$ be the set of profiles that maximizes Equation (1). Let $R \subset P$ be the set of profiles returned by our algorithm, then $b(R) \geq b(R')/2$.*

procedure SelectVectorsMaxMinDist(d, P, m)
  let p_1 be the profile that always goes down one unit between time points
  $R = \{p_1\}$; (⋆ The set of selected vectors ⋆)
  $L = P \setminus \{p_1\}$;
  for $i = 2$ to m **do**
    let $p \in L$ be the profile that maximizes:
      $\mathbf{min}_{p_1 \in R} d(p, p_1)$;
    $R = R \cup \{p\}$; $L = L \setminus \{p\}$;
  end for;
  return R;

Fig. 2. Greedy approximation algorithm to choose a set of m distinct profiles.

This theorem is proved by considering two cases regarding the relationship between the set of profiles identified by the our algorithm (R) and the optimal set (R'). For both cases we can show that there exists a profile $p \in R$ that is at a distance at most $b(R)$ from two different profiles from R'. Thus, we can use the triangle inequality to show that $b(R')$ is at most twice $b(R)$. A formal proof of this theorem can be found in the Appendix.

The above algorithm performs m iterations and each of these takes at most $m(2c+1)^{n-1}$ time for a total running time of $O(m^2(2c+1)^{n-1})$. Since m is small (m should be at most 100 in order to be manageable), the total running time of this algorithm is small for short time series datasets (small n).

It is interesting to briefly discuss a related problem known as the k-centers problem (Hochbaum, 1997) . In our notations, the k-centers problem tries to find a group R that minimizes the following equation:

$$\min_{R \subset P, |R|=k} \max_{p_1 \in P \setminus R, p_2 \in R} d(p_1, p_2). \quad (3)$$

In other words, we are looking for a subset R of size k such that the maximum distance from points not in R to points in R is minimized. The k-centers problem tries to select centers that are the best representatives for the group while our goal is to find the most distinct profiles. Although in general, an optimal solution to one of these problems is not necessarily an optimal solution to the other, the algorithm we presented above is also known to be the best possible approximation algorithm for k-centers (the proof is obviously different). Thus, this algorithm provides the best of both worlds: a distinct subset which is also a good representation of the initial set of profiles P.

2.2 Identifying significant model profiles

Given a set M of model profiles and a set of genes G, each gene $g \in G$ is assigned to a model expression profile $m_i \in M$ such that $d(e_g, m_i)$ is the minimum over all $m \in M$. Here e_g is the temporal expression profile for gene g. If the above distance is minimized by $h > 1$ model profiles (i.e. we have ties) then we assign g to all of these profiles, but weight the assignment in the counts as $1/h$. We count the number of genes assigned to each model profile and denote this number for profile m_i by $t(m_i)$.

Next, we would like to identify model profiles that are significantly enriched for genes in our experiment. Our null hypothesis is that the data are memoryless, i.e. the probability of observing a value at any time point is independent of past and future values. Thus, according to the null hypothesis, any profile we observe is a result of random fluctuation in the measured values for genes assigned to that profile. Model profiles that represent true biological function deviate significantly from the null hypothesis since many more genes than expected by random chance are assigned to them.

Determining a parametric model for our null hypothesis is complicated by the many noise factors that affect gene-expression measurements. Instead, we follow many previous methods for static gene expression analysis (Dudoit *et al.*, 2002; Tsai *et al.*, 2003) and use a permutation based test. In our case, permutation is used to quantify the expected number of genes that would have been assigned to each model profile if the data were generated at random. Note that under the null hypothesis, the order of the observed values is random (as each point is independent of any other point) and thus permutations are expected to result in profiles that are similar to the null distribution.

Since there are n time points, each gene has $n!$ possible permutations, and all of these can be computed for small values of n. For each possible permutation we assign genes to their closest model profile. Let s_i^j be the number of genes assigned to model profile i in permutation j (j is one of the $n!$ possible permutations). We set $S_i = \sum_j s_i^j$. Then, $E_i = S_i/(n!)$ is the expected number of genes for each profile model if the data were indeed generated according to the null hypothesis. Note that different model profiles may have different number of expected genes and so in general $E_i \neq |G|/m$ (see Section 3).

Since each gene is assigned to one of the profiles, we can assume that the number of genes in each profile is distributed as a binomial random variable with parameters $|G|$ and $E_i/|G|$. Thus, the (uncorrected) P-value of seeing $t(m_i)$ genes assigned to profile p_i is $P(X \geq t(m_i))$, where $X \sim \text{Bin}(|G|, E_i/|G|)$. If we were testing just one model expression profile for significance then we could consider the number of genes assigned to p_i to be statistically significant at the α significance level if $P(X \geq t(m_i)) < \alpha$. However, since we are testing m model profiles for significance, we need to correct for the multiple comparisons. We

thus apply a Bonferroni correction and consider the number of genes assigned to p_i to be statistically significant if $P(X \geq t(m_i)) < \alpha/m$. The running time of the permutation test method is $|G|n!m$ which for small m and n is at most quadratic in the number of genes.

2.3 Correlation coefficient

While the above algorithm and approximation guarantees works with any distance metric, in this paper we focus on the use of the correlation coefficient $\rho(x, y)$. The correlation coefficient has enjoyed great success in computational biology, especially when used in a clustering algorithm (Eisen *et al.*, 1998). An advantage of the correlation coefficient for our method is that it can group together genes with similar expression profiles even if their units of change are different. However, while the correlation coefficient is useful, it can take negative values and does not satisfy the triangle inequality and thus is not a metric.

Instead we use the value $gm(x, y) = 1-\rho(x, y)$. This function while always greater or equal to 0, is still not a metric since it does not satisfy the triangle inequality. However, $gm(x, y)$ does satisfy a generalized version of the triangle inequality, specifically, we prove on the supporting website.

LEMMA 1. $gm(x, z) \leq 2(gm(x, y) + gm(y, z))$.

We further remark that the proof of the lemma actually gives an even tighter upper bound on $gm(x, z)$ as a function of the specific values of $gm(x, y)$ and $gm(y, z)$. Note that the above lemma proves that the correlation coefficient is a transitive measure. This might serve as a justification and motivation for the success of the correlation coefficient as it shows that when using the correlation coefficient two highly dissimilar profiles cannot be very similar to a third profile.[2]

2.4 Grouping significant profiles

The assignment of genes to model profiles is deterministic. However, due to noise, it is impossible to rule out close profiles (even if not the closest) as being the true profile for individual genes. If we have a measurement of the noise (e.g. from repeat experiments) it is possible to determine a distance threshold δ below which two model profiles are considered similar (the difference between genes assigned to these two may be attributed to noise). Such model profiles represent similar enough expression patterns and thus should be grouped together.

In order to determine which model profiles should be grouped together we transform this problem into a graph theoretic problem. We define the graph (V, E) where V is the set of significant model profiles and E the set of edges. Two profiles $v_1, v_2 \in V$ are connected with an edge iff $d(v_1, v_2) \leq \delta$. Cliques in this graph correspond to sets of significant profiles which are all similar to one another. There are many ways to partition a graph into a set of cliques. Here we are interested in identifying large cliques of profiles which are all very similar to each other. This naturally leads a greedy algorithm to partition the graph into cliques and thus to group significant profiles.

The greedy algorithm we use grows a cluster C_i around each statistically significant profile p_i. Initially, $C_i = \{p_i\}$. Next, we look for a profile p_j such that p_j is the closest profile to p_i that is not already included in C_i. If $d(p_j, p_k) \leq \delta$ for all profiles $p_k \in C_i$ we add p_j to C_i and repeat this process, otherwise we stop and declare C_i as the cluster for p_i. After obtaining clusters for all significant profiles, we select the cluster with the largest number of genes (by counting the number of genes in each of the profiles that are included in this cluster), remove all profiles in that cluster and repeat the above process. The algorithm terminates when all profiles have been assigned to clusters. The running time of this algorithm is $O(m'^4)$, where m' is the number of significant profiles, which is generally small.

3 RESULTS

We present the results for simulated and biological data. Our simulation results illustrate empirically that our method performs consistently with theoretical expectations. We then present results for using our algorithm to study the immune response system in humans. For this data we have also compared our results with clustering algorithms that have been used in the past to cluster short time series expression data.

3.1 Simulated results

We generated a dataset simulating 5000 genes with 5 time points. The raw expression value at each time point was randomly drawn from a uniform (10,100) distribution (other distributions yielded similar results). Each value was drawn independently of all other values, and the distribution was identical for all time points. Next we transformed this data to a log ratio representation. We applied our algorithm using 50 model profiles with a maximum unit change between time points of two. As expected, our algorithm determined that none of the profiles had a significant number of genes. Figure 3a plots the number of genes assigned to each profile against the number of genes expected. The region above the diagonal line corresponds to gene assignments levels that would be statistically significant at an $\alpha = 0.05$ Bonferroni corrected level or equivalently at an $\alpha = 0.001$ uncorrected level. Note that if we assume that the number of expected genes for each profile is the same ($5000/50 = 100$ in our case) then anything above the horizontal line would be considered statistically significant. The distribution of profiles on the graph illustrates that different temporal expression profiles are more likely than others to occur by random chance,

[2]Note that since gm does not fully satisfy the triangle inequality, the factor $1/2$ guarantee of our profile selection algorithm does not hold. However, we can still obtain a factor $1/4$ guarantee using the above algorithm as we show on the supporting website.

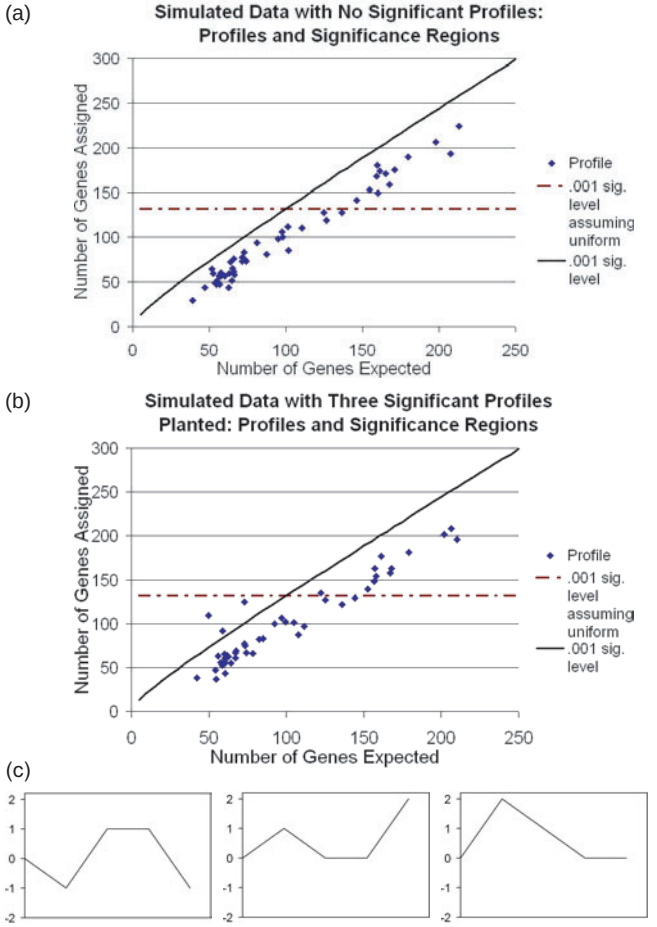

Fig. 3. Simulated data results. (**a**) Expected versus assigned number of genes for our first experiment. Points above the diagonal line correspond to profiles determined by our algorithm to be significantly enriched. The horizontal line corresponds to the same significance level if we assume that the number of expected genes for all profiles is the same. As can be seen our algorithm correctly determined that no profile is significantly enriched for genes, even though a number of profiles are above the horizontal line. (**b**) Similar plot for our second experiment. Our algorithm correctly identified all three planted profiles, even though each was planted with only 1% of the genes. (**c**) The three significant profiles found out the set of fifty considered. The fifty profiles considered are the same as they appear in Figure 4.

something which standard clustering algorithms do not take into account.

In our second experiment, we selected three profiles as appears in Figure 3c and assigned 50 genes (1%) to each of these profiles (the other 4850 genes were generated as described above). Log ratio values for genes assigned to a profile were set to a noisy version of the values of that profile by adding random noise to every time point of these genes (see Supplementary information for details). Figure 3b shows the results obtained for this data. The only three profiles which lie above the diagonal line are those for which the genes were planted. Thus, all three selected profiles were correctly recovered by our algorithm, and no other profile was determined to be significant. Note that the significant profiles had roughly half the number of genes assigned than a number of non-significant profiles. Such smaller, but more statistically significant cluster of genes could be overlooked by a traditional clustering algorithm.

3.2 Biological results

We tested our algorithm on immune response data from Guillemin *et al*. (2002). In the paper the authors used human cDNA microarrays to study the gene expression program of gastric AGS cells infected with various strains of *Helicobacter pylori*. *H.pylori* is one of the most abundant human pathogenic bacteria. In this paper we will analyze data from the response of the wild-type G27 strain. We use data obtained from two replicates on the same biological sample in which time series data were collected at 5 time points, 0, 0.5, 3, 6 and 12 h.

We first selected 2243 genes for further analysis from the 24,192 array probes. Genes were selected based on the agreement between the two repeats and their change at any of the experiment time points (see Supplementary website for details). We used a set of 50 model profiles (using more profiles yielded similar results, however, we believe that 50 is a manageable number and so we focus on this set here). For the results discussed below we generated the model profiles using a value of 2 for the maximum unit change parameter (c). Additional experiments with $c=1$ and $c=3$ returned very similar results (see Supplementary website for details). Of the 50 model profiles, 10 profiles in seven clusters were identified as significant. Figure 4 presents an image and plot of the clusters and profiles. Shaded profiles are significant and profiles with the same shade belong to the same cluster. We used a correlation of 0.7 ($\delta=0.3$) in our grouping method, where the value of 0.7 was obtained by using the similarity of the repeat data. Of the seven cluster of profiles one contained three profiles, one contained two profiles, and five were single profiles. Four of the 10 significant model profiles were significantly enriched for GO categories (as determined by the hypergeometric distribution), two of these profiles were assigned to the cluster containing three profiles while the others remained separate. We note that the array contained many unannotated genes, which could explain why not all profiles were significantly enriched for GO categories. Below we describe some of the significant profiles, and discuss their relevance to GO categories for which the profiles were enriched.

Profile 9 $(0,-1,-2,-3,-4)$ (Fig. 5) contained 131 genes that were downregulated during the entire experiment duration. This profile was significantly enriched for cell-cycle genes (P-value $< 10^{-10}$). Many of the cycling genes in this profile are known transcription factors, which could contribute to repression of cell-cycle genes, and, ultimately, the cell

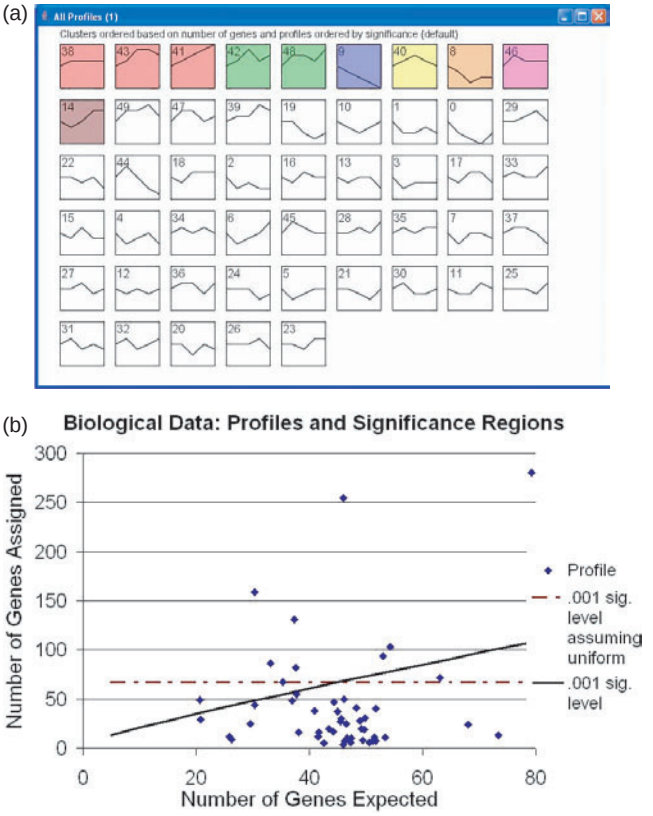

Fig. 4. Biological data results. (**a**) A screen shot of the main window to the software. Fifty distinct temporal profiles with a maximum unit change of two between time points is shown. The shaded profiles are statistically significant. Profiles of the same shade are grouped together. The algorithm was able to narrow the 50 initial profiles, to only 10 which were statistically significant. (**b**) A plot of the number of genes assigned to each profiles versus the expected number of genes. The ten above the diagonal line are those which are considered statistically significant. One of these profiles, profile 14, lies well below the horizontal line and would not be considered statistically significant if the number of genes assigned to each profile was assumed to be the same.

cycle (Guillemin *et al*., 2002; McCaffrey *et al*., 2002; Simon *et al*., 2001). Profile 14 $(0,-1,0,2,2)$, contained 49 genes. This profile is interesting since the raw number of genes assigned to the profile is not large and thus it could be missed by a clustering algorithm which ignores the sequential nature of the time series data. Genes assigned to this profile went slightly down at the beginning, but later were expressed at high levels. GO analysis indicates that many of these genes were relevant to cell structure and annotated as belonging to the categories cytoskeleton (P-value $= 9 \times 10^{-5}$), extracellular matrix (9×10^{-4}) and membrane (2×10^{-6}). Structural elongation of cells is a known phenotypical response to pathogens, and thus the enrichment of such genes in upregulated expression profile is consistent with this biological response (Guillemin *et al*., 2002; Huang *et al*., 2001). Profile 41 contained 86 genes that were going up during the entire experiment $(0,1,2,3,4)$. The most enriched GO category for this profile was response to stimulus (P-value $= 2 \times 10^{-5}$) which contains defense and immune response genes. Since the experiment involved pathogen infection, such a reaction from immune response genes is to be expected, and many of the unannotated genes in this profile might be also related to immune response.

We note that while the biological analysis in Guillemin *et al*. (2002) was largely anecdotal (focusing on a few key genes) many of these genes correspond to the above GO categories or to GO categories associated with the other significant profiles. Thus our work contributes a rigorous statistical justification for many of the observations made in the paper.

We compared our method with both, a general clustering algorithm (k-means) and an algorithm, designed specifically for time series data (CAGED) (Ramoni *et al*., 2002). We did not compare directly with hierarchical clustering since hierarchical clustering does not give a fixed number of clusters (cutting hierarchical clustering at a particular level in the tree resulted in few large clusters and many singletons). For k-means we used the Matlab 6.5 implementation of k-means with the correlation coefficient as the distance function (similar results were obtained when using Euclidean distance). Since k-means does not assign significance to the clusters it detects we used two version of k-means for this comparison. In the first version we clustered the entire set of 2243 genes with 10 clusters. In the second method, we generated 50 clusters and selected the 10 clusters with the most genes for further analysis. We used the third level of the GO hierarchy to compare our results with k-means. For each clustering result we computed the GO enrichment for the selected clusters and compared them to the enrichment detected using the profiles algorithm. Sixteen third-level GO categories had a P-value of at least 0.001 in one of the three clustering results. As Figure 6 shows, for most of the significant GO categories our algorithm identified a more coherent set of genes (resulting in a lower P-value) compared with either version of k-means. Some of the most biologically relevant categories, such as cellular physiological process, death, membrane and response to stimulus had P-values that were orders of magnitude lower using the profiles methods when compared with the k-means results. This is probably because of the inability of k-means to determine which of the clusters correspond to significant profiles and which are only the result of random noise.

For CAGED we used the recommended default settings including a Markov order of 1 except for consistency used correlation as our distance function (the results were similar for a Markov order of 2 and with Euclidean distance). CAGED returned five clusters. Four of the five clusters were not enriched for any GO category and the fifth was enriched with categories that are found in the entire set of 2243 genes. One of the main problems of CAGED was that it grouped too

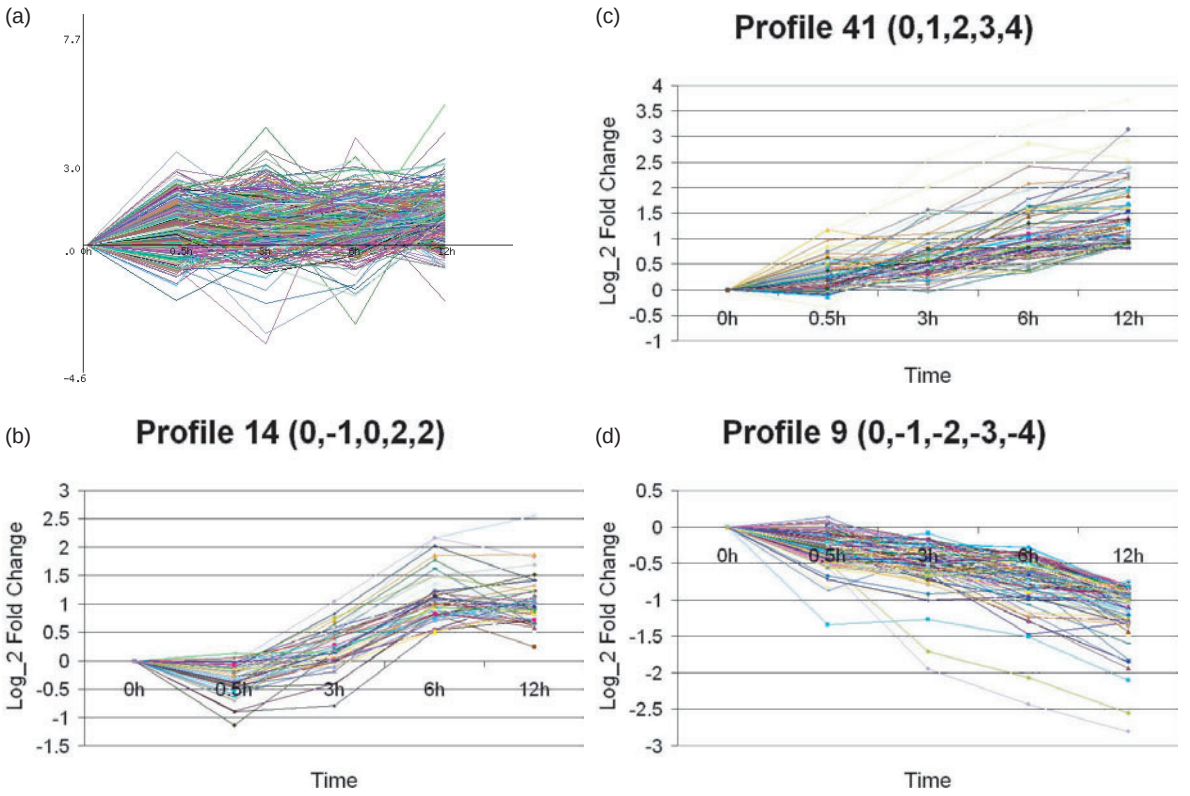

Fig. 5. A cluster from CAGED (**a**) containing all Profile 14 (**b**) genes and a substantial majority of Profile 41 (**c**) genes, among many other genes. As can be seen, the fact that so many genes are grouped together masks the presence of significant profiles identified by our algorithm resulting in low correlation with the relevant GO categories. Profile 9 is shown in (**d**).

many genes in one cluster. As can be seen in Figure 5, one of the CAGED clusters contained genes from both profiles 14 and 41, in addition to many other genes. The large set of genes masked the significant subsets that were contained in this profile, resulting in no significant GO category for this cluster. While CAGED is a very useful algorithm for long time series datasets, for short ones it seems like it does not have enough data to further separate the clusters. In contrast, our algorithm looks at all possible profiles (or a representative subset of them) allowing it to detect significant expression profiles even if only a small number of genes are associated with them.

4 CONCLUSIONS AND FUTURE WORK

Short time series expression datasets present unique challenges due to the large number of genes sampled and the small number of values for each gene. In this paper, we presented an algorithm which uses a set of model profiles to cluster the results of these experiments. The model profiles are selected independently from the data allowing our algorithm to determine the significance of the different clusters. This is a major advantage over other clustering algorithm that have been used for this task in the past since, due to noise and the small number of points, many patterns can be expected to arise at random.

Using simulated data we have shown that our algorithm can correctly identify small sets of genes planted in large random noise and can also distinguish between true and random patterns. Using immune response data we have shown that the patterns returned by our algorithm are in good agreement with the functional annotations of the associated sets of genes. Comparison with k-means and CAGED indicated that by focusing on the set of significant profiles our algorithm outperform these algorithms resulting in a much more coherent set of genes.

There are a number of possible future directions. First, if either c (the unit change) or n (number of time points) are large then even the potential set of model profiles can be too large to work with. In such a case we would like to develop a sampling strategy to help us first select a smaller set from which we would later choose the model profiles. It will also be interesting to extend this work by incorporating other types of data with the results of this method. For example, it would be useful to order the temporal profiles and figure out if later expressed profiles can be explained by binding motifs belonging to a transcription factor which is included in an early profile.

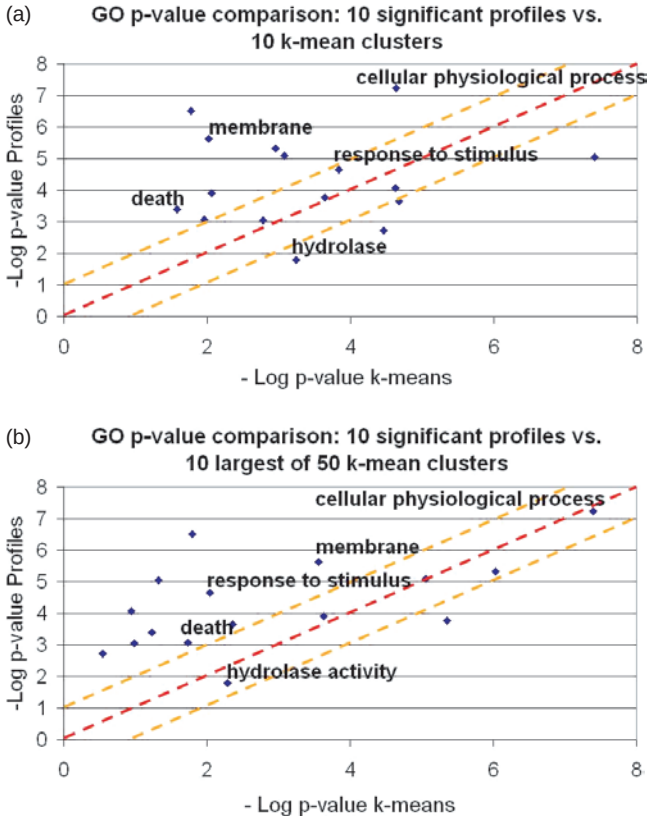

Fig. 6. Comparison of enriched third level GO categories between our algorithm and *k*-means. All categories that were enriched on one of the two algorithms were selected. *y*-axis: minus log *P*-value for GO enrichment using the profiles algorithm. *x*-axis: minus log *P*-value for *k*-means. (**a**) *k*-means with 10 clusters ($k = 10$) (**b**) *k*-means with 50 clusters focusing on the 10 most populated clusters. Points above the central diagonal line represent categories that were more enriched using the profile algorithm and below the line, categories more enriched using *k*-means. Points above (below) the light dashed lines represent differences greater than one order of magnitude between the two methods. As can be seen, most categories were much more enriched using the profiles algorithm. In particular, categories directly related to the experimental condition, such as cellular physiological process (cell cycle), death, membrane and response to stimulus were generally much better detected using our algorithm.

ACKNOWLEDGEMENTS

We would like to acknowledge Nilesh Dave, Anupam Gupta, Naftali Kaminski, Leonid Kontorovich and Ryan Williams for the useful discussions related to this work.

REFERENCES

Arbeitman,M.N., Furlong,E.E., Imam,F.J., Johnson,E., Null,B.H., Baker,B.S., Krasnow,M.A., Scott,M.P., Davis,R.W. and White,K.P. (2002) Gene expression during the life cycle of *Drosophila melanogaster*. *Science*, **298**, 2270–2275.

Bar-Joseph,Z., Gerber,G., Jaakkola,T.S., Gifford,D.K. and Simon,I. (2003) Continuous representations of time series gene expression data. *J. Comput. Biol.*, **3–4**, 341–356.

De Hoon,M.J., Imoto,S. and Miyano,S. (2002) Statistical analysis of a small set of time-ordered gene expression data using linear splines. *Bioinformatics*, **18**, 1477–1485.

Dudoit,S., Yee Hwa Yang,Y.H., Callow,M.J. and Speed,T.P. (2002) Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. *Statistica sinica*, **12**, 111–139.

Eisen,M.B., Spellman,P.T., Brown,P.O. and Botstein,D. (1998) Cluster analysis and display of genome-wide expression patterns. *Proc. Natl Acad. Sci.* USA, **95**, 14863–14868.

Gasch,A.P., Spellman,P.T., Kao,C.M., Carmel-Harel,O., Eisen,M.B., Storz,G., Botstein,D. and Brown,P.O. (2000) Genomic expression programs in the response of yeast cells to environmental changes. *Mol. Biol. Cell*, **11**, 4241–4257.

Gollub,J., Ball,C.A., Binkley,G., Demeter,J., Finkelstein,D.B., Hebert,J.M., Hernandez-Boussard,T., Jin,H., Kaloper,M., Matese,J.C., *et al*. (2003) The Stanford Microarray Database: data access and quality assessment tools. *Nucleic Acids Res.*, **31**, 94–96.

Guillemin,K., Salama,N., Tompkins,L. and Falkow,S. (2002) Cag pathogenicity island-specific responses of gastric epithelial cells to *Helicobacter pylori* infection. *Proc. Natl Acad. Sci. USA*, **99**, 15136–15141.

Hochbaum,D.S. (ed.) (1997) *Approximation Algorithms for NP-Hard Problems*. PWS Publishing Company, Boston, MA.

Huang,Q., Liu,D., Majewski,P., Schulte,L.C., Korn,J.M., Young,R.A., Lander,E.S. and Hacohen,N. (2001) The plasticity of dendritic cell responses to pathogens and their components. *Science*, **294**, 870–875.

Lu,X., Zhang,W., Qin,Z.S., Kwast,K. and Liu,J.S. (2004) Statistical resynchronization and Bayesian detection of periodically expressed genes. *Nucleic Acids Res.*, **32**, 447–455.

McCaffrey,R.L., Fawcett,P., O'Riordan,M., Lee,K.D., Havell,E.A., Brown,P.O. and Portnoy,D.A. (2002) A specific gene expression program triggered by gram-positive bacteria in the cytosol. *Proc. Natl Acad. Sci. USA*, **101**, 15136–15141.

Möller-Levet,C.S., Chu,K.H. and Wolkenhauer,O. (2003) DNA microarray data clustering based on temporal variation: Fcv with tsd preclustering. *Appl. Bioinformatics*, **2**, 35–45.

Peddada,S.D., Lobenhofer,E.K., Li,L., Afshari,C.A., Weinberg,C.R. and Umbach,D.M. (2003) Gene selection and clustering for time-course and dose-response microarray experiments using order-restricted inference. *Bioinformatics*, **19**, 834–841.

Ramoni,M.F., Sebastiani,P. and Kohane,I.S. (2002) Cluster analysis of gene expression dynamics. *Proc. Natl Acad. Sci. USA*, **99**, 9121–9126.

Schliep,A., Schonhuth,A. and Steinhoff,C. (2003) Using hidden Markov models to analyze gene expression time course data. *Bioinformatics*, **19**, i264–i272.

Shamir,R. and Sharan,R. (2002) Algorithmic approaches to clustering gene expression data. In Jiang,T., Smith,T., Xu,Y. and Zhang,M.Q. (eds), *Current Topics in Computational Biology*. MIT Press, pp. 259–299.

Simon,I., Barnett,J., Hannett,N., Harbison,C.T., Rinaldi,N.J., Volkert,T.L., Wyrick,J.J., Zeitlinger,J., Gifford,D.K., Jaakkola,T.S. and Young,R.A. (2001) Serial regulation of transcriptional regulators in the yeast cell cycle. *Cell*, **106**, 697–708.

Storch,K.F., Lipan,O., Leykin,I., Viswanathan,N., Davis,F.C., Wong,W.H. and Weitz,C.J. (2002) Extensive and divergent circadian gene expression in liver and heart. *Nature*, **418**, 78–83.

Tamayo,P., Slonim,D., Mesirov,J., Zhu,Q., Kitareewan,S., Dmitrovsky,E., Lander,E.S. and Golub,T.R. (1999) Interpreting patterns of gene expression with self organizing maps: methods and applications to hematopoietic differentiation. *Proc. Natl Acad. Sci. USA*, **96**, 2907–2912.

Tsai,C.A., Chen,Y.J. and Chen,J.J. (2003) Testing for differentially expressed genes with microarray data. *Nucleic Acids Res.*, **31**, e52.

Zhao,L.P., Prentice,R. and Breeden,L. (2001) Statistical modeling of large microarray data sets to identify stimulus–response profiles. *Proc. Natl Acad. Sci. USA*, **98**, 5631–5636.

APPENDIX

THEOREM 1. *Let d be a distance metric. Let $R' \subset P$ be the set of profiles that maximizes Equation (1). Let $R \subset P$ be the set of profiles returned by our algorithm, then $b(R) \geq b(R')/2$.*

PROOF. Set $b' = b(R')$ (b' is the optimal distance) and $b = b(R)$ (b is the distance returned by our algorithm). Let $\{r'_1, r'_2, \ldots, r'_{m-1}, r'_m\}$ be the profiles in R' and $\{r_1, r_2, \ldots, r_{m-1}, r_m\}$ be the profiles in R. Note that for any profile $p \in P$ there exists a profile $r_j \in R$ such that $d(p, r_j) \leq b$. If p is one of the profiles in R then let $r_j = p$, which gives $d(p, r_j) = d(r_j, r_j) = 0 \leq b$. If $p \notin R$ then there must be a profile in R with a distance at most b from p, otherwise the greedy algorithm would have selected p from R instead of r_m (we know that the minimum distance b was achieved by the last profile r_m). For each profile in R' we can find its closest profile in R. Next, we consider two possible cases, which are also the only possible cases:

Case 1. Two different profiles, $r'_i, r'_j \in R'$, are closest to the same profile $r_h \in R$: we note that $d(r'_i, r_h) \leq b$ and $d(r'_j, r_h) \leq b$ as mentioned above. Using the triangle inequality we get $2b \geq d(r'_i, r_h) + d(r'_j, r_h) \geq d(r'_i, r'_j) \geq b'$ and thus our solution is at least half of an optimal solution.

Case 2. No two vectors in R' are closest to the same vector in R: WLOG let r'_m be the vector which is closest to r_m (the last profile added by our algorithm). We next observe that there must exist $i \neq m$ such that $d(r'_m, r_i) \leq b$. This is so because if such a profile r_i did not exist then the greedy algorithm would have selected r'_m instead of r_m. Let r'_i be the profile from R' closest to r_i, then $d(r'_i, r_i) \leq b$ since all profiles are within b of a profile selected by the greedy algorithm. We thus have $2b \geq d(r'_m, r_i) + d(r'_i, r_i) \geq d(r'_i, r'_m) \geq b'$ which again shows that our solution is at least half of an optimal solution.