Phase-Independent Rhythmic Analysis of Genome-Wide Expression Patterns

Christopher James Langmead[*] **Anthony K. Yan**[*]

C. Robertson McClung[†] **Bruce Randall Donald**[*,‡,§,¶]

Abstract

We introduce a model-based analysis technique for extracting and characterizing rhythmic expression profiles from genome-wide DNA microarray hybridization data. These patterns are clues to discovering rhythmic genes implicated in cell-cycle, circadian, or other biological processes. The algorithm, implemented in a program called RAGE (Rhythmic Analysis of Gene Expression), decouples the problems of estimating a pattern's periodicity and phase. Our algorithm is linear-time in frequency and phase resolution, an improvement over previous quadratic-time approaches. Unlike previous approaches, RAGE uses a true distance metric for measuring expression profile similarity, based on the Hausdorff distance. This results in better clustering of expression profiles for rhythmic analysis. The confidence of each frequency estimate is computed using Z-scores. We demonstrate that RAGE is superior to other techniques on synthetic and actual DNA microarray hybridization data. We also show how to replace the discretized phase search in our method with an exact (combinatorially precise) phase search, resulting in a faster algorithm with no complexity dependence on phase resolution.

[*]Dartmouth Computer Science Department, Hanover, NH 03755, USA.

[†]Dartmouth Biology Department, Hanover, NH 03755, USA.

[‡]Dartmouth Chemistry Department, Hanover, NH 03755, USA.

[§]Dartmouth Center for Structural Biology and Computational Chemistry, Hanover, NH 03755, USA.

[¶]Corresponding author: 6211 Sudikoff Laboratory, Dartmouth Computer Science Department, Hanover, NH 03755, USA. Phone: 603-646-3173. Fax: 603-646-1672. Email: `brd@cs.dartmouth.edu`

Preprint: Proceedings of The Sixth Annual International Conference on Computational Molecular Biology (RECOMB), Washington, DC (2002) In press.

1 Introduction

The expression patterns of many genes associated with circannual (yearly), circadian (daily), cell-cycle and other periodic biological processes are known to be rhythmic. Conversely, the expression profiles of genes associated with aperiodic biological processes (e.g., tissue repair) are not rhythmic. The functional significance of previously uncharacterized genes, therefore, may be inferred if they exhibit rhythmic patterns of expression synchronized to some ongoing biological process.

DNA microarray experiments are an effective tool for identifying rhythmic genes when a time-series of expression levels are collected. Unlike Northern blots and real-time PCR, which study one gene at a time, DNA microarray hybridization time-series experiments can reveal the expression patterns of entire genomes. Chronobiologists are therefore able to assign putative functional properties to large numbers of genes based on the results of a single experiment. However, the large volume of data generated by hybridization experiments makes manual inspection of individual expression profiles impractical. Separating the subset of genes whose expression profiles are rhythmic from the thousands or tens of thousands that are not requires computer assistance. Ideally, the algorithms for providing such assistance should be efficient and have well-understood performance guarantees.

We have designed and implemented an algorithm to identify and characterize the properties of rhythmic genes from DNA microarray hybridization time-series data. Our approach specifically addresses issues of computational complexity, statistical significance and morphological similarity.

The identification of rhythmic genes may be viewed as a pattern-recognition problem — the goal is to identify sinusoidal RNA expression patterns in massively parallel gene expression data. Each expression profile may be viewed as a scalar function of time. A stored set of 'model' functions may be compared with an unknown function (expression profile) that has been obtained by

experiment. The model may be either (a) an 'ideal' synthetic sinusoid or (b) another gene expression profile. In (a), one tries to fit a family of different ideal sinusoids to the data, to determine if the data is periodic, and if so, the best fit. The sinusoids may differ in frequency, phase, and may be damped. In (b), the model may be a known rhythmic gene, in which case one attempts to find genes with similar profiles.

In either case, the difference between each model shape and unknown shape is computed, and the model that is closest to the unknown shape is reported as the best match. Strong arguments from the machine vision and pattern recognition literature argue that for such applications, the function used to measure the difference between model and data should be a metric [3, 21]. This means that for a class of expression profiles the difference function d should obey the following properties, for any three profiles X, Y, and Z:

$$d(X,Y) \geq 0 \text{ for all } X \text{ and } Y. \quad (1)$$

$$d(X,Y) = 0 \text{ if and only if } X = Y \text{ (Identity).} \quad (2)$$

$$d(X,Y) = d(Y,X) \text{ for all } X \text{ and } Y \text{ (Symmetry).} \quad (3)$$

$$d(X,Y) + d(Y,Z) \geq d(X,Z) \text{ for all } X, Y \text{ and } Z \text{ (Triangle Inequality).} \quad (4)$$

As argued in [10], the triangle inequality is of particular importance, because it guarantees that if several model expression profiles are similar to a given data expression profile, then these profiles also must be similar to one another. Thus, for example, it is not possible for two highly dissimilar model profiles to be similar to the data profile for the same gene. Current microarray analysis methods generally compare profiles using functions that are *not* metrics, and thus may report that several dissimilar models match the same data, which is highly counter-intuitive. In addition to obeying metric properties, an expression profile comparison method should also be easy to compute in order for it to be of practical use. The method we describe can be computed efficiently both in theory and in practice. Previous algorithms that use hierarchical clustering (e.g., [11]) run in time $O(n^2 l)$, where n is the number of genes represented in the microarray data and l is the number of time-series points. Other algorithms (e.g., [15]) that estimate both the frequency and phase of gene expression profiles using pattern recognition run in time $O(nmp\,l \log l)$, where m is the frequency resolution, and p is the phase resolution. These methods can take up to a week of wall-clock CPU time to analyze data from a single gene chip experiment and suffer from the use of non-metric similarity measurements. We describe an algorithm that runs in time $O(n(m+p)f(l))$, where $f(l)$ is the time to compute the Hausdorff distance ($f(l) = O(l^2)$ deterministic and $O(l)$ probabilistic time). Next, we replace our discretized phase search with an exact (combinatorially precise) phase search. This eliminates the factor of p entirely, resulting in an overall complexity of $O(nmf(l) + nl^3\alpha(l)\log l)$, where α is the extremely slow-growing inverse of Ackerman's function. In all cases, l may be treated as a small constant, since in today's technology, l is never more than a small constant $l_{max} \ll n$ (for example, typically, $l \leq 24$, and $n \approx 6500$ — See Table 1). This simplification obtains a complexity bound of $O(nmp)$ for previous algorithms vs. $O(n(m+p))$ and $O(nm)$ for ours. Our algorithm runs in 2-4 hours on a single processor Pentium-class workstation.

Our chief contributions are as follows:

1. The use of autocorrelation to define a phase-independent search over frequency- and phase-space. This allows us to perform two linear-time searches, one in frequency- and one in phase-space, as opposed to a quadratic-time search over frequency-phase space;
2. The use of the undirected Hausdorff (UH) distance to compare similarity of expression profiles. UH satisfies the axioms of a distance metric on the space of expression profiles, unlike previous measures, resulting in a robust and rigorous basis for clustering;
3. Testing our methods on publicly available gene expression data and a comparison of the results to previous analyses; and
4. The application of our methods to find circadian genes in a microarray data set that has previously been searched only for cell-cycle genes.

1.1 Organization of paper

We begin, in Section 2, with a review of the relevant biology and a summary of three publicly available DNA microarray hybridization time-series data sets. Section 3 categorizes existing techniques for extracting rhythmic profiles from microarray data, including a discussion of their limitations and computational complexity. In section 4, we detail our method and analyze its computational complexity. Section 5 presents the results of the application of RAGE to simulated and real biological data. Finally, section 6 discusses these results.

2 Background

There are many examples of DNA microarray time-series experiments in the literature (e.g., [8, 9, 14, 20, 22, 15, 24, 18, 27]). In many of these experiments, the primarily goal was to identify genes whose expression patterns were periodic over the length of the experiment. For example, cell-cycle regulated (e.g., [8, 24]) and circadian (e.g., [15, 27]) genes have been identified from their expression profiles in hybridization experiments.

Several research labs have made their raw data available to the public via [1] facilitating the development of improved techniques. The Campbell lab at Stanford has released the yeast data presented in [8] on the CDC28 mutant of yeast. The Botstein lab has released the data from their yeast experiment on the CDC15 mutant of yeast presented in [24]. The Brown lab at Stanford has released the human fibroblast data presented in [18]. The CDC15, CDC28, and fibroblast data sets are often used as benchmarks for novel microarray data processing techniques. In this section we briefly summarize the biological background relevant to these data sets.

2.1 Yeast and Fibroblast Data sets

The CDC15, and CDC28 experiments were designed to identify cell-cycle regulated genes in yeast (*Saccharomyces cerevisiae*). The eukaryotic cell-cycle is the 4 stage process by which a single cell replicates into two daughter cells. The four stages, named G1, S, G2 and M, have distinct roles. The chromosomes are prepared for replication in G1. The DNA and centrioles are replicated in S. The cell is prepared for separation in G2. Finally, the cell divides in M (mitosis) and the process begins again. This process takes about 90 minutes in yeast and 16 hours in fibroblasts. Thus, the authors of the CDC15, CDC28 and fibroblast experiments were looking for uncharacterized genes whose expression profiles were periodic with those frequencies.

The fibroblast experiment was also designed to identify cell-cycle regulated genes but in human fibroblast cells, instead of yeast. Unlike yeast, the cell-cycle for human fibroblasts is approximately 16 hours. Table 1 details the content of CDC15, CDC28, and fibroblast data sets.

3 Prior Work

A variety of techniques have been developed to extract the rhythmic genes from these data sets. The various techniques fall into two categories: *spectral* and *cluster-based* analyses. In this section we discuss each type, citing specific examples.

3.1 Spectral Techniques

The Fourier Transform is a standard tool for detecting periodicities in discretized signals. [24] used the Fourier transform as one component of a hybrid technique for determining the frequency and phase of gene expression profiles in the CDC15 and CDC28 data sets. The limitations of the Fourier transform are well understood. The range of detectable frequencies within a signal, and the resolution to which they can be resolved are particularly relevant to DNA microarray data. The frequency resolution obtainable on short time series, such as those generated in typical microarray experiments is often not adequate for resolving periodicities of interest. The interested reader is directed to Appendix A.1 for a longer discussion of these limitations.

The size limitations on the datasets are typically not biological but rather financial. For example, for *Arabidopsis* studies, each Affymetrix chip costs $400. That cost will increase to $500 when the full genome chip for *Arabidopsis* becomes available. There is also a $400 per chip processing fee. Thus, a 24-data point time series with a replication factor of 3 costs $57,600. When these costs drop and the number of points per experiment rises, the Fourier transform will become a more effective tool. Until that time, non-spectral techniques will likely dominate. We discuss these techniques in the next section.

Skiena and co-workers [12] give an algorithm for estimating frequencies of periodic genes, using the correlation coefficient and an unusual time-division strategy. While complexity bounds are not given, the algorithm appears to be efficient. The phase search in [12] is unique: they look for *aggregate* shifts. They find, for example, that the CDC15 data set is phase-shifted relative to the CDC28 set. Our method estimates the phase offset of individual genes relative to the start of the experiment.

3.2 Cluster-based Analysis

A clustering algorithm takes as input a set of items and a method for comparing the similarity between pairs of items. The outputs of the algorithm are subsets/clusters of the input set where the average similarity between pairs within a cluster is higher than the average similarity between items from different clusters. The second class of time-series analysis techniques clusters gene expression profiles to model profiles. Unknown genes are attributed the properties of the model to which they are most similar. The two most important distinctions among clustering methods are 1) how the models are generated and 2) which similarity measurements are used. The choice of models and similarity measurements affect both the complexity and accuracy of the resulting algorithm.

Experiment	Organism	Δt (minutes)	# samples	# periods	# genes
CDC15[24]	*S. cerevisiae*	10/20	24	3.2	6178
CDC28[8]	*S. cerevisiae*	10	17	1.8	6220
Fibroblast[18]	*H. sapiens*	15/30/60/120/240	12	1.5	9712

Table 1: CDC15, CDC28, and fibroblast data sets. Δt indicates the time period between successive time points. If there is more than one Δt listed, then the data was non-linearly sampled using a combination of the specified times. # periods indicates the number of cell-cycle periods that fit within the duration of the sample interval.

3.2.1 Model Generation

There are two primary means for generating models for a clustering algorithm. In the first approach the models are generated from the data itself. The clustering technique developed by [11] which was used for the original analysis of the CDC15, CDC28, and fibroblast data sets is an example of a data-generated clustering technique. It is often the case that among the microarray data there are a number of genes of known function. Such genes can be used as models. Data-derived models have a potential advantage in that they implicitly include noise models. There are non-trivial variations found among microarrays and in the steps leading up to hybridization. Consequently, the actual expression profiles of rhythmic genes sometimes deviate from an ideal model. When these variations are systematic across all genes, the use of data-derived models is especially useful. Whether or not such noise is in fact systematic is a valid concern.

The disadvantage of data-derived methods is that one has no control over the models. This places a particular burden on the similarity measurement used to compare profiles. The similarity measurement must be somewhat forgiving of small variations between otherwise similar shapes. Consider the following example (Fig 1 A & B), comparing the shapes of two sinusoids differing in phase by 90 degrees using the correlation coefficient (eq. 5):

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{[\sum_{i=1}^{n}(x_i - \overline{x})^2][\sum_{i=1}^{n}(y_i - \overline{y})^2]}} \qquad (5)$$

where x and y are the signals being compared and $\overline{x}$ and $\overline{y}$ are the respective means of those signals.

The correlation coefficient is used by quite a few clustering algorithms (e.g., [11, 12]). The correlation coefficient of the two signals in Fig. 1 A & B is 0, indicating that they are not similar. Clearly, the two shapes have a lot in common. Furthermore, the correlation coefficient violates 3 of the 4 criteria for being a metric outlined in Sec. 1. Perhaps most important violation is that the correlation coefficient does not satisfy the triangle inequality. A related statistical measurement, the cross-correlation, computes correlation coefficients over all relative shifts of the two input signals. Consequently, the cross-correlation is also a non-metric.

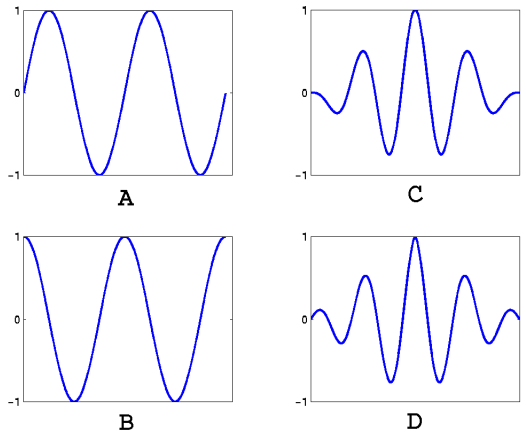

Figure 1: The sinusoid in A is 90 degrees out of phase with the sinusoid in B. The correlation coefficient (eq. 5) of A and B is 0, or no correlation. The signal in C is the autocorrelation of A. The signal in D is the autocorrelation of B. Note they are nearly identical due to the phase-independent nature of the autocorrelation. Furthermore, unlike the Fourier transform, the autocorrelation is not affected by non-linearly sampled data. Therefore, it is more suitable for a larger class of microarray experiments.

Another approach to model-based clustering is implemented in the program CorrCos [15]. CorrCos clusters gene expression data to *synthetic* sinusoidal models using cross correlations. The advantage of synthetic models is that the program has complete control over the models. CorrCos generates one thousand sinusoidal models of differing frequencies. For each frequency model, 101 phase variations are also generated. Each gene expression profile is then compared, using the cross-correlation, to each of the 101,000 synthetic models. The frequency and phase of the model most closely matching the expression profile are assigned to that gene.

The time complexity of CorrCos is $O(nmpl \log l)$, where n is the number of genes, m is the number of frequency models, p is the number of phase variants generated for each frequency model, and l is the length of the time series. $O(l \log l)$ is the time to compute a cross correlation. Hence this brute-force search over frequency and phase grows quadratically with the frequency and phase resolution (m and p) of the search.

In practice, CorrCos (compiled FORTRAN code) takes about 1 week to run on microarray data of size 32,000 genes. In contrast, our method only takes about 4 hours (interpreted MATLAB code) on the same data set.

In summary, there are a number of problems with the existing approaches for detecting and characterizing rhythmic genes in microarray time-series data. Spectral methods are not appropriate because the typical microarray experiment generates relatively short time-series. Consequently, cluster-based techniques are required. Of the cluster-based approaches, data-derived models are problematic because they lack flexibility and, when using a phase-sensitive similarity measurement such as the correlation coefficient, can lead to nonintuitive results. Synthetic model-based clustering addresses some of these problems at the expense of computational complexity. Any generate-and-test based technique using a phase-dependent similarity measurement will have similar properties.

In contrast, we introduce the *autocorrelation* to render our search algorithm phase-independent (Sec. 4.1). The (hypothetical) use of a *non-metric* similarity measurement (such as the correlation coefficient) on an autocorrelated phase-independent representation would still suffer from the drawbacks presented in Sec. 1. Therefore, we employ the *Hausdorff distance* (a true metric) on the autocorrelated signals in order to obtain superior matching and clustering performance (Sec. 4.2). By attacking the problem of time-series analysis using these new methods, we hope to strengthen the computational armamentarium of the chronobiologist.

4 RAGE (Rhythmic Analysis Of Gene Expression)

We have developed an algorithm named RAGE to addresses these problems. Like CorrCos, RAGE is a synthetic model-based clustering technique. However, RAGE is more efficient than CorrCos, running in time $O(n(m+p)f(l))$, where $f(l)$ is the time to compute the undirected Hausdorff distance between an l-point model function and an l-point gene expression profile (Sec. 4.2). Hence, the time complexity of RAGE grows linearly with increased frequency (m) and phase (p) resolution. RAGE achieves the better bound by using phase-independent transformations of the data and models. RAGE also utilizes a true mathematical metric, the undirected Hausdorff distance metric, for computing expression profile similarities. A summary of the RAGE algorithm is given in Fig. 2.

4.1 Phase-Independence

The computational complexity of the CorrCos algorithm stems from a brute-force search over the space of frequency and phase. We claim it is not necessary to search this entire space. It is possible to decouple the phase and frequency searches by transforming both the raw data and the models into a phase-independent form. The autocorrelation Ψ_x of a signal x is a frequency-sensitive but phase-independent representation of x:

$$\Psi_x(l) = \int_{-\infty}^{+\infty} x(t)x(t-l)dt \tag{6}$$

where x is the function, $x(t)$ is the function evaluated at t, and $x(t-l)$ is a translated version the function. Figures 1 C and D show the autocorrelation of the two signals from Fig. 1 A and B, respectively. Note that the two autocorrelations are nearly identical. The interested reader is directed to Appendix A.2 for more information on the autocorrelation. In our implementation, each signal is zero-meaned and normalized prior to computing the autocorrelation.

RAGE estimates the frequency of expression profiles using the autocorrelations of both the model and gene data. After the frequency has been estimated, phase-variations of the winning frequency model are generated and a second clustering is performed using these phase models and the original, non-autocorrelated data. Note that the phase search is conducted over a single frequency, and not all frequencies (unlike CorrCos).

4.2 Undirected Hausdorff Distance

RAGE uses the Undirected Hausdorff distance metric (UH) instead of the correlation coefficient to cluster expression profiles. The undirected Hausdorff distance (H, Eq. (7)) calls the directed Hausdorff distance (h, Eq. (8)) as a subroutine [10, 16]:

$$H(A,B) = \max\big(h(A,B), h(B,A)\big) \tag{7}$$

$$h(A,B) = \max_{a\in A}\min_{b\in B}\big(\rho(a,b)\big), \tag{8}$$

where A and B are two point sets in the plane and ρ is a distance metric. In our implementation ρ is the Euclidean distance in the plane between two points. ρ can, in fact, be any metric. The UH distance essentially measures the maximum outlier between two sets of points. It is commonly used in the computer vision community for object recognition tasks. We treat the model and data expression profiles as scalar functions; the Hausdorff distance is used to compare the similarity of their graphs, treated as point sets, in the plane. The deterministic time-complexity $f(l)$ for computing the Hausdorff distance is $O(l^2)$. UH distance can be computed on curves (such as our functions) in expected time $O(l)$ [5]. One potential limitation of the UH is that it can be sensitive to outliers. However, this can be easily solved using *quantiles* [10]. The basic strategy is to compute the UH on a subset of the complete set of

```
Let G ← Gene Data
Let A ← AutoCorr(G)
for ω = ω_lo to ω_hi
        M_ω ← AutoCorr(GenerateModel(ω, 0))
for i = 1 to num-genes
        Ω_i ← ClusterFreq(A_i,M)
for i = 1 to num-genes
        for φ = φ_lo to φ_hi
                P_φ ← GenerateModel(Ω_i,φ)
        Φ_i ← ClusterPhase(G_i,Ω_i,P)
for i = 1 to num-genes
        Z_i ← Z-Score(G_i, Ω_i, G)
```

Figure 2: RAGE algorithm. The function AutoCorr takes a set of gene expression profiles as input and returns a set of autocorrelated gene expression profiles. GenerateModel takes a frequency (ω) and phase (ϕ) as input and returns a sinusoid of the specified frequency and phase. The length of the model is assumed to be the same size as the length of the expression profiles in G. M is the set of autocorrelated (i.e., phase-independent) models, and M_ω is the model of frequency ω. ClusterFreq takes as an autocorrelated expression profile and a set of models and returns the frequency of the model that has the lowest Hausdorff distance to the input expression profile. Ω is the set of frequency assignments, as determined by ClusterFreq and Ω_i is the frequency assigned for each gene G_i in G. P is the set of phase-sensitive models, and P_ϕ is the model of phase ϕ and frequency Ω_i. ClusterPhase takes as input a single gene expression profile, the frequency that has been assigned to that gene and a set of phase-sensitive models and returns the phase of the model that has the lowest Hausdorff distance to the input expression profile. Note that P, the set of models for a given frequency (Ω_i) can be cached for better performance. Φ is the set of phase-assignments, as determined by ClusterPhase, for each of the genes in G, and Φ_i is the phase for gene G_i. Z-Score takes as input an expression profile, the frequency assigned to that gene and G. Z-score randomly chooses a constant-sized set R of genes from G. The distribution of Hausdorff distances between G_i and the expression profiles in R is used to compute the Z-score of the Hausdorff distance between the model of frequency Ω_i and G_i.

distances measured using ρ. For example, h might be computed using only the 75 % quantile of maximum distances. However, by using quantiles some metric properties of the UH are lost. We did not use quantiles in our implementation.

4.3 Statistical Significance

Any cluster-based approach computes the similarity between each gene and model. The gene is assigned to the model with the highest similarity. In algorithms such as CORRCOS and RAGE, all of the models are rhythmic. Thus, aperiodic genes are initially clustered to rhythmic models. A post-processing step is required to filter such genes from each cluster. There are number of ways to approach this problem. RAGE employs a statistical approach.

The Z-score ($z = \frac{x-\mu}{\sigma}$ where x is a member of the distribution, μ is the mean of the distribution and σ is the standard deviation), is a common statistical technique for estimating the significance of a given score. Given a normal distribution of similarity scores, the Z-score of any individual similarity measurement is the number of standard deviations from the mean that score lies. The Z-score can in turn be transformed into a probability ($P(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}}$). Hence, the confidence of a given estimate is $1 - P(z)$.

For each gene, RAGE pre-computes a distribution of UH distances between the gene and a constant number of randomly selected genes from the data set[1]. The confidence associated with the UH distance of the winning frequency model is computed using this distribution. One-sided probabilities are used so that only statistically small UH distances are considered. The biologist may then sort the genes by these confidence scores.

4.4 Algorithmic Complexity

The analysis of RAGE is as follows. The conversion of the raw gene data to its autocorrelated form can be computed in time $O(nl \log l)$ where n is the number of genes and l is the length of the time-series for each gene. The generation of the set of models, M is computed in time $O(ml \log l)$ where m is the number of models (determined by the desired frequency resolution, m). The clustering of each gene to its nearest model is done by computing the Hausdorff distance

[1] In our experiments, 500 other gene expression profiles were randomly selected.

between each of the n genes and each of the m models. This can be done in time $O(nmf(l))$, where $f(l)$ is the time to compute the Hausdorff distance (Sec. 4.2). The creation of the set P of phase models takes $O(npl)$ time where p is the number of phase models (determined by the desired phase resolution, p). The clustering of each (frequency assigned) gene to the nearest phase model takes $O(npf(l))$. Finally, computing the Z-scores takes time $O(cnf(l))$ where c is a constant corresponding to the number of genes randomly selected to compute the distribution required for calculating Z-scores. Note that c is determined by the need for statistical significance. It does not grow as the size of n increases. This gives an overall complexity of $O(nl\log l + ml\log l + nmf(l) + npl + npf(l) + cnf(l))$. Since l is $O(1)$ in today's experiments, the expression reduces to $O(n(m+p))$, asymptotically faster than CorrCos' complexity of $O(nmp)$. In the future, if genechip experiments become cheaper, l will not be a constant, although we can still expect $l \ll n$, and $l \ll m$ is likely for the foreseeable future. Hence, our algorithm would run in time $O(n(m+p)l^2)$ (deterministic) or $O(n(m+p)l)$ (probabilistic) time, as opposed to $O(nmpl\log l)$ for CorrCos.

4.5 Exact Phase Search and Improved Algorithm

We can improve on this algorithm as follows. For each of the n genes, the phase search as presented above takes time $O(pf(l))$ per gene. We can replace this discretized phase search with an exact (combinatorially precise) phase search, with running time $O(l^3\alpha(l)\log l)$ per gene, where α is the inverse of Ackerman's function. This eliminates the factor of p entirely, resulting in an overall complexity of $O(nmf(l) + nl^3\alpha(l)\log l)$. We achieve this as follows. Given two point sets A and B in the plane, [10] gives an algorithm for computing

$$\operatorname*{argmin}_t H(A_t, B), \tag{9}$$

where A_t is a rigid translation of set A by vector $t \in \mathbb{R}^2$. If A and B have p and q points, respectively, the algorithm runs in time $O(pq(p+q)\alpha(pq)\log(pq))$.

Definition 1 (Voronoi Surface [10]) *Given a set S of points in $\mathbb{R}^2$, and a metric ρ on $\mathbb{R}^2$, consider the function $d(x) = \min_{s\in S}\rho(x,s)$. The* Voronoi surface of S *is the graph of d, $\{(x, d(x)) \mid x \in \mathbb{R}^2\} \subset \mathbb{R}^3$.*

The Voronoi surface is a two dimensional surface in a three dimensional space. Intuitively, the Voronoi surface looks like an "irregular egg carton". Following [10], to determine the translation t that minimizes Eq. (9), we must identify the value of t that minimizes the upper envelope U of all the Voronoi surfaces defined by the sets $S_i = a_i \ominus B$ and $S'_j = A \ominus b_j$, for all $a_i \in A$ and all $b_j \in B$. $\ominus$ denotes the *Minkowski difference*, so that $S_i = \{a_i - b \mid b \in B\}$, and $S'_j = \{a - b_j \mid a \in A\}$. The Voronoi surface has $O(pq(p+q)\alpha(pq))$ local minima. Hence,

Claim 2 [10] *The minimum undirected Hausdorff distance under translation of two sets of points in the plane (and the translation that achieves this minimum) can be computed in time $O(pq(p+q)\alpha(pq)\log(pq))$.*

We use this algorithm as follows. B is the expression profile for a gene G_i. $A = \Omega_i$ is a frequency model, and A_t is the frequency model translated by t. We restrict t to be a phase shift (pure one-dimensional x-translation), as follows. The algorithm finds the minimum of the upper envelope U. We are only interested in a one-dimensional translation (the phase) along the x-axis. We can use the [10] algorithm, by intersecting the upper envelope U of the Voronoi Surfaces, with the plane $\pi_y^{-1}(0)$, where $\pi_y : \mathbb{R}^3 \longrightarrow \mathbb{R}^1$, $\pi_y(x,y,z) = y$, and then finding minima in $U \cap \pi_y^{-1}(0)$. It is also possible to construct the Voronoi surface of a collection of one-dimensional distance functions directly, as a 1D surface in a 2D space.

5 Results

rage has been applied to both synthetic and real microarray data. In section 4, we demonstrated that rage is computationally more efficient than CorrCos due to the use of a phase-independent search. Our second claim is that the Hausdorff distance metric is superior to the correlation coefficient as a distance measure. To assess the benefit of using the Hausdorff distance metric, we implemented two versions of rage. The first one is as described in this paper. The second implementation (rage-cc) is exactly the same as rage except that it uses the correlation coefficient to compare function shapes instead of the Hausdorff distance metric.

5.1 Simulated Data

Our first comparison of rage vs. rage-cc was on a set of 6500 simulated gene profiles. rage outperforms rage-cc in accuracy of phase and frequency estimation. That is, the average difference between the *actual* periodicity of the synthetic gene and the estimated frequency is smaller when using the Hausdorff metric. The interested reader is directed to Appendix A.3 for more details on this experiment.

5.2 CDC15, CDC28 and Fibroblast Data

We applied rage and rage-cc to the CDC15, CDC28 and fibroblast data sets. The results of that analysis are in Tables 2 A-C. rage is more accurate than rage-cc

CDC15	RAGE	RAGE-CC
Mean $\Delta\omega$ (minutes)	4.39	7.99
St. Dev $\Delta\omega$ (minutes)	15.05	24.41

(A)

CDC28	RAGE	RAGE-CC
Mean $\Delta\omega$ (minutes)	8.57	11.62
St. Dev $\Delta\omega$ (minutes)	15.8	35.05

(B)

Fibroblast	RAGE	RAGE-CC
Mean $\Delta\omega$ (hours)	2.68	3.05
St. Dev $\Delta\omega$ (hours)	2.89	3.12

(C)

Gene	Periodicity (hours)
Cry1	21
BMAL	24
Per2	24

(D)

Table 2: Results on actual microarray data. RAGE is more accurate than RAGE-CC at estimating the periodicity of cell-cycle regulated genes. A & B compare RAGE and RAGE-CC's accuracy at estimating the periodicity of 104 known cell cycle-regulated yeast genes identified by [26, 7, 23, 19, 2, 13, 25, 17, 6]. The periodicity of those genes is assumed to be 90 minutes. $\Delta\omega$ is the average difference (in minutes) between each program's estimate and 90 minutes. (C) compares RAGE and RAGE-CC's accuracy at estimating the periodicity of the 517 fibroblast genes identified in [18]. The periodicity of those genes is assumed to be 16 hours. $\Delta\omega$ is the average difference (in hours) between the program's estimate and 16 hours. (D) 3 of the 7 genes identified as circadian in [27] are present in the fibroblast cell-cycle data set. RAGE correctly estimates the frequency of those genes.

at estimating the frequency on the real microarray data. Furthermore, on a list of 104 genes reported to be cell cycle-regulated [26, 7, 23, 19, 2, 13, 25, 17, 6], RAGE finds 6 of the 9 genes that were missed by [24]. Figures 3 A-C show some selected clusters of genes that RAGE estimated as highly periodic.

As noted in Sec. 2.1, the human fibroblast data set was collected over a 24-hour time period. While not a true circadian experiment, it is known that the clock genes are expressed in peripheral tissues such as fibroblasts [4]. Hence, it should be possible to look for circadian genes in the same data set. This had not been done before; we ran RAGE to try it. Our results were compared to [27], who found circadian genes in cultured *rat* fibroblasts, using (different) microarray experiments and data. RAGE processed the human microarray fibroblast data set and found three out of the seven genes (*Cry1, Per2* and *BMAL*) that are close human homologs of the rat circadian genes identified by [27]. In Table 2 D, RAGE's estimate of periodicity for *Cry1, Per2* and *BMAL* is reported. RAGE successfully estimates the periodicity of these three genes as being circadian. Figure 3 D shows the expression profiles for *Cry1, Per2* and *BMAL* from the human fibroblast data set.

6 Conclusion

Genome-wide RNA expression time-series experiments are an important source of biological information. The discovery of periodic gene expression profiles is especially useful for the study of rhythmic processes like the cell cycle and the circadian clock. The sheer volume of data generated by microarray experiments prohibits manual inspection of all the data. Therefore, algorithms for identifying rhythmic genes are needed.

Fourier-based techniques are not yet appropriate for microarray data because the number of time-points in a typical experiment is too small to yield adequate frequency resolution. Model-based techniques are therefore needed. Of the existing model-based techniques, those using synthetic models, like CorrCos, are very accurate, but are computationally inefficient. We have presented a novel technique that is computationally much more efficient. It gains its efficiency through the use of a phase-independent search of frequency-space. Furthermore, RAGE uses a true mathematical metric to compute the similarity between gene expression profiles. The Hausdorff distance is a more accurate measurement and therefore, RAGE tends to give more accurate frequency and phase estimates on both simulated and actual microarray data. The Hausdorff distance may prove useful in other microarray data applications, such as clustering. We are presently exploring its performance in that application.

Acknowledgments. We would like to thank Ryan Lilien and all members of Donald Lab for helpful discussions and suggestions. We thank Dr. M. Straume, author of CorrCos [15], for reading and commenting on a draft of this paper. Thanks to Drs. J. M. Groh and H. Farid for their suggestions on the signal processing and statistical aspects of RAGE. This work is supported by the following grants to B.R.D.: a Guggenheim Foundation Fellowship, National Science Foundation grants IIS-9906790, EIA-9901407, EIA-9802068, CDA-9726389, EIA-9818299, CISE/CDA-9805548, IRI-9896020, IRI-9530785, EIA-0102710, and EIA-0102712, U.S. Department of Justice contract 2000-DT-CX-K001, and an equipment grant from Microsoft Research. C.R.M. was supported by National Science Foundation grants MCB-9723482 and MCB-0091008.

References

[1] Stanford Microarray Database. http://genome-www4.stanford.edu/MicroArray/SMD/.

[2] H. Araki, R. K. Hamatake, A. Morrison, A. L. Johnson, L. H. Johnston, and Sugino A. Cloning DPB3, the gene encoding the third subunit of DNA polymerase II of *Saccharomyces cerevisiae*. *Nucleic Acids Res*, 19:4867–4872, 1991.

[3] E. Arkin, L. P. Chew, D. P. Huttenlocher, K. Kedem, and J. S. B. Mitchell. An efficiently computable metric for comparing polygonal shapes. *IEEE Trans. Patt. Anal. Mach. Intell.*, 13(3):209–216, 1991.

[4] A. Balsalobre, F. Damiola, and U. Schibler. A Serum Shock Induces Circadian Gene Expression in Mammalian Tissue Culture Cells. *Cell*, 93:929–937, 1998.

[5] E. Belogay, C. Cabrelli, U. Molter, and R. Shonkwiler. Calculating the Hausdorff distance between curves. *Information Processing Letters*, 64(1):17–22, 1997.

[6] L. H. Caro, G. J. Smits, P. van Egmond, J. W. Chapman, and F. M. Klis. Transcription of multiple cell wall protein-encoding genes in *Saccharomyces cerevisiae* is deferentially regulated during the cell cycle. *FEMS Microbiol. Lett*, 161:345–349, 1998.

[7] J. W. Chapman and Johnston L. H. The yeast gene, *dbf4*, essential for entry into s phase is cell cycle regulated. *Exp. Cell Res.*, 180:419–428, 1989.

[8] R. Cho, M. Campbell, E. Winzeler, L. Steinmetz, A. Conway, L. Wodicka, T. Wolfsberg, A. Gabrielian, D. Landsman, D. Lockhart, and R. Davis. A genome-wide transcriptional analysis of the mitotic cell cycle. *Mol. Cell*, 2:65–73, 1998.

[9] A. Claridge-Chang, H. Wijnen, F. Naef, C. Boothroyd, N. Rajewsky, and M. W. Young. Circadian Regulation of Gene Expression Systems in the Drosophila Head. *Neuron*, 32:657–671, 2001.

[10] B. R. Donald, D. Kapur, and J. Mundy. *Symbolic and Numerical Computation for Artificial Intelligence*, chapter 8, "Distance metrics for comparing shapes in the plane," by D. Huttenlocher and K. Kedem, pages 201–219. Academic Press, Harcourt Jovanovich, London, 1992.

[11] M. Eisen, P. T. Spellman, D. Botstein, and P. O. Brown. Cluster Analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci.*, 95(25):14863–14868, 1998.

[12] V. Filkov, S. Skiena, and J. Zhi. Analysis Techniques for Microarray Time-Series Data. *Proc. of the 5th Ann.Intl. Conf. on Comput. Biol.*, pages 124–131, 2001.

[13] I. Fitch, C. Dahmann, U. Surana, A. Amon, K. Nasmyth, L. Goetsch, B. Byers, and Futcher B. Characterization of four B-type cyclin genes of the budding yeast *Saccharomyces cerevisiae*. *Mol. Biol. Cell*, 3:805–818, 1992.

[14] C. Grundschober, F. Delaunay, A. Phlhofer, G. Triqueneaux, V. Laudet, T. Bartfai, and P. Nef. Circadian Regulation of Diverse Gene Products Revealed by mRNA Expression Profiling of Synchronized Fibroblasts. *J. Biol. Chem.*, 276:46751–46758, 2001.

[15] S. Harmer, J. B. Hogenesch, M. Straume, H. S. Chang, B. Han, T. Zhu, X. Wang, J. A. Kreps, and S. A. Kay. Orchestrated Transcription of Key Pathways in Arabidopsis by the Circadian Clock. *Science*, 290:2110–2113, 2000.

[16] D. P. Huttenlocher and K. Kedem. Computing the minimum Hausdorff distance for point sets under translation. *Proc. 6th ACM Symp. Computational Geom.*, pages 340–349, 1990.

[17] J. C. Igual, A. L. Johnson, and L. H. Johnston. Coordinated regulation of gene expression by the cell cycle transcription factor Swi4 and the protein kinase C MAP kinase pathway for yeast cell integrity. *EMBO J.*, 15:5001–5013, 1996.

[18] V. R. Iyer, M. B. Eisen, D. T. Ross, G. Schuler, T. Moore, J. C. F. Lee, J. M. Trent, L. M. Staudt, J. Jr Hudson, M. S. Boguski, D. Lashkari, D. Shalon, D. Botstein, and P. O. Brown. The transcriptional program in the response of human fibroblasts to serum. *Science*, 283:83–87, 1999.

[19] L. H. Johnston, J. H. White, A. L. Johnson, G. Lucchini, and P. Plevani. Expression of the yeast DNA primase gene, PRI1, is regulated within the mitotic cell cycle and in meiosis. *Mol. Gen. Genet.*, 221:44–48, 1990.

[20] M. J. McDonald and M. Rosbash. Microarray analysis and organization of circadian gene expression in Drosophila. *Cell*, 107:567–578, 2001.

[21] D. Mumford. The problem of robust shape descriptors. *Proc. 1st Int. Conf. Comput. Vision*, pages 602–606, 1987.

[22] R. Schaffer, J. Landgraf, M. Accerbi, V. Simon, M. Larson, and E. Wisman. Microarray analysis of diurnal and circadian-regulated genes in Arabidopsis. *Plant Cell*, 13:113–123, 2001.

[23] W. Siede, G. W. Robinson, D. Kalainov, T. Malley, and E. C. Friedberg. Regulation of the RAD2 gene of *Saccharomyces cerevisiae*. *Mol. Microbiol*, 3:1697–1707, 1989.

[24] P. Spellman, G. Sherlock, M. Q. Zhang, R. I. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher. Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by Microarray Hybridization. *Mol. Biol. Cell*, 9(12):3273–3297, 1998.

[25] J. Wan, H. Xu, , and M. Grunstein. Cdc14 of *Saccharomyces cerevisiae*. Cloning, sequence analysis, and transcription during the cell cycle. *J. Biol. Chem.*, 267:11274–11280, 1992.

[26] J. H. White, S. R. Green, D. G. Barker, L. B. Dumas, and L. H. Johnston. The *cdc8* transcript is cell cycle regulated in yeast and is expressed coordinately with *cdc9* and *cdc21* at a point preceding histone transcription. *Exp. Cell Res.*, 171:223–231, 1987.

[27] K. Yagita, F. Tamanini, G. T. J. van der Horst, and H. Okamura. Molecular Mechanisms of the Biological Clock in Cultured Fibroblasts. *Science*, 292:278–281, 2001.

A Appendix

A.1 Fourier Transform

The Nyquist frequency is the maximum frequency that can be detected within a sampled signal. It is one half the sampling rate. The minimum frequency that can be detected in a sampled signal is determined by the sampling interval. If the signal is collected over n time units, then the lowest detectable frequency is 1 cycle per n time units. The resolution of a spectrum is determined by the number of points collected. If there are m points in the signal, the resulting spectrum will have m equally spaced frequency intervals from the lowest frequency to the Nyquist frequency.

An important assumption made by the Fourier Transform is that the data is linearly sampled. The CDC15 data set, for example, is non-linearly sampled, making the Fourier transform inappropriate. Of course, it is sometimes possible to discard data points in order to create a linearly-sampled signal, but as noted above, eliminating data points reduces the resolution of the resulting spectrum.

For example, the CDC15 data set was collected over 290 minutes. The sampling was non-linear. It is possible to create a linearly sampled data set of 15 time points spaced 1 every 20 minutes. The Nyquist frequency of this reduced set is one cycle per forty minutes and the minimum frequency is one cycle per 290 minutes. Since there are 15 time points in the reduced set, the resolution of the spectrum is one cycle per $(290 - 40)/(15 - 1) = 17.9$ minutes. In other words, two periodic functions must differ by one cycle per 17.9 minutes (or more) in order to be well-resolved (distinguishable) by the Fourier Transform.

[24] used a variant of the Discrete Fourier Transform (eq. 10).

$$X(\omega) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{\frac{-2\pi i \omega n}{N}} \quad (10)$$

[24] notes "...the magnitude of the Fourier transform was unstable for small variations of ω ..." and was forced to average the results of multiple transforms in order to come to a final frequency estimate. The issue was primarily one of resolution in the actual signal. Not surprisingly, the authors resorted to a hybrid approach including non-spectral methods to estimate frequencies. The final frequency and phase estimate was a combination of both spectral and non-spectral methods.

A.2 Autocorrelation

The autocorrelation is truly phase-independent for periodic signals of infinite length. For finite-length signals, the numeric error associated with signals differing only in phase, is proportional to the difference in phase and inversely proportional to the number of periods in the signal. In our experiments on simulated data modeled on the CDC15, CDC28 and fibroblast data sets, that error represents approximately a 1% error in frequency estimation. By contrast, the caculated frequency resolution of the real data sets used in this paper represents between a 15% and 25% error (see Sec. A.1). The noise in the real data sets is also likely to mask the error associated with the autocorrelation.

A.3 Simulated Data Results

A set of 6,500 simulated gene expression profiles were generated. Of those, 1500 were periodic and the remaining ones were aperiodic random signals. The elements of that set are detailed in Table 3 A. The results of running RAGE and RAGE-CC on that data set are reported in Table 3 B. RAGE outperforms RAGE-CC in accuracy of phase and frequency estimation. That is, the average difference between the *actual* periodicity of the synthetic gene and the estimated frequency is smaller when using the Hausdorff metric. RAGE also has a lower rate of false-positives. We conclude that the Hausdorff metric is superior to the correlation coefficient.

Models	Number	% noise
Random	5000	N/A
10 ± 2-hr periodic	500	5%
15 ± 2-hr periodic	500	5%
24 ± 2-hr periodic	500	5%

(A)

	RAGE	RAGE-CC
Number	1498	2199
Mean Δ frequency (hours)	2.1	8.5
Mean Δ phase (hours)	2.2	15.7
False-Positives	6%	33%
False-Negatives	7%	1%

(B)

Table 3: (A) Synthetic Gene Expression Profiles. A set of 6,500 signals were generated. 5,000 of these were random signals. The remaining 1,500 were sinusoids. The sinusoids were grouped into three categories, 10, 15 and 24 -hr periodic of 500 genes each. The actual frequency of each sinusoid was randomly chosen at ± 2-hrs from the category. All genes were assigned random phases. (B) Simulation Results. After running RAGE and RAGE-CC on the simulated gene data, the accuracy of the results were evaluated. Given a sorted list of the confidence scores assigned to each gene, the top 50% of the non-zero scorers were investigated. The number of genes in that set is reported in the first row. Since the actual phase and frequency of the simulated genes were known, it is possible to compare the estimated frequency and phases against their true values. The 2nd and 3rd rows give the mean deviation of the estimated frequency and phase from their true values. *False-positives* indicates what percentage of the identified genes were from the set of random genes. *False-negatives* indicates the percentage of the 1,500 periodic genes were missed. Smaller numbers for false positives and negatives are better.

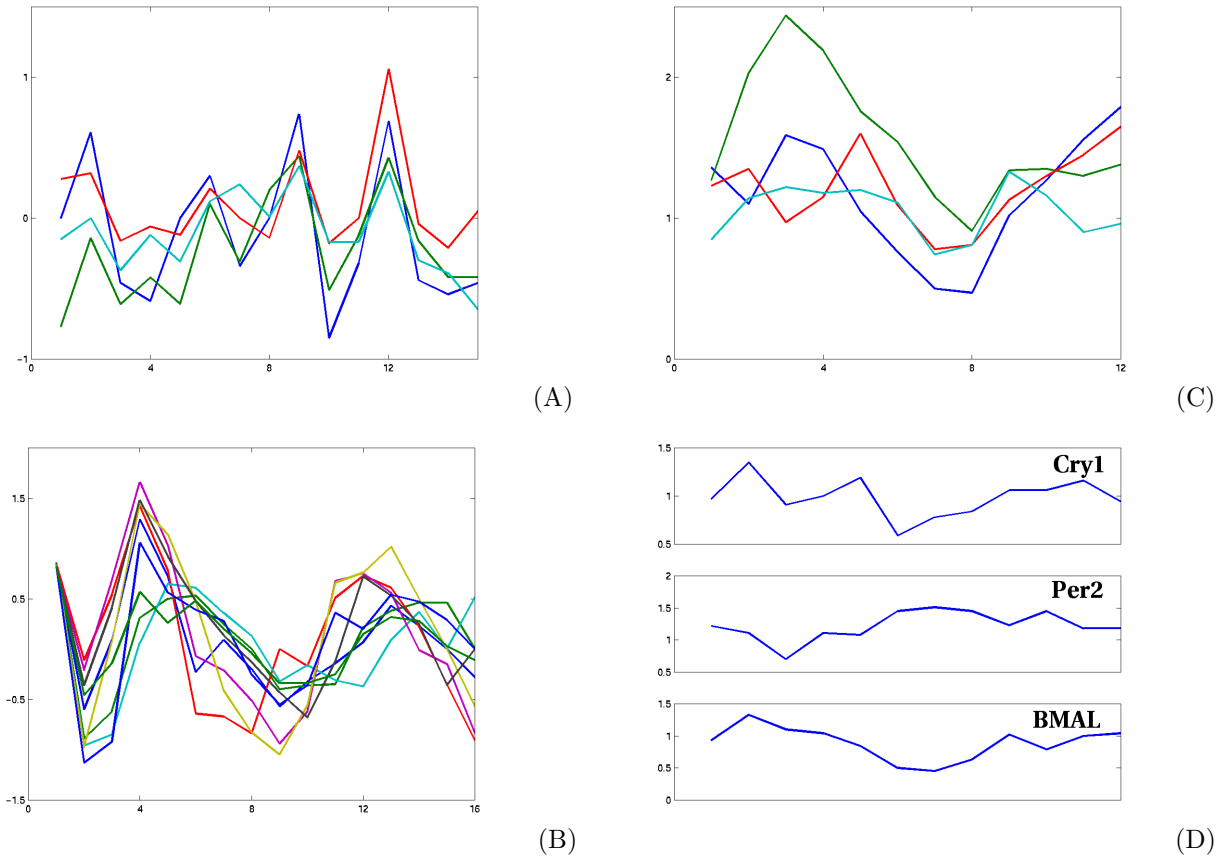

Figure 3: (A-C) Representative clusters from results on actual microarray data. (D) shows Circadian genes identified by RAGE. (A) RAGE identified 561 genes from the CDC15 data set as 90 $\pm$ 10 minute periodic with a confidence score $\geq$ 50%. 4 of these (YOR076C, YCR017C, YKL186C, YLL065W) are shown whose periodicities were judged equal (80 minutes) with roughly equal phases (11 minutes). (B) RAGE identified 748 genes from the CDC28 data set as 90 $\pm$ 10 minute periodic with a confidence score $\geq$ 50%. 9 of these (YKL045W, YNL283C, YNL309W,YJR112W, YFR027W, YKL113C, YML060W, YOR373W, YPR076W) are shown whose periodicities were judged equal (81 minutes) with roughly equal phases (0 minutes). (C) RAGE identified 81 genes from the fibroblast data set as 16 $\pm$ 2 hr periodic with a confidence score $\geq$ 50%. 4 of these (W38444, N70172, W52203, AA044583) are shown whose periodicities were judged equal (15 hours) with roughly equal phases (11 hours). (D) The expression profile of three genes (*Cry1, Per2* and *BMAL*) from the human fibroblast data set. These genes were identified as circadian by RAGE. *Cry1, Per2* and *BMAL* are close human homologs of rat circadian genes identified by [27].