17

Microarrays

Statistical Methods for Circadian Rhythms

Rikuhiro Yamada and Hiroki R. Ueda

Summary

Microarrays are promising tools that are increasingly being applied to the study of circadian rhythms. The large and complex datasets they generate, however, mean they require a new approach on how to design experiments, handle datasets, translate results, and derive conclusions. This technology also requires statistical methods for the correct interpretation of data generated by the microarrays. In this chapter, we provide an overview of analytical methods applied to microarray experiments for the identification of genes with circadian expression.

Key Words: Circadian rhythm; microarray; *p*-value; *fp*-value.

1. Introduction

One of the most remarkable advances in molecular biology over the past decade is the availability of genomic sequence information and the development of high-throughput and genome-based technologies such as microarrays (DNA chips). Microarray studies look at the mRNA expression of tens of thousands of genes and simultaneously measure the fluorescence emitted by hybridized gene-specific probes. One of the main purposes of these analyses is to identify genes with characteristic expression patterns that recapitulate the observed physiology.

One particular aim in applying microarray studies to circadian rhythms is to identify clock-controlled genes that exhibit circadian rhythmicity in their level of expression. The purpose of this chapter is to provide a general overview on the analytical methods used for the identification of clock-controlled genes from the tens of thousands of genes on the microarray. First, the hybridization intensities of multiple microarrays are normalized to balance them appropriately so that meaningful biological comparisons can then be made. The actual circadian rhythmicity is then assessed by calculating correlation coefficients

From: *Methods in Molecular Biology, vol. 362: Circadian Rhythms: Methods and Protocols*
Edited by: E. Rosato © Humana Press Inc., Totowa, NJ

between experimental expression profiles and theoretical cosine waves. Finally, statistical significance and the probability of false positives are evaluated by calculating the *p*-value and *fp*-value, respectively.

In this chapter, we do not intend to give a comprehensive and detailed description of microarray statistical methods available for circadian studies because of their rapid evolvement and because no clear consensus yet exists on which method is best for identifying circadian rhythmicity in gene expression levels. Rather, we will focus on a method based on basic concepts but that is open to further development.

As a prerequisite to reading the chapter, we assume that readers have some experience of spreadsheet applications such as Microsoft Excel, and some knowledge of Mathematica (Wolfram Research, Champaign, IL). It is also advisable to have a basic knowledge of statistical tests ***(1,2)***.

2. Materials

1. Wolfram Research Mathematica (preferably version 5.0 or later).
2. Microsoft Excel (or other spreadsheet software).

3. Methods

In this section, we provide a step-by-step guide to the statistical analysis of microarray data for the identification of genes that exhibit circadian expression. After formatting, the expression data are normalized and then assayed by crosscorrelation with cosine waves cycling with circadian rhythmicity. Finally, *p*-value and *fp*-value are calculated to evaluate statistical significance and the probability of false-positives.

3.1. How Many Chips?

In circadian studies, animals or other organisms are first entrained to a 12 h:12 h light–dark (LD) cycle for days and then are released into free-running constant dark (DD) conditions. RNA is harvested during LD and/or DD cycles, most commonly at 4-h intervals over 48 h ***(3–8)***. Twelve microarrays are therefore generally used for one experiment. Several studies, however, have suggested that it would be preferable to use more arrays over the course of an experiment. Panda et al.***(6)*** used two arrays for each time point over 2 d in DD condition (24 arrays in total), and Claridge-Chang et al. ***(9)*** used three arrays for each time point over 2 d of LD followed by DD condition (36 arrays in total). Using fewer arrays may be more appropriate for more specific purposes where, for example, the effects of mutations or light stimuli are measured ***(10–12)***. Further information regarding this and other studies may be found in **Table 1** ***(3–22)*** and in some excellent reviews ***(23,24)***. In this chapter, we assume that only one array has been used for each of the 12 datapoints over 2 d.

3.2. Preparation of Data Files

All expression data should be placed into a table consisting of number of probes (rows) × number of chips (columns) **(Table 2)**. This can be performed through basic manipulation in a spreadsheet application such as Microsoft Excel. The subsequent data table should be saved as a tab-separated text file. Here, we save this file as C:\work\data.txt.

3.3. Normalization

In spite of great care in keeping experimental conditions constant, random effects are unavoidable. In circadian research we usually use multiple chips (12 chips in our case) to measure temporal changes of mRNA expression. As stochastic variability is inevitable, proper mathematical procedures must be implemented to allow for cross-chip comparisons. "Normalization" is a term used to describe processes that reduce the impact of random effects on the data, with many methods having been proposed ***(25,26)***. In this section we adopt the following: we scale the average expression level on each chip so as to be equal among all chips, as we assume that all chips have been stained with roughly the equal amount of total mRNA. Another popular technique is to scale the expression levels so as to have equal medians for all the chips. Although more sophisticated techniques are now currently available ***(25)***, normalization of the average or the median are still first-choice methods **(Fig. 1)**. In the following subheading, we describe the program codes for Mathematica to perform these normalization steps.

3.3.1. Load Packages and Expression Profile Data

1. Before starting, load the Mathematica packages required for the subsequent analyses.

```
Needs["Statistics'MultiDescriptiveStatistics'"]
Needs["Statistics'ContinuousDistributions'"]
```

2. Load the previously prepared raw expression data, using the following Mathematica code:

```
dataTable =
ReadList["C:\\work\\data.txt",{Word, Number, Number,
Number, Number, Number, Number, Number, Number, Number,
Number, Number, Number}];
```

   Now the variable "dataTable" is a table (two-dimensional matrix), whose rows represents genes, and whose columns represents probe IDs (column 1) and expression profiles (column 2 to column 13).

3. Separate probe IDs from expression profiles using the following code:

```
idList=Transpose[dataTable][[1]];
rawExpressionTable=Transpose[Drop[Transpose
[dataTable],{1}]];
```

Table 1
Summary of Microarray Studies on Circadian Rhythms

Authors	Year	Sample	DNA Chip	Design	Analysis method
Harmer et al. ***(8)***	2000	*Arabidopsis*	HDO	12 time-points, 4-h interval, LL, $n = 2$	Cross correlation with cosine waves
Schaffer et al. ***(19)***	2001	*Arabidopsis*	cDNA	4 time-points, 6-h interval, LD, $n = 1–4$ 1 time-point, DD, $n = 2$ 2 time-points, LL, $n = 1$	Two time-point comparison
Claridge-Chang et al. ***(9)***	2001	*Drosophila* head	HDO	12 time-points, 4-h interval, LD followed by DD, $n = 3$	Fourier analysis
McDonald and Rosbach ***(20)***	2001	*Drosophila* head	HDO	6 time-points, 4-h interval, DD, $n = 3–5$	Cross correlation with cosine waves
Grundschober et al. ***(21)***	2001	Rat-1 fibroblasts	HDO	20 time-points, 4-h interval, DD, $n = 1$	Spectral analysis
Kit et al. ***(14)***	2002	Rat liver Rat kidney	cDNA	2 time-points, 12-h interval, LD, $n = 1$	Two time-point comparison
Humphries et al. ***(15)***	2002	Rat pineal gland	cDNA	2 time-points, 12-h interval, LD, $n = 3$	Two time-point comparison
Akhtar et al. ***(18)***	2002	Mouse liver Mouse hypothalamus	cDNA	7 time-points, 4-h interval, DD, $n = 2$	Anchored comparison Moving window analysis
Duffield et al. ***(17)***	2002	Rat-1 fibroblasts	cDNA	13 time-points, 4-h interval, DD, $n = 1$	Cosine wave fitting
Ueda et al. ***(4)***	2002	*Drosophila* head	HDO	12 time-points, 4-h interval, LD and DD, $n = 1$	Cross correlation with cosine waves
Storch et al. ***(5)***	2002	Mouse heart Mouse liver	HD)	12 time-points, 4-h interval, DD, $n = 1$	Autocorrelation analysis
Panda et al. ***(6)***	2002	Mouse SCN Mouse liver	HDO	12 time-points, 4-h interval, DD, $n = 2$	Cosine wave fitting
Lin et al. ***(13)***	2002	*Drosophila* head	HDO	6 time-points, 4-h interval, LD, $n = 2–3$, and DD, $n = 2$	Autocorrelation analysis
Ueda et al. ***(3)***	2002	Mouse SCN Mouse liver	HDO	12 time-points, 4-h interval, LD and DD, $n = 1$	Cross correlation with cosine waves

Ceriani et al. ***(7)***	2002	*Drosophila* head	HDO	12 time-points, 4-h interval, LD and DD, $n = 2$	Cosine wave fitting
Hirota et al. ***(16)***	2002	Rat-1 fibroblasts	HDO	3 time-points, 0 h, 1 h, 4 h, $n = 1$	Time-point comparison
Nowrousian et al. ***(22)***	2003	*Neurospora*	cDNA	5 time-points, 4-h interval, 1 cycle, DD, $n = 3$ and temprature entrainment	Time-point comparison Cosine wave fitting
Oishi et al. ***(10)***	2003	Mouse liver	HDO	2 time-points, 12-h interval, 1 cycle, DD, $n = 1$	Two time-point comparison
Salter et al. ***(12)***	2003	*Arabidopsis*	HDO	7 time-points, $n = 1$	Time-point comparison
Grechez-Cassiau et al. ***(11)***	2004	Mouse liver	HDO	2 time-points, 12-h interval, 1 cycle, DD, $n = 2–3$	Two time-point comparison

Studies are listed by publication date.

HDO, high-density oligonucleotide microarray; cDNA, complementary DNA microarray; SCN, suprachlasmatic nucleus; LD, light–dark; DD, constant darkness.

Table 2
Profiles of Gene Expression Over 2 d at 4-h Intervals

1415670_at	313.6	332.7	313.1	425	599.7	463.8	429.2	324.6	554.4	461.2	575.6	349.5
1415671_at	680.4	799	805.5	1019.7	1031.7	1008.5	1006.5	707.5	756.8	1123.4	1195.1	675
1415672_at	1281.6	1484.1	872.7	1058.8	1184	1084	1227.2	931.4	1059.4	1214.8	1203	764.2
1415673_at	124.3	95.3	80.4	110.3	132.9	112	103.9	58	64.9	108.1	101.5	65.4
1415674_a_at	307.4	335.8	312.1	376.6	350.2	340.7	394.6	289	284.8	385	375.8	245.7
1415675_at	258.8	229.4	231.9	282.3	245.4	271.7	315.5	228.3	167.5	227.4	242.4	170.9
1415676_a_at	1094.3	1415.7	1330.2	1327.6	1242.9	1221.8	1722.2	1248.4	1092.6	1446.3	1311.5	1173.6
1415677_at	441.3	480.6	557.8	737.4	434.2	523.2	789.9	635	372.1	850.9	524.1	625
1415678_at	828.6	930.7	884	967.3	950	818.9	749.2	687	685	984.4	792.3	570.2
1415679_at	1274.7	1409.7	1202	1358.2	1286.6	1249.3	1500.2	993.2	1185.1	1428.4	1565.9	958.2

This table is created with Microsoft Excel. The first column shows the "Affymetrix Probe Set Ids" and the following columns indicate the expression level for each gene. The first 10 out of 22,690 rows are shown here. There is no header row to simplify Mathematica codes.

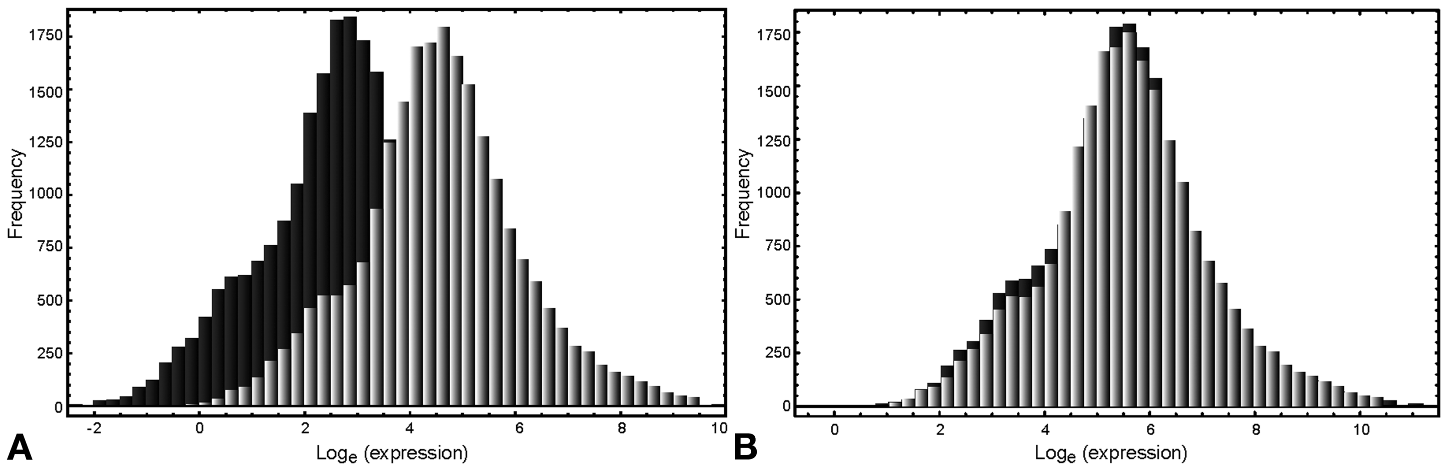

Fig. 1. Schematic representation of the normalization procedure. Gene expression data from two different chips are shown before **(A)** and after **(B)** normalization to illustrate how these procedures transform the data sets. The normalized distributions, shown in (B), are shifted and aligned at their centers. Gene expression comparisons between the two distributions can now be made without systematic experimental bias.

The first code exchanges rows and columns of dataTable, and then extracts the first column (probe IDs). The second code exchanges rows and columns of dataTable, and then drops the first column (probe IDs), and exchanges its rows and columns again. The produced "idList" is an array of probe IDs, and "rawExpressionTable" is a table (two-dimensional matrix), whose rows represent genes, and whose columns represent expression profiles.

3.3.2. Equalize Average or Median of Each Chip

1. Scale the level of expression of each probe so that the average expression level for each chip becomes 1000 (*see* **Note 1**) using the following Mathematica code:

```
normalizationFactors=1000/Mean[rawExpressionTable];
normalizedExpressionTable=rawExpressionTable.Diagonal
Matrix[normalizationFactors];
```

The first line calculates scaling factors and put them in a vector. The second line multiplies "rawExpressionTable" with a diagonal matrix of the scaling factors to produce normalized expression profiles "normalizedExpressionTable," whose rows contains normalized expression profiles of each gene.
Alternatively, scale the expression levels for each probe so that the median of each chip becomes 1000 (*see* **Note 1**), using the following Mathematica code:

```
normalizationFactors=1000/Median[rawExpressionTable];
normalizedExpressionTable=rawExpressionTable.Diagonal
Matrix[normalizationFactors];
```

3.4. Evaluation of Circadian Expression

Several procedures exist by which to evaluate whether the expression of a gene is under circadian control. One of them is based on the assumption that the expression profile of a gene exhibiting circadian rhythmicity approximates a cosine wave with a period of 24 h (*see* **Note 2**). A significant correlation can therefore be found between a rhythmically expressed gene and a theoretical cosine wave cycling with an appropriate phase, as can be seen in the following:

1. Generate 60 cosine waves with the equation defined below (*see also* **Fig. 2**).

$$C_i = \cos(2\pi(\frac{1}{24}t - \frac{1}{60}i)) \; (t = 0,4,8,...,44) \; (i = 0,1,2,...59)$$

The following properties apply:

- 24-h period.
- 48 h long (two cycles).
- Interval between adjacent phases equal to 0.4 h.

The above formula is expressed in Mathematica as the following:

```
cosines=Table[Cos[2Pi(t/24-i/60)],{i,0,59,1},{t,0,44,4}];
```

2. Calculate the correlation coefficient between each expression profile and each of the 60 cosine waves (C_i). The highest correlation coefficient among them should

be selected as the representative value of circadian rhythmicity. We have termed this value *max correlation* (*maxCorr*). For a gene *k*, $maxCorr_k$ is defined as follows:

$$maxCorr_k = \max(Correlation(expression\_profile_k, C_i))\ (\mathrm{i} = 0,1,2,...59).$$

A list of *maxCorr*s can be calculated by the following Mathematica code:

```
maxCorrs = {};
peakTimes = {};
For[g = 1, g <= Length[normalizedExpressionTable], g++,
  normalizedExpression = normalizedExpressionTable[[g]];
  corrs = Table[Correlation[normalizedExpression,
  cosines[[i]]]
  ,{i, 1, Length[cosines]}];
  maxCorr = Max[corrs];
  peakTime = 0.4*(Position[corrs, maxCorr][[1, 1]] - 1);
  AppendTo[maxCorrs, maxCorr];
  AppendTo[peakTimes, peakTime];
];
```

Note that "peakTime" indicates the estimated *peak time* of normalized expression data, which is estimated by the peak time of the best-correlated cosine curve.

3.5. Statistical Significance: p *Value*

In this context, the *p* value can be defined as the probability that a random expression profile shows *max correlation* greater than a given value **(Fig. 3)**. The *p* value is experimentally calculated by generating random expression profiles, calculating *maxCorr* for each random profile, and finally, counting the frequency of a random expression profile showing *maxCorr* greater than the chosen value. When a *maxCorr* value increases, the associated *p* value decreases; in other words, a smaller p value indicates that random expression profiles are more unlikely to show a defined maxCorr.

1. Generate 100,000 random expression profiles and calculate *maxCorr* for each of them, thereby creating the *maxCorr* distribution of random expression profiles. The Mathematica code to obtain this distribution is as follows:

```
randomCorrs={};
For[i=1,i<= 100000,i++,
  randomExpression=Table[Random[NormalDistribution[0,1]],
  {t,1,12}];
  corrs=Table[Correlation[randomExpression,cosines[[i]]]
  ,{i,1,Length[cosines]}];
  randomCorr = Max[corrs];
  AppendTo[randomCorrs,randomCorr];
];
```

This process usually takes several hours on an up-to-date PC (such as Pentium4 3GHz; *see* **Note 3**).

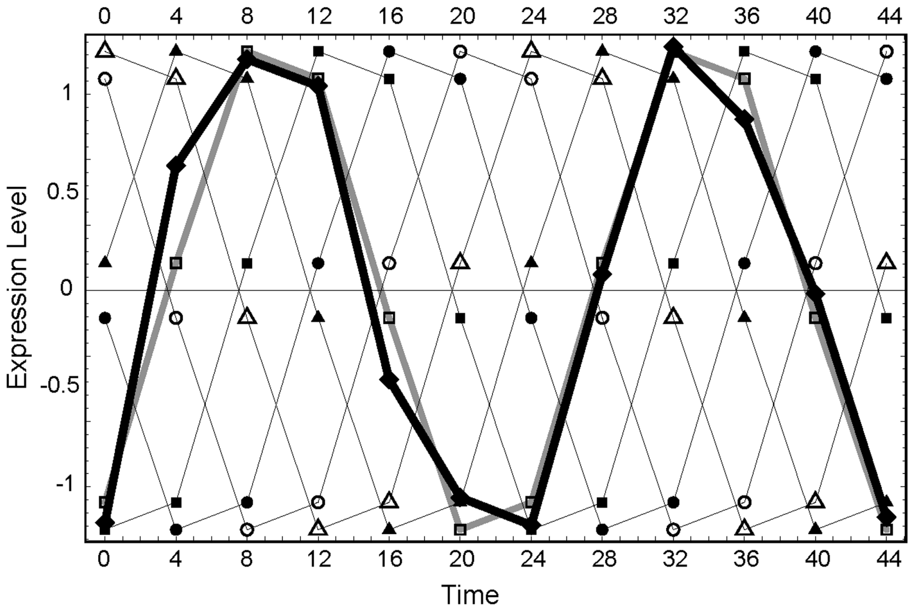

Fig. 2. Crosscorrelation between the expression profile of a gene and theoretical cosine waves. The experimental profile (black line) is compared with 60 cosine waves of fixed periodicity (e.g., 24 h), varying in phase from 0 to 24 h. A gray line indicates the best-correlated cosine wave. For convenience, only 6 out of the 60 cosine waves are shown here.

2. Save the result in a file:

```
Save["C:\\work\\randomCorrs.txt",randomCorrs];
```

3. Using the following Mathematica code it is possible to reload the random correlation data at any time, even after restarting Mathematica:

```
<<"C:\\work\\randomCorrs.txt";
```

4. Count the number of times that a random expression profile shows *maxCorr*s greater than a specified value. The following Mathematica code can be used to perform this (*see* **Note 4**):

```
CountGreater[maxCorr_,value_]:=Length[
Select[randomCorrs,(# > value)&]]
```

5. Calculate the *p* value with the following code:

```
sortRandomCorrs=Sort[randomCorrs];
pValues=Table[CountGreater[sortRandomCorrs,
maxCorrs[[i]]]/Length[sortRandomCorrs],{i,1,
Length[maxCorrs]}];
```

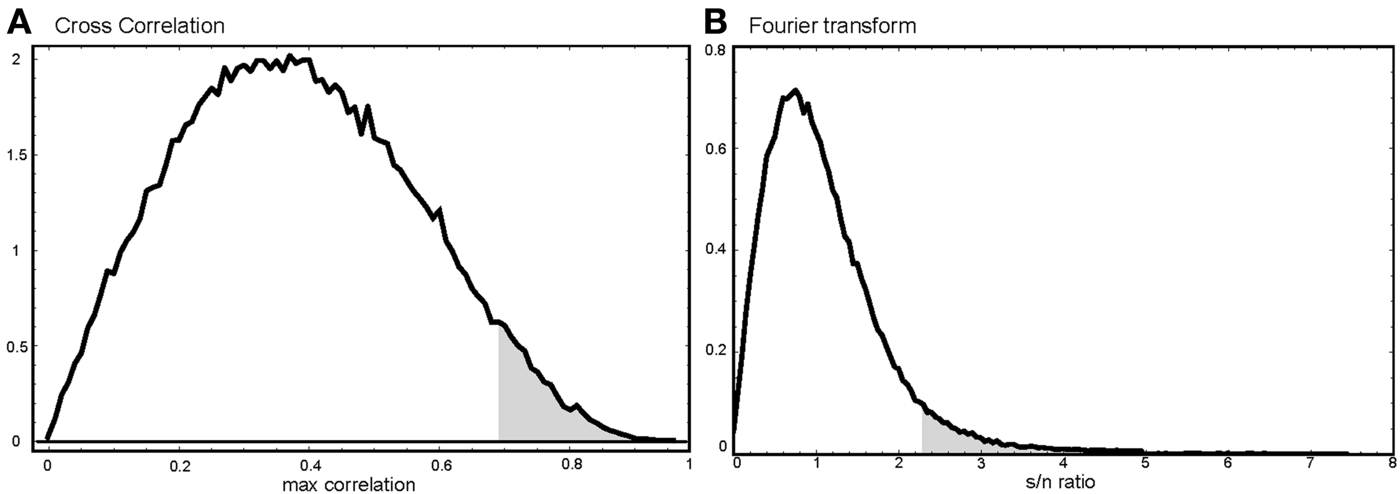

Fig. 3. *maxCorr* and *s/n ratio* distributions calculated from 100,000 random expression profiles. The curves indicate the probability of obtaining a particular value of *maxCorr* (**A**) or *s/n ratio* (**B**), when 100,000 random expression profiles are generated. The shaded area, compared with the total under the probability curve, indicates the p value associated with a particular value of *maxCorr* (= 0.7, A) or *s/n ratio* (= 2.3, B). In this graph the p values are about 0.05.

3.6. Probability of False-Positives: fp Value

500 "positive" genes out of 10,000 genes may be obtained by chance, when you call genes with $p \leq 0.05$ as "positive." This means, assuming that the relevant research involves the expression of 10,000 genes on a microarray and that 1000 positive genes ($p \leq 0.05$) are obtained, 500 genes are expected to be false positives among the 1000 positive genes. The probability of false-positives will therefore be 500/1000 = 0.5 (*see* **Notes 5** and **6**).

1. This probability is defined as the *fp* value **(Fig. 4)** and is calculated by the following Mathematica code:

```
sortMaxCorrs=Sort[maxCorrs];
sortRandomCorrs=Sort[randomCorrs];
randomCorrsSize=Length[randomCorrs];
maxCorrsSize=Length[maxCorrs];
fpValues=Table[
  (CountGreater[sortRandomCorrs,maxCorrs[[i]]]/
  randomCorrsSize)
*(maxCorrsSize/CountGreater[sortMaxCorrs,maxCorrs[[i]]])
,{i,1,maxCorrsSize}];
```

2. As the *fp* values are calculated experimentally and not theoretically, they may not exhibit monotonous decreasing that parallels smaller *p* values. To correct for this, we use an additional process shown below.

```
pv=pValues;
fp=fpValues;
idxPV=Transpose[{Range[Length[pv]],pv}];
sortedIdxPV=Sort[idxPV,(#1[[2]]>#2[[2]])&];
minFP=1;
idxSmoothedFP=Table[
  idx=sortedIdxPV[[i]][[1]];
  {idx,minFP=Min[fp[[idx]],minFP]},
  {i,Length[sortedIdxPV]}
  ];
idxSmoothedFP=Sort[idxSmoothedFP,(#1[[1]] < #2[[1]])&];
fpValuesSM=Transpose[idxSmoothedFP][[2]];
```

3.7. Fourier Analysis

The previous section stated that the identification of rhythmically expressed genes is based on the maximum correlation coefficients between expression profiles and cosine waves. In this section, we describe an alternative approach by Fourier analysis. The Fourier transform decomposes an expression profile into a linear combination of sinusoids of different periods, and circadian rhythmicity is measured by comparing the amplitude of the 24-h period sinusoid with that of other period sinusoids. In this section, we describe these procedures. Codes described in **Subheadings 3.1.1.** and **3.1.2.** and the function

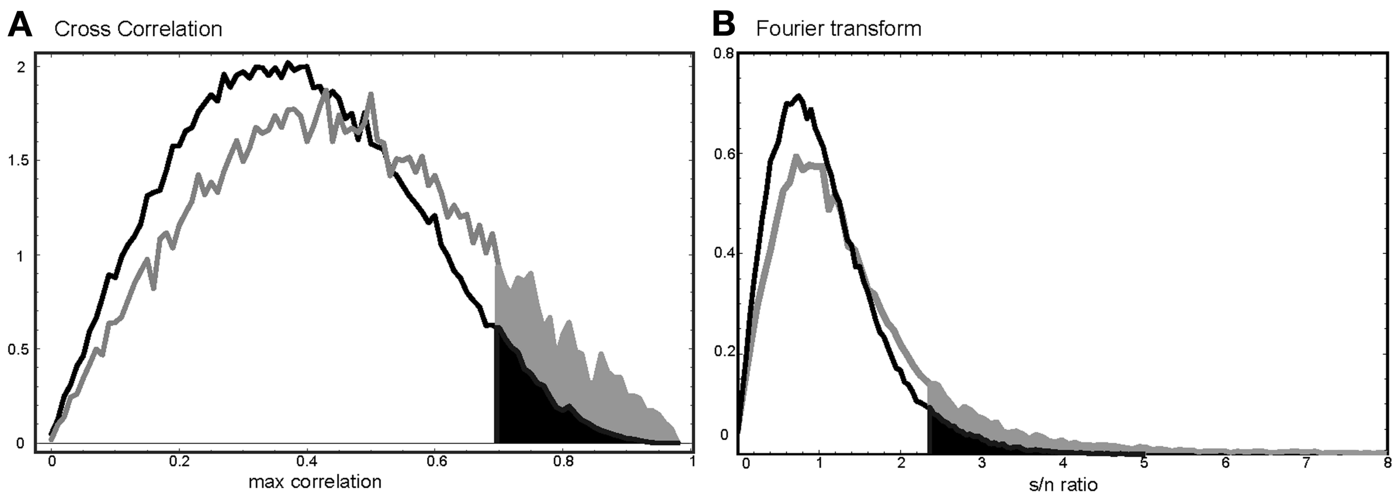

Fig. 4. Distributions of *maxCorr* and *s/n ratio* calculated from real and random expression profiles. The black curve indicates the distribution of *maxCorr* (**A**) or *s/n ratio* (**B**) from random expression profiles whereas the gray curve refers to the real expression profiles. The proportion between the black area and the total of the black and gray areas defines the *fp* value.

"CountGreater" are also used here and should be executed before implementation of the codes described in this section.

3.7.1. Discrete Fourier Transform

1. Before applying the Fourier transform, subtract the average expression level from the expression level of each gene.

```
subtractedExpressionTable =
Table[normalizedExpressionTable[[i]]-Mean[normalized
  ExpressionTable[[i]]],{i,1,Length[normalized
  ExpressionTable]}];
```

2. Mathematica can perform Fourier transform with just one function.

```
fourierTable=Table[Fourier[subtractedExpression
  Table[[i]]],
{i,1,Length[subtractedExpressionTable]}];
```

3. From a 12-sample-points time course, derive the amplitude of the 24-h period sinusoid from the third component of each "fourierTable[[i]]" (i = 1,2,…,n) and calculate the *signal/noise* (*s/n*) *ratio* (**Fig. 5**) with the following Mathematica code:

```
snRatios=Table[
  amplitudes=Abs[fourierTable[[i]]];
  amplitudes[[3]]/Mean[Part[amplitudes,{1,2,4,5,6,7}]]
  ,{i,1,Length[fourierTable]}
];
```

4. *PeakTime* can also be calculated from the third component of each "fourierTable[[i]]" (i = 1,2,…,n) by calculating its argument.

```
frPeakTimes=Table[
  components=fourierTable[[i]];
  Mod[24*Arg[components[[3]]]/(2Pi),24],
{i,1,Length[fourierTable]}
];
```

3.7.2. Statistical Significance

Using a similar approach as for the correlation coefficients, statistical significance and the probability of false-positives for *s/n* ratios can be assessed by calculating *p*- and *fp* values, respectively.

1. Generate random expression profiles and create a distribution of *s/n* ratios as follows:

```
randomSNRatios=Table[
  randomExpression=Table[Random[NormalDistribution
  [0,1]],{12}];
  randomFourierAbs=Abs[Fourier[randomExpression]];
  randomFourierAbs[[3]]/Mean[Part[randomFourierAbs,
  {1,2,4,5,6,7}]]
,{100000}];
```

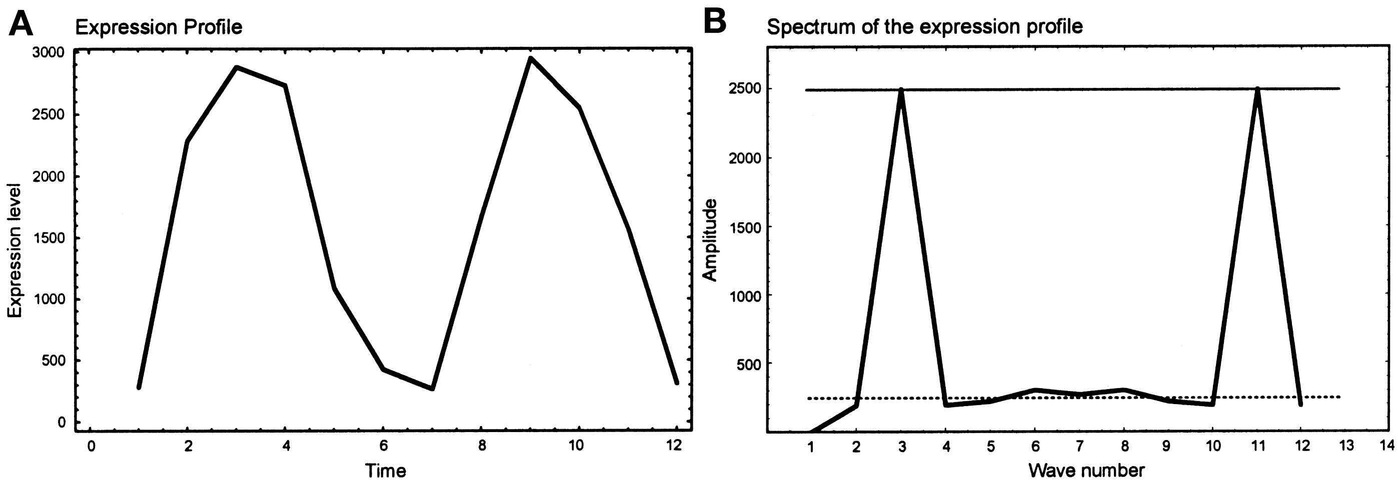

Fig. 5. (**A**) Expression profile of a gene and (**B**) its spectrum generated after Fourier transform. The solid line represents the amplitude of the 24-h-period sinusoid, whereas the dotted line represents the average amplitude of other period sinusoids. The *s/n ratio* is defined as the ratio of these two values (value of solid line/value of dotted line).

2. Use this distribution to calculate *p* values **(Fig. 3B)**:

```
sortedRandomSNRatios=Sort[randomSNRatios];
snRatioSize=Length[snRatios];
randomSNRatioSize=Length[randomSNRatios];
frPValues=Table[CountGreater[sortedRandomSNRatios
  snRatios[[i]]]/randomSNRatioSize,{i,1,snRatioSize}];
```

3. Use it also to calculate *fp* value **(Fig. 4B)**:

```
snRatioSize=Length[snRatios];
randomSNRatioSize=Length[randomSNRatios];
sortedSNRatios=Sort[snRatios];
sortedRandomSNRatios = Sort[randomSNRatios];
frFPValues=Table[
(CountGreater[sortedRandomSNRatios,snRatios[[i]]]/
  randomSNRatioSize
*(snRatioSize/CountGreater[sortedSNRatios,snRatios[[i]]])
,{i,1,Length[snRatios]}];
```

4. Use the following code to ensure that the *fp* values are monotonously decreasing along with the *p* values:

```
pv=frPValues;
fp=frFPValues;
idxPV=Transpose[{Range[Length[pv]],pv}];
sortedIdxPV=Sort[idxPV,(#1[[2]]>#2[[2]])&];
minFP=1;
idxSmoothedFP=Table[
  idx=sortedIdxPV[[i]][[1]];
  {idx,minFP=Min[fp[[idx]],minFP]},
  {i,Length[sortedIdxPV]}
  ];
idxSmoothedFP=Sort[idxSmoothedFP,(#1[[1]] < #2[[1]])&];
frFPValuesSM=Transpose[idxSmoothedFP][[2]];
```

5. Save the result in a file also containing additional statistics.

```
tableForFile=Table[
  {idList[[i]],
  N[avgs[[i]]],
  N[sdvs[[i]]],
  N[frPeakTimes[[i]]],
  snRatios[[i]],
  N[frPValues[[i]]],
  N[frFPValuesSM[[i]]]},{i,1,Length[idList]}];
Export["C:\\work\\output_fr.txt",tableForFile,"TSV"];
```

3.8. Other Statistics

So far, we have demonstrated how to calculate the *p* value and *fp* value for the expression profile of each gene, which provides enough information to determine whether a gene exhibits rhythmic expression or not. This section describes how to calculate other useful statistics.

3.8.1. Average Expression

The average expression of a gene over a time course provides useful information for evaluating the general level of expression in the samples.

```
avgs=Mean[Transpose[normalizedExpressionTable]];
```

Generally, the expression profile of genes with a low average expression is unreliable, as experimental noise can obscure the true signal.

3.8.2. Standard Deviation of Expression

In this context, standard deviation refers to the size of the variation in the expression profile of a transcript; it is an estimate of the amplitude of expression of cycling genes.

```
sdvs=StandardDeviation[Transpose[normalizedExpression
  Table]];
```

3.8.3. Average Peak Time

If you have two sets of samples—for instance, one under LD and the other under DD conditions—their average peak time might be useful. Defining "peakTimeLD" and "peakTimeDD" as peak time in LD and DD respectively, calculate the average peak time with the following code (*see* **Note 7**):

```
peakTimeAvgs=Table[(Mod[(peakTimeLD[[i]]+0.5*Mod
  [peakTimeDD[[i]] -peakTimeLD[[i]],24,-12]),24])&,
  {i,1,Length[peakTimeLD]}];
```

3.9. Write Results to File

Entering analyzed data into a file that can be viewed and edited by spreadsheet software such as Microsoft Excel is useful for further analysis. In Mathematica, a tab-separated file in which each line consists of "id list," "average of expression," "standard deviation of expression," "peak time," "max correlation," "*p* value," and "*fp* value" can be written with the following code (*see* **Note 8**):

```
tableForFile=Table[
  {idList[[i]],
  N[avgs[[i]]],
  N[sdvs[[i]]],
  N[peakTimes[[i]]],
  maxCorrs[[i]],
  N[pValues[[i]]],
  N[fpValuesSM[[i]]]},{i,1,Length[idList]}];
Export["C:\\work\\output.txt",tableForFile,"TSV"];
```

4. Notes

1. The number 1000 is arbitrary; it is possible to select a different number.
2. In other studies *(3,4)* different periods (20–28 h) are used to identify genes with a circadian expression pattern. The basic strategy described in **Subheading 3.4.** still applies and can be expanded to longer or shorter periods (e.g., 20–28 h).
3. A much faster Mathematica code to obtain the *maxCorr* distribution of random expression profiles is as follows:

```
randomExpressionTable=Table[Random[NormalDistribution
   [0,1]],{100000},{12}];
sinBase=Table[Sin[2*Pi(i/24)],{i,0,44,4}];
sinBase=sinBase / Sqrt[sinBase.sinBase];
cosBase=Table[Cos[2*Pi(i/24)],{i,0,44,4}];
cosBase=cosBase/ Sqrt[cosBase.cosBase];
f={cosBase,sinBase}.((#-Mean[#])/Sqrt[(#-Mean[#])
   (#-Mean[#])]) & /@randomExpressionTable;
randomCorrs=Sqrt[#.#]& /@ f;
```

   This code uses the concept of Fourier transformation. Although this code runs much faster, it is advisable to save the result in a file.
4. A faster alternative to the previous code is the following:

```
CountGreater=Compile[{{l,_Real,1},{val,_Real}},
   ei=len=Length[l];              (* end index *)
   si=1;                          (* start index *)
   mi=Floor[si+ei/2];             (* middle index *)
   If[val>l[[len]],Return[0]];
   If[val<l[[1]],Return[len]];
   While[mi != si,
      If[l[[mi]]>val,
         ei=mi;
         mi=Floor[(si+ei)/2];
         ,
         si=mi;
         mi=Floor[(si+ei)/2];
         ];
      ];
   len-si
];
```

5. In our analyses we empirically use an *fp* value of 0.1, corresponding to 10% of false positives, as a threshold. You may increase this value to increase the sensitivity of identification, or may decrease it to increase the specificity of identification.
6. *fp* value is a conservative form of false discovery rate . Storey and Tibshirani have proposed a statistic known as *q* value *(28)* that corrects the tendency of the *fp* value to overestimate false positives. You can easily calculate the *q* values for your data set by feeding your list of *p* values assigned to each gene to the software made available by Storey et al. at their website (http://faculty.washington.edu/~jstorey/qvalue/).

7. A simple arithmetic average is inappropriate for calculating the average peak time. For example the average time between 23:00 and 1:00 should be 0:00, not 12:00.
8. In this example the data are recorded into an "output.txt" file. You can easily add annotation information to this file. Affymetrix provides annotation information for each target gene on their microarrays found on its website *(27)*.

Acknowledgments

We thank Michael Royle and Douglas Sipp at SCIA (Office for Science Communications and International Affairs) of CDB for carefully going over the draft and pointing out many errors and helping us improve the manuscript significantly.

References

1. Curran-Everett, D., Taylor, S., and Kafadar, K. (1998) Fundamental concepts in statistics: elucidation and illustration. *J. Appl. Physiol.* **85,** 775–786.
2. Curran-Everett, D. (2000) Multiple comparisons: philosophies and illustrations. *Am. J. Physiol. Regul. Integr. Comp. Physiol.* **279,** R1–R8.
3. Ueda, H. R., Chen, W., Adachi, A., et al. (2002) A transcription factor response element for gene expression during circadian night. *Nature* **418,** 534–539.
4. Ueda, H. R., Matsumoto, A., Kawamura, M., Iino, M., Tanimura, T., and Hashimoto, S. (2002) Genome-wide transcriptional orchestration of circadian rhythms in *Drosophila. J. Biol. Chem.* **277,** 14,048–14,052.
5. Storch, K. F., Lipan, O., Leykin, I., et al. (2002) Extensive and divergent circadian gene expression in liver and heart. *Nature* **417,** 78–83.
6. Panda, S., Antoch, M. P., Miller, B. H., et al. (2002) Coordinated transcription of key pathways in the mouse by the circadian clock. *Cell* **109,** 307–320.
7. Ceriani, M. F., Hogenesch, J. B., Yanovsky, M., Panda, S., Straume, M., and Kay, S. A. (2002) Genome-wide expression analysis in Drosophila reveals genes controlling circadian behavior. *J. Neurosci.* **22,** 9305–9319.
8. Harmer, S. L., Hogenesch, J. B., Straume, M., et al. (2000) Orchestrated transcription of key pathways in *Arabidopsis* by the circadian clock. *Science* **290,** 2110–2113.
9. Claridge-Chang, A., Wijnen, H., Naef, F., Boothroyd, C., Rajewsky, N., and Young, M. W. (2001) Circadian regulation of gene expression systems in the *Drosophila* head. *Neuron* **32,** 657–671.
10. Oishi, K., Miyazaki, K., Kadota, K., et al. (2003) Genome-wide expression analysis of mouse liver reveals CLOCK-regulated circadian output genes. *J. Biol. Chem.* **278,** 41,519–41,527.
11. Grechez-Cassiau, A., Panda, S., Lacoche, S., et al. (2004) The transcriptional repressor STRA13 regulates a subset of peripheral circadian outputs. *J. Biol. Chem.* **279,** 1141–1150.
12. Salter, M. G., Franklin, K. A., and Whitelam, G. C. (2003) Gating of the rapid shade-avoidance response by the circadian clock in plants. *Nature* **426,** 680–683.

13. Lin, Y., Han, M., Shimada, B., et al. (2002) Influence of the period-dependent circadian clock on diurnal, circadian, and aperiodic gene expression in *Drosophila melanogaster. Proc. Natl. Acad. Sci. USA* **99,** 9562–9567.
14. Kita, Y., Shiozawa, M., Jin, W., et al. (2002) Implications of circadian gene expression in kidney, liver and the effects of fasting on pharmacogenomic studies. *Pharmacogenetics* **12,** 55–65.
15. Humphries, A., Klein, D., Baler, R., and Carter, D. A. (2002) cDNA array analysis of pineal gene expression reveals circadian rhythmicity of the dominant negative helix-loop-helix protein-encoding gene, Id-1. *J. Neuroendocrinol.* **14,** 101–108.
16. Hirota, T., Okano, T., Kokame, K., Shirotani-Ikejima, H., Miyata, T., and Fukada, Y. (2002) Glucose down-regulates Per1 and Per2 mRNA levels and induces circadian gene expression in cultured Rat-1 fibroblasts. *J. Biol. Chem.* **277,** 44,244–44,251.
17. Duffield, G. E., Best, J. D., Meurers, B. H., Bittner, A., Loros, J. J., and Dunlap, J. C. (2002) Circadian programs of transcriptional activation, signaling, and protein turnover revealed by microarray analysis of mammalian cells. *Curr. Biol.* **12,** 551–557.
18. Akhtar, R. A., Reddy, A. B., Maywood, E. S., et al. (2002) Circadian cycling of the mouse liver transcriptome, as revealed by cDNA microarray, is driven by the suprachiasmatic nucleus. *Curr. Biol.* **12,** 540–550.
19. Schaffer, R., Landgraf, J., Accerbi, M., Simon, V., Larson, M., and Wisman, E. (2001) Microarray analysis of diurnal and circadian-regulated genes in *Arabidopsis. Plant Cell* **13,** 113–123.
20. McDonald, M. J., and Rosbash, M. (2001) Microarray analysis and organization of circadian gene expression in *Drosophila. Cell* **107,** 567–578.
21. Grundschober, C., Delaunay, F., Puhlhofer, A., et al. (2001) Circadian regulation of diverse gene products revealed by mRNA expression profiling of synchronized fibroblasts. *J. Biol. Chem.* **276,** 46,751–46,758.
22. Nowrousian, M., Duffield, G. E., Loros, J. J., and Dunlap, J. C. (2003) The frequency gene is required for temperature-dependent regulation of many clock-controlled genes in *Neurospora crassa. Genetics*,**164,** 923–933.
23. Duffield, G. E. (2003) DNA microarray analyses of circadian timing: the genomic basis of biological time. *J. Neuroendocrinol.* **15,** 991–1002.
24. Etter, P. D., and Ramaswami, M. (2002) The ups and downs of daily life: profiling circadian gene expression in *Drosophila. Bioessays* **24,** 494–498.
25. Bolstad, B. M., Irizarry, R. A., Astrand, M., and Speed, T. P. (2003) A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics* **19,** 185–193.
26. Quackenbush, J. (2002) Microarray data normalization and transformation. *Nat. Genet.* **32 Suppl,** 496–501.
27. Liu, G., Loraine, A.E., Shigeta, R., et al. (2003) NetAffx: Affymetrix probesets and annotations. *Nucleic Acids Res.* **31,** 82–86.
28. Storey, J. D., and Tibshirani, R. (2003) Statistical significance for genomewide studies. *Proc. Natl. Acad. Sci. USA* **100,** 9440–9445.