PLOS ONE

RESEARCH ARTICLE

Confidence in Phase Definition for Periodicity in Genes Expression Time Series

Mohammed El Anbari*, Abeer Fadda, Andrey Ptitsyn

Division of Biomedical Informatics, Sidra Medical and Research Center, Doha, Qatar

* melanbari@sidra.org

Abstract

Circadian oscillation in baseline gene expression plays an important role in the regulation of multiple cellular processes. Most of the knowledge of circadian gene expression is based on studies measuring gene expression over time. Our ability to dissect molecular events in time is determined by the sampling frequency of such experiments. However, the real peaks of gene activity can be at any time on or between the time points at which samples are collected. Thus, some genes with a peak activity near the observation point have their phase of oscillation detected with better precision then those which peak between observation time points. Separating genes for which we can confidently identify peak activity from ambiguous genes can improve the analysis of time series gene expression. In this study we propose a new statistical method to quantify the phase confidence of circadian genes. The numerical performance of the proposed method has been tested using three real gene expression data sets.

Citation: El Anbari M, Fadda A, Ptitsyn A (2015) Confidence in Phase Definition for Periodicity in Genes Expression Time Series. PLoS ONE 10(7): e0131111. doi:10.1371/journal.pone.0131111

Editor: Ying Xu, University of Georgia, UNITED STATES

Received: January 13, 2015

Accepted: May 28, 2015

Published: July 10, 2015

Data Availability Statement: All relevant data are within the paper and its Supporting Information files.

Funding: The authors have no support or funding to report.

Competing Interests: The authors have declared that no competing interests exist.

Introduction

Analysis of periodic patterns is an essential part of many studies of gene expression involving timeline sampling or targeting of rhythmically expressed genes. Recent publications report a large proportion of the entire transcriptome oscillating in a circadian (i.e. approximately daily) rhythm [1–3]. The number of genes for which circadian baseline can be identified as statistically significant over stochastic noise is traditionally thought to be under 10% [4–6], but more recently estimated as 43% [1] or even close to 100% [7], depending on the algorithms applied. Significance of the signal-to-noise ratio is the focus of most studies targeting rhythmic expression. The absolute amplitude and time of the peak (i.e. phase) of rhythmic gene expression are also analyzed and reported. However, we feel that one aspect of rhythmic gene expression required additional consideration. It has been observed that low sampling frequency presents a significant challenge to all studies of periodic gene expression ([7] for review). Most gene expression studies only report 6 or 9 observation points per period and not more than two consecutive periods in the entire timeline. Some oscillating genes may have peak expression coinciding at, or near, the observation point (i.e. the time when the animal is sacrificed and tissue samples are taken for analysis). However, other genes may peak at any time between sparsely placed observations. Since our ability to differentiate events in time is restricted by the low

sampling rate, how can we be sure that genes are expressed in the phase we identified? Would it be possible to make a quantitative estimation of confidence that a gene peaks at a certain time of the day? With such a metric we could separate a fraction of genes for which we know the true time of peak and analyze the function of genes at a given time with less noise (i.e. genes highly expressed, but peaking at a different time) mixed in. To answer these questions and enable time-wise analysis of gene function and interactions, we propose a novel algorithm for the estimation of confidence of phase assignment in analysis to timeline expression profiles.

To answer the questions posed for this study we propose to use the bootstrap, which is a general technique for estimating unknown quantities associated with statistical models. Often the bootstrap is used to find

- standard errors for estimators,
- confidence intervals for unknown parameters,
- p-values for test statistics under a null hypothesis.

The maximum entropy bootstrap [8] is a resampling method for observations that are not necessarily independent and/or identically distributed. These conditions match typical observations of gene expressions time series. The maximum entropy bootstrap is an algorithm that constructs a large number of replicates (such as $R = 999$) that retain the basic shape, local peaks and troughs and time independence of the original time series, by being strongly dependent on it. The maximum entropy bootstrap is particularly useful for short time series.

Materials and Methods

Notations

I: indicator function.
n: the sample size.
p: the number of genes.
$\chi = \{X_1, \ldots, X_n\}$: random sample from population.
$\chi^* = \{X_1^*, \ldots, X_n^*\}$: resample obtained by sampling from χ.
α: level of confidence.
$\hat{\theta}$: estimate of θ, computed from χ.
$\hat{\theta}^*$: bootstrap version of $\hat{\theta}$, computed from χ^*.

Phase estimation

We consider a gene expression time series $\{x_1, \ldots, x_n\}$. Without loss of generality, suppose that the measurements are taken in time points $t = 0, 4, 8, \ldots, 44h$. We can then construct a collection of intervals named *phases* and labeled $G_0, G_1, \ldots, G_5$ such that

$$G_0 = [-2, 2], G_1 = [2, 6], G_2 = [6, 10], G_3 = [10, 14], G_4 = [14, 18], G_5 = [18, 22]. \quad (1)$$

Let θ be the first *peak time* or the *phase* of the gene expression. We are interested in *estimating* and in a later step *constructing a confidence interval* for θ. More precisely, we want to construct an interval contained in one of the classes G_i, and that contains the estimated parameter $\hat{\theta}$ with high probability.

The expression profile of a gene exhibiting circadian rhythmicity approximates to a cosine wave with a period $T = 24$h. A significant correlation can therefore be found between rhythmically expressed gene and a theoretical cosine wave cycling with an appropriate phase. The process of estimating θ consists of the following steps:

1. Generate 6 cosine waves with the equation given below

$$\mathcal{C}_{\varphi}(t) = \cos\left(\frac{2\pi}{T}t - \varphi\right), t = 0, 4, 8, ..., 44; \varphi \in \{0, \pi/3, 2\pi/3, \pi, 4\pi/3, 5\pi/3\}. \quad (2)$$

The following properties apply: the periods are 24h, 48h long (two cycles), and the intervals between adjacent phases is 4h. Fig 1 is a graphical representation of the cosine (Eq 2).

2. Calculate the correlation coefficient between the gene expression profile and each of the 6 cosine waves $\mathcal{C}_{\varphi}$. Let $R = \{\hat{\rho}_0, \hat{\rho}_1, ..., \hat{\rho}_5\}$ denotes the obtained vector of correlations. Let

$$\hat{\rho} = \max R, \quad (3)$$

be the highest correlation and $\hat{\varphi}$ the phase of the corresponding cosine wave. The optimal $\mathcal{C}_{\hat{\varphi}}$ is selected to be the representative of the circadian rhythmicity if the correlation is significant. Our parameter of interest θ is then estimated by the peak of the *best-correlated* cosine curve, and it is equal to

$$\hat{\theta} = \hat{\varphi}T/2\pi = 12\hat{\varphi}/\pi. \quad (4)$$

Data resampling using Maximum Entropy Bootstrap Algorithm

Several bootstrap methods have been proposed for time series data. The most well-known is the *Moving Block Bootstrap*. This procedure works by dividing the observations in blocks of length b and then resampling the blocks (See Fig 2 for an illustration). The main problem with the block bootstrap is that the block length, b, which is a form of smoothing parameter, needs to be chosen. If the blocks are too short, the bootstrap samples cannot mimic the original sample. In this case dependency is broken whenever we start a new block. If, on the other hand, the blocks are too long, we will lose the randomness of the replicates. For these reasons, in this study we apply the maximum entropy bootstrap algorithm proposed by [8]. It does not impose strong assumptions on the distribution of the time series like stationarity. A full description of the algorithm can be found in [9]. The replications are generated by the following steps

1. Form order statistics $x_{(t)}$ by sorting increasingly the original data, and keep the vector of ordering index.
2. Using the ordering statistics obtained at step 1, compute the intermediate points $z_{(t)} = (x_{(t)}+x_{(t+1)})/2$ for $t = 1, \ldots, n-1$.
3. For $t = 1, \ldots, n$, construct the deviation $x_{(t)} - x_{(t-1)}$, and calculate the trimmed mean m_{trm} of the obtained observations. The lower limit for left tail is $z_0 = x_{(1)} - m_{\text{trm}}$ and upper limit for right tail is $z_n = x_{(n)}+m_{\text{trm}}$. z_0 and z_n are the new limiting intermediate points.
4. Compute the mean of the maximum entropy (ME) density within each interval while satisfying the *mean-preserving constraint*.
5. Generate uniformly distributed numbers on the [0, 1] interval, then calculate sample quantiles of the Maximum Entropy at the generated points and sort them.
6. Using the ordering index of step 1, reorder the sorted sample. This process permits to conserve the dependance relationships among observations in the original data.
7. The steps 2 to 6 are repeated many times, in our analysis we use $R = 999$.

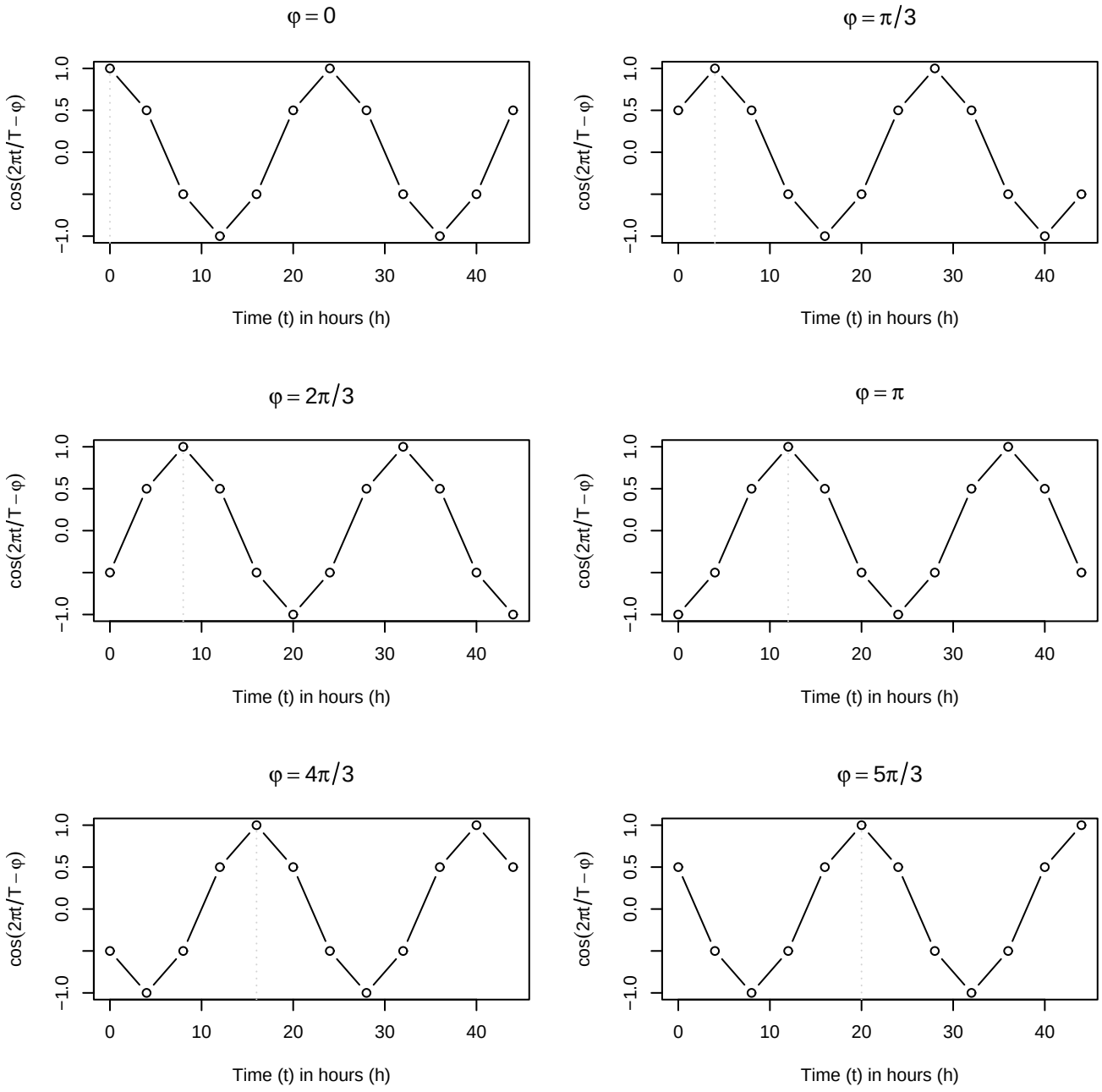

Fig 1. Graph of the ideal cosines. Graph representing the cosine waves: $cos(\frac{2\pi}{T}t - \varphi)$ for $t = 0,4,8, \ldots, 44$ and $\varphi \in \{0, \pi/3, 2\pi/3, \pi, 5\pi/3\}$. The dotted vertical line shows the first *peak time*.

doi:10.1371/journal.pone.0131111.g001

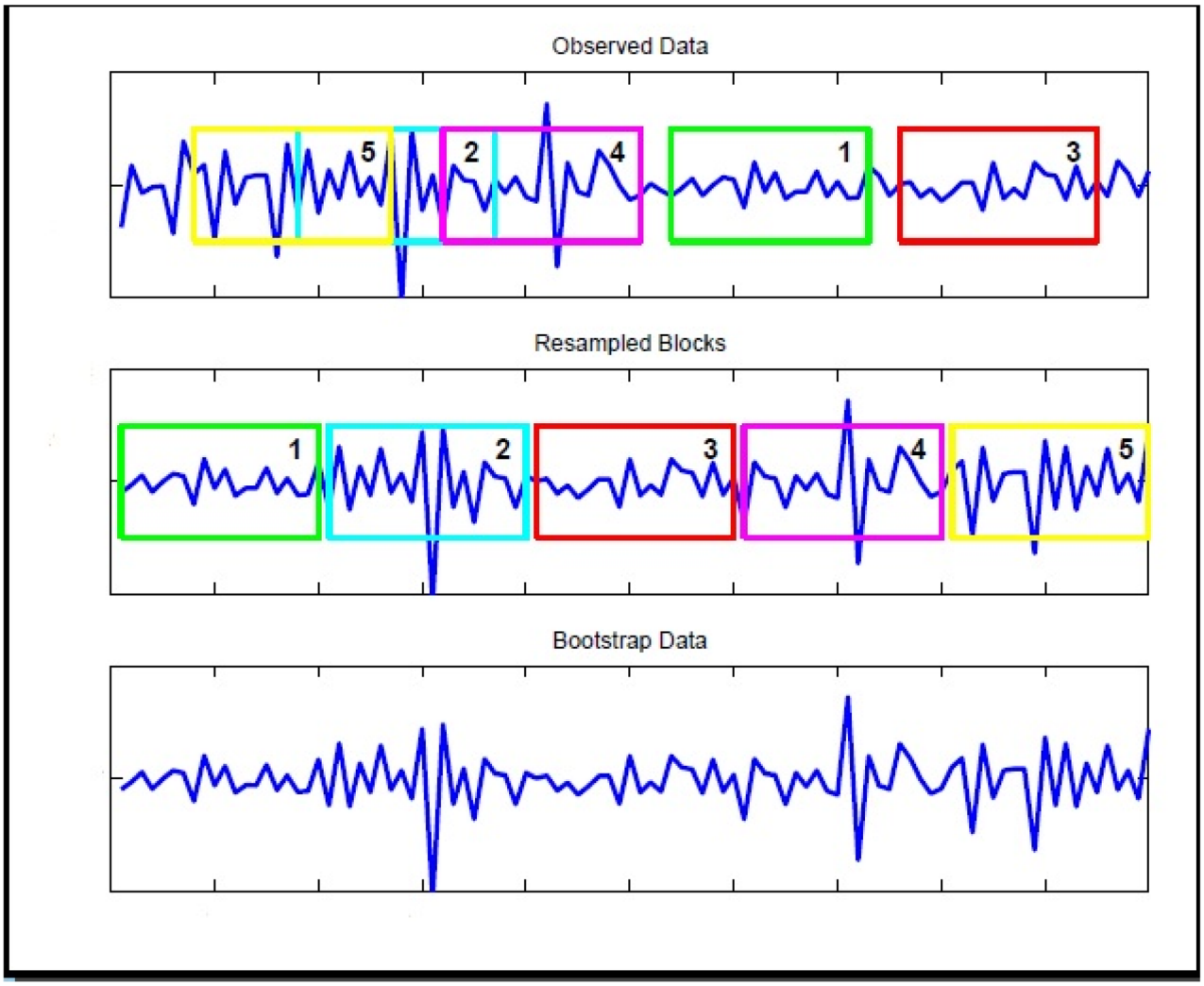

Fig 2. Graph of the moving block bootstrap principle. Graph showing the principal of moving block bootstrap. The moving block bootstrap randomly selects blocks of the original data (top) and concatenate them together (center) to form a resample (bottom).

doi:10.1371/journal.pone.0131111.g002

A complete simulated example for illustration of each step of the algorithm can be found in [9]. Fig 3 shows one gene expression time series from the IWAT data, along with 24 different replicates of the series chosen randomly from 999 used in the analysis. Due to the fact that the maximum entropy algorithm tries to retain all the properties of the data, one can see that the replicates remain close to the original time series.

The Bootstrap Approach for *p*-value

Let $\hat{\tau}$ denote the realized value of a test statistic τ computed for a particular sample. Then $\mathbb{P}(\tau \geq \hat{\tau} \mid H_0)$ is the definition of the p-value in situations where large values of τ support the alternative hypothesis. The process of calculating p-value consists of the following steps:

1. Specify a way to generate bootstrap samples that resemble the real data while satisfying the null hypothesis H_0. In our case we will use the **Maximum Entropy Bootstrap Algorithm**.
2. Let **MEBA** denote this **bootstrap data-generating process**.

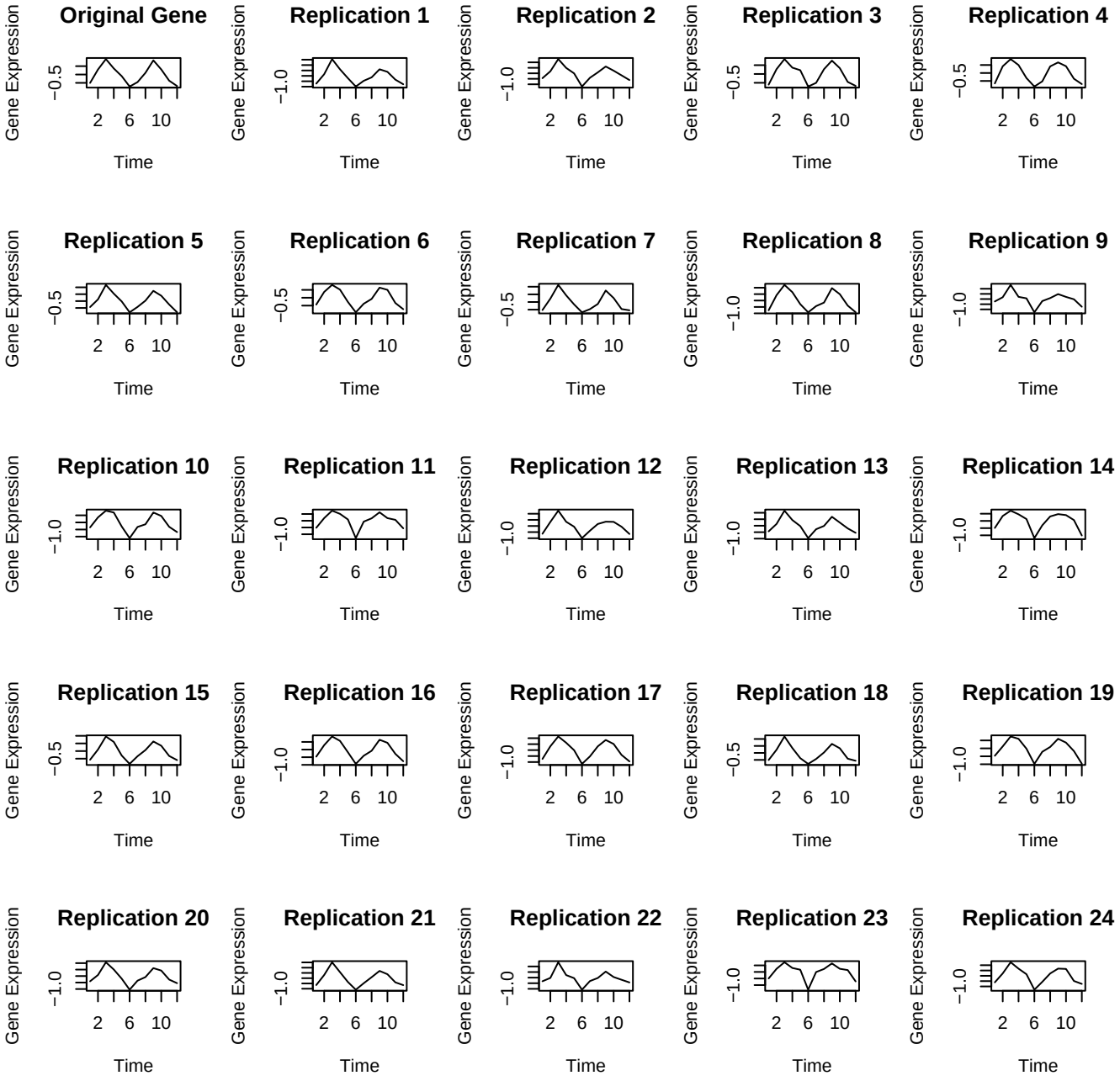

Fig 3. An Example of data resampling using the Maximum Entropy Bootstrap Algorithm. (Top left panel): A gene expression time series from the IWAT data. (Remaining:) Set of 24 replications randomly chosen from 999 maximum entropy bootstrap samples used in the analysis.

doi:10.1371/journal.pone.0131111.g003

3. Using **MEBA**, generate $R = 999$ bootstrap samples indexed by j. From each of them, compute a bootstrap test statistic τ_j^*. To estimate a bootstrap p-value, we use

$$\hat{p}^*(\hat{\tau}) = \frac{1 + \sum_{j=1}^{R} \mathrm{I}_{\{\tau_j^* > \hat{\tau}\}}}{1 + R}. \tag{5}$$

Arguments in favor of the latter formulae for calculating p-value instead of the classical formulae $\sum_{j=1}^{R} \mathrm{I}_{\{\tau_j^* > \hat{\tau}\}}/R$, can be found in [10], p. 148, 161). For example, if 73 of the τ_j^* are greater than $\hat{\tau}$, then $\hat{p}^*(\hat{\tau}) = (1 + 73)/(1 + 999) = 0.074$.

4. Reject the null hypothesis H_0 if $\hat{p}^*(\hat{\tau}) < \alpha$. Where α is a given constant satisfying $0 < \alpha < 1$. In general we take $\alpha = 0.05$.

This algorithm will be used to assess significance of the correlation between a gene expression time series and one of the cosine (Eq 2).

Bootstrap Percentile Confidence Interval

The main focus of this paper is to give an accurate approximate confidence interval for *peak time* parameter $\hat{\theta}$. Computing such confidence intervals with distributions that are difficult to represent mathematically, is very challenging. The bootstrap is another class of general methods for constructing confidence intervals without making strong distributional assumptions about the data or the statistic being calculated. There are several ways to construct bootstrap confidence intervals. They vary in ease of calculation and accuracy. There have been three main lines of development: Efron's original percentile method [11], the bootstrap t interval introduced in [12], and the double bootstrap interval introduced in [13]. In this work, due to its simplicity and good performance, we use the Bootstrap Percentile Confidence Interval.

Let $\hat{\theta}$ be an estimator of θ on the measured data $X_1, \ldots, X_n$, and $\hat{\theta}^*$ be its analog on a bootstrapped sample $X_1^*, \ldots, X_n^*$, then:

$$K_{\mathrm{boot}}(x) = \mathbb{P}_*(\{\hat{\theta}^* \leq x\}). \tag{6}$$

Where K_{boot} is the empirical distribution function of the bootstrap values. Efron's (1979) original $100(1 - 2\alpha)\%$ bootstrap *percentile interval* is to just take the empirical 100α and $100(1 - \alpha)$ percentiles from the bootstrap values $\hat{\theta}_1^*, \ldots, \hat{\theta}_R^*$. Then the $100(1 - 2\alpha)\%$ percentile interval is

$$[\underline{\theta}_{\mathrm{bp}}, \overline{\theta}_{\mathrm{bp}}] = [K_{\mathrm{boot}}^{-1}(\alpha), K_{\mathrm{boot}}^{-1}(1 - \alpha)], \tag{7}$$

where K_{boot}^{-1} is the inverse or the generalized inverse distribution function or quantile function. The name percentile comes from the fact that $K_{\mathrm{boot}}^{-1}(\alpha)$ and $K_{\mathrm{boot}}^{-1}(1 - \alpha)$ are percentiles of the bootstrap distribution K_{boot} in (Eq 6). In practice, we proceed as follows:

1. Generate R bootstrap samples of size n using the maximum entropy algorithm.
2. Estimate the parameter θ of interest for each bootstrap sample: $\hat{\theta}_b^*$ for $b = 1, \ldots, R$.
3. Order the bootstrap replications of $\hat{\theta}$ such that $\hat{\theta}_{(1)}^* \leq \hat{\theta}_{(2)}^* \leq \ldots \hat{\theta}_{(R)}^*$. The lower and upper confidence bounds are the $R.\alpha^{\mathrm{th}}$ and $R.(1 - \alpha)^{\mathrm{th}}$ ordered elements, respectively. The estimated $(1 - 2\alpha)$ confidence interval of $\hat{\theta}$ is

$$[\underline{\theta}_{\mathrm{bp}}, \overline{\theta}_{\mathrm{bp}}] = [\hat{\theta}_{(R.\alpha)}^*, \hat{\theta}_{(R.(1-\alpha))}^*]. \tag{8}$$

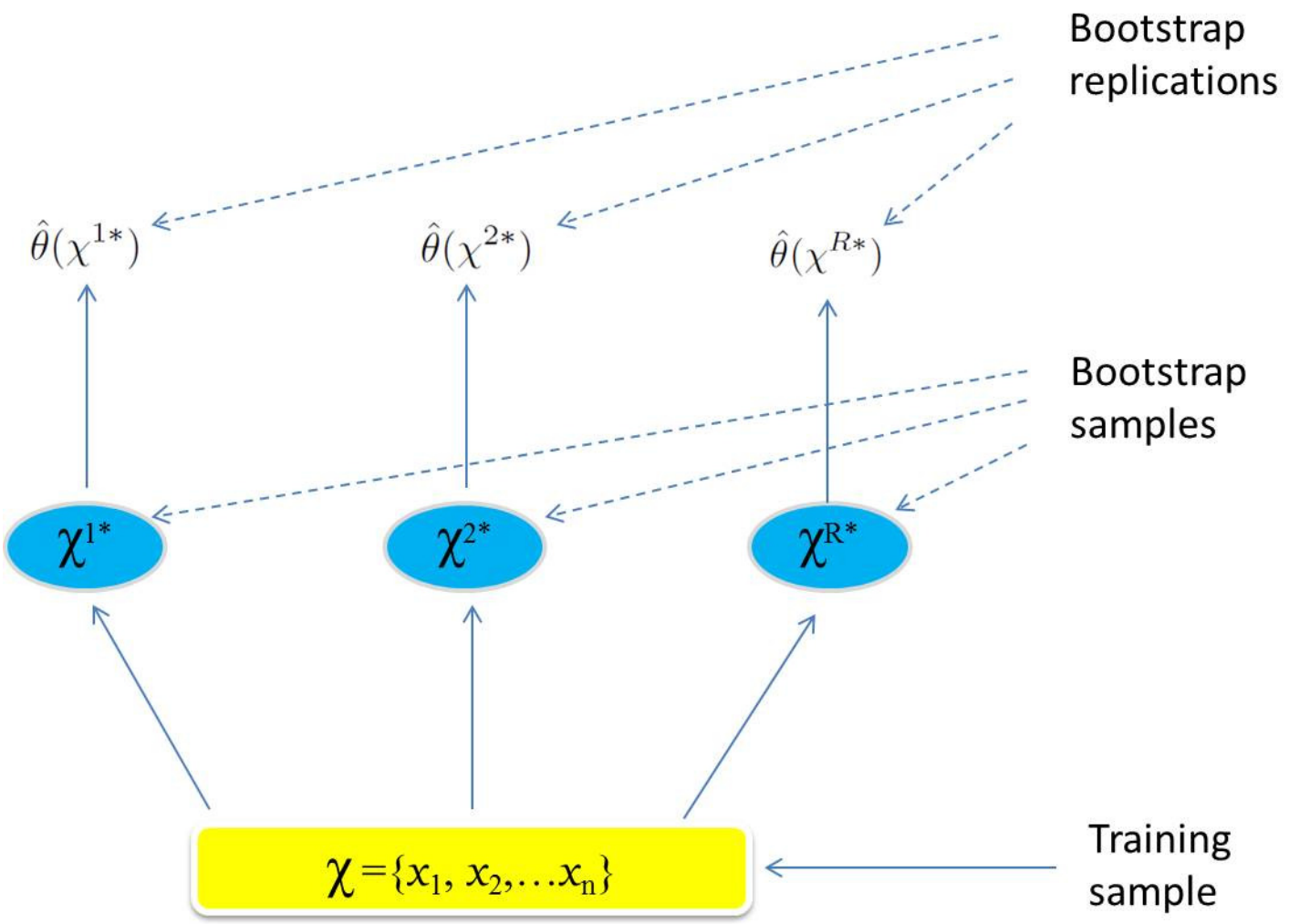

Fig 4. The Bootstrap Percentile confidence interval principle. Schematic of the bootstrap process. We want to estimate a confidence interval for the phase $\theta(\chi)$. R training sets, $\chi^{1}*, \ldots, \chi^{R}*$ each of size n are generated using an appropriate resampling mechanism. The quantity of interest $\theta(\chi)$ is computed from each bootstrap training set, and the values $\theta(\chi_1^*), \ldots, \theta(\chi_R^*)$ are used to construct a confidence interval for the quantity $\theta(\chi)$.

doi:10.1371/journal.pone.0131111.g004

Fig 4 summarizes the steps of the Bootstrap Percentile confidence interval principle.

Remark 1. If $R.\alpha$ is not an integer, the following procedure can be used:

Let $k = [(R + 1)\alpha]$, the largest integer $\leq (R + 1)\alpha$. Then we define the empirical α and $(1 - \alpha)$ quantities by the k^{th} largest and $(R + k - 1)^{\text{th}}$ values of $\hat{\theta}^*_{(b)}$, respectively. So if $R = 999$ and $\alpha = 2.5\%$ these are the 25^{th} and 975^{th} ordered elements.

We have now all the pieces needed to accomplish the phase confidence analysis. Algorithm 1 summarizes the details of the proposed approach

Algorithm 1: Confidence in phase definition for periodicity in genes expression time series

Data: $\chi = \{x_1, \ldots, x_n\}$: n realizations of a gene expression time series, the number of replications R, and a confidence level α.

Result: Bootstrapped p-value, Bootstrap Percentile Confidence Interval $[\underline{\theta}_{\text{bp}}, \bar{\theta}_{\text{bp}}]$.

for $b \leftarrow 1$ **to** R **do**
    Using the maximum entropy bootstrap algorithm, generate a bootstrap sample χ^{b^*};
    Calculate the maximum correlation $\hat{\rho}_b$ using formula (Eq 3);
    Estimate the peak time $\hat{\theta}_b$ using formula (Eq 4);
Calculate the bootstrapped p-value $\hat{p}^*(\hat{\rho})$ using formula (Eq 5);
if $\hat{p}^*(\hat{\rho}) \leq \alpha$ **then**
    the gene is considered as circadian.
Calculate the Bootstrap Percentile Confidence Interval $[\underline{\theta}_{\text{bp}}, \bar{\theta}_{\text{bp}}]$ using formula (Eq 8).
if *it exist* $i \in \{0, \ldots, 5\}$ such that $[\underline{\theta}_{\text{bp}}, \bar{\theta}_{\text{bp}}] \subset G_i$ **then**
    the gene is assigned to the phase G_i, where G_j are defined in (Eq 1).

Results, Discussion, and Conclusions

We conducted experiments on three real previously published data sets. The data are derived from microarray study of gene expression in three tissues in mice referred as Inguinal White Adipose tissue (IWAT), Brown Adipose Tissue (BAT) and Liver. Each individual data set contains more than 22,000 gene expression profiles. Each profile consists of 12 time points of 4-h interval difference. See [14] for detailed description. In the first step of our analysis, we estimated the phase of each gene using the Eq (4), and we identified the circadian gene expression based on the Algorithm 1. We note here that our aim is not to identify all the circadian genes, but we are more interested in genes for which the peak time is near to one of the time points where the measurements are taken. Detection of circadian genes can be sophistically performed using Fisher's g-test, autocorrelation or permutation test (See [15] for more details). This estimation revealed 646 oscillatory genes in the IWAT data, 680 in the BAT data, and 747 in the Liver data for which the bootstrapped p-value was ≤ 0.05, representing 6.9%, 7.15%, and 7.6% of the number of oscillatory genes obtained by applying a permutation test, respectively.

We used our proposed method to calculate a 95% confidence interval $[\underline{\theta}_{\text{bp}}, \bar{\theta}_{\text{bp}}]$ for the *peak time* of the oscillating genes, and then we assigned a phase to each of them using the following rule: a circadian gene is assigned to a Phase G_i if $[\underline{\theta}_{\text{bp}}, \bar{\theta}_{\text{bp}}] \subset G_i$.

The Results of phase classification are summarized in Table 1 and Fig 5. In the IWAT data, and with a confidence levels of at least 95%, 28 genes peak at Phase G_0, 47 at Phase G_1, 223 at Phase G_2, 128 at Phase G_3, 117 at Phase G_4, and 103 at Phase G_5, representing 4.33%, 7.27%, 34.52%, 19.81%, 18.11%, and 15.94% of the oscillating genes, respectively. In the BAT data, 57 peak at Phase G_0, 110 at Phase G_1, 158 at Phase G_2, 145 at Phase G_3, 147 at Phase G_4, and 63 at

Table 1. Number of genes in each phase for the IWAT, BAT and Liver data sets.

Phase/Data	IWAT	BAT	Liver
Phase G_0	28	57	68
Phase G_1	47	110	136
Phase G_2	223	158	164
Phase G_3	128	145	176
Phase G_4	117	147	136
Phase G_5	103	63	67
Total	646	680	747

doi:10.1371/journal.pone.0131111.t001

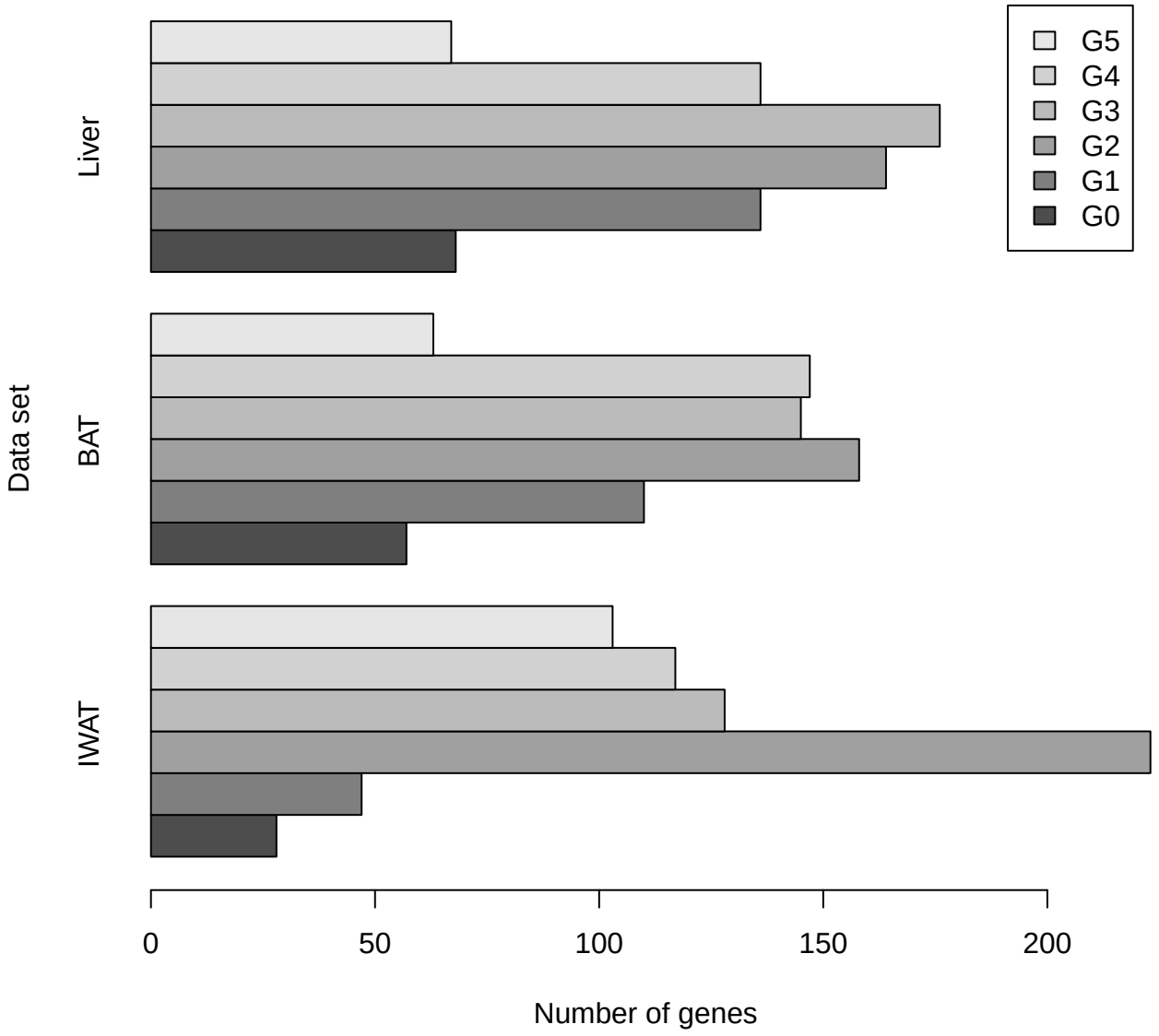

Fig 5. Barplot of the number of genes against phases. Bar plot summarizing the number of genes in each phase for the IWAT, BAT and Liver data sets from the results in Table 1.

doi:10.1371/journal.pone.0131111.g005

Phase G_5, representing 8.38%, 16.17%, 23.23%, 21.32%, 21.61%, and 9.26% of the oscillating genes, respectively. For the Liver data set, 68 genes peak at Phase G_0, 136 at Phase G_1, 164 at Phase G_2, 176 at Phase G_3, 136 at Phase G_4, and 67 at Phase G_5, representing 9.10%, 18.20%, 21.95%, 23.56%, 18.20%, and 8.97% of the oscillating genes, respectively. The method for estimation of phase assignment confidence that we proposed allows some useful observation even on the testing data. For instance, we may ask how uniform is gene expression over time? For

the experiments collecting data in circadian timeline we can formulate the Null-hypothesis stating that the same number of genes can be confidently assigned to each phase group. The alternative hypothesis would state that at least one phase group has significantly different number of genes. Both hypotheses are consistent with the overall number of rhythmically expressed genes and cannot be testes without quantitative estimation of confidence of phase assignment. In our test data we apply the same $p = 0.05$ threshold, but observe fewer genes peaking at one of the phases. In biological terms this means the in murine adipose tissue there is a period (morning hours) when the overall gene expression activity is lower compared to all other times of the day.

However, it is even more important that our method can be applied to increase precision of observation in many studies involving timeline observation of gene expression. The sampling frequency still imposes limitation on our ability to separate molecular events (such as peak of gene expression) in time. To know the time of peak expression more precisely the experiment has to be repeated with higher a number of time points (for example, one sample every 2 hours rather than every 4 hours). However, with our method we can refine the existing data. For the groups peaking at a certain time we can be confident (at a selected confidence level) that certain genes peak at a certain time and filter out genes peaking sometime between out observation time points. This confidence is essential for functional annotation of co-expressed genes and can be critical in analysis of permutation of gene activity in reaction to environment or medication.

Strengths and boundaries

We compare the proposed method with some competing algorithms, namely Fisher's *g*-test [16], Permutation test [15], and JTK-CYCLE [17]. All methods except the permutation test are implemented in R, and run on an Itel core *i*7 at 3.40 GHz. The permutation test is implemented in C++. Tables 2, 3 and 4 show some results for the IWAT, BAT and Liver data sets.

In this paper, we are interested in genes that may have a peak expression coinciding or near one of the observation points. We approximate their expression profiles by an ideal cosine wave of the form:

$$\mathcal{C}_{\varphi}(t) = \cos\left(\frac{2\pi}{T}t - \varphi\right), t = 0, 4, 8, ..., 44; \varphi \in [0, 2\pi[. \quad (9)$$

We know that for circadian genes we have $T = 24$h. For the data sets used in this paper, the measurements time are $t \in \{0,4,8, \ldots, 44\}$. Since we are interested by the first peak expression time, the possible time points to be considered are $t \in \{0,4,8, \ldots, 20\}$. If we solve for equations $\mathcal{C}_{\varphi}(t) = 0$ for $t \in \{0,4,8, \ldots, 20\}$, we obtain $\varphi \in \{0, \pi/3, 2\pi/3, \pi, 4\pi/3, 5\pi/3\}$, this explains the use of $\pi/3$ as a resolution power of estimated phase in Eq (2). If we choose different values of the resolution power of estimated phase, the peak time of the generated ideal cosine waves will not necessarily coincide with one of the time points when the measurements were taken. Nevertheless, the method is general. It can work for periods other than 24 hours, for different spacing time points, and it can works with a larger number of cosines waves with smaller phases. For example, for any integer k we can generate $2k$ cosine waves using the equation:

$$\mathcal{C}_i = \cos\left(2\pi\left(\frac{1}{24}t - \frac{1}{2k}i\right)\right) = \cos\left(\pi\left(\frac{1}{12}t - \frac{1}{k}i\right)\right), t = 0, 4, 8, ..., 44; i = 0, 1, 2, ..., 2k - 1. \quad (10)$$

Table 2 shows some timing results for $k = 30$, which generates 60 cosine waves. Results are given for $R \in \{9,99,999\}$ bootstrap replications. Like any method based on resampling, the proposed method can be computationally expensive, because it involves fitting the same statistical method a large number of times using different replications of the original data. We can see

Table 2. IWAT, BAT and Liver data sets: timings (in minutes (m) or in hours (h)) for the proposed method and a variant of it that uses a set of 60 cosine waves with smaller phases generated using the Eq (10). The number of bootstrap replications R is in {9, 99, 999}.

Data set	IWAT			BAT			Liver		
Method	***R* = 9**	***R* = 99**	***R* = 999**	***R* = 9**	***R* = 99**	***R* = 999**	***R* = 9**	***R* = 99**	***R* = 999**
Proposed Method using Eq (2)	3.71(m)	32.20(m)	5.53(h)	3.67(m)	32.54(m)	5.75(h)	3.61(m)	32.39(m)	5.79(h)
Proposed Method using Eq (10)	11.40(m)	1.86(h)	19.83(h)	11.39(m)	1.82(h)	19.83(h)	11.45(m)	1.81(h)	19.73(h)

doi:10.1371/journal.pone.0131111.t002

Table 3. IWAT, BAT and Liver data sets: timings (seconds) for Fisher's *g*-test, Permutation test, JTK-CYCLE, and the proposed method on one bootstrap replication.

Data set Method	IWAT	BAT	Liver
Fisher's *g*-test	11.79(secs)	12.27(secs)	11.87(secs)
JTK-CYCLE	16.49(secs)	14.34(secs)	14.59(secs)
Proposed Method (One replication)	22.50(secs)	22.63(secs)	22.41(secs)

doi:10.1371/journal.pone.0131111.t003

that the average CPU timings increases with number of generated cosine waves and the number of bootstrap replications.

Table 3 shows some timing results for the three different datasets; Fisher's *g*-test is faster, followed by JTK-CYCLE and then the proposed method (one replication). We note here that the computing performance of the proposed method can be enhanced considerably (See **Remark 4**).

Table 4 shows the number of identified circadian genes. The Permutation test identifies the highest number, followed by the JTK-CYCLE and then Fisher's *g*-test. Our method is not developed for detecting all the circadian genes, but rather it detects, with high confidence, the circadian gene for which the peak time (the phase) is near one of the time points; estimates this phase, and constructs a confidence interval for it. This explains the small number of circadian genes detected by our method compared to the competitors.

Remark 2. This experiment design is rather typical for circadian biology. Some experiments collect samples at different intervals, such as 3h or, rarely, every 2h. Higher sampling frequency improves resolution ability, but costs a lot more and is harder to implement.

Remark 3. Gene expression profiles are analyzed independently, thus it is possible that a researcher may find few or none of the gene confidently peaking at a given time. In fact, in the data set on which we tested the method, gene expression has a quiet period at which relatively few genes are active.

Table 4. IWAT, BAT and Liver data sets: number of circadian genes identified using Fisher's *g*-test, Permutation test and JTK-CYCLE respectively.

Data set Method	IWAT	BAT	Liver
Fisher's *g*-test	4177	4547	5030
Permutation test	9321	9441	9775
JTK-CYCLE	6646	6868	7354
Proposed Method	646	680	747

doi:10.1371/journal.pone.0131111.t004

Remark 4. We note that the computational performance of the proposed method can be enhanced. In fact, if we avoid using *loops* in R script that process one element per iteration, and instead we use *apply* family of functions that process whole rows, columns, or lists, the computing time is reduced significantly. In this case we need just 0.001 second to run the method for one replication using Eq (2), and we need 0.008 second to run the method using higher number of cosine waves using Eq (10).

Supporting Information

S1 R Codes. R Analysis Codes.
(PDF)

S1 Data. IWAT data measurements.
(TXT)

S2 Data. BAT data measurements.
(TXT)

S3 Data. Liver data measurements.
(TXT)

Acknowledgments

We thank Christopher Leonard from QScience, Qatar Foundation, for improving the quality of the manuscript.

Author Contributions

Analyzed the data: ME. Wrote the paper: ME AP. Developed the algorithm and implemented the code: ME. Discussed the results: AF AP. Formulated the problem: AP.

References

1. Zhang R, Lahens NF, Ballance HI, Hughes ME, Hogenesch JB. A circadian gene expression atlas in mammals: implications for biology and medicine. Proceedings of the National Academy of Sciences. 2014; 111(45):16219–16224. doi: 10.1073/pnas.1408886111
2. Klevecz RR, Li CM, Marcus I, Frankel PH. Collective behavior in gene regulation: the cell is an oscillator, the cell cycle a developmental process. FEBS journal. 2008; 275(10):2372–2384. doi: 10.1111/j.1742-4658.2008.06399.x PMID: 18410382
3. Ptitsyn AA, Reyes-Solis G, Saavedra-Rodriguez K, Betz J, Suchman EL, Carlson JO, et al. Rhythms and synchronization patterns in gene expression in the Aedes aegypti mosquito. BMC genomics. 2011; 12(1):153. doi: 10.1186/1471-2164-12-153 PMID: 21414217
4. Panda S, Antoch MP, Miller BH, Su AI, Schook AB, Straume M, et al. Coordinated transcription of key pathways in the mouse by the circadian clock. Cell. 2002; 109(3):307–320. doi: 10.1016/S0092-8674(02)00722-5 PMID: 12015981
5. Storch KF, Lipan O, Leykin I, Viswanathan N, Davis FC, Wong WH, et al. Extensive and divergent circadian gene expression in liver and heart. Nature. 2002; 417(6884):78–83. doi: 10.1038/nature744 PMID: 11967526
6. Bray MS, Shaw CA, Moore MW, Garcia RA, Zanquetta MM, Durgan DJ, et al. Disruption of the circadian clock within the cardiomyocyte influences myocardial contractile function, metabolism, and gene expression. American Journal of Physiology-Heart and Circulatory Physiology. 2008; 294(2):H1036–H1047. doi: 10.1152/ajpheart.01291.2007 PMID: 18156197
7. Ptitsyn AA, Gimble JM. True or false: All genes are rhythmic. Annals of medicine. 2011; 43(1):1–12. doi: 10.3109/07853890.2010.538078 PMID: 21142579
8. Vinod HD. Maximum entropy ensembles for time series inference in economics. Journal of Asian Economics. 2006; 17(6):955–978. doi: 10.1016/j.asieco.2006.09.001

9. Vinod HD, López-de Lacalle J. Maximum entropy bootstrap for time series: the meboot R package. Journal of Statistical Software. 2009; 29(5):1–19.
10. Davison AC, Hinkley DV. Bootstrap methods and their application. vol. 1. Cambridge university press; 1997.
11. Efron B. Bootstrap methods: another look at the jackknife. The annals of Statistics. 1979;p. 1–26.
12. Efron B, Efron B. The jackknife, the bootstrap and other resampling plans. vol. 38. SIAM; 1982.
13. Hall P. On the bootstrap and confidence intervals. The Annals of Statistics. 1986;p. 1431–1452.
14. Zvonic S, Ptitsyn AA, Conrad SA, Scott LK, Floyd ZE, Kilroy G, et al. Characterization of peripheral circadian clocks in adipose tissues. Diabetes. 2006; 55(4):962–970. doi: 10.2337/diabetes.55.04.06.db05-0873 PMID: 16567517
15. Ptitsyn AA, Zvonic S, Conrad SA, Scott LK, Mynatt RL, Gimble JM. Circadian clocks are resounding in peripheral tissues. PLoS Comput Biol. 2006; 2(3):e16. doi: 10.1371/journal.pcbi.0020016 PMID: 16532060
16. Wichert S, Fokianos K, Strimmer K. Identifying periodically expressed transcripts in microarray time series data. Bioinformatics. 2004; 20(1):5–20. doi: 10.1093/bioinformatics/btg364 PMID: 14693803
17. Hughes ME, Hogenesch JB, Kornacker K. JTK CYCLE: an efficient nonparametric algorithm for detecting rhythmic components in genome-scale data sets. Journal of biological rhythms. 2010; 25(5):372–380. doi: 10.1177/0748730410379711 PMID: 20876817