POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Uncovering Circadian Genes with Bayesian Fourier Analysis

Bayesian Statistics Course - Mathematical Engineering

Michele Birardi, Celeste De Bernardinis, Giulia de Sanctis, Alessandro Di Gianni, Marta Krychkovska, Jacopo Lazzari

Academic year:
2024-2025

Abstract: Circadian rhythms are 24-hour biological cycles that help organisms anticipate and adapt to environmental changes. They are controlled by the circadian clock, a network of mutually interacting proteins that generate transcriptional and translational feedback loops. The identification of clock-genes is performed through examination of their expression levels or "transcripts". Statistical methods are needed to detect periodic pathways among a very high number of gene expression profiles. The purpose of this study was to use a Bayesian approach that identifies periodic signals in gene expression profiles while accounting for dependence in the functional data. We built on Montagna, Irincheeva, and Tokdar (2018) to develop a C++ algorithm for detecting periodic signals, with an R wrapper for usability. The algorithm has been tested on a real-world dataset of Arabidopsis Thaliana. We know of 26 clock-associated genes in Arabidopsis, and we use them as a benchmark to evaluate our approach. We use a Markov-Chain-Monte-Carlo (MCMC) algorithm to update the model parameters. Using the posterior samples, we can estimate the posterior probability of a gene being circadian. We rely on this estimate to rank the genes from most to least likely of being circadian. Our results show that the model is able to identify the 26 periodic genes with a reasonable level of accuracy, and outperforms competing approaches in terms of AUC. This method provides a novel approach to identifying circadian genes and it has an improved performance compared to other widely-used rhythmicity detection techniques that do not directly accommodate for dependence across genes.

Key-words: Bayesian latent factor models, Circadian rhythms, Latent threshold methods, Dependent functional data

1. Introduction

This project is based on the paper *High-dimensional Bayesian Fourier Analysis for Detecting Circadian Gene Expression* by Montagna, S., Irincheeva, I., and Tokdar, S. T. (2018) [1].

Circadian rhythms are cycles of biological activity based on a 24-hour period which allow organisms to anticipate and adapt to predictable daily oscillations in the environment (Hughes et al. 2010) [2]. Circadian rhythms are controlled by the circadian clock, namely a network of mutually interacting proteins that generate transcriptional and translational feedback loops (Jouffe et al. 2013) [3]. The molecular mechanisms underlying the circadian clock have been investigated in many organisms (Wichert et al., 2004 [4]), [3]. In plants, circadian rhythmicity has been extensively studied in the Arabidopsis Thaliana (Anderson et al. 2006 [6], Edwards et al. 2006 [5], Liverani 2009 [7]) . In animals, sleep-wake cycles are circadian-regulated to maximize the availability of food as well as to avoid predation. In humans, blood pressure, hormone production, metabolism and other biological cycles are clock-regulated, and disruptions to the circadian rhythms have been linked to a variety of pathologies [2]. Consequently, there is a considerable interest in identifying genes that control the timing of many physiological processes.

The identification of clock-genes is performed through examination of their expression levels or "transcripts". With regard to the functioning of a cell, deoxyribonucleic acid (DNA) is first duplicated into messenger ribonucleic acid (mRNA), and the RNA is then used for protein synthesis. To quantify the expression of a specific gene, it is possible to measure the concentration of RNA molecules associated with this gene. By using this principle, microarray analysis allows investigators to measure many hundreds or thousands of transcripts simultaneously, and then statistical methods are needed to detect periodic pathways among a very high number of gene expression profiles.

Several authors have proposed methods for periodicity identification in biomedical research over the last couple of decades. Chudova et al. (2009) [8] give an excellent review of the main existing techniques, which can be broadly classified as time domain or frequency domain analyses. Time domain methods are not very effective at finding periodic signals that are not perfectly sinusoidal. On the other hand, Chudova et al. (2009) [8] remark that frequency domain methods are most effective on long time series. However, this is not a typical feature of circadian studies, which are usually designed to collect data every 2 or 4 hours over two circadian cycles. Therefore, coarse sampling and short periods of data collection are typical features of these studies.

A key assumption in the most widely-used rhythmicity detection techniques is the independence across transcripts. Although practical from a computational perspective, the independence assumption is often too strong to be realistic in many applications.

Montagna S. et al. (2018) [1] propose a Bayesian approach that identifies periodic signals in gene expression profiles while accounting for dependence in the functional data. Specifically, the true underlying signal for each transcript is decomposed into a series expansion of sine and cosine (Fourier) waves. Conditional dependence across genes at each time point is accommodated via a latent factor framework. Dimensionality reduction and sparsity are induced through careful modelling of the latent factors as well as the Fourier basis coefficients.

The main objectives of this project are to transition the existing code from MATLAB to C++ in order to achieve performance improvements and minimize execution time. The optimized algorithm will then be validated using a real-world dataset, the Arabidopsis Thaliana [5] dataset, with its performance evaluated in comparison to the original implementation. Additionally, a comprehensive literature review will be conducted to compare the performance of our model with that of other existing models for circadian gene detection.

2. Data

2.1. Synthetic Data

During the code translation phase, we worked exclusively with synthetic data, where 33 genes were simulated as being circadian. Here we briefly explain how the process of doing simulations with synthetic data works.

We simulated y_i, $i = 1, \dots, p = 500$, from a $T = 24$-dimensional normal distribution with mean $\mathbf{B}\boldsymbol{\theta}_i + \boldsymbol{\eta}\boldsymbol{\lambda}_i$ and covariance matrix $\sigma^2 \mathbf{I}_T$, with $\sigma^2 = 0.5$. The design matrix $\mathbf{B}$ includes the Fourier bases with possible periods $\{4, 6, 8, 12, 24\}$ hours, thus $q = 5$. The true number of factors was set equal to $k = 6$, and the number of non-zero elements in each column of $\mathbf{A}$ was chosen linearly between $2 \times (10 \log p)$ and $10 \log p + 1$. In practice, this resulted in a number of non-zero elements between 99 and 124 across the different columns of $\boldsymbol{\Lambda}$. We randomly allocated the location of the zeros in each column and simulated the non-zero elements independently from a normal distribution with mean 0 and variance 9. The latent factors $\boldsymbol{\eta}$ were independently generated by sampling from a standard normal distribution. The $p \times q$ true latent thresholds for Θ were independently generated from a $\text{Unif}(0, 6)$. The rows of $\mathbf{W}$ were independently generated by sampling from a standard normal distribution, and the true values of the latent coefficients $\{\tilde{\Theta}_i\}_{i=1}^{p}$ were generated by sampling from their prior distribution given the true values of $\mathbf{W}$ and Λ.

We ran the Gibbs sampler for 10000 iterations with a burn-in of 1000, and collected every 5th sample to thin the chain. The hyperparameters a_σ and b_σ for σ_i^{-2} were 1 and 0.5, respectively, while $\rho = 3$, $a_1 = 2.1$, $a_2 = 3.1$, $a_\Theta = 1$, $\beta_\Theta = 5$ and we used $k = 5$ as the starting number of factors.
Only 33 of the 500 simulated profiles were simulated to be circadian. Therefore, the signal-to-noise ratio is quite weak in this dataset.

Circadian genes do not necessarily follow a perfectly sinusoidal trajectory. In fact, the model includes a jitter component, introducing deviations from the sinusoidal pattern. Plotting the synthetic data can be useful to better understand the characteristics of genes known to be circadian.

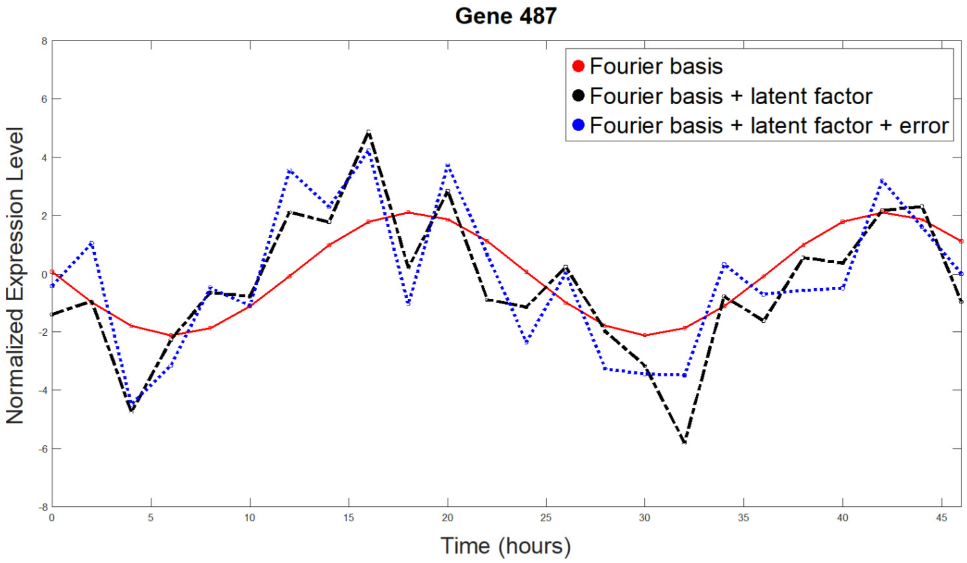


Figure 1: Decomposition of the trajectory of a gene simulated as circadian according to the model.

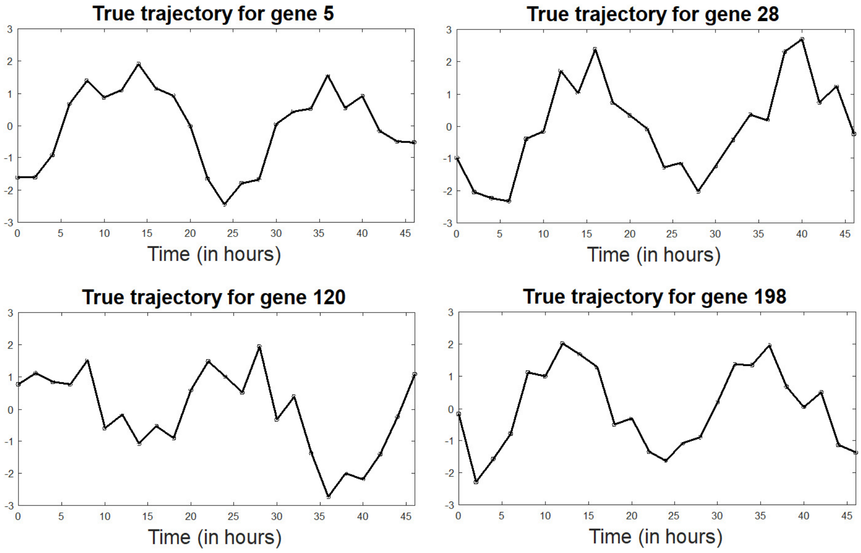


Figure 2: 48-hour trajectories of 4 different genes simulated as being circadian.

2.2. Real Data

Our reference dataset on which we performed our analysis is the Arabidopsis thaliana dataset [5]. This plant is particularly well-known in the scientific community as a model organism in plant sciences and was one of the first plant genomes to be fully sequenced. Dodd et al. [11] identify 26 known clock-associated genes in Arabidopsis. We use these 26 well-established circadian genes as a benchmark to evaluate our approach.

The dataset contains measurements taken from eight-day-old Columbia seedlings grown under cycles of 12 hours of light followed by 12 hours of darkness, which were subsequently transferred to constant light at a temperature of 22°C. Plant samples were collected at 13 time points, spanning two circadian cycles in 4-hour intervals, starting 26 hours after the last dark-light transition. In this context, we have $p = 22810$ genes and $T = 13$ time points.

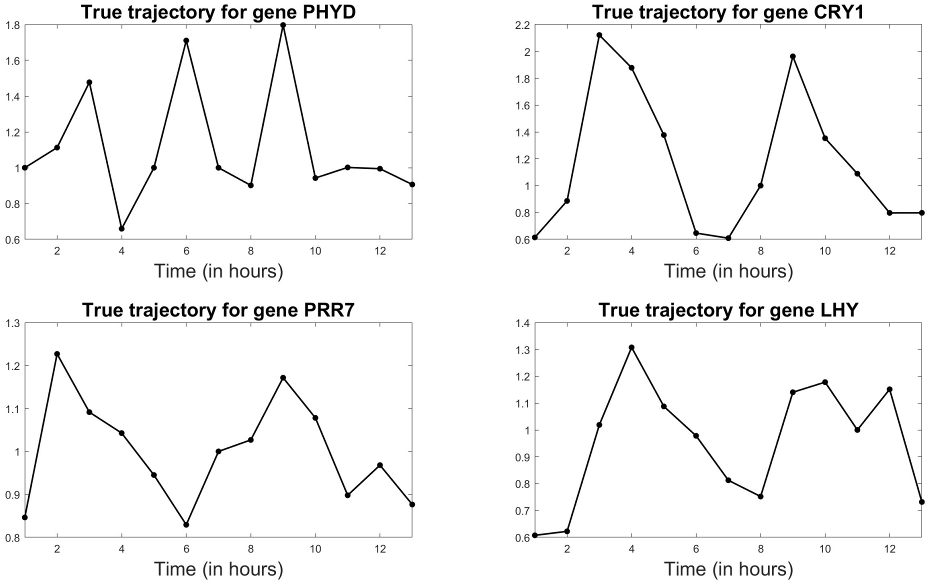


Figure 3: 48-hour trajectories of 4 different genes known to be circadian.

3. Methodology

We consider data from a gene expression experiment in the form of a $p \times T$ matrix $\boldsymbol{Y} = [\boldsymbol{y}_j]$, where entry y_{ij} denotes the observed mRNA concentration for gene i at time t_j, for $i = 1, \ldots, p$, and with p denoting the total number of genes. In circadian microarray studies, data is typically collected over two complete circadian cycles and the sampling rate is usually either two or four hours.

We assume that the y_{ij} are error-prone measurements of an underlying smooth true trajectory

$$y_{ij} = f_i(t_j) + \nu_{ij}. \tag{1}$$

Suppose that the de-trended and centered true signal for gene i at time t_j, $f_i(t_j)$, can be decomposed as

$$f_i(t_j) = \sum_{m=1}^{q} \left(b_{2m-1}(t_j)\theta_{i,2m-1} + b_{2m}(t_j)\theta_{i,2m}\right) = \boldsymbol{\theta}_i^\top \boldsymbol{b}_m(t_j) = \boldsymbol{\theta}_i^\top \boldsymbol{b}_j,$$

where for $m = 1, \ldots, q$, we define $\boldsymbol{\theta}_i = (\theta_{i,2m-1}, \theta_{i,2m})^\top$ and $\boldsymbol{b}_m = [b_{2m-1}(t_j), b_{2m}(t_j)]^\top$. The vector

$$\boldsymbol{b}_j = [b_1(t_j), b_2(t_j), \ldots, b_{2q-1}(t_j), b_{2q}(t_j)]^\top$$

represents a set of $2q$ fixed basis functions evaluated at time t_j. One popular basis for a space of periodic functions is the **Fourier basis**

$$b(t) = \left[\sin\left(\frac{2\pi}{\omega_1}t\right), \cos\left(\frac{2\pi}{\omega_1}t\right), \ldots, \sin\left(\frac{2\pi}{\omega_q}t\right), \cos\left(\frac{2\pi}{\omega_q}t\right)\right]^\top,$$

where $\{\omega_m\}$ denotes the periodicity of the signal and is time represented by a unit-interval increase. The q period lengths ω_m are assumed known and fixed.

The term ν_{ij} in Equation 1 models the deviation between the observed measurement at time t_j, y_{ij}, and the underlying smooth profile.

3.1. Dependence across genes

To accommodate for dependence across probes at time j, we adopt a **sparse factor model**:

$$\boldsymbol{\nu}_j = \boldsymbol{A}\boldsymbol{\eta}_j + \boldsymbol{\epsilon}_j, \tag{2}$$

where $\boldsymbol{\nu}_j = [\nu_{1j}, \ldots, \nu_{pj}]^\top$, $\boldsymbol{A} = [\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_p]^\top$ is a $p \times k$ factor loading matrix with elements $\{\lambda_{ih}\}_{i=1,\ldots,p;h=1,\ldots,k}$, $\boldsymbol{\eta}_j = (\eta_{1j}, \ldots, \eta_{kj})^\top$ is a $k \times 1$ vector of latent factors at time j which explains deviations of the expression levels at time j from their corresponding truth (it explains why proteins at time j may be systematically over- or under-expressed with respect to the “truth"), and $\boldsymbol{\epsilon}_j$ is a residual error.

The full model for subject i at time t_j is

$$y_{ij} = f_i(t_j) + \lambda_i^\top \boldsymbol{\eta}_j + \epsilon_{ij}, \quad \epsilon_{ij} \sim N(0, \sigma_i^2), \tag{3}$$

where the first term $f_i(t_j) = \boldsymbol{b}_j^\top \boldsymbol{\theta}_j$ captures periodic oscillations (if present), and the second term $\lambda_i^\top \boldsymbol{\eta}_j$ captures across-proteins dependence (if present).

The term $\boldsymbol{\theta}_i$ is the $2q \times 1$ vector of fixed periodic basis function coefficients for protein i. Greater T corresponds to more flexibility in modeling local deviations. We follow standard practice in normalizing the data prior to analysis and hence do not include an intercept term in 3. We use $\boldsymbol{y}_i = (y_{i1}, \ldots, y_{iT})^\top$ to denote the i-th row of $\boldsymbol{Y}$ (the i-th protein observed at times $1, \ldots, T$), and $\boldsymbol{y}^{(j)} = (y_{1j}, \ldots, y_{pj})^\top$ to denote the j-th column of $\boldsymbol{Y}$ (proteins observed at the time j). We can rewrite equation 3 in vector notation as:

$$\boldsymbol{y}_i = \boldsymbol{B}\boldsymbol{\theta}_i + \boldsymbol{\eta}\lambda_i + \boldsymbol{\epsilon}_i, \tag{4}$$

where

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{b}_1^\top \\ \vdots \\ \boldsymbol{b}_T^\top \end{bmatrix} \in R^{T\times 2q}, \quad \boldsymbol{\lambda}_i \in R^k, \quad \boldsymbol{\eta} = \begin{bmatrix} \boldsymbol{\eta}_1^\top \\ \vdots \\ \boldsymbol{\eta}_T^\top \end{bmatrix} \in R^{T\times k} \quad \boldsymbol{\epsilon_i} = (\epsilon_{i1}, \dots, \epsilon_{iT})^\top \sim N_T(\mathbf{0}, \sigma_i^2 \boldsymbol{I}_T).$$

with $T \times T$ identity matrix $\boldsymbol{I}_T$.
Or:

$$\boldsymbol{y}^{(j)} = \boldsymbol{\Theta}\boldsymbol{b}_j + \boldsymbol{\lambda}\boldsymbol{\eta} + \boldsymbol{\epsilon}^{(j)}, \tag{5}$$

where

$$\boldsymbol{\Theta} = \begin{bmatrix} \boldsymbol{\theta}_1^\top \\ \vdots \\ \boldsymbol{\theta}_p^\top \end{bmatrix} \in R^{p\times 2q}, \quad \boldsymbol{\Lambda} = \begin{bmatrix} \boldsymbol{\lambda}_1^\top \\ \vdots \\ \boldsymbol{\lambda}_p^\top \end{bmatrix} \in R^{p\times k}, \quad \boldsymbol{\epsilon}^{(j)} = (\epsilon_{1j}, \dots, \epsilon_{pj})^\top \sim N_p(\mathbf{0}, \boldsymbol{\Sigma}), \quad \boldsymbol{\Sigma} = \mathrm{diag}(\sigma_1^2, \dots, \sigma_p^2).$$

The latent factors $\boldsymbol{\eta}$ explain the dependence structure across proteins at time t_j.

Hereafter we follow standard practice and assign a normal prior to the latent factors at time t_j, $\boldsymbol{\eta_j} \sim N(\mathbf{0}, \boldsymbol{I}_k)$. Proteins are assumed to be independent given the latent factors, and dependence among proteins is induced by marginalizing over the distribution of the factors, so marginally $\boldsymbol{y}^{(j)} \sim N(\boldsymbol{\Theta b_j}, \boldsymbol{\Lambda\Lambda^T} + \boldsymbol{\Sigma})$. In practical applications involving a moderate to large p, the number of factors k is typically much smaller than p, thus inducing a sparse characterization of the unknown covariance matrix $\boldsymbol{\Lambda\Lambda^T} + \boldsymbol{\Sigma}$.

4. Prior elicitation

With regards to the modeling of the basis coefficients $\{\boldsymbol{\theta}_i\}_{i=1}^p$ we need to induce sparsity, therefore avoid overfitting, whilst retaining an easy interpretation of the method. The latter is particularly crucial for the modeling of $\boldsymbol{\theta}_i$ since inference on this set of parameters is the primary interest of this project.
We adopt the latent threshold model (LTM) of Nakajima and West (2013) [10]. The LTM is a direct extension of standard Bayesian variable selection which assigns non-zero prior probabilities to zero values of regression parameters, and continuous priors centered at zero otherwise.
We adopt the same variable selection prior for the periodic basis coefficients. Denote with

$$\boldsymbol{\theta}_{im} = (\theta_{i,2m-1}, \theta_{i,2m})^\top$$

the vector of the $(2m-1)$th and $2m$th components of $\boldsymbol{\theta_i}$, with $m = 1, \dots, q$. Thus, θ_{2m-1} is the coefficient of the $(2m-1)$th sine basis, and θ_{2m} is the coefficient of the $(2m)$th cosine basis, both harmonics of period w_m. To ensure a correct interpretation of periodicity, we need to jointly switch off θ_{2m-1} and θ_{2m} only if supported by the data. Therefore, we assume:

$$\boldsymbol{\theta}_{i,m} = \tilde{\boldsymbol{\theta}}_{i,m} 1\left(\|\tilde{\boldsymbol{\theta}}_{i,m}\| \geq \varpi_{i,m}\right), \tag{6}$$

where $\varpi_{i,m}$ is a latent threshold. The idea behind equation 6 is that the value of the ω_m-periodic basis coefficients is shrunk to zero when their norm falls below a $2m$th- (and protein-) specific threshold.
Furthermore, we assume that $\tilde{\boldsymbol{\theta}}_{\boldsymbol{i}} = \{\tilde{\boldsymbol{\theta}}_i\}_{m=1}^q$ is modeled as

$$\tilde{\boldsymbol{\theta}}_i = \boldsymbol{W}\boldsymbol{\lambda_i} + \boldsymbol{\alpha}_i^\theta \quad \text{and} \quad \boldsymbol{\alpha}_i^\theta \overset{\text{iid}}{\sim} N_{2q}(\mathbf{0},\ \mathbf{I}), \tag{7}$$

where $\mathbf{W}$ is a $2q \times k$ matrix and $\boldsymbol{\lambda}_i$ is the vector of factor loadings for protein i. We assume $\mathbf{W}_j \sim N_k(\mathbf{0}, \mathbf{I})$, $j = 1, \dots, 2q$. This simple structure on $\boldsymbol{\theta}_{i,m}$ in 6 allows to flexibly take into account the dependence among parameters $\boldsymbol{\theta}_{i,m}$, and λ_i.

To continue, we adopt a multiplicative gamma process shrinkage prior (MGPSP) on the loadings

$$\lambda_{ih} \mid \phi_{ih}, \tau_h \overset{\text{iid}}{\sim} N(0, \phi_{ih}^{-1}\tau_h^{-1}), \quad \phi_{ih} \overset{\text{iid}}{\sim} Ga(\rho/2, \rho/2), \quad \tau_h = \prod_{l=1}^{h} \zeta_l$$

$$\zeta_1 \sim Ga(a_1, 1), \quad \zeta_l \overset{\text{iid}}{\sim} Ga(a_2, 1), \quad l \geq 2, \quad i = 1, \ldots, p,$$

with $h = 1, \ldots, k$, the number of latent factors. τ_h is a global shrinkage parameter for the h-th column, and ϕ_{ih} is a local shrinkage parameter for the elements in the h-th column. In matrix notation, row i of $\mathbf{\Lambda}$ has prior

$$\boldsymbol{\lambda}_i^\top \mid \{\phi_{ih}\}_{h=1}^{k}, \{\tau_h\}_{h=1}^{k} \overset{\text{iid}}{\sim} \mathcal{N}_k(\mathbf{0}, \boldsymbol{D}_i), \tag{8}$$

with $\boldsymbol{D}_i = \text{diag}\left(\phi_{i1}^{-1}\tau_1^{-1}, \ldots, \phi_{ik}^{-1}\tau_k^{-1}\right)$.

This prior introduces a changing rate of shrinkage. As the column index of $\mathbf{\Lambda}$ increases there is more shrinkage and thus factor splitting is avoided by concentrating more and more shrunken loadings in the last columns of $\mathbf{\Lambda}$. The MGPSP was originally proposed by Bhattacharya and Dunson (2011) [9] for sparse modeling of high-dimensional covariance matrices. The authors embedded the MGPSP into an adaptive Gibbs sampler which allowed for block update of the rows of the $\mathbf{\Lambda}$ while accounting for an adaptive choice of the number of factors, k. The main idea consisted of monitoring the columns $\mathbf{\Lambda}$ whose loadings were all within some pre-specified neighborhood of zero. If the number of such columns dropped to zero, one extra column was added to $\mathbf{\Lambda}$, otherwise the redundant columns were discarded. In our method, we adopt the same adaptive block Gibbs sampler in order to retain only the important factors and therefore also reduce computational time.

To conclude the model formulation, we need to specify prior distributions on the latent threshold parameters. The straightforward extension of Nakajima and West (2013) [10] to our scenario leads to a dependent prior for $\varpi_{i,m}$ of the type $\varpi_{i,m} \overset{\text{iid}}{\sim} \text{Unif}(0, K_\theta)$ for $i = 1, \ldots, p$ and $m = 1, \ldots, q$, where K_θ is a fixed parameter in our case.

5. Posterior update

Given the observed data $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^{p}$, we wish to infer the periodic basis functions coefficients $\{\boldsymbol{\theta}_i\}_{i=1}^{p}$, the factor loading matrix $\mathbf{\Lambda}$, the $T \times k$ matrix of latent factors $\boldsymbol{\eta}$, and all the relevant hyper-parameters. We use Gibbs sampling by successively drawing samples from the full conditional distributions of each parameter in turn, given all other parameters.

The conditional distribution of $\mathbf{Y}$ implied by 4 is

$$\mathbf{Y} \mid \mathbf{B}, \mathbf{C}, \mathbf{\Theta}, \boldsymbol{\eta}, \mathbf{\Lambda}, \mathbf{\Sigma} = \prod_{i=1}^{p} N(\mathbf{y}_i \mid \mathbf{B}\boldsymbol{\theta}_i + \boldsymbol{\eta}\boldsymbol{\lambda}_i, \sigma_i^2 \mathbf{I}_T) \tag{9}$$

and the likelihood function is

$$P(\mathbf{Y}, \mathbf{\Theta}, \tilde{\mathbf{\Theta}}, \mathbf{\Sigma}, \boldsymbol{\eta}, \mathbf{\Lambda}, \boldsymbol{\phi}, \boldsymbol{\tau}, \boldsymbol{\varpi}) = \prod_{i=1}^{p} \Bigg\{ N(\mathbf{y}_i \mid \mathbf{B}\boldsymbol{\theta}_i + \boldsymbol{\eta}\boldsymbol{\lambda}_i, \sigma_i^2 \mathbf{I}_T)\, \text{Ga}(\sigma_i^{-2} \mid a_\sigma, b_\sigma) \times \tag{10}$$

$$N_k(\boldsymbol{\lambda}_i^\top \mid \mathbf{0}, \mathbf{D}_i(\boldsymbol{\phi}, \boldsymbol{\tau}))\, p(\boldsymbol{\phi} \mid \rho)\, p(\boldsymbol{\tau} \mid a_1, a_2) \prod_{j=1}^{T} N_k(\boldsymbol{\eta}_j \mid \mathbf{0}, \mathbf{I}_k) \times$$

$$N_{2q}(\tilde{\boldsymbol{\theta}}_i \mid \mathbf{W}\boldsymbol{\lambda}_i, \text{Var}(\boldsymbol{\alpha}_i^\theta)) \times p(K_\theta) \times \prod_{j=1}^{2q} N_k(\mathbf{W}_j \mid \mathbf{0}, \mathbf{I}_k) \times p(\boldsymbol{\varpi}) \Bigg\},$$

where $p(\boldsymbol{\phi}|\rho)$ and $p(\boldsymbol{\tau}|a_1, a_2)$ are the densities of prior distributions induced by MGPSP on vectors of all $\{\phi_{ih}\}_{i=1,\dots,p;\ h=1,\dots,k}$ and all $\{\tau_h\}_{h=1,\dots,k}$ respectively, and $p(\boldsymbol{\varpi})$ is the density of prior distribution induced on vectors of all $\{\varpi_{i,m}\}_{i=1,\dots,p;\ m=1,\dots,q}$ In what follows we use "–" to denote the "rest" of the model, i.e. all random variables not explicitly mentioned in the current state of the Markov Chain. Using the introduced notations we describe a MCMC algorithm for simulation of the full joint posterior distribution of the model parameters.

- **Update of W**: We place a conjugate normal prior on the columns of the $k \times 2q$ matrix $\mathbf{W}^\top$, so $\mathbf{W}_l \sim N_k(\mathbf{0}, \mathbf{I})$, $l = 1, \dots, 2q$. This is equivalent to a prior on the rows of matrix $\mathbf{W}$, $\mathbf{W}_l^\top$. Conditioning on the current estimate of $\tilde{\theta}_{i,l} \sim N(\boldsymbol{\lambda}_i^\top \mathbf{W}_l, 1)$ and other model parameters, the posterior update of $\mathbf{W}_l$ is

$$\mathbf{W}_l \mid - \sim N_k\left(\left(\sum_{i=1}^{p} \boldsymbol{\lambda}_i\boldsymbol{\lambda}_i^\top + \mathbf{I}\right)^{-1}\left(\sum_{i=1}^{p} \tilde{\theta}_{i,l}\boldsymbol{\lambda}_i\right), \left(\sum_{i=1}^{p} \boldsymbol{\lambda}_i\boldsymbol{\lambda}_i^\top + \mathbf{I}\right)^{-1}\right).$$

- **Update of $\boldsymbol{\lambda}_i^\top$**: We place a MGPSP on row i of $\boldsymbol{\Lambda}$ (equivalently, column i of $\boldsymbol{\Lambda}^\top$) as in 8. The likelihood contribution factorizes as

$$L(\boldsymbol{\lambda}_i \mid \tilde{\boldsymbol{\Theta}}, \boldsymbol{\Theta}, \eta, \boldsymbol{\Sigma}, \mathbf{W}) \propto N_T(\mathbf{y}_i \mid \mathbf{B}\boldsymbol{\theta}_i + \boldsymbol{\eta}\boldsymbol{\lambda}_i, \sigma_i^2\mathbf{I}_T) \times N(\tilde{\boldsymbol{\theta}}_i \mid \mathbf{W}\boldsymbol{\lambda}_i, \mathrm{Var}(\boldsymbol{\alpha}_i^\theta)). \tag{11}$$

We assume $\mathrm{Var}(\boldsymbol{\alpha}_i^\theta) = \mathbf{I}_{2q}$. The posterior update of $\boldsymbol{\lambda}_i$ is

$$\boldsymbol{\lambda}_i \mid - \sim N_k\left(\mathbf{V}_{\boldsymbol{\lambda}_i}\mathbf{M}_{\boldsymbol{\lambda}_i}, \mathbf{V}_{\boldsymbol{\lambda}_i}\right), \quad i = 1, \dots, p, \tag{12}$$

where

$$\mathbf{M}_{\boldsymbol{\lambda}_i} = \sigma_i^{-2}\boldsymbol{\eta}^\top(\mathbf{y}_i - \mathbf{B}\boldsymbol{\theta}_i) + \mathbf{W}^\top\tilde{\boldsymbol{\theta}}_i,$$
$$\mathbf{V}_{\boldsymbol{\lambda}_i} = \left(\sigma_i^{-2}\boldsymbol{\eta}^\top\boldsymbol{\eta} + \mathbf{W}^\top\mathbf{W} + \mathbf{D}_i^{-1}\right)^{-1}.$$

- **Update of $\tilde{\boldsymbol{\theta}}_i$**: We sample the conditional posterior $p(\tilde{\boldsymbol{\theta}}_i \mid -)$ sequentially for $i = 1, \dots, p$ using a Metropolis-Hastings (MH) sampler conditional on the other model parameters. The MH proposal originates from a non-thresholded version of the model. Fixing $\mathbb{1}\{\|\tilde{\boldsymbol{\theta}}_{i,m}\| \geq \varpi_{i,m}\} \equiv 1$ for $m = 1, \dots, q$, we take the proposal distribution to be $N(\tilde{\boldsymbol{\theta}}_i \mid \mathbf{m}_i, \mathbf{M}_i)$ with

$$\mathbf{M}_i = (\sigma_i^{-2}\mathbf{B}^\top\mathbf{B} + \mathbf{I}_{2q})^{-1}, \quad \mathbf{m}_i = \mathbf{M}_i(\sigma_i^{-2}\mathbf{B}^\top\tilde{\mathbf{y}}_i + \mathbf{W}\boldsymbol{\lambda}_i)$$

with $\tilde{\mathbf{y}}_i = \mathbf{y}_i - \boldsymbol{\eta}\boldsymbol{\lambda}_i$. The candidate is accepted with probability

$$\alpha(\tilde{\boldsymbol{\theta}}_i, \tilde{\boldsymbol{\theta}}_i^*) = \min\left\{1, \frac{N(\mathbf{y}_i \mid \mathbf{B}\boldsymbol{\theta}_i^* + \boldsymbol{\eta}\boldsymbol{\lambda}_i, \sigma_i^2\mathbf{I}_T)N(\tilde{\boldsymbol{\theta}}_i^* \mid \mathbf{W}\boldsymbol{\lambda}_i, \mathbf{I})N(\tilde{\boldsymbol{\theta}}_i \mid \mathbf{m}_i, \mathbf{M}_i)}{N(\mathbf{y}_i \mid \mathbf{B}\boldsymbol{\theta}_i + \boldsymbol{\eta}\boldsymbol{\lambda}_i, \sigma_i^2\mathbf{I}_T)N(\tilde{\boldsymbol{\theta}}_i \mid \mathbf{W}\boldsymbol{\lambda}_i, \mathbf{I})N(\tilde{\boldsymbol{\theta}}_i^* \mid \mathbf{m}_i, \mathbf{M}_i)}\right\}$$

where $\tilde{\boldsymbol{\theta}}_i$ ($\boldsymbol{\theta}_i$) is the current estimate and $\tilde{\boldsymbol{\theta}}_{i,m}^*$ ($\boldsymbol{\theta}_{i,m}^* = \tilde{\boldsymbol{\theta}}_{i,m}^*\mathbb{1}(\|\tilde{\boldsymbol{\theta}}_{i,m}^*\| \geq \varpi_{i,m})$) is the candidate, with $\boldsymbol{\theta}_i^* = \{\boldsymbol{\theta}_{i,m}^*\}_{m=1}^{q}$.

- **Update of $\boldsymbol{\varpi}_{i,m}$**: the update can be performed via Gibbs sampling, conditioning on the current estimate of $\tilde{\boldsymbol{\theta}}_{i,m} = \{\tilde{\theta}_{i,2m-1}, \tilde{\theta}_{i,2m}\}^\top$ and the other model parameters, for $i = 1, \dots, p$ and $m = 1, \dots, q$. If $\|\tilde{\boldsymbol{\theta}}_{i,m}\| > K_\theta$ (the upper bound of the uniform prior on $\varpi_{i,m}$), the posterior update of $\varpi_{i,m}$ is:

$$\varpi_{i,m} \mid - \sim \mathrm{Unif}(0, K_\theta).$$

Otherwise, sample

$$\varpi_{i,m} \mid - \sim \begin{cases} \mathrm{Unif}(0, \|\tilde{\boldsymbol{\theta}}_{i,m}\|), & \text{with probability } \pi^*, \\ \mathrm{Unif}(\|\tilde{\boldsymbol{\theta}}_{i,m}\|, K_\theta), & \text{with probability } 1 - \pi^*, \end{cases}$$

with:

$$\pi^* = \frac{A}{A+D},$$

$$A = N\big(\mathbf{y}_i \mid \mathbf{B}_{-m}\boldsymbol{\theta}_{i,-m} + \mathbf{B}_m\tilde{\boldsymbol{\theta}}_{i,m} + \boldsymbol{\eta}\boldsymbol{\lambda}_i, \sigma_i^2\mathbf{I}_T\big) \cdot \|\tilde{\boldsymbol{\theta}}_{i,m}\|,$$
$$D = N\big(\mathbf{y}_i \mid \mathbf{B}_{-m}\boldsymbol{\theta}_{i,-m} + \boldsymbol{\eta}\boldsymbol{\lambda}_i, \sigma_i^2\mathbf{I}_T\big) \cdot (K_\theta - \|\tilde{\boldsymbol{\theta}}_{i,m}\|),$$

with $N(\mathbf{y}_i \mid \mathbf{m}, \mathbf{v})$ denoting the Gaussian density function with mean $\mathbf{m}$ and covariance matrix $\mathbf{v}$, evaluated at $\mathbf{y}_i$. Matrix $\mathbf{B}_{-m}(\boldsymbol{\theta}_{i,-m})$ corresponds to the matrix of periodic bases (vector of periodic basis coefficients) with columns (components) $m = \{2m-1, 2m\}$ excluded. Instead, $\mathbf{B}_m(\tilde{\boldsymbol{\theta}}_{i,m})$ denotes the $\{2m-1, 2m\}$-th columns of matrix $\mathbf{B}$ (the $\{2m-1, 2m\}$-th components of $\tilde{\boldsymbol{\theta}}_i$).
Further,

$$\sigma_i^{-2} \mid - \sim \text{Ga}\left(a_\sigma + \frac{T}{2}, b_\sigma + \frac{\|\mathbf{y}_i - \mathbf{B}\boldsymbol{\theta}_i - \boldsymbol{\eta}\boldsymbol{\lambda}_i\|^2}{2}\right), \quad i = 1, \ldots, p;$$

$$\boldsymbol{\eta}_j \mid - \sim N_k\Big[\mathbf{V}_{\boldsymbol{\eta}_j}\mathbf{M}_{\boldsymbol{\eta}_j}, \mathbf{V}_{\boldsymbol{\eta}_j}\Big], \quad j = 1, \ldots, T, \quad \text{where}$$
$$\mathbf{M}_{\boldsymbol{\eta}_j} = \boldsymbol{\Lambda}^\top\boldsymbol{\Sigma}^{-1}\big(\mathbf{y}^{(j)} - \boldsymbol{\Theta}\mathbf{b}_j\big),$$
$$\mathbf{V}_{\boldsymbol{\eta}_j} = \big(\mathbf{I}_k + \boldsymbol{\Lambda}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\Lambda}\big)^{-1};$$

$$\phi_{ih} \mid - \sim \text{Ga}\left(\frac{\rho+1}{2}, \frac{\rho + \tau_h\lambda_{ih}^2}{2}\right), \quad i = 1, \ldots, p \quad \text{and} \quad h = 1, \ldots, k;$$

$$\zeta_1 \mid - \sim \text{Ga}\left(a_1 + \frac{pk}{2}, 1 + \frac{1}{2}\sum_{l=h}^{k}\tau_l^{(1)}\sum_{i=1}^{p}\phi_{il}\lambda_{il}^2\right);$$

$$\zeta_h \mid - \sim \text{Ga}\left(a_2 + \frac{p}{2}(k-h+1), 1 + \frac{1}{2}\sum_{l=1}^{k}\tau_l^{(h)}\sum_{i=1}^{p}\phi_{il}\lambda_{il}^2\right), \quad h \geq 2,$$

$$\text{where} \quad \tau_l^{(h)} = \prod_{t=1, t\neq h}^{l}\zeta_t \quad \text{for} \quad h = 1, \ldots, k.$$

6. Period detection

The LTM on the periodic basis coefficients eases the identification of those proteins that are more likely to be periodically expressed. Denote with TS the total number of thinned posterior samples post-burn-in obtained by running a Markov-Chain-Monte-Carlo (MCMC) algorithm to update the model parameters, i.e. $TS = \frac{\text{Tot. \# runs} - \text{burn-in}}{\text{thin}}$.
We can easily derive the posterior probability of any simple periodicity by counting the proportion of posterior samples for which $\{\theta_{i,2m-1}, \theta_{i,2m}\}$ are not shrunk to zero while the remaining $\boldsymbol{\theta}_i$'s are switched off. For example, the posterior probability that protein i is circadian can be computed as

$$P(\text{Protein } i \text{ is circadian}) = \frac{1}{TS}\sum_{g=1}^{TS}\mathbb{1}\left(\left\{\theta_{i,l}^{(g)}\right\}_{l=1}^{2q-2} \equiv 0 \,\text{and}\, \left\{\theta_{i,2q-1}^{(g)}, \theta_{i,2q}^{(g)}\right\} \neq 0\right) \tag{13}$$

If we were interested in quantifying the probability of a protein being periodically expressed without making any specific reference to its period, we could simply count the proportion of posterior samples for which any pair $\{\theta_{i,2m-1}, \theta_{i,2m}\}$ is *not* shrunk whereas the remaining parameters are switched off.

Biologists are interested in identifying clock proteins without incurring into too many false discoveries. Therefore, we need to compile a list of proteins for which the hypothesis of 24 hour periodicity is probably true. We want the list to be as large as possible, whilst simultaneously bounding the rate of false discoveries by some threshold, say k^*. We can rank the proteins according to increasing values of

$$\beta_i = 1 - \Pr(\text{Protein } i \text{ is circadian})$$

and declare all proteins β_i below a threshold κ, as clock-controlled proteins

$$\beta_i^* = 1(\beta_i \leq \kappa). \tag{14}$$

where β_i^* is an indicator for the decision to report protein i as circadian.
Müller et al. (2004) [12] show that 14 is the optimal decision rule under several loss functions that combine false negative and false discovery counts and/or rates, and the choice of the loss function determines the specific value of κ. In addition, the authors show that the result is true for any probability model with non-zero prior probability for periodic and non-periodic expression. In particular, the probability model can include dependence across proteins, as in our case.

7. Alternative methods and literature review

Over time, several computational methods have been developed to detect circadian rhythms in high-throughput gene expression data. In the following, we briefly present three alternative methods for identifying circadian genes, which will also be used to evaluate and compare the results obtained by our model.

Method 1: Lomb–Scargle (LS) periodograms
The Lomb-Scargle periodogram algorithm is an effective parametric method that is used to detect periodic gene expression profiles, especially when data may be collected at arbitrary time points or when a significant proportion of data is missing. This approach models gene expression data as a combination of a periodic function and normally distributed random errors.

The steps for detecting periodicity include:
1) Calculating the Lomb-Scargle periodogram for each gene at multiple test frequencies to identify the highest power peak;
2) Performing hypothesis testing to assess statistical significance of the peaks in the periodograms, with the null hypothesis being that a given gene is non-periodic versus that it is periodic;
3) Adjusting for multiple testing using the Benjamini-Hochberg false discovery rate approach to identify significantly periodic genes while controlling the rate of false positives.

Numerical experiments confirmed the effectiveness of the Lomb-Scargle periodogram for both single and multiple periodicities in simulated unevenly spaced time points.

Method 2: Autoregressive Spectral Estimation and Harmonic Regression (ARSER)
The ARSER algorithm combines time-domain and frequency-domain analyses, allowing for the identification and characterization of periodic patterns in short and noisy time-series datasets.

The methodology consists of three main steps:
1) *Data Preprocessing*: The initial step involves preparing the time-series data by de-trending and smoothing to ensure stationarity and reduce noise.
2) *Period Detection*: To identify the dominant periods in the time series, autoregressive (AR) spectral estimation is employed. This high-resolution technique estimates the power spectral density and identifies peaks

within the circadian period range (20–28 hours). The Akaike Information Criterion (AIC) is used to select the optimal periods from the detected peaks in the AR spectrum, ensuring robust period estimation even in noisy datasets.

3) *Harmonic Regression Modeling*: Once the periods are identified, harmonic regression is used to model the cyclic components in the time series. The rhythmic patterns are described using four key parameters: period (duration of a complete cycle), amplitude (magnitude of oscillation), phase (timing of the peak relative to the start of the cycle) and mean level (baseline around which the oscillations occur). The regression model is validated statistically using an F-test to assess the significance of the fitted coefficients. To control for false positives due to multiple testing, false discovery rate correction is applied.

Method 3: JTK-cycle

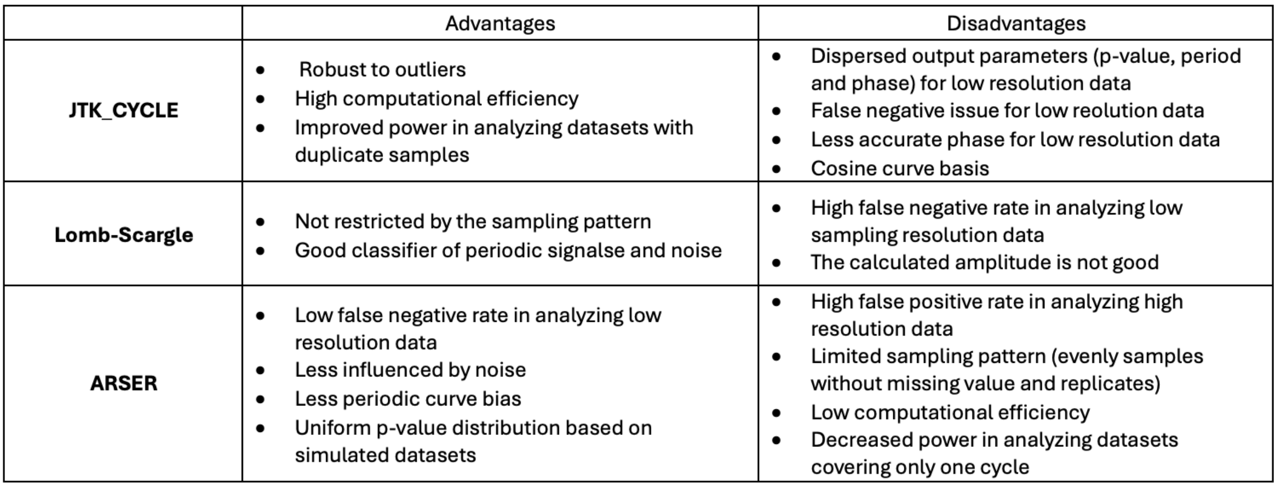
The Jonckheere-Terpstra-Kendall (JTK) algorithm builds on the Jonckheere-Terpstra (JT) test, which is a non-parametric tool for detecting monotonic trends across ordered independent groups, and on Kendall's tau, which is a measure of rank correlation.

The algorithm applies the JT test to a family of alternative hypothesized group orderings. To increase computational efficiency, JTK algorithm exploits the mathematical equivalence between the exact null JT distribution and the exact null distribution of Kendall's tau correlation between a continuous random variate and an ordinal grouping factor. [2]

JTK-cycle applies the JTK algorithm to a range of user-defined period lengths and phases. The algorithm finds the optimal combination of period and phase that minimizes the exact p-value of Kendall's tau correlation between an experimental time series and each tested cyclical ordering. Each minimal p-value is then Bonferroni-adjusted for multiple testing, ensuring conservative results.

A relevant feature of JTK-cycle is that the optimal periods and phases found by this algorithm remain invariant under monotonic transformations of the time series. This ensures the algorithm's robustness when analyzing datasets that require scaling adjustments or transformations to stabilize variance or normalize distributions. Moreover, the optimal periods and phases identified by JTK-cycle are highly resistant to outliers and extreme values. This feature makes it particularly suitable for noisy or heteroscedastic data, ensuring reliable detection of periodic patterns.

Below we include a table that compares the advantages and disadvantages of the three methods, followed by a brief overview of some additional techniques for detecting gene periodicity.



	Advantages	Disadvantages
JTK_CYCLE	• Robust to outliers • High computational efficiency • Improved power in analyzing datasets with duplicate samples	• Dispersed output parameters (p-value, period and phase) for low resolution data • False negative issue for low reolution data • Less accurate phase for low resolution data • Cosine curve basis
Lomb-Scargle	• Not restricted by the sampling pattern • Good classifier of periodic signalse and noise	• High false negative rate in analyzing low sampling resolution data • The calculated amplitude is not good
ARSER	• Low false negative rate in analyzing low resolution data • Less influenced by noise • Less periodic curve bias • Uniform p-value distribution based on simulated datasets	• High false positive rate in analyzing high resolution data • Limited sampling pattern (evenly samples without missing value and replicates) • Low computational efficiency • Decreased power in analyzing datasets covering only one cycle

Figure 4: Table with the main features of the methods presented above.

- **Bayesian Fourier Clustering** (BFC) is a Bayesian approach used to cluster gene expression profiles based on Fourier coefficients. This method provides a more flexible framework compared to sinusoidal-based methods, as it can capture complex rhythmic patterns beyond simple sine waves. By clustering genes according to their period, amplitude, and phase, BFC is able to classify circadian-regulated genes more efficiently. However, it is computationally intensive and may introduce artifacts, particularly when dealing with low-amplitude rhythms in sparse datasets. Despite these challenges, its ability to handle complex and non-sinusoidal rhythms makes it an attractive option for circadian gene detection.

- **Tempo** is a Bayesian variational inference algorithm, specifically developed for circadian phase inference in single-cell RNA-seq data. Unlike traditional methods, Tempo models gene expression using a Negative Binomial distribution and incorporates a cell-specific circadian phase as well as gene-specific parameters (mesor, amplitude, and acrophase). This makes Tempo particularly effective for single-cell datasets, where gene expression patterns can vary significantly between individual cells. Additionally, Tempo performs well with sparse and noisy data and provides uncertainty quantification for phase estimates, making it a robust tool for single-cell circadian rhythm studies. However, Tempo is best suited for single-cell RNA-seq data and may not perform as well with bulk RNA-seq or more dense datasets.

- **Autoregressive Bayesian Spectral Regression** (ABSR) offers another advanced approach to detect periodicity in gene expression data. By combining autoregressive models with spectral analysis, ABSR is well-suited for handling short, noisy time-series data. The method excels in estimating the period and amplitude even in datasets with low temporal resolution. ABSR is particularly robust when dealing with non-sinusoidal rhythms and complex gene expression profiles, where traditional methods may fail. However, ABSR is computationally demanding and requires careful model tuning to ensure optimal performance. Despite these challenges, it offers superior accuracy in noisy data environments and is effective in cases where other methods may struggle.

8. Analysis

8.1. Code

The main objective of this project is to translate the Gibbs Sampling Algorithm from MATLAB to C++. This is necessary to have faster code, since MATLAB is too slow for this application. To improve efficiency, we write C++ code with standard C++ 11 and we decided to use a library called Armadillo [13]. We chose to use it for two main reasons. Firstly, it provides optimized implementations of numerical operations such as the Cholesky or SVD decomposition, as well as for operations between matrices, both of which feature heavily in our code. Secondly, Armadillo is written with a high-level syntax and functionality that is deliberately similar to that of MATLAB, which thus makes it an ideal choice when translating from MATLAB.
The translation resulted in a ten-fold speed up of the code, below is a table that compares the execution time of the code. The code was run for 10000 iterations with 1000 as burn-in.

Data	MATLAB Execution Time (s)	C++ Execution Time (s)
Synthetic Data	585	54
Real Data (subset of 5000 genes)	12837	1120
Real Data (all 22810 genes)	n/a	6809

Table 1: Comparison of execution time of algorithm in MATLAB and C++. Test on synthetic data was run on an Apple M2 Processor from 2022 and the the test on real data was run on an AMD Ryzen 7 8845HS processor.

In order to make our code more user-friendly, we decided to provide a wrapper for the code in R. This should also make the post-processing step a lot easier since C++ is not well equipped to make plots and graphs. We allow the users some flexibility with the wrapper and ask them to provide:

- Matrix `Y` of gene transcripts
- Vector `t_ij` of time-points
- Vector `t_g` of fine grid of measurement times for prediction
- File path to save output
- List of matrices to save for post-processing
- Number of iterations
- Number of burn-in iterations

8.2. Analysis with Synthetic Data

Our analysis began with the synthetic data (Section 2.1). Below we plot the estimated trajectories of six genes that were simulated as being circadian. We see that they all show a dual peak shape.

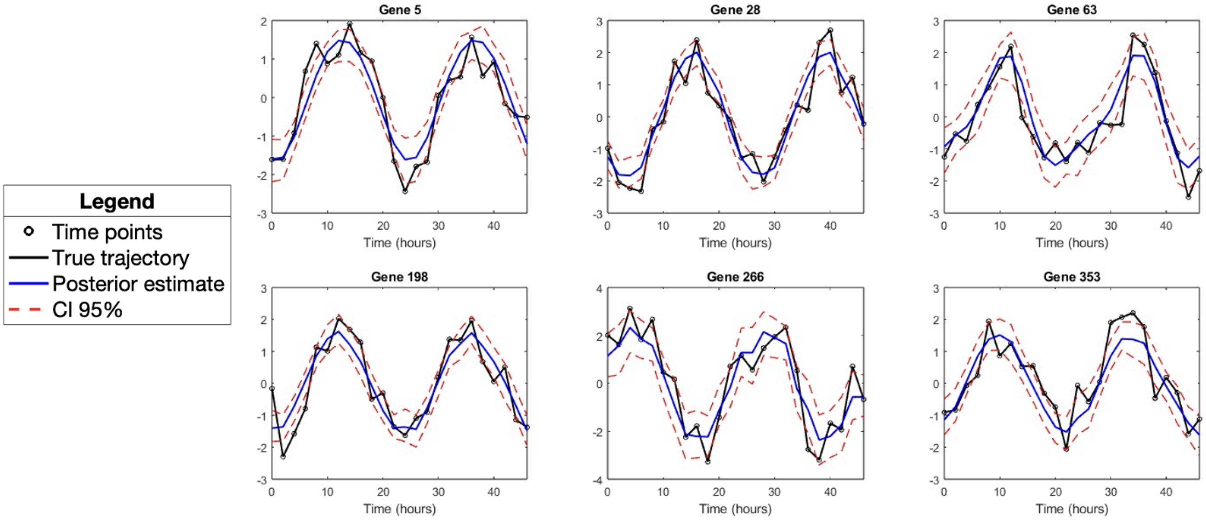


Figure 5: Estimated trajectories of six genes known as circadian from the synthetic dataset.

Below we plot the trajectories of four genes amongst the ones classified as having the highest probability of being circadian and four genes amongst the ones classified as having the lowest probability of being circadian. The "most likely" circadian genes all have a probability above 0.90 of being circadian, whereas the "least likely" circadian genes all have a probability of 0 of being circadian.

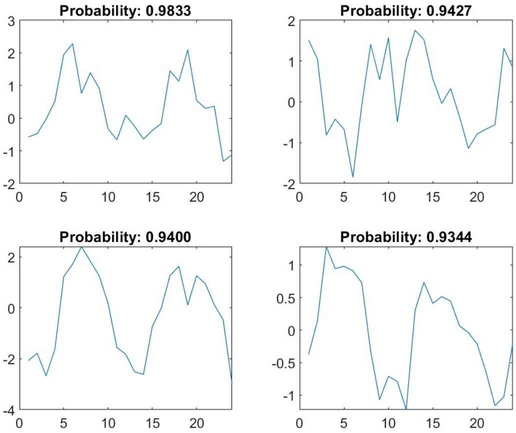


Figure 6: The most likely circadian genes.

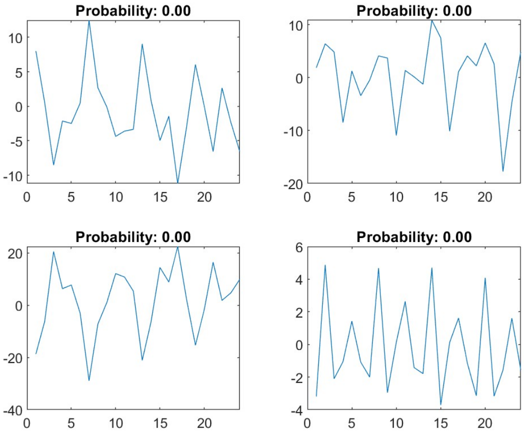


Figure 7: The least likely circadian genes.

In the figure below, we plotted the frequency of genes that have a certain probability of being circadian. There is a plot for the genes that were simulated as being circadian and those that were not. We see that the genes simulated as being circadian have a much higher frequency of genes with higher probabilities of being circadian. In particular, for the circadian genes, the frequency peak is at around 0.7 whereas for the non-circadian genes the peak is at 0 and no gene has a probability higher than approximately 0.25. We are very satisfied with these results because they imply that the model correctly identifies the genes that are actually circadian and doesn't assign circadianicity to genes that are not.

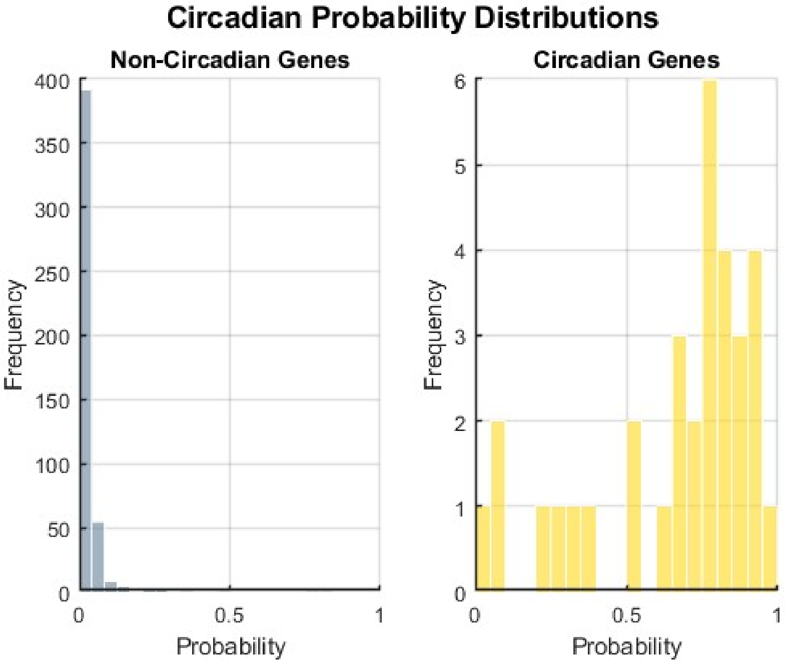


Figure 8: Frequency of circadian probability distributions.

Presented here is a table that compares the performance of our method with the performance of other methods found in the literature for identifying circadian genes. Apart from its performance in identifying genes in the top 1%, our model is on par with the other three models.

Method	Top 1%	Top 5%	Top 10%	Top 25%	Top 60%
DepLF	5	23	31	32	33
LS	20	25	27	30	31
ARSER	28	31	31	32	33
JTK-cycle	26	27	30	31	31

Table 2: Summary of rankings of the known clock genes in the synthetic data. Genes were ranked by estimated posterior circadian probability for the proposed approach (DepLF); by p-value for LS, ARSER, and JTK-cycle.

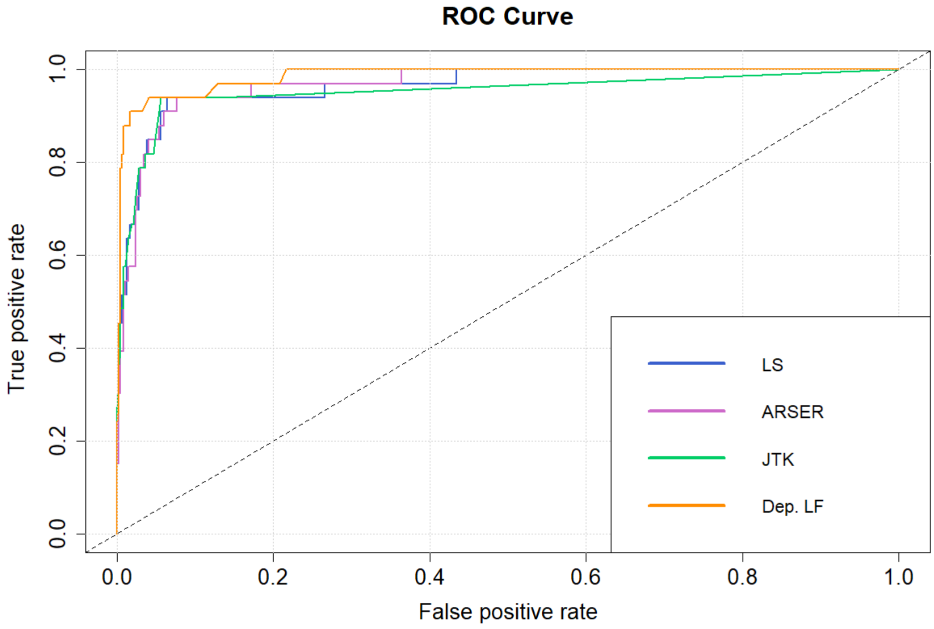


Figure 9: ROC curve for identifying circadian genes in the simulated example.

DepLF	LS	ARSER	JTK-cycle
0.9857	0.9648	0.9678	0.9529

Table 3: AUC values.

The ROC curve confirms an excellent performance of our model. In fact, the AUC is the highest of the four models considered.

8.3. Analysis with Real Data

We now apply our method to the Arabidopsis Thaliana dataset, which considers 22 810 genes over 13 time points. Below, we plot the estimated trajectories for six out of the twenty-six genes we know to be circadian.

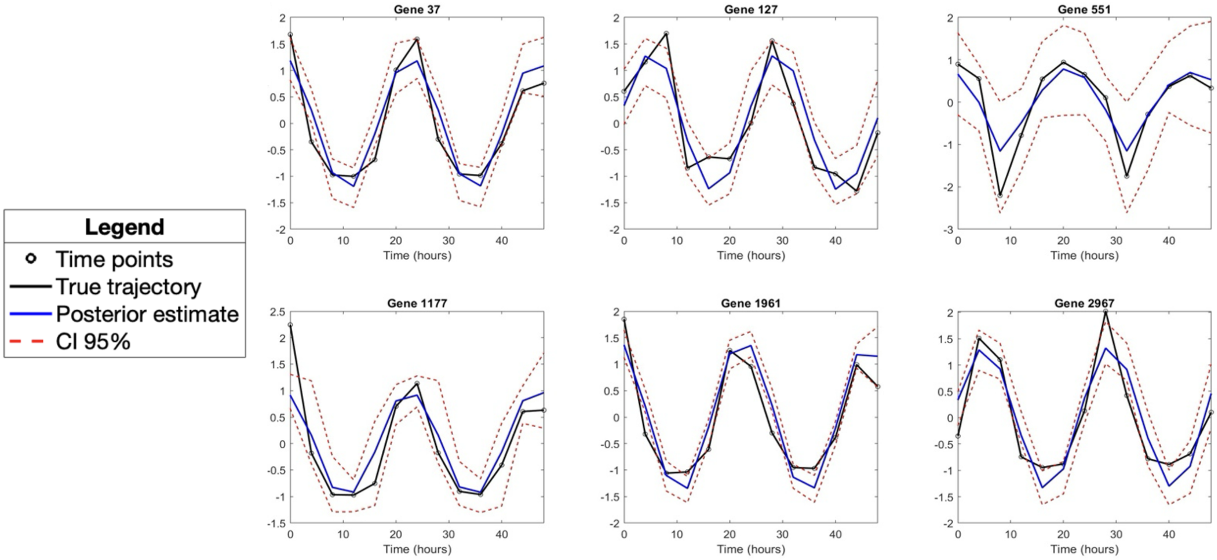
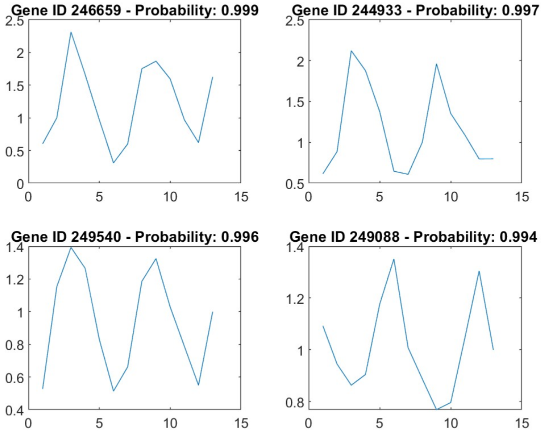


Figure 10: Estimated trajectories of six genes known as circadian from the Arabidopsis Thaliana dataset.

Once again, we plot the trajectories of genes with the highest and lowest probabilities of being circadian. We are satisfied to see that the genes with the highest probability of being circadian all have a probability higher than 93% and instead those with the lowest probability all have a probability of 0. In particular, the genes with the highest probability of being circadian all have a distinctive two-peak shape, whereas those with the lowest probability all exhibit random trajectories.



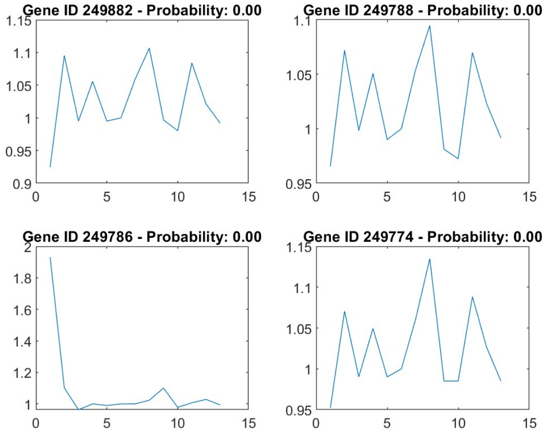


Figure 11: The most likely circadian genes.

Figure 12: The least likely circadian genes.

This table reports the probability of being circadian for all the 26 genes that we know are circadian a priori. The lower probability of certain genes can be attributed to the fact that certain genes which are known by biologists to be circadian do not exhibit sinusoidal behavior. Given how our model is built, the fact that they do not exhibit sinusoidal behavior leads to them being assigned a low probability of circadianicity.

Gene ID	P(Circ)	Gene ID	P(Circ)	Gene ID	P(Circ)
214929	0.9628	215926	0.6604	215677	0.3758
214933	0.9561	214570	0.6037	215861	0.2663
214912	0.9500	215401	0.5737	216270	0.2079
214993	0.9444	214525	0.5570	216132	0.1773
214992	0.8988	215466	0.5258	216142	0.1745
204996	0.8988	215500	0.5064	216148	0.1718
215082	0.8349	215532	0.4688	217013	0.0639
215139	0.7882	215592	0.4375	214797	0.0334
215150	0.7760	215629	0.4097		

Table 4: Posterior probabilities of being circadian for the 26 known genes, sorted in descending order.

This table compares the performance of our method with the performance of the same methods with which we compared performance on the synthetic data. Our model lags behind its counterparts when identifying circadian genes in the top 1% and top 5% but performs as well as the other methods when identifying circadian genes in the top 10, 25 and 60%.

Method	Top 1%	Top 5%	Top 10%	Top 25%	Top 60%
DepLF	1	5	12	21	25
LS	0	5	15	20	23
ARSER	19	21	23	25	26
JTK-cycle	17	20	20	20	22

Table 5: Summary of rankings of 26 known clock genes in the Arabidopsis Thaliana genome. Genes were ranked by estimated posterior circadian probability for the proposed approach (DepLF); by p-value for LS, ARSER, and JTK-cycle.

8.4. Convergence Analysis

To assess the convergence of the proposed method, we compute the trace plot of the $\boldsymbol{\Theta}\mathbf{b}_j + \boldsymbol{\Lambda}\boldsymbol{\eta}_j$ vector for a fixed time point t_j. Our model has very few identifiable parameters (a consequence of using the MGPSP prior), and thus this is the only vector we can use to check for convergence of the Gibbs Sampler, which is indeed achieved.

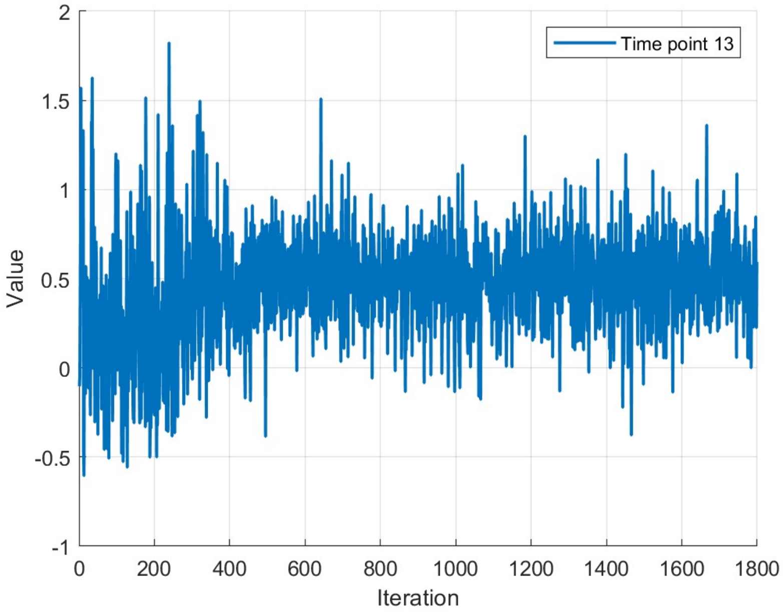


Figure 13: Trace plot of $\mathbf{\Theta b}_j + \mathbf{\Lambda}\boldsymbol{\eta}_j$.

The trace plot is generated after having discarded the burn-in iterations and having applied thinning. The chain appears to be stationary, with values fluctuating around a constant mean, suggesting that the algorithm has reached convergence. There are no obvious increasing or decreasing trends, which is a good sign for convergence. Finally, the variance of the samples appears relatively stable over time.

9. Conclusions

In conclusion, we believe that this project was completed successfully. The Gibbs Sampler was translated from MATLAB to C++ and the algorithm in C++ achieves much better performances than its counterpart in MATLAB. When it is applied to synthetic data or a subset of 5000 genes, our code achieves a tenfold speedup. In the case of the entire Arabidopsis dataset, MATLAB was unable to run that code and in C++ it takes just under 2 hours.

We are also relatively satisfied with the model performance. It performs particularly well on synthetic data compared to other models for detecting circadian genes. Our model has the highest AUC, successfully identifying 31 out of 33 circadian genes with an associated probability higher than 90% (refer to the Top 10% in Table 2). In the case of real genetic data, our model slightly underperforms compared to the competitors, being able to identify a smaller number of circadian genes. This discrepancy is more significant for the Top 1% and Top 5%, while for the other thresholds, our model's performance aligns closely with that of the other models (see Table 5).

In our approach, we considered a slightly simplified version of the model proposed by Montagna et al. [1], specifically excluding a term that accounts for local deviations from the underlying periodic oscillation. We believe that this minor modification contributes to our model's slight underperformance on real data. The natural extension of this project would be to include this term back into the model, allowing for a more accurate comparison with existing methods in the literature.

10. Bibliography and citations

References

[1] Montagna, S., Irincheeva, I., Tokdar, S. T. (2018), *High-dimensional Bayesian Fourier analysis for detecting circadian gene expressions.*, ArXiv.

[2] Hughes, M. E., Hogenesch, J. B. and Kornacker, K. (2010), *JTK CYCLE: an efficient nonparametric algorithm for detecting rhythmic components in genome-scale data sets*, Journal of Biological Rhythms 25(5), 372–380.

[3] Jouffe, C. and Cretenet, G. and Symu, L. and Martin, E. and Atger, F. and Naef, F. and Gachon, F. (2013), *The circadian clock coordinates ribosome biogenesis*, PLoS Science 11.

[4] Wichert, S., Fokianos, K. and Strimmer, K. (2004), *Identifying periodically expressed transcripts in microarray time series data*, Bioinformatics 20(1), 5–20.

[5] Edwards, K. D., Anderson, P. E., Hall, A., Salathia, N. S., Locke, J. C. W., Lynn, J. R., Straume, M., Smith, J. Q. and Millar, A. J. (2006), *Flowering locus C mediates natural variation in the high-temperature response of the Arabidopsis circadian clock*, The Plant Cell 18, 639–650.

[6] Anderson, P. E., Smith, J. Q., Edwards, K. D. and Millar, A. J. (2006), *Guided conjugate bayesian clustering for uncovering rhythmically expressed genes*, Working paper.
`http://wrap.warwick.ac.uk/35566/`

[7] Liverani, S. (2009), *Bayesian clustering of curves and the search of the partition space*, PhD thesis, University of Warwick, UK.

[8] Chudova, D., Ihler, A., Lin, K. K., Andersen, B. and Smyth, P. (2009), *Bayesian detection of nonsinusoidal periodic patterns in circadian expression data*, Bioinformatics 25(23), 3114–3120.

[9] Bhattacharya, A. and Dunson, D. B. (2011), *Sparse Bayesian infinite factor models*, Biometrika 98(2), 291–306.

[10] Nakajima, J. and West, M. (2013), *Bayesian analysis of latent threshold dynamic models*, Journal of Business and Economic Statistics 31, 151–164.

[11] Dodd, A. N., Gardner, M. J., Hotta, C. T., Hubbard, K. E., Dalchau, N., Love, J., Assie, J.-M., Robertson, F. C., Jakobsen, M. K., Goncalves, J., Sanders, D., Webb, A. A. R. (2007), *The Arabidopsis circadian clock incorporates a cADPR-based feedback loop*, Science. 318, 1789-1792.

[12] Müller, P., Parmigiani, G., Robert, C. and Rousseau, J. (2004), *Optimal sample size for multiple testing: the case of gene expression microarrays*, Journal of the American Statistical Association 99, 990–1001.

[13] Sanderson, C., Curtin, R. (2025), *Armadillo: An Efficient Framework for Numerical Linear Algebra.*, arXiv:2502.03000.

[14] Sanderson, C., Curtin, R. (2019), *Practical Sparse Matrices in C++ with Hybrid Storage and Template-Based Expression Optimisation.*, Mathematical and Computational Applications, Vol.24, No.3.

[15] Glynn, E.F., Chen, J., Mushegian, A.R. (2006), *Detecting periodic patterns in unevenly spaced gene expression time series using Lomb–Scargle periodograms*, Bioinformatics, Vol. 22, 3, 310–316.

[16] Yang, R., Su, Z. (2010), *Analyzing circadian expression data by harmonic regression based on autoregressive spectral estimation.* Bioinformatics, 26(12).

[17] Auerbach, B.J., FitzGerald, G.A., Li, M. (2022), *Tempo: an unsupervised Bayesian algorithm for circadian phase inference in single-cell transcriptomics.* Nat Commun 13, 6580.

[18] Newton, M. A., Noueiry, A., Sarkar, D. and Ahlquist, P. (2004), *Detecting differential gene expression with a semiparametric hierarchical mixture method*, Biostatistics 5(2), 155–176.