

REFERENCE DOCUMENT FOR SIMULATING MAMMALIAN CIRCADIAN CLOCKS WITH A NETWORK OF PHASE-ONLY OSCILLATORS

STEPHANIE R TAYLOR

INTRODUCTION

This guide is intended to be used as a reference for completing projects in CS336. We will be simulating the mammalian circadian clocks as a set of heterogeneous oscillators that send signals to each other to make small adjustments and, if the set-up is correct, synchronize their rhythms. We will be using this as a platform to (a) see the benefit of parallel computing in the context of scientific research, and (b) determine what constitutes a “correct” set-up. For example, is there such a thing as too much signaling? Too little? If so, what are the bounds?

CIRCADIAN CLOCK

The Center for Circadian Biology at UCSD has some helpful materials for learning about circadian clocks

- Read the “Basics of Chronobiology” (you can skip Box 1 and Box 2 if you want, and skim over the part about temperature compensation):
<http://ccb.ucsd.edu/the-bioclock-studio/education-resources/basics/index.html>
- Read “A Basic Sketch of Endogenous, Self-Sustained Oscillators”
<http://ccb.ucsd.edu/the-bioclock-studio/education-resources/basics/part2.html>
skipping section 2.2 (about data collection).

In this class, we are going to be using a simple model of the circadian clock in mammals. The clock in mammals is actually thousands of clocks that communicate with each other and synchronize their rhythms. They reside in neurons in the suprachiasmatic nucleus (SCN).

For a review on the mammalian circadian clock, read “Coordination of circadian timing in mammals” (Reppert & Weaver, *Nature*, 2002), which is in Courses/CS336/CourseMaterials. You can skim the parts describing the transcriptional feedback loops. Instead of modeling the transcriptional feedback loops directly, we will be simply modeling the “hand” of the clock.

CIRCADIAN TERMINOLOGY

- **Oscillator:** Something (e.g. a set of proteins in a cell or a set of mathematical equations) whose values regularly repeat. We sometimes refer to the circadian clock as an oscillator and sometimes as a multi-oscillator. The mammalian circadian clock is composed of thousand of oscillators that coordinate their rhythms so that altogether, they act as one big oscillator.
- **Phase:** Refers to where in the cycle the clock is. You can think of this as the angle of the hand of a clock. In the circadian world, phase is measured in 3 units:
 - hours. This unit is a natural one for a 24-hour clock.
 - radians. This unit is used to relate the clock's hand to the most common unit used for angles.
 - circadian hours (1 circadian hour is $1/24^{th}$ of a cycle). This unit it used so that we can compare phases across oscillators with different periods.
- **Circadian Time (CT).** Circadian Time is a convention used for indicating the phase of a circadian oscillation. It was developed in the context of animal behavior – mice become active around dusk, which is usually labeled CT12.
 - CT0 refers to the clock's time that usually coincides with dawn.
 - CT6 refers to the clock's time that usually coincides with noon.
 - CT12 refers to the clock's time that usually coincides with dusk.
 - CT18 refers to the clock's time that usually coincides with midnight.
- **Free-Running Period (FRP).** This is the intrinsic period of a clock. We call it “free-running” because it describes the period of oscillation when the clock is allow to run free of any inputs.
- **Phase Response Curve (PRC).** The phase response curve maps the effect of a stimulus (such as a signal sent from another cell or light coming in through the eye) on the phase of the clock. In the circadian biology literature, the PRC is usually a curve that has been experimentally measured (e.g. it will have a point on it showing that if a 1-hour pulse of light was given to a mouse at CT12, then the mouse would skip ahead in its rhythm).
- **Velocity Response Curve (VRC).** The VRC maps the effect of a stimulus on the *velocity of the* phase of the clock. No one has experimentally measured a VRC. Often, VRC shapes are modeled after PRC shapes and are used in mathematical models. In some of the models, they refer to what Stephanie calls a VRC as PRC.
- **Vasoactive Intestinal Peptide (VIP).** Neurons in the mammalian clock use VIP to communicate. It is not the only means of communication, but for the sake of our model, we will be using VIP as a stand-in for all peptides. When VIP is sent to a cell, it affects the clock's phase velocity.
- **Light/Dark Cycle (LD).** In the natural environment, there are a 24-hour light/dark cycle. Circadian clocks entrain (adapt) to that cycle, so understanding the clock means understanding how it responds to the light/dark cycle.
- **Constant Conditions (DD).** But we also want to understand how the clock works when it does not have any input from the environment. For nocturnal animals, we

typically study them in constant darkness. It is called DD, to contrast with LD (dark/dark versus light/dark).

DIFFERENTIAL EQUATION MODEL

The clock model tracks the phase of each cell over time. It is based on a classic model by Arthur T Winfree (Winfree, 1967; Winfree 2001) that captures the the phase of coupled oscillators. We adapt the model a small amount here, to change the units of the phase variable and to give a name to the signaling agent – we call it VIP. Each cell produces the signaling agent VIP, with the amount depending on its phase. This VIP is sent to all other cells.

We use differential equations to keep track of the phase ϕ_i of each oscillator. The units of ϕ are hours, so I usually refer to this model as simulating in “real time”. The units of the independent variable t are also hours, so when an oscillator is oscillating unperturbed, its phase increases at a rate of 1 (hour per hour). If a cell receives a signal, that affects its phase velocity. Depending upon the phase, the signal may slow it down or speed it up. That phase-dependent response is captured in the velocity response curve (VRC). The differential equation for oscillator i then tracks how the phase progresses (basal rate of 1 plus the input times its phase-dependent response). The system together is a series of n differential equations:

$$\begin{aligned}\frac{d\phi_1}{dt} &= 1 + v_1(\mathbf{A}, \vec{\sigma}) \text{VRC}(\phi_1) \\ \frac{d\phi_2}{dt} &= 1 + v_2(\mathbf{A}, \vec{\sigma}) \text{VRC}(\phi_2) \\ &\vdots \\ \frac{d\phi_n}{dt} &= 1 + v_n(\mathbf{A}, \vec{\sigma}) \text{VRC}(\phi_n)\end{aligned}$$

where v_i is the VIP input for the i^{th} cell and the VRC is a cosine curve. The phase ϕ_i is measured in hours, but must be converted to radians to call the cosine curve, so we have

$$\text{VRC}(\phi_i) = \cos\left(\phi_i \frac{2\pi}{\tau_i}\right)$$

where τ_i is the intrinsic (free-running) period of the i^{th} oscillator.

The VIP input v_i for the i^{th} cell depends on the VIP output of all n oscillators $\vec{\sigma}$. We compute the average amount of VIP sent from each of the cells. That means, we add up the VIP inputs and then divide by the number of cells that send VIP to cells i . Below, I

write the equation for each cell i , but notice that it is the same for every cell, so you should just compute it once. The equation is

$$v_i(\vec{o}) = \frac{1}{n} \sum_j^n o_j$$

The output o_j of the j^{th} cell depends on its current phase ϕ_j . VIP is released between CT4 and CT8. Since ϕ_j is measure in “real time”, we must translate it to circadian time, by dividing by the intrinsic period and multiplying by 24. Also, since the phase increases without bounds during the simulation, we need to use modular arithmetic to figure out how many hours into its current cycle it is.

$$o_j = \begin{cases} s, & \text{if } 4 \leq \text{mod}(\phi_j, \tau_j) \frac{24}{\tau_j} \leq 8 \\ 0, & \text{otherwise} \end{cases}$$

where s is the strength of the VIP release.

The vector of τ values and the VIP strength s are parameters (constants) for this version of model.

Version of Model with Specific Network Topology. Instead of assuming that VIP coupling is all-to-all or “mean field”, it is more realistic to model a specific communication network topology. We indicate which cells send VIP to which other cells using a matrix $\mathbf{A}$. If there are n cells, then we have an $n \times n$ matrix. The entry $A_{i,j}$ (i^{th} row, j^{th} column) will be 1 if cell j sends VIP to cell i and 0 if it doesn’t.

For this version, the VIP input v_i for the i^{th} cell depends on the i^{th} row of the connectivity matrix $\mathbf{A}$ and the VIP output of all n oscillators $\vec{o}$. We compute the average amount of VIP sent from each of the cells to cell i . That means, we add up the VIP inputs and then divide by the number of cells that send VIP to cells i . The equation is

$$v_i(\mathbf{A}, \vec{o}) = (\mathbf{A}_{i*} \cdot \vec{o}) / \text{nnz}(\mathbf{A}_{i*})$$

where nnz computes the “number of non-zeros” and o_j is defined as in the original version of the model.

The vector of τ values, the VIP strength s , and the connectivity matrix $\mathbf{A}$ are parameters (constants) for this version of model.

HOW TO SIMULATE THE MODEL – FORWARD EULER

Simulating the model is the equivalent to finding the values of $\phi(t)$. To find those values, we need to solve the equations using numerical approximation. To do that, we start with an *initial condition* (the values of the phases of all the oscillators at the initial time, usually $t = 0$ h) and then numerically integrate the system at specified time points. For example, we typically begin at $t = 0$ and find the values of each phase at times that are 6 minutes ($1/10^{th}$ of an hour) apart, ending after several cycles (often 6 to 10 days).

There are many methods to accomplish this numerical integration. We are going to use the simplest - the “Forward Euler” method. It uses the basic intuition that if you know where you are (what ϕ is) and how fast you are moving (what $\frac{d\phi}{dt}$ is), then you can calculate where you will be a short while (Δt h) later:

$$\phi(t + \Delta t) = \phi(t) + \Delta t \frac{d\phi}{dt}$$

We compute the values of the phase for each cell at each time step:

```

for tidx = 1 : NT
    phaseVelocity = phaseEq( phases[tidx*NX:(tidx+1)*NX-1], parameters )
    for cidx = 1 : NX
        phases[(tidx+1)*NX+cidx] = phases[(tidx+1)*NX+cidx] + deltaT
            * phaseVelocity[cidx]
    
```

where phaseEq is a function that evaluates the right-hand side of the differential equations for the phases. It takes in the phases for the current time step (index tidx) and the parameters, and returns an array of phase velocities. (Note: This is pseudo-code - the C function will also need to take in NX, so that it knows how long the phase array is).

The result of the simulation is a *timeseries* - a series of values over time.

For more information, see Wikipedia’s entry on the Euler Method: https://en.wikipedia.org/wiki/Euler_method

DETERMINING SYNCHRONY

We will say that a set of oscillators is synchronized if they are “close enough” in period and phase. Below, I describe two methods for describing synchrony. The first makes it easy to detect both period-synchrony and phase-synchrony, but is not a standard measure. The second is a standard measure, but it measures only phase-synchrony. In our projects, we will implement the first method. The second is an extension.

- **Event Detection:** A simple method for determining how synchronized oscillators are is to identify an event that will happen once per cycle. Then track which time that event happens for every oscillator. If the times are all close to each other, then

the oscillators are likely synchronized. If the times are far apart, the the oscillators are not. This is the method we will use for the projects and the event we will track is the moment the phase crosses CT6 “on the way up”. In other words, if the phase of an oscillator is CT5.96 at the time step with index 125 and is CT6.12 at the time step with index 126, then we say the event happened at time step 126. We will detect the event for every oscillator for a given number of cycles. Then, for each cycle, compute the standard deviation of the event times. If we compute that standard deviation in units of hours, then we are looking for a standard deviation of less than 0.1 to indicate the oscillators are fully synchronized.

- Phase Coherence: A common measure of phase coherence is the radius of the *complex order parameter* (Kuramoto, 1984; Strogatz, 2000). For this we imagine all the phases are points on the unit circle. There is an arrow pointing from the center towards the mean location of the points on the circle. The arrow is longer if those points are closer together. Using phases in units of radians we compute the right-hand-side of

$$r e^{i\psi} = \frac{1}{n} \sum_j^n e^{i\phi_j}$$

where i is the imaginary number $\sqrt{-1}$. After computing the right-hand side, we will have a complex number if we take the absolute value of that complex number, we will find the value of r (the length of the arrow). If r is close to 1, then the system is synchronized. If it is close to 0, then the system is not synchronized. (Note: this method will determine that oscillators are perfectly synchronized even if they aren’t oscillating!. Yikes. Suppose our signaling just makes everything halt, but halt at the same phase. They will still look synchronized. Or suppose the signaling does the unthinkable and makes the clocks run backwards. Again, they could look synchronized but be behaving quite badly. So we generally use this measure in concert with an analysis of the period of oscillation.)

REFERENCES

- Y Kuramoto, *Chemical Oscillations, Waves, and Turbulence*, Springer, New York, 1984.
- SH Strogatz, From Kuramoto to Crawford: exploring the onset of synchronization in populations of coupled oscillators, *Physica D*, 2000. Strogatz 2000
- AT Winfree, Biological rhythms and the behavior of populations of coupled oscillators. *J Theor Biol*, 1967.
- AT Winfree, *The Geometry of Biological Time*, Springer, New York, 2001.
- SM Reppert & DR Weaver, Coordination of circadian timing in mammals, *Nature*, 2002.