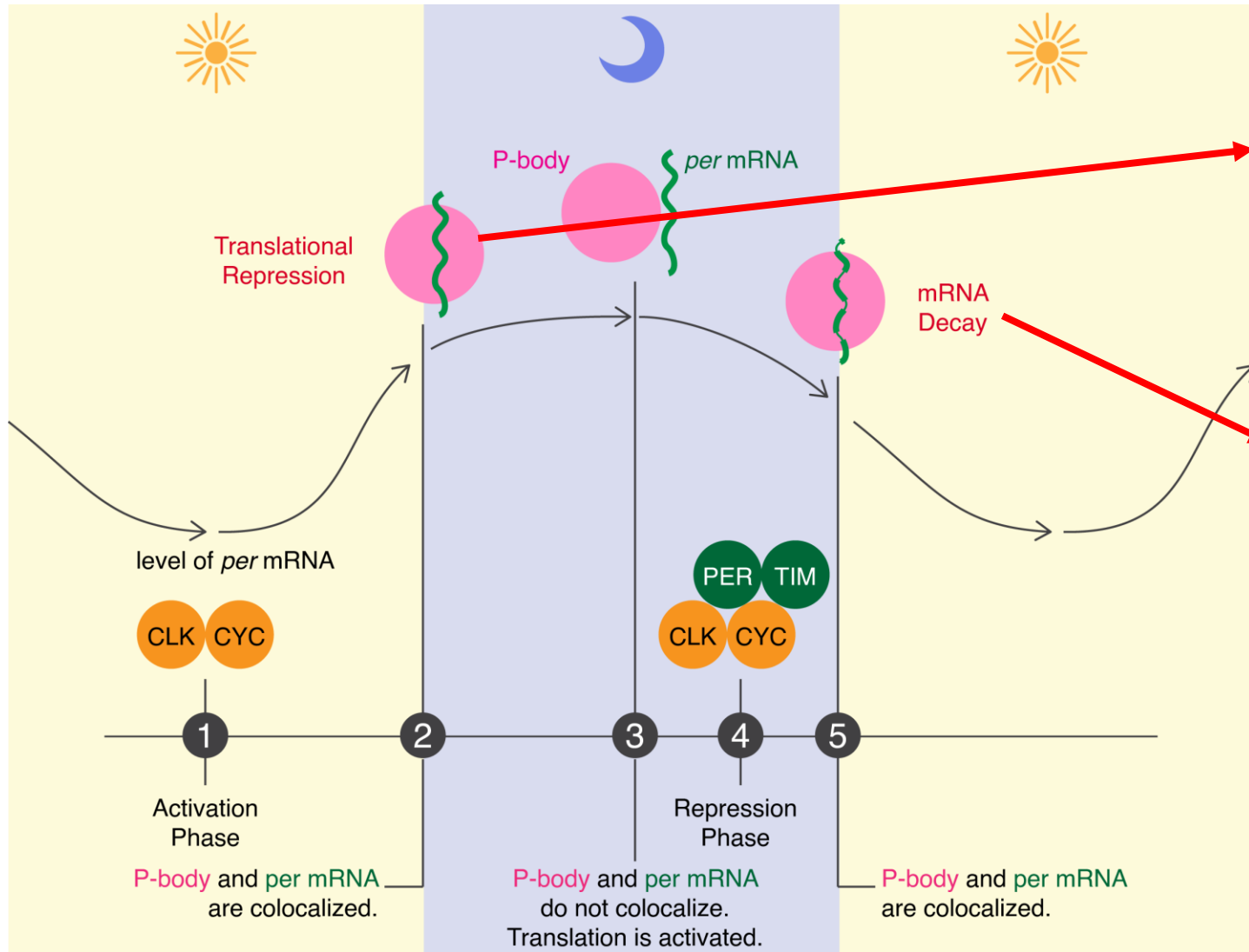
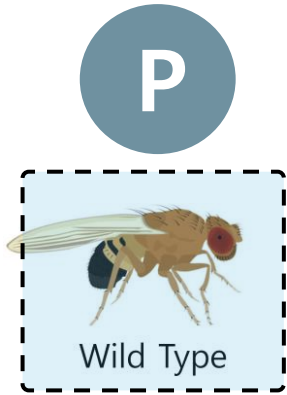


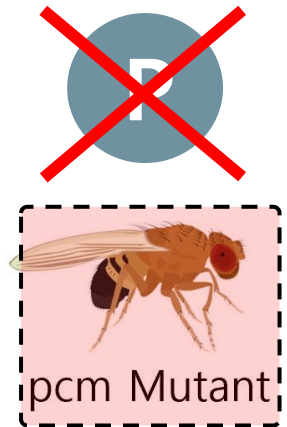
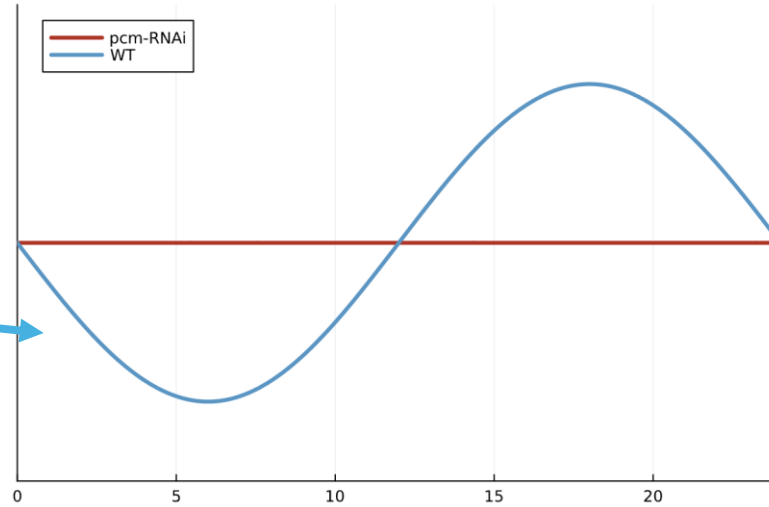
Review: We have two hypotheses about how P bodies control circadian clocks



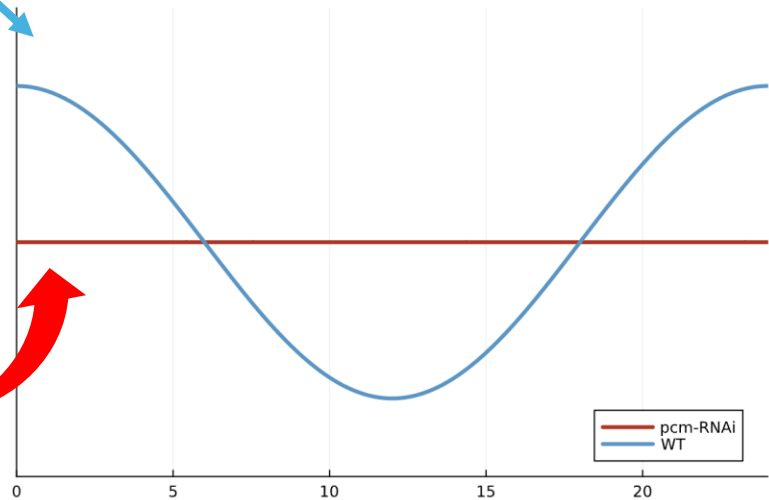
Through the estimated decay rate of per and translation rate, we can validate our hypothesis



The decay rate of per

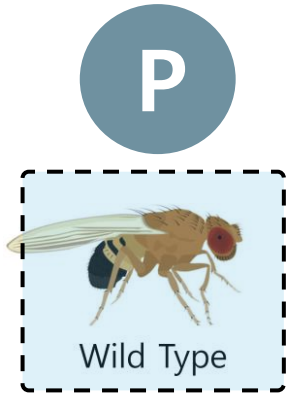


The translation rate of PER

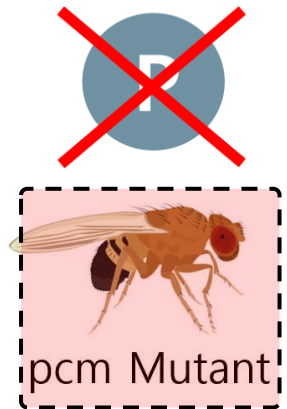
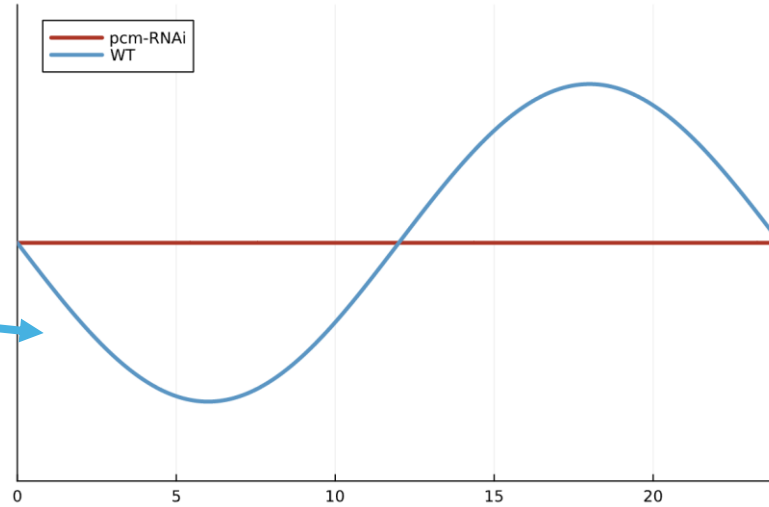


CT(h)

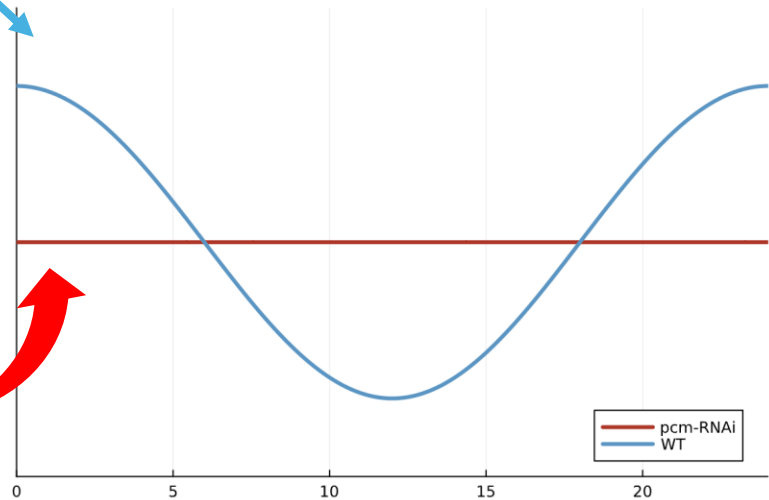
Through the estimated decay rate of per and translation rate, we can validate our hypothesis



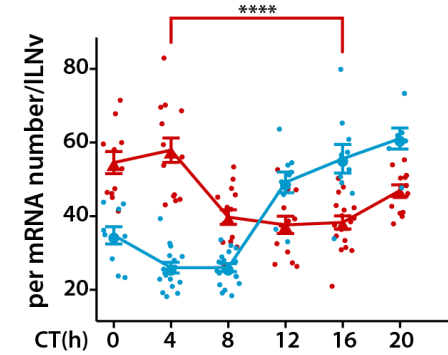
The decay rate of per



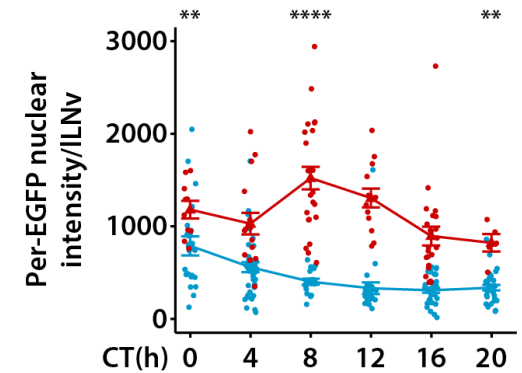
The translation rate of PER



CT(h)

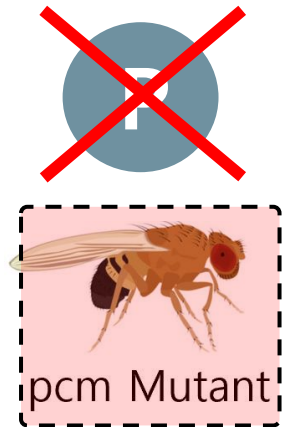
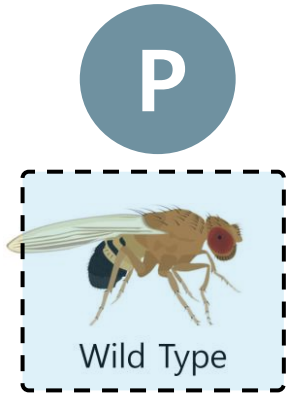


$$\text{rhythm}_D(t) = \left(\frac{\text{amp}_D}{2} \right) \cos\left(\frac{2\pi(t - \text{ph}_D)}{24} \right) + b_1$$



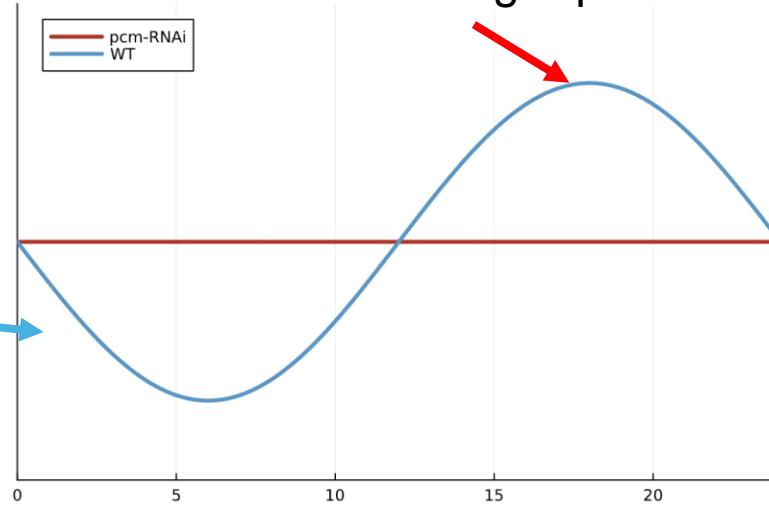
$$\text{rhythm}_{T_L}(t) = \left(\frac{\text{amp}_T}{2} \right) \cos\left(\frac{2\pi(t - \text{ph}_T)}{24} \right) + 1$$

Through the estimated decay rate of per and translation rate, we can validate our hypothesis



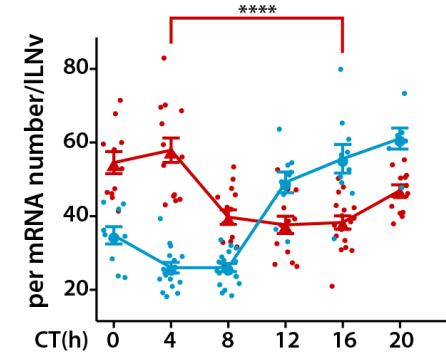
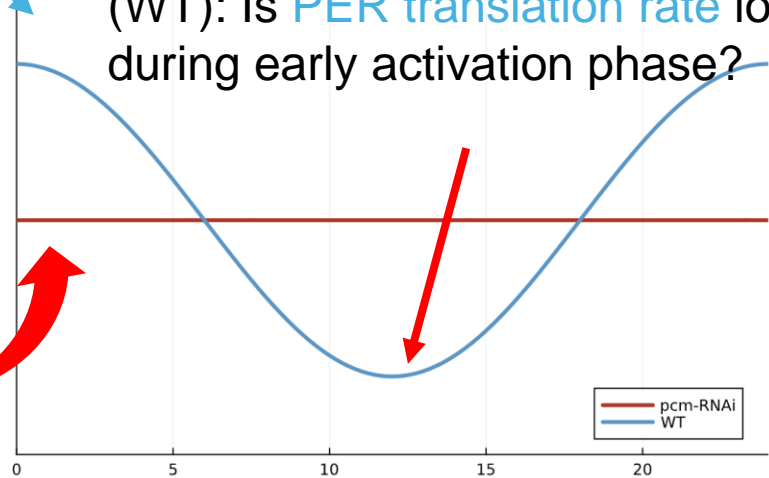
(WT): Is **per (mRNA)** decay rate maximized during repression phase?

The decay rate of per

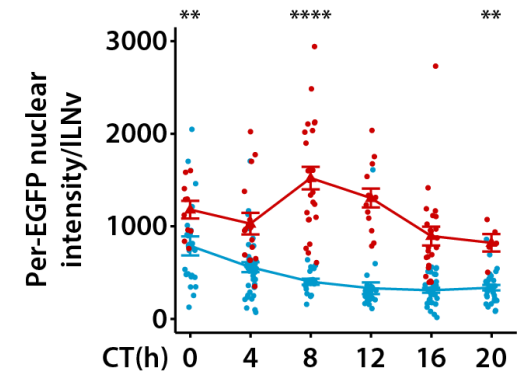


(WT): Is **PER translation rate** lowest during early activation phase?

The translation rate of PER



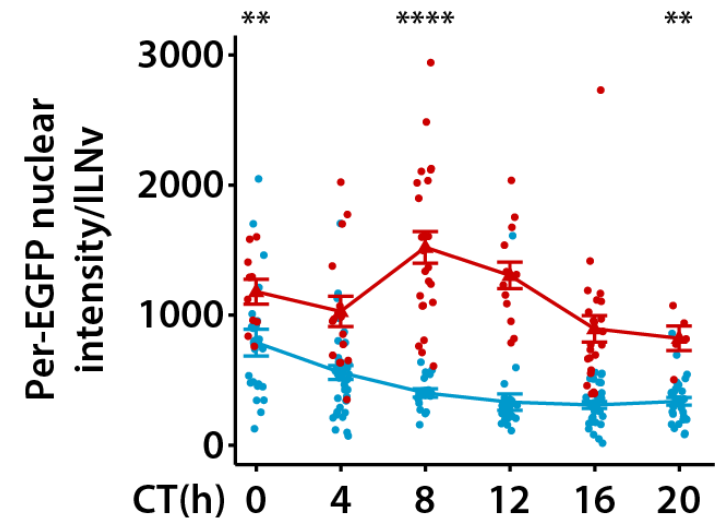
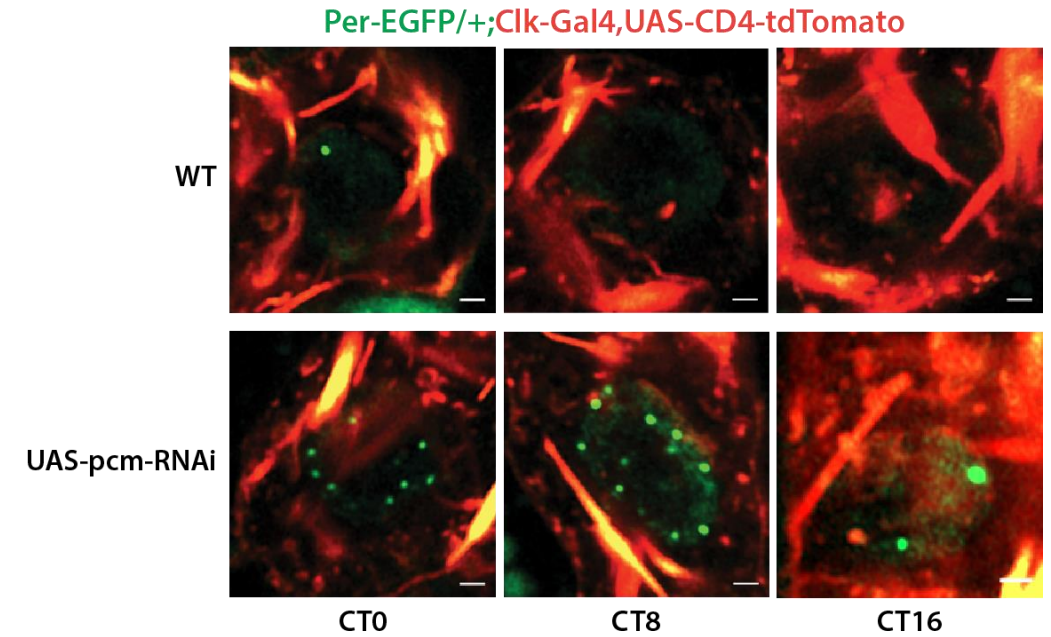
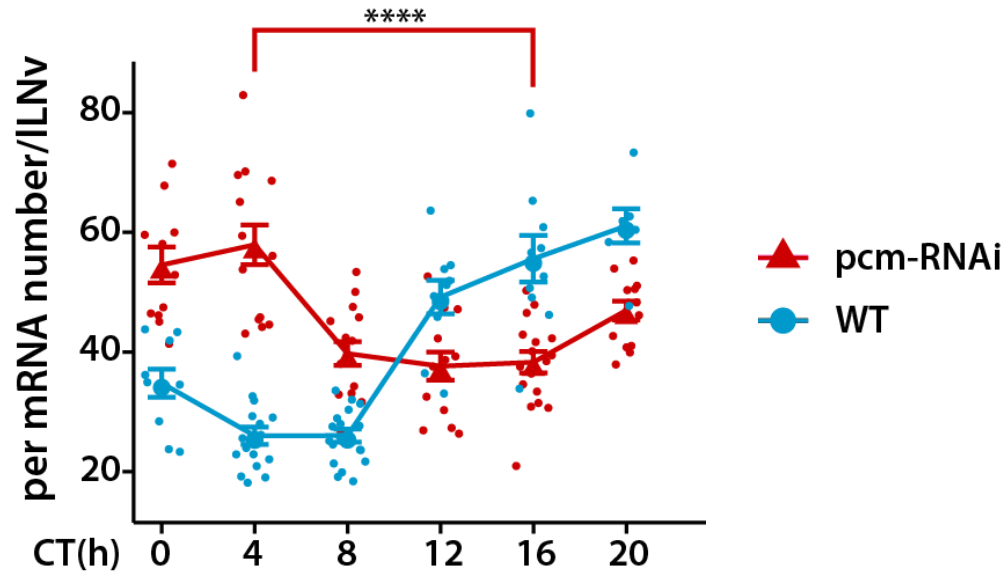
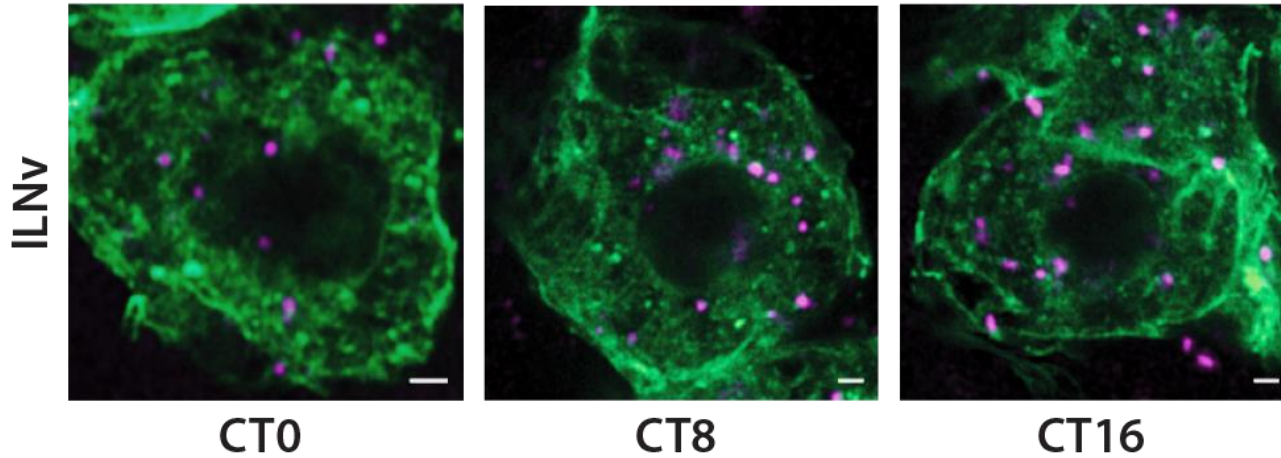
$$\text{rhythm}_D(t) = \left(\frac{\text{amp}_D}{2} \right) \cos\left(\frac{2\pi(t - \text{ph}_D)}{24} \right) + b_1$$



$$\text{rhythm}_{T_L}(t) = \left(\frac{\text{amp}_T}{2} \right) \cos\left(\frac{2\pi(t - \text{ph}_T)}{24} \right) + 1$$

We will fit models to four time-series (per mRNA & nucleic PER of WT & pcm KD)

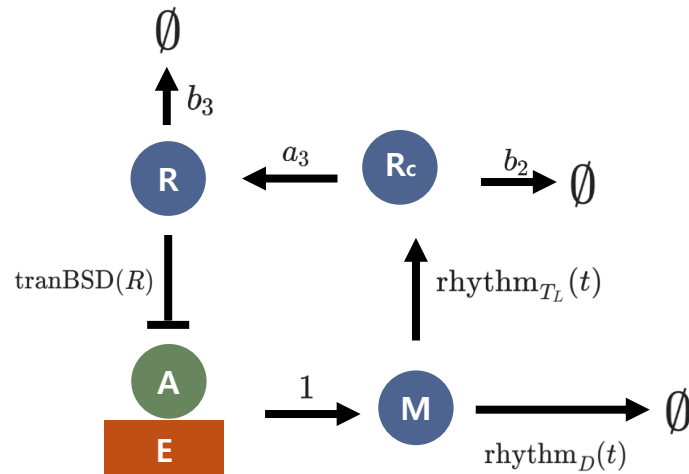
LD LD DD2 ;Clk-Gal4/+;UAS-mCD8-GFP/+ **per exon**



Two models (9,13 parameters each) will be fitted with time series data.

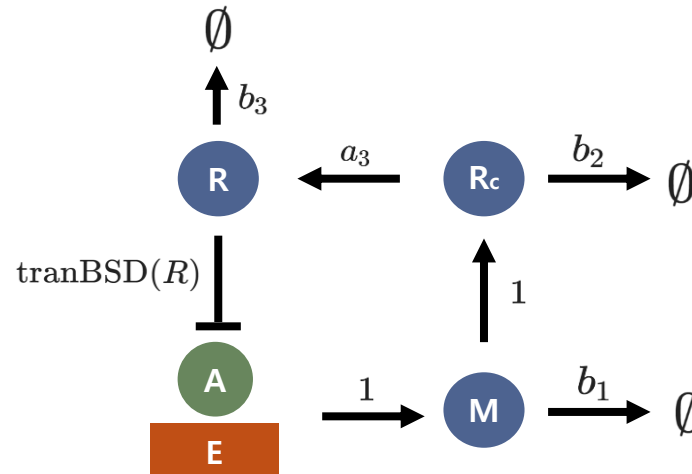
Wild Type

$$\begin{aligned}\dot{M} &= \text{tranBSD}(R) - \text{rhythm}_D(t, b_1) \cdot M \\ \dot{R}_c &= \text{rhythm}_{TL}(t, 1) \cdot M - b_2 \cdot R_c - a_3 \cdot R_c \\ \dot{R} &= a_3 \cdot R_c - b_3 \cdot R\end{aligned}$$

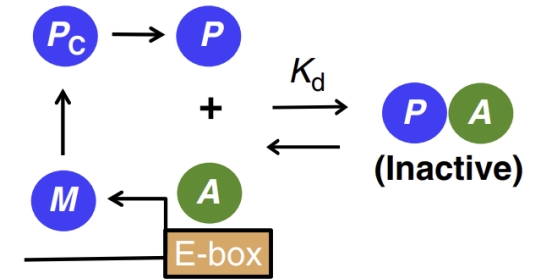


pcm-Mutant

$$\begin{aligned}\dot{M} &= \text{tranBSD}(R) - b_1 \cdot M \\ \dot{R}_c &= M - b_2 \cdot R_c - a_3 \cdot R_c \\ \dot{R} &= a_3 \cdot R_c - b_3 \cdot R\end{aligned}$$



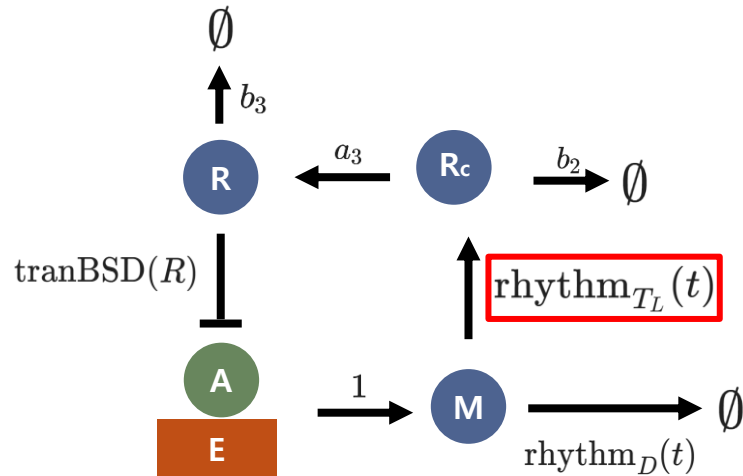
Kim-Forger Model (2012)



Two models (9,13 parameters each) will be fitted with time series data.

Wild Type

$$\begin{aligned}\dot{M} &= \text{tranBSD}(R) - \text{rhythm}_D(t, b_1) \cdot M \\ \dot{R}_c &= \text{rhythm}_{TL}(t, 1) \cdot M - b_2 \cdot R_c - a_3 \cdot R_c \\ \dot{R} &= a_3 \cdot R_c - b_3 \cdot R\end{aligned}$$

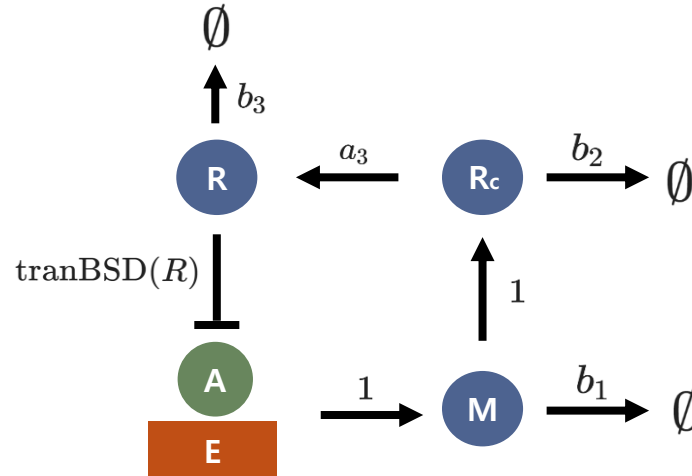


$(1, a_3, b_1, b_2, b_3, A_T, K_a, K_s, K_b, K_d, \text{amp}_D, \text{ph}_D, \text{amp}_T, \text{ph}_T)$

Total: 13 Parameters

pcm-Mutant

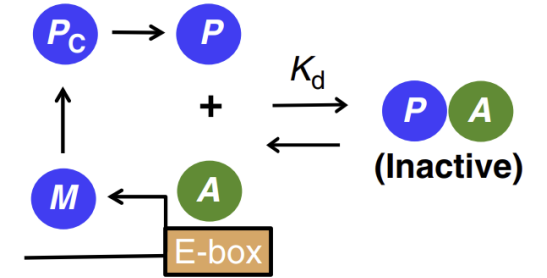
$$\begin{aligned}\dot{M} &= \text{tranBSD}(R) - b_1 \cdot M \\ \dot{R}_c &= M - b_2 \cdot R_c - a_3 \cdot R_c \\ \dot{R} &= a_3 \cdot R_c - b_3 \cdot R\end{aligned}$$



$(1, a_3, b_1, b_2, b_3, A_T, K_a, K_s, K_b, K_d)$

Total: 9 Parameters

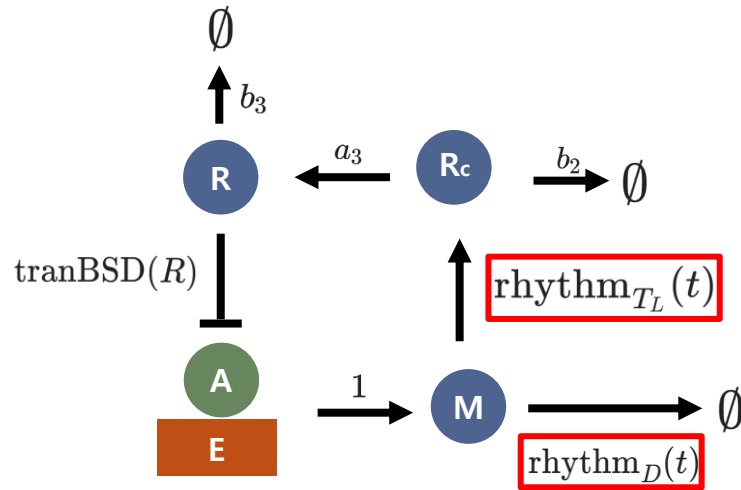
Kim-Forger Model (2012)



Two models (9,13 parameters each) will be fitted with time series data.

Wild Type

$$\begin{aligned}\dot{M} &= \text{tranBSD}(R) - \text{rhythm}_D(t, b_1) \cdot M \\ \dot{R}_c &= \text{rhythm}_{TL}(t, 1) \cdot M - b_2 \cdot R_c - a_3 \cdot R_c \\ \dot{R} &= a_3 \cdot R_c - b_3 \cdot R\end{aligned}$$

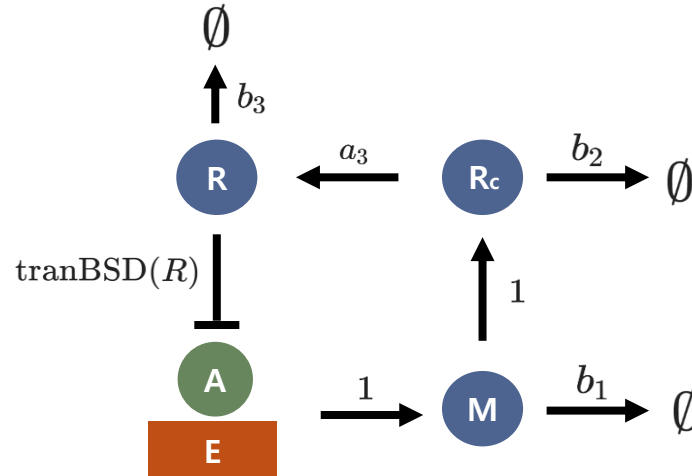


$(1, a_3, b_1, b_2, b_3, A_T, K_a, K_s, K_b, K_d, \text{amp}_D, \text{ph}_D, \text{amp}_T, \text{ph}_T)$

Total: 13 Parameters

pcm-Mutant

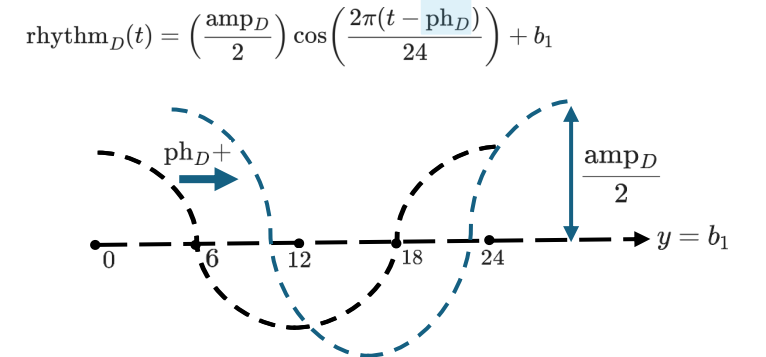
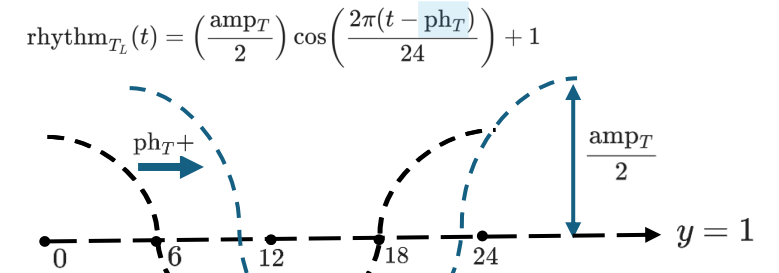
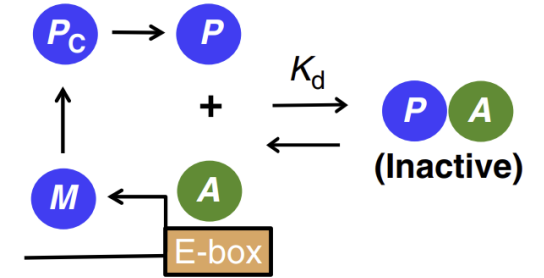
$$\begin{aligned}\dot{M} &= \text{tranBSD}(R) - b_1 \cdot M \\ \dot{R}_c &= M - b_2 \cdot R_c - a_3 \cdot R_c \\ \dot{R} &= a_3 \cdot R_c - b_3 \cdot R\end{aligned}$$



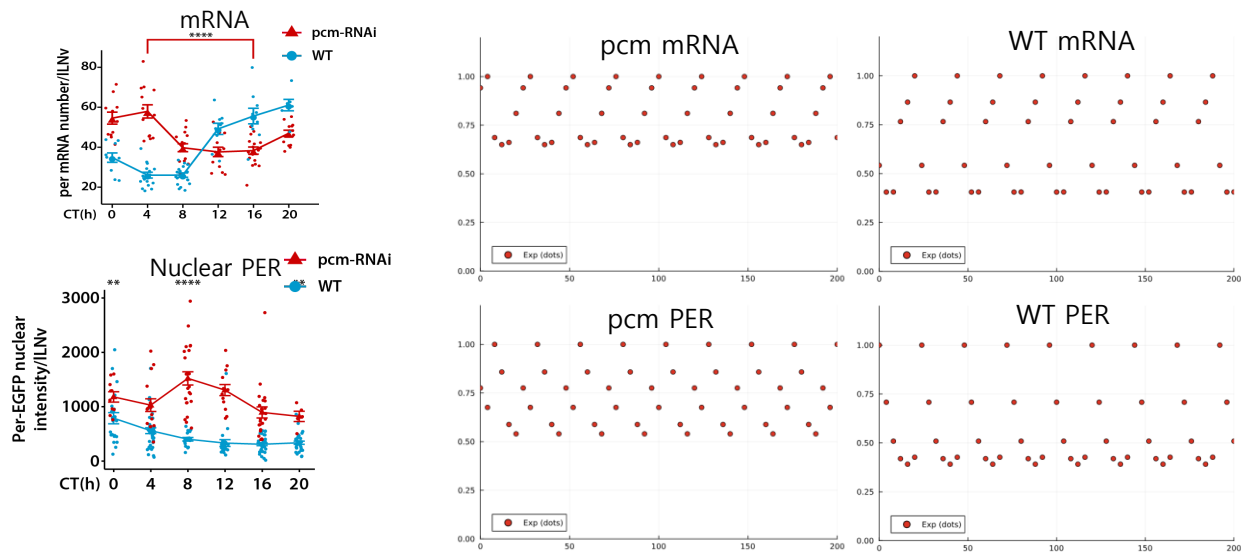
$(1, a_3, b_1, b_2, b_3, A_T, K_a, K_s, K_b, K_d)$

Total: 9 Parameters

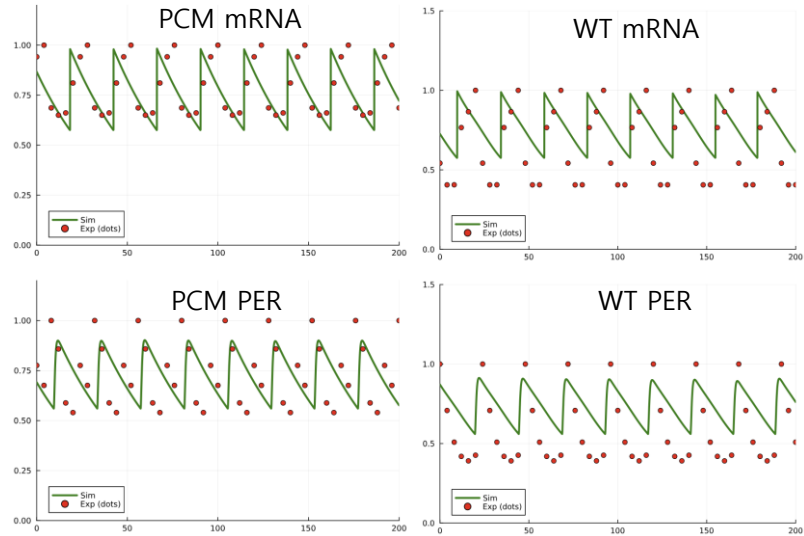
Kim-Forger Model (2012)



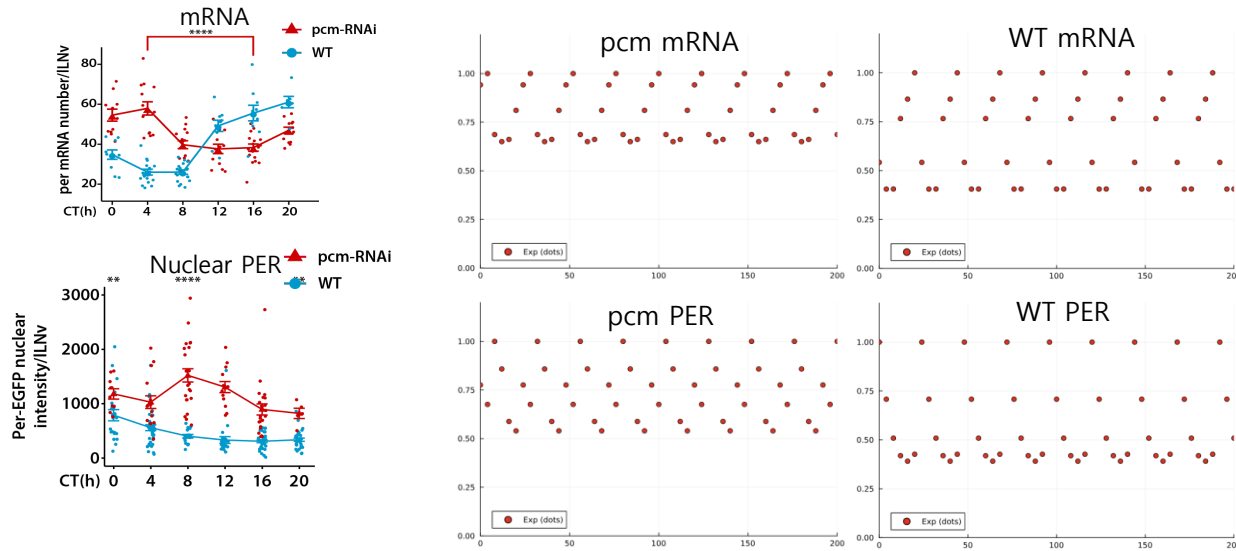
Simulated Annealing (SA) with appropriate cost function is needed to generate good model fit with stable, consistent oscillations in decay/translation rate for WT.



Simulated Annealing (SA)
+ Cost Function
→ Model Fitting



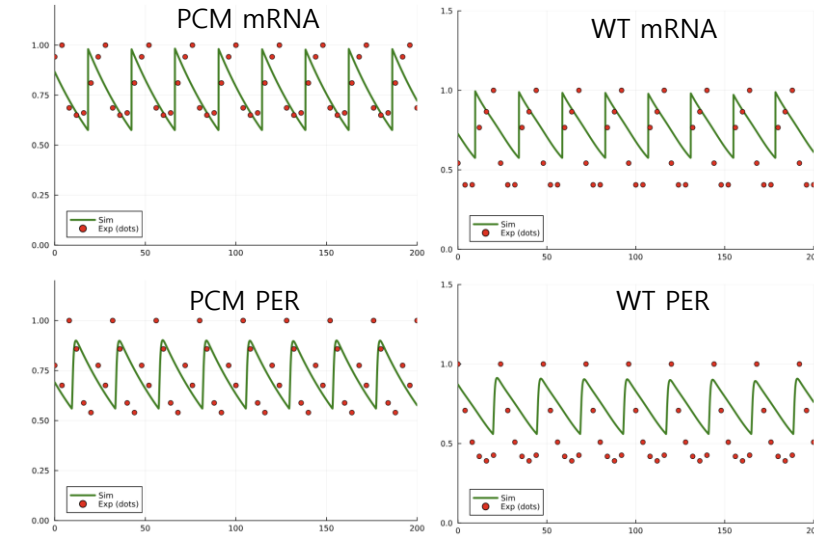
Simulated Annealing (SA) with appropriate cost function is needed to generate good model fit with stable, consistent oscillations in decay/translation rate for WT.



Simulated Annealing (SA)

+ Cost Function

→ Model Fitting



Simulated Annealing (SA)

```

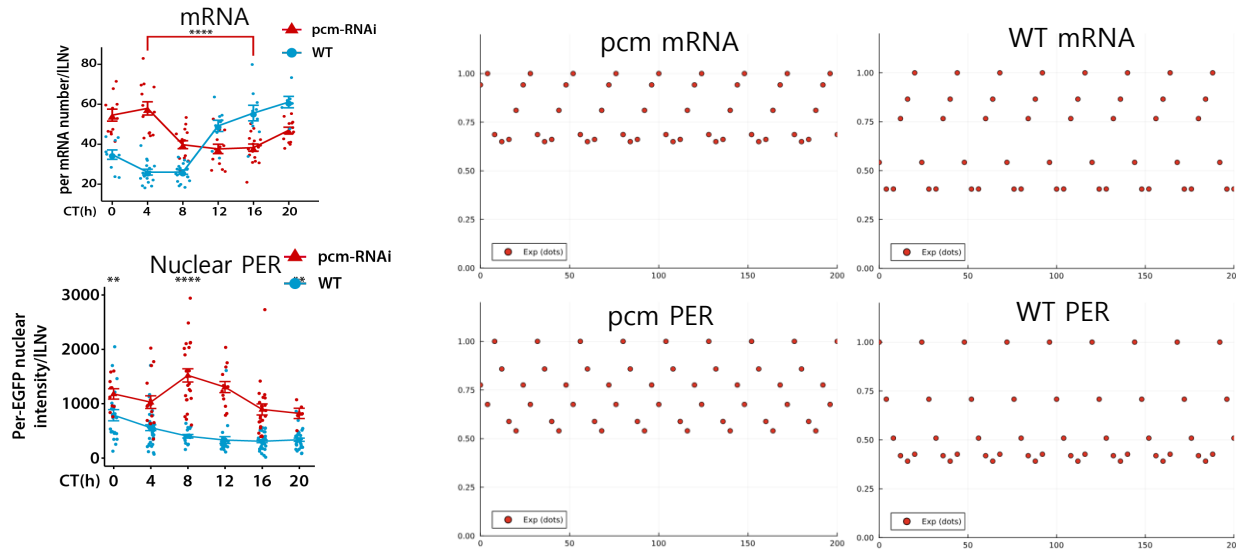
for i = 1 to max_iter do
  s_candidate ← perturb(s_current) // Generate new candidate state
  f_candidate ← f(s_candidate) // Evaluate candidate state
  Δf ← f_candidate - f_current
  if Δf < 0 or exp(-Δf/T) > rand() then
    s_current ← s_candidate
    f_current ← f_candidate
  end if
  if f_current < f_best then
    s_best ← s_current
    f_best ← f_current
  end if
  T ← T × α // Decrease temperature
  if i mod 5000 == 0 then
    Print progress: i, s_candidate, f_current, f_best, T
  end if
end for

```

Temperature,
Cooling Rate

Return s_best, f_best

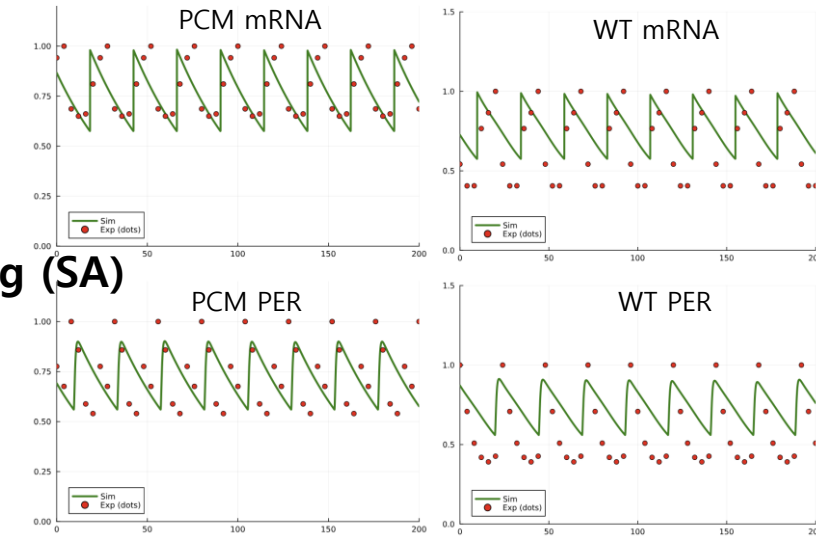
Simulated Annealing (SA) with appropriate cost function is needed to generate good model fit with stable, consistent oscillations in decay/translation rate for WT.



Simulated Annealing (SA)

+ Cost Function

→ Model Fitting



Simulated Annealing (SA)

```

for i = 1 to max_iter do
    s_candidate ← perturb(s_current) // Generate new candidate state
    f_candidate ← f(s_candidate) // Evaluate candidate state
    Δf ← f_candidate - f_current
    if Δf < 0 or exp(-Δf/T) > rand() then
        s_current ← s_candidate
        f_current ← f_candidate
    end if
    if f_current < f_best then
        s_best ← s_current
        f_best ← f_current
    end if
    T ← T × α // Decrease temperature
    if i mod 5000 == 0 then
        Print progress: i, s_candidate, f_current, f_best, T
    end if
end for

Return s_best, f_best
    
```

$$\text{Cost: } \sqrt{\left(1 - \frac{\text{period}}{24}\right)^2 + \left(1 - \frac{\text{rel amp } M}{\text{rel amp } mPer}\right)^2 + \left(1 - \frac{\text{relamp } R_n}{\text{rel amp } PER}\right)^2 + \left(1 - \frac{\text{peak(th) phase } R_n}{\text{peak(th) phase } PER}\right)^2 + \left(1 - \frac{\text{peak(th) phase } M}{\text{peak(th) phase } mPer}\right)^2}$$

Period ≈ 24h

$$\text{relamp} = \frac{\text{amp}[\text{end}]}{\text{peak}[\text{end}]} \approx 1 - \min(\text{pcmMper or pcmPER})$$

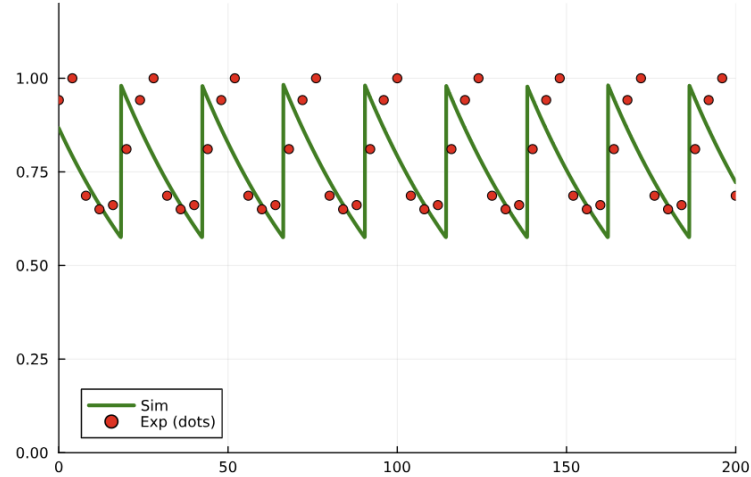
Stable Oscillations

$$\max\left(0, 5 \times \left(2 - \frac{\text{amp}[\text{end}]}{\text{amp}[\text{end} - 1]} - \frac{\text{amp}[\text{end} - 1]}{\text{amp}[\text{end} - 2]}\right)\right)$$

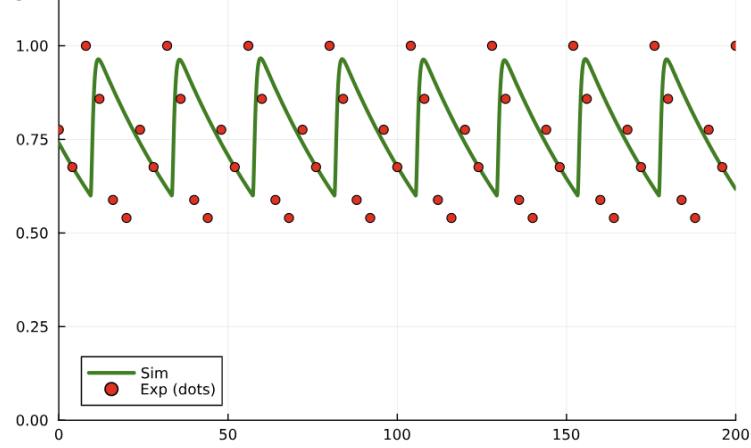


Pcm model ODE solution fits pcm mutant time trace data.

pcm mRNA



pcm PER



$$\text{cost} = \text{cost}_{amp}(\text{mRNAs}) + \text{cost}_{amp}(\text{proteins})$$

$$+ \sqrt{\underbrace{\left(1 - \frac{\text{relamp}(\text{mRNAs})}{1 - \min(\text{pcmMper})}\right)^2}_{\text{cost}_{relamp}(\text{mRNAs})} + \underbrace{\left(1 - \frac{\text{relamp}(\text{proteins})}{1 - \min(\text{pcmPER})}\right)^2}_{\text{cost}_{relamp}(\text{PER})} + \underbrace{\left(1 - \frac{\text{period}(\text{proteins})}{24}\right)^2}_{\text{cost}_{period}(\text{PER})}}$$

Best value: 0.22608254311121734

Cost components at best state:

ddmeasure(mRNAs)[1]: 0.0

ddmeasure(proteins)[1]: 0.0

$(1 - \text{ddmeasure}(\text{mRNAs})[5] / (1 - \text{minimum}(\text{pcmMper})))^2$: 0.013170807449025554

$(1 - \text{ddmeasure}(\text{proteins})[5] / (1 - \text{minimum}(\text{pcmPER})))^2$: 0.037941814406165446

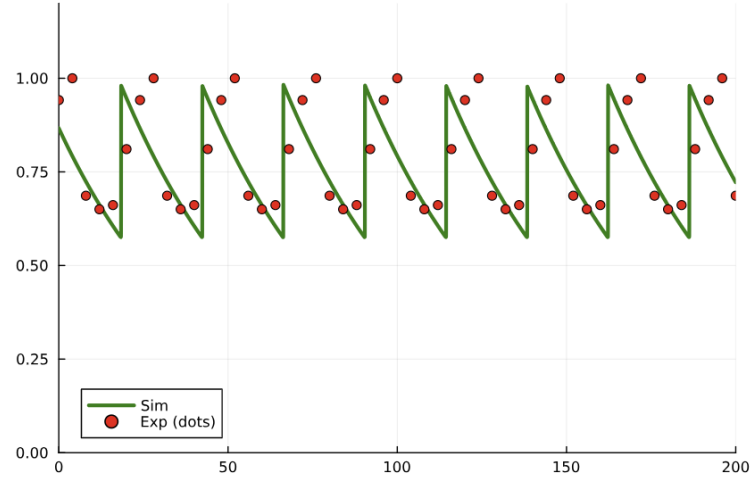
$(1 - \text{ddmeasure}(\text{proteins})[3] / 24)^2$: 6.944444444442915e-7

Iteration 45200: Best value = 0.22608254311121734, Temperature = 1.522264525982424e-10
 Iteration 45400: Best value = 0.22608254311121734, Temperature = 1.3773674571581714e-10
 Iteration 45600: Best value = 0.22608254311121734, Temperature = 1.2462624462814756e-10
 Iteration 45800: Best value = 0.22608254311121734, Temperature = 1.1276366934180646e-10

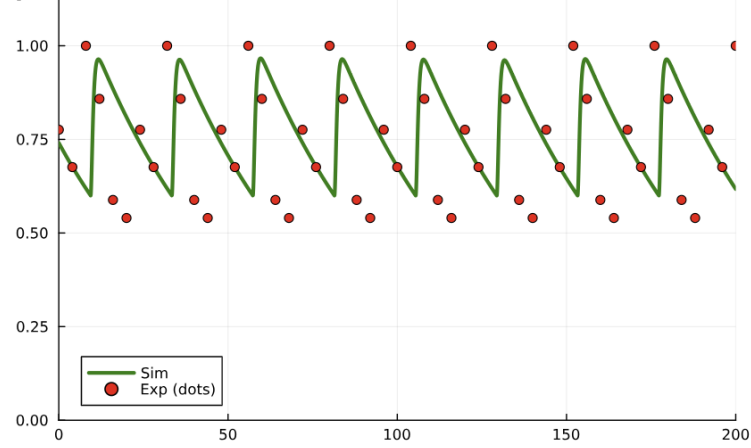
$$\max\left(0, 5 \times \left(2 - \frac{\text{amp}[\text{end}]}{\text{amp}[\text{end} - 1]} - \frac{\text{amp}[\text{end} - 1]}{\text{amp}[\text{end} - 2]}\right)\right)$$

Pcm model ODE solution fits pcm mutant time trace data.

pcm mRNA



pcm PER



Iteration 45200: Best value = 0.22608254311121734, Temperature = 1.522264525982424e-10
 Iteration 45400: Best value = 0.22608254311121734, Temperature = 1.3773674571581714e-10
 Iteration 45600: Best value = 0.22608254311121734, Temperature = 1.2462624462814756e-10
 Iteration 45800: Best value = 0.22608254311121734, Temperature = 1.1276366934180646e-10

$$\text{cost} = \text{cost}_{amp}(\text{mRNAs}) + \text{cost}_{amp}(\text{proteins})$$

$$+ \sqrt{\underbrace{\left(1 - \frac{\text{relamp}(\text{mRNAs})}{1 - \min(\text{pcmMper})}\right)^2}_{\text{cost}_{relamp}(\text{mRNAs})} + \underbrace{\left(1 - \frac{\text{relamp}(\text{proteins})}{1 - \min(\text{pcmPER})}\right)^2}_{\text{cost}_{relamp}(\text{PER})} + \underbrace{\left(1 - \frac{\text{period}(\text{proteins})}{24}\right)^2}_{\text{cost}_{period}(\text{PER})}}$$

Best value: 0.22608254311121734

Cost components at best state:

ddmeasure(mRNAs)[1]: 0.0

ddmeasure(proteins)[1]: 0.0

(1 - ddmeasure(mRNAs)[5] / (1 - minimum(pcmMper)))^2: 0.013170807449025554

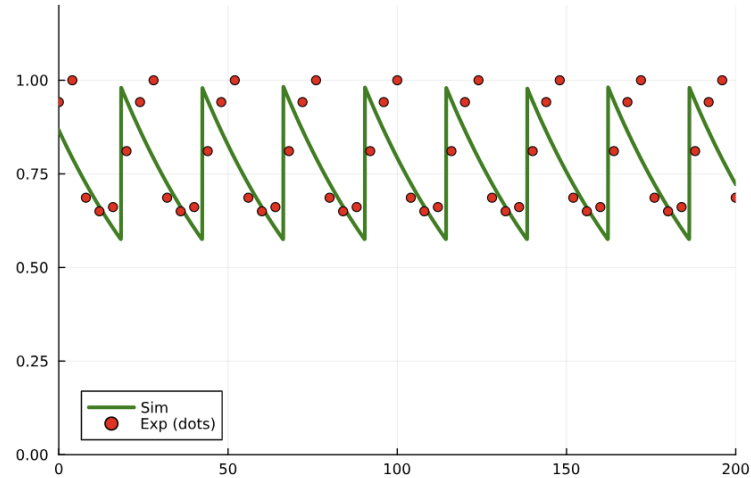
(1 - ddmeasure(proteins)[5] / (1 - minimum(pcmPER)))^2: 0.037941814406165446

(1 - ddmeasure(proteins)[3] / 24)^2: 6.9444444444442915e-7

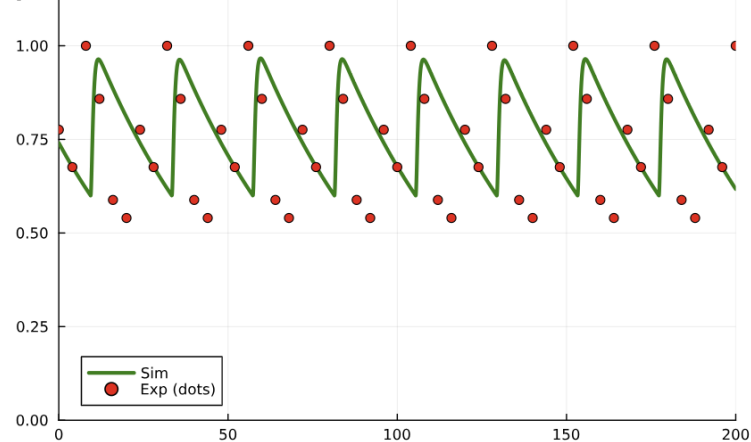
$$\max\left(0, 5 \times \left(2 - \frac{\text{amp}[\text{end}]}{\text{amp}[\text{end} - 1]} - \frac{\text{amp}[\text{end} - 1]}{\text{amp}[\text{end} - 2]}\right)\right)$$

Pcm model ODE solution fits pcm mutant time trace data.

pcm mRNA



pcm PER



Iteration 45200: Best value = 0.22608254311121734, Temperature = 1.522264525982424e-10
 Iteration 45400: Best value = 0.22608254311121734, Temperature = 1.3773674571581714e-10
 Iteration 45600: Best value = 0.22608254311121734, Temperature = 1.2462624462814756e-10
 Iteration 45800: Best value = 0.22608254311121734, Temperature = 1.1276366934180646e-10

$$\text{cost} = \text{cost}_{\text{amp}}(\text{mRNAs}) + \text{cost}_{\text{amp}}(\text{proteins})$$

$$+ \sqrt{\underbrace{\left(1 - \frac{\text{relamp}(\text{mRNAs})}{1 - \min(\text{pcmMper})}\right)^2}_{\text{cost}_{\text{relamp}}(\text{mRNAs})} + \underbrace{\left(1 - \frac{\text{relamp}(\text{proteins})}{1 - \min(\text{pcmPER})}\right)^2}_{\text{cost}_{\text{relamp}}(\text{PER})} + \underbrace{\left(1 - \frac{\text{period}(\text{proteins})}{24}\right)^2}_{\text{cost}_{\text{period}}(\text{PER})}}$$

Best value: 0.22608254311121734

Cost components at best state:

ddmeasure(mRNAs)[1]: 0.0

ddmeasure(proteins)[1]: 0.0

(1 - ddmeasure(mRNAs)[5] / (1 - minimum(pcmMper)))^2: 0.013170807449025554

(1 - ddmeasure(proteins)[5] / (1 - minimum(pcmPER)))^2: 0.037941814406165446

(1 - ddmeasure(proteins)[3] / 24)^2: 6.9444444444442915e-7

$$a_3 = 2.947834891470722$$

$$b_1 = 0.02234956055138963$$

$$b_2 = 0.015991595872378122$$

$$b_3 = 2.076437798547174$$

$$AT = 0.016806368455979742$$

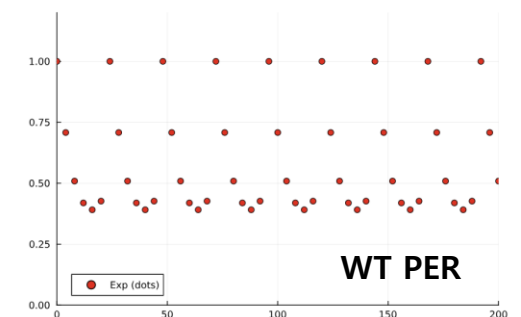
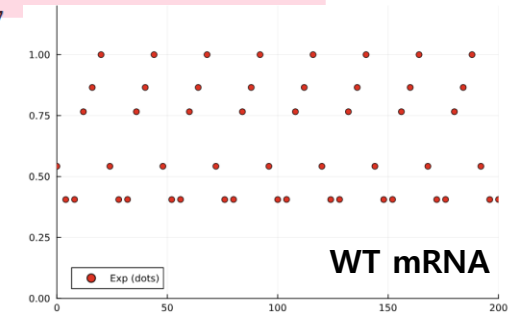
$$K_a = 6.091560110356015 \times 10^{-12}$$

$$K_s = 3.64850378800174 \times 10^{-11}$$

$$K_b = 1.763125206116844 \times 10^{-5}$$

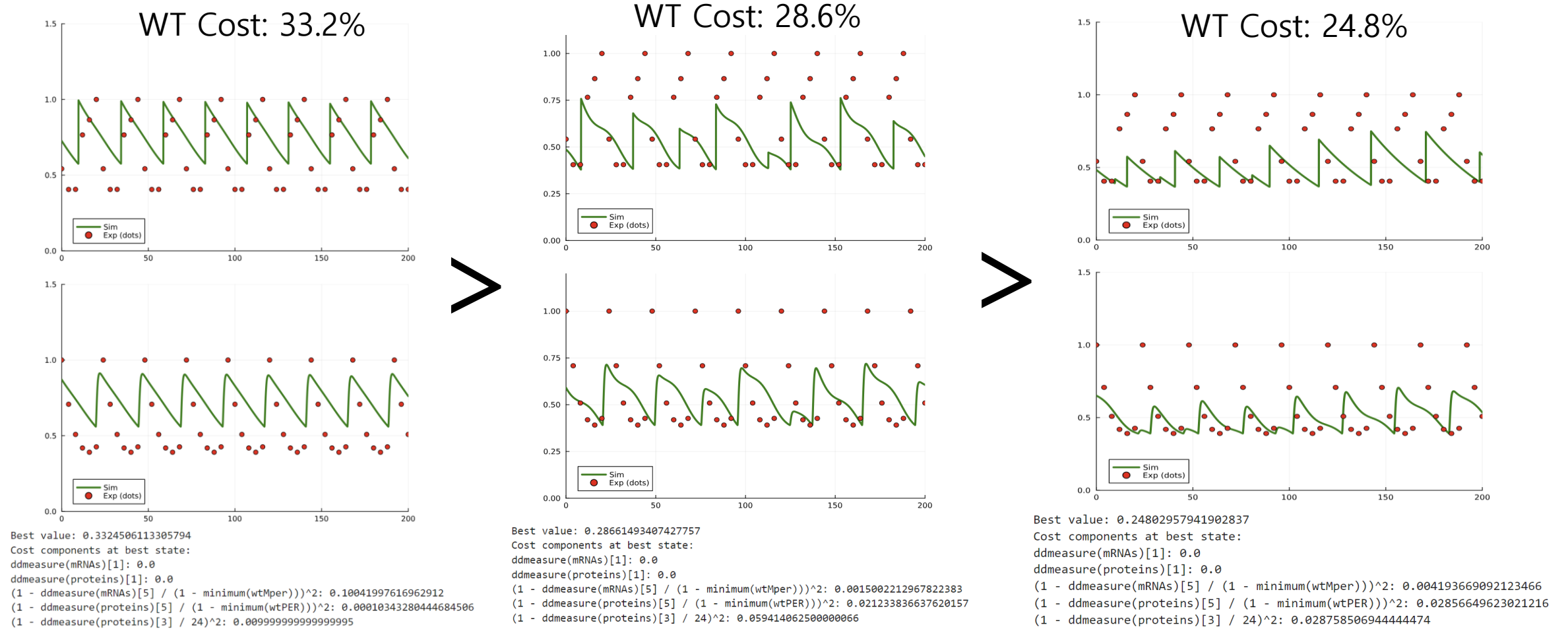
$$K_d = 1923.5904640470426$$

+ $\text{amp}_T, \text{ph}_T$
 $\text{amp}_D, \text{ph}_D$??



$$\max\left(0, 5 \times \left(2 - \frac{\text{amp}[\text{end}]}{\text{amp}[\text{end} - 1]} - \frac{\text{amp}[\text{end} - 1]}{\text{amp}[\text{end} - 2]}\right)\right)$$

Cost function needs to be modified for WT for better data fit / consistent circadian oscillation.



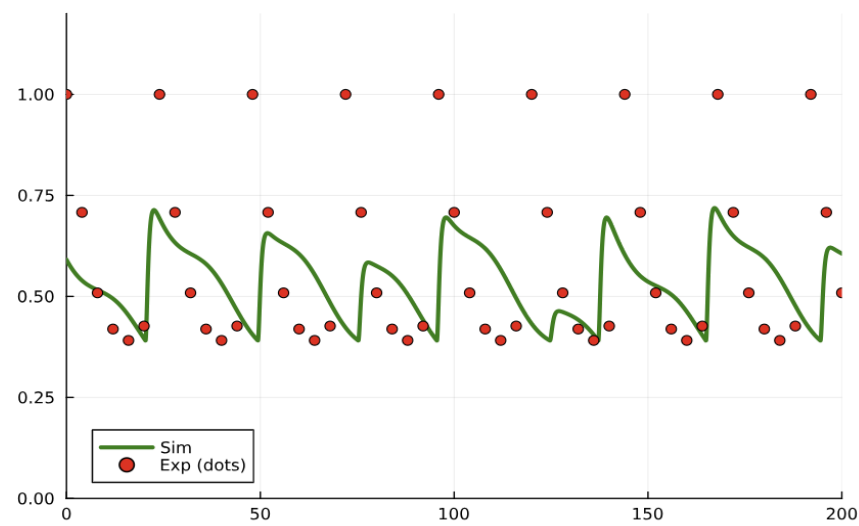
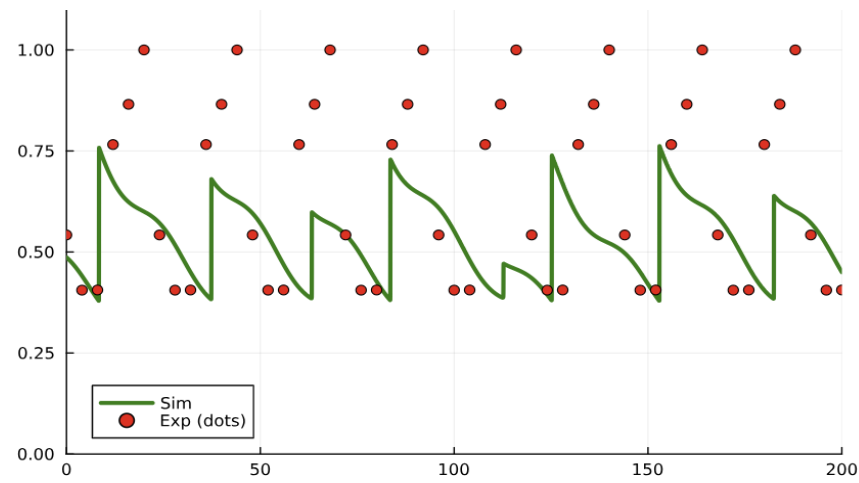
pcm cost function

$$\text{cost} = \text{cost}_{\text{amp}}(\text{mRNAs}) + \text{cost}_{\text{amp}}(\text{proteins}) + \sqrt{\underbrace{\left(1 - \frac{\text{relamp}(\text{mRNAs})}{1 - \min(\text{pcmMper})}\right)^2}_{\text{cost}_{\text{relamp}}(\text{mRNAs})} + \underbrace{\left(1 - \frac{\text{relamp}(\text{proteins})}{1 - \min(\text{pcmPER})}\right)^2}_{\text{cost}_{\text{relamp}}(\text{PER})} + \underbrace{\left(1 - \frac{\text{period}(\text{proteins})}{24}\right)^2}_{\text{cost}_{\text{period}}(\text{PER})}}$$



Modify... (i.e. Add Extra Cost Terms (WT))

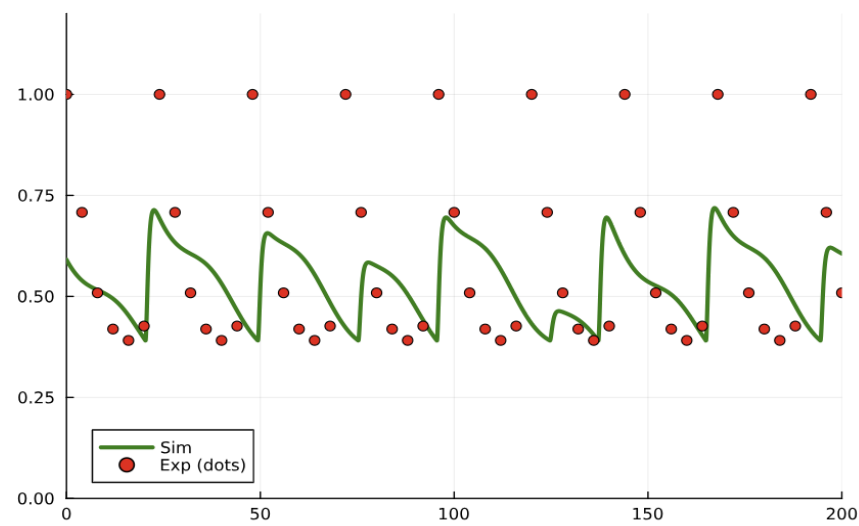
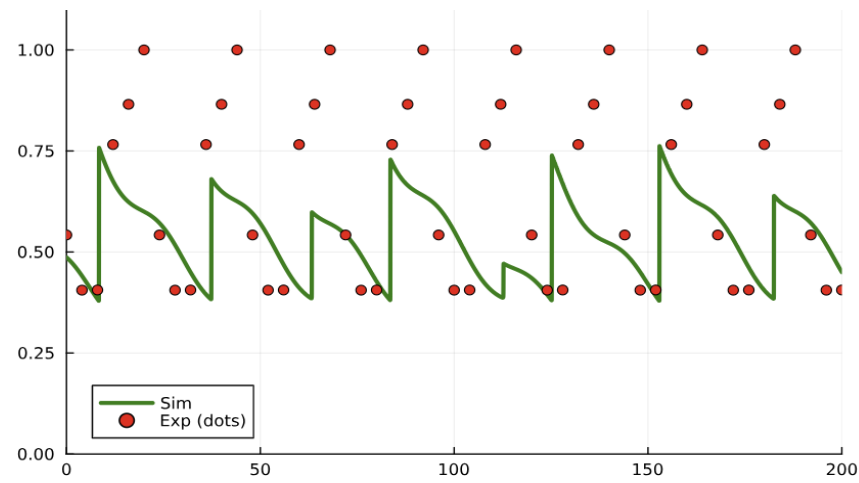
Future Direction: Cost function must be modified to fit data's amplitude/period and to reduce variance in period, amplitude.



	Cycle 1	Cycle 2	Cycle 3	[...]	Cycle N
Amplitude	0.35	0.15	0.1		0.08
Period	27	21	23		25

Reduce Variance..

Future Direction: Cost function must be modified to fit data's amplitude/period and to reduce variance in period, amplitude.



	Cycle 1	Cycle 2	Cycle 3	[...]	Cycle N
Amplitude	0.35	0.15	0.1		0.08
Period	27	21	23		25

Reduce Variance..

Measure CVs

```
function ddmeasure(ts) :
    ts_smooth ← movingAverage(ts, 100)
    peaks, troughs ← findExtrema(ts_smooth)

    if min(|peaks|, |troughs|) ≥ 5 :
        amps ← |peaks1:n - troughs1:n|
        periods ← 0.01 × diff(peakIndices)

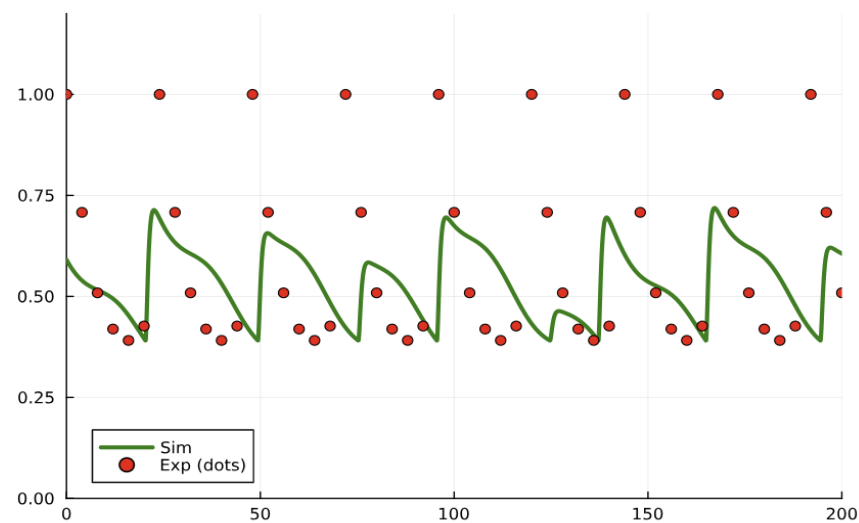
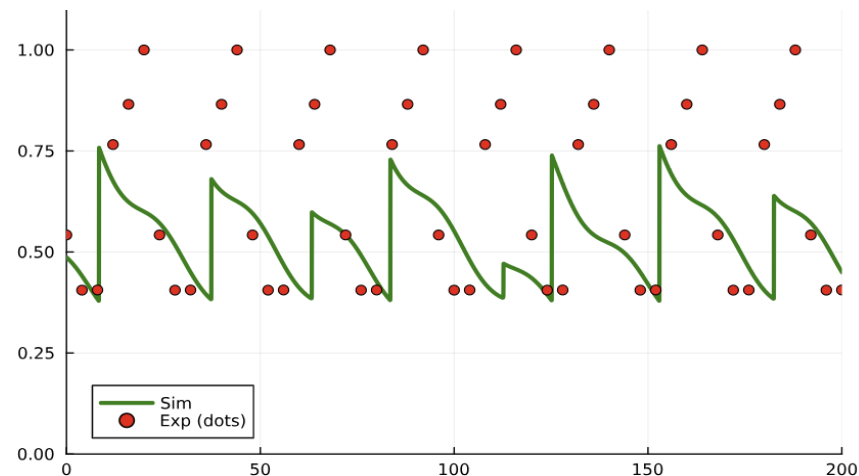
        // Original metrics
        costf ← max(0, 5(2 -  $\frac{amps_{-1}}{amps_{-2}} - \frac{amps_{-2}}{amps_{-3}}$ ))
        amp, period ← amps-1, periods-1
        level ← (peaks-1 + troughs-1)/2
        relamp ← amps-1/peaks-1

        // New metrics
        μamp, σamp ← mean(amps), std(amps)
        μperiod, σperiod ← mean(periods), std(periods)
        camp_var ← σamp/μamp
        cperiod_var ← σperiod/μperiod
        cperiod_24h ← |μperiod - 24|/24

        return costf, amp, period, level, relamp, camp_var, cperiod_var, cperiod_24h, μamp
    else:
        return 10, 0, 0, ts-1, 0, 10, 10, 10, 0
```

+ CV (amplitude)
+ CV (period)

Future Direction: Cost function must be modified to fit data's amplitude/period and to reduce variance in period, amplitude.



	Cycle 1	Cycle 2	Cycle 3	[...]	Cycle N
Amplitude	0.35	0.15	0.1		0.08
Period	27	21	23		25

Reduce Variance..

Measure CVs



Include in New_Cost

```
function ddmeasure(ts) :
    ts_smooth ← movingAverage(ts, 100)
    peaks, troughs ← findExtrema(ts_smooth)

    if min(|peaks|, |troughs|) ≥ 5 :
        amps ← |peaks1:n - troughs1:n|
        periods ← 0.01 × diff(peakIndices)

        // Original metrics
        costf ← max(0, 5(2 -  $\frac{amps_{-1}}{amps_{-2}} - \frac{amps_{-2}}{amps_{-3}}$ ))
        amp, period ← amps-1, periods-1
        level ← (peaks-1 + troughs-1)/2
        relamp ← amps-1/peaks-1

        // New metrics
        μamp, σamp ← mean(amps), std(amps)
        μperiod, σperiod ← mean(periods), std(periods)
        camp_var ← σamp/μamp
        cperiod_var ← σperiod/μperiod
        cperiod_24h ← |μperiod - 24|/24

        return costf, amp, period, level, relamp, camp_var, cperiod_var, cperiod_24h, μamp
    else:
        return 10, 0, 0, ts-1, 0, 10, 10, 10, 0
```

+ CV (amplitude)
+ CV (period)

```
function ddsa(params) :
    a3, b1, b2, b3, AT, Ka, Ks, Kb, Kd, ampD, phD, ampT, phT ← params
    ampfixed_1, ampfixed_3 ← amp_mRNA, amp_PER
    mRNAs, proteins ← solveODE(modelWT, params, (0, 300))
    mRNAmetrics, proteinmetrics ← ddmeasure(mRNAs), ddmeasure(proteins)

    // Original costs
    cmRNA, cprotein ← mRNAmetrics1, proteinmetrics1
    cmRNA_amp ← (1 - mRNAmetrics5/(1 - min(wtMper)))2
    cprotein_amp ← (1 - proteinmetrics5/(1 - min(wtPER)))2
    cprotein_period ← (1 - proteinmetrics3/24)2

    // New costs
    cmRNA_amp_var, cmRNA_period_var, cmRNA_period_24h ← mRNAmetrics6:8
    cprotein_amp_var, cprotein_period_var, cprotein_period_24h ← proteinmetrics6:8
    cmRNA_amp_fixed ← |mRNAmetrics9 - ampfixed_1|/ampfixed_1
    cprotein_amp_fixed ← |proteinmetrics9 - ampfixed_3|/ampfixed_3

    costoriginal ← cmRNA + cprotein + √(cmRNA_amp + cprotein_amp + cprotein_period)
    costnew ← ∑ of all new costs
    costtotal ← costoriginal + costnew

    return costtotal, (all cost components)
```