



CERAMATHS



Université  
Polytechnique  
HAUTS-DE-FRANCE

## Physically-inspired Gaussian processes: Application in biology

SIAM Conference on Uncertainty Quantification 2022

MS87 Incorporating Structural Information in Kernel Methods

---

Andrés F. López-Lopera

April 13, 2022

Céramaths, Univ. Polytechnique Hauts-de-France, France

# Joint work with...



Nicolas Durrande  
Principal Research Scientist  
SecondMind, UK



Mauricio A. Álvarez  
Senior Lecturer in Machine Learning  
Department of Computer Science  
University of Manchester, UK

Physically-inspired Gaussian processes for post-transcriptional regulation in *Drosophila*,  
in *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, vol. 18, no. 2, pp.  
656-666, 2021, doi: 10.1109/TCBB.2019.2918774.

R codes: <https://github.com/anfelopera/PhysicallyGPDrosophila>

# Table of contents

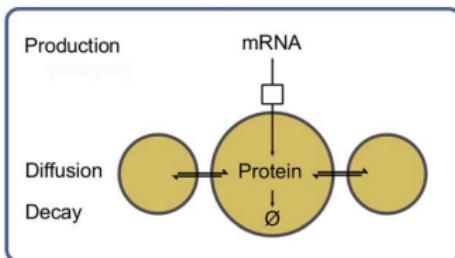
1. Motivation
2. Gaussian processes
3. Physically-inspired Gaussian processes
4. Biological application
5. Remarks and conclusions

## Motivation

---

# Post-transcriptional regulation

- Post-transcriptional regulation at the mRNA level:



$$\frac{\partial \mathbf{y}(x, t)}{\partial t} = S u(x, t) - \lambda \mathbf{y}(x, t) + D \frac{\partial^2 \mathbf{y}(x, t)}{\partial x^2} \quad (\text{reaction-diffusion process})$$

where

- $\mathbf{y}(x, t)$ : relative protein concentration, at location  $x$  and instant  $t$ ;
- $u$ : messenger RNA (mRNA);
- $S, D, \lambda$ : translation, decay and diffusion rate parameters, respectively.

# Challenges

- Data acquisition in the modelling cycle can be expensive.
  - “Cheaper” (*probabilistic*) *meta-models* are useful.
- mRNAs (*u*) are commonly unknown and must be inferred.
  - *Protein concentration* data (*y*) can be exploited.
  - The *reaction-diffusion equation* gives physical information.

# Challenges

- Data acquisition in the modelling cycle can be expensive.
  - “Cheaper” (*probabilistic*) *meta-models* are useful.
- mRNAs ( $u$ ) are commonly unknown and must be inferred.
  - *Protein concentration* data ( $y$ ) can be exploited.
  - The *reaction-diffusion equation* gives physical information.
- \* We focus on physically-inspired models based on *Gaussian processes*:
  - They provide well-founded probabilistic frameworks.
  - Parameters ( $S, D, \lambda$ ) can be encoded into *kernel* functions.
  - $u$  can be inferred using  $y$  data.

## Gaussian processes

---

- GPs form a flexible **prior over functions** [Rasmussen and Williams, 2005].
- Let  $\{Y(x); x \in \mathbb{R}^d\}$  be a GP. Then,  $Y$  is defined by its **mean  $\mu$**  and **kernel  $k_\Theta$** :

$$Y \sim \mathcal{GP}(\mu, k_\Theta), \quad (1)$$

where  $k := k_\Theta$  is parametrised by  $\Theta$  (covariance parameters) and is given by,

$$k(x, x') = \text{cov} \{Y(x), Y(x')\} = \mathbb{E} \{[Y(x) - \mu(x)][Y(x') - \mu(x')]\}.$$

- In practice:
  - *Centred GPs* are considered, i.e.  $\mu(\cdot) = 0$ ;
  - Parameters  $\Theta$  are estimated via *maximum likelihood*.

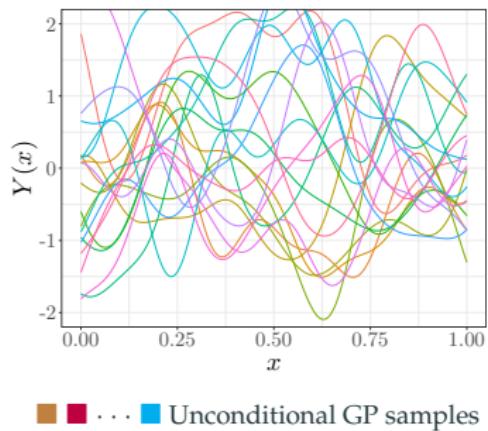
# Gaussian processes (GPs)

- For a finite number of design points  $(x_1, \dots, x_n)$ , we have:

$$\mathbf{Y}_n = [Y(x_1), \dots, Y(x_n)]^\top \sim \mathcal{N}(\boldsymbol{\mu}(= \mathbf{0}), \mathbf{K}), \quad (2)$$

with covariance matrix  $\mathbf{K} = (k(x_i, x_j))_{1 \leq i, j \leq n}$ .

- Effect of kernels on unconditional GP samples with  $\Theta = (\sigma^2 = 1, \theta = 0.1)$ :



**Squared Exponential (SE) kernel:**

$$k_{\sigma^2, \theta}(x, x') = \sigma^2 \exp \left\{ -\frac{(x - x')^2}{\theta^2} \right\},$$

with variance  $\sigma^2$  and length-scale  $\theta$ .

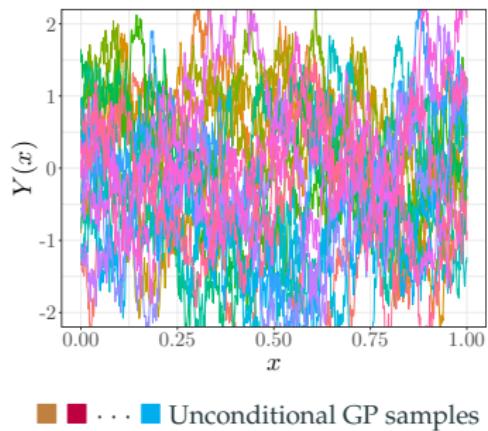
# Gaussian processes (GPs)

- For a finite number of design points  $(x_1, \dots, x_n)$ , we have:

$$\mathbf{Y}_n = [Y(x_1), \dots, Y(x_n)]^\top \sim \mathcal{N}(\boldsymbol{\mu}(= \mathbf{0}), \mathbf{K}), \quad (2)$$

with covariance matrix  $\mathbf{K} = (k(x_i, x_j))_{1 \leq i, j \leq n}$ .

- Effect of kernels on unconditional GP samples with  $\Theta = (\sigma^2 = 1, \theta = 0.1)$ :



Exponential kernel:

$$k_{\sigma^2, \theta}(x, x') = \sigma^2 \exp \left\{ -\frac{|x - x'|}{\theta} \right\},$$

with variance  $\sigma^2$  and length-scale  $\theta$ .

## Conditional distribution

- Let  $\mathbf{Y}_* = [Y(x_1^*), \dots, Y(x_m^*)]^\top$  be a Gaussian vector extracted from  $\mathbf{Y}$ .
- For an observed vector  $\mathbf{y} = [y_1, \dots, y_n]^\top$ , and noise  $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \tau^2 \mathbf{I})$ , we have:

$$\mathbf{Y}_* | \{\mathbf{Y}_n + \boldsymbol{\varepsilon} = \mathbf{y}\} \sim \mathcal{N}(\mathbf{m}, \boldsymbol{\Sigma}), \quad (\text{conditional distribution}) \quad (3)$$

with conditional parameters,

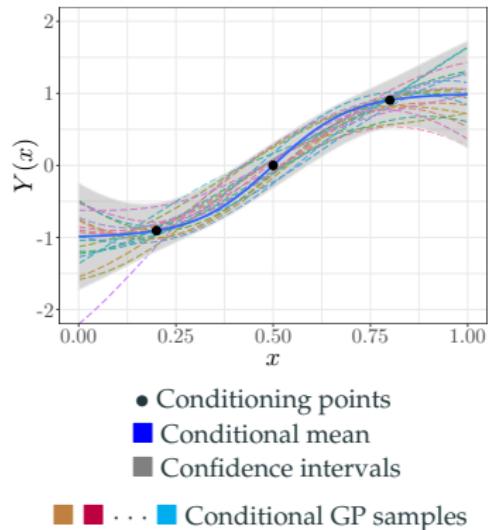
$$\mathbf{m} = \mathbf{K}_{\mathbf{Y}_*, \mathbf{Y}_n} \mathbf{C}^{-1} \mathbf{y}, \quad \boldsymbol{\Sigma} = \mathbf{K}_{\mathbf{Y}_*, \mathbf{Y}_*} - \mathbf{K}_{\mathbf{Y}_*, \mathbf{Y}_n} \mathbf{C}^{-1} \mathbf{K}_{\mathbf{Y}_n, \mathbf{Y}_*},$$

with  $\mathbf{C} = \mathbf{K}_{\mathbf{Y}_n, \mathbf{Y}_n} + \tau^2 \mathbf{I}$ .

- Note that computing  $\mathbf{C}^{-1}$  is challenging when  $n$  is large.

# Gaussian processes (GPs)

- Conditional GP with  $\hat{\Theta} = (\hat{\sigma}^2 = 0.9, \hat{\theta} = 0.4)$  and  $\tau^2 = 0.03$ :



Squared Exponential (SE) kernel:

$$k_{\sigma^2, \theta}(x, x') = \sigma^2 \exp \left\{ -\frac{(x - x')^2}{\theta^2} \right\},$$

with variance  $\sigma^2$  and length-scale  $\theta$ .

# Gaussian processes (GPs): what about adding prior knowledge?

- Standard GPs are “purely” *data-driven* frameworks.
- If we have prior knowledge, how can it be encoded in the prior?

# Gaussian processes (GPs): what about adding prior knowledge?

- Standard GPs are “purely” *data-driven* frameworks.
- If we have prior knowledge, how can it be encoded in the prior?
- \* Here we show that physical information can also be encoded into GPs!

## Physically-inspired Gaussian processes

---

# Linear operations of GPs

- Let  $\mathbf{U} \sim \mathcal{GP}(0, k_{\mathbf{u}, \mathbf{u}})$  with covariance function:

$$k_{\mathbf{u}, \mathbf{u}}(\mathbf{x}, \mathbf{x}') = \text{cov} \{ \mathbf{U}(\mathbf{x}), \mathbf{U}(\mathbf{x}') \} = \mathbb{E} \{ \mathbf{U}(\mathbf{x}) \mathbf{U}(\mathbf{x}') \}.$$

## Linear operations of GPs

- Let  $\mathbf{U} \sim \mathcal{GP}(0, k_{\mathbf{u}, \mathbf{u}})$  with covariance function:

$$k_{\mathbf{u}, \mathbf{u}}(\mathbf{x}, \mathbf{x}') = \text{cov} \{ \mathbf{U}(\mathbf{x}), \mathbf{U}(\mathbf{x}') \} = \mathbb{E} \{ \mathbf{U}(\mathbf{x}) \mathbf{U}(\mathbf{x}') \}.$$

- For any linear operator  $\mathcal{L}$  that commutes with the covariance,

$$\mathbf{Y} = \mathcal{L} \circ \mathbf{U} := \mathcal{L}(\mathbf{U}),$$

we have that  $\mathbf{Y} \sim \mathcal{GP}(0, k_{\mathbf{y}, \mathbf{y}})$  with covariance function:

$$k_{\mathbf{y}, \mathbf{y}}(\mathbf{x}, \mathbf{x}') = \text{cov} \{ \mathcal{L} \circ \mathbf{U}(\mathbf{x}), \mathcal{L} \circ \mathbf{U}(\mathbf{x}') \} = \mathcal{L} \circ \mathcal{L}' \circ k_{\mathbf{u}, \mathbf{u}}(\mathbf{x}, \mathbf{x}').$$

- Furthermore, cross-covariance functions can be established:

$$k_{\mathbf{y}, \mathbf{u}}(\mathbf{x}, \mathbf{x}') = \text{cov} \{ \mathcal{L} \circ \mathbf{U}(\mathbf{x}), \mathbf{U}(\mathbf{x}') \} = \mathcal{L} \circ k_{\mathbf{u}, \mathbf{u}}(\mathbf{x}, \mathbf{x}'),$$

$$k_{\mathbf{u}, \mathbf{y}}(\mathbf{x}, \mathbf{x}') = \text{cov} \{ \mathbf{U}(\mathbf{x}), \mathcal{L} \circ \mathbf{U}(\mathbf{x}') \} = \mathcal{L}' \circ k_{\mathbf{u}, \mathbf{u}}(\mathbf{x}, \mathbf{x}').$$

# Linear operations of GPs

- Hence, the joint process  $(\mathbf{U}, \mathbf{Y})$  is also (centred) GP-distributed:

$$\begin{bmatrix} \mathbf{U} \\ \mathbf{Y} \end{bmatrix} = \mathcal{N} \left( \mathbf{0}, \begin{bmatrix} \mathbf{K}_{\mathbf{u}, \mathbf{u}} & \mathbf{K}_{\mathbf{y}, \mathbf{u}}^\top \\ \mathbf{K}_{\mathbf{y}, \mathbf{u}} & \mathbf{K}_{\mathbf{y}, \mathbf{y}} \end{bmatrix} \right),$$

and, therefore, conditional GP formulas can be applied.

- \* e.g.  $\mathbf{U}$  can be inferred using observations of  $\mathbf{Y}$ :

$$\mathbf{U} | \{\mathbf{Y} = \mathbf{y}\} \sim \mathcal{N} \left( \mathbf{0}, \mathbf{K}_{\mathbf{u}, \mathbf{u}} - \mathbf{K}_{\mathbf{y}, \mathbf{u}}^\top \mathbf{K}_{\mathbf{y}, \mathbf{y}}^{-1} \mathbf{K}_{\mathbf{y}, \mathbf{u}} \right). \quad (4)$$

## Physically-inspired Gaussian processes: dynamical system

- Let  $\{Y(x, t); x \in \mathbb{R}, t \in \mathbb{R}\}$  and  $\{U(x, t); x \in \mathbb{R}, t \in \mathbb{R}\}$  be random processes.
- We consider the (linear) reaction-diffusion equation:

$$\frac{\partial Y(x, t)}{\partial t} = SU(x, t) - \lambda Y(x, t) + D \frac{\partial^2 Y(x, t)}{\partial x^2}, \quad (5)$$

where

- $Y$ : relative gap protein concentration;
  - $U$ : messenger RNA (mRNA);
  - $S, D, \lambda$ : translation, decay and diffusion rate constants, respectively.
- For simplicity, we assume homogeneous conditions:

$$Y(x, t = 0) = 0, \quad Y(x = 0, t) = Y(x = l, t) = 0, \text{ for } x \in [0, l], l \in \mathbb{R}^+.$$

## GP-mRNA model

- Physically-inspired GPs result from placing GP priors over  $\textcolor{green}{Y}$  or  $\textcolor{red}{U}$ :
  - e.g. assuming  $\textcolor{red}{U} \sim \mathcal{GP}(0, k_{\textcolor{red}{u}, \textcolor{red}{u}})$  and computing the process  $\textcolor{green}{Y}$ .
- By solving the *reaction-diffusion equation*, we have [Álvarez et al., 2009]:

$$\textcolor{green}{Y}(x, t) = \textcolor{brown}{S} \int_0^t \int_0^{\textcolor{brown}{l}} \textcolor{red}{U}(\xi, \tau) \textcolor{blue}{G}(x, \xi, t - \tau) d\xi d\tau, \quad (6)$$

with *Green's function*  $G(x, \xi, t)$  given by,

$$\textcolor{blue}{G}(x, \xi, t) = \frac{2}{l} \exp\{-\lambda t\} \sum_{n=1}^{\infty} \sin\left(\frac{n\pi}{l}x\right) \sin\left(\frac{n\pi}{l}\xi\right) \exp\left\{-D\left(\frac{n\pi}{l}\right)^2 t\right\}.$$

# Physically-inspired Gaussian processes: GP-mRNA model

· Covariance  $k_{\textcolor{green}{y},\textcolor{green}{y}}((x, t), (x', t'))$ :

$$\text{cov} \{ \textcolor{green}{Y}(x, t), \textcolor{green}{Y}(x', t') \}$$

$$= \textcolor{brown}{S}^2 \mathbb{E} \left\{ \int_0^t \int_0^{\textcolor{red}{l}} \textcolor{red}{U}(\xi, \tau) \textcolor{blue}{G}(x, \xi, t - \tau) d\xi d\tau \cdot \int_0^{t'} \int_0^{\textcolor{red}{l}} \textcolor{red}{U}(\xi', \tau') \textcolor{blue}{G}(x', \xi', t' - \tau') d\xi' d\tau' \right\}$$

$$= \textcolor{brown}{S}^2 \int_0^t \int_0^{t'} \int_0^{\textcolor{red}{l}} \int_0^{\textcolor{red}{l}} \textcolor{blue}{G}(x, \xi, t - \tau) \textcolor{blue}{G}(x', \xi', t' - \tau') k_{\textcolor{red}{u},\textcolor{red}{u}}((\xi, \tau), (\xi', \tau')) d\xi' d\xi d\tau' d\tau.$$

# Physically-inspired Gaussian processes: GP-mRNA model

- Covariance  $k_{\textcolor{green}{y},\textcolor{green}{y}}((x, t), (x', t'))$ :

$$\text{cov} \{ \textcolor{green}{Y}(x, t), \textcolor{green}{Y}(x', t') \}$$

$$\begin{aligned} &= \textcolor{brown}{S}^2 \mathbb{E} \left\{ \int_0^t \int_0^{\textcolor{red}{l}} \textcolor{red}{U}(\xi, \tau) \textcolor{blue}{G}(x, \xi, t - \tau) d\xi d\tau \cdot \int_0^{t'} \int_0^{\textcolor{red}{l}} \textcolor{red}{U}(\xi', \tau') \textcolor{blue}{G}(x', \xi', t' - \tau') d\xi' d\tau' \right\} \\ &= \textcolor{brown}{S}^2 \int_0^t \int_0^{t'} \int_0^{\textcolor{red}{l}} \int_0^{\textcolor{red}{l}} \textcolor{blue}{G}(x, \xi, t - \tau) \textcolor{blue}{G}(x', \xi', t' - \tau') k_{\textcolor{red}{u},\textcolor{red}{u}}((\xi, \tau), (\xi', \tau')) d\xi' d\xi d\tau' d\tau. \end{aligned}$$

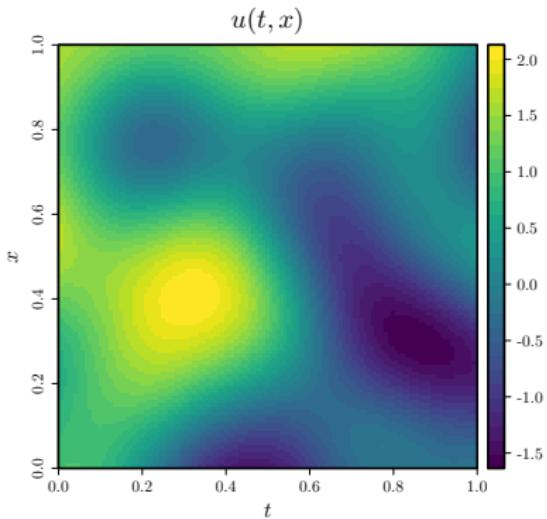
- Cross-covariance  $k_{\textcolor{green}{y},\textcolor{red}{u}}((x, t), (x', t'))$ :

$$\text{cov} \{ \textcolor{green}{Y}(x, t), \textcolor{red}{U}(x', t') \} = \textcolor{brown}{S} \int_0^t \int_0^{\textcolor{red}{l}} \textcolor{blue}{G}(x, \xi, t - \tau) k_{\textcolor{red}{u},\textcolor{red}{u}}((\xi, \tau), (x', t')) d\xi d\tau.$$

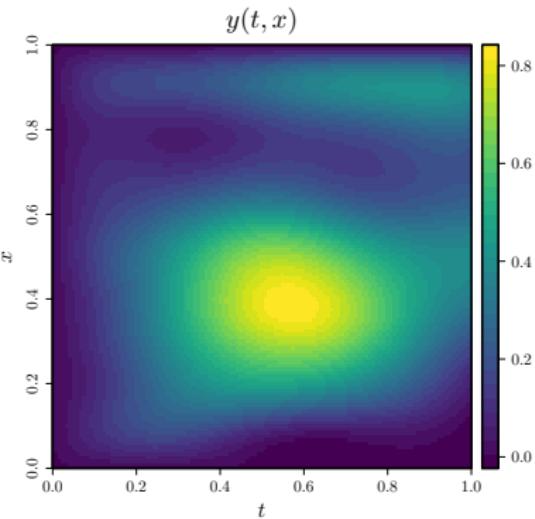
- Note that  $k_{\textcolor{green}{y},\textcolor{green}{y}}$  and  $k_{\textcolor{green}{y},\textcolor{red}{u}}$  have closed-forms only for some kernels  $k_{\textcolor{red}{u},\textcolor{red}{u}}$ , e.g.:

$$k_{\textcolor{red}{u},\textcolor{red}{u}}((x, t), (x', t')) = \exp \left( - \frac{(x - x')^2}{\theta_x^2} \right) \exp \left( - \frac{(t - t')^2}{\theta_t^2} \right). \quad (7)$$

## GP-mRNA model: toy example – functions to approximate



(a) mRNA

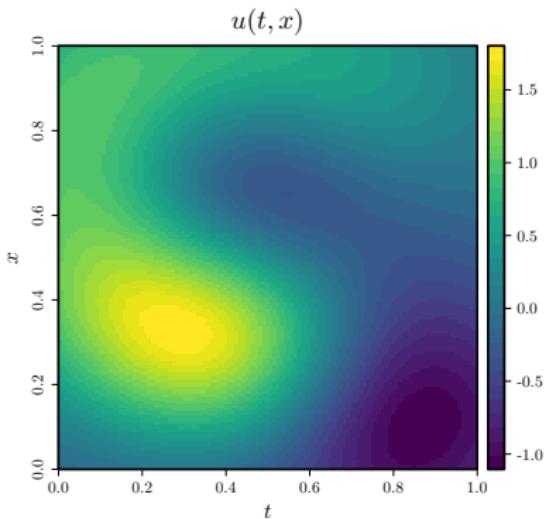


(b) Protein

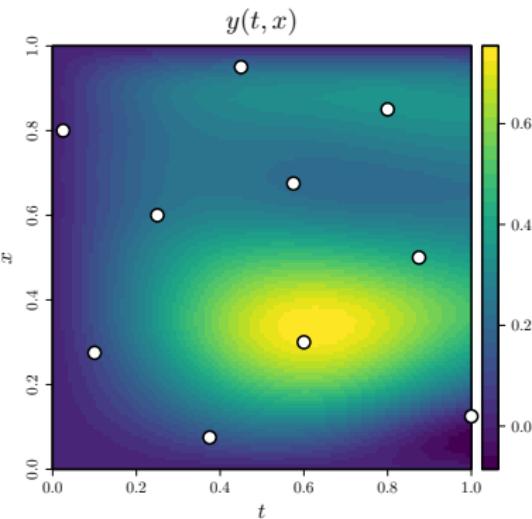
Sample from the GP-mRNA model

$$Q^2 = 1 - \text{SMSE}.$$

## GP-mRNA model: toy example – inference of $\mathbf{U}$ using $\mathbf{Y}$



$$(a) Q^2 = 0.674$$

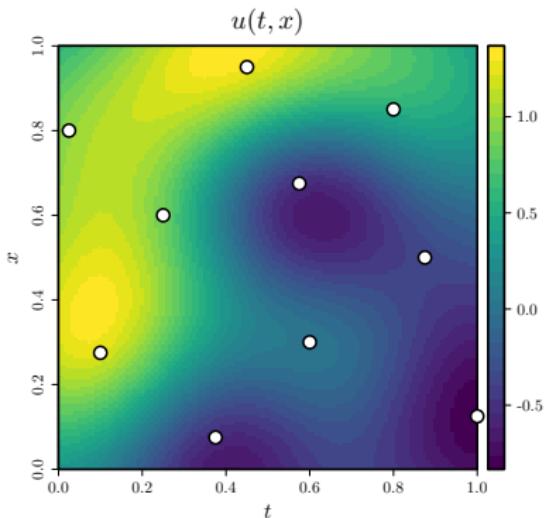


$$(b) Q^2 = 0.842$$

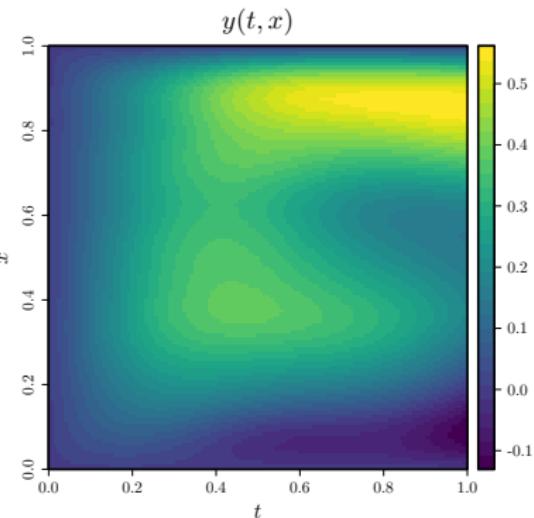
- 10 conditioning points from a maximin Latin hypercube design

$$Q^2 = 1 - \text{SMSE}.$$

## GP-mRNA model: toy example – inference of $\mathbf{Y}$ using $\mathbf{U}$



$$(a) Q^2 = 0.520$$

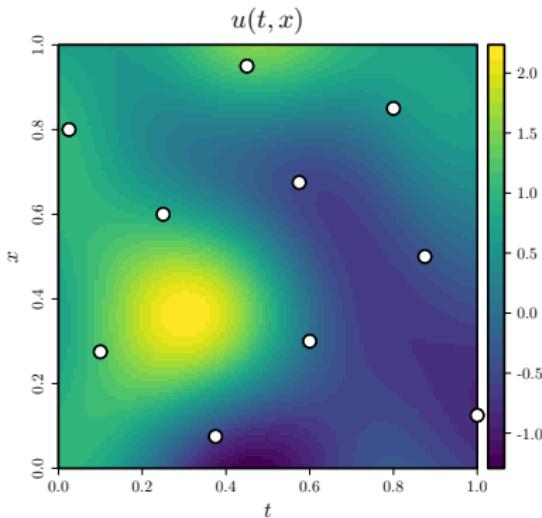


$$(b) Q^2 = 0.276$$

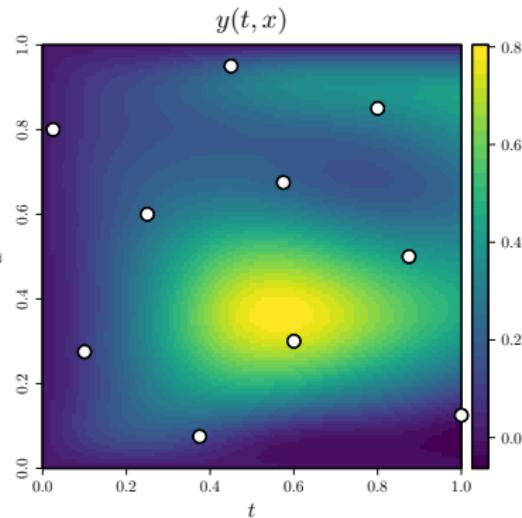
- 10 conditioning points from a maximin Latin hypercube design

$$Q^2 = 1 - \text{SMSE}.$$

## GP-mRNA model: toy example – inference using $\mathbf{U}$ and $\mathbf{Y}$



(a)  $Q^2 = 0.872$

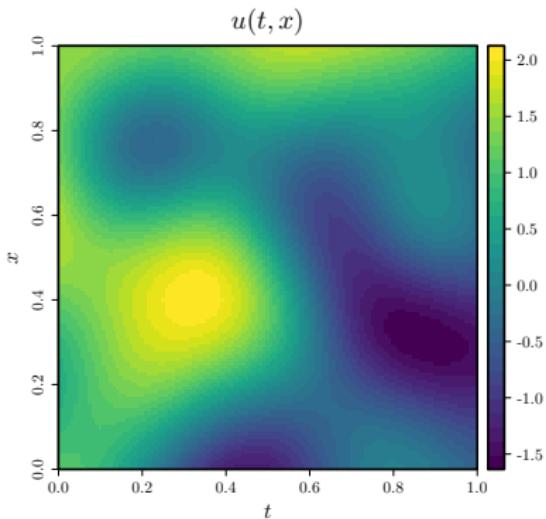


(b)  $Q^2 = 0.950$

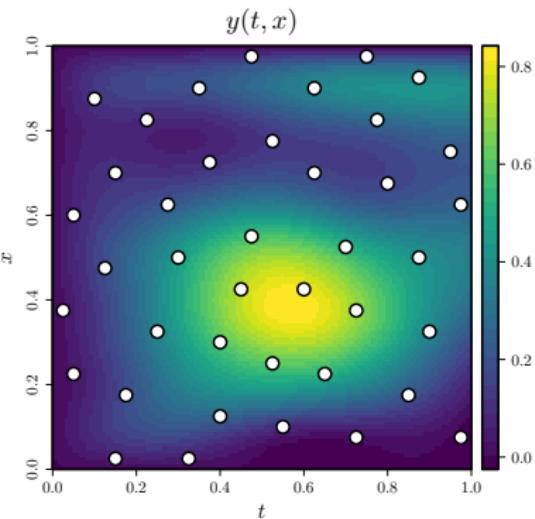
- 10 conditioning points from a maximin Latin hypercube design

$$Q^2 = 1 - \text{SMSE}.$$

## GP-mRNA model: toy example – inference of $\mathbf{U}$ using $\mathbf{Y}$



$$(a) Q^2 = 0.988$$

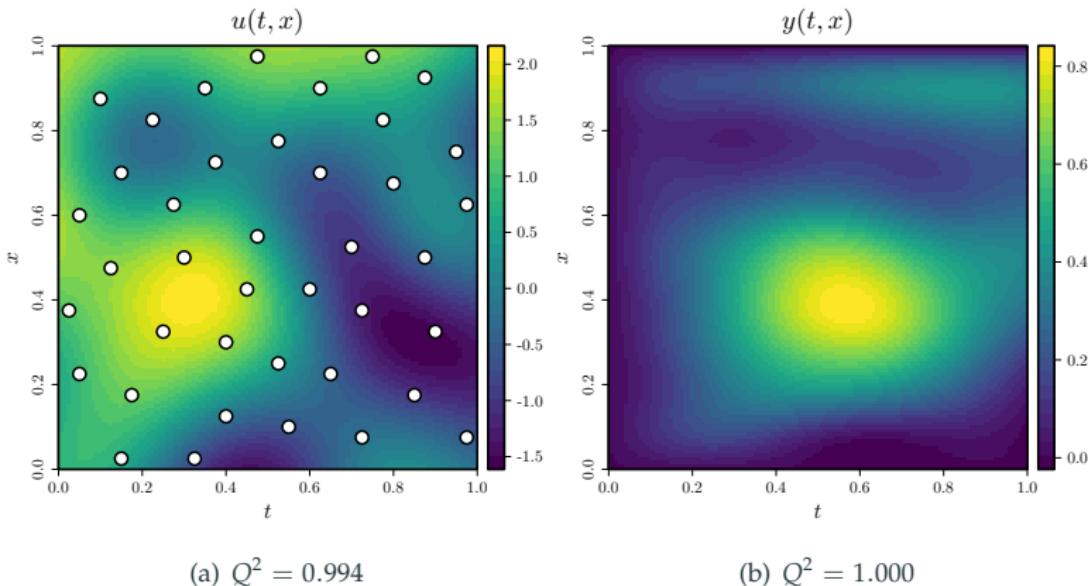


$$(b) Q^2 = 0.999$$

- o 40 conditioning points from a maximin Latin hypercube design

$$Q^2 = 1 - \text{SMSE}.$$

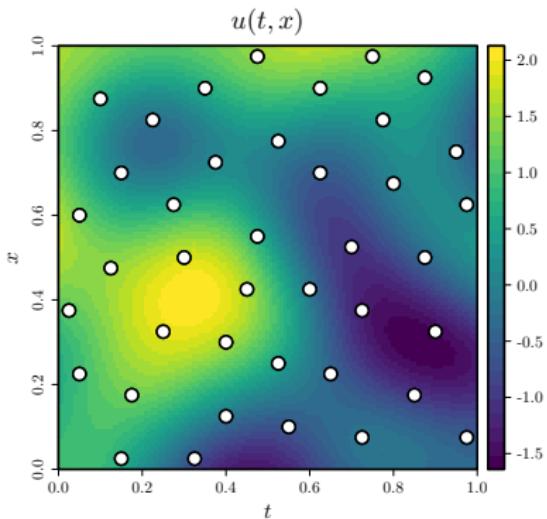
## GP-mRNA model: toy example – inference of $\mathbf{Y}$ using $\mathbf{U}$



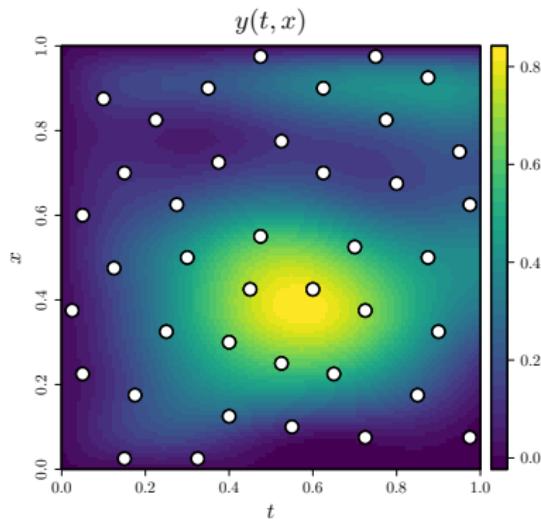
- o 40 conditioning points from a maximin Latin hypercube design

$$Q^2 = 1 - \text{SMSE}.$$

## GP-mRNA model: toy example – inference using $\mathbf{U}$ and $\mathbf{Y}$



$$(a) Q^2 = 1.000$$



$$(b) Q^2 = 1.000$$

- o 40 conditioning points from a maximin Latin hypercube design

$$Q^2 = 1 - \text{SMSE}.$$

## GP-Protein model

- Now, we can assume  $\mathbf{Y} \sim \mathcal{GP}(0, k_{\mathbf{y}, \mathbf{y}})$  and compute the process  $\mathbf{U}$ .
- The *reaction-diffusion equation* can be written as:

$$\mathbf{U}(x, t) = \frac{1}{S} \left[ \frac{\partial \mathbf{Y}(x, t)}{\partial t} + \lambda \mathbf{Y}(x, t) - D \frac{\partial^2 \mathbf{Y}(x, t)}{\partial x^2} \right]. \quad (8)$$

- Note that  $k_{\mathbf{y}, \mathbf{y}}$  has to be a differentiable kernel, e.g.,

$$k_{\mathbf{y}, \mathbf{y}}((x, t), (x', t')) = \exp \left( -\frac{(x - x')^2}{\theta_x^2} \right) \exp \left( -\frac{(t - t')^2}{\theta_t^2} \right).$$

- Covariance  $k_{\textcolor{red}{u},\textcolor{red}{u}}((x, t), (x', t'))$ :

$$\text{cov} \{ \textcolor{red}{U}(x, t), \textcolor{red}{U}(x', t') \}$$

$$\begin{aligned} &= \frac{1}{S^2} \mathbb{E} \left\{ \left[ \frac{\partial Y(x, t)}{\partial t} + \lambda Y(x, t) - D \frac{\partial^2 Y(x, t)}{\partial x^2} \right] \left[ \frac{\partial Y(x', t')}{\partial t'} + \lambda Y(x', t') - D \frac{\partial^2 Y(x', t')}{\partial x'^2} \right] \right\} \\ &= \frac{1}{S^2} \mathcal{L} \circ k_{\textcolor{green}{y},\textcolor{green}{y}}((x, t), (x', t')), \end{aligned}$$

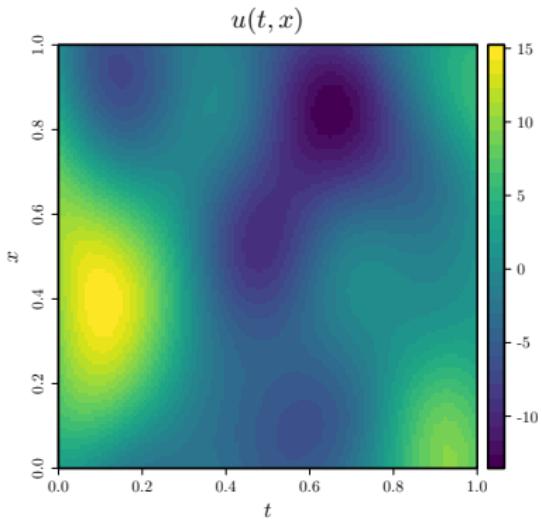
with

$$\mathcal{L} = \frac{\partial^2}{\partial t \partial t'} + \lambda \left( \lambda + \frac{\partial}{\partial t} + \frac{\partial}{\partial t'} - D \frac{\partial^2}{\partial x'^2} \right) - D \left( \lambda \frac{\partial^2}{\partial x^2} + \frac{\partial^3}{\partial t \partial x'^2} + \frac{\partial^3}{\partial x^2 \partial t} - D \frac{\partial^4}{\partial x^2 \partial x'^2} \right).$$

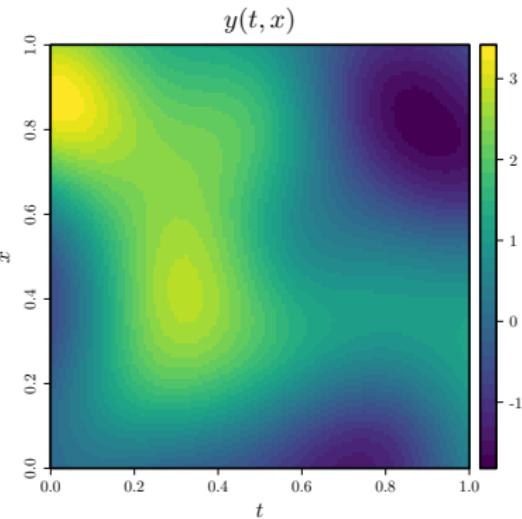
- Cross-covariance  $k_{\textcolor{green}{y},\textcolor{red}{u}}((x, t), (x', t'))$ :

$$\text{cov} \{ \textcolor{green}{Y}(x, t), \textcolor{red}{U}(x', t') \} = \frac{1}{S} \left[ \lambda + \frac{\partial}{\partial t'} - D \frac{\partial^2}{\partial x'^2} \right] \circ k_{\textcolor{green}{y},\textcolor{green}{y}}((x, t), (x', t')).$$

## GP-Protein model: toy example – functions to approximate



(a) mRNA

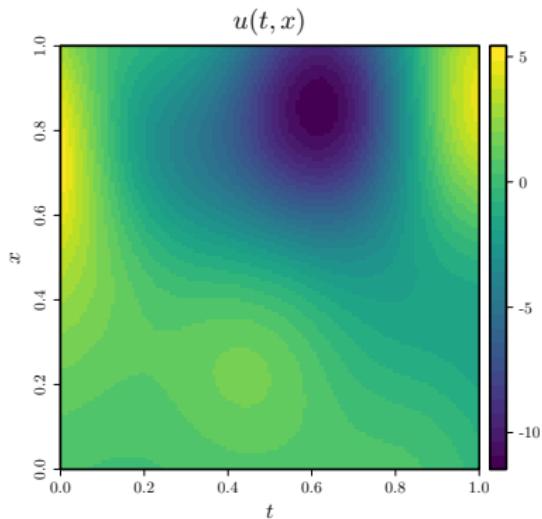


(b) Protein

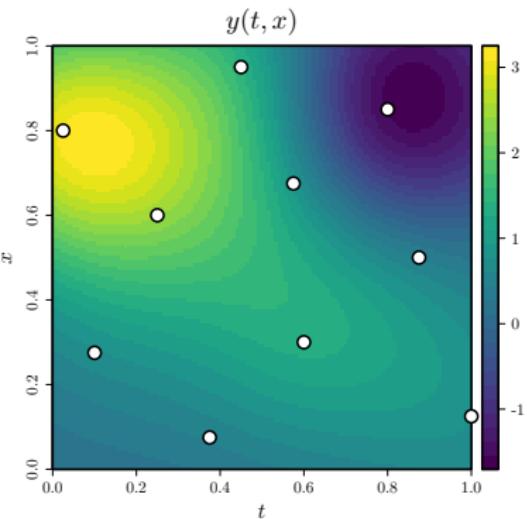
Sample from the GP-Protein model

$$Q^2 = 1 - \text{SMSE}.$$

## GP-Protein model: toy example – inference of $\mathbf{U}$ using $\mathbf{Y}$



$$(a) Q^2 = 0.373$$

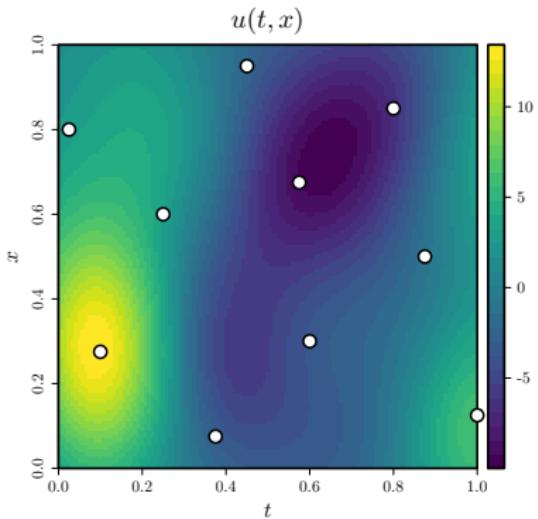


$$(b) Q^2 = 0.714$$

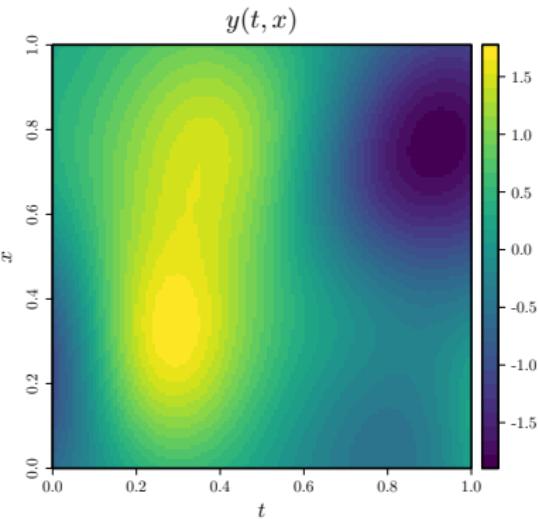
- 10 conditioning points from a maximin Latin hypercube design

$$Q^2 = 1 - \text{SMSE}.$$

## GP-Protein model: toy example – inference of $\mathbf{Y}$ using $\mathbf{U}$



(a)  $Q^2 = 0.756$

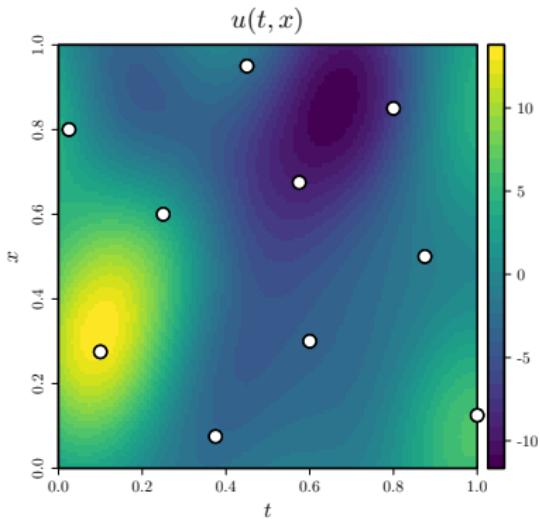


(b)  $Q^2 = 0.467$

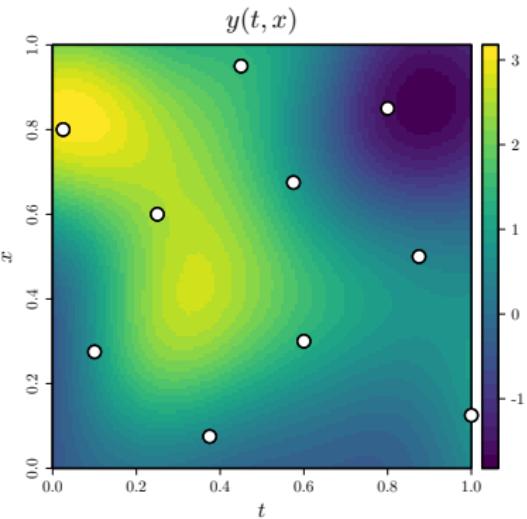
- 10 conditioning points from a maximin Latin hypercube design

$$Q^2 = 1 - \text{SMSE}.$$

## GP-Protein model: toy example – inference using $\mathbf{U}$ and $\mathbf{Y}$



(a)  $Q^2 = 0.893$

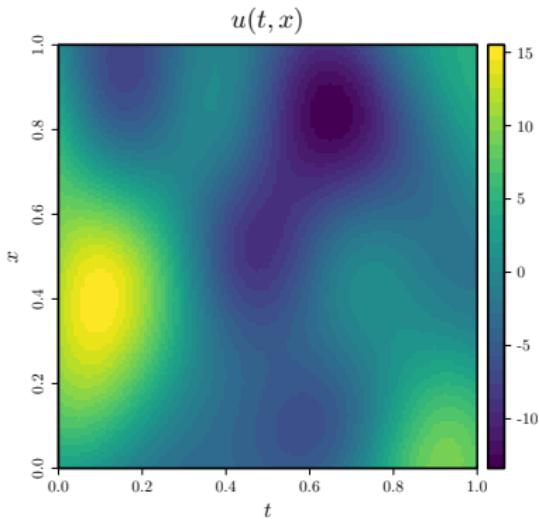


(b)  $Q^2 = 0.948$

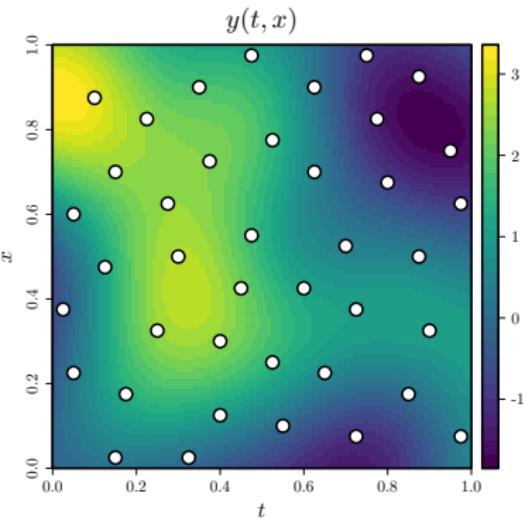
- 10 conditioning points from a maximin Latin hypercube design

$$Q^2 = 1 - \text{SMSE}.$$

## GP-Protein model: toy example – inference of $\mathbf{U}$ using $\mathbf{Y}$



(a)  $Q^2 = 0.990$

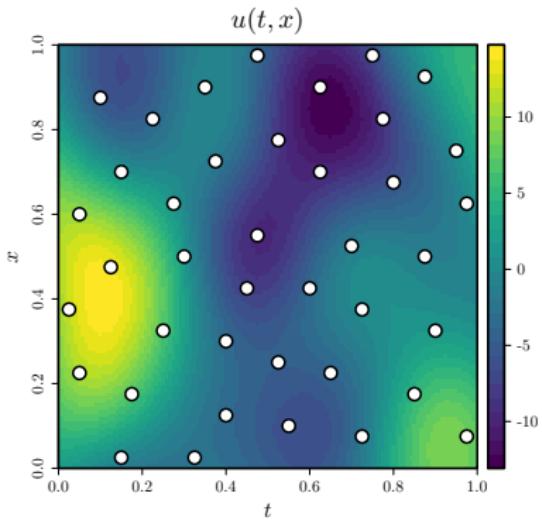


(b)  $Q^2 = 0.999$

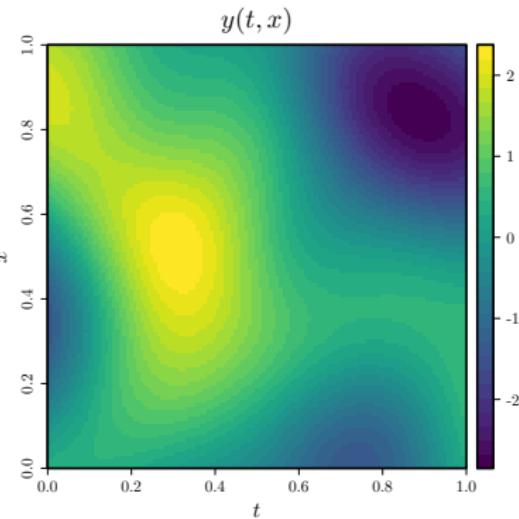
- o 40 conditioning points from a maximin Latin hypercube design

$$Q^2 = 1 - \text{SMSE}.$$

## GP-Protein model: toy example – inference of $\mathbf{Y}$ using $\mathbf{U}$



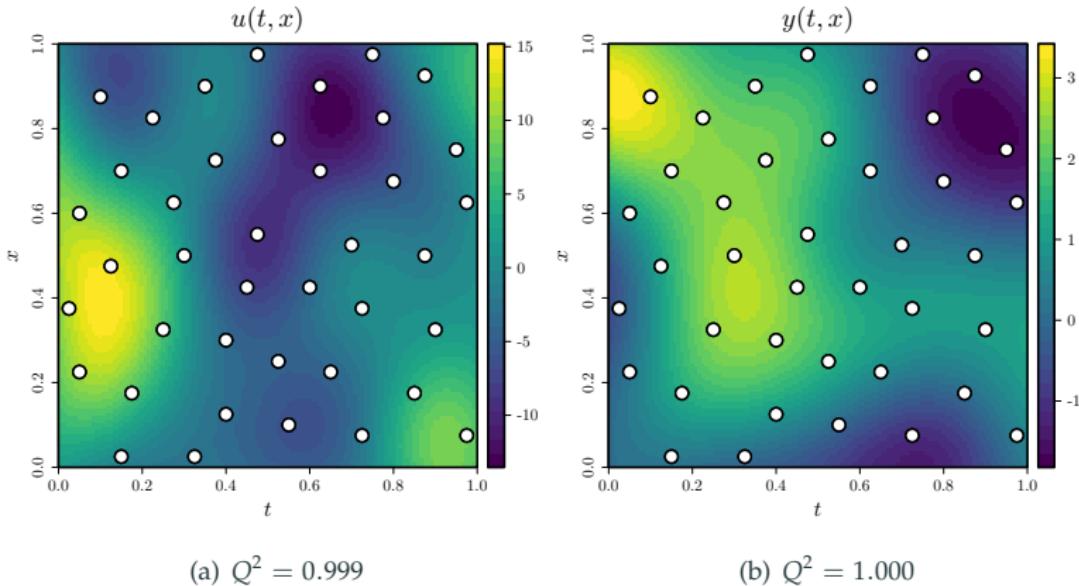
$$(a) Q^2 = 0.997$$



$$(b) Q^2 = 0.656$$

- 40 conditioning points from a maximin Latin hypercube design

$$Q^2 = 1 - \text{SMSE}.$$



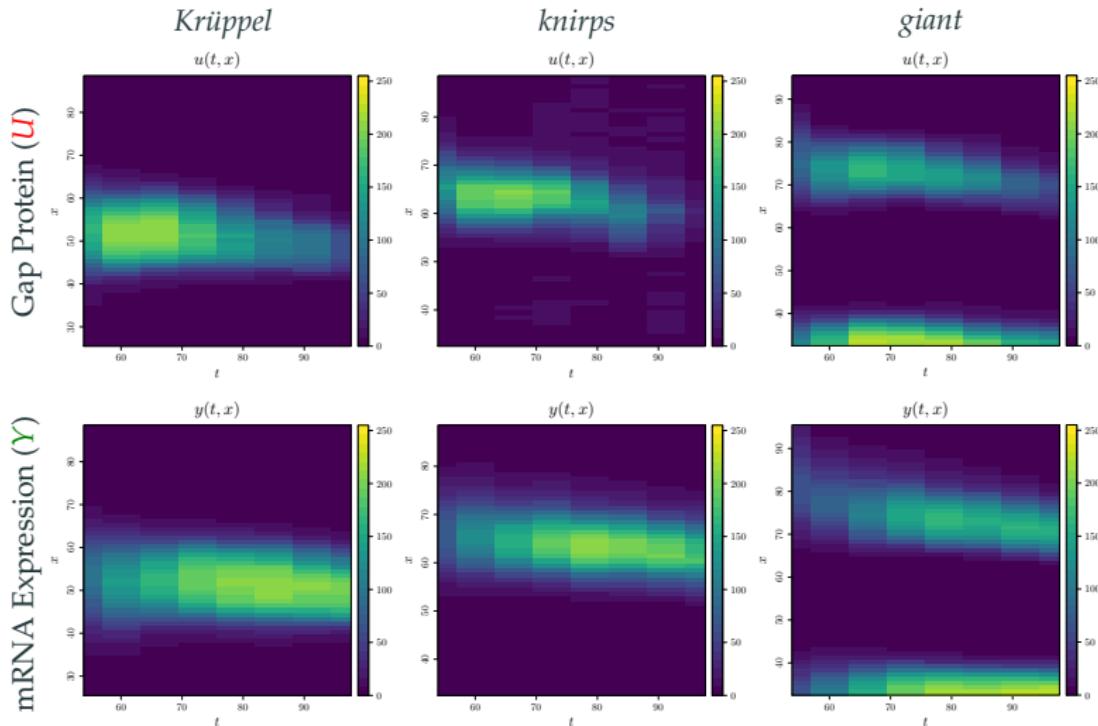
- o 40 conditioning points from a maximin Latin hypercube design

$$Q^2 = 1 - \text{SMSE}.$$

## Biological application

---

# Biological application: dataset



Gap gene mRNA expression data from [Becker et al., 2013].

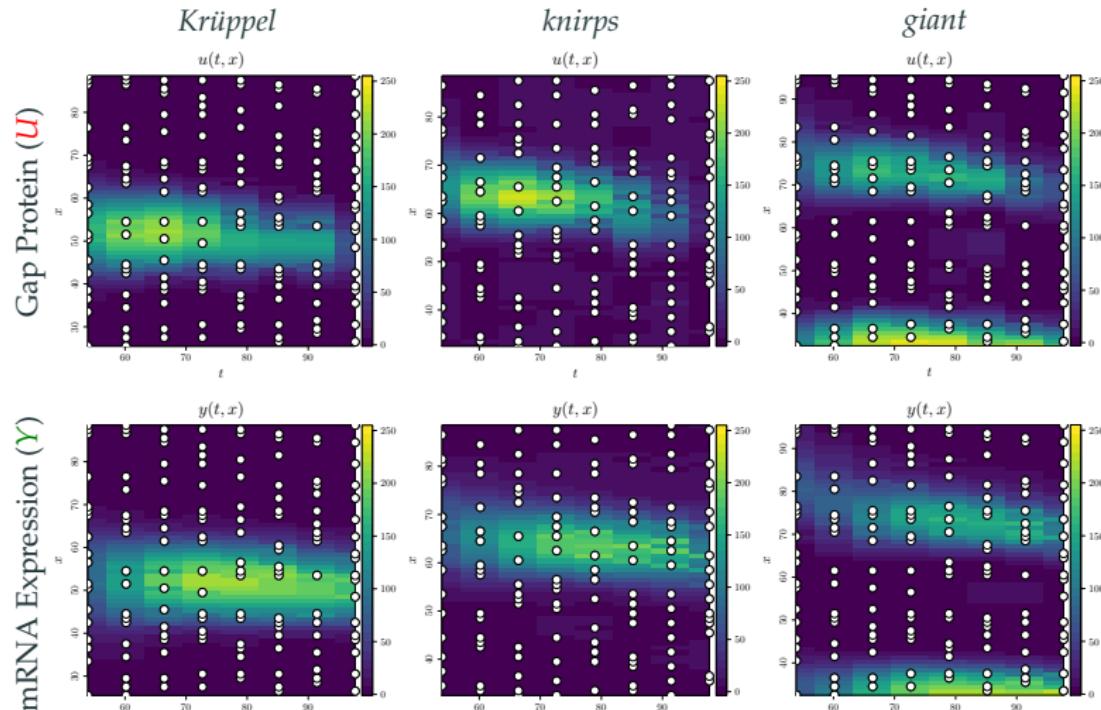
# Biological application: prediction assessment

Prediction assessment ( $Q^2[\%]$ ) for 10 replicates using 30% of data for training models.

Trunk	GP-mRNA		GP-Protein	
Gap	mRNA	Gap Protein	mRNA	Gap Protein
Gene	$\mu \pm \sigma$	$\mu \pm \sigma$	$\mu \pm \sigma$	$\mu \pm \sigma$
Training data only from the gap protein concentration ( $Y$ )				
<i>Krüppel</i>	<b>90.5 ± 2.0</b>	<b>90.8 ± 0.6</b>	<b>92.0 ± 0.6</b>	<b>90.6 ± 0.5</b>
<i>knirps</i>	<b>81.1 ± 2.5</b>	<b>88.7 ± 0.8</b>	<b>77.6 ± 4.7</b>	<b>88.6 ± 0.7</b>
<i>giant</i>	<b>91.2 ± 1.9</b>	<b>92.3 ± 0.6</b>	<b>93.2 ± 1.3</b>	<b>92.8 ± 0.5</b>
Training data only from the mRNA concentration ( $U$ )				
<i>Krüppel</i>	<b>86.7 ± 1.4</b>	<b>97.5 ± 0.7</b>	<b>84.0 ± 2.1</b>	<b>60.6 ± 1.2</b>
<i>knirps</i>	<b>82.9 ± 2.1</b>	<b>86.7 ± 1.3</b>	<b>80.7 ± 3.2</b>	<b>55.2 ± 12.7</b>
<i>giant</i>	<b>91.2 ± 0.7</b>	<b>93.9 ± 0.3</b>	<b>88.2 ± 3.1</b>	<b>84.3 ± 1.7</b>
Training data from both biological quantities ( $U, Y$ )				
<i>Krüppel</i>	<b>96.8 ± 0.5</b>	<b>97.9 ± 0.3</b>	<b>98.6 ± 0.6</b>	<b>99.6 ± 0.2</b>
<i>knirps</i>	<b>91.2 ± 2.9</b>	<b>95.0 ± 0.7</b>	<b>94.5 ± 3.5</b>	<b>99.4 ± 0.3</b>
<i>giant</i>	<b>95.2 ± 1.4</b>	<b>96.2 ± 0.6</b>	<b>97.7 ± 1.7</b>	<b>99.3 ± 0.2</b>

\* Training points were chosen from a maximin Latin hypercube design.

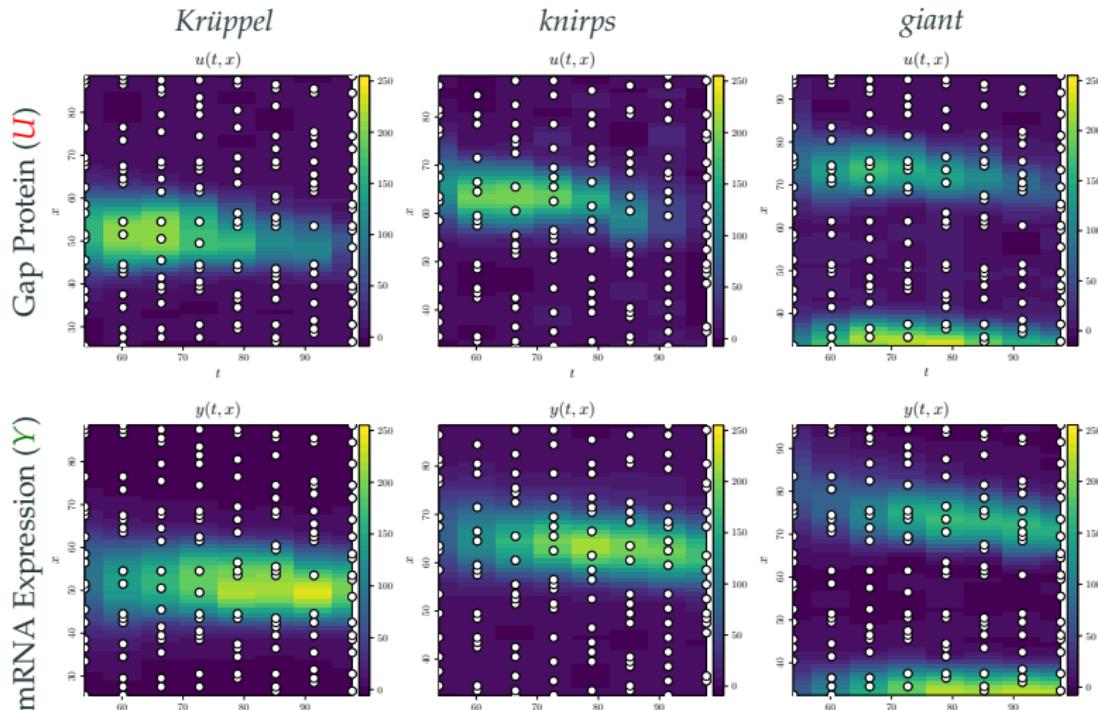
# Biological application: GP-mRNA



○ training data

Prediction using 30% of the dataset from both biological quantities.

# Biological application: GP-Protein



## **Remarks and conclusions**

---

## Remarks and conclusions

- 2 physically-inspired GP models were studied for the post-transcriptional regulation in Drosophila.
  - They rely on a reaction-diffusion equation.
  - Mechanistic parameters are encoded into kernels.
  - Difference between models lies on whether GP priors are placed.
- Choice of the model may depend on the availability and nature of data:
  - Placing priors over unobserved processes.
  - Initial and boundary conditions.
  - Regularity assumptions (e.g. smoothness).
- Both frameworks can be applied to other (linear) dynamical systems, e.g.:
  - Mass-Spring-Damper equation [Álvarez et al., 2009],
  - Wave equation [Alvarado et al., 2014],
  - Heat equation [Raissi et al., 2017].

## References

- P. Alvarado, M. Álvarez, G. Daza, and A. Orozco. An LFM for describing electric propagation in deep brain stimulation: A simulation study. In *EMBC*. 2014.
- M. Álvarez, D. Luengo, and N. Lawrence. Latent force models (LFM). In *AISTATS*, 2009.
- K. Becker, A. Balsa, D. Cicin, A. Hoermann, H. Janssens, J. Banga, and J. Jaeger. Reverse-engineering post-transcriptional regulation of gap genes in *Drosophila melanogaster*. *PLoS Computational Biology*, 2013.
- P. Gao, A. Honkela, M. Rattray, and N. Lawrence. GP modelling of latent chemical species: applications to inferring transcription factor activities. *Bioinformatics*, 2008.
- N. Lawrence, G. Sanguinetti, and M. Rattray. Modelling transcriptional regulation using GPs. In *NeurIPS*. 2007.
- A. F. López-Lopera, N. Durrande, and M. Álvarez. Physically-inspired GPs models for post-transcriptional regulation in *Drosophila*. *IEEE/ACM TCBB*, 2021.
- M. Raissi, P. Perdikaris, and G. Karniadakis. Machine learning of linear differential equations using GPs. *Journal of Computational Physics*, 2017.
- C. E. Rasmussen and C. K. I. Williams. *GPs for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.
- J. Vásquez, M. Álvarez, and A. Orozco. LFM for describing transcriptional regulation processes in the embryo development problem for the *Drosophila*. In *EMBC*, 2014.