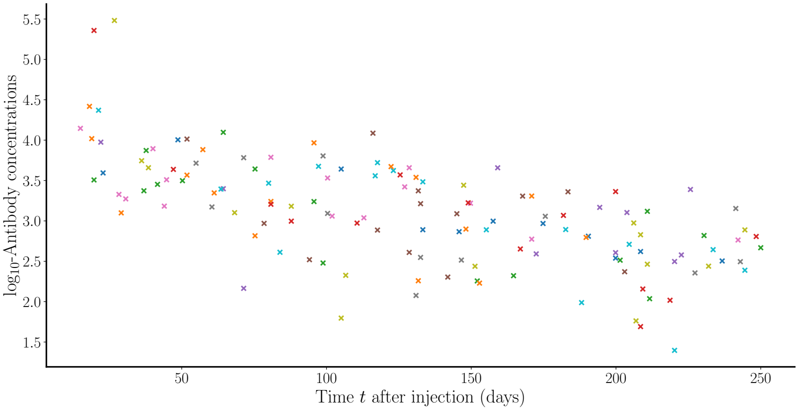
# Gaussian Processes for the inference of partially known mechanistic models used for clinical trial data analysis

Julien Martinelli ¯_(ツ)_/¯

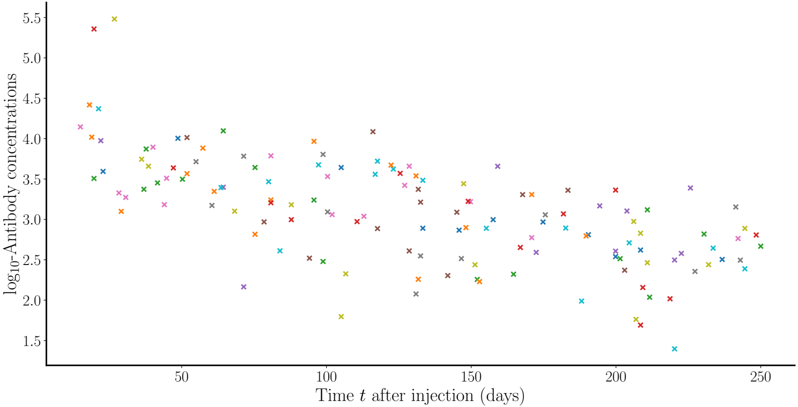
November 13rd, 2023

# Motivation



5.5
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
$\log_{10}$-Antibody concentrations
50
100
150
200
250
Time $t$ after injection (days)

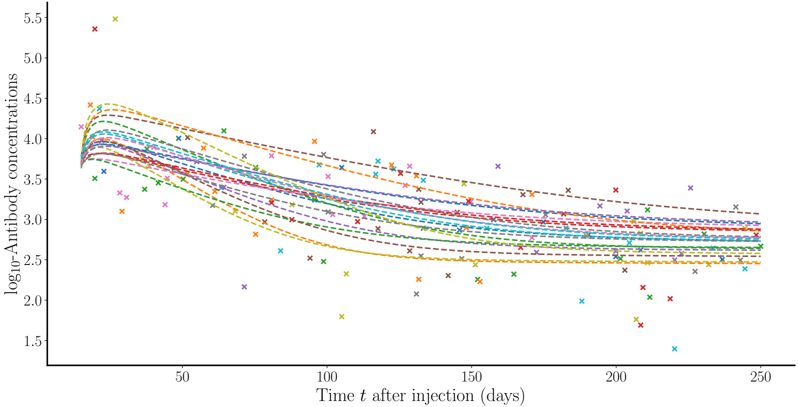
# Motivation




Obs $y_i(t) = f(t; \theta_i) + \varepsilon$ for a **known mechanistic model** $f$ and $1 \leq i \leq M$ patients.

$$f(t; \theta_i) = e^{-\delta_{Ab,i}(t-t_0)} Ab_{0,i} + \phi_{S,i} \frac{e^{-\delta_{S,i}(t-t_0)} - e^{-\delta_{Ab,i}(t-t_0)}}{\delta_{Ab,i} - \delta_{S,i}} + \phi_L \frac{e^{-\delta_L(t-t_0)} - e^{-\delta_{Ab,i}(t-t_0)}}{\delta_{Ab,i} - \delta_L}$$

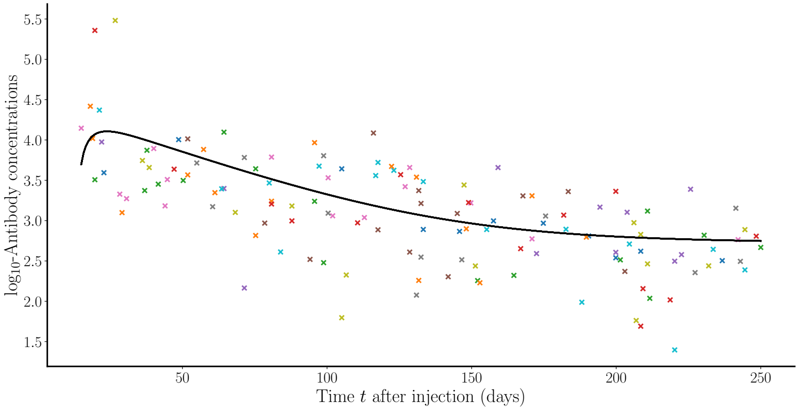
# Motivation




Latent trajectories $f(t;\theta_i)$ with **unkwown parameters** $\theta_i = \theta + b_i$ (mixed-effects)

$$f(t;\theta_i) = e^{-\delta_{Ab,i}(t-t_0)} Ab_{0,i} + \phi_{S,i} \frac{e^{-\delta_{S,i}(t-t_0)} - e^{-\delta_{Ab,i}(t-t_0)}}{\delta_{Ab,i} - \delta_{S,i}} + \phi_L \frac{e^{-\delta_L(t-t_0)} - e^{-\delta_{Ab,i}(t-t_0)}}{\delta_{Ab,i} - \delta_L}$$

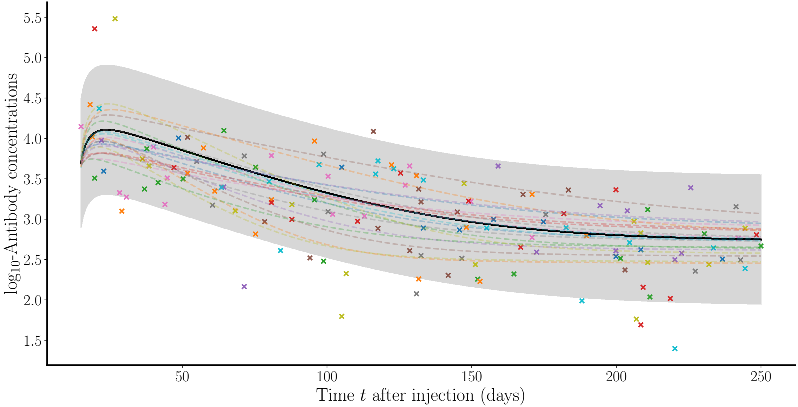
# Motivation




We want to say something about the population mean behavior characterized by $\theta$.

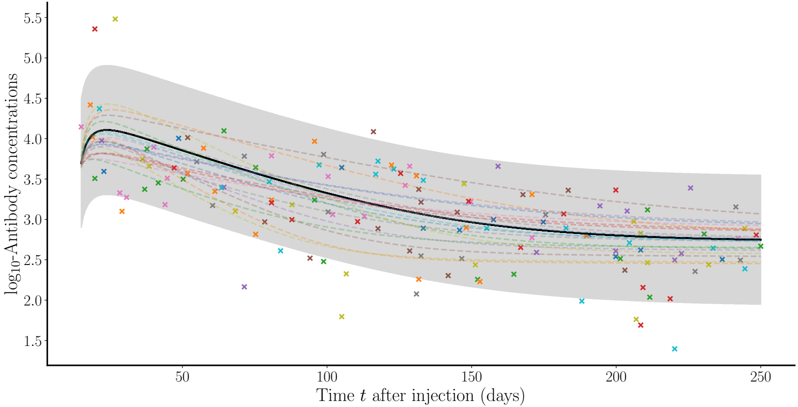
$$f(t;\theta_i) = e^{-\delta_{Ab,i}(t-t_0)} Ab_{0,i} + \phi_{S,i} \frac{e^{-\delta_{S,i}(t-t_0)} - e^{-\delta_{Ab,i}(t-t_0)}}{\delta_{Ab,i} - \delta_{S,i}} + \phi_L \frac{e^{-\delta_L(t-t_0)} - e^{-\delta_{Ab,i}(t-t_0)}}{\delta_{Ab,i} - \delta_L}$$

# Motivation




While being able to incorporate **prior information** about $\theta$ and $\{b_i\}_{i=1}^{M}$, leading to principled uncertainty quantification.

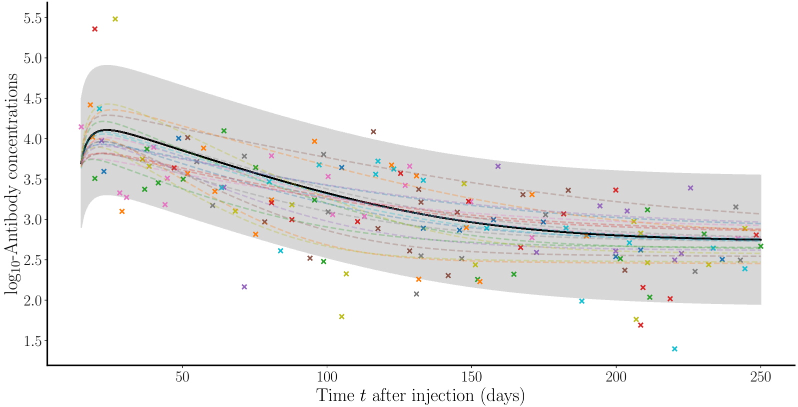
# Motivation




**Can we still do that when $f$ is partially known, or even unknown?**

$f_i(t) = \mu_0(t) + g_i(t) \iff$ learn **functions** not parameters

# Motivation




**Can we still do that when $f$ is partially known, or even unknown?**

$f_i(t) = \mu_0(t) + g_i(t) \iff$ learn **functions** not parameters

**Answer:** yes (hopefully ¯_(ツ)_/¯), using **Gaussian Processes**

# Outline

1. Gaussian Processes in a nutshell
2. Analogies, extensions
3. Application: learning partially known vector fields from heterogeneous data

# Gaussian processes (GPs)

A GP is a stochastic process acting as a **prior distribution over function spaces**

$$f(x) \sim \mathcal{GP}(m_{\theta_m}(x), k_{\theta_k}(x, x'))$$

$m_{\theta_m}(x) = \mathbb{E}[f(x)]$ is the **mean function**, $k_{\theta_k}(x, x') = \text{Cov}[f(x), f(x')]$ the **kernel**.
(Hyper-)Parameterized by $(\theta_m, \theta_k)$.

# Gaussian processes (GPs)

A GP is a stochastic process acting as a **prior distribution over function spaces**

$$f(x) \sim \mathcal{GP}(m_{\theta_m}(x), k_{\theta_k}(x, x'))$$

$m_{\theta_m}(x) = \mathbb{E}[f(x)]$ is the **mean function**, $k_{\theta_k}(x, x') = \text{Cov}[f(x), f(x')]$ the **kernel**.
(Hyper-)Parameterized by $(\theta_m, \theta_k)$.

GPs generalize the multivariate normal distribution to infinite-dimensional spaces
For any collection of function values $\mathrm{f} = [f(x_1), \dots, f(x_n)]$

$$\mathrm{f} \sim \mathcal{N}(\mathrm{m}, \mathrm{K})$$

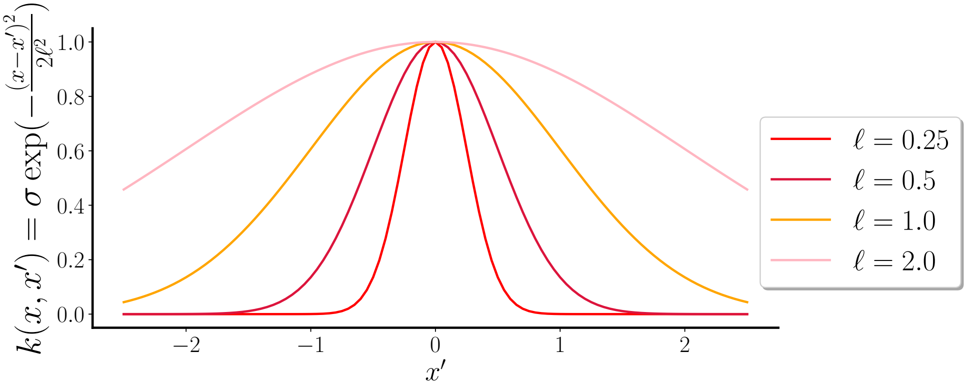
With $\mathrm{m} = [m_{\theta_m}(x_1), \dots, m_{\theta_m}(x_n)]$ and $\mathrm{K} = (k_{\theta_k}(x_i, x_j))_{1 \leq i,j \leq n}$

# Example - Radial Basis Function Kernel

$$\mathsf{Cov}[f(x), f(x')] := k_{\theta_k}(x, x') = \sigma_{\mathsf{amp}} \exp\left(-\frac{(x - x')^2}{2\ell^2}\right) \qquad \theta_k = (\sigma_{\mathsf{amp}}, \ell)$$

# Example - Radial Basis Function Kernel

$$\mathrm{Cov}[f(x), f(x')] := k_{\theta_k}(x, x') = \sigma_{\text{amp}} \exp\left(-\frac{(x - x')^2}{2\ell^2}\right) \qquad \theta_k = (\sigma_{\text{amp}}, \ell)$$




$\sigma_{\text{amp}}$ handles the variance magnitude and $\ell$ how fast correlation decreases

# Animations are always better to understand ¯_(ツ)_/¯

# Nice thing about GPs: posterior predictive available in closed-form

Let $\mathscr{D} = (x_i, y_i)_{i=1}^{n} = (\mathrm{X}, \mathrm{y})$ with $y_i = f(x_i) + \varepsilon$. For a new function value $f_*$ located at $x_*$,

$$f_* | \mathrm{y} \sim \mathscr{N}(m_{\theta_m}(x_* | \mathscr{D}), \sigma^2(x_* | \mathscr{D}))$$

$$m(x_* | \mathscr{D}) = m_{\theta_m}(x_*) + k_{\theta_k}(x_*, \mathrm{X})^T (\mathrm{K} + \sigma^2_{\text{noise}} I)^{-1} (\mathrm{y} - \mathrm{m})$$

$$\sigma^2(x_* | \mathscr{D}) = k_{\theta_k}(x_*, x_*) - k_{\theta_k}(x_*, \mathrm{X})^T (\mathrm{K} + \sigma^2_{\text{noise}} I)^{-1} k_{\theta_k}(\mathrm{X}, x_*)$$

Where $k_{\theta_k}(x_*, \mathrm{X})^T = [k_{\theta_k}(x_*, x_1), \dots, k_{\theta_k}(x_*, x_n)]$.

# Nice thing about GPs: posterior predictive available in closed-form

Let $\mathscr{D} = (x_i, y_i)_{i=1}^n = (\mathrm{X}, \mathrm{y})$ with $y_i = f(x_i) + \varepsilon$. For a new function value $f_*$ located at $x_*$,

$$f_*|\mathrm{y} \sim \mathcal{N}(m_{\theta_m}(x_*|\mathscr{D}), \sigma^2(x_*|\mathscr{D}))$$
$$m(x_*|\mathscr{D}) = m_{\theta_m}(x_*) + k_{\theta_k}(x_*, \mathrm{X})^T(\mathrm{K} + \sigma^2_{\text{noise}} I)^{-1}(\mathrm{y} - \mathrm{m})$$
$$\sigma^2(x_*|\mathscr{D}) = k_{\theta_k}(x_*, x_*) - k_{\theta_k}(x_*, \mathrm{X})^T(\mathrm{K} + \sigma^2_{\text{noise}} I)^{-1} k_{\theta_k}(\mathrm{X}, x_*)$$

Where $k_{\theta_k}(x_*, \mathrm{X})^T = [k_{\theta_k}(x_*, x_1), \dots, k_{\theta_k}(x_*, x_n)]$.

**Hyperparameters** $(\theta_m, \theta_k, \sigma_{\text{noise}})$ learned through marginal likelihood maximization.

# Nice thing about GPs: posterior predictive available in closed-form

Let $\mathscr{D} = (x_i, y_i)_{i=1}^n = (\mathrm{X}, \mathrm{y})$ with $y_i = f(x_i) + \varepsilon$. For a new function value $f_*$ located at $x_*$,

$$f_*|\mathrm{y} \sim \mathcal{N}(m_{\theta_m}(x_*|\mathscr{D}), \sigma^2(x_*|\mathscr{D}))$$

$$m(x_*|\mathscr{D}) = m_{\theta_m}(x_*) + k_{\theta_k}(x_*, \mathrm{X})^T(\mathrm{K} + \sigma^2_{\text{noise}} I)^{-1}(\mathrm{y} - \mathrm{m})$$

$$\sigma^2(x_*|\mathscr{D}) = k_{\theta_k}(x_*, x_*) - k_{\theta_k}(x_*, \mathrm{X})^T(\mathrm{K} + \sigma^2_{\text{noise}} I)^{-1} k_{\theta_k}(\mathrm{X}, x_*)$$

Where $k_{\theta_k}(x_*, \mathrm{X})^T = [k_{\theta_k}(x_*, x_1), \ldots, k_{\theta_k}(x_*, x_n)]$.

**Hyperparameters** $(\theta_m, \theta_k, \sigma_{\text{noise}})$ learned through marginal likelihood maximization.

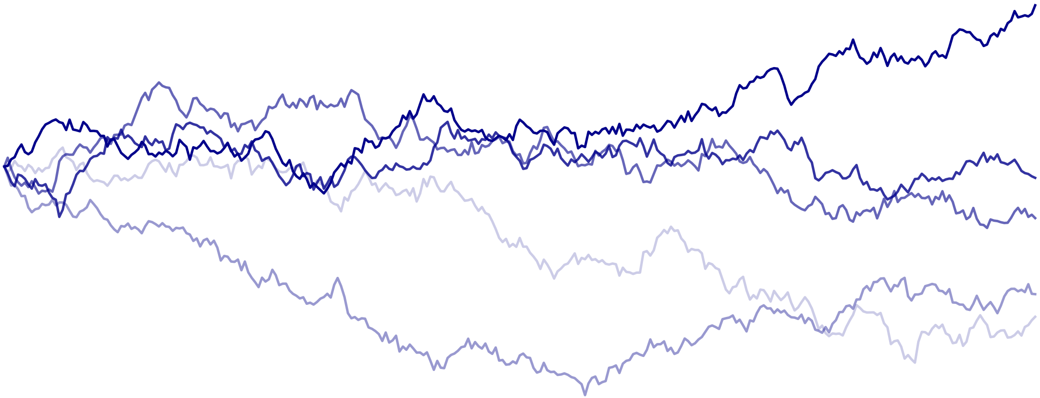
For a zero-mean prior $m$, the posterior mean can be written as

$$m(x_*|\mathscr{D}) = \sum_{i=1}^{n} \alpha_i k_{\theta_m}(x_*, x_i)$$

with $\alpha = (K + \sigma^2_{\text{noise}} I)^{-1}\mathrm{y}$. **GPs: probabilistic counterpart of kernel methods.**

# Animations are always better to understand ¯_(ツ)_/¯

# You probably used GPs at some point without even noticing

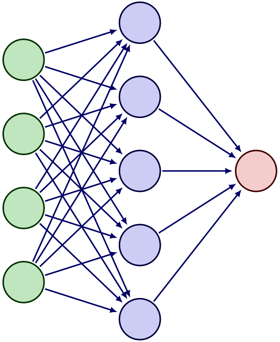


Brownian Motion is a GP where the kernel is $k(x, x') = \min(x, x')$

# You probably used GPs at some point without even noticing

In the infinite number of neurons, 1-layer Neural Networks can be written as GPs

$$f(x) = b + \sum_{l=1}^{L} v_l s(w_l x + b_l)$$



Under the assumption of i.i.d Gaussian weights $\{v_l\}_l$, $\{w_l\}_l$ and biases $b$, $\{b_l\}_l$,

$$\mathbb{E}[f(x)] = 0 \text{ and } \mathrm{Cov}[f(x), f(x')] = \sigma_b^2 + \sigma_v^2 L \mathbb{E}_{w,b}[s(wx + b)s(wx' + b)]$$

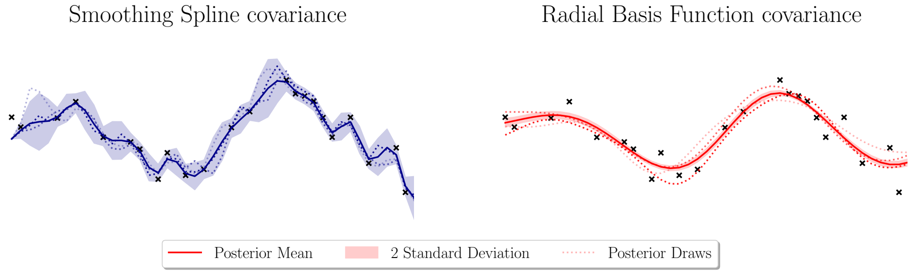
Scale the output variance with $\sigma_v^2 = \frac{\omega}{L}$ and apply CLT to get the final kernel.

# You probably used GPs at some point without even noticing

The cubic smoothing spline estimate $\hat{f}$ of the function $f$ is also a GP

$$\underset{\hat{f}}{\operatorname{argmin}} \sum_{i=1}^{n} (\hat{f}(x_i) - y_i)^2 + \lambda \int_0^1 \hat{f}''(x)^2 \mathrm{d}x$$

$$\iff \hat{f} \sim \mathcal{GP}\left(0, \sigma_{\text{amp}} \left( \frac{|x - x'|}{2} \min(x, x')^2 + \frac{\min(x, x')^3}{3} \right) + \sigma_{\text{noise}} \delta_{xx'} \right)$$



# You probably used GPs at some point without even noticing

**Kalman Filters** are a particular type of GPs equipped with the Markov property

Classical GP regression problem **(⋆)**

$$U(t) \sim \mathscr{GP}(0, k(t, t'))$$
$$Y_t = U(t_k) + \xi_k$$

# You probably used GPs at some point without even noticing

**Kalman Filters** are a particular type of GPs equipped with the Markov property
Classical GP regression problem (⋆)

$$U(t) \sim \mathscr{GP}(0, k(t,t'))$$
$$Y_t = U(t_k) + \xi_k$$

Will lead to the same solution as the smoothing problem (⋆⋆)

$$d\bar{U}(t) = A\bar{U}(t) + BdW(t)$$
$$U(t_0) = U_0 \sim \mathscr{N}(0, P_0)$$
$$U = H\bar{U}$$

(⋆): you provide the kernel $k$. (⋆⋆): you provide the SDE matrices $A, B$.

# Extensions

## Nonstationary kernels

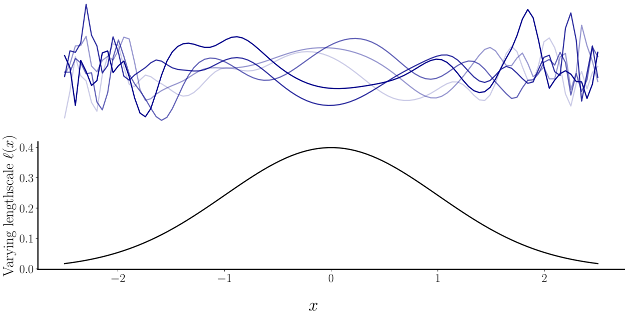
Classical kernels $k_{\theta_k}(x, x')$ can be written $k_{\theta_k}(h)$ with $h = (x - x')$:
$\implies$ output correlation only depends on the distance between inputs, not their location, **stationnarity**: $p(x_1, \dots, x_n) = p(x_{1+\tau}, \dots, x_{n+\tau})$.

# Extensions

## Nonstationary kernels

Classical kernels $k_{\theta_k}(x, x')$ can be written $k_{\theta_k}(h)$ with $h = (x - x')$:
$\implies$ output correlation only depends on the distance between inputs, not their location, **stationnarity**: $p(x_1, \dots, x_n) = p(x_{1+\tau}, \dots, x_{n+\tau})$.

E.g. make hyperparameters a function of the input $k(x, x') = \sigma_{\text{amp}} \exp\left(-\frac{1}{2}\frac{(x-x')^2}{\ell(x)^2+\ell(x')^2}\right)$



# Extensions

## Multitask GPs for multiple outputs

Extend the input space with a *patient dimension*: $x \leftarrow (x, i)$ and define

$$k((x,i),(x',i')) = k_\theta(x,x')k_{\mathsf{task}}(i,i').$$

Typically, $k_{\text{task}}$ is the inter-patient covariance matrix, estimated from data.

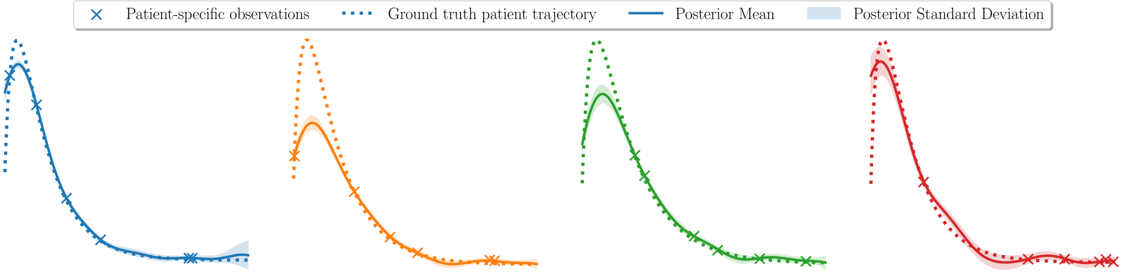
# Extensions

## Multitask GPs for multiple outputs

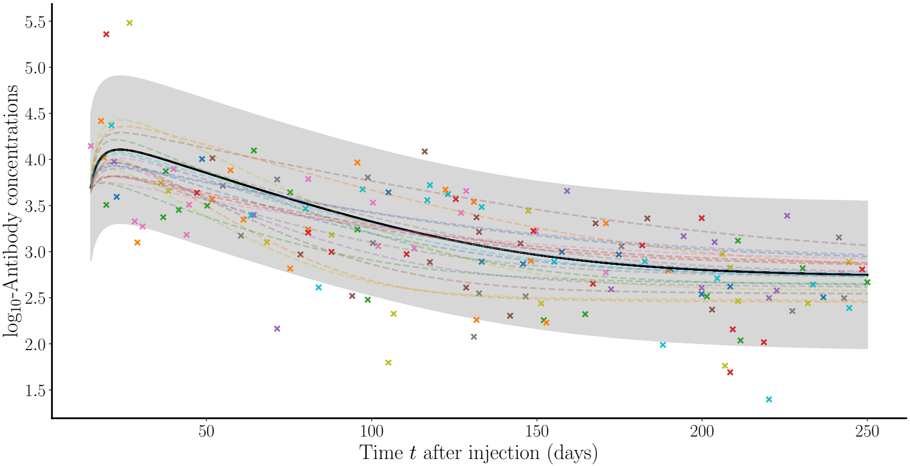
Extend the input space with a *patient dimension*: $x \leftarrow (x, i)$ and define

$$k((x,i),(x',i')) = k_\theta(x,x')k_{\text{task}}(i,i').$$

Typically, $k_{\text{task}}$ is the inter-patient covariance matrix, estimated from data.



# Back to the original problem




$$y_i(t) = \mu_0(t) + f_i(t) + \varepsilon_i(t), \qquad i = 1, \dots, M$$

# MAGMA - Multi task Gaussian processes with common mean

Arthur Leroy, Pierre Latouche, Benjamin Guedj and Servane Gey, 2022

$$
\begin{aligned}
y_i(t) &= \mu_0(t) + f_i(t) + \varepsilon_i(t) \\
\mu_0(\cdot) &\sim \mathcal{GP}(m_0(\cdot), k_{\theta_0}(\cdot,\cdot)) \\
f_i(\cdot) &\sim \mathcal{GP}(0, c_{\theta_i}(\cdot,\cdot)) \\
\varepsilon_i(\cdot) &\sim \mathcal{N}(0, \sigma^2_{\text{noise},i} I)
\end{aligned}
$$

# MAGMA - Multi task Gaussian processes with common mean

Arthur Leroy, Pierre Latouche, Benjamin Guedj and Servane Gey, 2022

$$y_i(t) = \mu_0(t) + f_i(t) + \varepsilon_i(t)$$
$$\mu_0(\cdot) \sim \mathcal{GP}(m_0(\cdot), k_{\theta_0}(\cdot,\cdot))$$
$$f_i(\cdot) \sim \mathcal{GP}(0, c_{\theta_i}(\cdot,\cdot))$$
$$\varepsilon_i(\cdot) \sim \mathcal{N}(0, \sigma^2_{\text{noise},i} I)$$

Assumptions:

- $f_i$'s independent, $\varepsilon_i$'s independent
- $\forall i, \mu_0, f_i, \varepsilon_i$ are independent

# MAGMA - Multi task Gaussian processes with common mean

Arthur Leroy, Pierre Latouche, Benjamin Guedj and Servane Gey, 2022

$$y_i(t) = \mu_0(t) + f_i(t) + \varepsilon_i(t)$$
$$\mu_0(\cdot) \sim \mathcal{GP}(m_0(\cdot), k_{\theta_0}(\cdot,\cdot))$$
$$f_i(\cdot) \sim \mathcal{GP}(0, c_{\theta_i}(\cdot,\cdot))$$
$$\varepsilon_i(\cdot) \sim \mathcal{N}(0, \sigma^2_{\text{noise},i} I)$$

Assumptions:

- $f_i$'s independent, $\varepsilon_i$'s independent
- $\forall i, \mu_0, f_i, \varepsilon_i$ are independent

$\implies$ $\{y_i|\mu_0\}_i$ **are independent**

$$\mathrm{y}_i(\mathrm{t}_i)|\mu_0(\mathrm{t}_i) \sim \mathcal{N}\left(\mathrm{y}_i; \mu_0(\mathrm{t}_i), \Psi^{\mathrm{t}}_{\theta_i,\sigma^2_{\text{noise},i}}\right)$$

$m_0$ is the (hyper)-prior mean, and encodes **mechanistic knowledge**.
It can be parametrized as well.

# Population mean *a posteriori* distribution

Hyperparameters: $\Theta = (\theta_0, \{\theta_i\}_i, \{\sigma^2_{\text{noise,i}}\}_i)$. Assuming for simplicity $t_i = t_{i'} = t$,

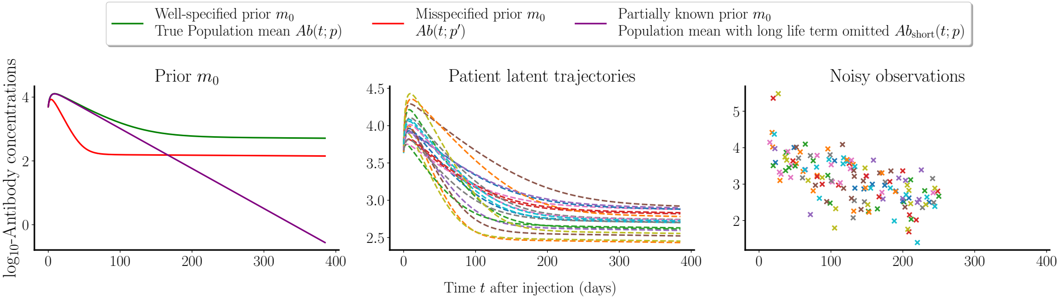
$$p(\mu_0(t) | \{y_i\}_i, \Theta) = \mathcal{N}(\hat{m}_0(t), \hat{K}^t)$$

$$\hat{K} = \left( {K^t_{\theta_0}}^{-1} + \sum_{i=1}^{M} {\Psi^t_{\theta_i, \sigma^2_{\text{noise},i}}}^{-1} \right)^{-1}$$

$$\hat{m}_0(t) = \hat{K}^t \left( {K^t_{\theta_0}}^{-1} m_0(t) + \sum_{i=1}^{M} {\Psi^t_{\theta_i, \sigma^2_{\text{noise},i}}}^{-1} y_i \right)$$

- $\hat{\theta}_0$ and $(\hat{\theta}_i, \hat{\sigma}^2_{\text{noise},i})$ obtained independently **like in usual mixed-effect models**
- We can investigate how $m_0$ and $\hat{m}_0$ differ, what happens if $m_0$ is misspecified...

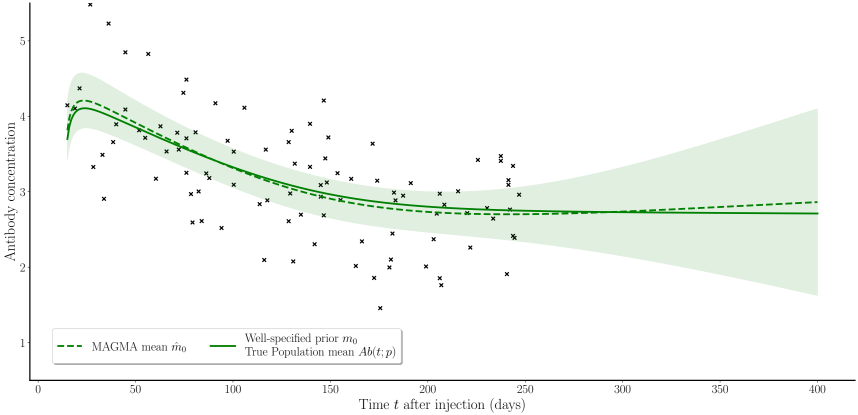
# Case study




$$Ab(t;\theta_i) = e^{-\delta_{Ab,i}(t-t_0)}Ab_{0,i} + \phi_{S,i}\frac{e^{-\delta_{S,i}(t-t_0)} - e^{-\delta_{Ab,i}(t-t_0)}}{\delta_{Ab,i}-\delta_{S,i}} + \phi_L\frac{e^{-\delta_L(t-t_0)} - e^{-\delta_{Ab,i}(t-t_0)}}{\delta_{Ab,i}-\delta_L}$$

- $M = 15$ patients
- $\approx 5-8$ observations per patient at different time points
- No mixed-effect for the long-life parameters $\delta_L$ and $\phi_L$
- Noise is added to the observations

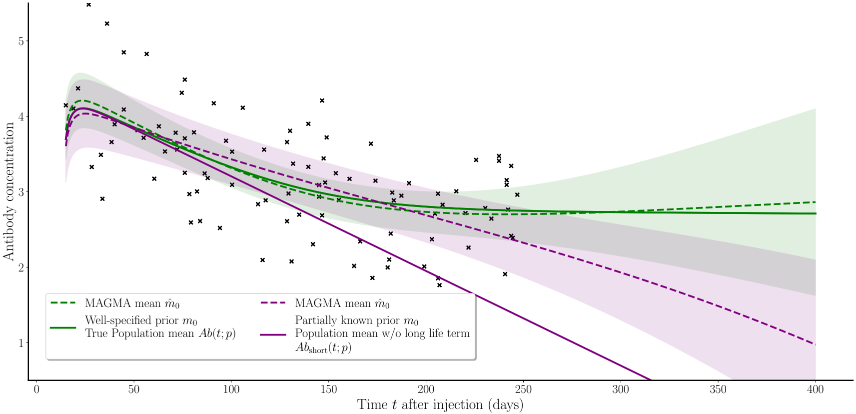
# Comparing learned mean functions $\hat{m}_0$ for different priors $m_0$




$\hat{m}_0$ slightly deviates from the (well-specified) prior $m_0$ to better fit the data

*Post hoc* sanity check of the prior: $m_0$ included in the CIs computed from $\hat{m}_0$

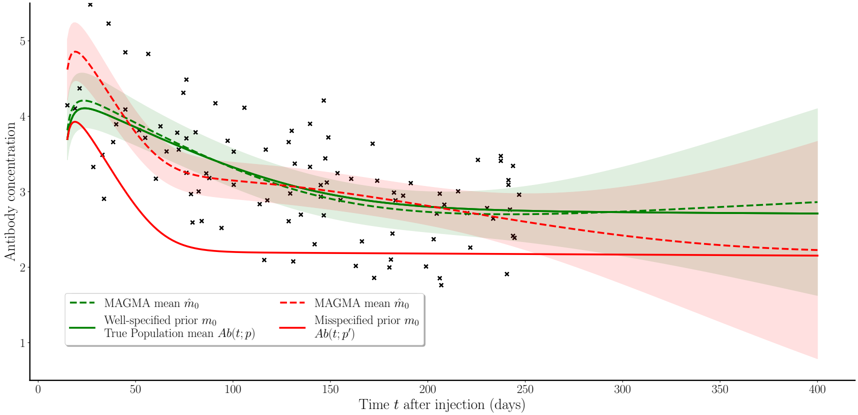
# Comparing learned mean functions $\hat{m}_0$ for different priors $m_0$




$\hat{m}_0$ clearly deviates from the (misspecified) prior $m_0$ to better fit the data

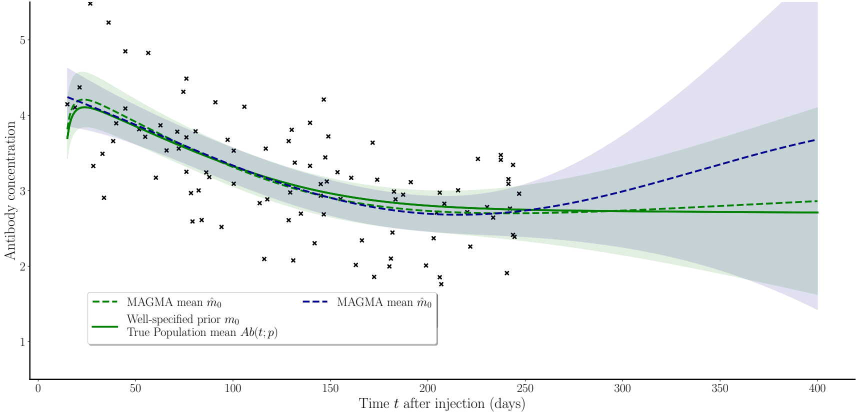
*Post hoc* sanity check: over the long run, $m_0$ without long-life term is **not** included in $\hat{m}_0$'s confidence intervals!

# Comparing learned mean functions $\hat{m}_0$ for different priors $m_0$



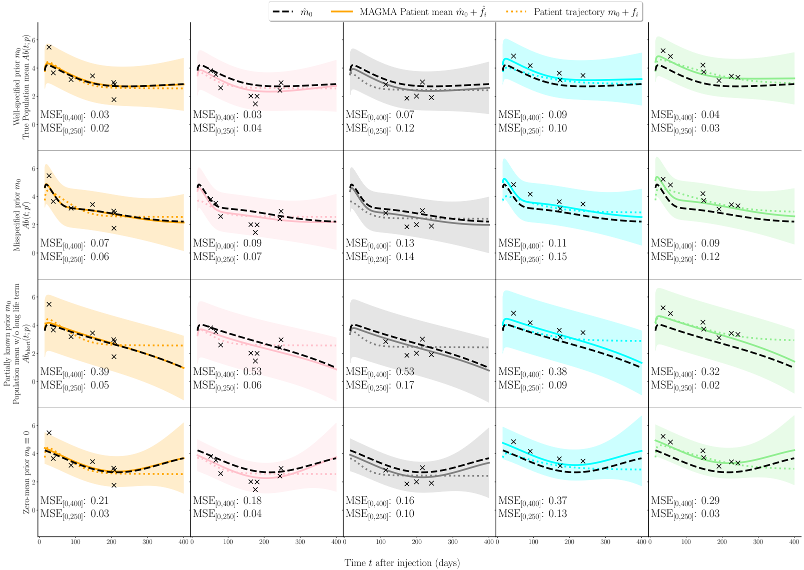

For the misspecified case, $\hat{m}_0$ adapts its mean level
In the presence of data, confidence intervals clearly rule out the misspecified prior

# Comparing learned mean functions $\hat{m}_0$ for different priors $m_0$




When data is abundant, even a zero-mean prior $m_0 \equiv 0$ yields a correct estimate of the population dynamics

# Individual results for 5 out of 15 patients



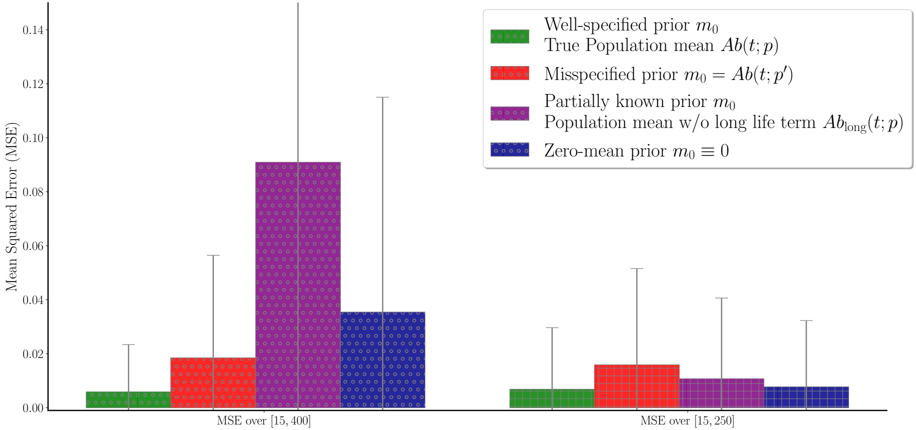

Metric:

$$\int \left(\hat{f}_i(t) - f_i(t)\right)^2 \mathrm{d}t$$

Using ground truth prior mean is best (top row)

Prior without long-life term worst performer (row 3) **over the long run**

# Results averaged over 20 different datasets for $M = 15$ patients




- When considering the whole time horizon, the prior clearly matters
- Over $[15, 250]$, except for misspecified prior, performances are roughly similar

# Roadmap

- Often, the dynamics are defined through ODEs with no closed-form solution

$$\begin{cases} y_i(t) = X_i(t) + \varepsilon_i(t) \\ \dot{X}_i(t) = \mu_0(X_i(t)) + f_i(X_i(t)) \\ X_i(0) = x_{0,i} \end{cases}$$

$p(\mu_0|\mathrm{y})$ **is not Gaussian anymore!** Requires MCMC, Variational Inference...

# Roadmap

- Often, the dynamics are defined through ODEs with no closed-form solution

$$\begin{cases} y_i(t) = X_i(t) + \varepsilon_i(t) \\ \dot{X}_i(t) = \mu_0(X_i(t)) + f_i(X_i(t)) \\ X_i(0) = x_{0,i} \end{cases}$$

  $p(\mu_0|\mathrm{y})$ **is not Gaussian anymore!** Requires MCMC, Variational Inference...

- Handling $D$**-dimensional** ODE systems, $D > 1$
- What if we do not know the full dynamics of **unobserved** variables

# Roadmap

- Often, the dynamics are defined through ODEs with no closed-form solution

$$\begin{cases} y_i(t) = X_i(t) + \varepsilon_i(t) \\ \dot{X}_i(t) = \mu_0(X_i(t)) + f_i(X_i(t)) \\ X_i(0) = x_{0,i} \end{cases}$$

  $p(\mu_0|\mathrm{y})$ **is not Gaussian anymore!** Requires MCMC, Variational Inference...
- Handling **$D$-dimensional** ODE systems, $D > 1$
- What if we do not know the full dynamics of **unobserved** variables
- **Bayesian Experimental Design**
  - E.g., given the current model, when should patient $i$ be called for the next measurement so that population predictive uncertainty is maximally reduced?

# Conclusion

- GPs $\mathscr{GP}(m, k)$ are powerful tools for **nonparametric regression**
  - The kernel $k$ captures abstract function attributes (smoothness, stationarity)...
  - ...While also handling complex correlation structures among subjects
  - The mean function $m$ encompasses **mechanistic knowledge**

# Conclusion

- GPs $\mathcal{GP}(m, k)$ are powerful tools for **nonparametric regression**
  - The kernel $k$ captures abstract function attributes (smoothness, stationarity)...
  - ...While also handling complex correlation structures among subjects
  - The mean function $m$ encompasses **mechanistic knowledge**
- GPs act as a **bridge** between statistical and mechanistic modeling frameworks
  - Their strength lies in the low-data regime, typical in health-related problems

# Conclusion

- GPs $\mathscr{GP}(m, k)$ are powerful tools for **nonparametric regression**
  - The kernel $k$ captures abstract function attributes (smoothness, stationarity)...
  - ...While also handling complex correlation structures among subjects
  - The mean function $m$ encompasses **mechanistic knowledge**
- GPs act as a **bridge** between statistical and mechanistic modeling frameworks
  - Their strength lies in the low-data regime, typical in health-related problems
- The current challenge is to place GP priors over **vector fields**

# Conclusion

- GPs $\mathscr{GP}(m,k)$ are powerful tools for **nonparametric regression**
  - The kernel $k$ captures abstract function attributes (smoothness, stationarity)...
  - ...While also handling complex correlation structures among subjects
  - The mean function $m$ encompasses **mechanistic knowledge**
- GPs act as a **bridge** between statistical and mechanistic modeling frameworks
  - Their strength lies in the low-data regime, typical in health-related problems
- The current challenge is to place GP priors over **vector fields**

**Thank you for your attention ¯_(ツ)_/¯**