IMPERIAL

Multi-task Bayesian Optimisation for Competitor DNA Molecule Design

Ruby Sedgwick[1], John P. Goertz[1], Molly M. Stevens[1,2], Ruth Misener[1], Mark van der Wilk[1,2]

[1]Imperial College London

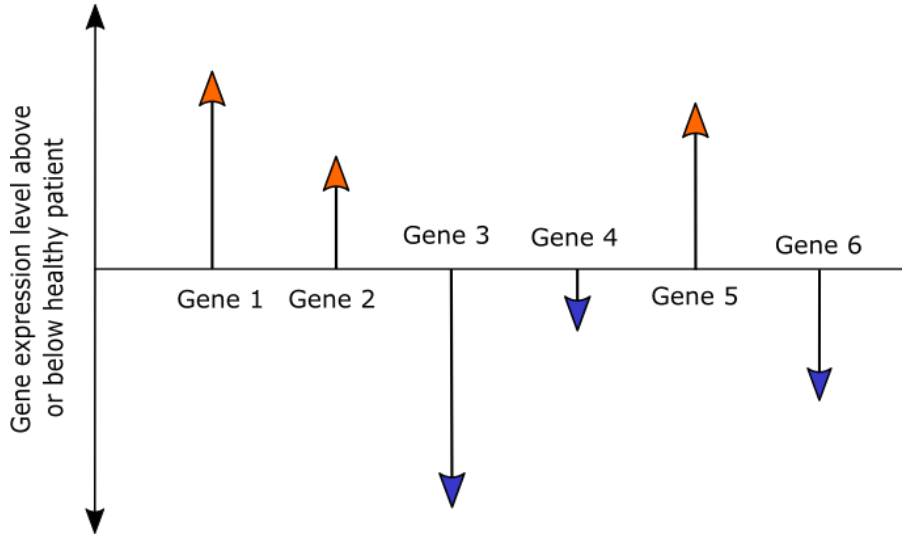
[2] University of Oxford

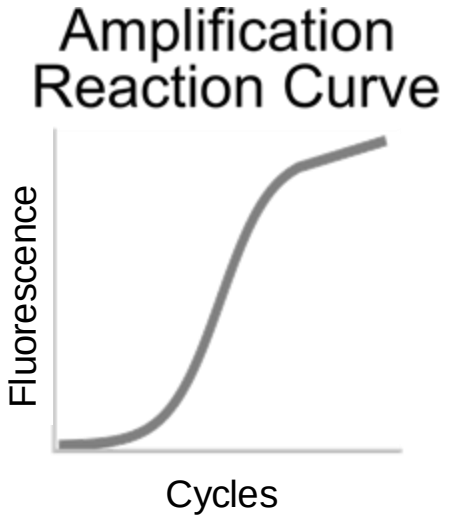
Presentation Overview

- Motivation
- Mathematical Formulation
- Design of Experiments Workflow
- Transfer Learning Surrogate Models
- Bayesian Optimisation
- Experimental Results
- Summary and Extensions

Motivation

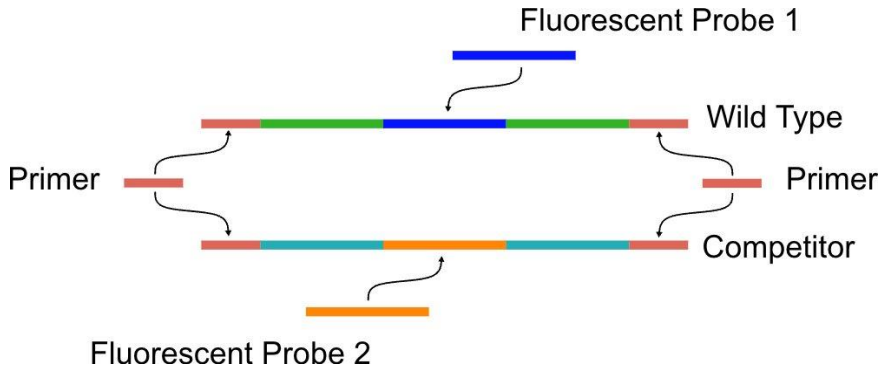
- Diagnostic device for detect levels of gene expressions a.k.a. RNA molecules in the blood
- PCR based for quicker, cheaper diagnosis of diseases
- PCR is a method for amplifying DNA
  - This is done by repeatedly dividing and rebuilding the DNA molecules

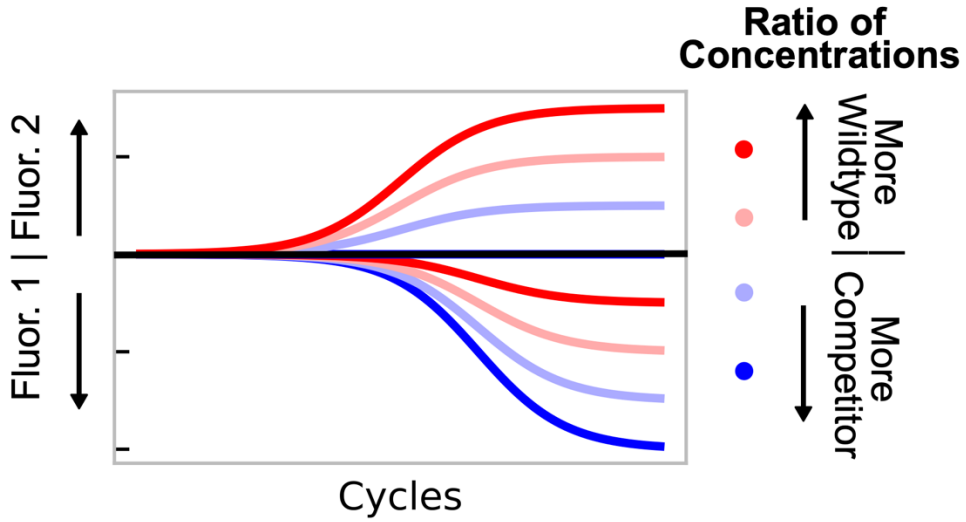




How Does the Device Work?

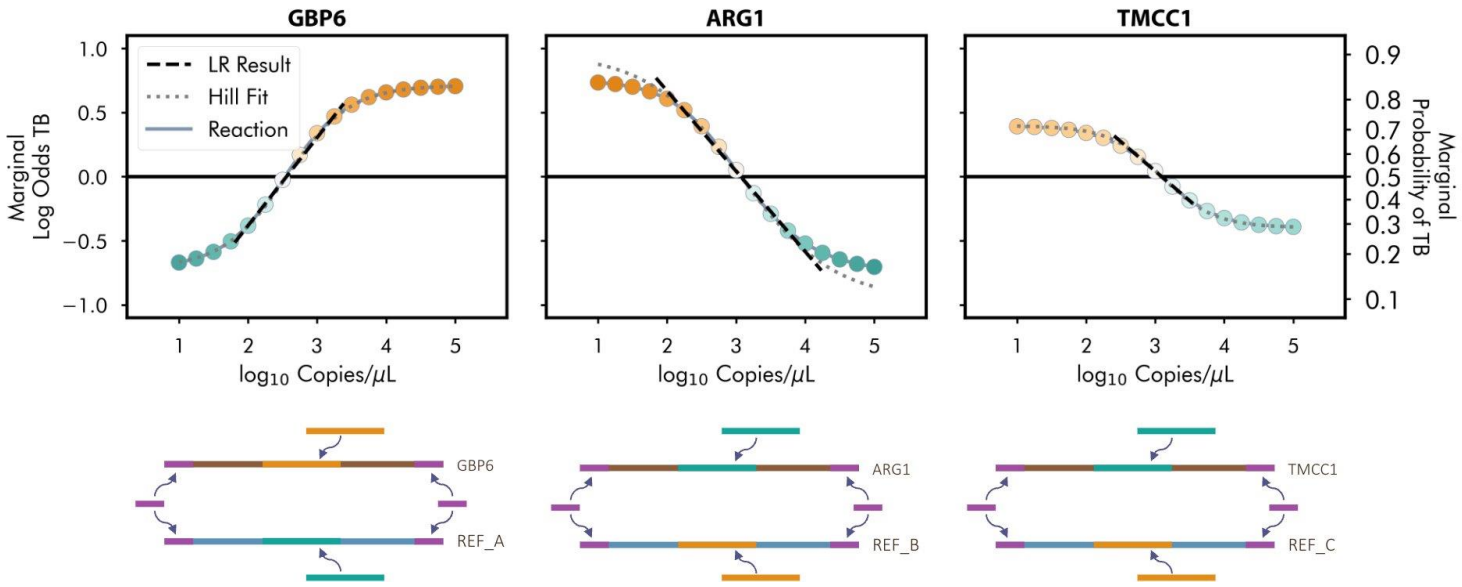
- In our device, synthetic competitor DNA molecules compete with the wild type for resources
- By comparing the difference in fluorescence at the end of the reaction, we can get an end point readout





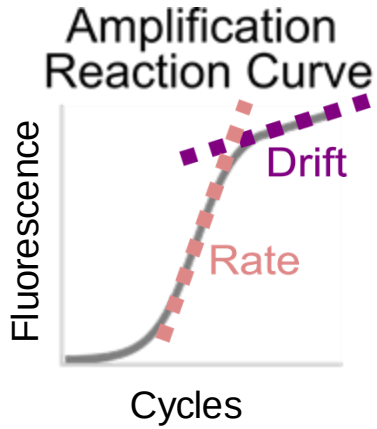
Multiplexed Sensing

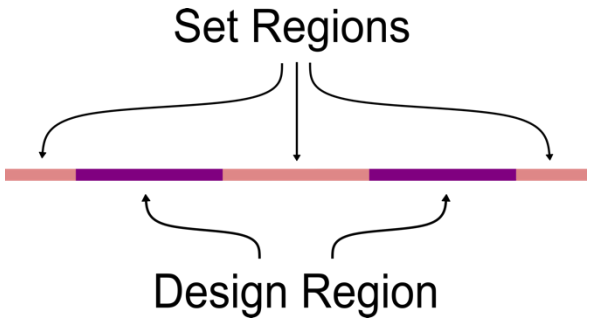
- Each device detects multiple gene expressions
  - Each gene expression requires a unique competitor

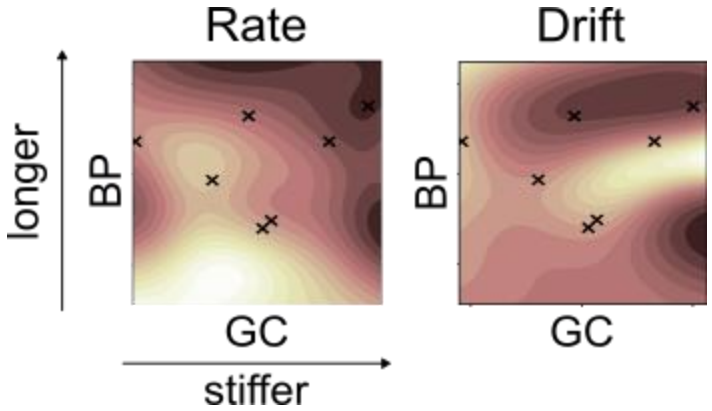


Single Competitor Design

- Objective:
  - for the rate to be as close to a target rate T_{rate} as possible
  - for the drift to be below a threshold, T_{drift}
- Designing DNA is a large combinatorial problem
  - We simplify this problem into a 2D approximately continuous design space







BP= number of base pairs, GC = % guanine-cytosine

Optimisation Objective

- Converting this into a mathematical objective:

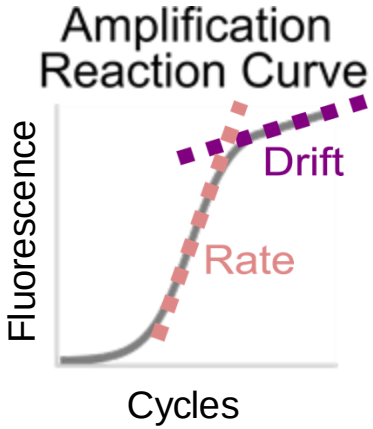
$$\underset{\mathrm{GC,BP}}{\operatorname{argmin}} \sqrt{(\mathrm{f_{rate}} - \mathrm{T_{rate}})^2} + \max(0, \mathrm{f_{drift}} - \mathrm{T_{drift}})$$

- For the multi-task case this becomes:

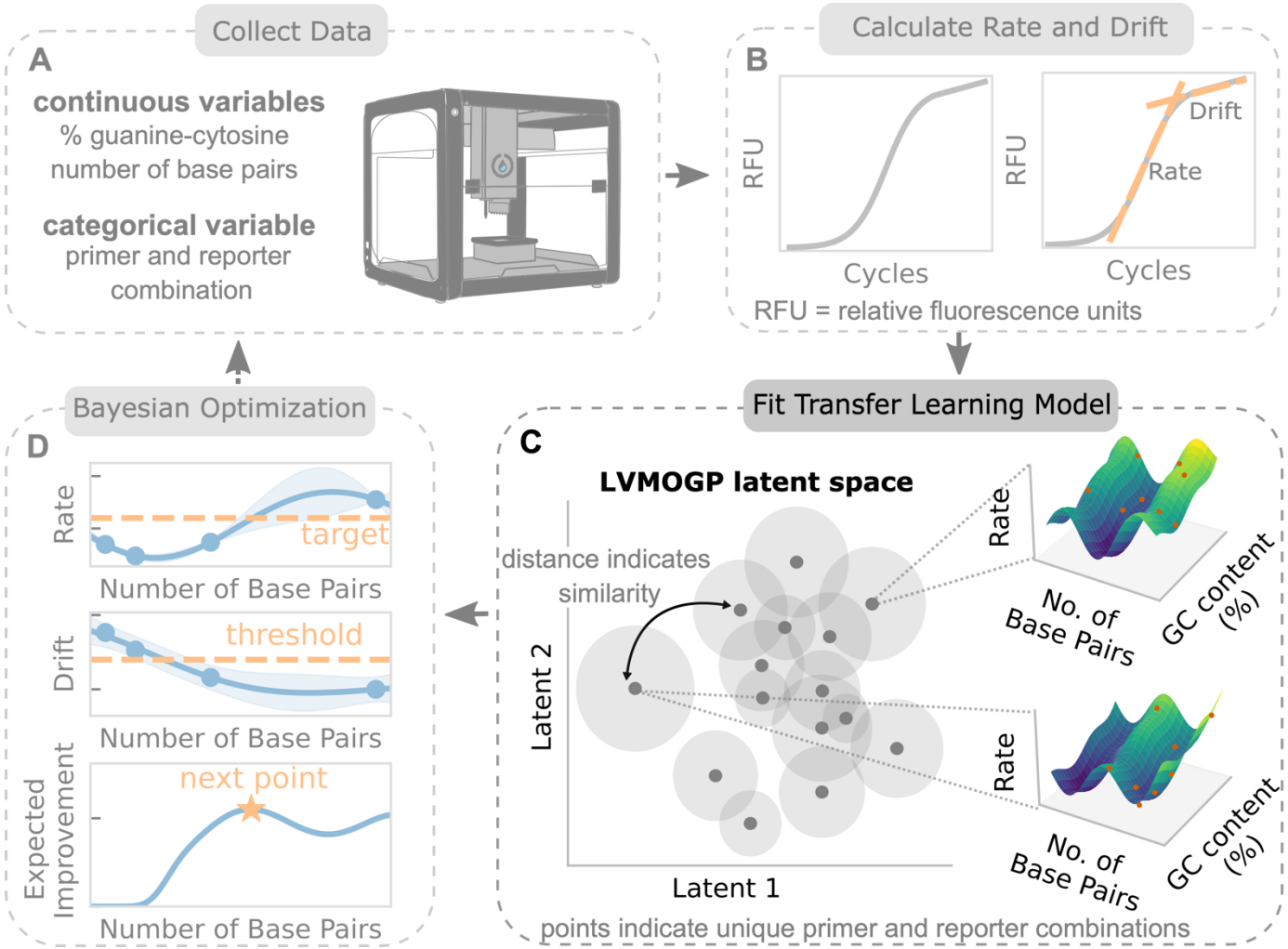
$$for\ i\ in\ competitors:$$

$$\underset{\mathrm{GC,BP}}{\operatorname{argmin}} \sqrt{(\mathrm{f_{rate,i}} - \mathrm{T_{rate,i}})^2} + \max(0, \mathrm{f_{drift,i}} - \mathrm{T_{drift}})$$

BP = number of base pairs, GC = % guanine-cytosine $\mathrm{f_{rate}}$ = rate, $\mathrm{f_{drift}}$= drift, $\mathrm{T_{rate}}$ = target rate, $\mathrm{T_{drift}}$=drift threshold

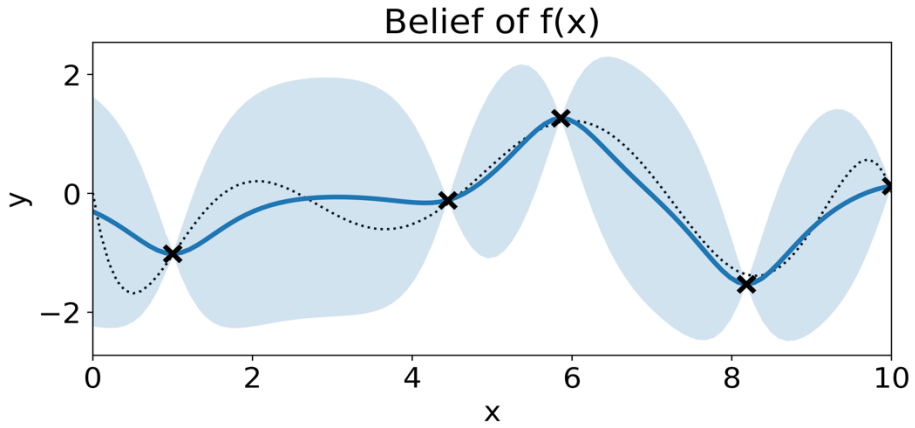


Design of Experiments Workflow



Surrogate Model

- We use Gaussian processes as they give good uncertainty measures and work well in low data regimes
- $\mathbf{y}(\mathbf{x}) \in \mathbb{R}$, is assumed to be a function of input $\mathbf{x} \in \mathbb{R}^{\mathrm{D}}$ plus some noise defined by the noise variance, $\boldsymbol{\sigma}_{\mathbf{n}}^{\mathbf{2}}$:



$$y(\boldsymbol{x}) = f(\boldsymbol{x}) + \epsilon, \qquad \epsilon \sim \mathcal{N}(0, \sigma_n^2)$$

- A Gaussian process is fully defined by its mean function $m : \mathbb{R}^{\mathrm{D}} \mapsto \mathbb{R}$ and covariance function $\mathrm{k} : \mathbb{R}^{\mathrm{D}} \times \mathbb{R}^{\mathrm{D}} \mapsto \mathbb{R}$.

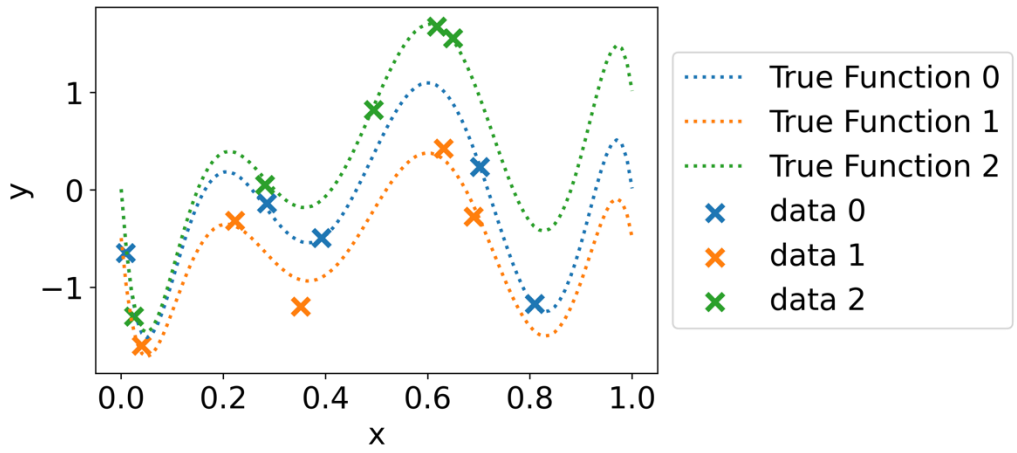
$$f(\boldsymbol{x}) \sim \mathcal{GP}(m(\boldsymbol{x}), \mathrm{k}(\boldsymbol{x}, \boldsymbol{x}'))$$

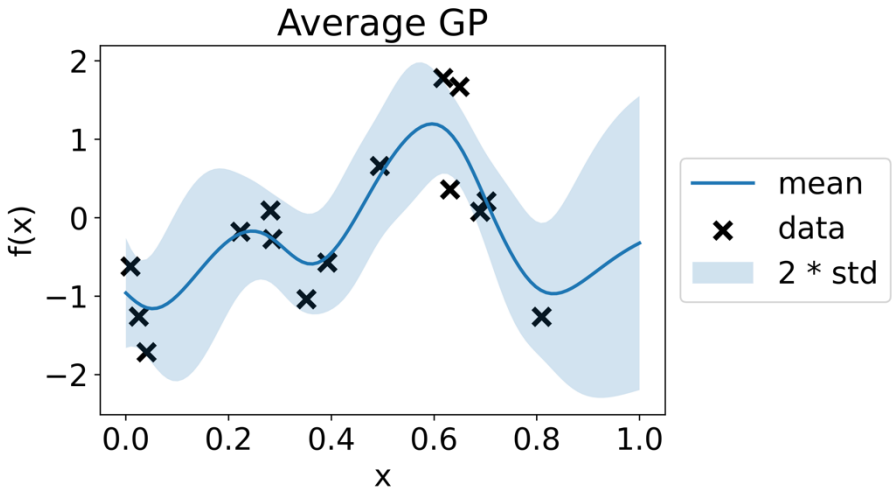
Surrogate Model

- “Average” Gaussian Process
- Multioutput Gaussian Process
- Linear Model of Coregionalisation
- Latent Variable Multioutput Gaussian Process

Average Gaussian Process

- This is the simplest model, where all data is taken to be from the same output, regardless of it's true output function
- Can be thought of as "total transfer"

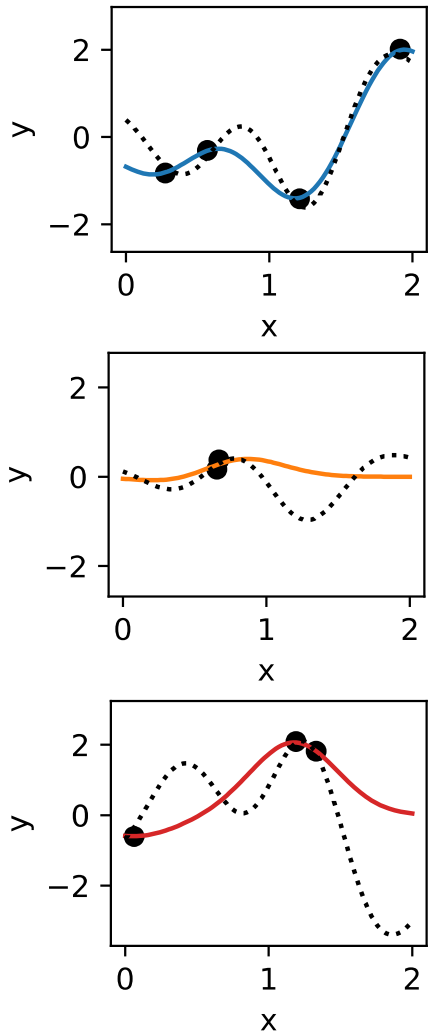




Multi-output Gaussian processes

- The multi-output Gaussian process (MOGP) extends the standard Gaussian process to multiple outputs, so $\mathbf{y}(\mathbf{x}) \in \mathbb{R}^{\mathbf{P}}$.
- It assumes all outputs have the same kernel function and hyperparameters but function values on different outputs are uncorrelated, giving the kernel structure:

$$\begin{bmatrix} f_1 \\ f_2 \end{bmatrix} \sim \mathcal{N}\begin{pmatrix} K(X_1, X_1) & \mathbf{0} \\ \mathbf{0} & K(X_2, X_2) \end{pmatrix}$$



Linear Model of Coregionalisation

- The linear model of coregionalisation (LMC) extends the MOGP to model linear correlations between output surfaces by assuming they are linear combinations of Gaussian process latent functions:

$$f_p(\boldsymbol{x}) = \boldsymbol{W}_p\, \boldsymbol{g}(\boldsymbol{x}) + \kappa_p\, \upsilon_p(\boldsymbol{x})$$

$f_p(\boldsymbol{x})$: p^{th} output function

$\boldsymbol{W}_p$: vector of weights

$\boldsymbol{g}(\boldsymbol{x})$: $\mathrm{g}(\mathrm{x}) = \{\mathrm{g_q}(\mathrm{x})\}_{\mathrm{q=1}}^{\mathrm{Q}}$ shared latent functions

κ_p: a learnt constant

$\upsilon_p(\boldsymbol{x})$: latent function that allows some independence of behaviour

Linear Model of Coregionalisation

- This leads to a Kronecker structured kernel with a joint distribution between two functions given by:

$$\begin{bmatrix} f_1 \\ f_2 \end{bmatrix} \sim \mathcal{N}\left(\begin{matrix} \sum_{q=1}^{Q} b_{11}\, k_q(X_1, X_1) & \sum_{q=1}^{Q} b_{12} k_q(X_1, X_2) \\ \sum_{q=1}^{Q} b_{21} k_q(X_2, X_1) & \sum_{q=1}^{Q} b_{22} k_q(X_2, X_2) \end{matrix} \right)$$

where $b_{pp'}$ is an element of $\boldsymbol{B} = \boldsymbol{W}\boldsymbol{W}^T + diag(\kappa)$ and Q is the number of different kernels the latent functions have.

- We use a special case of LMC called the Intrinsic Model of Coregionalization where $Q = 1$.

Latent Variable Multioutput Gaussian Process[1]

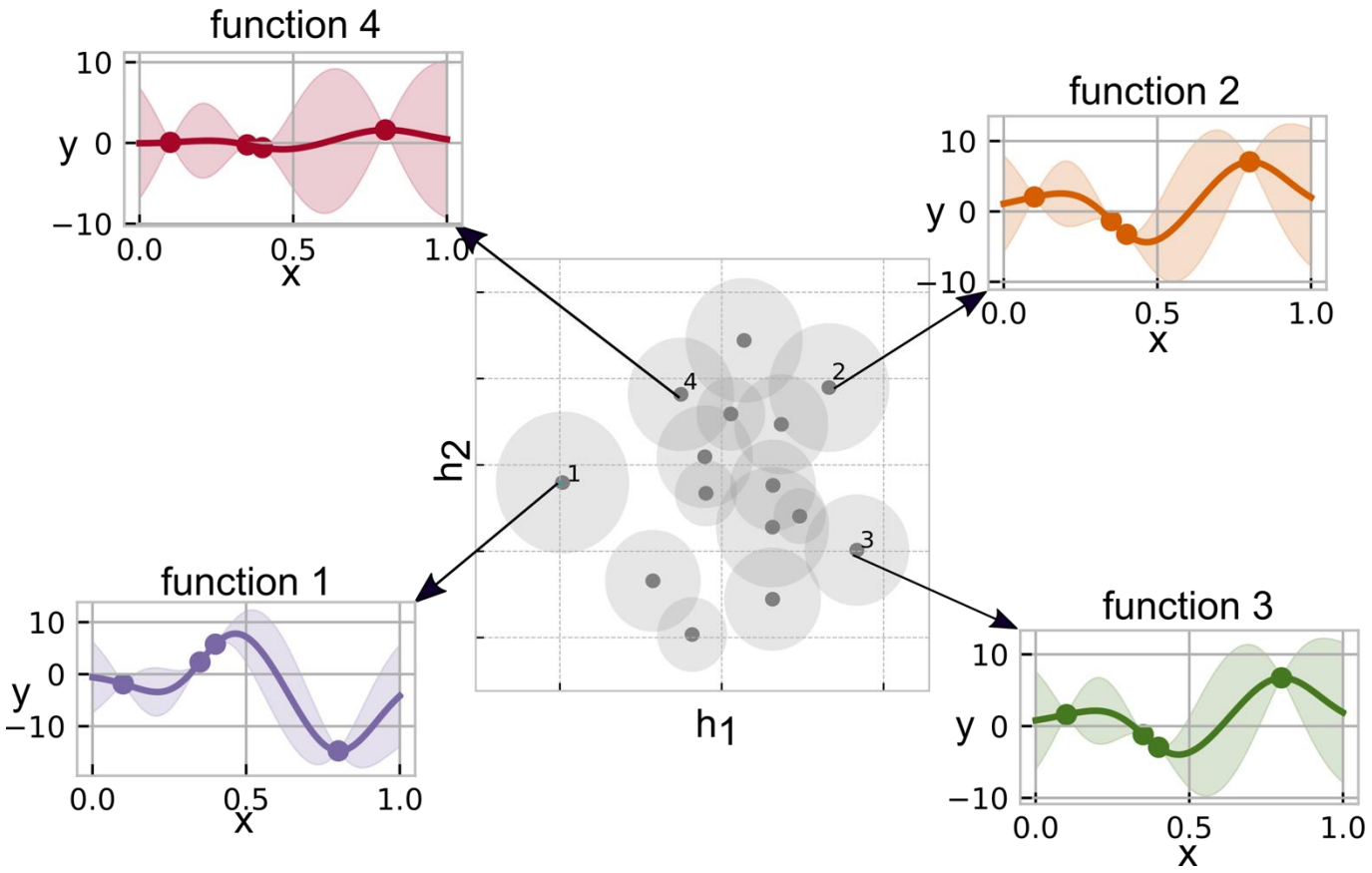
- The latent variable multi-output Gaussian process (LVMOGP) augments the input domain with extra latent dimensions:

$$y_p(\boldsymbol{x}) = f\left(\boldsymbol{x}, \boldsymbol{h}_p\right) + \epsilon, \qquad \underbrace{\boldsymbol{h}_p \sim \mathcal{N}\left(\mu_{h_p}, \Sigma_{h_p}\right)}_{\text{Latent variable}}, \qquad \epsilon \sim \mathcal{N}(0, \sigma_n^2).$$

- Dimensionality of the latent variables is $H = \left[\boldsymbol{h}_1, \ldots, \boldsymbol{h}_p\right]^T \in \mathbb{R}^{P \times Q_H}$ where Q_H is the dimensions of the latent space
- The LVMOGP is trained using variational inference

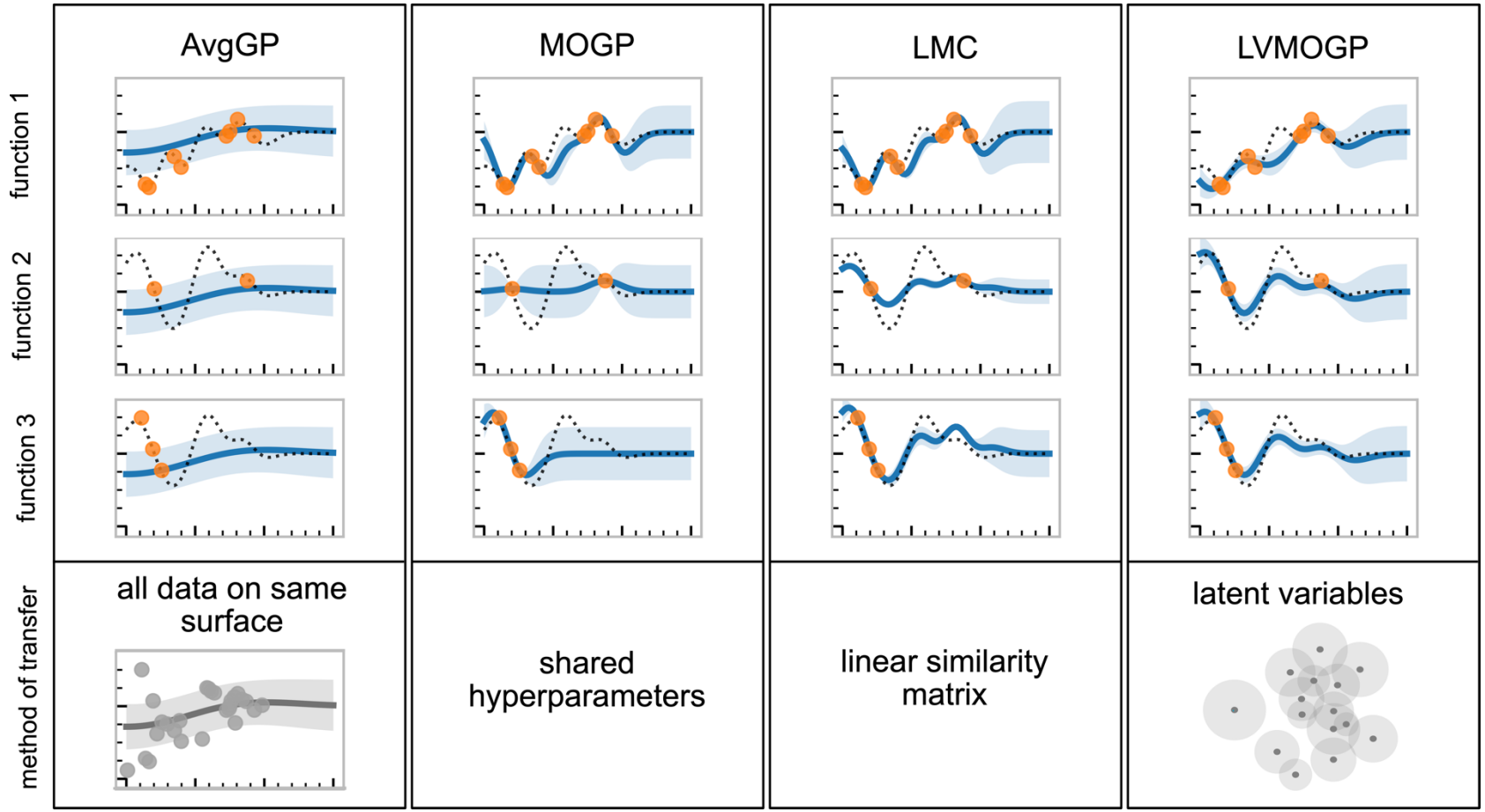
[1] Dai, Z., Álvarez, M. and Lawrence, N. (2017) Efficient Modeling of Latent Information in Supervised Learning using Gaussian Processes. In Advances in Neural Information Processing Systems.

Latent Variable Multioutput Gaussian Process[1]



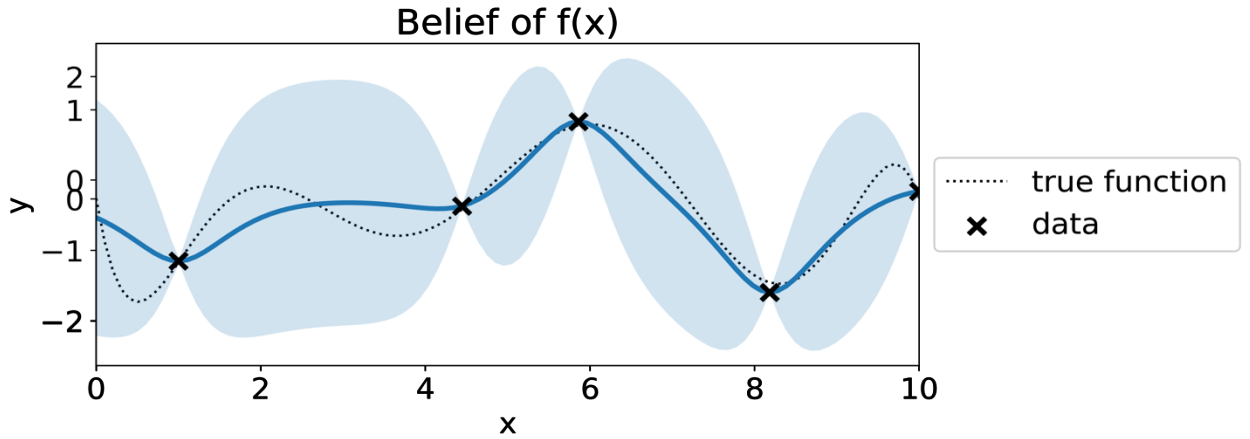
[1] Dai, Z., Álvarez, M. and Lawrence, N. (2017) Efficient Modeling of Latent Information in Supervised Learning using Gaussian Processes. In Advances in Neural Information Processing Systems.

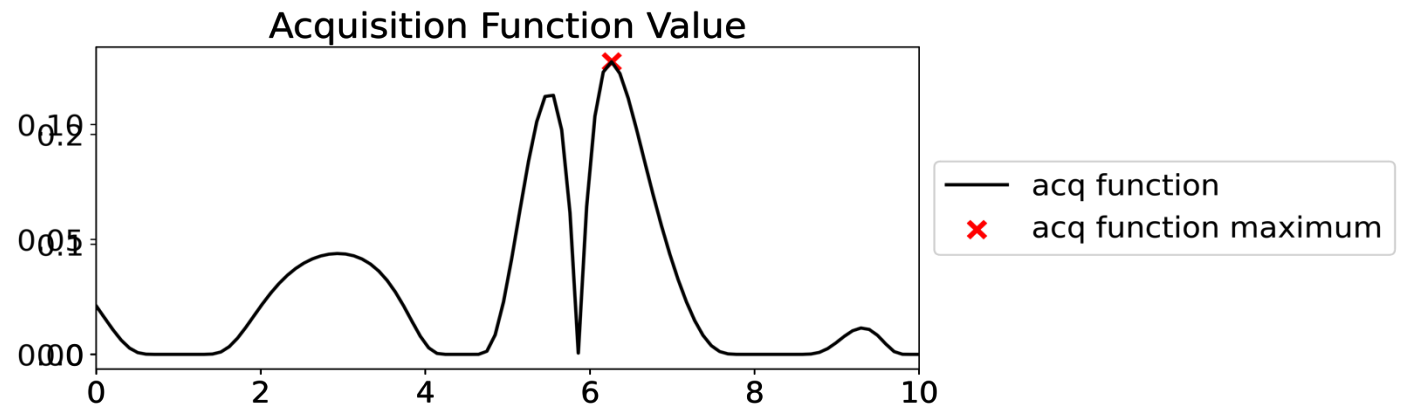
Multi-task Gaussian Process Surrogates



Bayesian Optimisation

- Now we have our surrogate functions, we want to optimise our molecules.





Bayesian Optimisation

- We wish to minimise the difference between the rate and the target rate.
- To do this we use the result of Uhrenholt et al.[2] where a new stochastic variable is defined as:

$$\delta|\mathrm{x} = \|\mathrm{y}_{\mathrm{rate}}(\mathrm{x}) - \mathrm{T}_{\mathrm{rate}}\|_2^2.$$

- The expected improvement for this variable can then be written as:

$$\alpha_{EI} = \boldsymbol{\delta}_{min} G_\lambda(\boldsymbol{\delta}_{min}/\gamma^2) - \gamma^2 \mathbb{E}\left[t \middle| t < \frac{\boldsymbol{\delta}_{min}}{\gamma^2}\right] G_\lambda(\boldsymbol{\delta}_{min}/\gamma^2),$$

Min value of $\boldsymbol{\delta}$ observed so far

Approximate cumulative χ^2 distribution with non-centrality parameter λ

$t = \delta\gamma^{-2}$

Root mean of variances of each output evaluated at training points

[2] Uhrenholt, Anders Kirk and Bjøern Sand Jensen (Apr. 2019). "Efficient Bayesian Optimization for Target Vector Estimation". In: Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics.

Penalty Term

- We also want to penalise any point with a drift value over a given threshold
- We use the probability of feasibility:

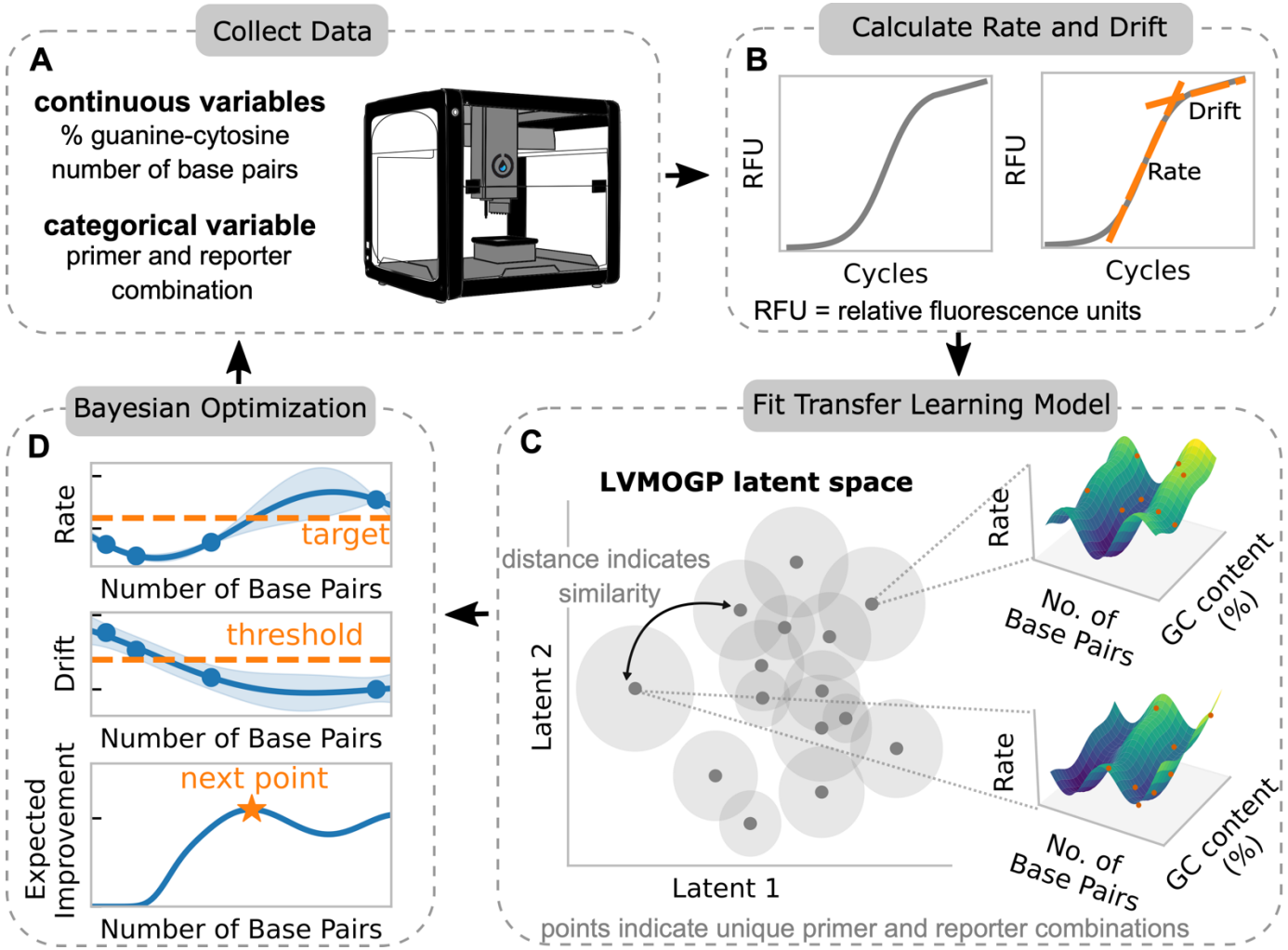
$$PF(\boldsymbol{x}) = p\left(f_{drift}(\boldsymbol{x}) \leq T_{drift}\right).$$

Drift threshold

- To get our final acquisition function we then multiply the expected improvement by the probability of feasibility:

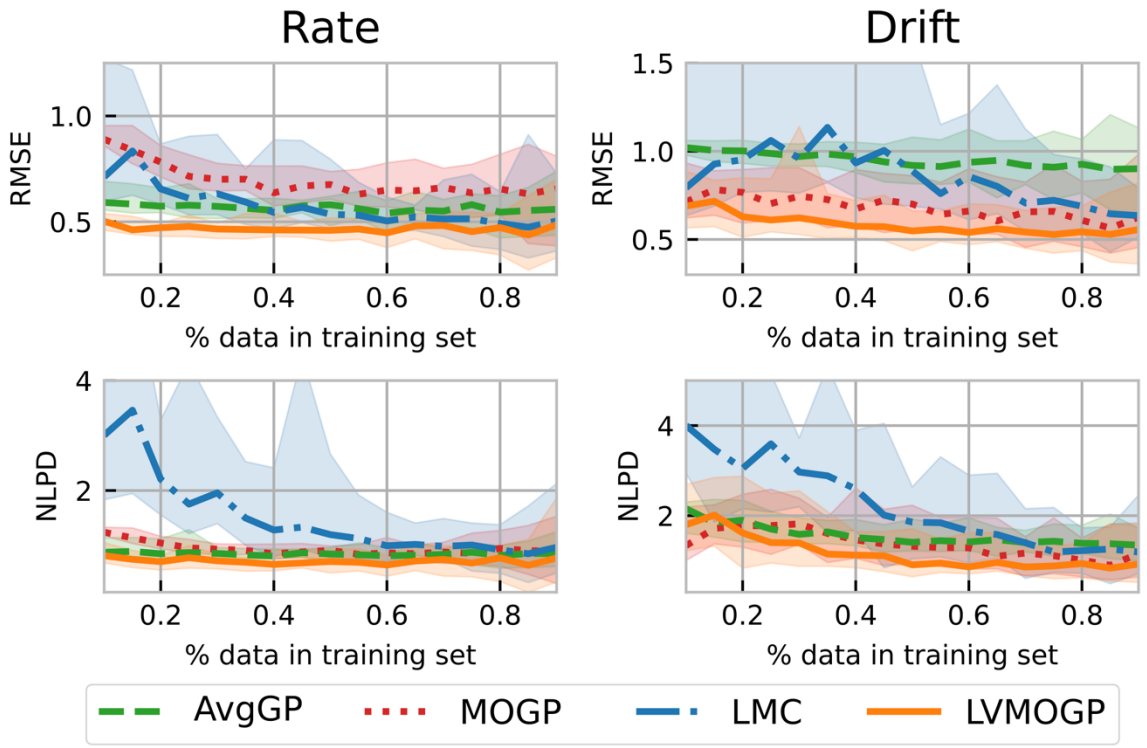
$$\alpha = PF(\boldsymbol{x})\alpha_{EI}(\boldsymbol{x}).$$

Design of Experiments Workflow



Cross Validation

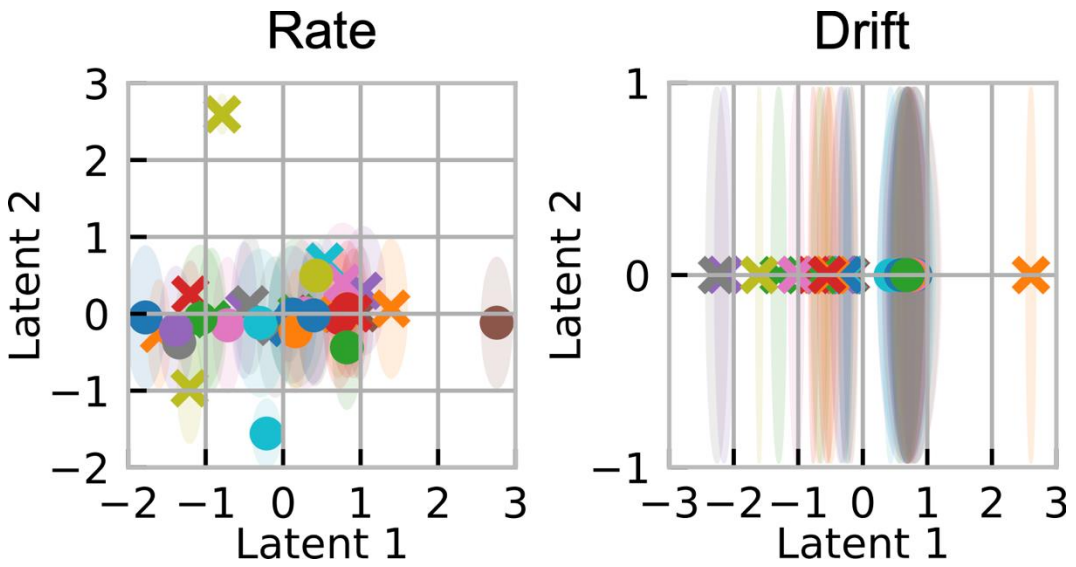
- We performed cross validation on our dataset to assess fit



RMSE= root mean squared error, NLPD = negative log predictive density

Latent Space

- The latent space of the LVMOGP uses the ARD property of the kernel

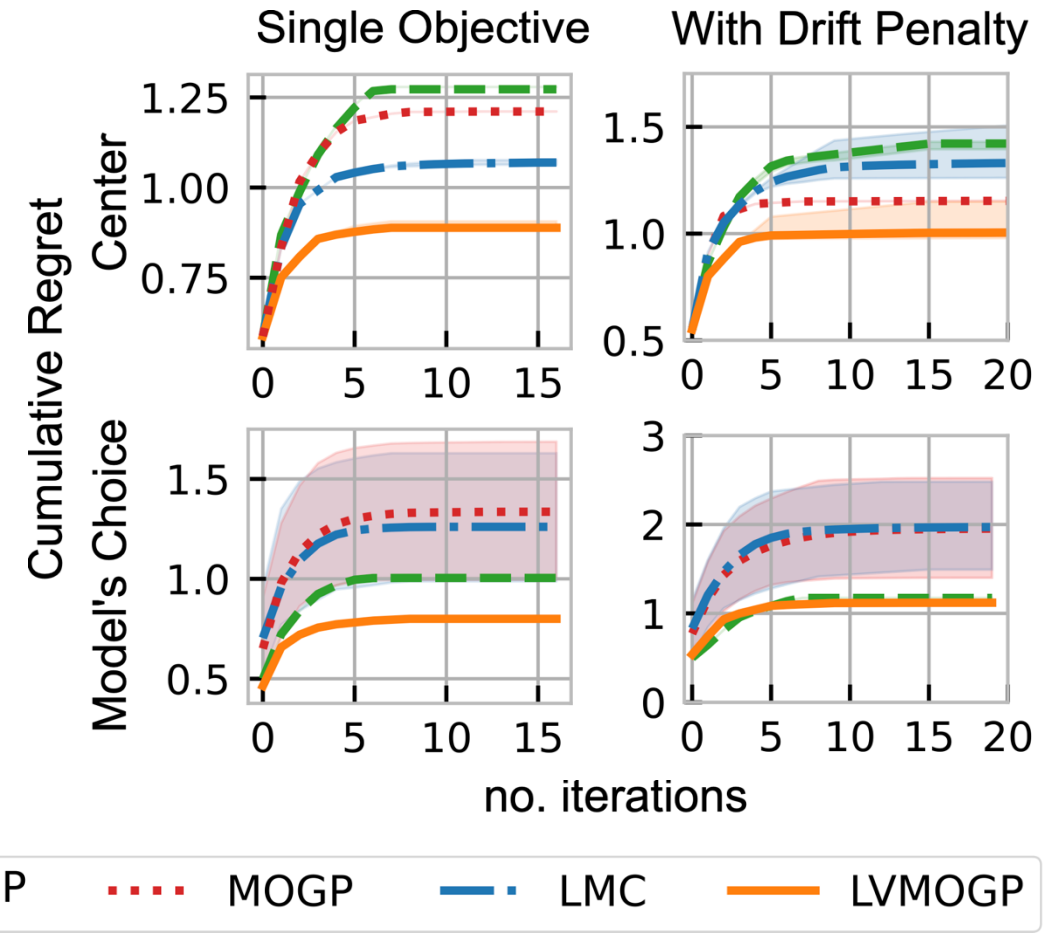


Bayesian Optimisation

- We performed retrospective Bayesian optimisation where each of the models is only allowed to choose the next point from the existing dataset
- Choice of starting point:
  - Centre
  - Model's choice
- Choice of learning problem:
  - Learning all surfaces at the same time
  - Learning one surface at a time, with all others in the training set

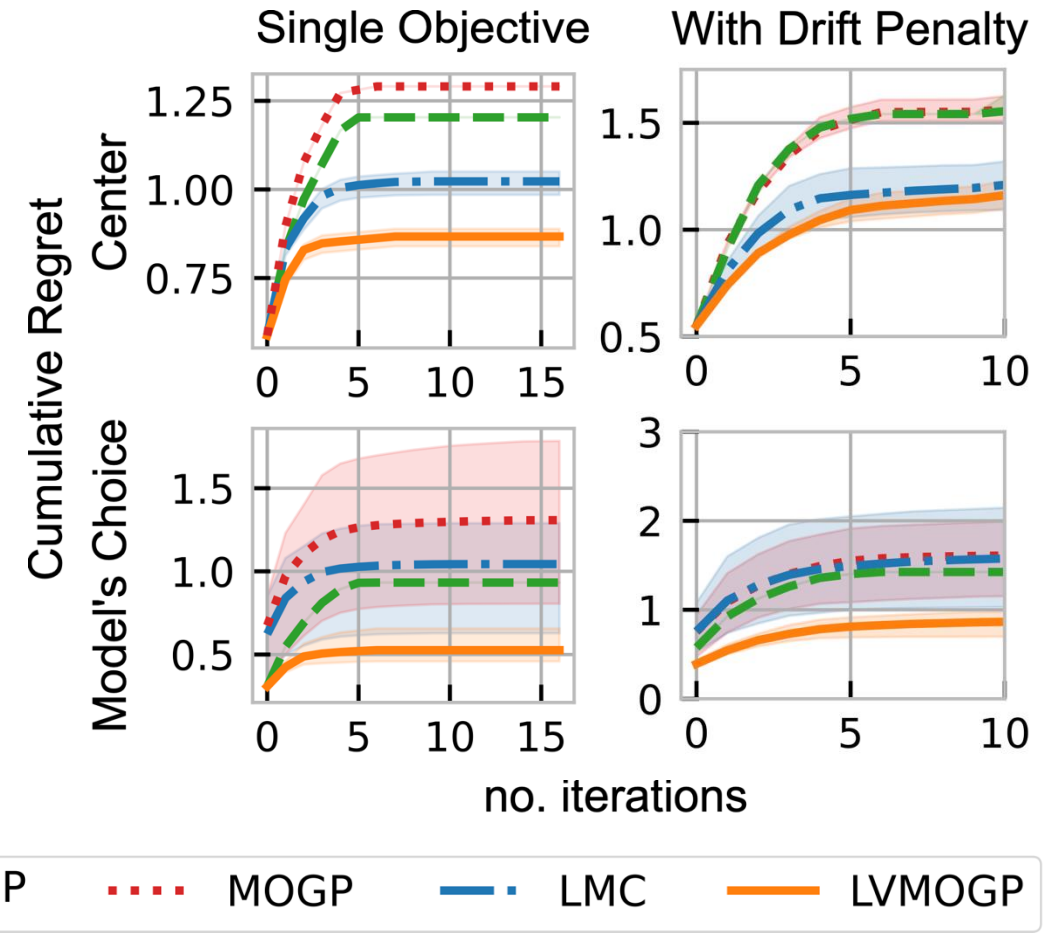
Bayes Opt: Learning Many Surfaces

- Two fully observed surfaces and then learning all other surfaces at the same time
- Cumulative regret
- LVMOGP has less cumulative regret
- The models that can choose their first point do better



Bayes Opt: Learning One Surface

- Learning only one surface, all others are fully observed
- LVMOGP has lower cumulative regret
- LMC performs better than in the learning many scenario



Summary

- We converted the problem of designing competitor DNA molecules into an optimisation problem
- We compared a number of multi-task learning surrogate functions and found that:
  - The LVMOGP had the best predictive accuracy
  - This translated to the least regret in Bayesian optimisation
- These results show this method can reduce the number of experiments needed both when developing many competitors at the same time and when optimising a new one

Possible Extensions

- Exploration of other acquisition functions
  - Especially ones that take the multi-task learning into account
- Better models for drift
- Further investigation of the ARD properties of the LVMOGP latent space
- The variational inference of the LVMOGP is a very non-convex problem and sensitive to initalisation
  - Better inference and simpler optimisation procedure would make this method more useable
- This approach can be extended to other problems

Thank you!

Ruby Sedgwick
r.sedgwick19@ic.ac.uk

Dr. Mark van der Wilk
mark.vdwilk@cs.ox.ac.uk

Prof. Ruth Misener
r.misener@ic.ac.uk

Dr. John Goertz
john@signatur.bio

Prof. Molly Stevens
molly.stevens@dpag.ox.ac.uk

paper

website