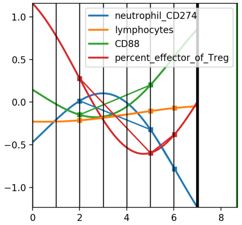
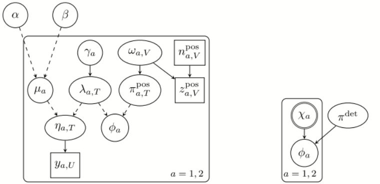
Lorenz Wernisch

- Principal scientist, MRC Biostatistics Unit
  - Single cell gene expression, gene regulation, biomarkers, health records (ICU, brain injury)
  - Bayesian modelling, Gaussian processes, machine learning, deep learning
- Lecturer/Reader/Chair in Computational Biology, Birkbeck College, London
- Researcher, European Bioinformatics Institute, Genome Campus, Hinxton
- PhD in theoretical computer science, Berlin
- MSc in mathematics, Vienna

Research topics



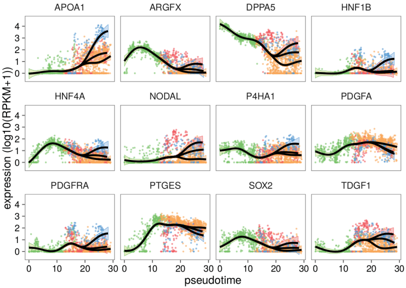
ICU infection prediction



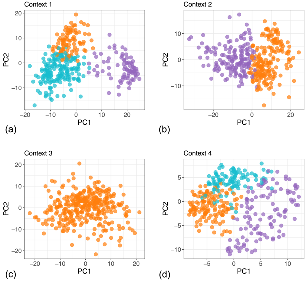
(a) ICU submodel

(b) Severity submodel

Model surgery

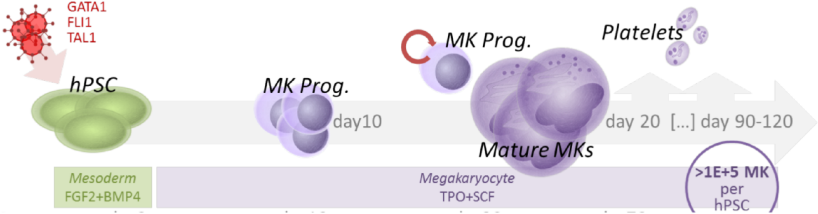


Pseudotime branching



Context clustering

Stem cells to Megakaryocytes



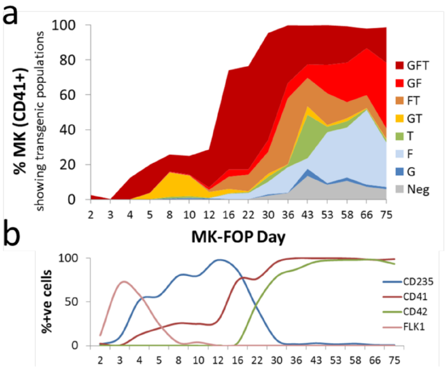
Human pluriopotent stem cells to Megakaryocytes via induction of genes GATA1, FLI1, TAL1

Increase yield and maturity of MK cells

When and how much induction is required?

Cedric Ghevaert, Cambridge Blood Centre

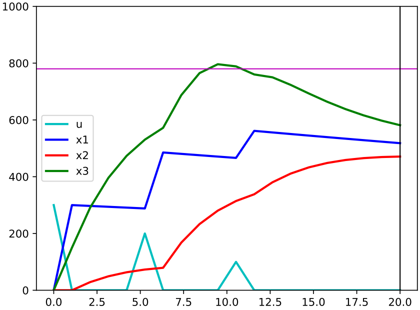
Flow cytometry measurements



Cell surface markers and gene transcription profile indicate maturity

Yield of MK cells from cell cultures

Toy abstraction of problem

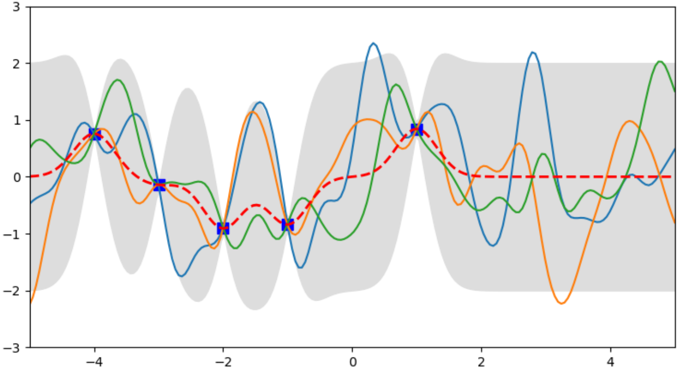


System of three variables $x_{1,t}, x_{2,t}, x_{3,t}$ measured daily over 20 days

We control input u_t to push $x_{3,20}$ (green) to a target value on day 20

$$\begin{pmatrix} x_{1,t} \\ x_{2,t} \\ x_{3,t} \end{pmatrix} = \begin{pmatrix} x_{1,t-1} \\ x_{2,t-1} \\ x_{3,t-1} \end{pmatrix} + \begin{pmatrix} f_1(x_{1,t-1}, x_{2,t-1}, x_{3,t-1}, u_{t-1}) \\ f_2(x_{1,t-1}, x_{2,t-1}, x_{3,t-1}, u_{t-1}) \\ f_3(x_{1,t-1}, x_{2,t-1}, x_{3,t-1}, u_{t-1}) \end{pmatrix}$$

GP uncertainty and samples



Regions of uncertainty, amount controlled by θ

Smoothness controlled by lengthscale ℓ

Estimate by ML or MAP

Dynamical system via Gaussian processes

Idea: use data to approximate the difference

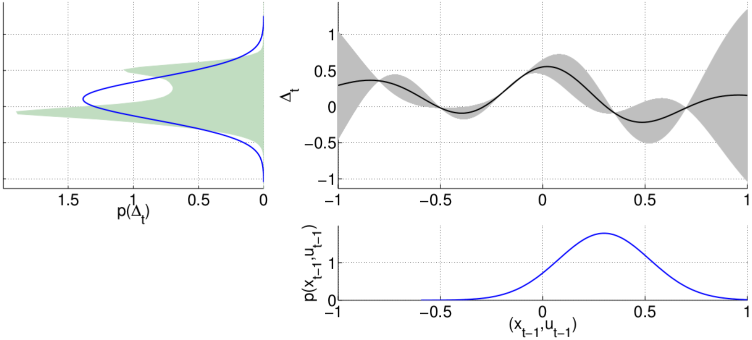
$$x_{i,t} - x_{i,t-1} = f_i(x_{1,t-1}, \ldots, x_{d,t-1}, u_{t-1})$$

via d GPs $f_i \sim GP(\mu(x,u), \Sigma(x,u))$, $i = 1, \ldots, d$

Iterative cycle for *optimising control* towards target:

- Approximate response x to u by GPs
- Optimize control input u towards target using GP approximation
- Acquire new data applying control to real system

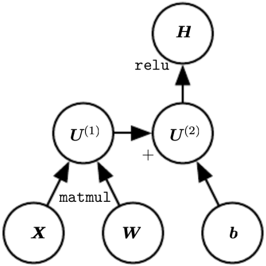
Transmitting uncertainty



(Gaussian) uncertainty in *inputs* results in (non-Gaussian) distribution in output

Pilco: approximate by Gaussian via moment matching

Alternative: dataflow graphs



TensorFlow works with *stateful dataflow graphs*

Data flow between nodes in a directed graph

Nodes represent eg: arithmetic operations, control clauses (if else), matrix manipulations, random number generators

Automatic differentiation

Optimize u using GP approximation

Minimize cost $c(u)$, eg for trajectory $x_3(u_0, \ldots, u_{19})$

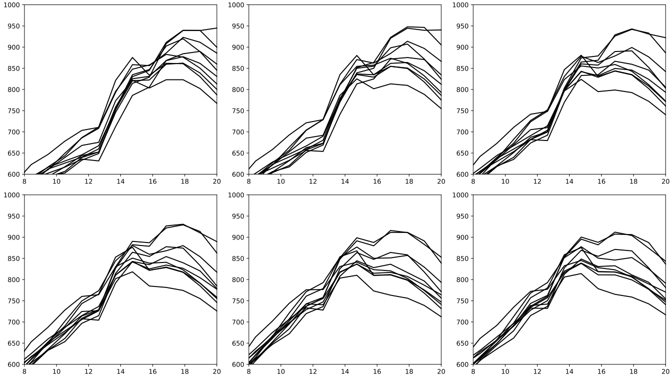
$$c(u) = (x_{3,20}(u_0, \ldots, u_{19}) - x_{\mathrm{target}})^2 + \lambda \sum_t |u_t|$$

How to define trajectory using GPs?

Bad idea: use means of GP for $f_i(x_{t-1}, u_{t-1})$

Better idea: sample several *random* trajectories using GP uncertainty and optimise eg mean

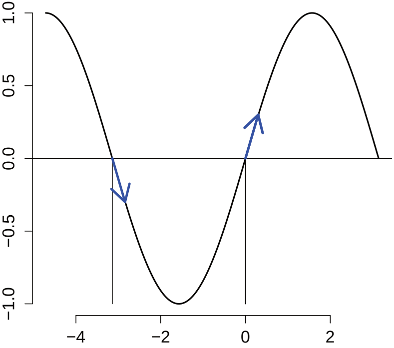
Random trajectories

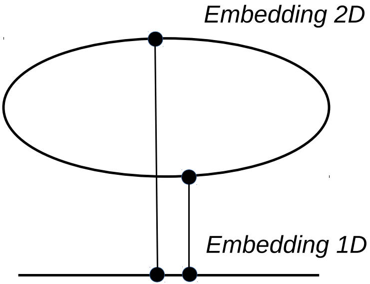


Expected loss $C(u) = 1/m \sum_k c(x^{(k)})$

Reparametrisation trick for Gaussian:
With $\epsilon \sim N(0, I)$: $z(u) = \mu(u) + C(u)\epsilon$

Key model parameter: latent dimension





$\dot{y} = f(y)$ or $x_t = x_{t-1} + f(t_{t-1})$ not representable in 1D:

Identical y mapped to different $f(y)$

Takens' theorem

Observe one variable $y(1), y(2), y(3), \ldots, y(L)$ from dynamical system

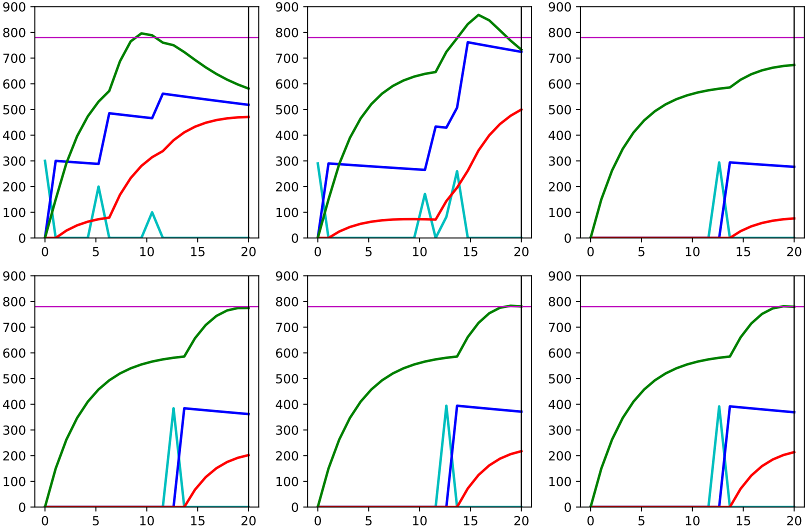
Find lag d with (eg by small autocorrelation)

Choose m and form delay embedding (eg $d = 2$, $m = 3$):

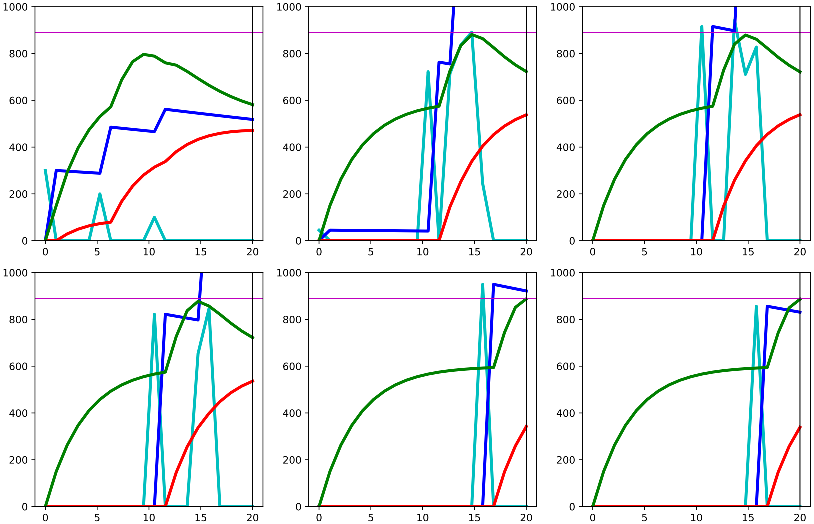
$$\begin{pmatrix} y(1) \\ y(3) \\ y(5) \end{pmatrix}, \begin{pmatrix} y(2) \\ y(4) \\ y(6) \end{pmatrix}, \begin{pmatrix} y(3) \\ y(5) \\ y(7) \end{pmatrix}, \ldots$$

Choose m with few "false" neighbors (monitor distance increase when adding $m + 1$st expansion)

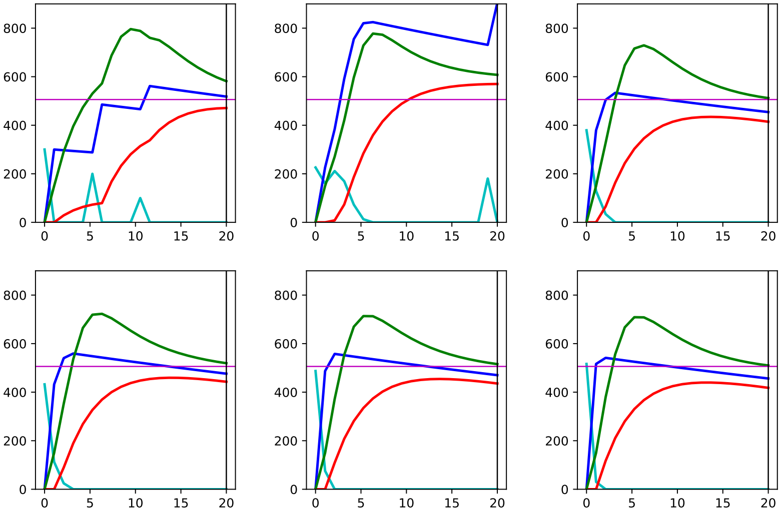
Aim: green to 780



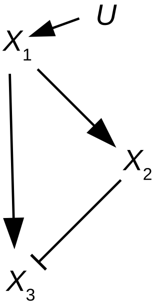
Aim: green to maximum

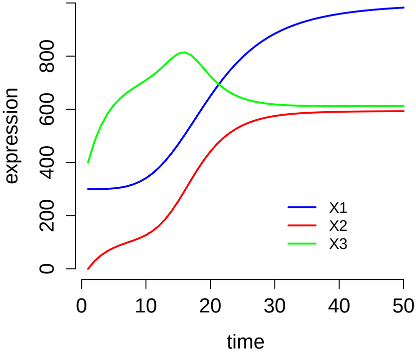


Aim: green to minimum



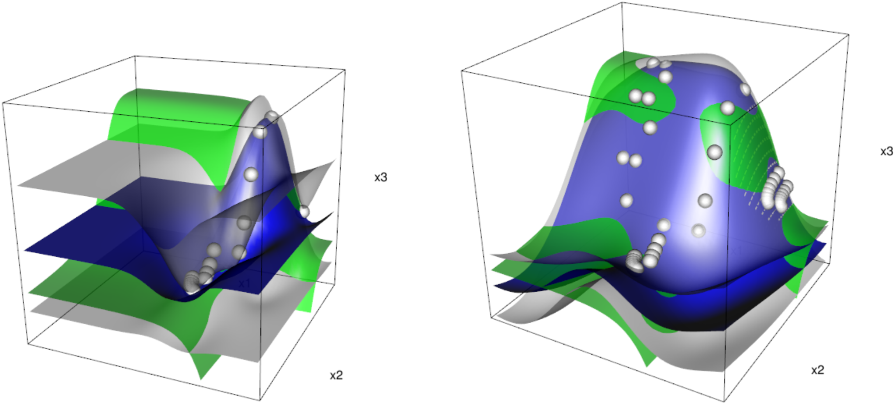
Feedforward loop





$$
\begin{aligned}
X_1(t) &= (1-\lambda_1)X_1(t-1) + U_{\mathrm{activate}}(t) \\
X_2(t) &= (1-\lambda_2)X_2(t-1) + h^{+}(X_1(t-1)) \\
X_3(t) &= (1-\lambda_3)X_3(t-1) + h^{+}(X_1(t-1)) \\
&\quad + h^{-}(X_2(t-1))
\end{aligned}
$$

GP reconstruction of $f_3(x_1, x_2)$



NANOG, SOX, OCT4 network

$$\frac{d[O]}{dt} = \frac{\eta_1 + a_1[A_+] + a_2[OS] + a_3[OS][N]}{1 + \eta_2 + b_1[A_+] + b_2[OS] + b_3[OS][N]}$$

$$- \gamma_1[O] - k_{1c}[O][S] + k_{2c}[OS]$$

$$\frac{d[S]}{dt} = \frac{\eta_3 + c_1[A_+] + c_2[OS] + c_3[OS][N]}{1 + \eta_4 + d_1[A_+] + d_2[OS] + d_3[OS][N]}$$

$$- \gamma_2[S] - k_{1c}[O][S] + k_{2c}[OS]$$

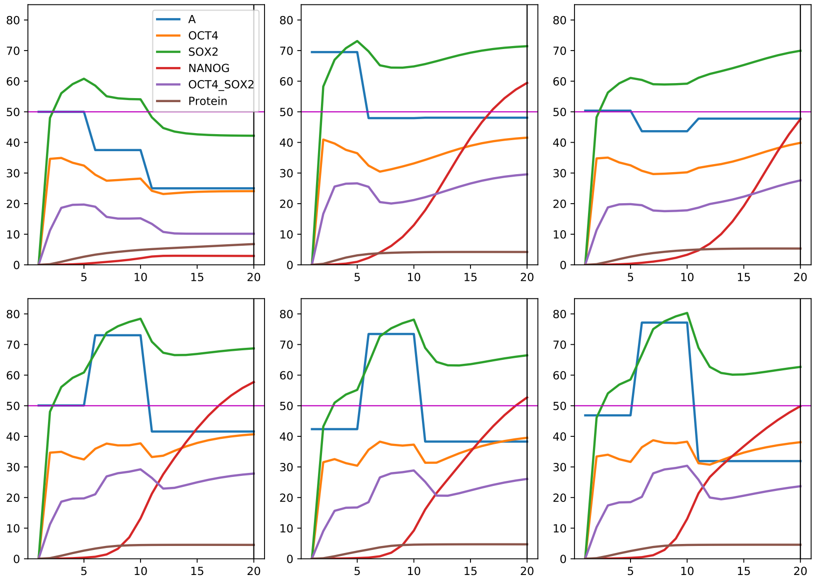
$$\frac{d[OS]}{dt} = k_{1c}[O][S] - k_{2c}[OS] - k_{3c}[OS]$$

$$\frac{d[N]}{dt} = \frac{\eta_5 + e_1[OS] + e_2[OS][N]}{1 + \eta_6 + f_1[OS] + f_2[OS][N] + f_3[B_-]} - \gamma_3[N].$$

Chickarmane et al., Transcriptional Dynamics of the Embryonic Stem Cell Switch, PLoS Comp. Biol.

SMBL Biomodel BIOMD0000000203

Aim: NANOG (red) to 50



Afterthoughts

Dynamic control problem widely applicable

Reconstruct network through Automatic Relevance Determination

Taking uncertainty into account is essential

Regularising effect of using expectation of cost

Converges surprisingly quickly

TensorFlow enables straightforward, flexible implementation