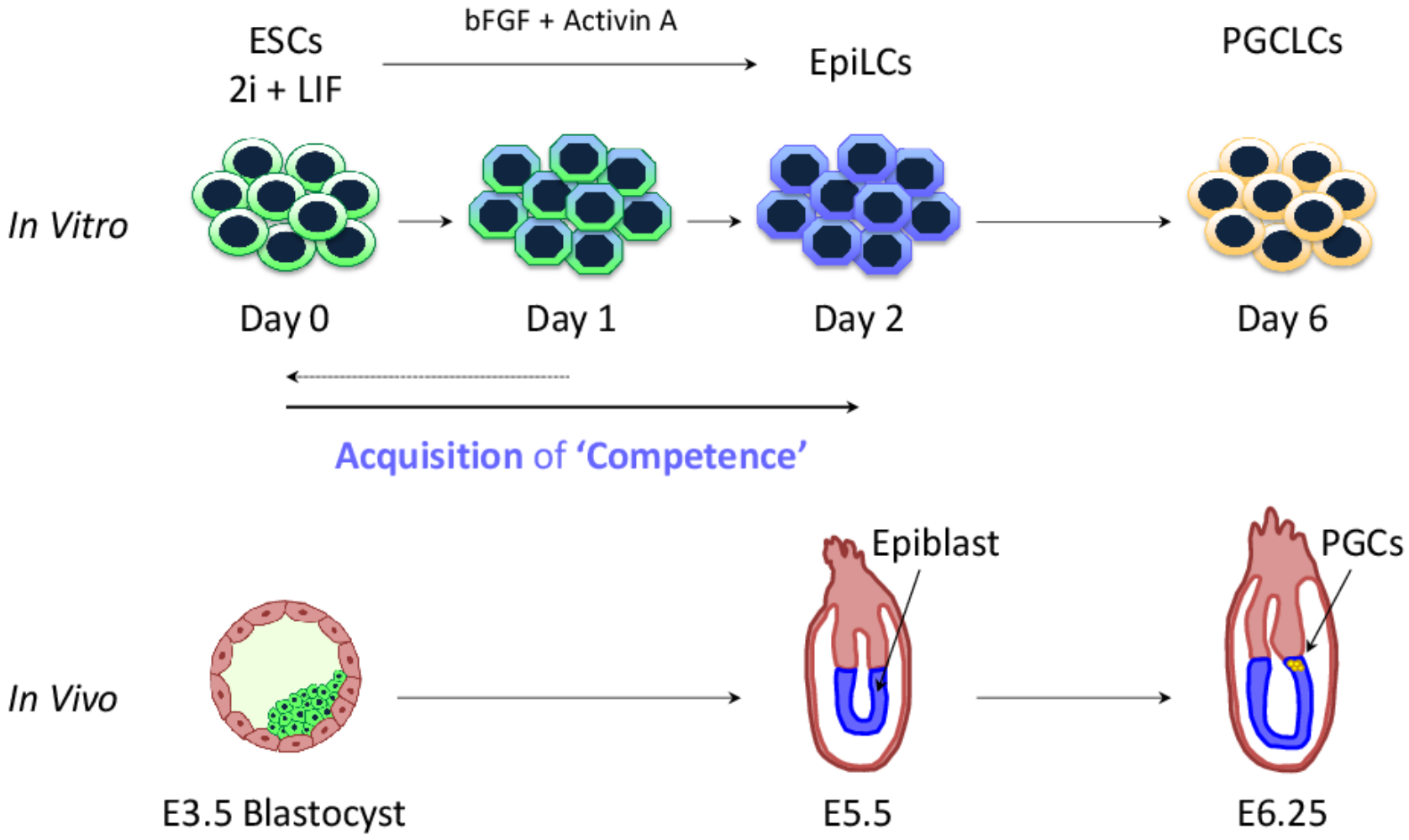
Gene regulatory networks from single cell data

Lorenz Wernisch, John Reid, Magdalena Strauss

MRC-Biostatistics Unit
Cambridge, UK

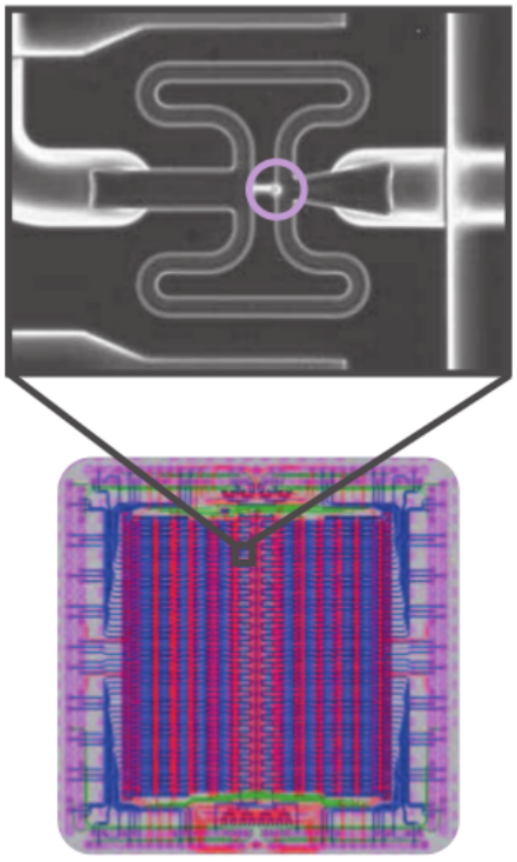
13 Jan 2017

Cell development

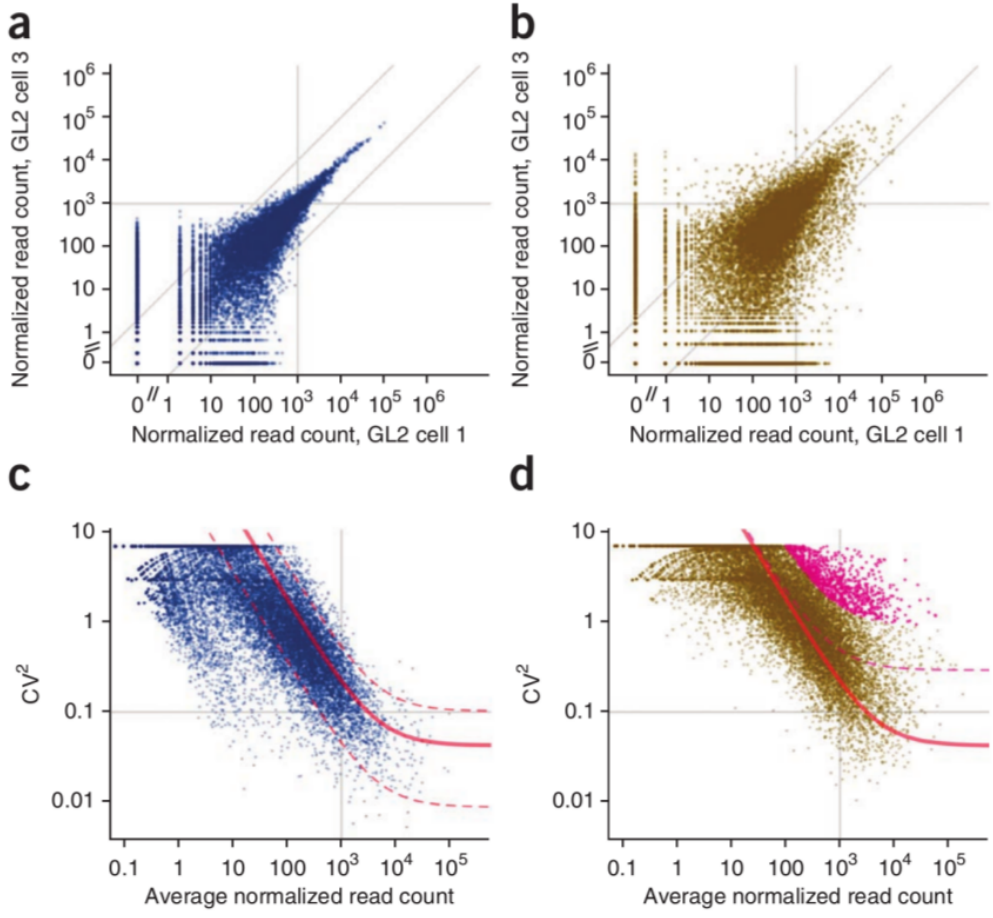


PGCs: Primordial germ cells, **Julia Tischler** (Gurdon Institute)

Single cell RNA-Seq

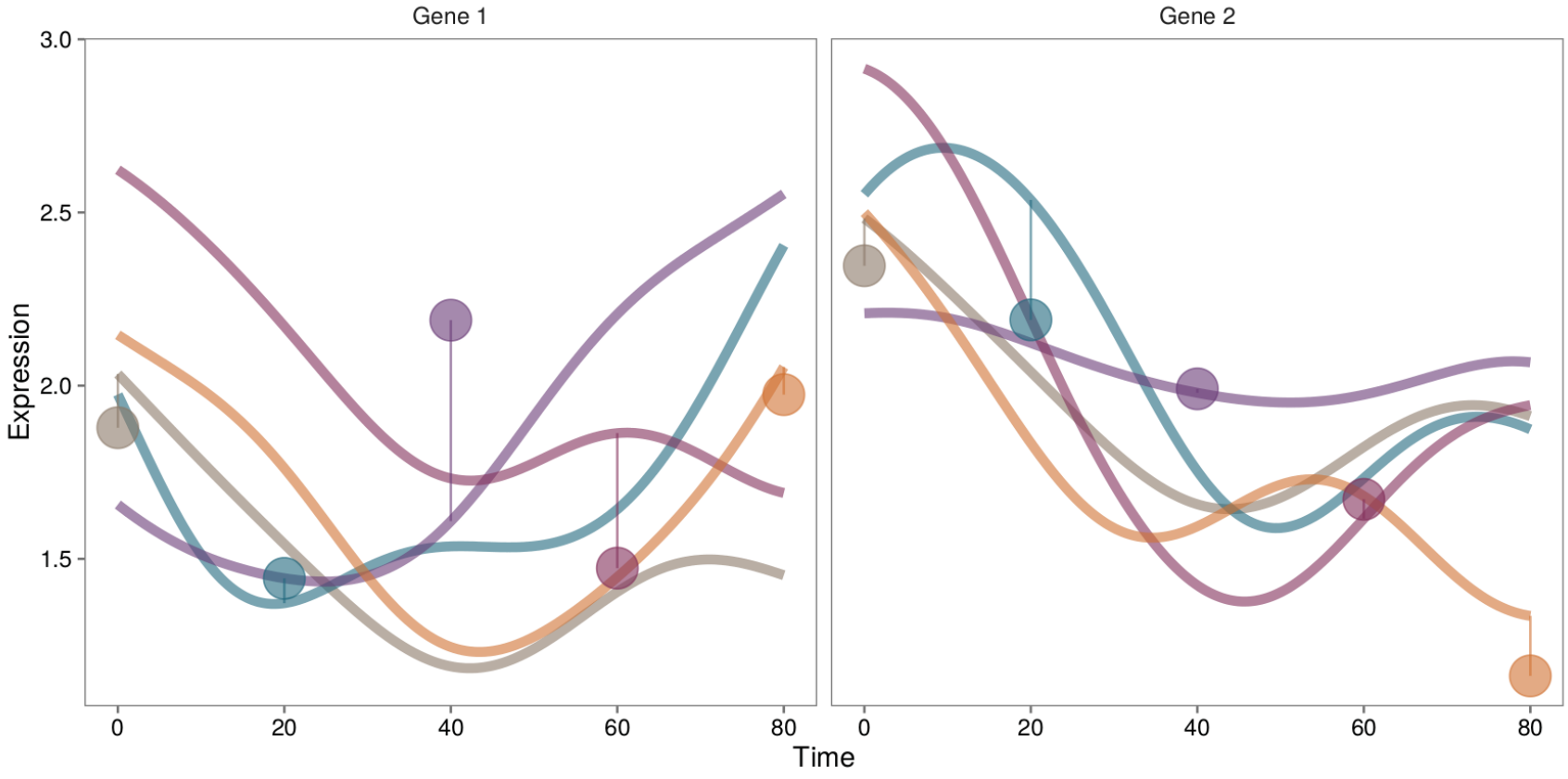


Shalek et al. 2014



Brennecke et al. 2013

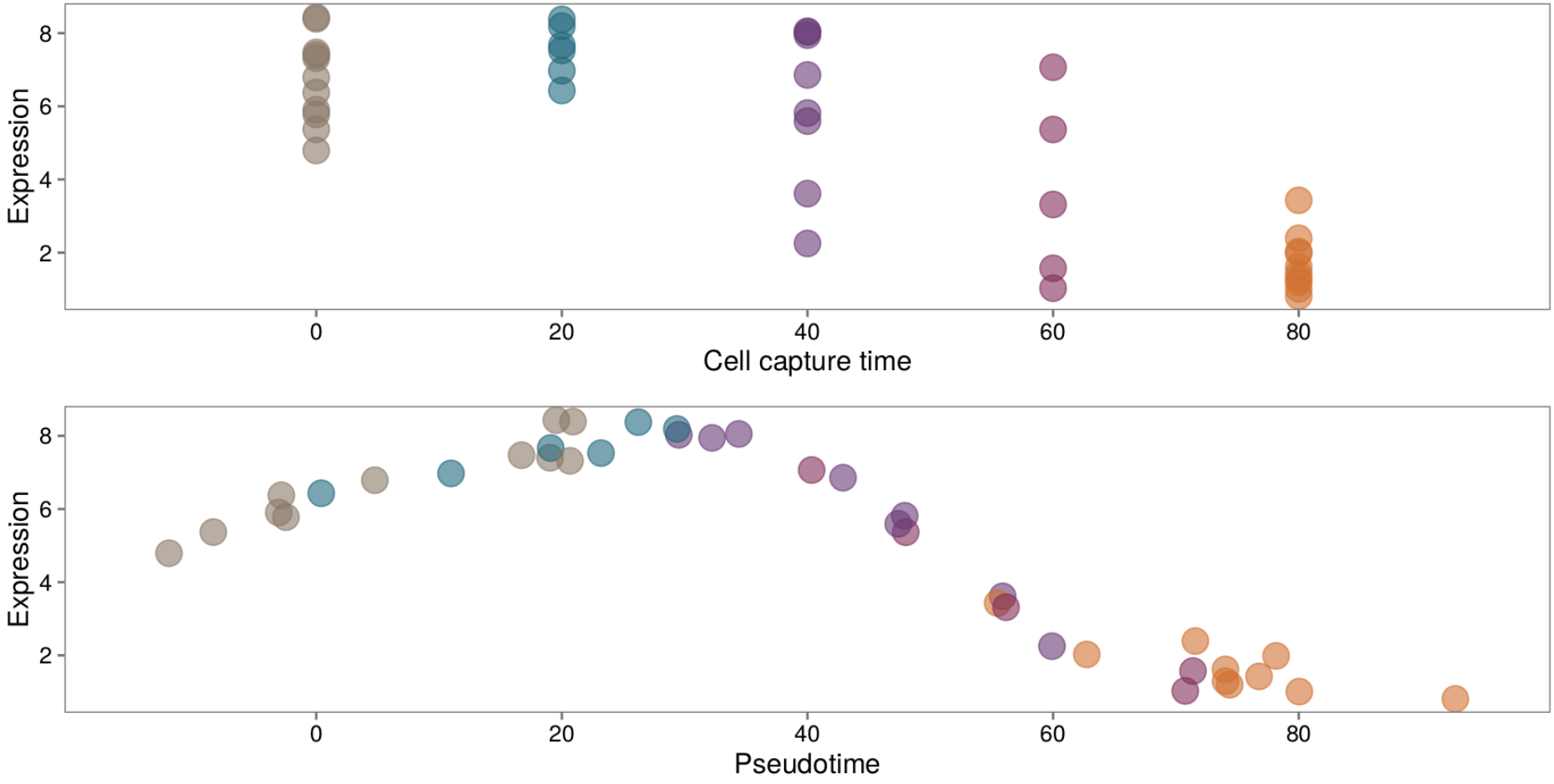
Single cell snapshot data



Deviation from **common** development path:

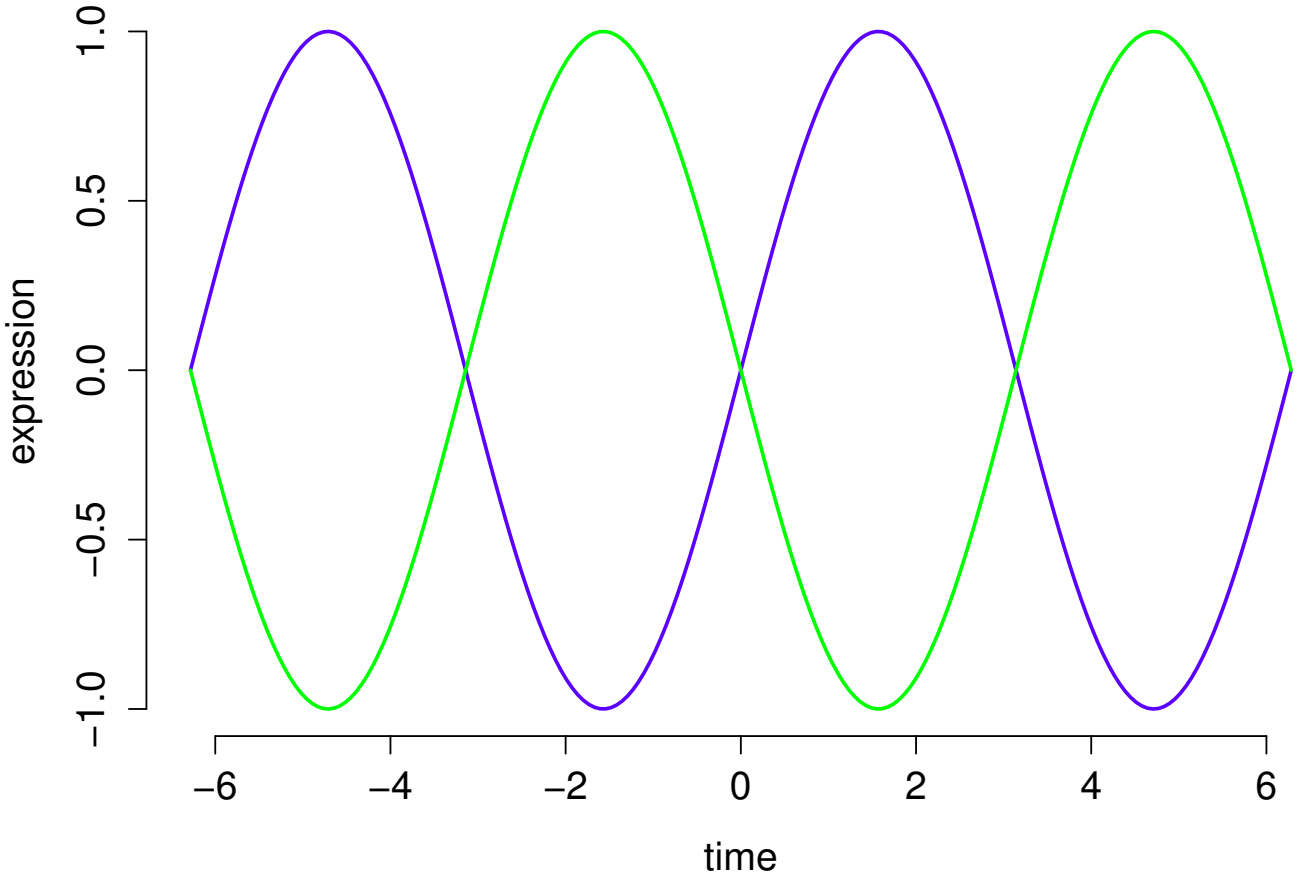
- Individual development of single cells
- Measurement noise

Pseudotime



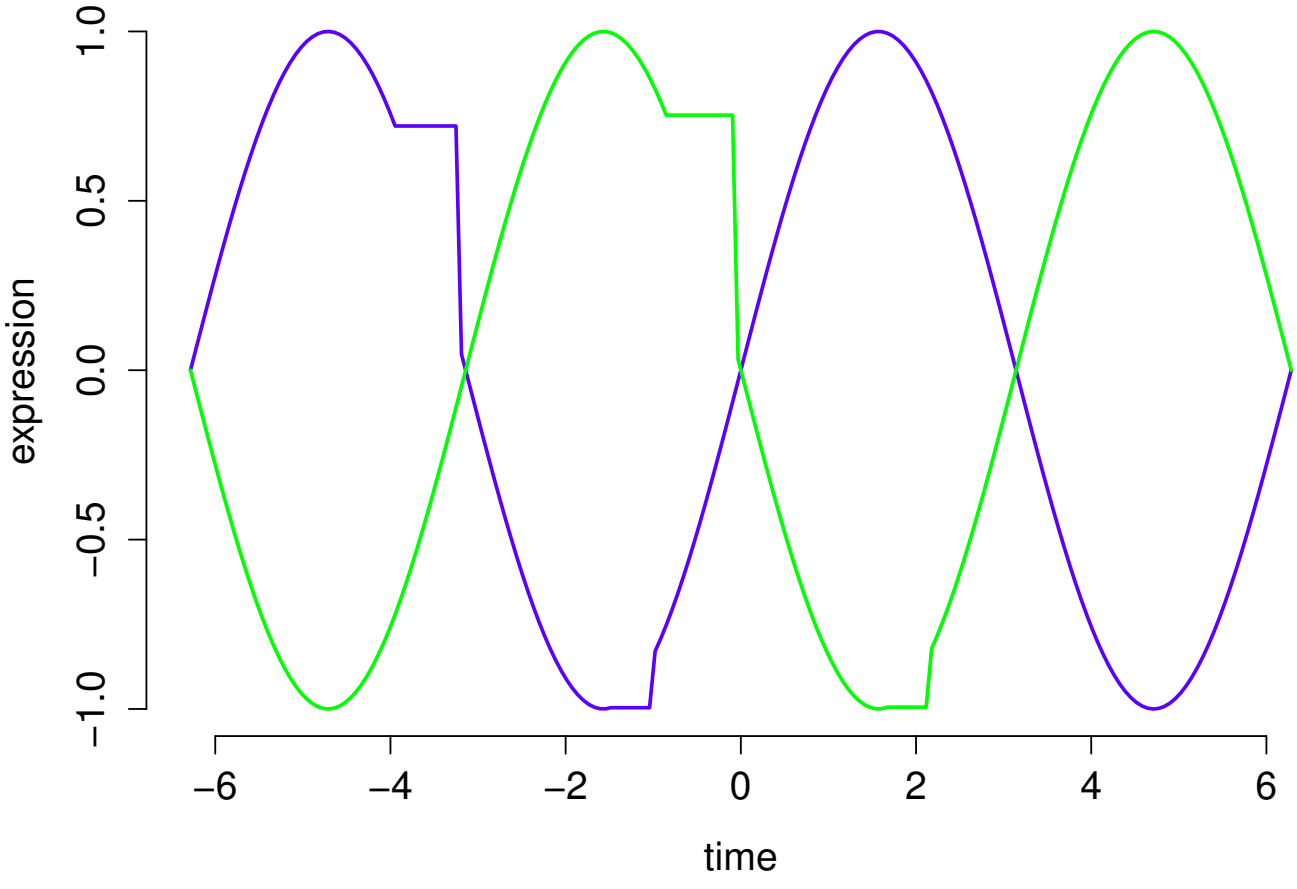
Order cells according to common development path

Causal inference from time series data



Two genes, green and blue. Which is regulating which?

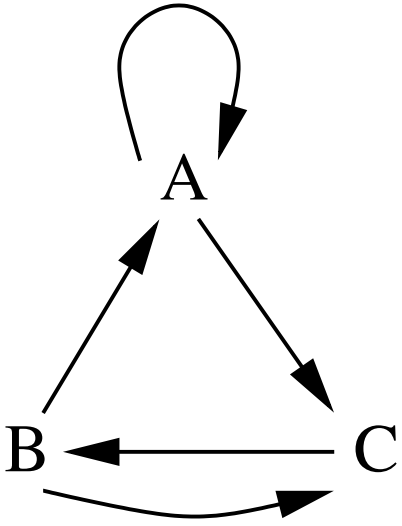
Time series data with noise

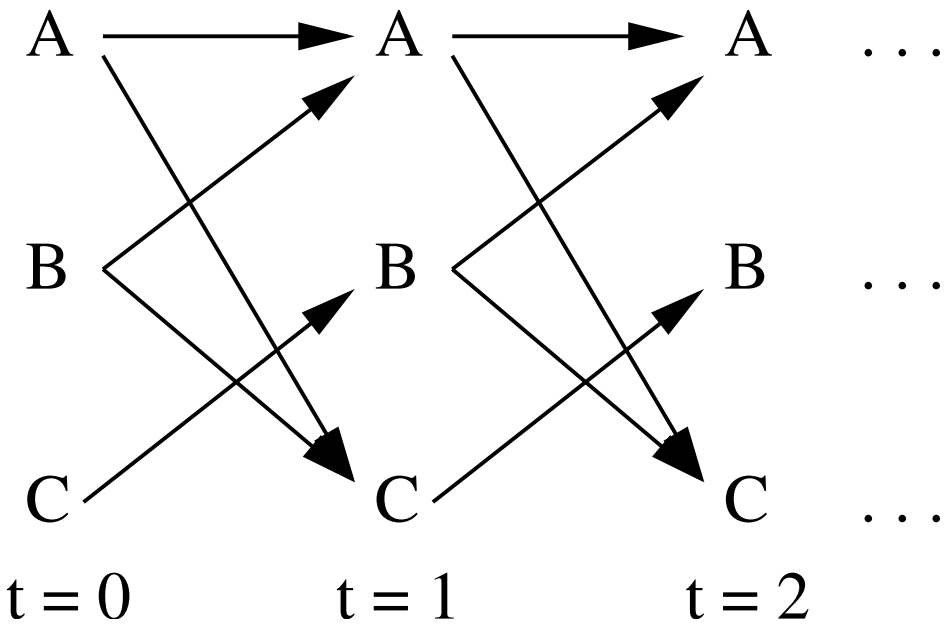


Key feature for causal inference: **process noise**

Resolve circularity with temporal data

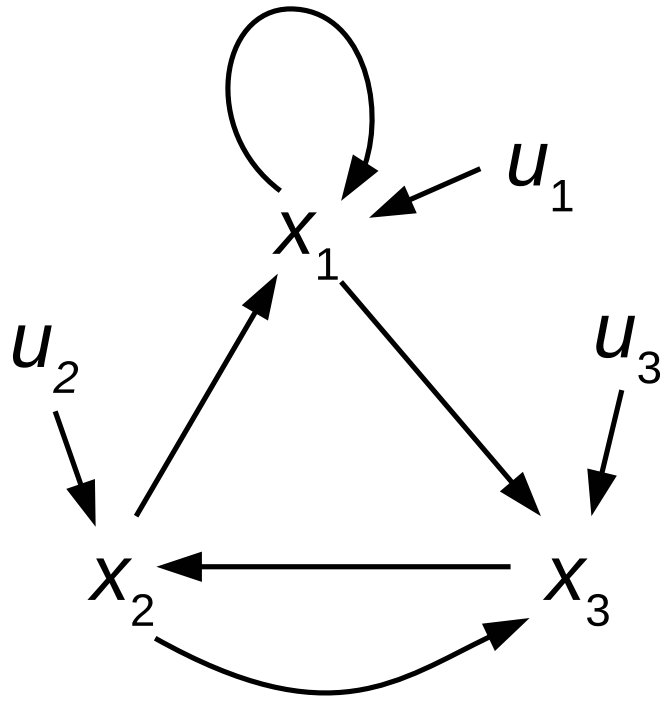
Regress time t on time $t-1$





But what if we only have snapshot data?

Structural Equation Model (SEM)



$x_t = Fx_{t-1} + u$

$u \sim N(0, \Delta)$ fixed over time

Stationary state

$x = Fx + u$

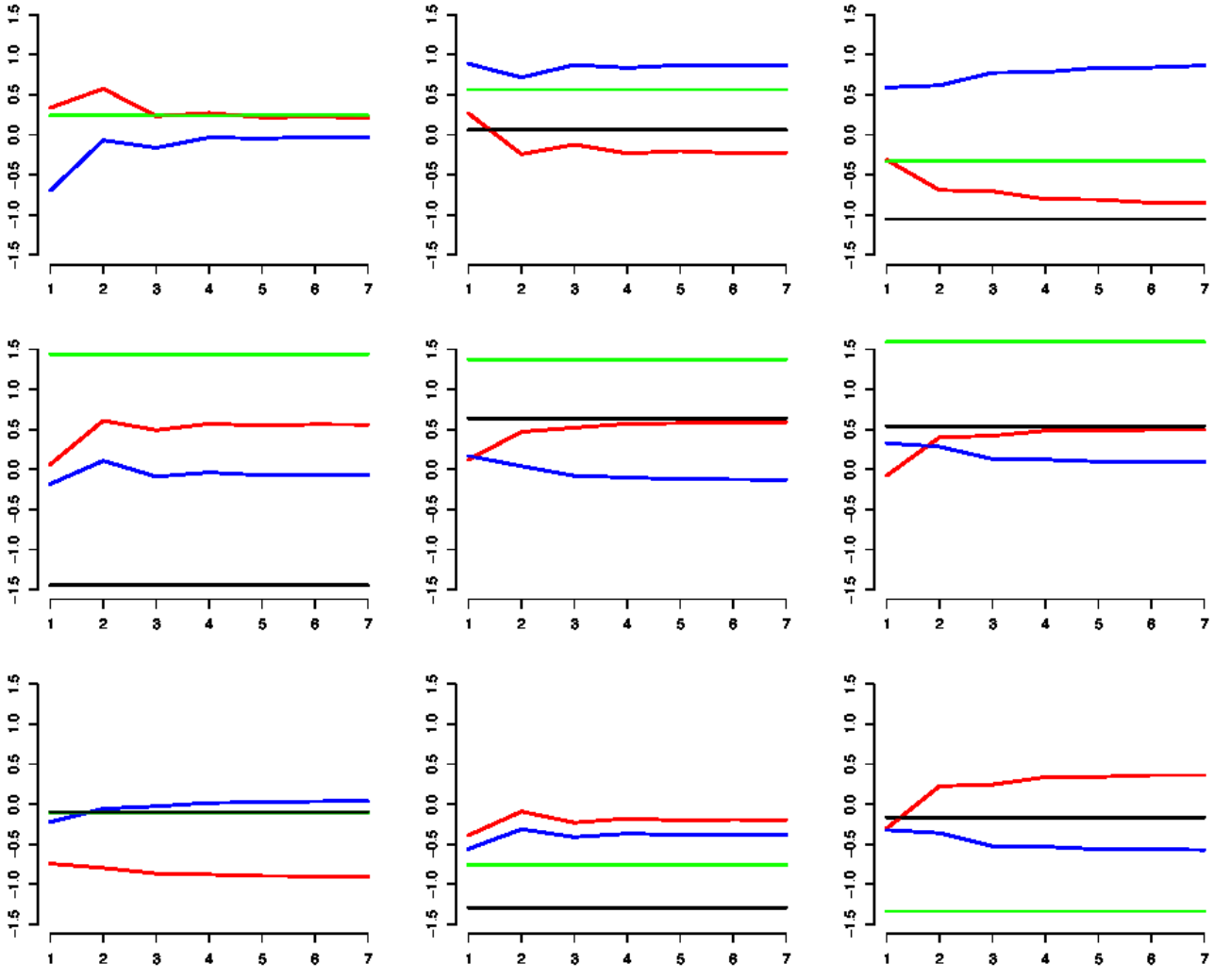
Matrix F, external input u

$x = (I - F)^{-1}u$

Covariance $\text{var}(x) = (I - F)^{-1}\Delta((I - F)^{-1})^T$

Traditional SEM: fit to empirical covariance matrix

Stationary states

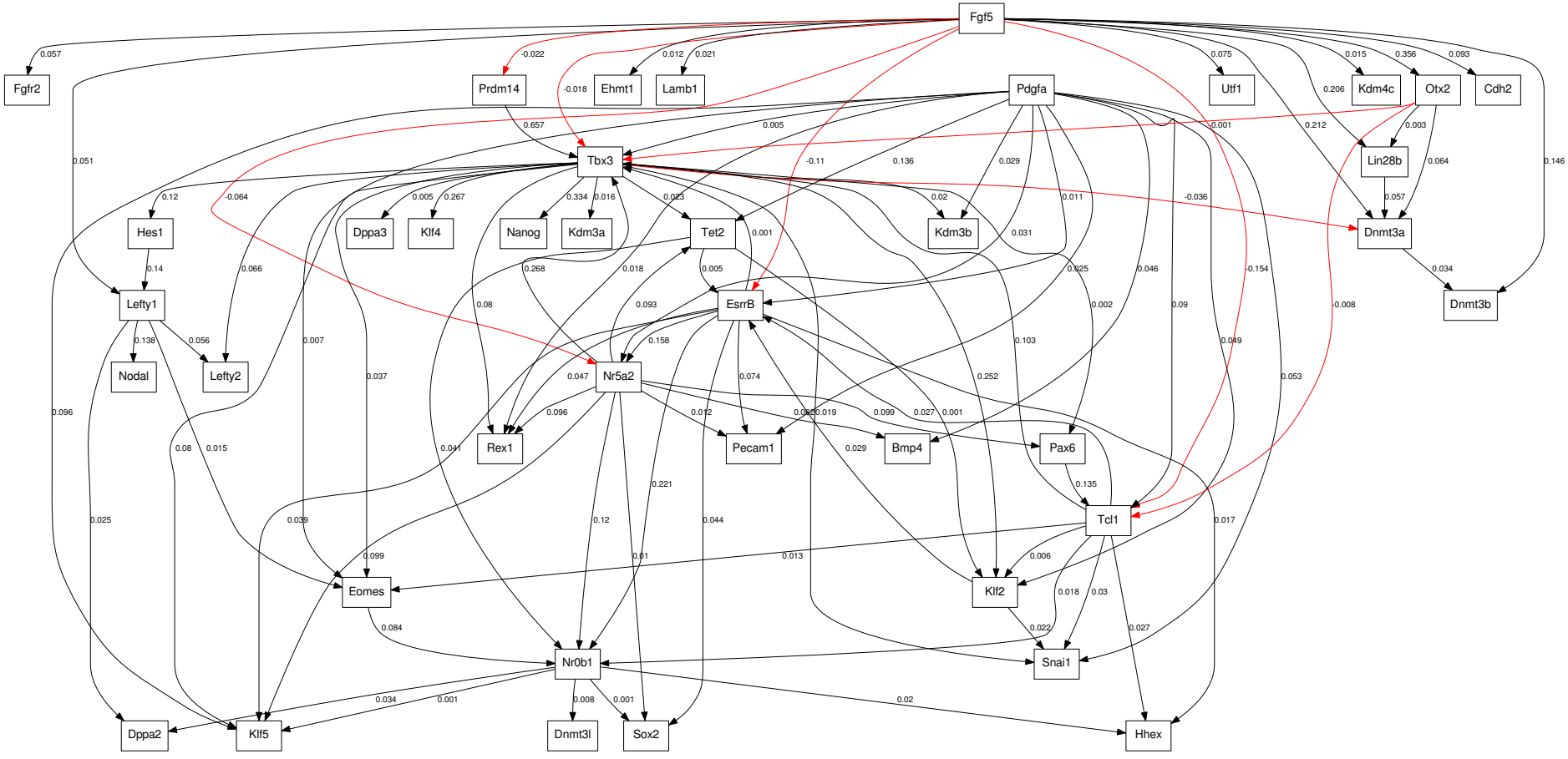


Each cell set off with different input $u_c \sim N(0, \Delta)$

Assume SCs stationary state, regress genes on each other

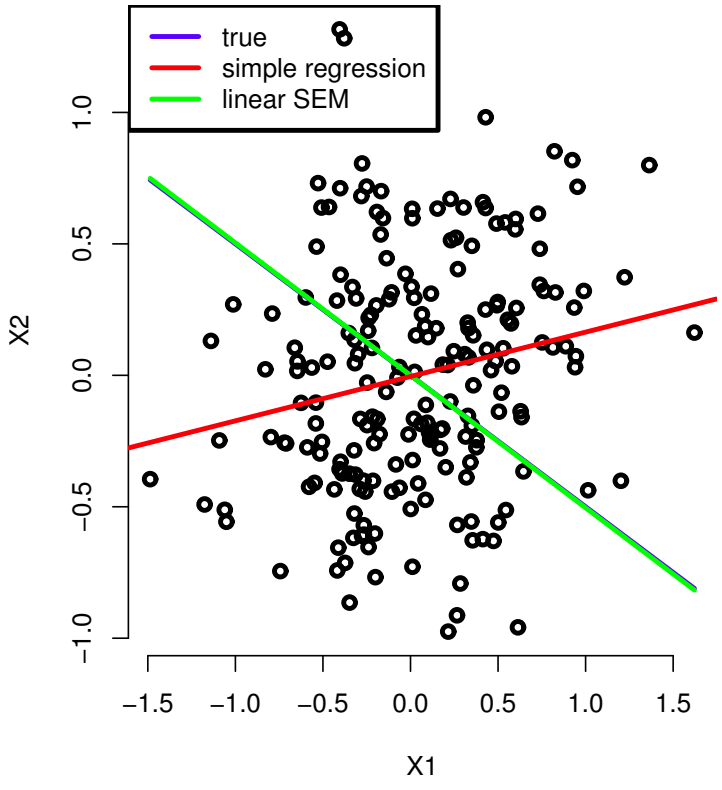
Regression based networks

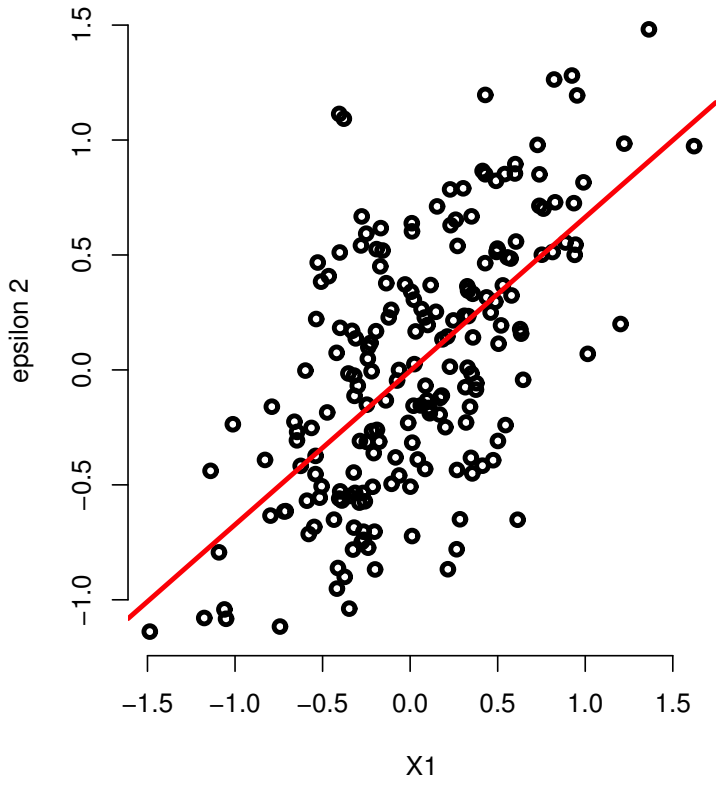
Sparse regression (elastic net with stability selection)



Popular GENIE3, TIGRESS

Simple regression misleading





$$x_1 = 0.8x_2 + u_1$$
$$x_2 = -0.5x_1 + u_2$$

Simple regression (red) misleading

General dynamical framework

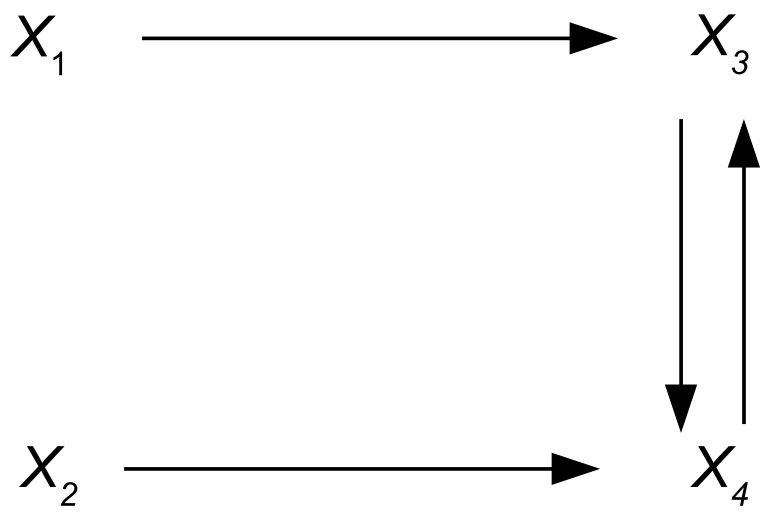
$$x_{1,t} = f_1(x_{t-1}; \theta_1, u_1)$$
$$\dots$$
$$x_{G,t} = f_G(x_{t-1}; \theta_G, u_G)$$

Cell-specific $u = (u_g)$ **constant throughout time** and **independent**

Estimate θ from **stationary state data**

Maximise **independence of residuals** (SEM), instead of **minimizing their size** (regression)

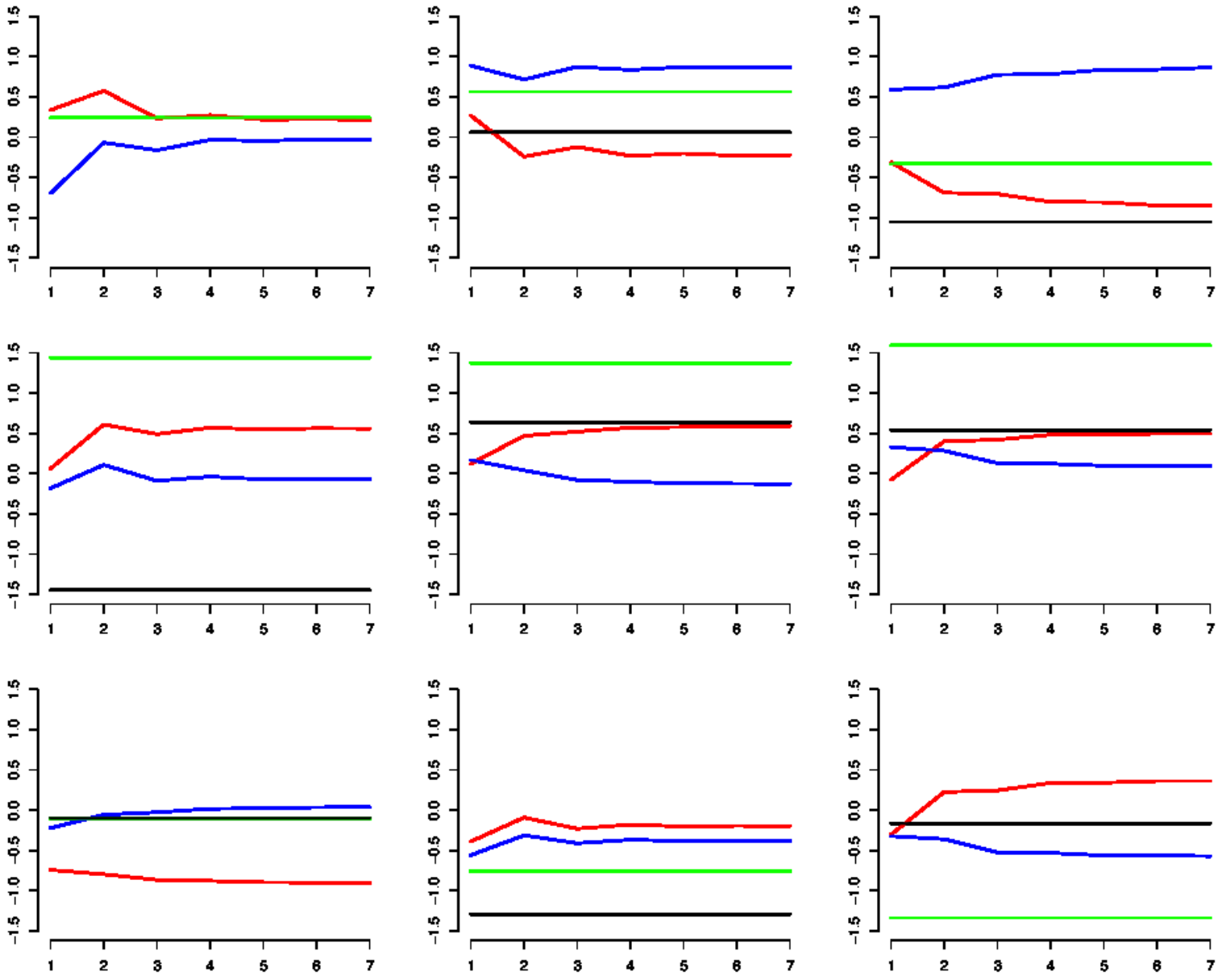
Simulation example



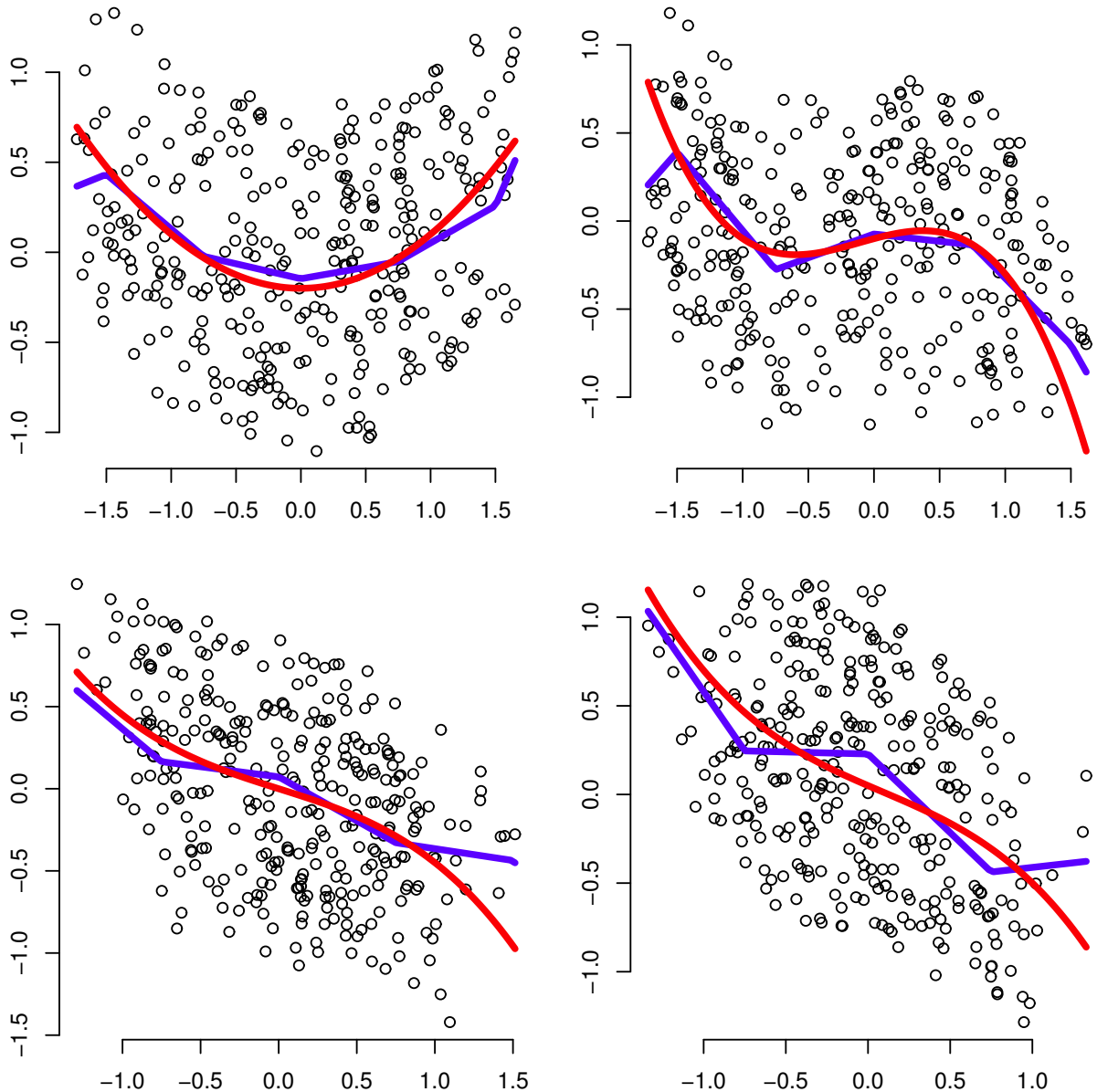
Nonlinear system

$$
\begin{aligned}
x_{1,t} &= \epsilon_1 \\
x_{2,t} &= \epsilon_2 \\
x_{3,t} &= -0.15 + -0.4x_{4,t-1} + 0.3x_{1,t-1}^2 \\
&\quad + 0.05x_{4,t-1}^2 - 0.2x_{4,t-1}^2 + \epsilon_3 \\
x_{4,t} &= -0.1 + 0.2x_{2,t-1} - 0.3x_{3,t-1} - 0.1x_{3,t-1}^2 \\
&\quad - 0.3x_{2,t-1}^3 - 0.15x_{3,t-1}^3 + \epsilon_4 \\
\epsilon_1, \epsilon_2 &\sim \text{Unif}(-1.73, 1.73) \\
\epsilon_3, \epsilon_4 &\sim \text{Unif}(-0.95, 0.95)
\end{aligned}
$$

300 dynamic simulations



Network with maximal independence

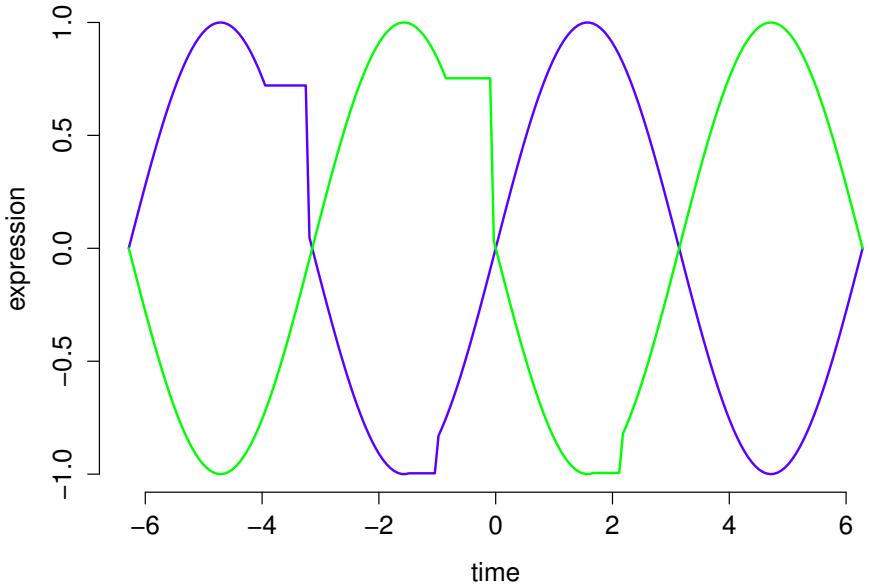


MCMC search: network structure, and parameters

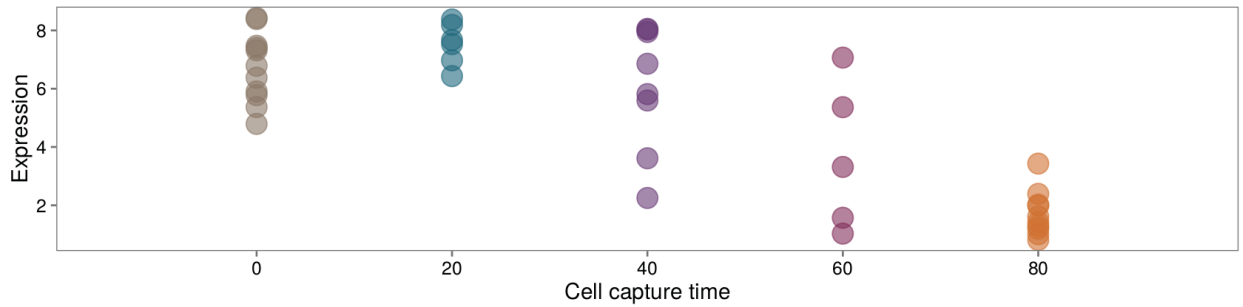
General independence criteria (eg HSIC)

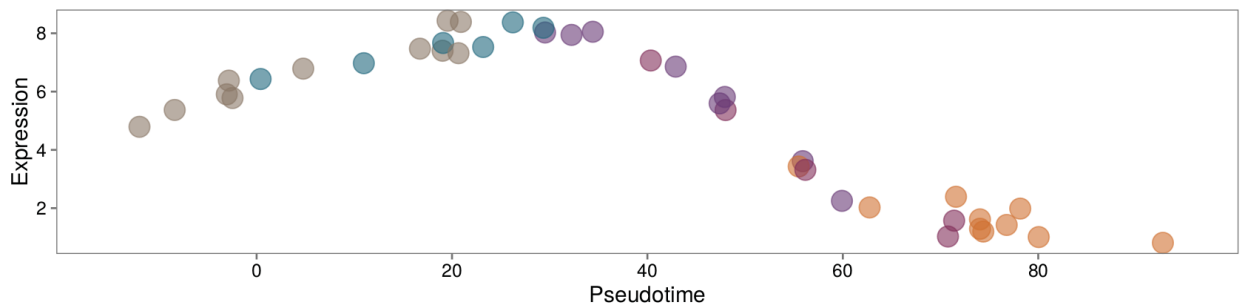
Petras Verbyla

Beyond snapshot data

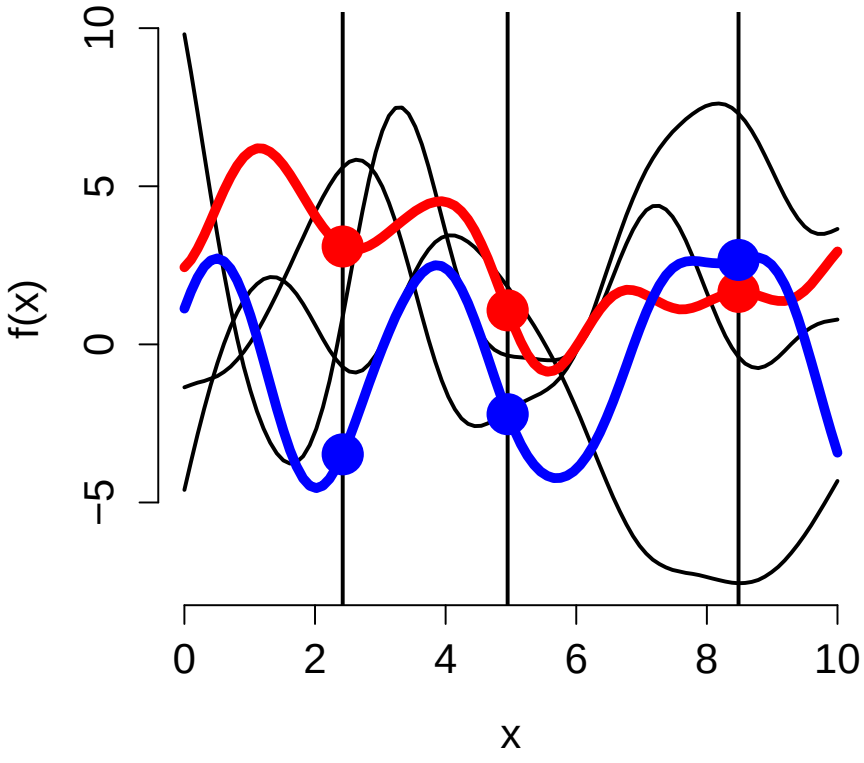


Obtain pseudotime
"dynamical data"





Gaussian process prior



Family of functions via covariance K on input points x

$$y \sim N(0, K_{xx})$$

Prediction for x^* from (x, y)

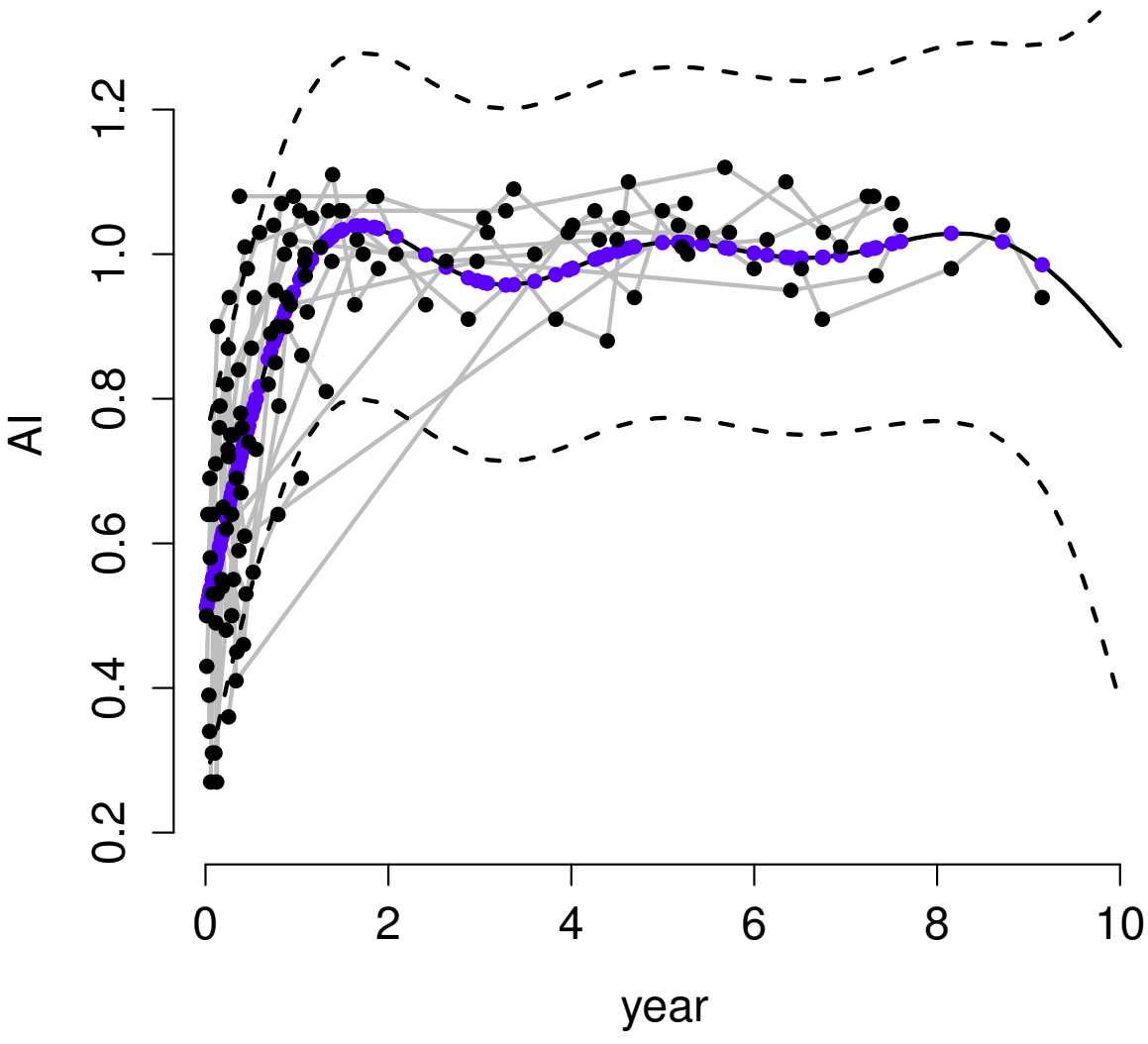
$$y^* \sim N(K_{x^*x}K_{xx}^{-1}y, \Sigma)$$

$$\Sigma = K_{x^*x^*} - K_{x^*x}K_{xx}^{-1}K_{xx^*}$$

Gaussian $\mathrm{cov}(x, x^*) = \theta_1 \exp(-\theta_2(x - x^*)^2)$

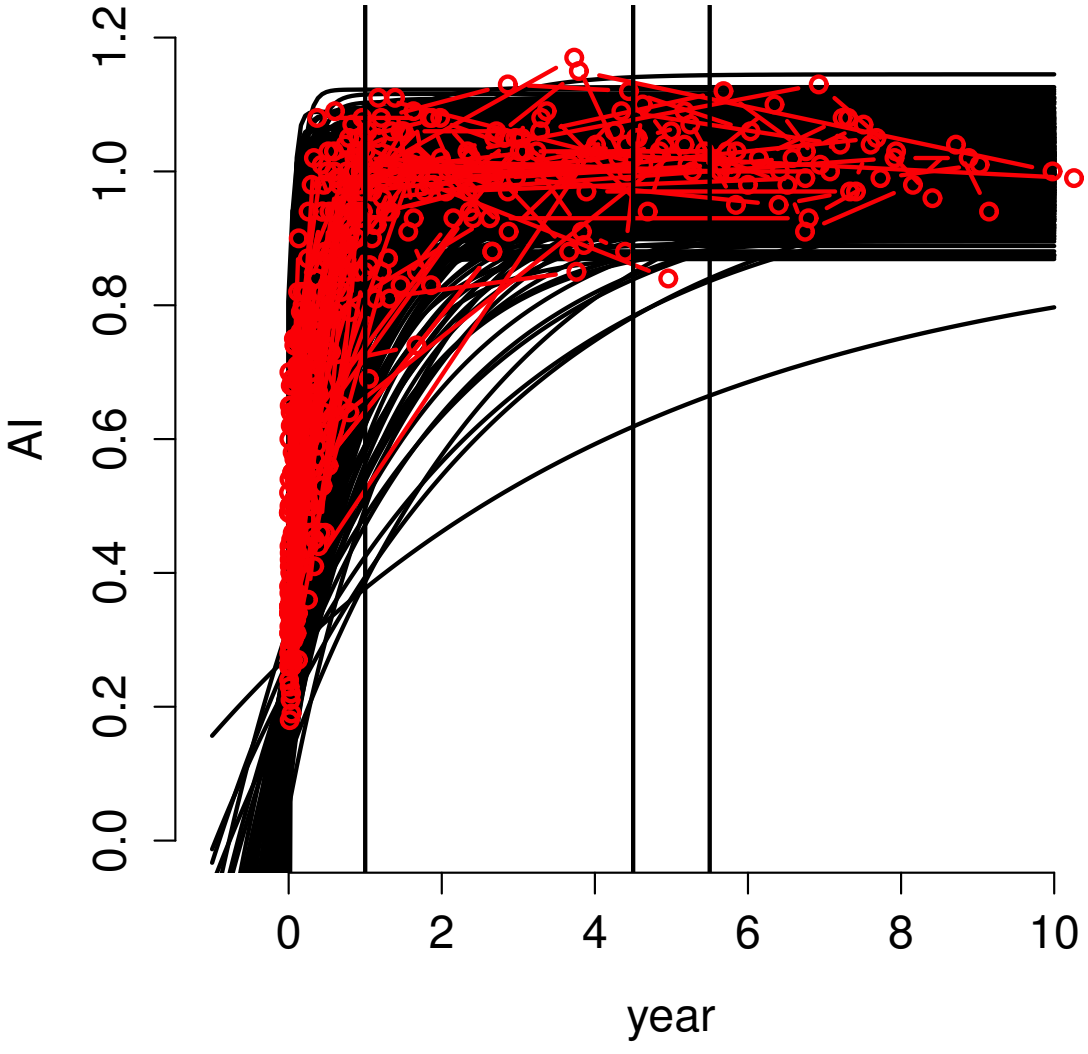
Matern $\mathrm{cov}(x, x^*) = \theta_1(1 + \theta_2|x - x^*|)\exp(-\theta_2|x - x^*|)$

Hierarchical GP with Gaussian kernels



Prediction (blue) based on GP common to all individuals

Covariance kernel from functional model



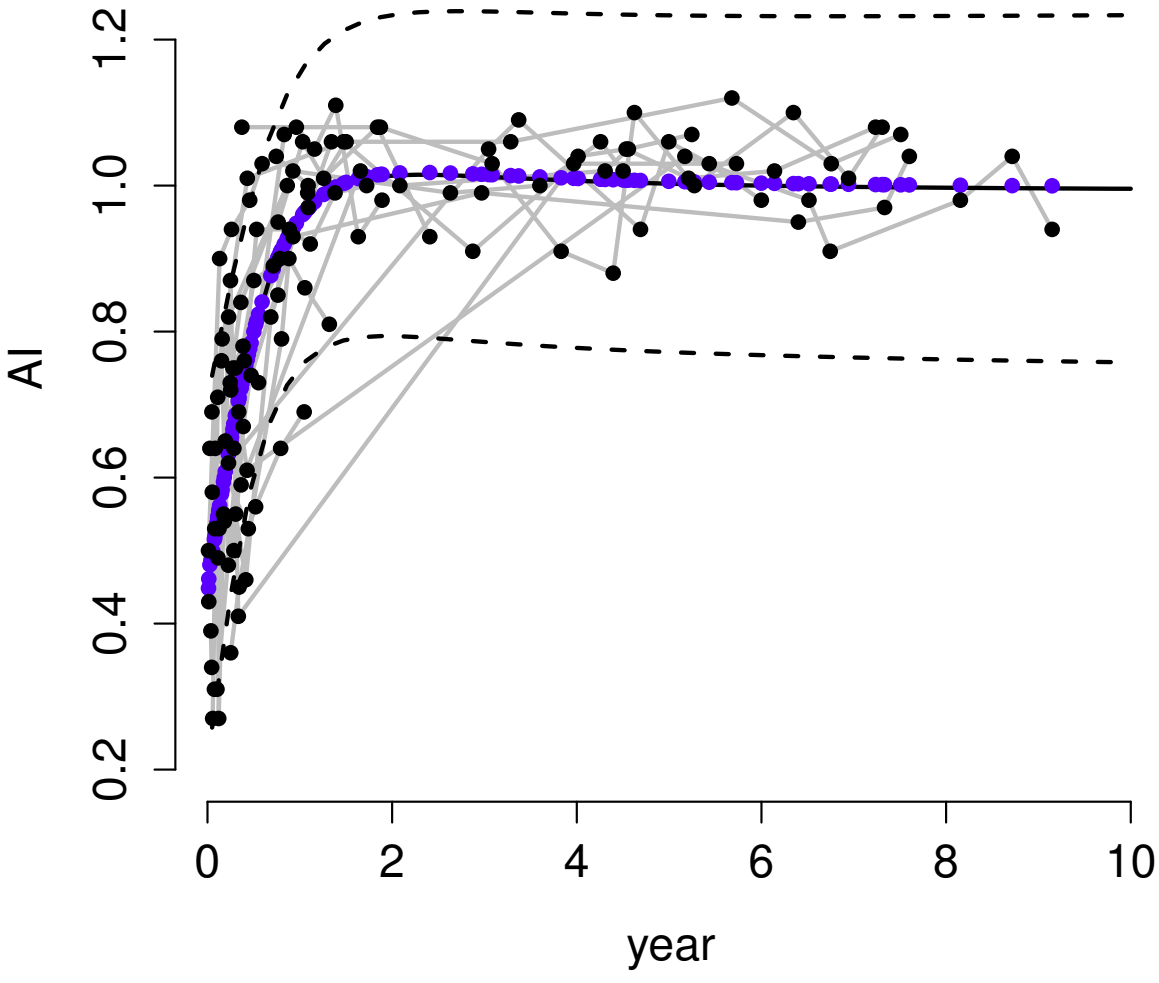
Parametric functions:

$$a+(b-a)(1-e^{-\lambda(t-t0)})$$

Sample shapes from parameter priors

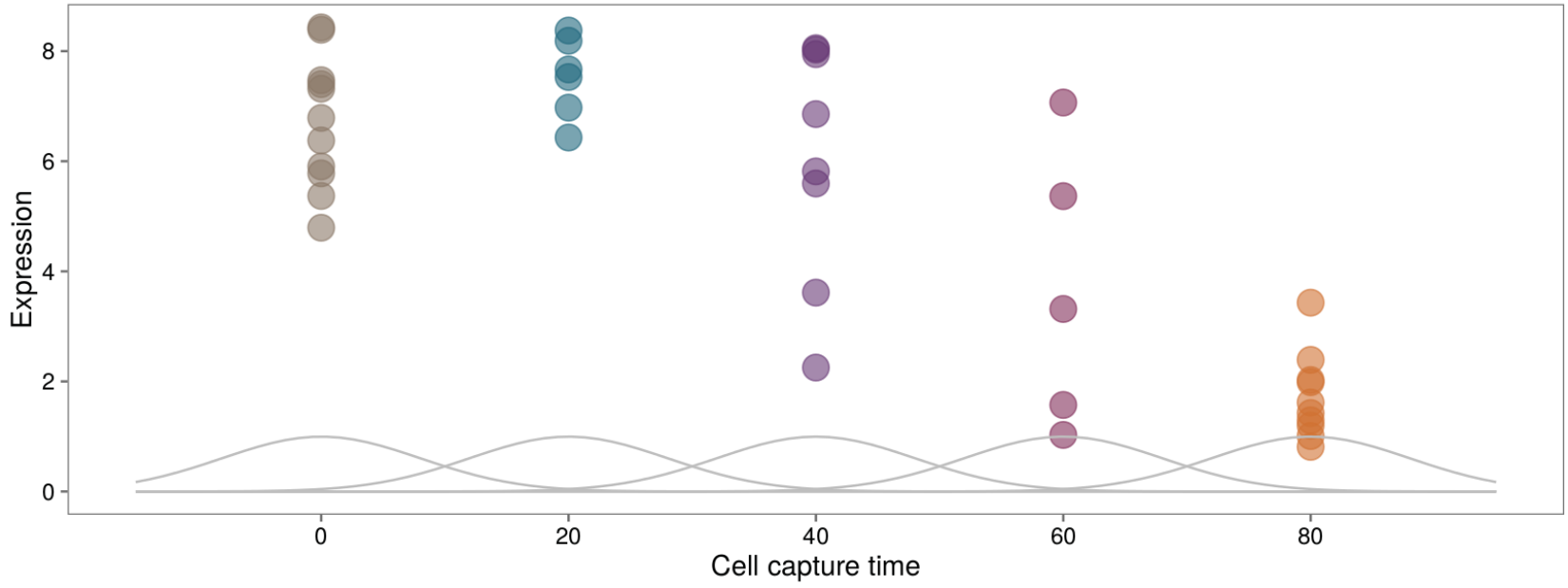
Estimate **covariance matrix** for points of interest

Hierarchical functional GP model



Use "functional" covariance from simulation for functional forms

GP for pseudotime

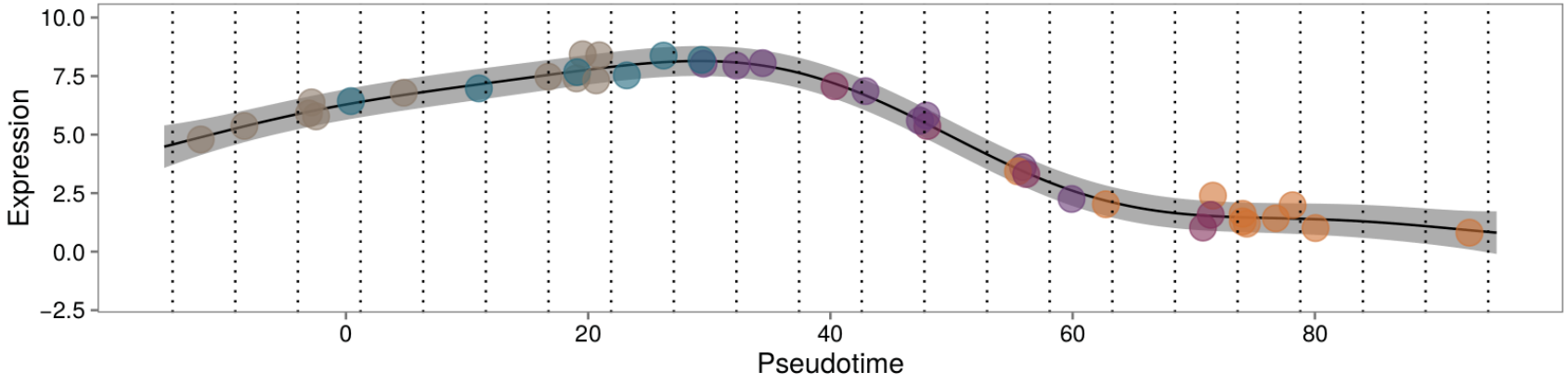


Find time points t_c for cell c, gene g expression $x_c^{(g)}$

Priors on t_c, GP likelihood on points $(t_c, x_c^{(g)})$

$$p(x, t, \theta) = p(\theta) \prod_{\text{cell } c} p(t_c) \prod_{\text{gene } g} p_N(x^{(g)} \mid 0, K(t, \theta))$$

Pseudotime ordering



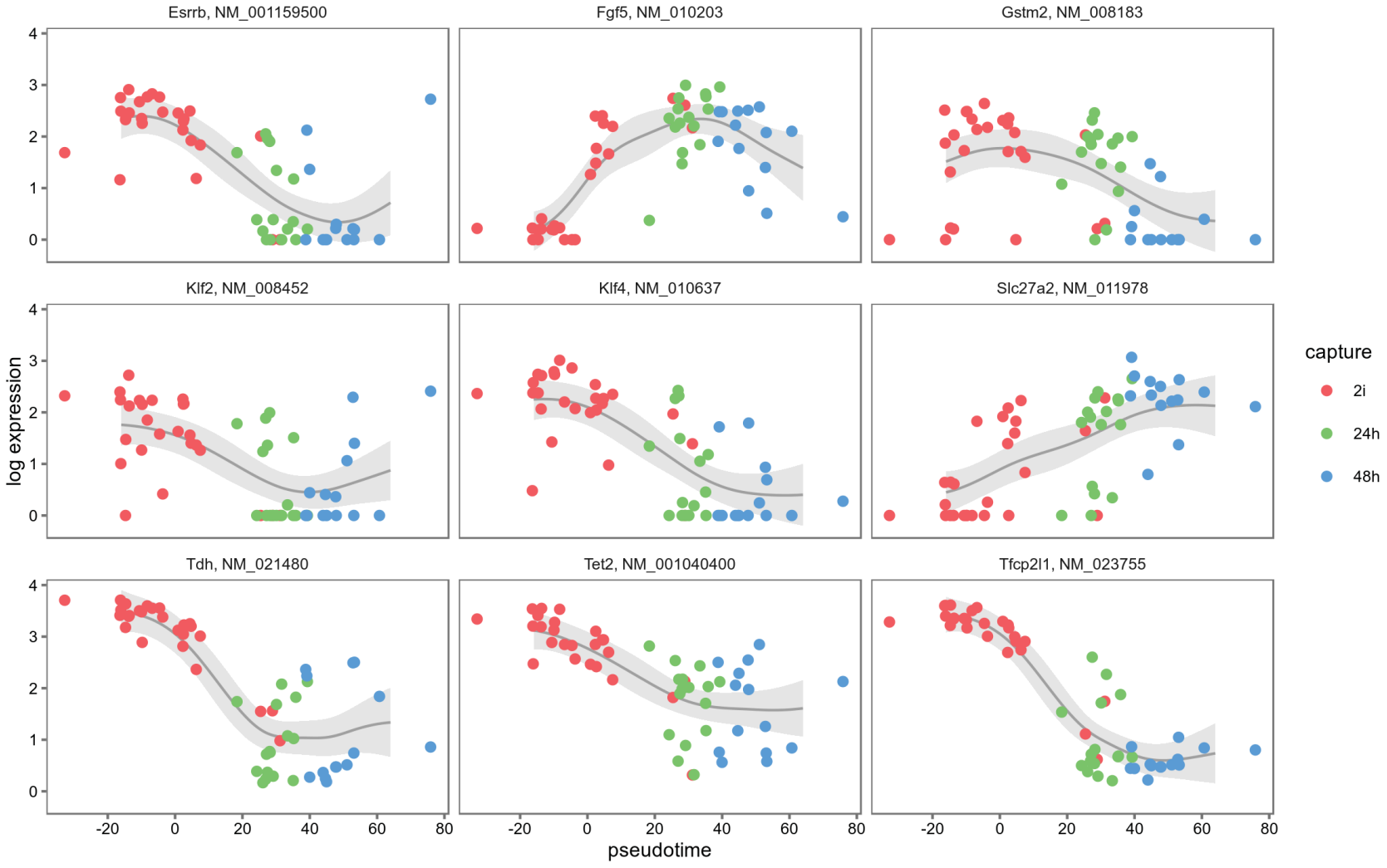
Sparse GP with inducing grid points u

$x^{(g)} \sim N(0, Q_{tt} + \mathrm{diag}(K_{tt} - Q_{tt}))$

$Q_{tt} = K_{tu} K_{uu}^{-1} K_{ut}$ (K_{uu} small, easy to invert)

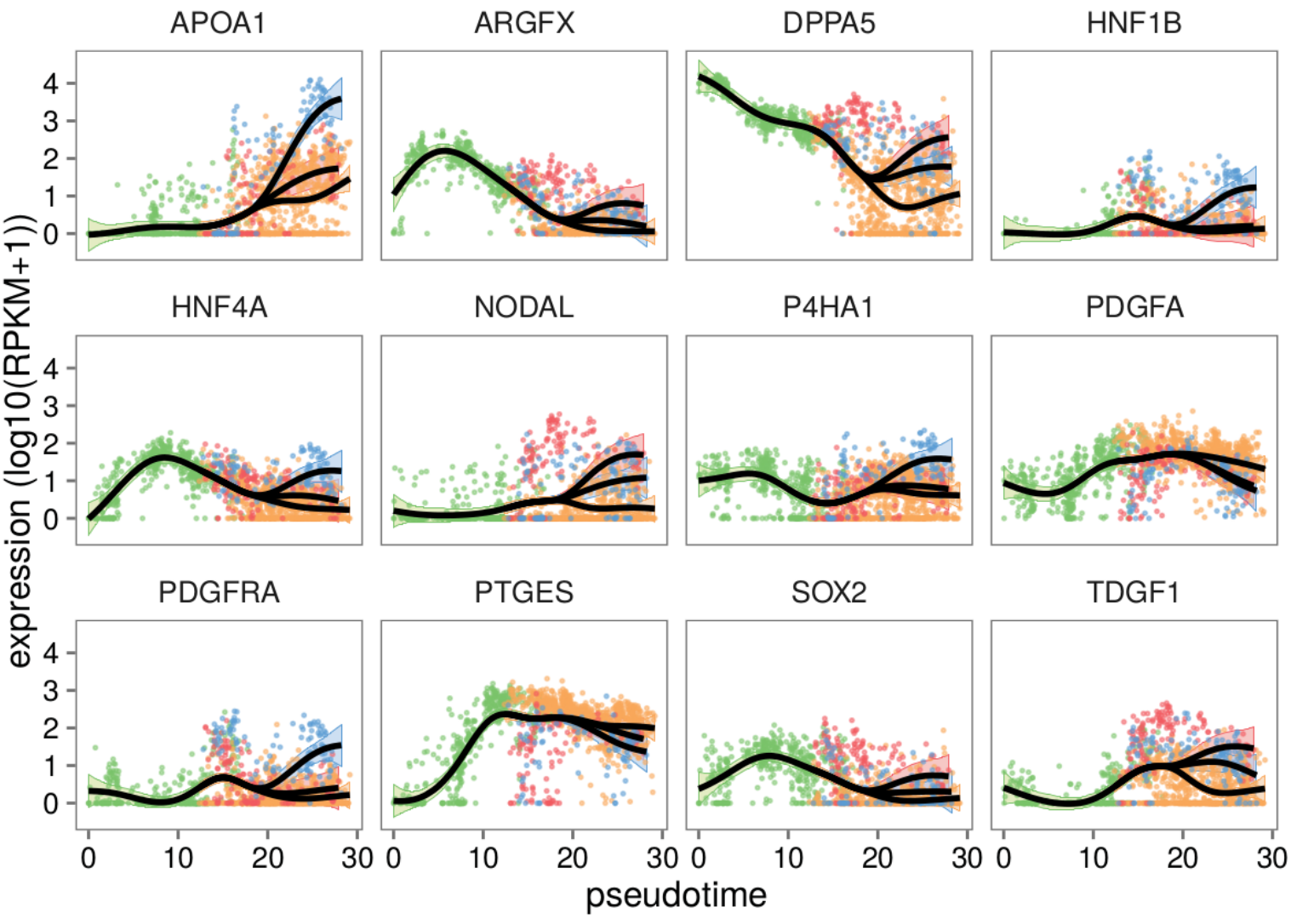
Reid and Wernisch, Bioinformatics 2016
CRAN package DeLorean, Magdalena Strauss

Embryonic stem cells



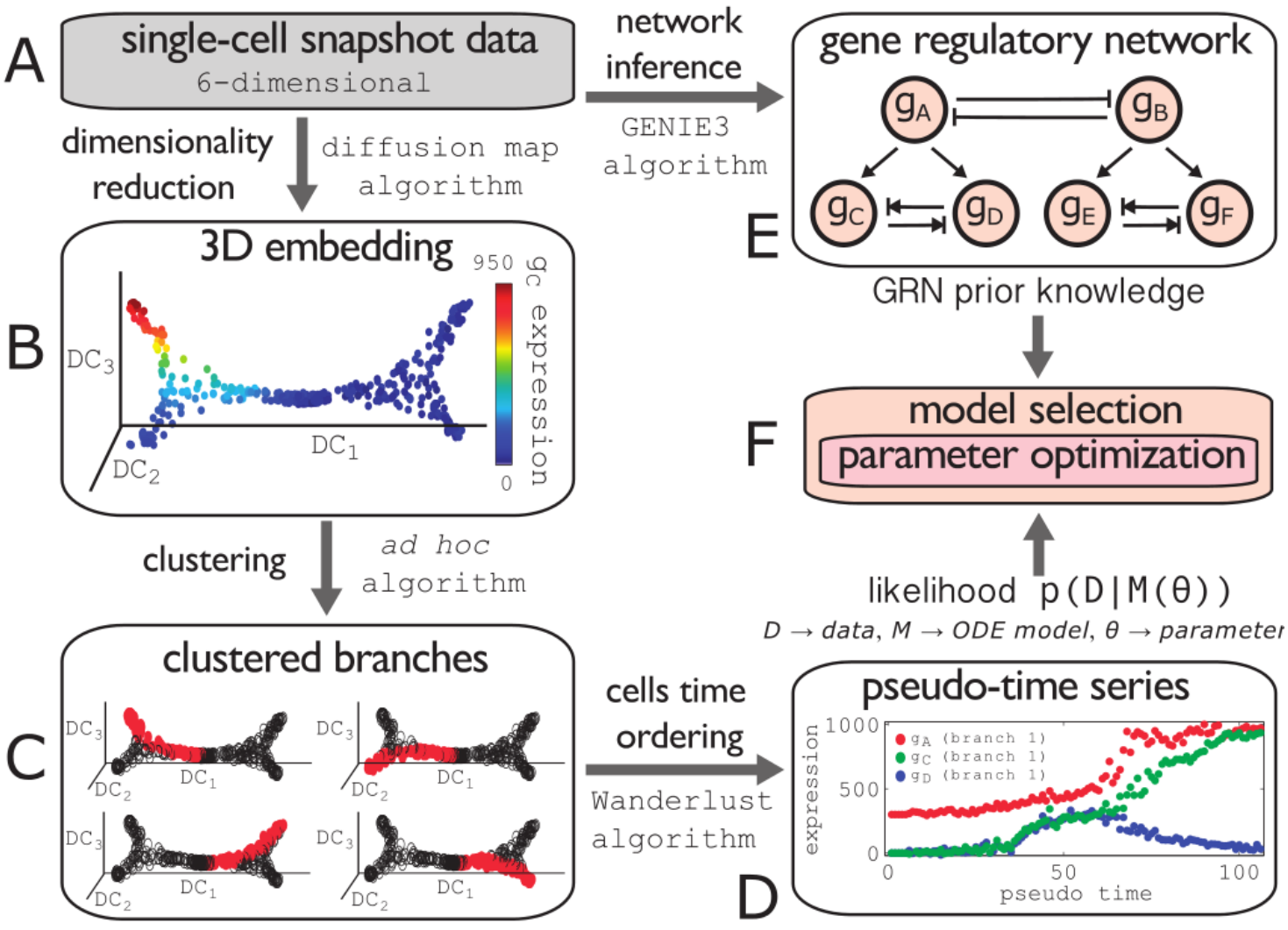
ESCs to PGLCs, Julia Tischler (Gurdon Institute)

Branching GPs



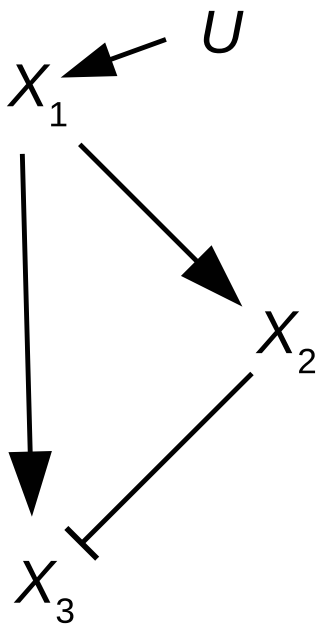
Human embryonic cells, Petropoulos et al. 2016

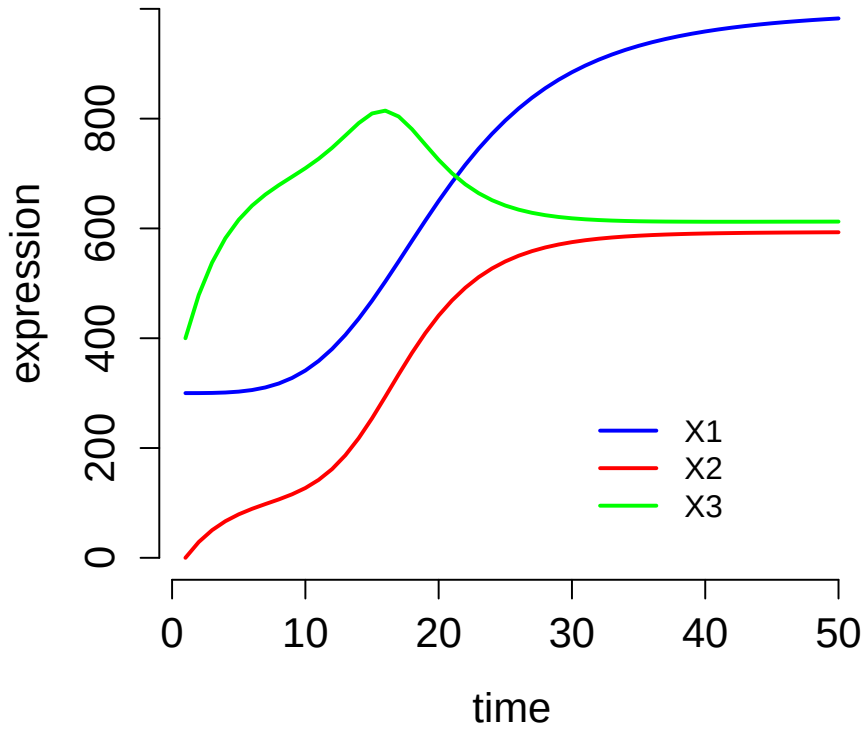
Networks from pseudotime



Ocone, Haghverdi, Mueller, Theis, Bioinf. 2015

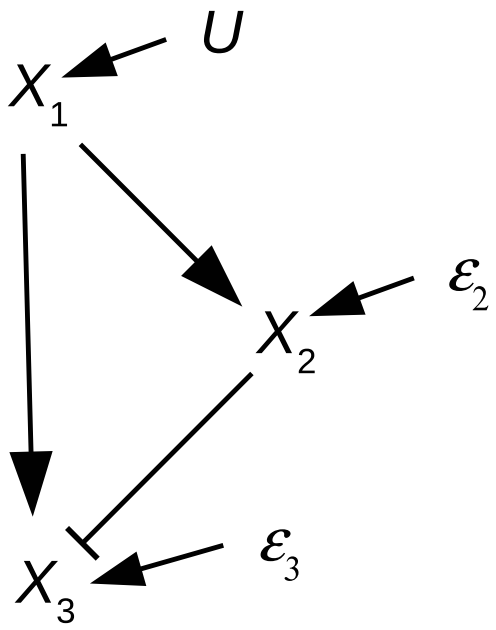
Feedforward loop without process noise

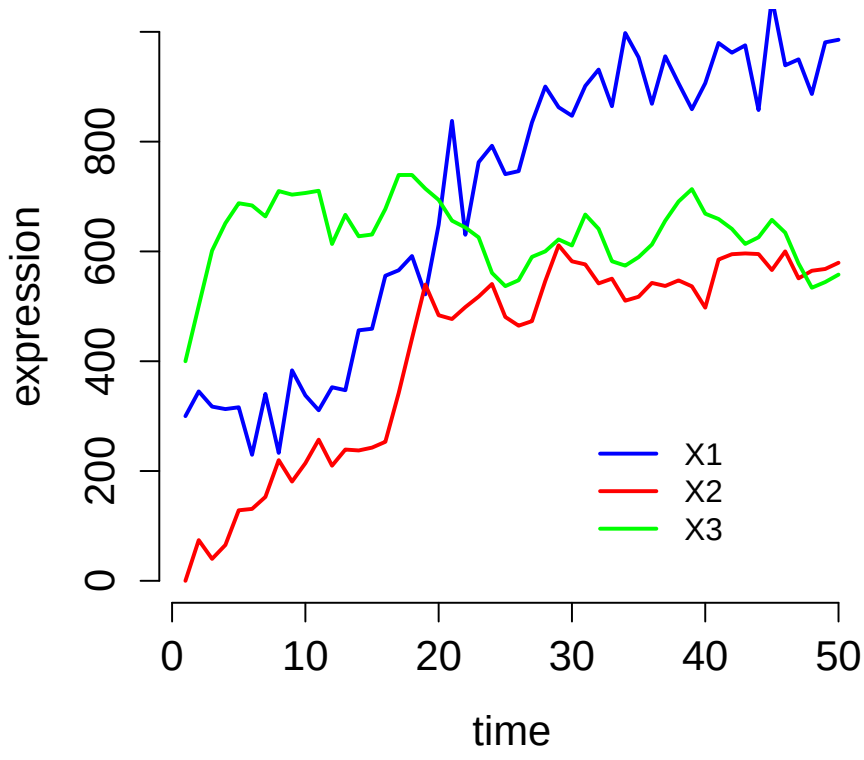




$$X_1(t) = (1-\lambda_1)X_1(t-1) + U_{\text{activate}}(t)$$
$$X_2(t) = (1-\lambda_2)X_2(t-1) + h^+(X_1(t-1))$$
$$X_3(t) = (1-\lambda_3)X_3(t-1) + h^+(X_1(t-1)) + h^-(X_2(t-1))$$

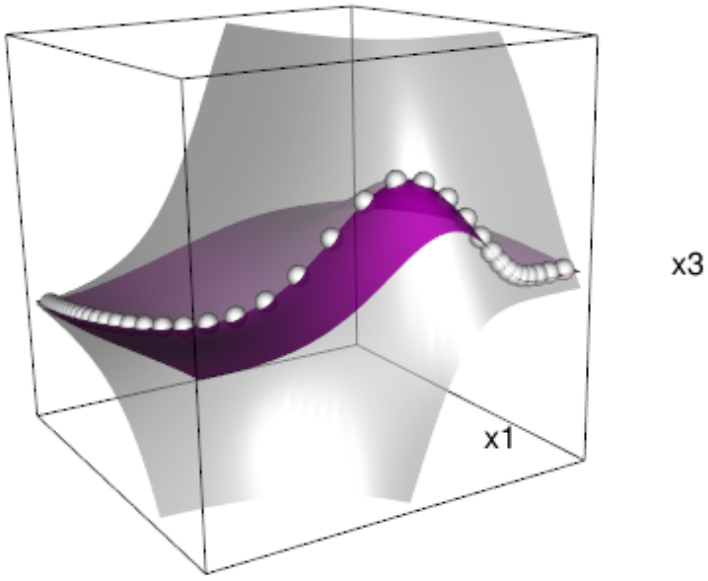
Feedforward loop with process noise



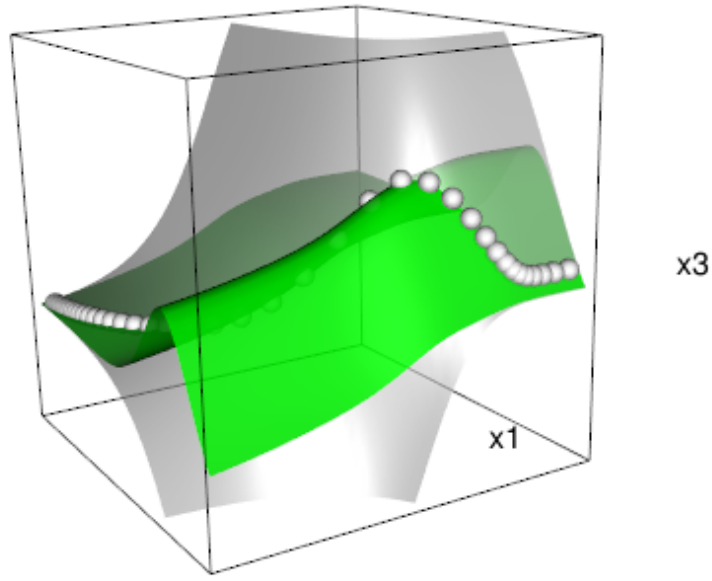


$$X_1(t) = (1-\lambda_1)X_1(t-1) + U_{\epsilon,\text{activate}}(t)$$
$$X_2(t) = (1-\lambda_2)X_2(t-1) + h^+(X_1(t-1)) + \epsilon_2(t)$$
$$X_3(t) = (1-\lambda_3)X_3(t-1) + h^+(X_1(t-1)) + h^-(X_2(t-1)) + \epsilon_3(t)$$

Approximating the transition function



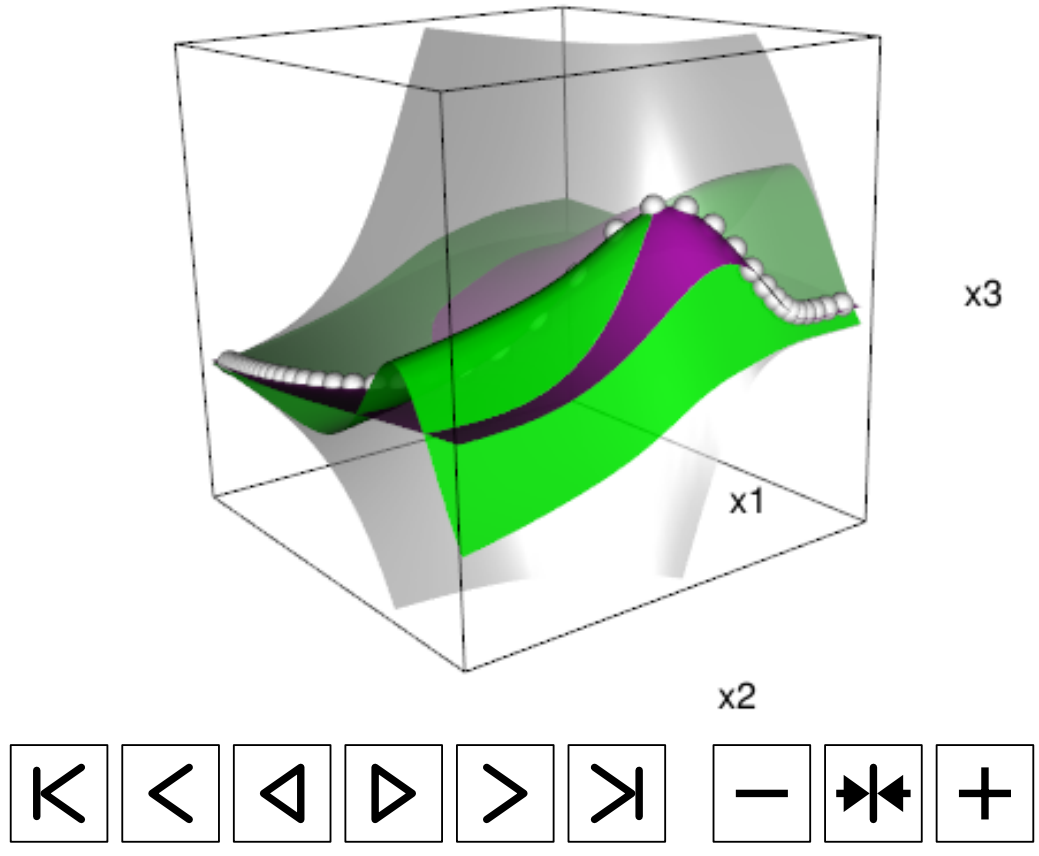
Gauss Cov



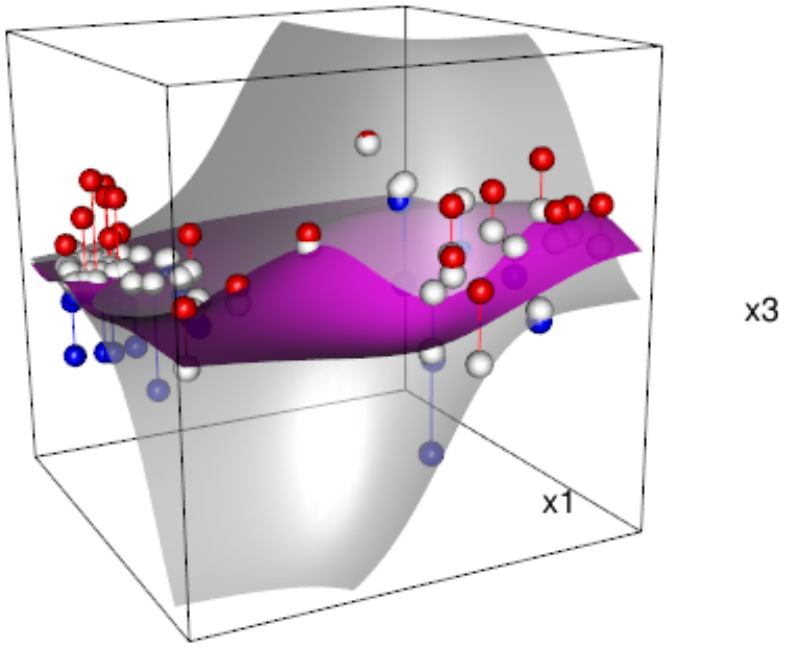
Functional Cov

Grey: $X_3(t) = f_{\text{hill}-\text{or}}(X_1(t-1), X_2(t-1))$
Colored: GP with covariance trained on
$(X_1(t-1), X_2(t-1)) \rightarrow X_3(t)$

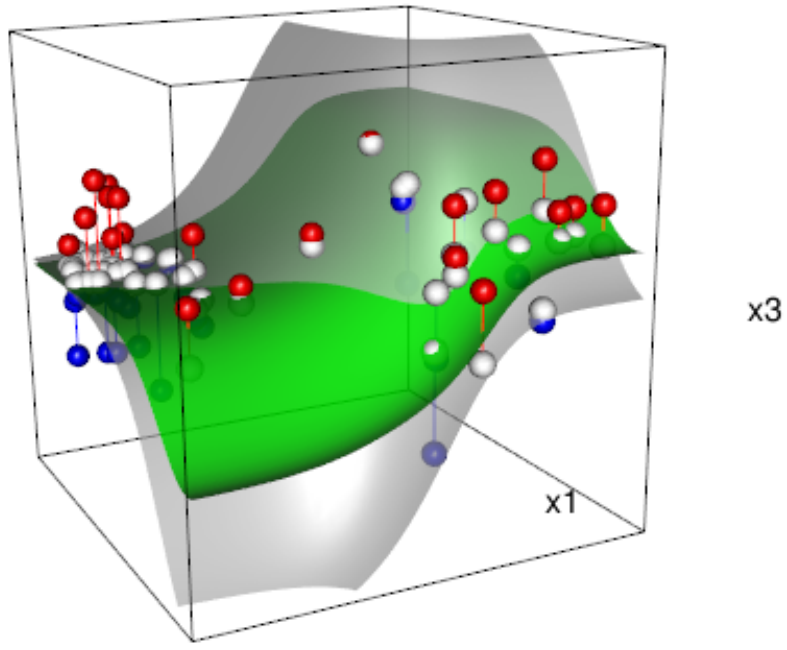
GP approximation



Approximating the transition function



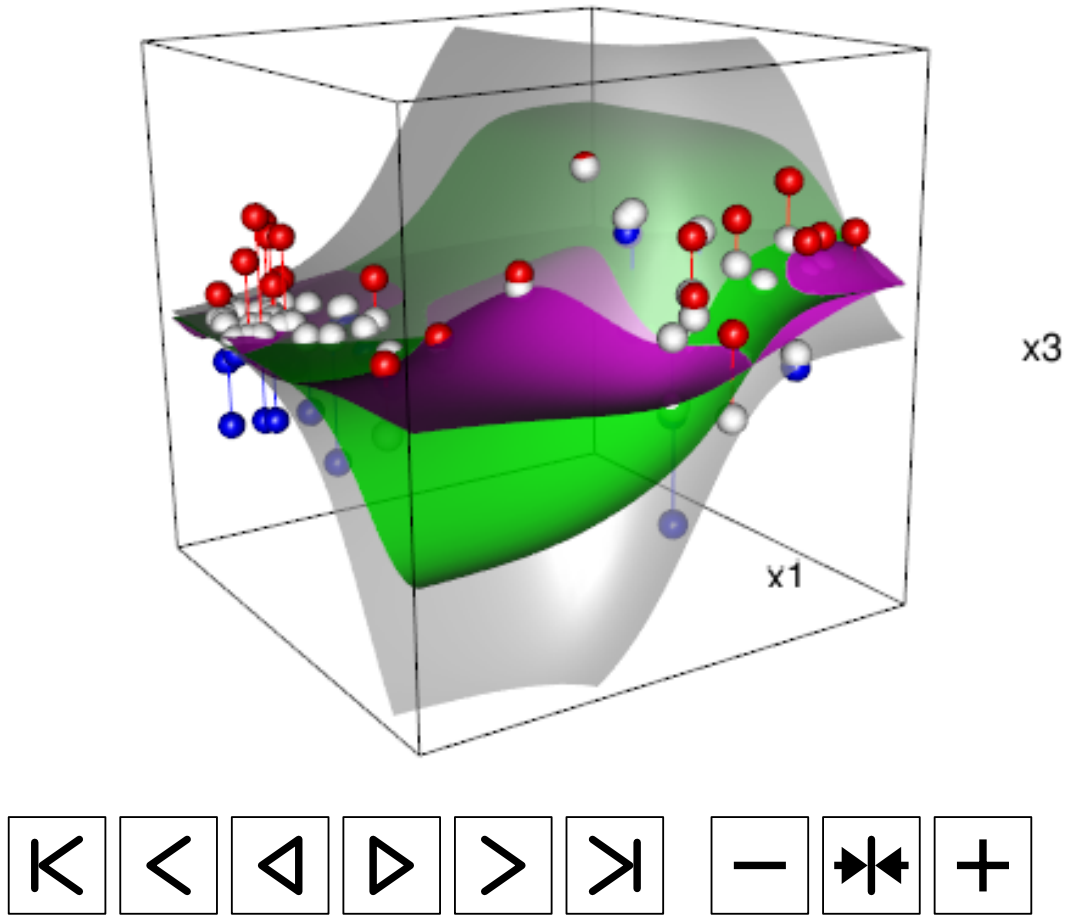
Gauss Cov



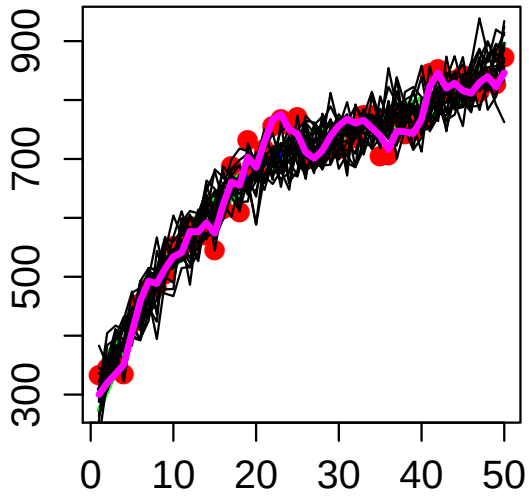
Functional Cov

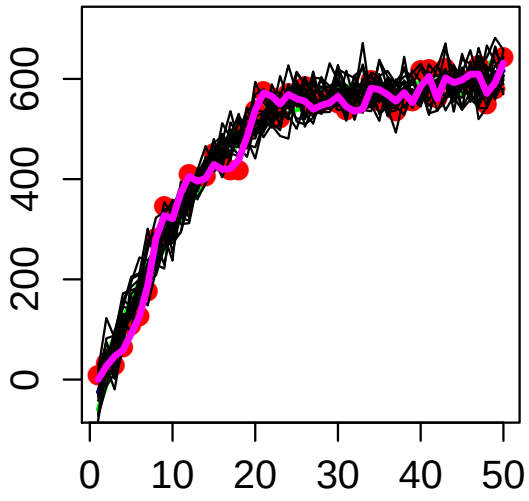
Grey: $X_3(t) = f_{\text{hill}-\text{or}}(X_1(t-1), X_2(t-1))$
Colored: GP with covariance trained on
$(X_1(t-1), X_2(t-1)) \rightarrow X_3(t)$

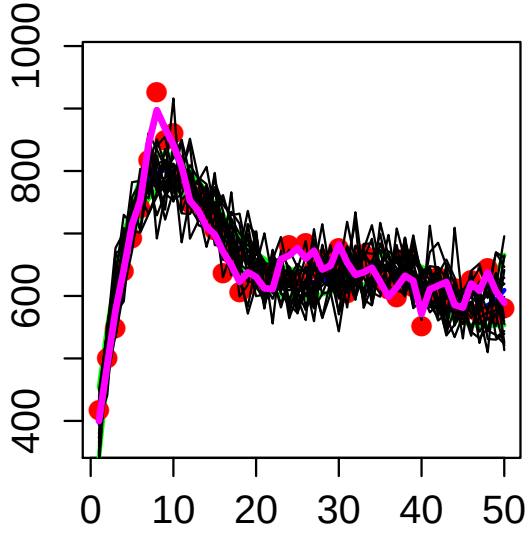
GP approximation



Accounting for uncertainty







Uncertainty:

- pseudoordering
- cell process noise
- measurement

Guess amount of process vs measurement noise

Reconstruct possible trajectories via **Gaussian process state-space model** (GP-SSM)

Latent state-space model

Given: mean m_g, covariance K_g for all gene trajectories

Aim: reconstruct transition function f and **latent trajectories** $\mathbf{x}^{(g)} = (\mathbf{x}_t^{(g)})$

$$f \sim \mathrm{GP}(0, K)$$
$$\mathbf{x}_t = f(\mathbf{x}_{t-1}) + \epsilon, \text{ for all } t$$
$$m_g \sim N(x^{(g)}, K_g), \text{ for all } g$$

Particle Gibbs

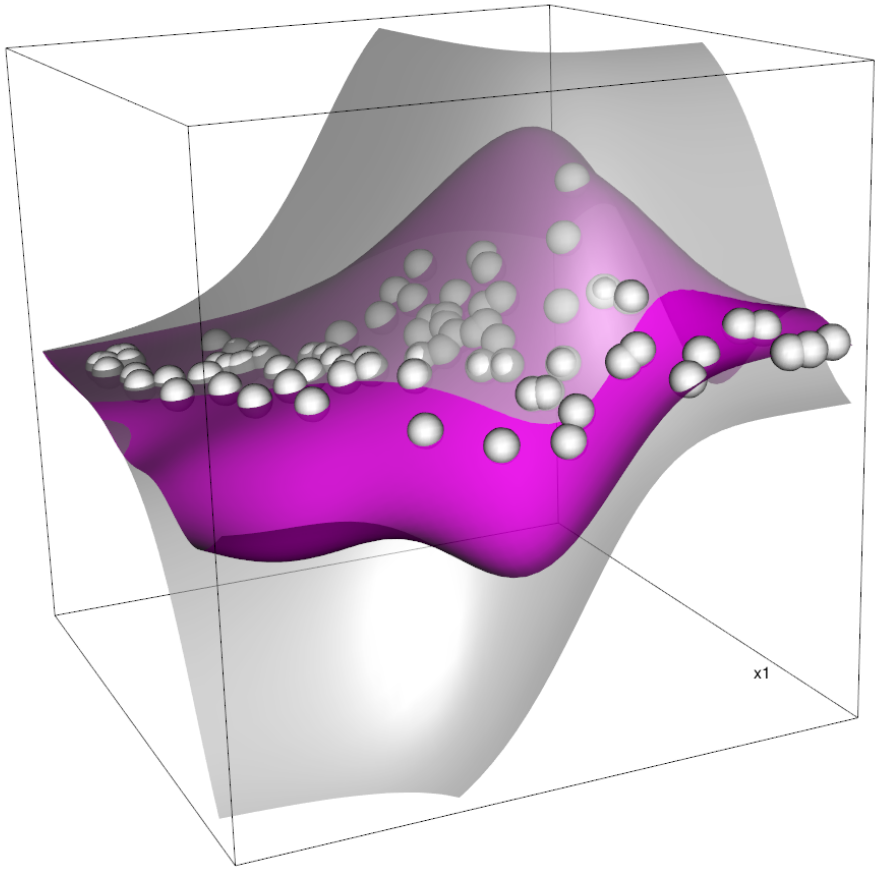
- Initialise latent trajectories $\mathbf{x} = (\mathbf{x}_g)$
- Loop:
  - Sample f (inducing points u) from $\mathbf{x}$
  - Sample new $\mathbf{x}_{\text{new}}$ from $\mathbf{x}$ using PGAS

Particle Gibbs Ancestor Sampler (PGAS) provides transition kernel $\mathbf{x} \rightarrow \mathbf{x}_{\text{new}}$ using **reference particle**

- N particles represent trajectories
- Loop $t = 1, \ldots, T$
  - Sample $N - 1$ from previous particles using GP
  - Sample reference particle N using $\mathbf{x}$
  - Reweight according to likelihood

Roger Frigola, Time Series Learning, thesis 2015

Estimated inducing points



GP estimate through posterior mean of inducing points u from PGAS sampler

Conclusions

- **Process noise** crucial for **causal inference** from observations
- **Plenty of useful variation** in **snapshot** single cell data: use regression (SEM) for short-term dynamics
- **Variation lost** in **pseudotime** data for long-term dynamics: difficult to regain with latent state-space models

People

MRC Biostatistics Unit

John Reid
Magdalena Strauss
Paul Kirk
Petras Verbyla

Gurdon Institute

Julia Tischler
Chris Penfold