Modelling gene expression dynamics with Gaussian process inference

Alexis Boukouvalas
PROWLER.io
September 25^{th} 2019

Talk Outline

Part 1. Introduction to Gaussian process regression

- From a multivariate Gaussian to a Gaussian process
- Covariance functions
- Gaussian processes for inference: Bayesian Regression

Part 2. Hierarchical models: batches and clusters

- Modelling time-series batches
- Combined modelling of cluster and batch variation

Part 3. Branching Gaussian processes

- Modelling branching time course data with labels
- Modelling branching without labels: single-cell data

Part 4. Dimensionality reduction and pseudotime inference

- GPLVM for pseudotime inference with capture times
- New extension of GPLVM: pseudotime with branching

Part 1. Introduction to Gaussian process regression

Probability distributions over functions

$$f(t) \sim \mathcal{GP}\left(\text{mean}(t), \text{cov}(t, t')\right)$$

Covariance function $k = \text{cov}(t, t')$ defines typical properties,

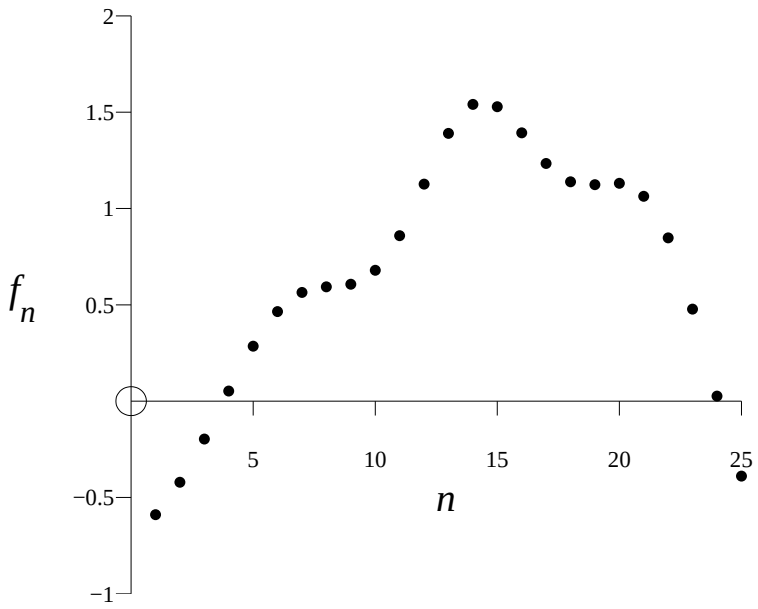
- Static . . . Dynamic
- Smooth . . . Rough
- Stationary. . . non-Stationary
- Periodic. . . Chaotic

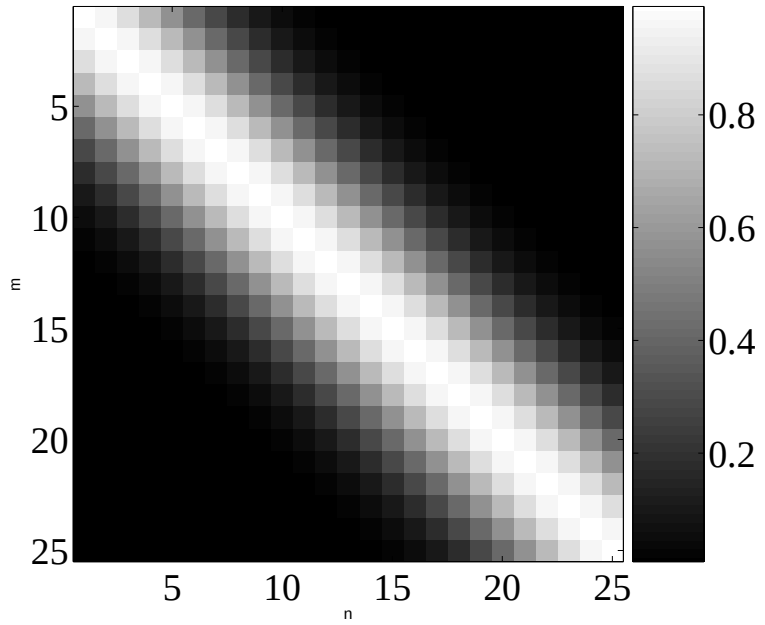
The covariance function has parameters tuning these properties

Bayesian Machine Learning perspective: Rasmussen & Williams "Gaussian Processes for Machine Learning" (MIT Press, 2006)

From a multivariate Gaussian to a Gaussian process

Samples from a 25-dimensional multivariate Gaussian distribution:



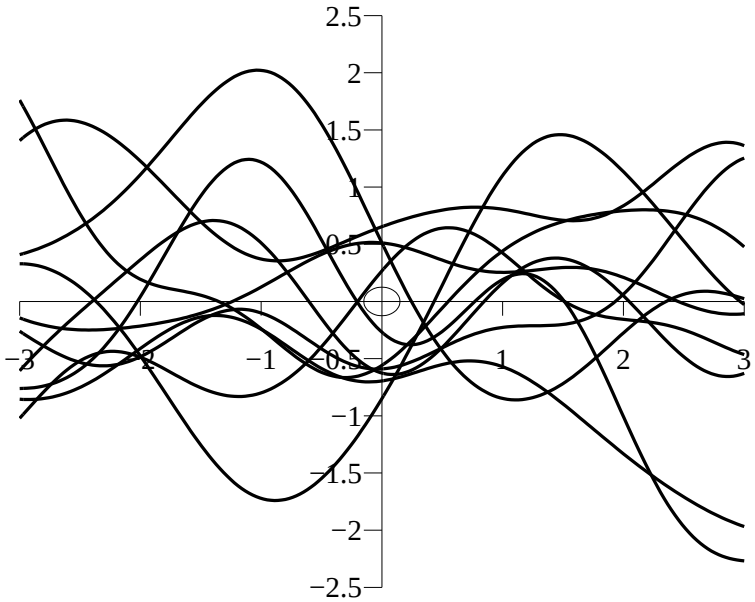


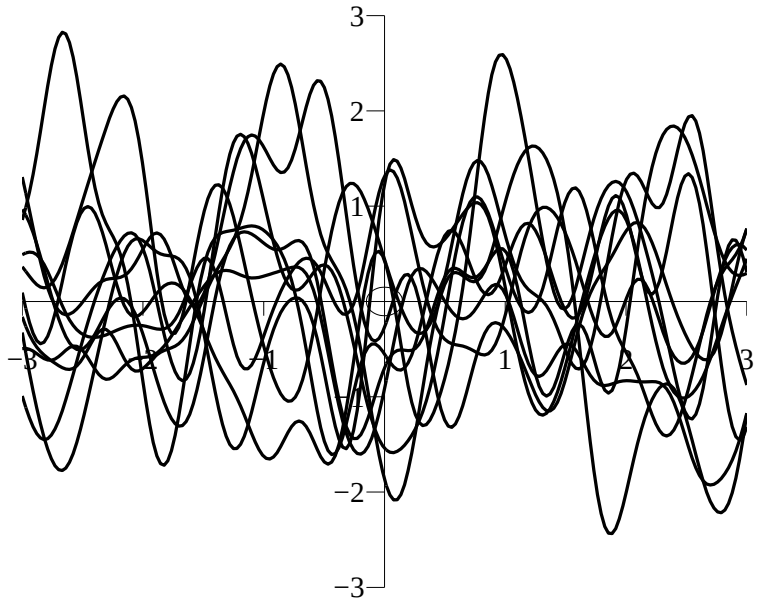
$$[f_1, f_2, \ldots, f_{25}] \sim \mathcal{N}(0, C)$$

Learning and Inference in Computational Systems Biology, MIT Press

From a multivariate Gaussian to a Gaussian processes

Take dimension $\rightarrow \infty$

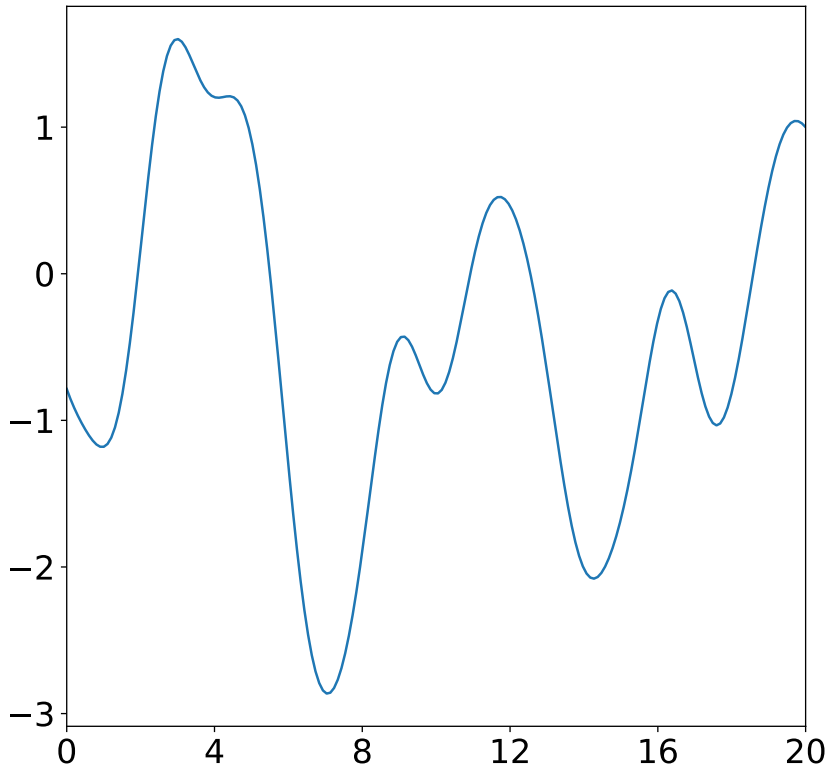




$$f \sim \mathcal{GP}(0, k) \qquad k\left(t, t'\right) = \alpha \exp\left(-\frac{(t - t')^2}{l^2}\right)$$

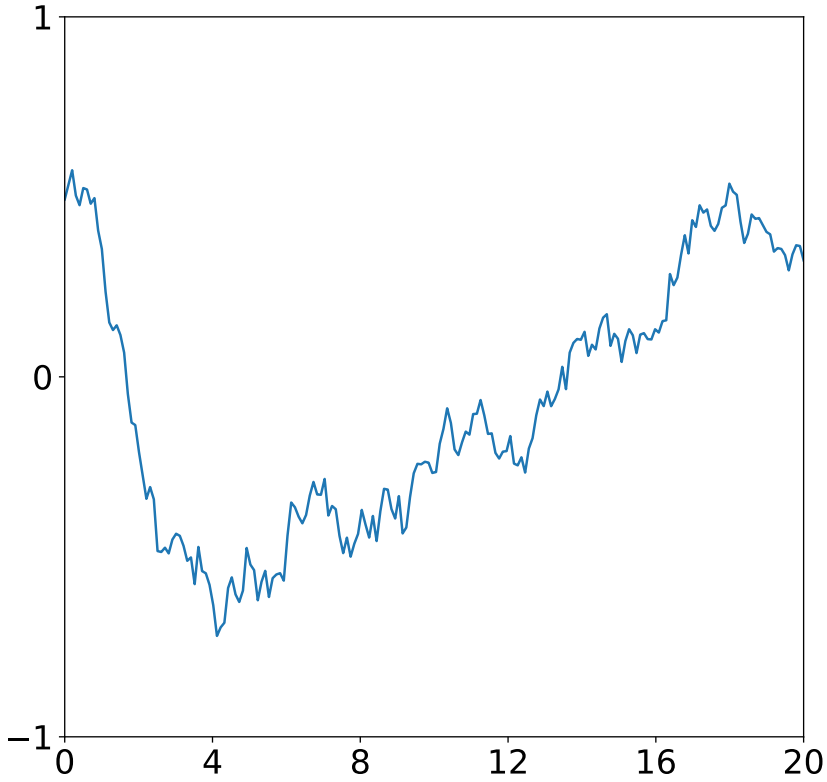
Learning and Inference in Computational Systems Biology, MIT Press

Covariance functions - Squared Exponential (aka RBF)



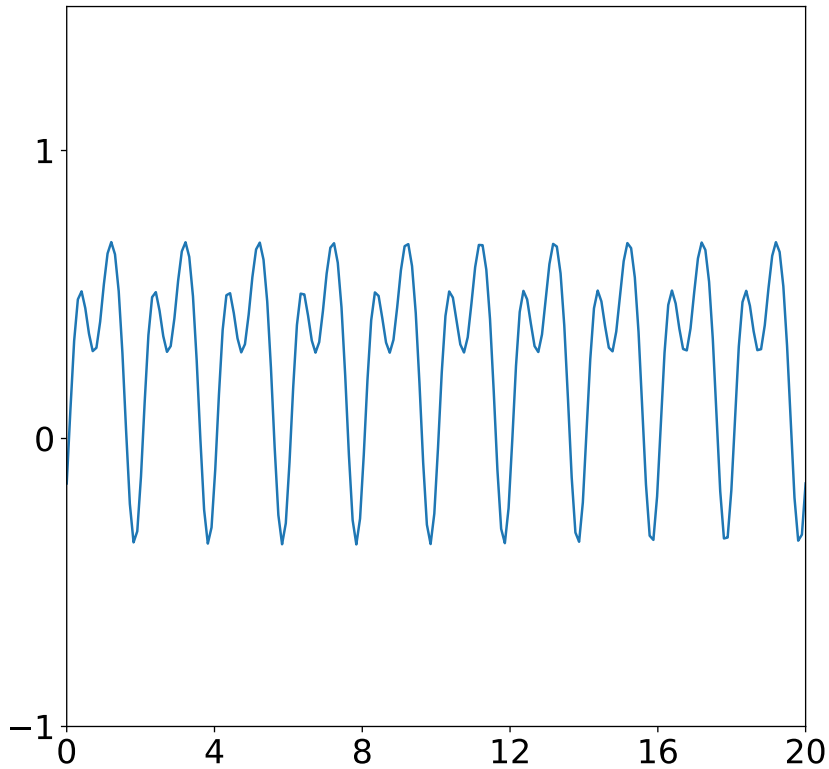
$$k\left(t, t'\right) = \alpha \exp\left(-\frac{\left(t - t'\right)^2}{l^2}\right)$$

Covariance functions - Ornstein Uhlenbeck (OU process)



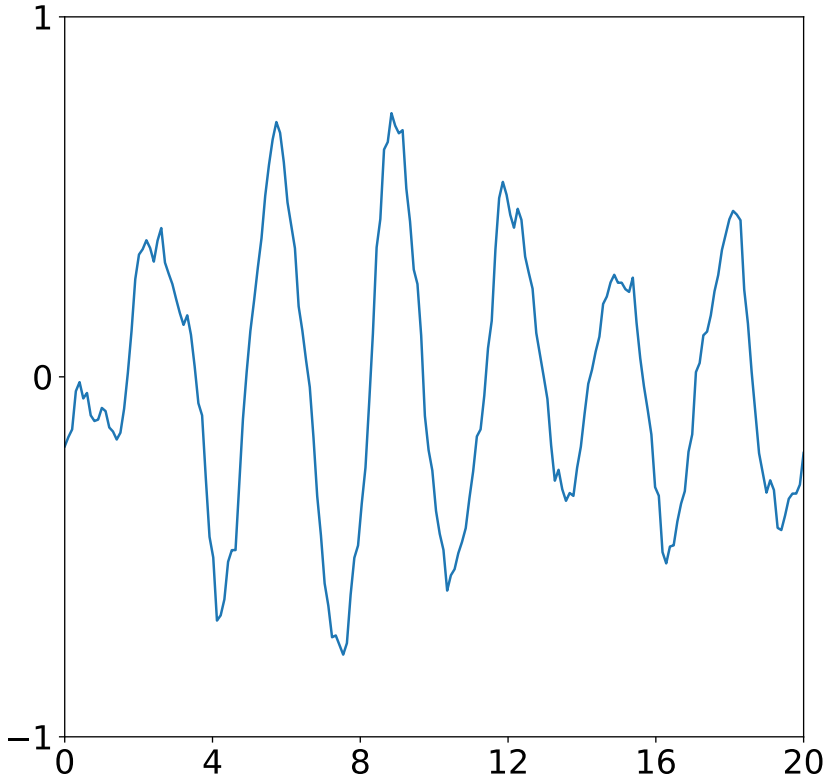
$$k\left(t, t'\right) = \alpha \exp\left(-\frac{\left|t - t'\right|}{l}\right)$$

Covariance functions - Periodic smooth process



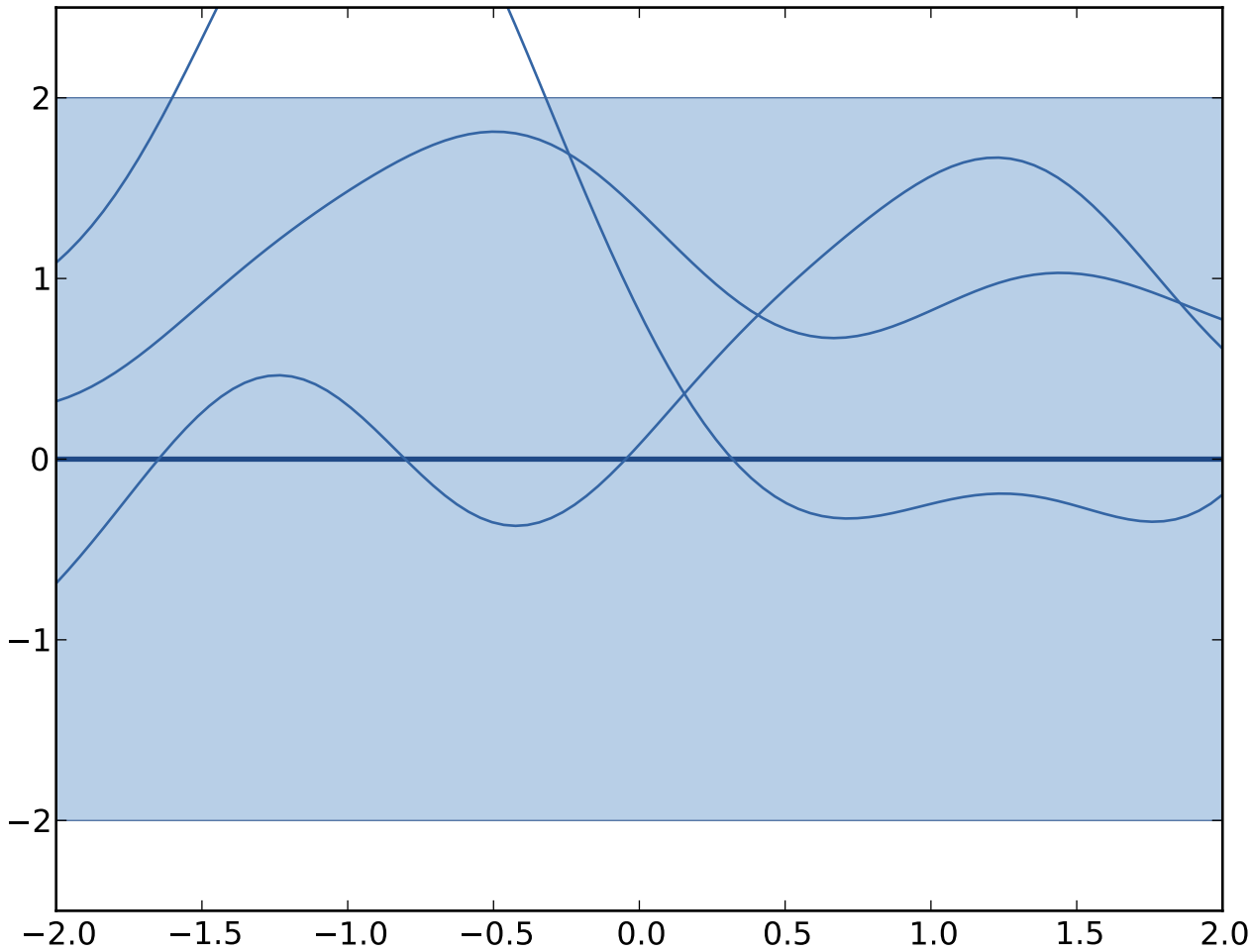
$$k\left(t, t'\right) = \alpha \exp\left(-\frac{sin^2\left(\pi \frac{t-t'}{\lambda}\right)}{l^2}\right)$$

Covariance functions - Quasi-periodic OU process

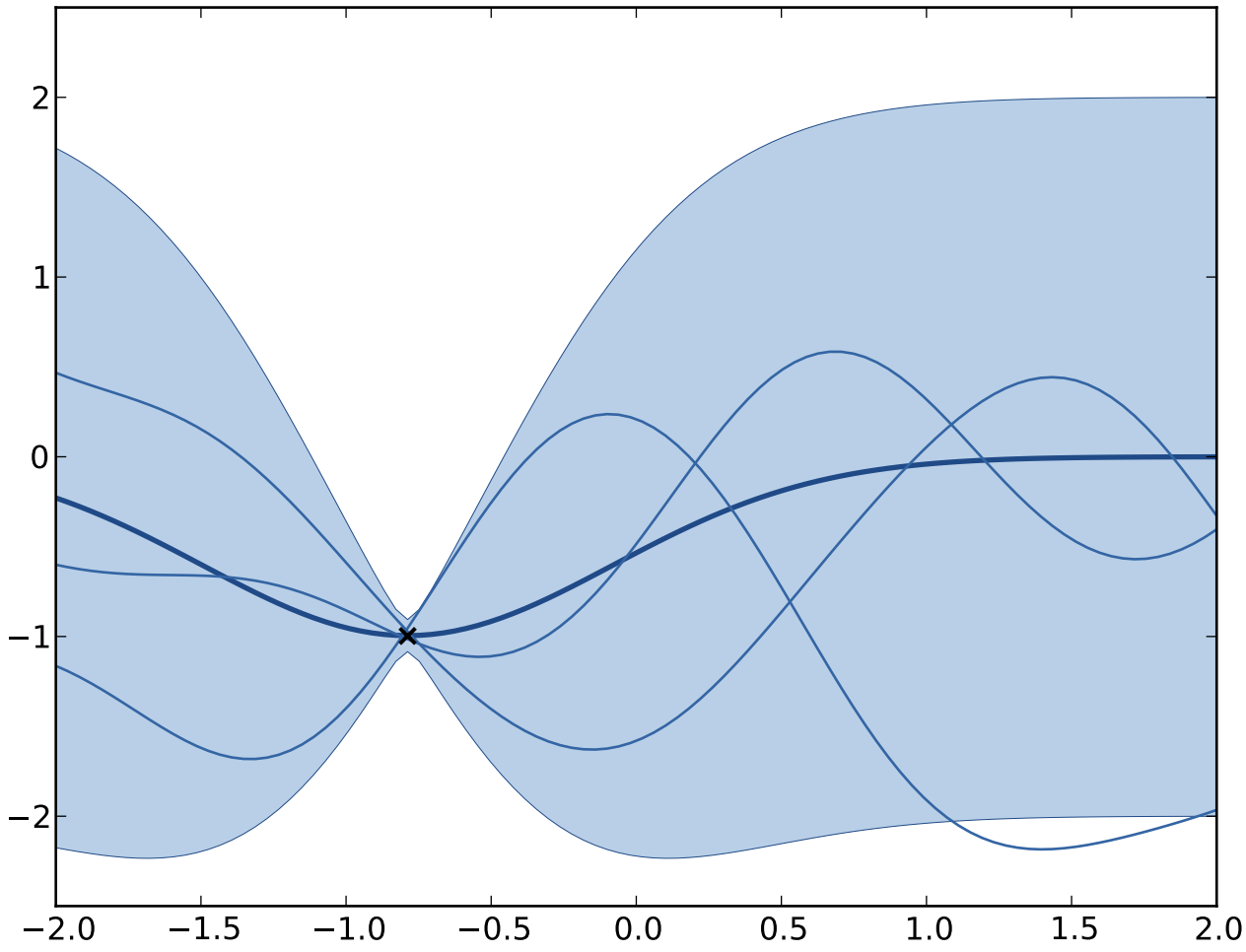


$$k\left(t, t'\right) = \alpha \exp\left(-\frac{|t - t'|}{l}\right) cos(\beta|t - t'|)$$

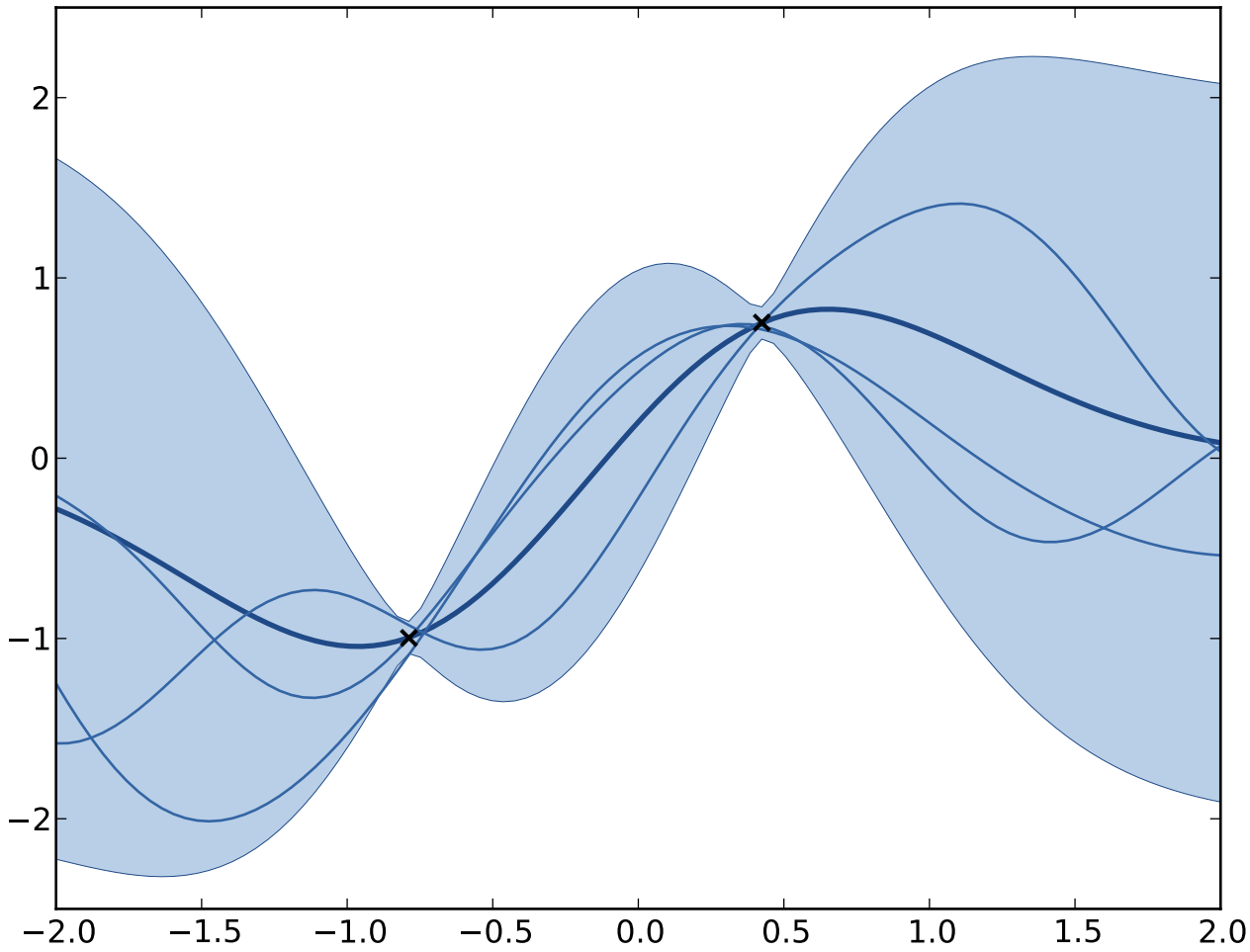
Gaussian processes for inference: Bayesian Regression



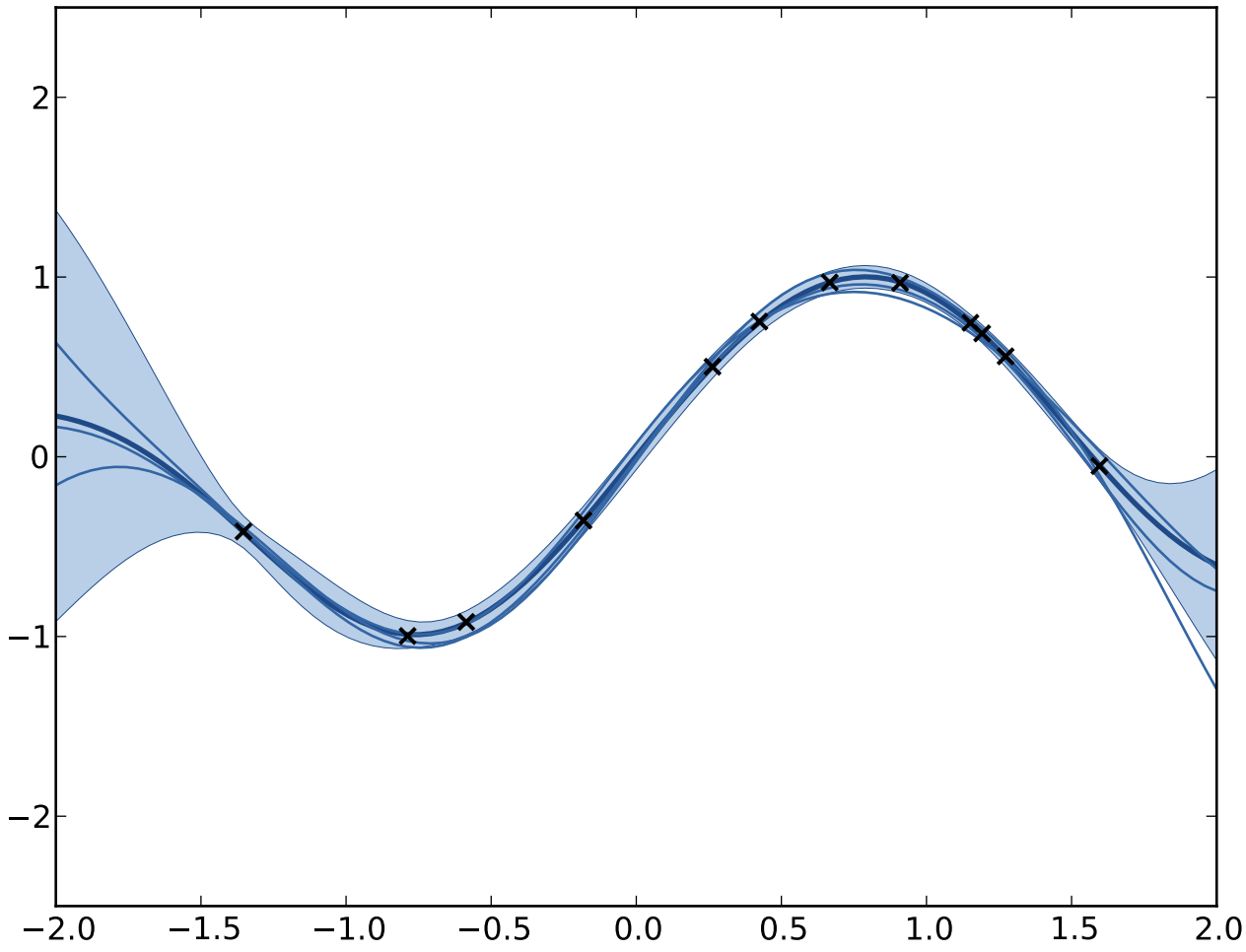
Bayesian Regression



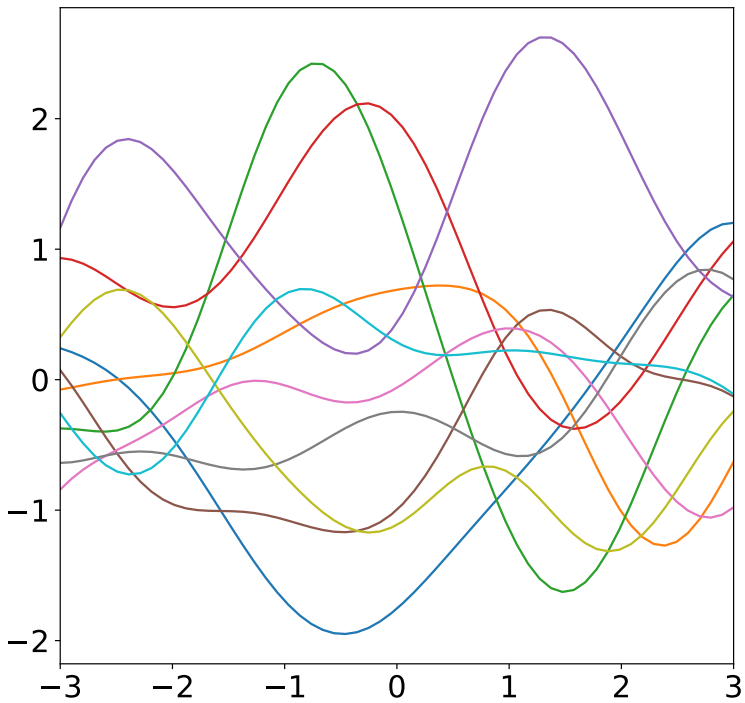
Bayesian Regression

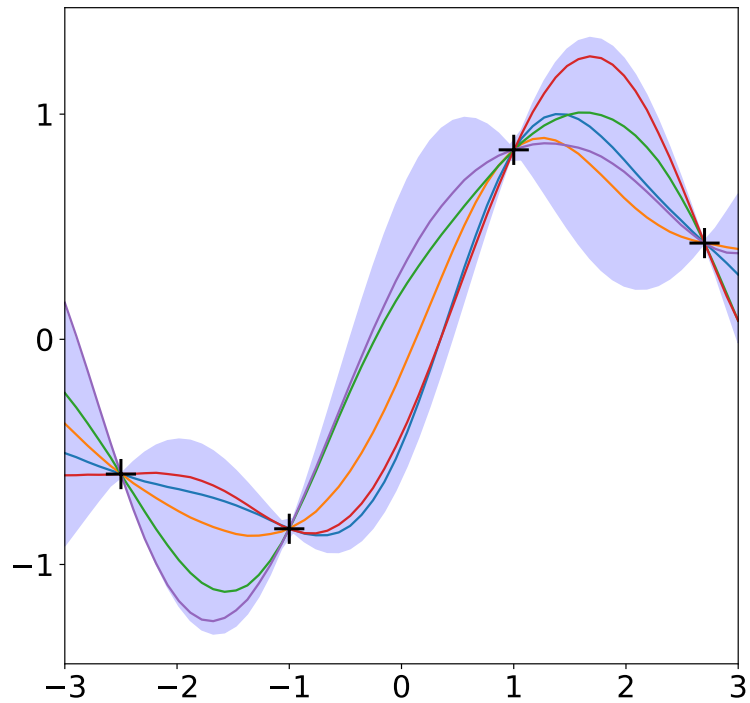


Bayesian Regression



Bayesian Regression





Posterior distribution (right) captures all functions consistent with the prior (left) that pass close to the data

Hyper-parameter learning

We can calculate the model likelihood exactly,

$$\begin{aligned} p(\mathbf{Y}|\mathbf{X}) &= \int \mathcal{N}(\mathbf{Y}|\mathbf{f}, \sigma^2\mathbf{I})\,\mathcal{N}(\mathbf{f}|\mathbf{0}, K(\mathbf{X},\mathbf{X}))\mathrm{d}\mathbf{f} \\ &= \mathcal{N}(\mathbf{Y}|\mathbf{0}, K(\mathbf{X},\mathbf{X}) + \sigma^2\mathbf{I}) \ . \end{aligned}$$

This allows estimation of the kernel hyper-parameters by numerically maximising the likelihood, or using Bayesian MCMC.

In today's labs we'll use a maximum likelihood approach.

Some practical considerations

Naive Gaussian process inference is slow

- Many datapoints - covariance inversion scales as $O(N^3)$

  Solution: **sparse inference** with k inducing points is $O(k^2 N)$

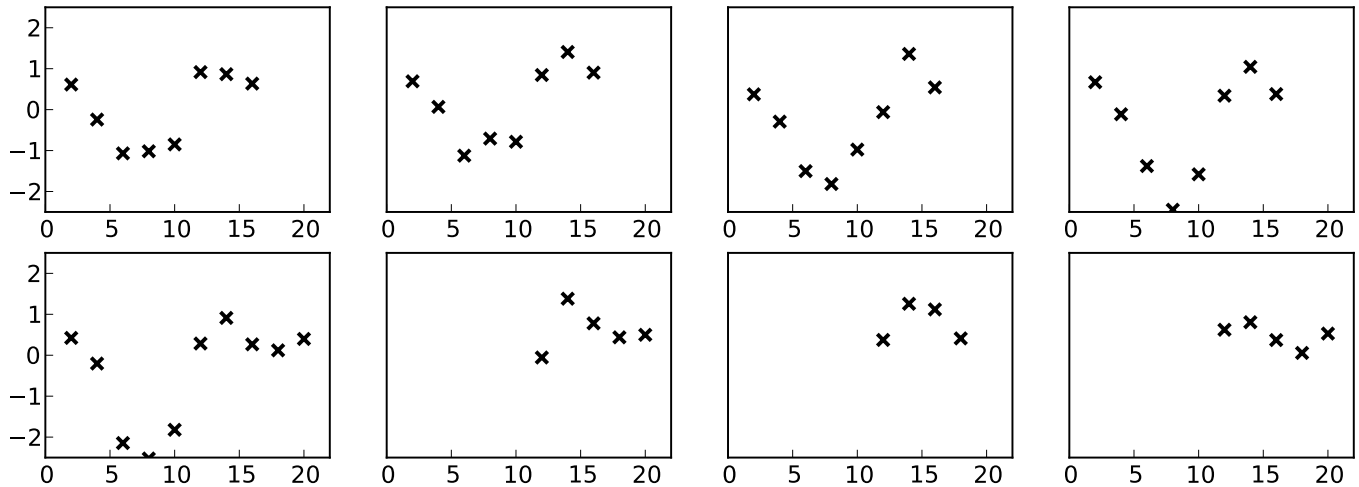
- Models with latent variables, e.g. GPLVM/branching models

  Solution: **variational inference**

- Computing derivatives can be time-consuming

  Solution: **GPflow package** uses TensorFlow autodiff

Part 2. Hierarchical models: batches and clusters

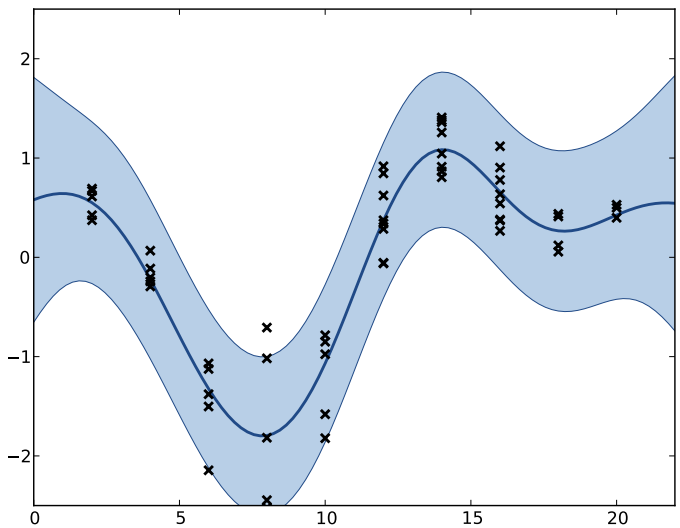


Data from Kalinka et al. "Gene expression divergence recapitulates the developmental hourglass model" *Nature* 2010

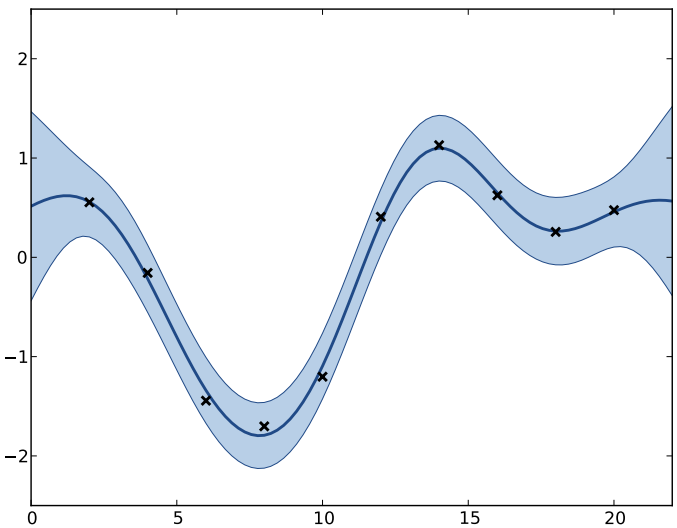
Joint work with James Hensman and Neil Lawrence

Naive processing options for time course batches

Lumped



Averaged



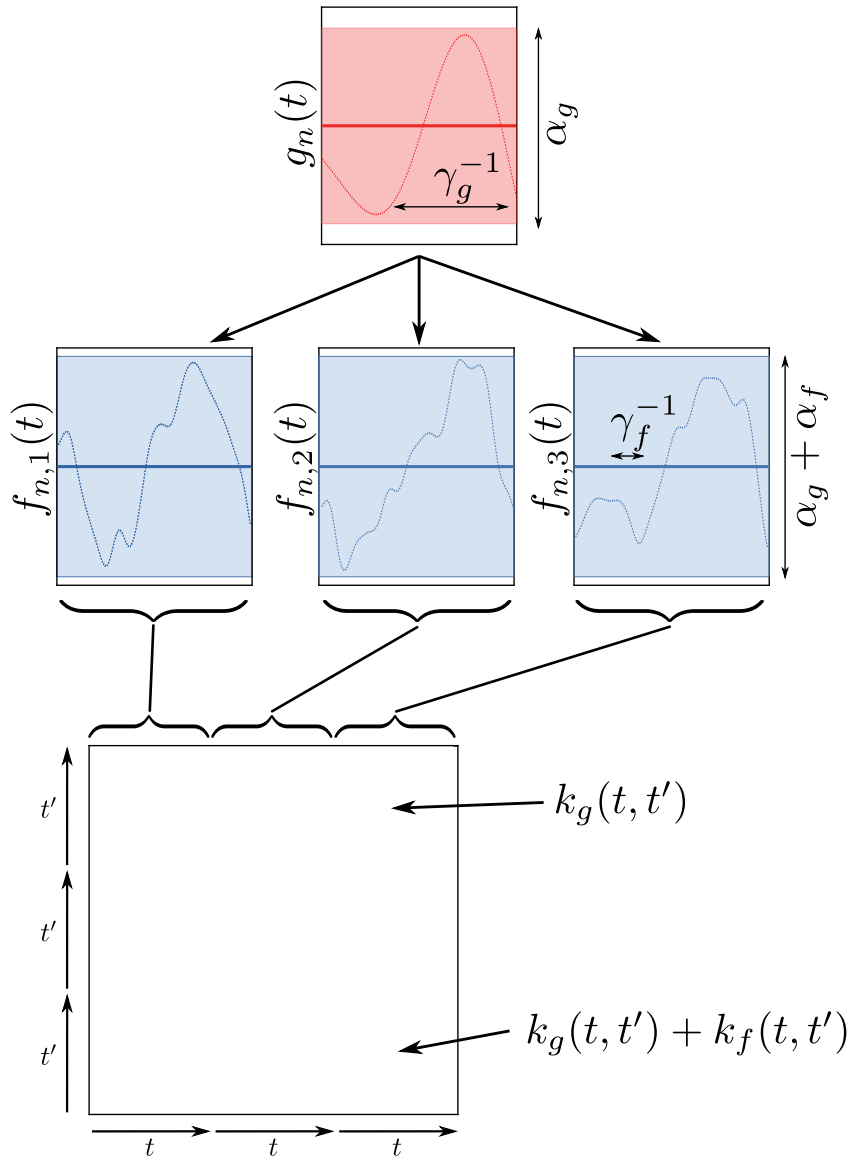
Hierarchical Gaussian process

gene:

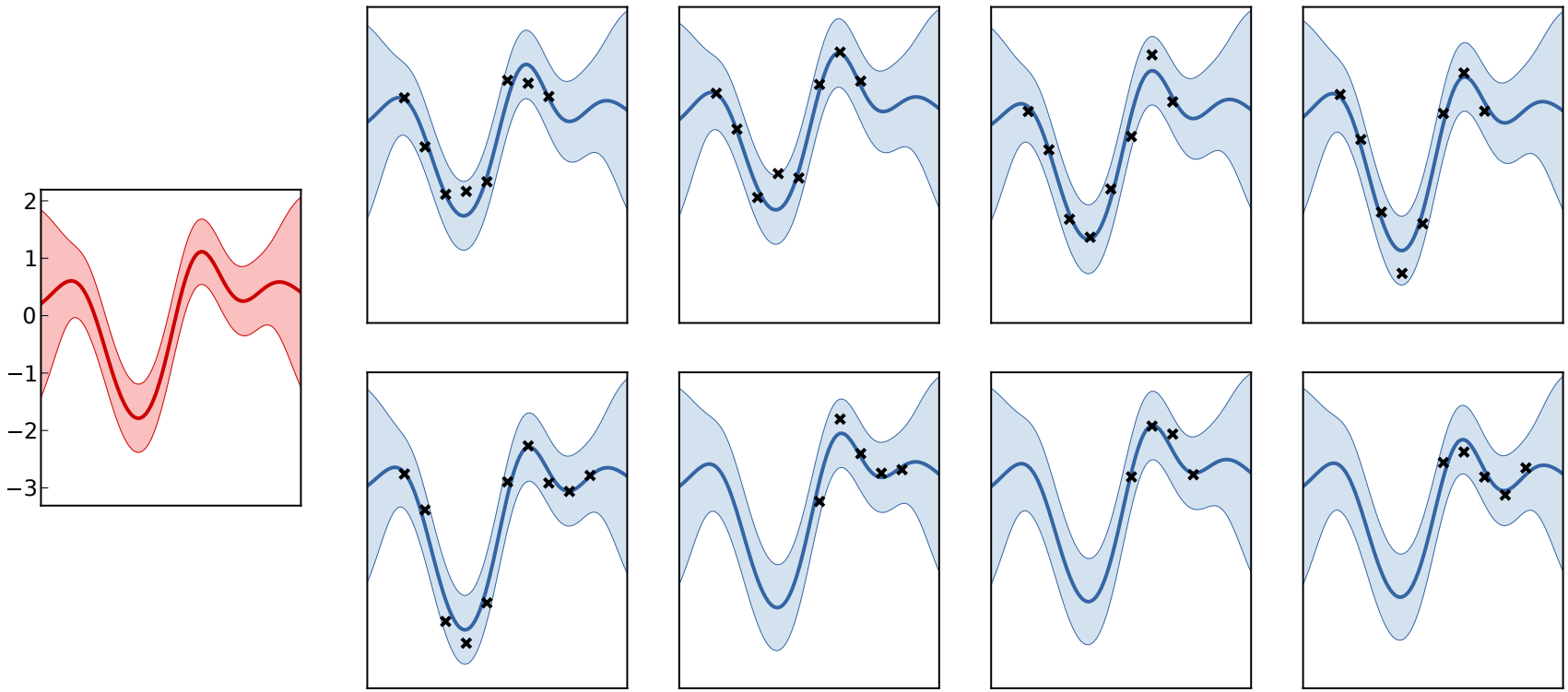
$$g(t) \sim \mathcal{GP}\left(0, k_g(t, t')\right)$$

replicate:

$$f_i(t) \sim \mathcal{GP}\left(g(t), k_f(t, t')\right)$$



Hierarchical Gaussian process



J. Hensman, N.D. Lawrence, M.Rattray "Hierarchical Bayesian modelling of gene expression time series across irregularly sampled replicates and clusters" *BMC Bioinformatics* 2013

Hierarchical Gaussian process for clustering

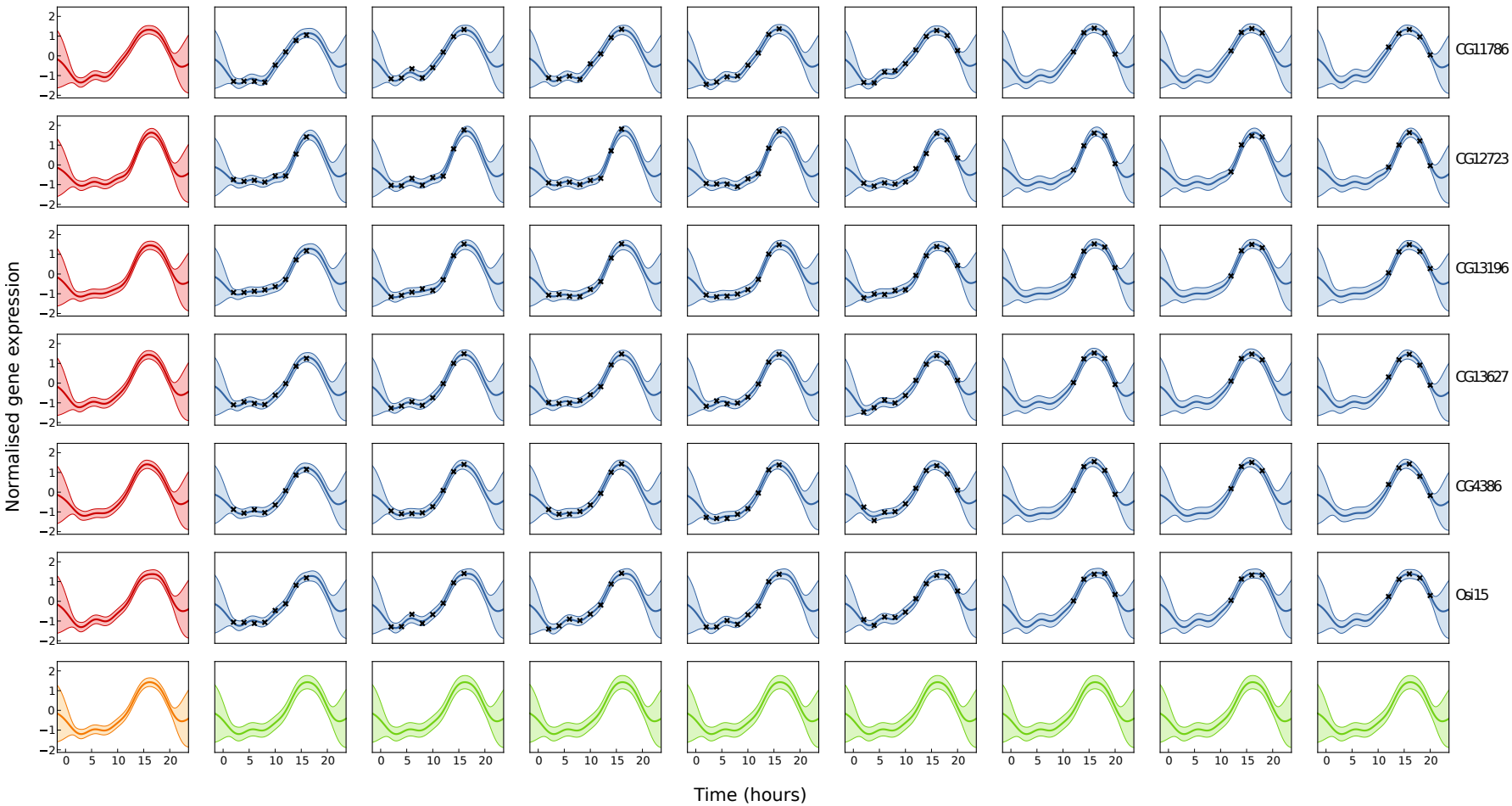
An extended hierarchy

$$h(t) \sim \mathcal{GP}\Big(0, k_h(t, t')\Big) \text{ cluster}$$

$$g_i(t) \sim \mathcal{GP}\Big(h(t), k_g(t, t')\Big) \text{ gene}$$

$$f_{ir}(t) \sim \mathcal{GP}\Big(g_i(t), k_f(t, t')\Big) \text{ replicate}$$

Hierarchical Gaussian process for clustering



Hierarchical Gaussian process for clustering

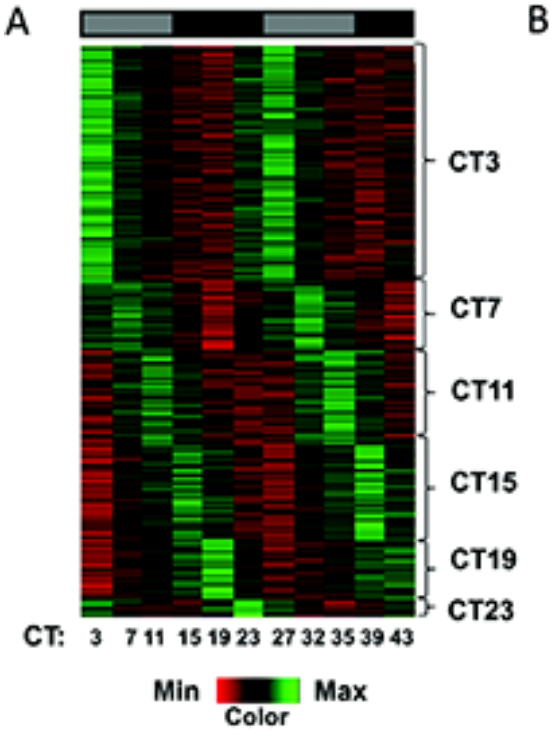
Modifying an existing algorithm to include this model of replicate and cluster structure leads to more meaningful clustering

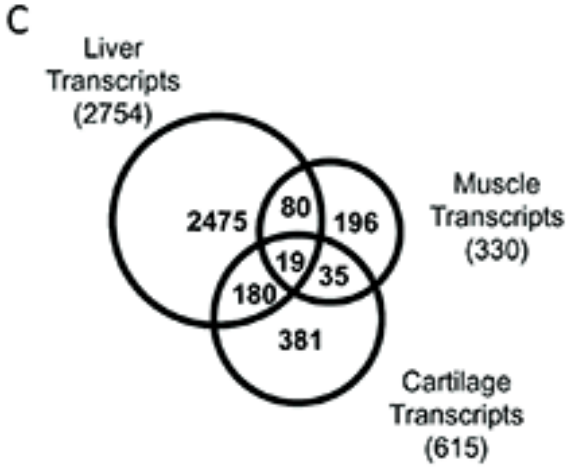
	MF	BP	CC	$\mathcal{L}$	N. clust.
agglomerative HGP	**0.46**	**0.16**	**0.50**	**7360.8**	50
agglomerative GP	0.39	0.13	0.36	6203.7	128
Mclust (concat.)	0.39	0.07	0.25	1324.0	26
Mclust (averaged)	0.40	0.08	0.24	-736.2	20

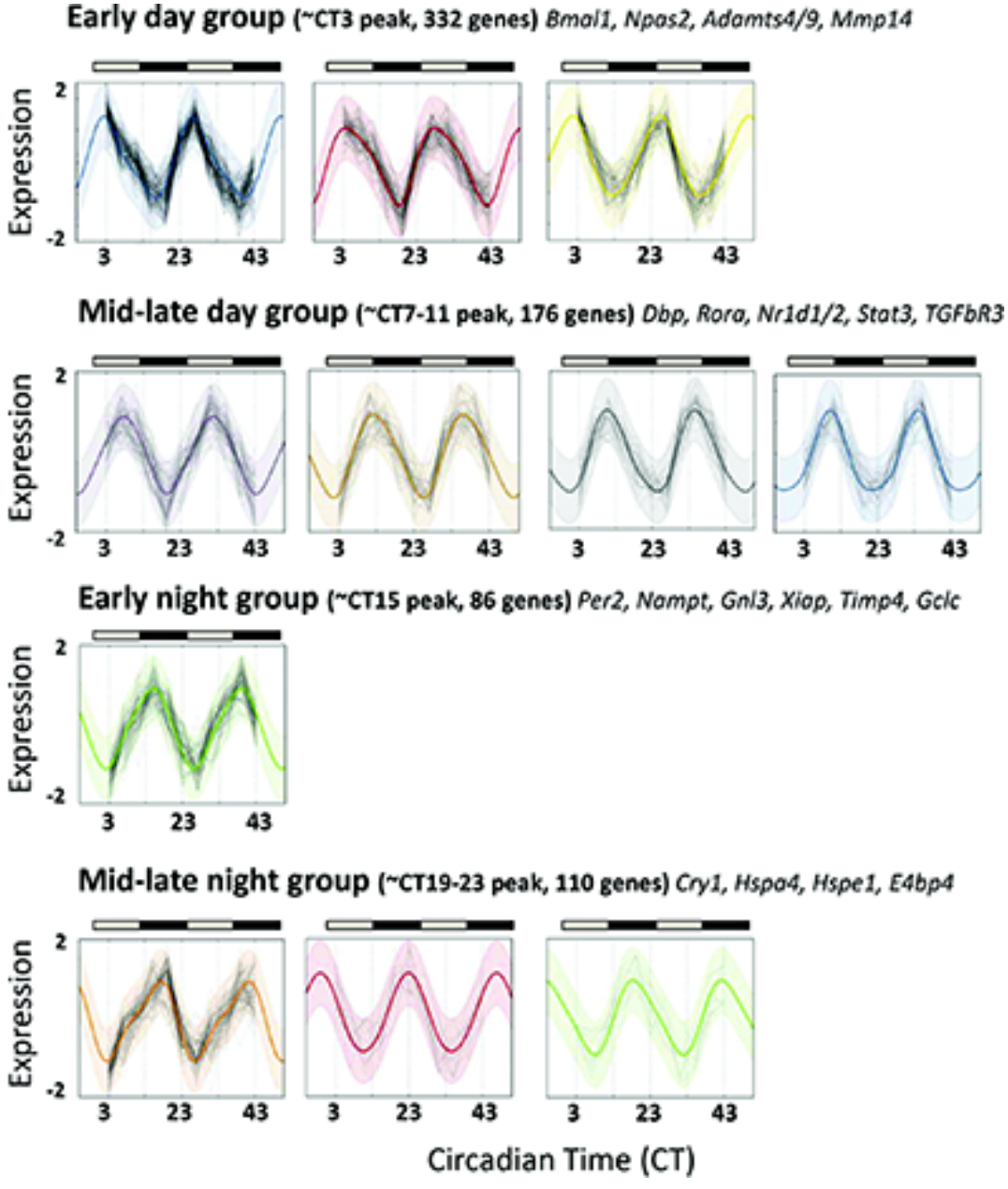
Variational Bayes algorithm is more efficient, allowing Bayesian clustering of $>$10K profiles with a Dirichlet Process prior

J. Hensman, M.Rattray, N.D. Lawrence "Fast non-parametric clustering of time-series data" *IEEE TPAMI* 2015

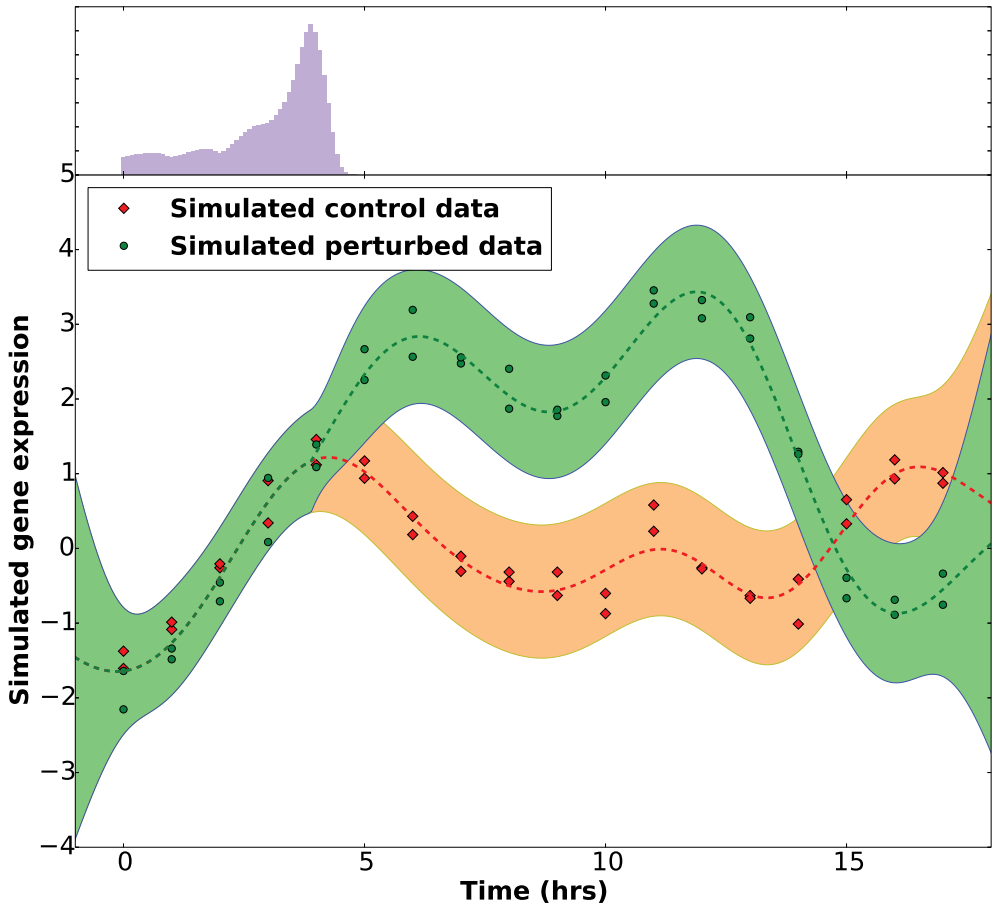
Clustering with a periodic covariance





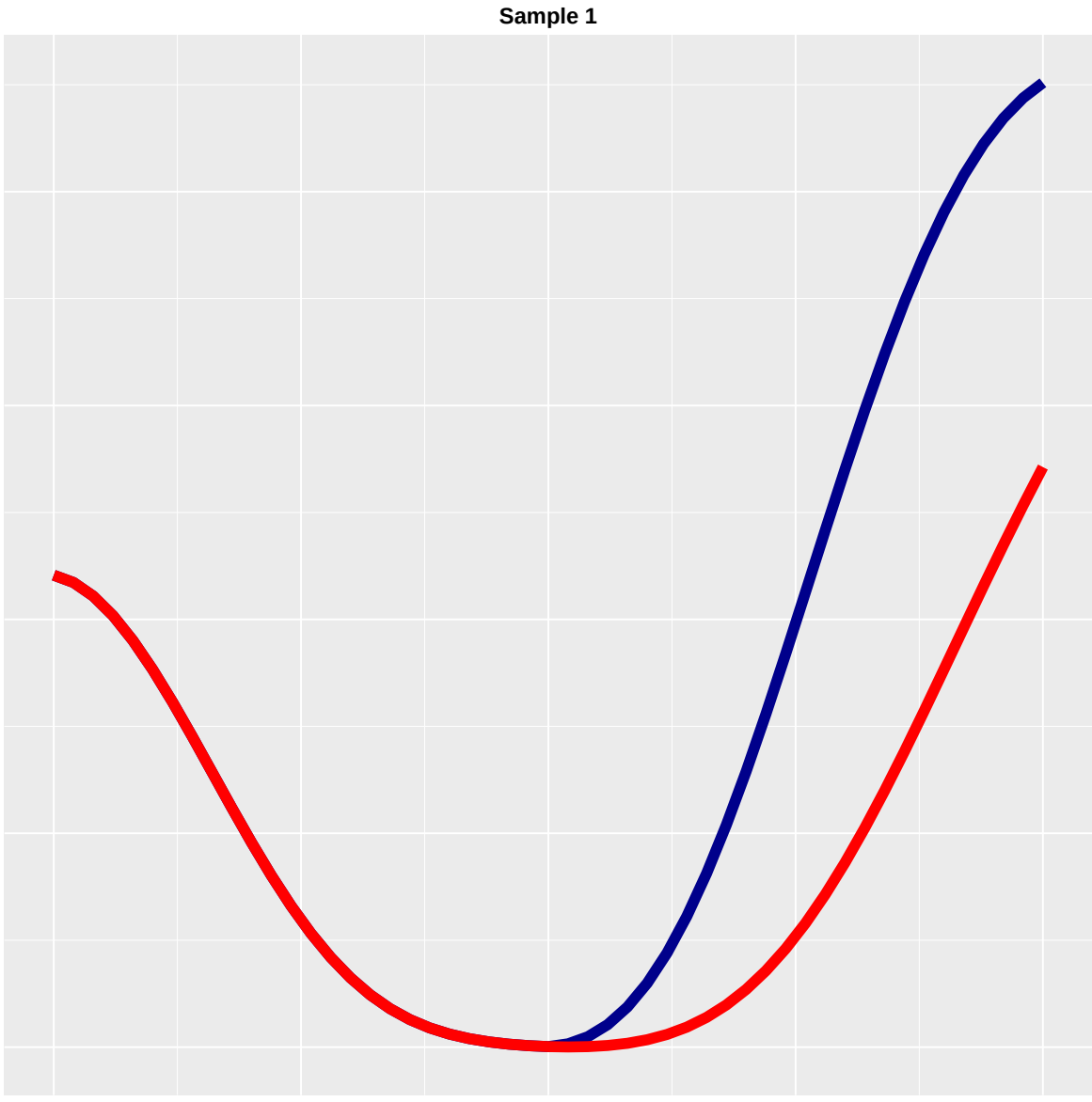


Part 3. Branching Gaussian processes

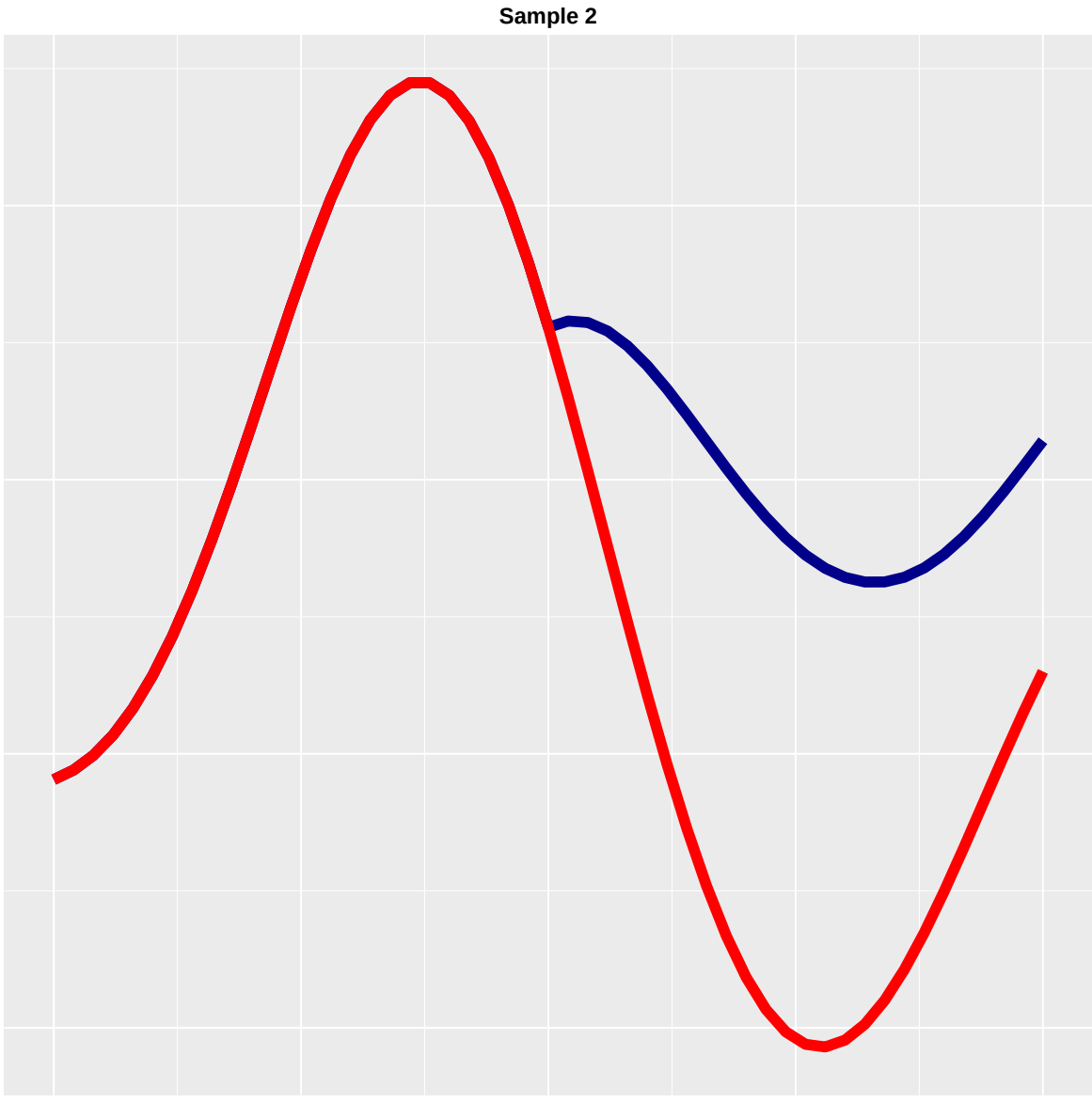


work with Jing Yang, Chris Penfold and Murray Grant

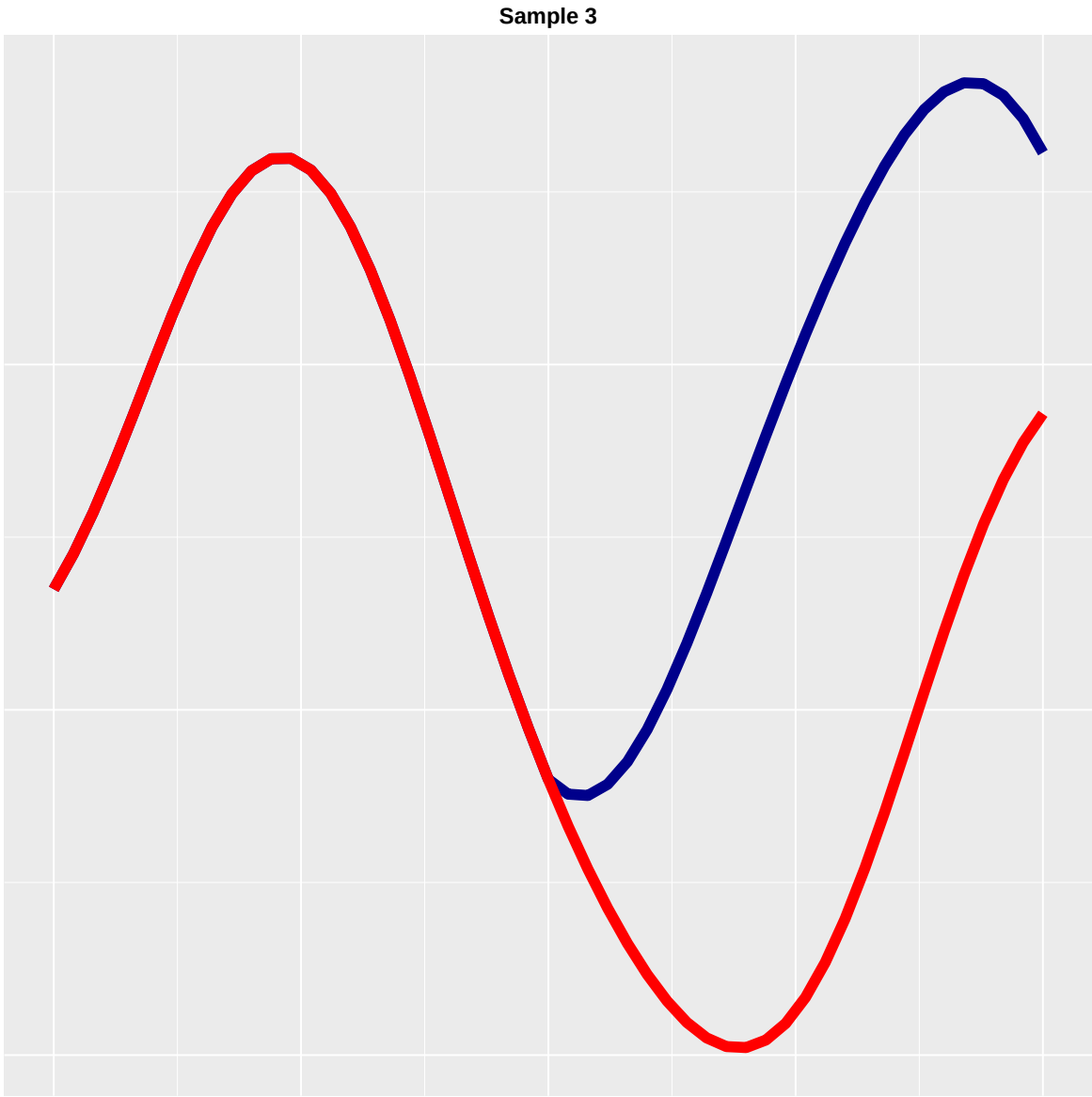
Samples from a branching model



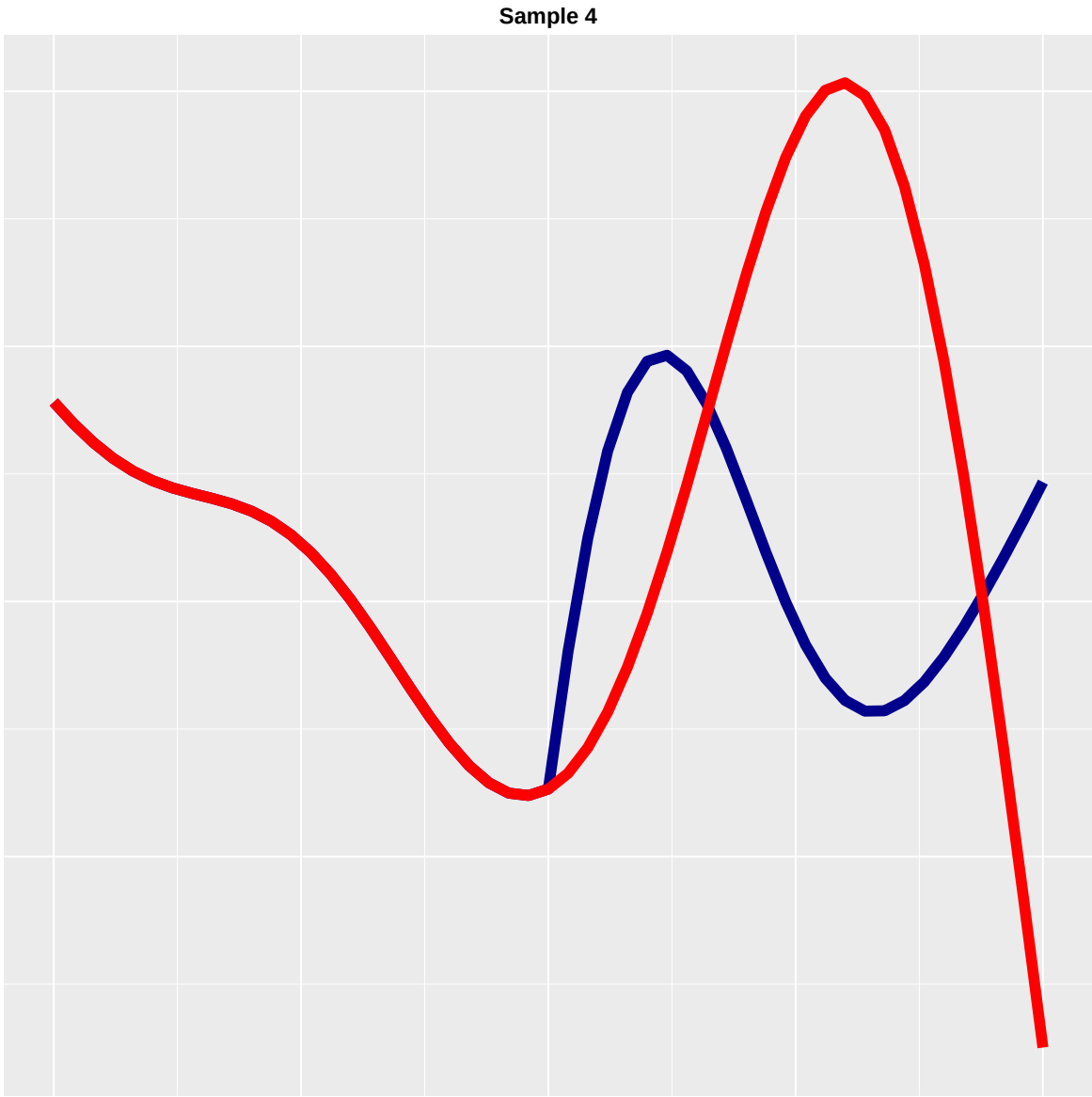
Samples from a branching model



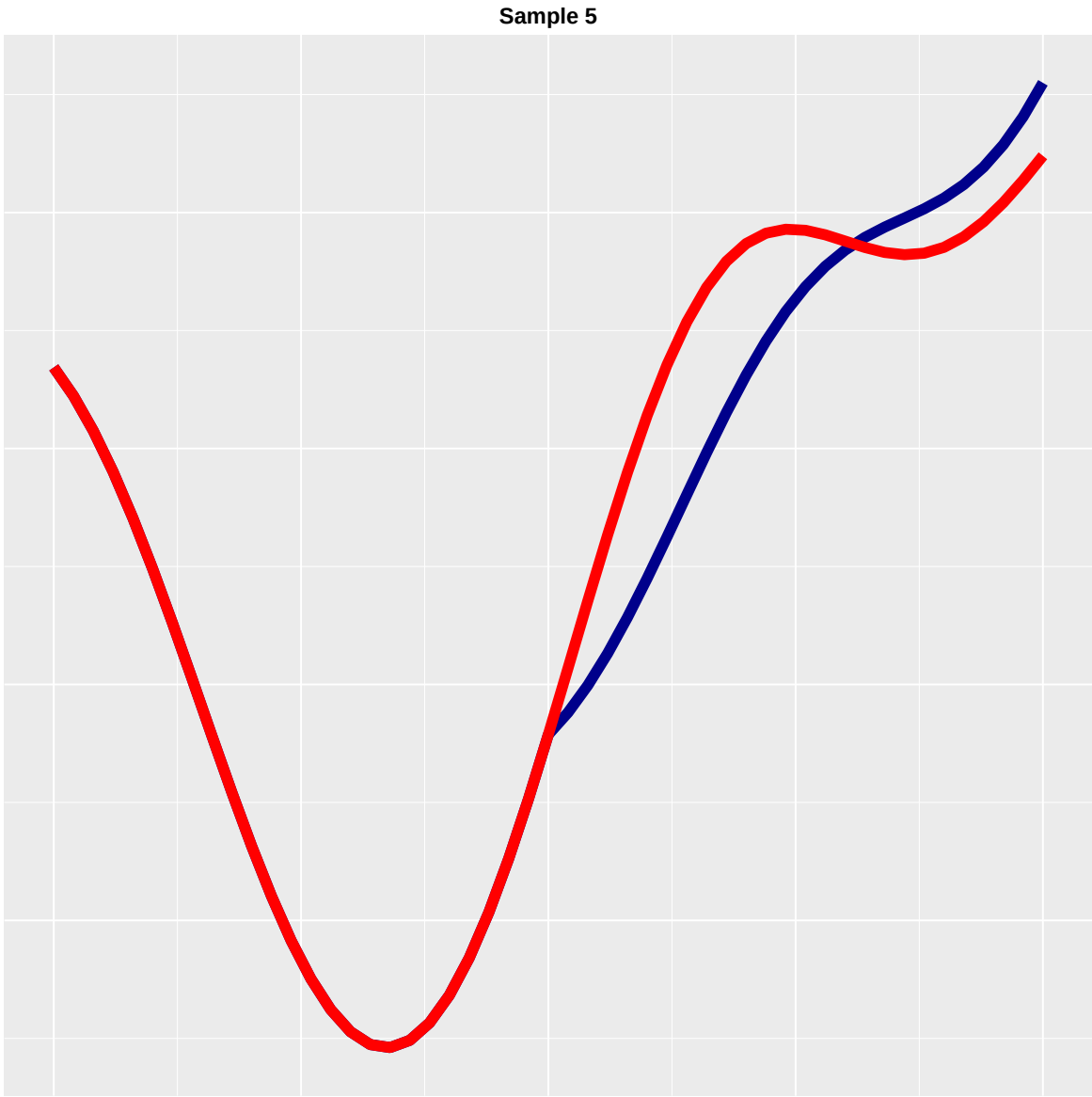
Samples from a branching model



Samples from a branching model

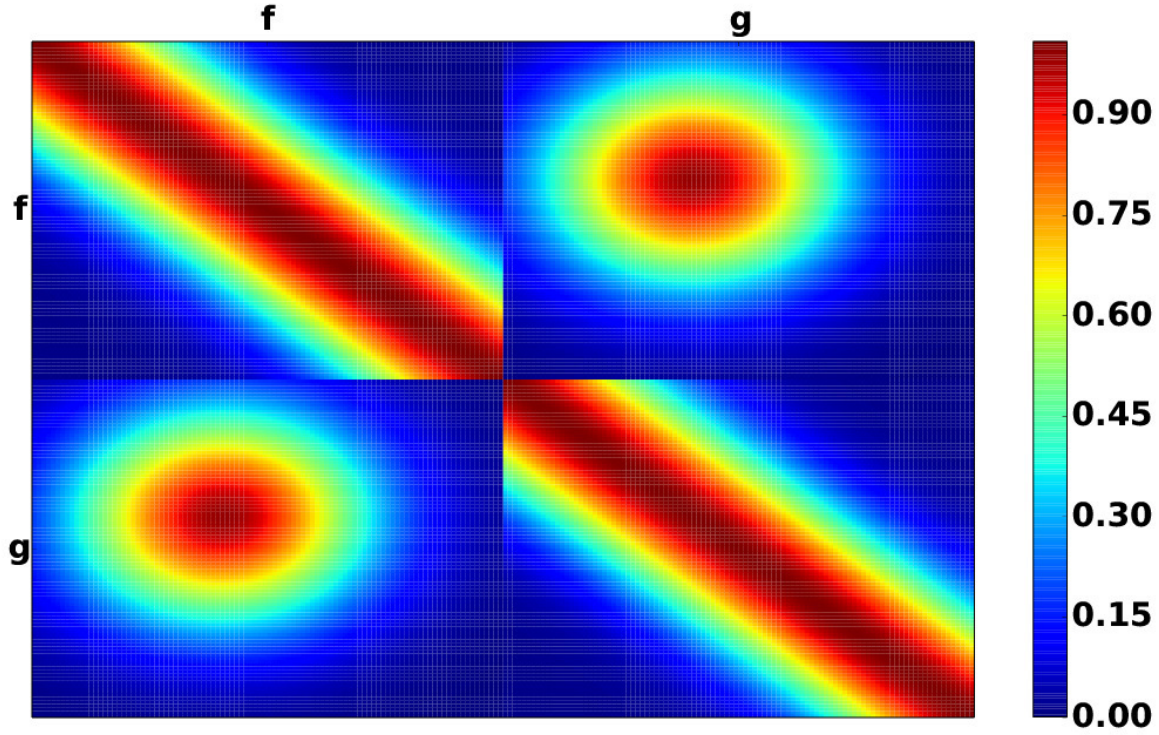


Samples from a branching model



Joint distribution to two functions crossing at t_p

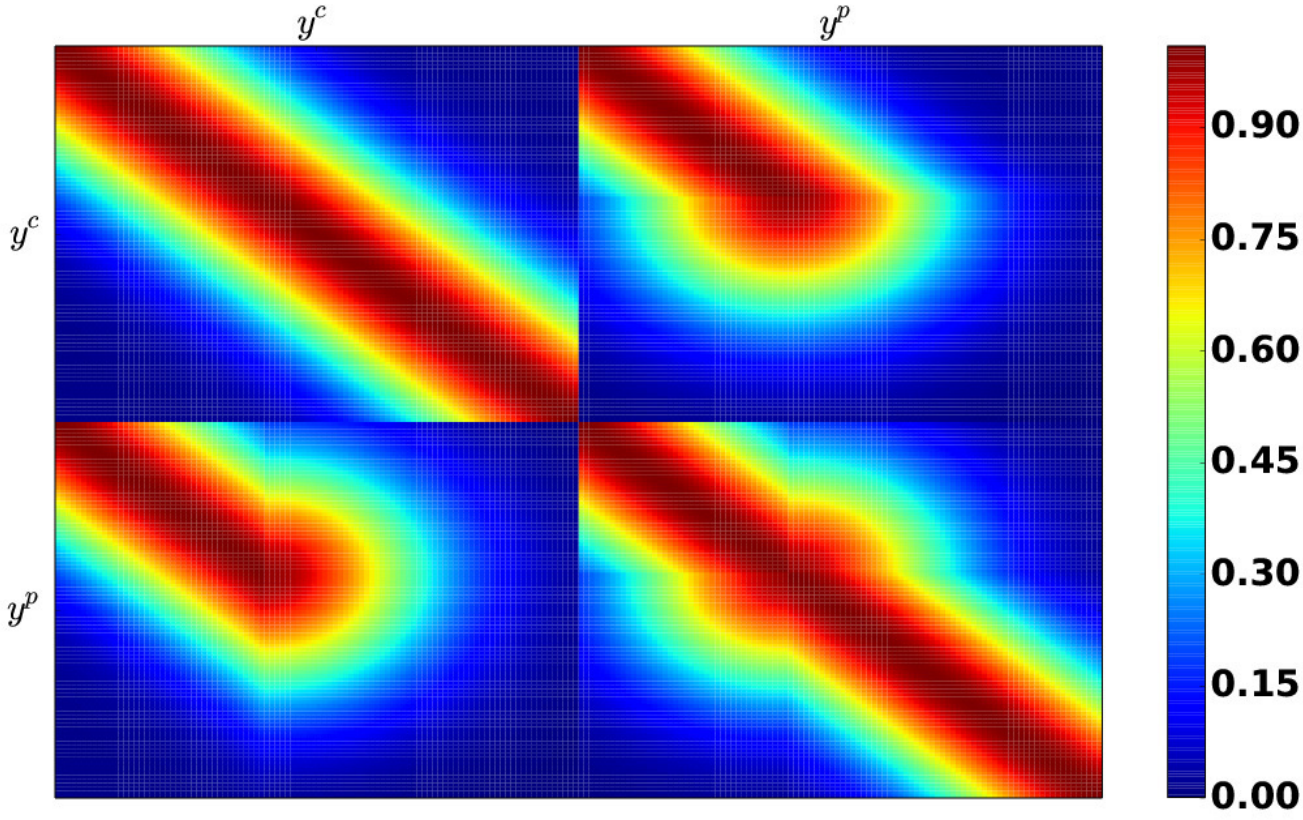
$$f \sim \mathcal{GP}(0, K) \ , \quad g \sim \mathcal{GP}(0, K) \ , \quad g(t_p) = f(t_p)$$



$$\Sigma = \begin{pmatrix} K_{ff} & K_{fg} \\ K_{gf} & K_{gg} \end{pmatrix} = \begin{pmatrix} K(\mathbf{T}, \mathbf{T}) & \frac{K(\mathbf{T}, t_p) K(t_p, \mathbf{T})}{k(t_p, t_p)} \\ \frac{K(\mathbf{T}, t_p) K(t_p, \mathbf{T})}{k(t_p, t_p)} & K(\mathbf{T}, \mathbf{T}) \end{pmatrix}$$

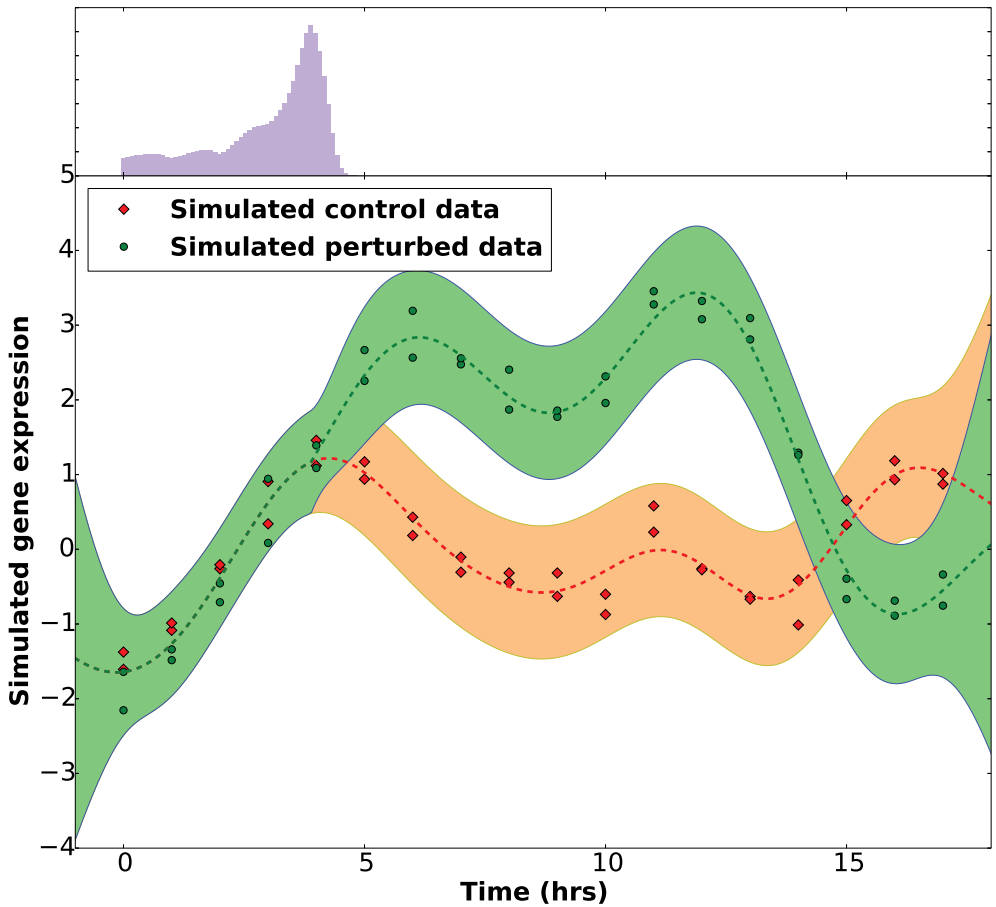
Joint distribution of two datasets diverging at t_p

$$
\begin{aligned}
y^c(t_n) &\sim \mathcal{N}(f(t_n), \sigma^2) \\
y^p(t_n) &\sim \mathcal{N}(f(t_n), \sigma^2) \quad \text{for } t_n \leq t_p \\
y^p(t_n) &\sim \mathcal{N}(g(t_n), \sigma^2) \quad \text{for } t_n > t_p
\end{aligned}
$$



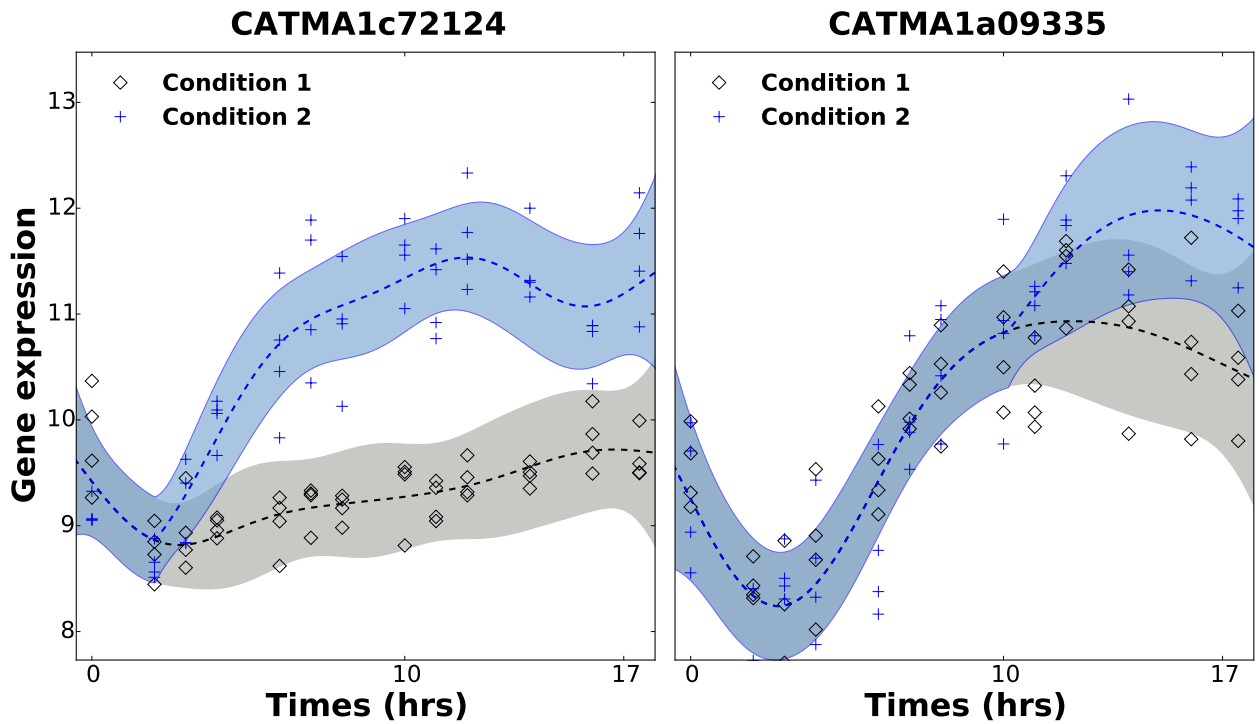
Posterior probability of the perturbation time t_p

$$p(t_p|y^c(\mathbf{T}), y^p(\mathbf{T})) \simeq \frac{p(y^c(\mathbf{T}), y^p(\mathbf{T})|t_p)}{\sum_{t=t_{\min}}^{t=t_{\max}} p(y^c(\mathbf{T}), y^p(\mathbf{T})|t)}$$



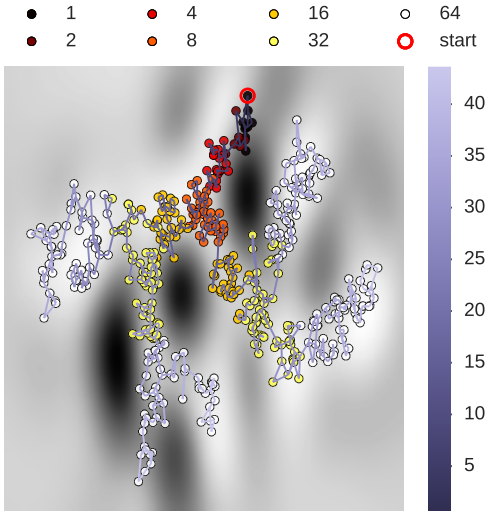
Application: plant response to bacterial challenge

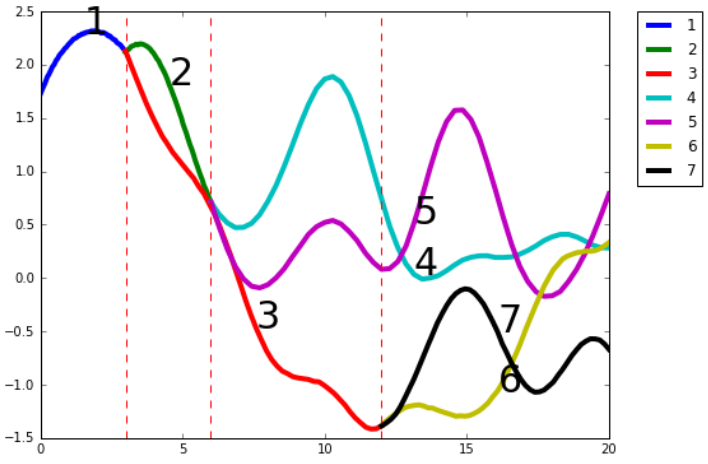
Infection with virulent *Pseudomonas syringage* pv. tomato DC3000 vs. disarmed strain DC3000*hrpA*

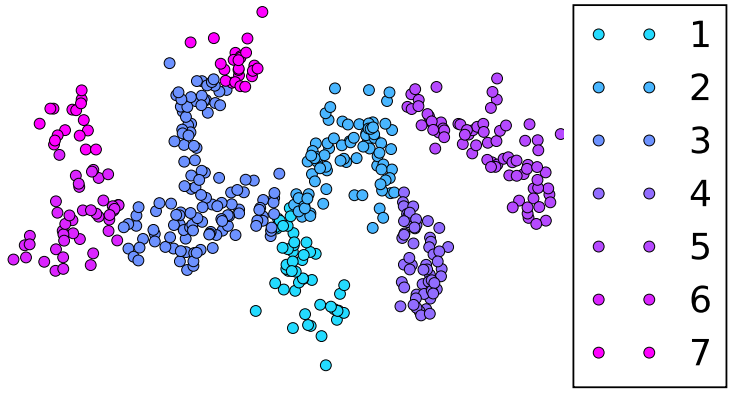


Yang *et al.* "Inferring the perturbation time from biological time course data" *Bioinformatics* (2016), 32 (19): 2956-2964

Modelling branching without labels: single-cell data







Single-cell snapshot data

High-throughput expression quantification in single cells

Cells may be at different points in a differentiation process

Several new problems for identifying branching dynamics:

- Where is the cell in the process? $\rightarrow$ *Pseudotime*
- Which branch does the cell lie on? $\rightarrow$ *Association*
- Which genes are involved? $\rightarrow$ *Branching evidence*
- When do they change? $\rightarrow$ *Branching time*

work with Alexis Boukouvalis and James Hensman

Our task: Branching evidence and branching time

- Assume each cell's pseudotime is known
- Similar to two-sample branching time-series problem: have to compute probability of data for every branching location
- But for genes branching earlier than global cellular branching we don't have the branch labels
- New inference task: inference over binary branch labels
- Assignment to branches is soft/probabilistic

Model definition

$F = \{f_1, f_2 \ldots, f_M\}$ is a branching Gaussian Process

$Z \in \{0,1\}^{N \times M}$ indicates which branch each cell comes from

$$p\left(Y|F,Z\right) = \mathcal{N}\left(Y|ZF, \sigma^2 I\right)$$

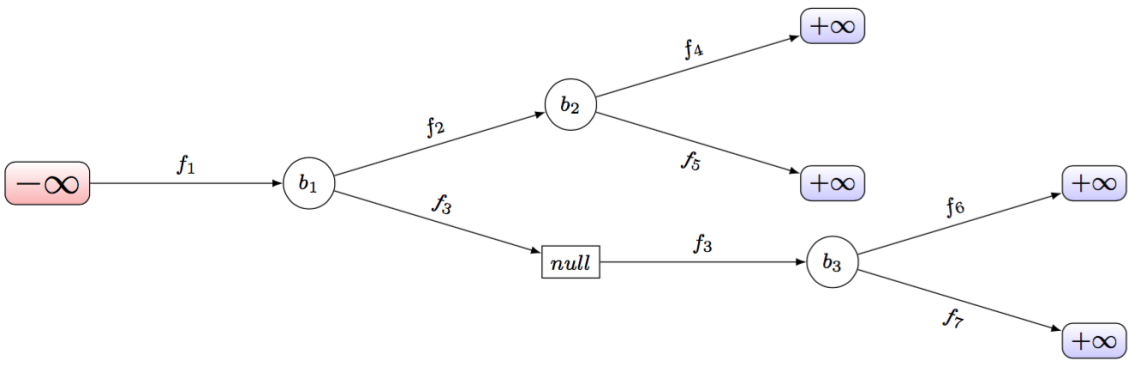
The likelihood conditional on the branching process is,

$$p\left(Y|F\right) = \int p\left(Y|F,Z\right) p(Z) dZ$$

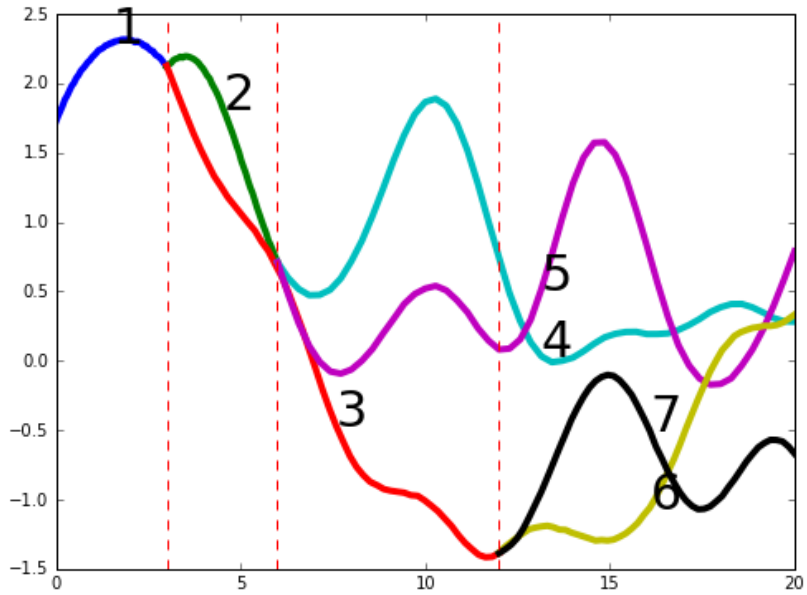
Global branching (from e.g. Monocle 2) can provide prior $p(Z)$

Model definition

Notional prior



Sample from the model



Inference

We construct a variational lower bound

$$\begin{aligned}
\log p(Y|F) &= \log \int p(Y, Z|F) \frac{q(Z)}{q(Z)} dZ \\
&= \log \left(\mathbb{E}_{q(Z)} \left[\frac{p(Y, Z|F)}{q(Z)} \right] \right) \\
&\geq \mathbb{E}_{q(Z)} \left[\log \frac{p(Y, Z|F)}{q(Z)} \right] \\
&= \mathbb{E}_{q(Z)} \left[\log p(Y, Z|F) \right] - \mathbb{E}_{q(Z)} \left[\log q(Z) \right]
\end{aligned}$$

which can be evaluated assuming a mean-field approximation

$$q(Z) = \prod_{tm} \phi_{tm}$$

and then F can be integrated out to get marginal likelihood $p(Y)$

Inference

Variational inference provides some useful things:

(1) Posterior probability of which branch cells belongs to

(2) Posterior probability of branching time

where latter is calculated using approximate marginal likelihood

$$p(t_b|Y) = \frac{p(Y|t_b)}{\sum_{t_b} p(Y|t_b)}$$

which is tractable for a single branching

(3) Bayes factor: branching versus not branching

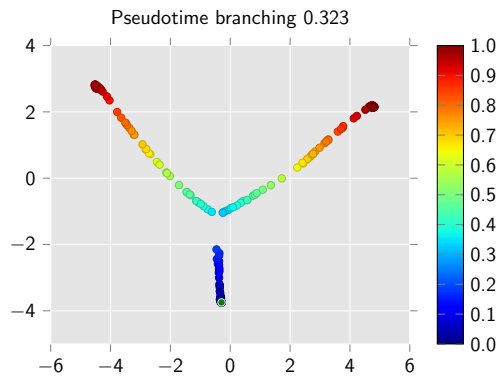
Synthetic data evaluation

Each gene has a different branching dynamics across time:

Scenario	Branching	Description
0	[0.2, 20], [1.1, 20]	Single branching
1	[0.2, 20], [0.6, 20]	All genes branching
2	[0.2, 15], [0.6, 15], [1.1, 10]	Multiple branching points
3	[0.2, 15], [0.6, 15], [1.1, 10]	Short lengthscale
4	[0.1, 3], [0.7, 27], [1.1, 10]	Majority of late branching genes
5	[0.1, 5], [0.3, 5], [0.5, 5], [0.7, 5], [1.1, 20]	Many branching locations
6	[0.2, 20], [1.1, 20]	High branching variance

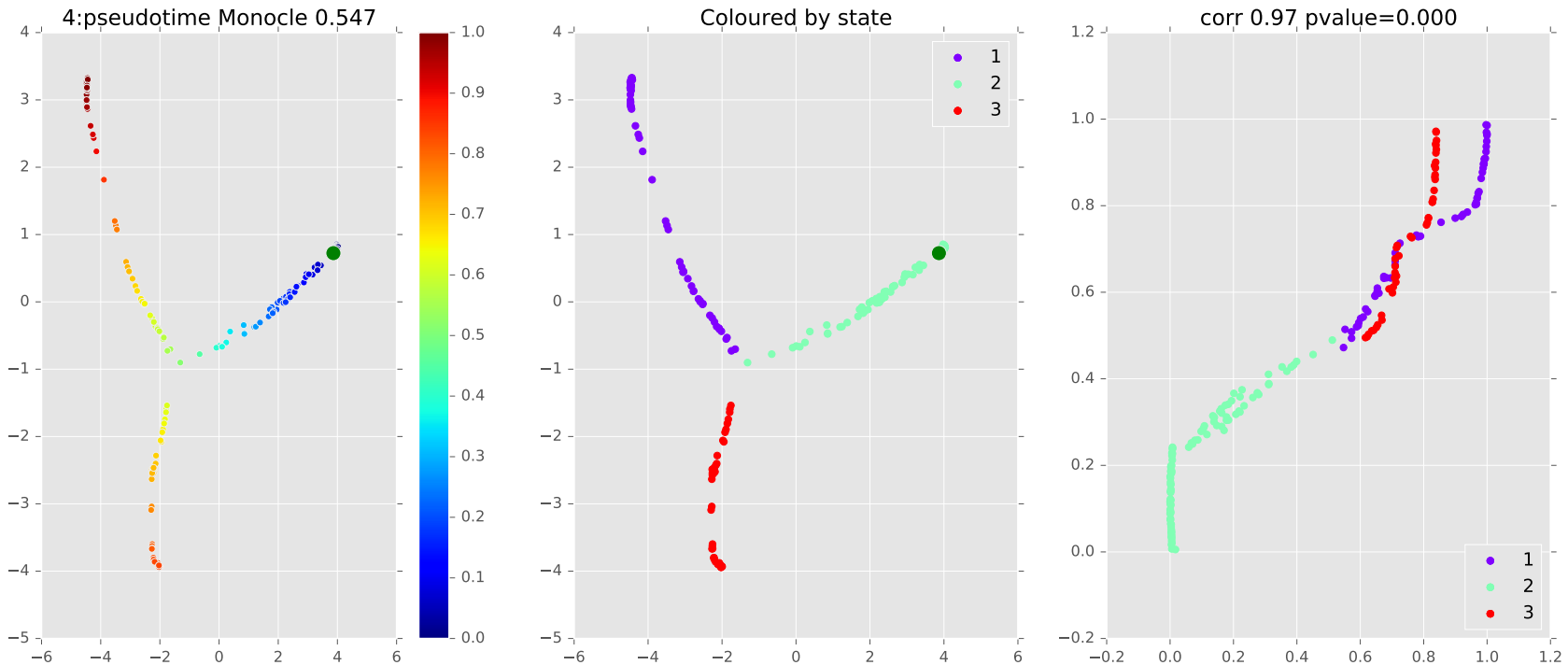
Hide time labels to simulate a single-cell RNA-Seq experiment

Use Monocle 2 algorithm to learn manifold and pseudotime:



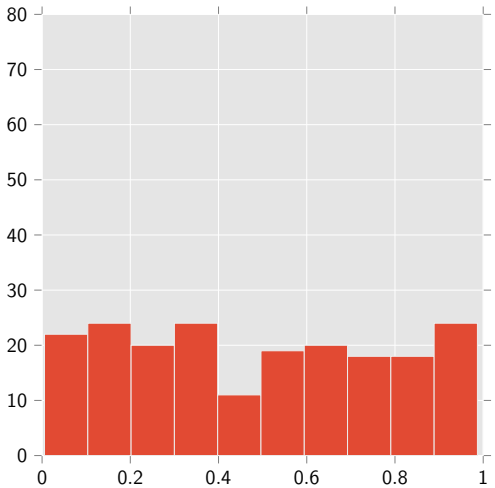
Qiu et al. 'Single-cell mRNA quantification and differential analysis with Census '(2017).

Monocle 2 performance

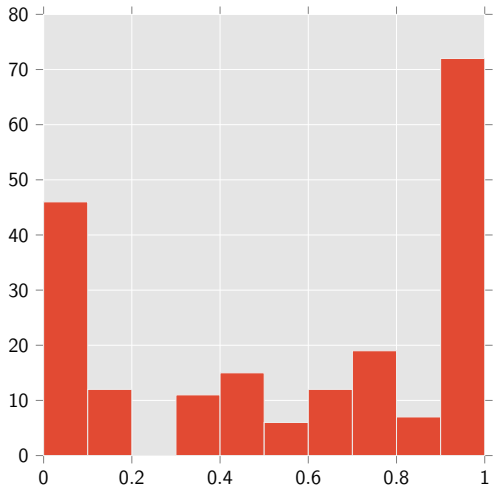


Time distortion

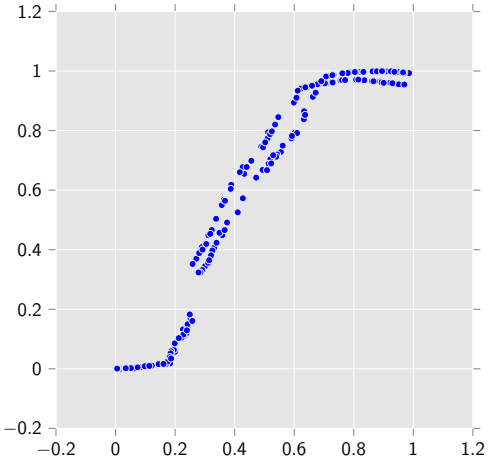
True time density



Pseudotime density



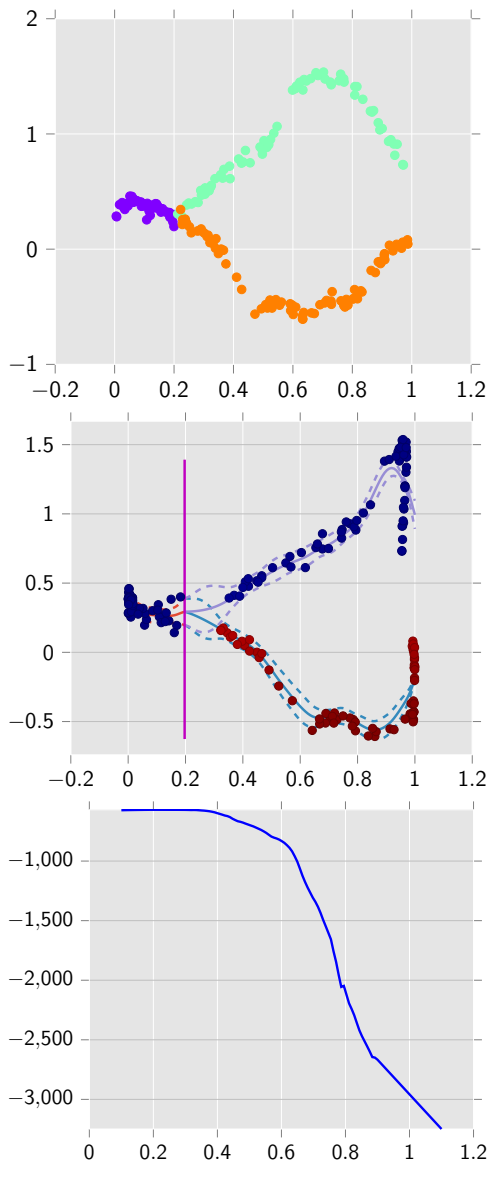
Rank correlation



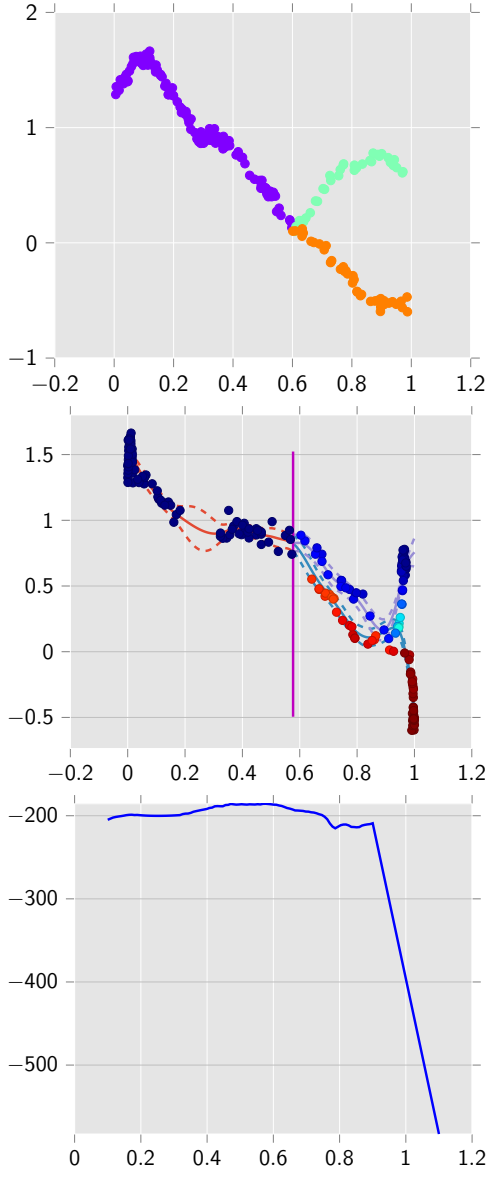
Time is compressed on the edges even though the rank correlation between the two times is very high (0.97)

Synthetic data: example fits

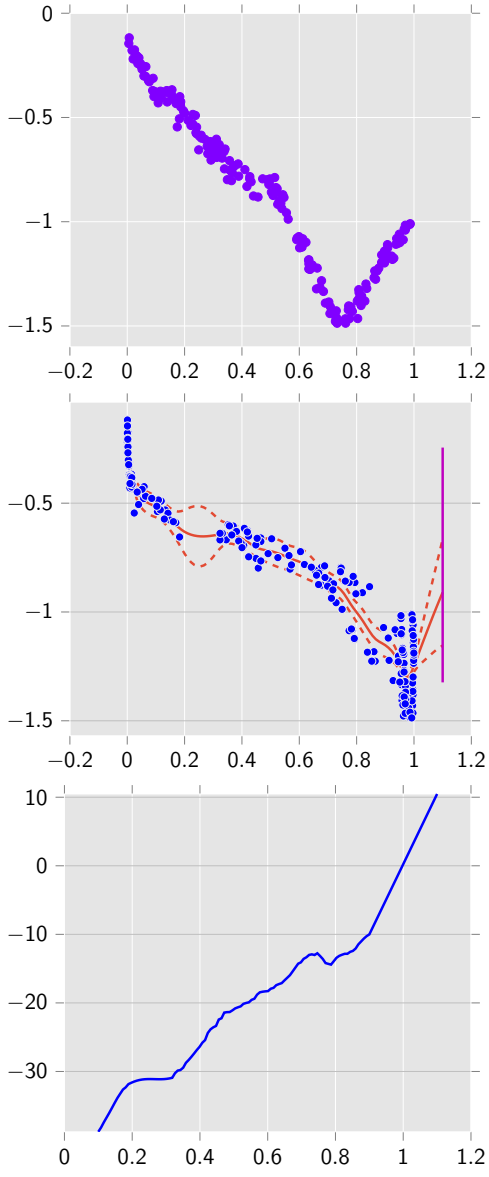
Early branching



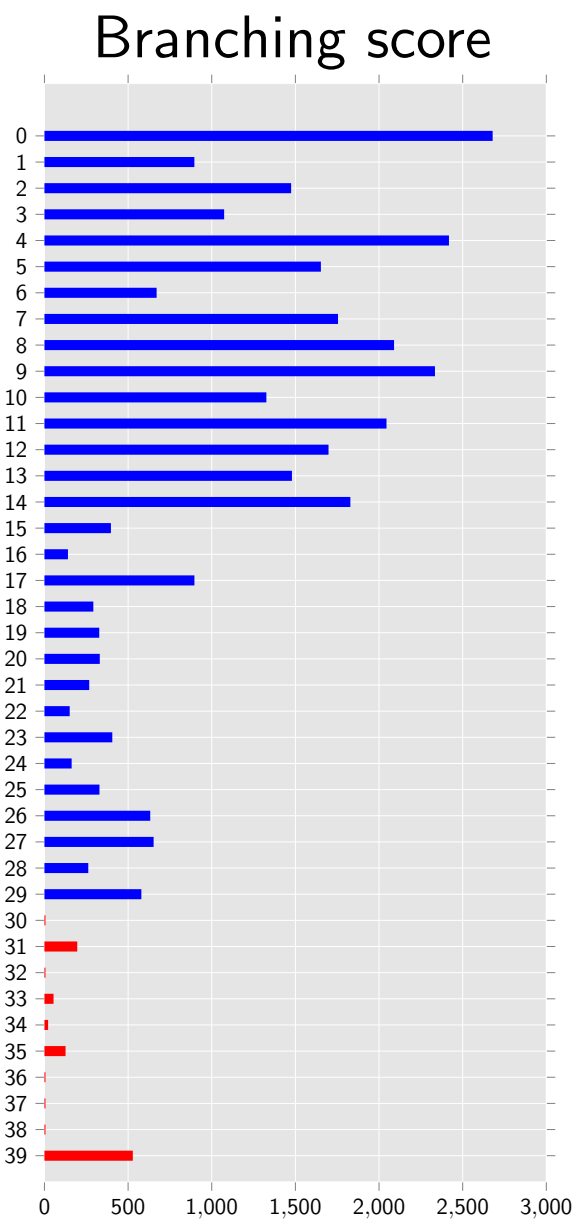
Late branching

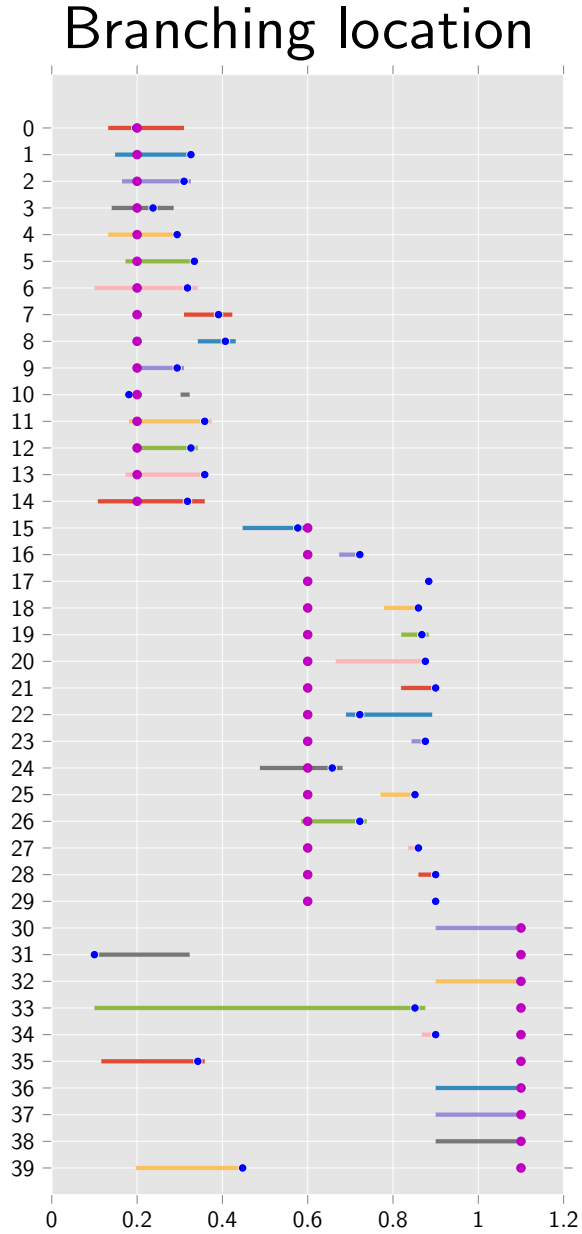


No branching

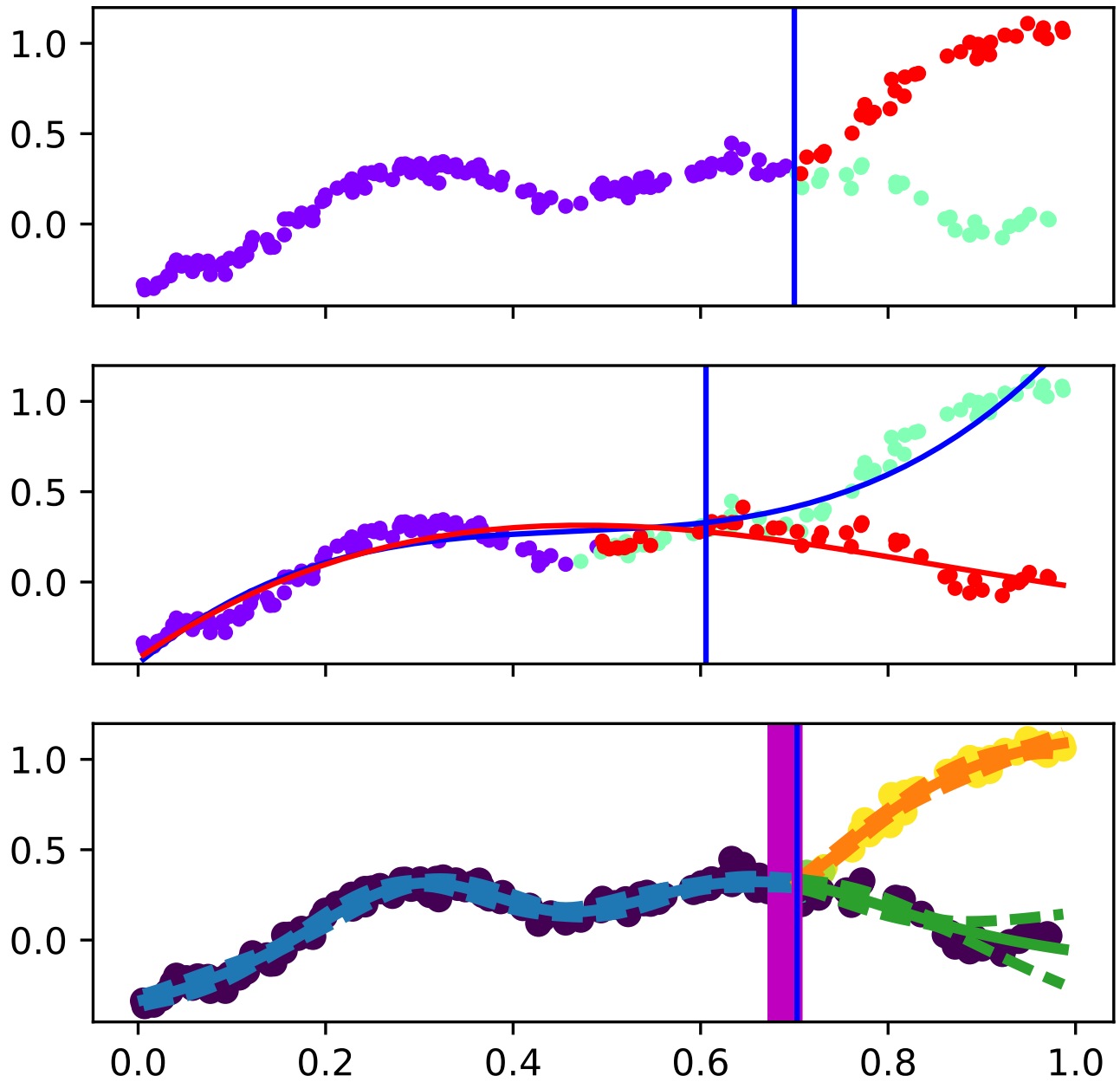


Synthetic data: branching probability and location

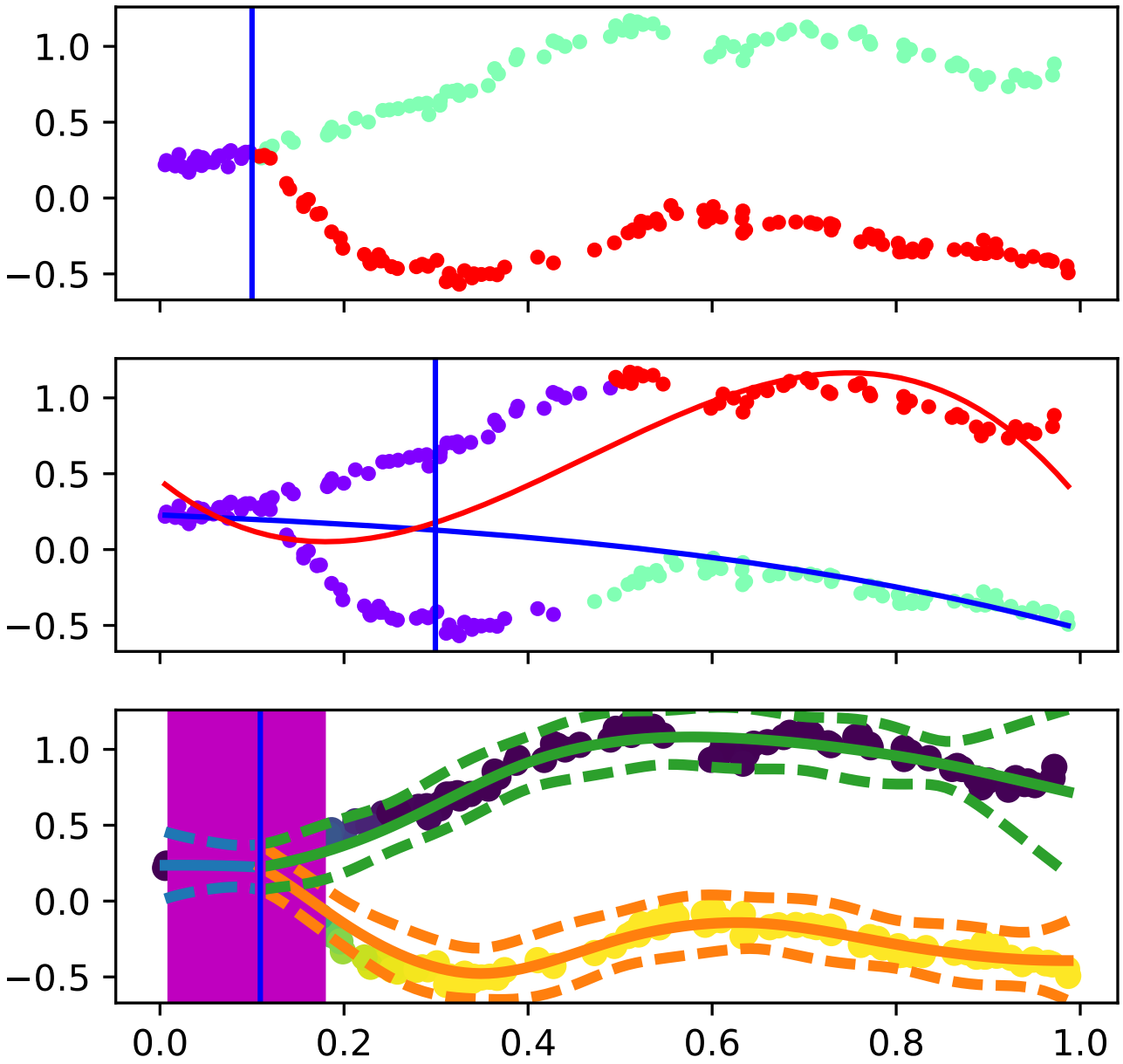




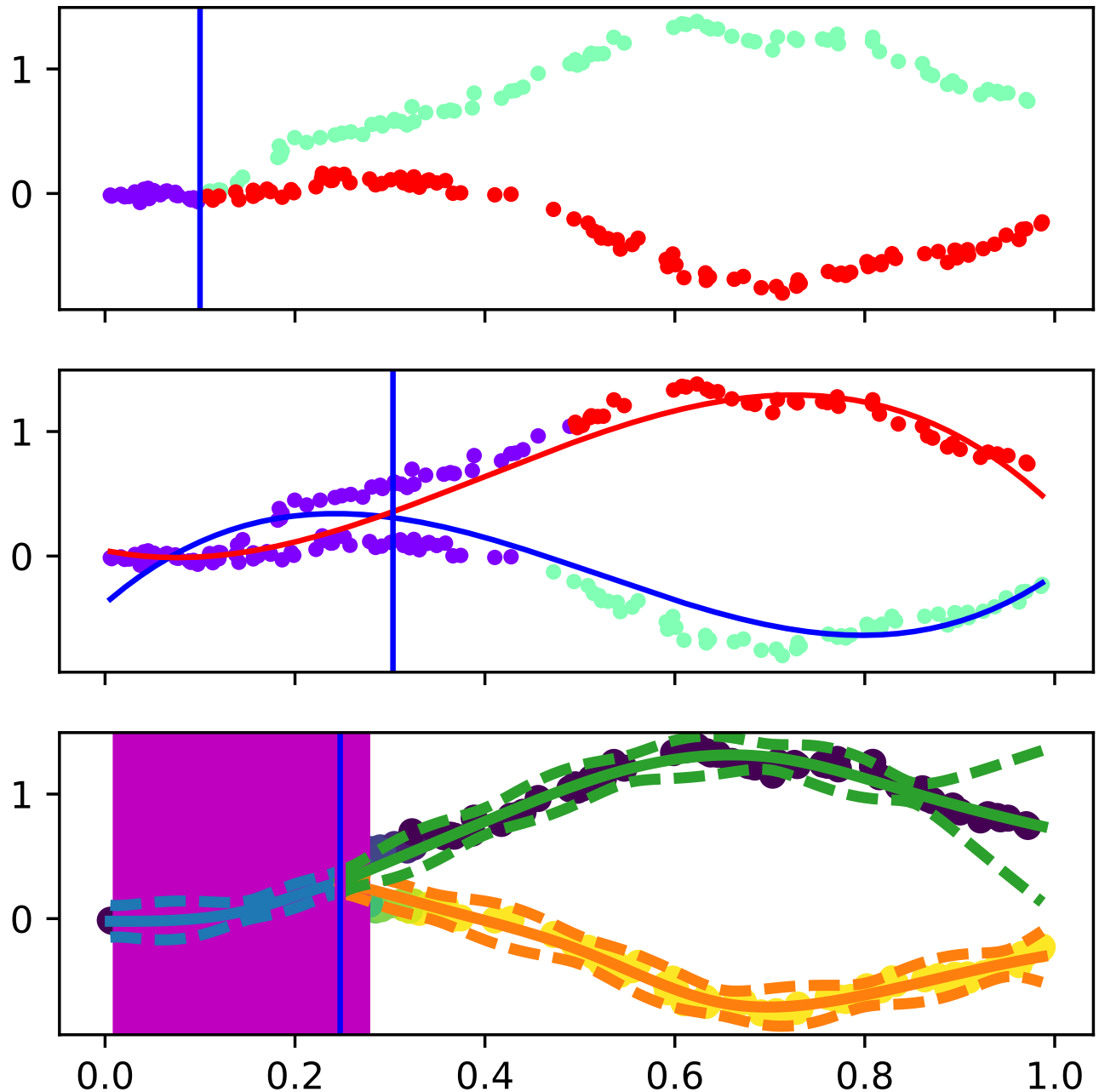
Comparison with splines (BEAM package)



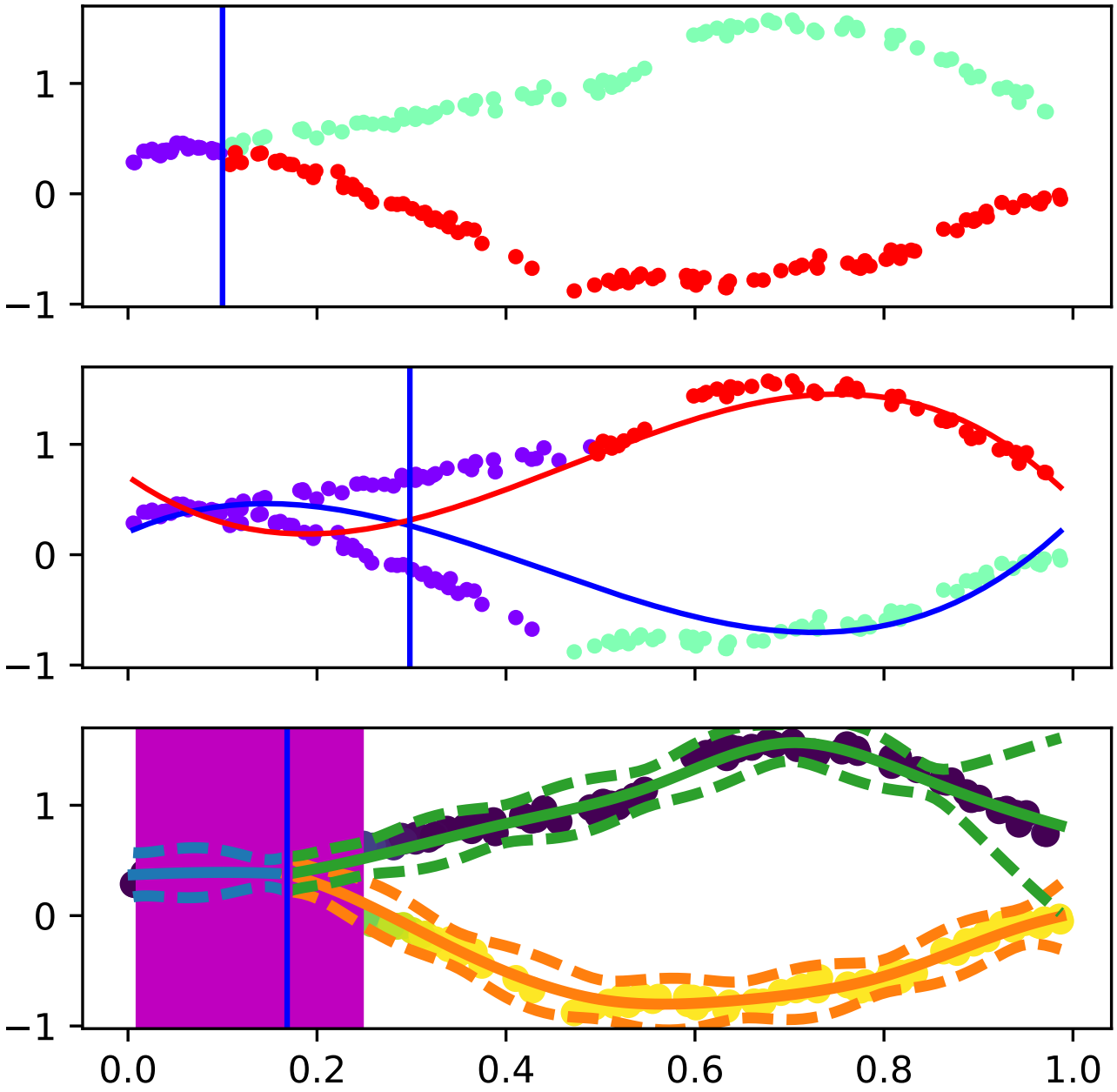
Comparison with splines



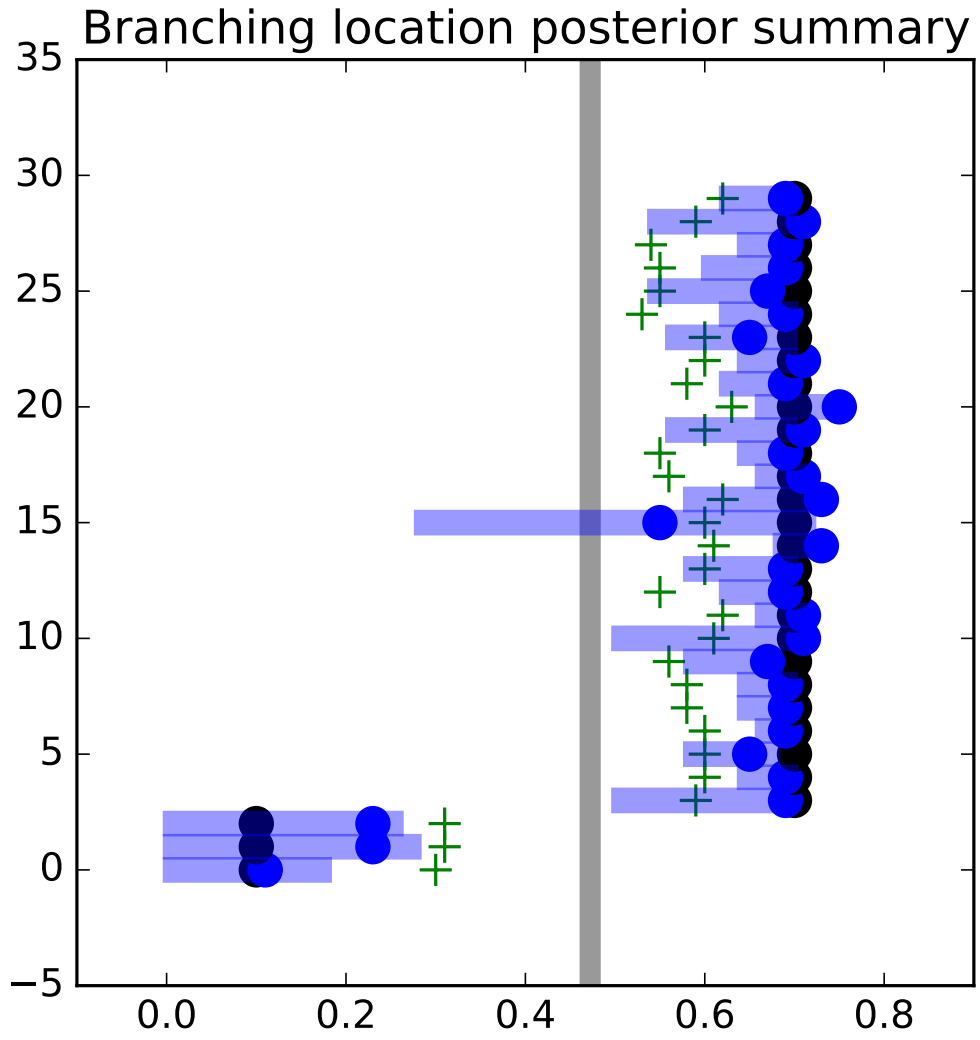
Comparison with splines



Comparison with splines



BEAM more biased towards global branch time



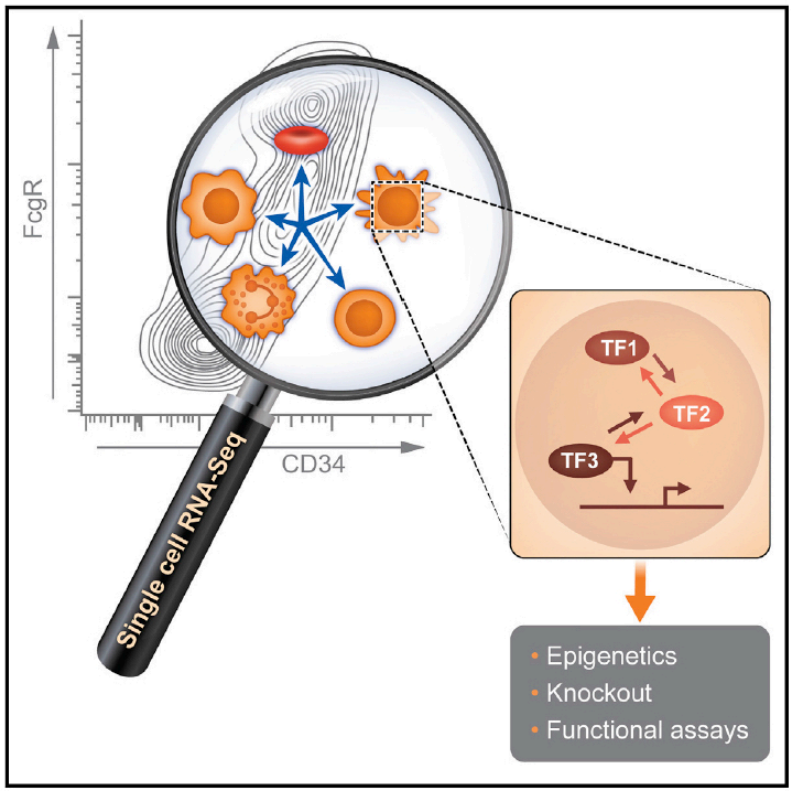
Example application

Article

Cell

Transcriptional Heterogeneity and Lineage Commitment in Myeloid Progenitors

Graphical Abstract



Authors

Franziska Paul, Ya'ara Arkin, Amir Giladi, ..., Bo Torben Porse, Amos Tanay, Ido Amit

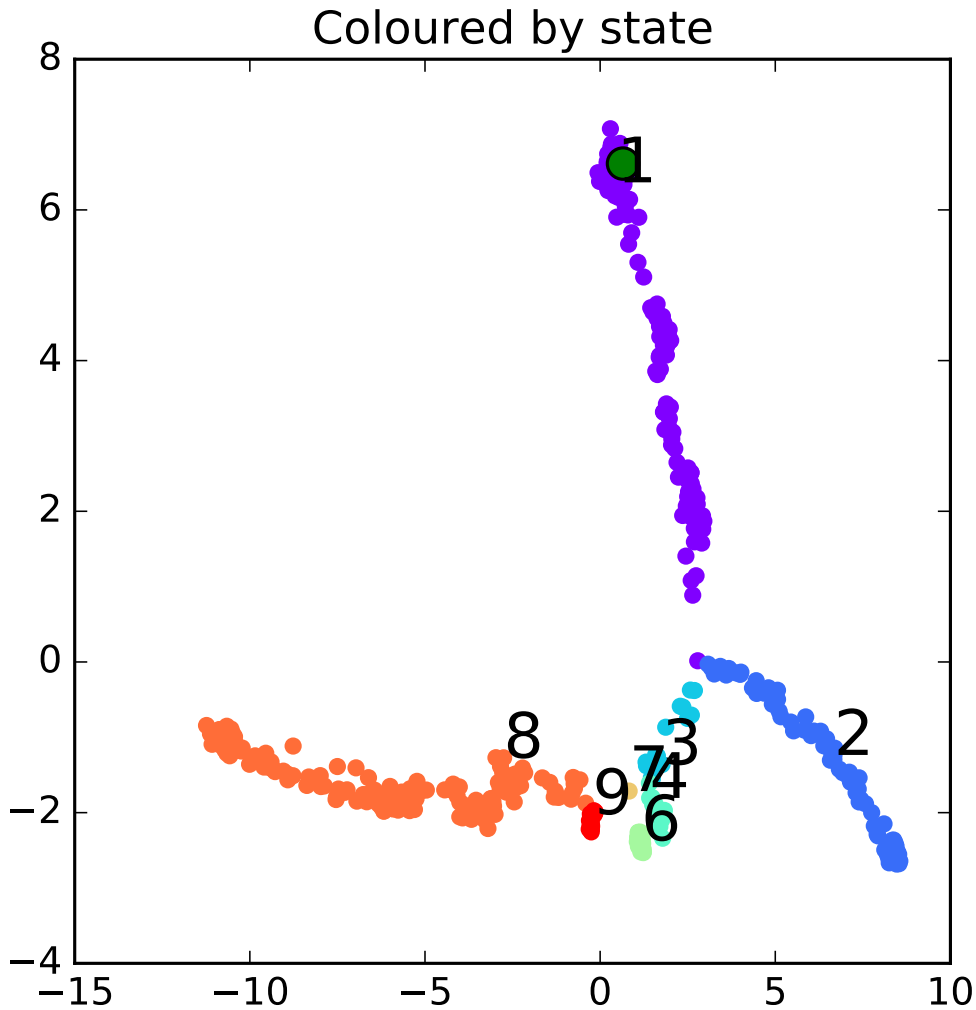
Correspondence

amos.tanay@weizmann.ac.il (A.T.), ido.amit@weizmann.ac.il (I.A.)

In Brief

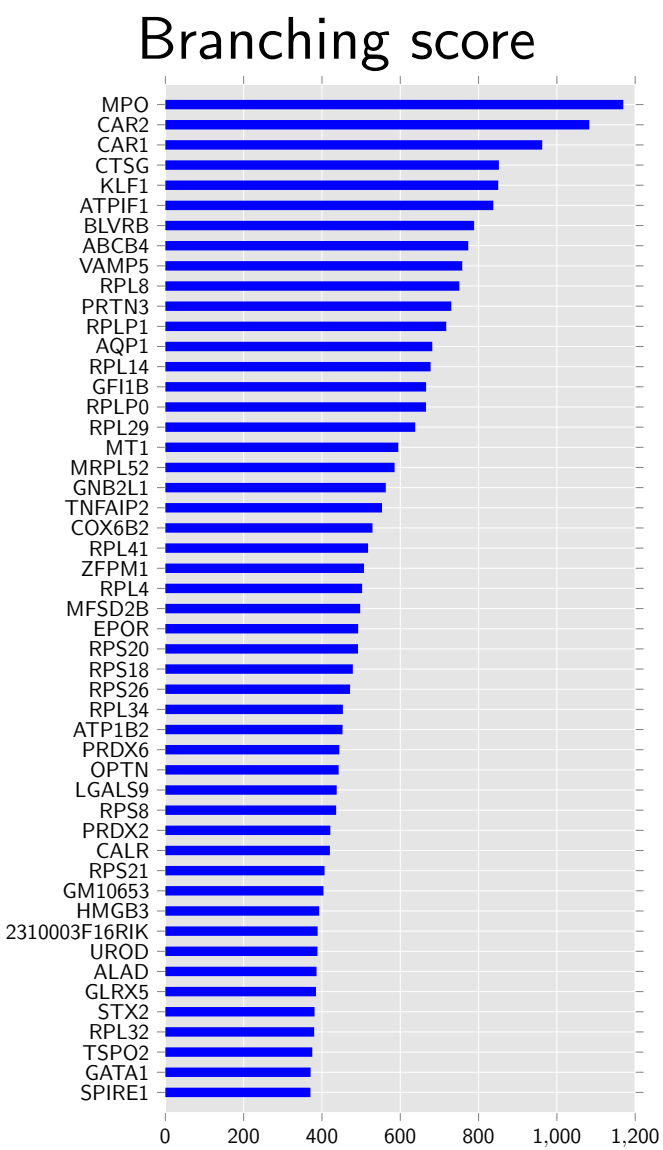
Single-cell transcriptomic analysis of bone marrow myeloid progenitor populations reveals early transcriptional priming toward seven different fates and absence of progenitors of mixed lineages, challenging the current models of hematopoiesis based on progressive loss of differentiation potential.

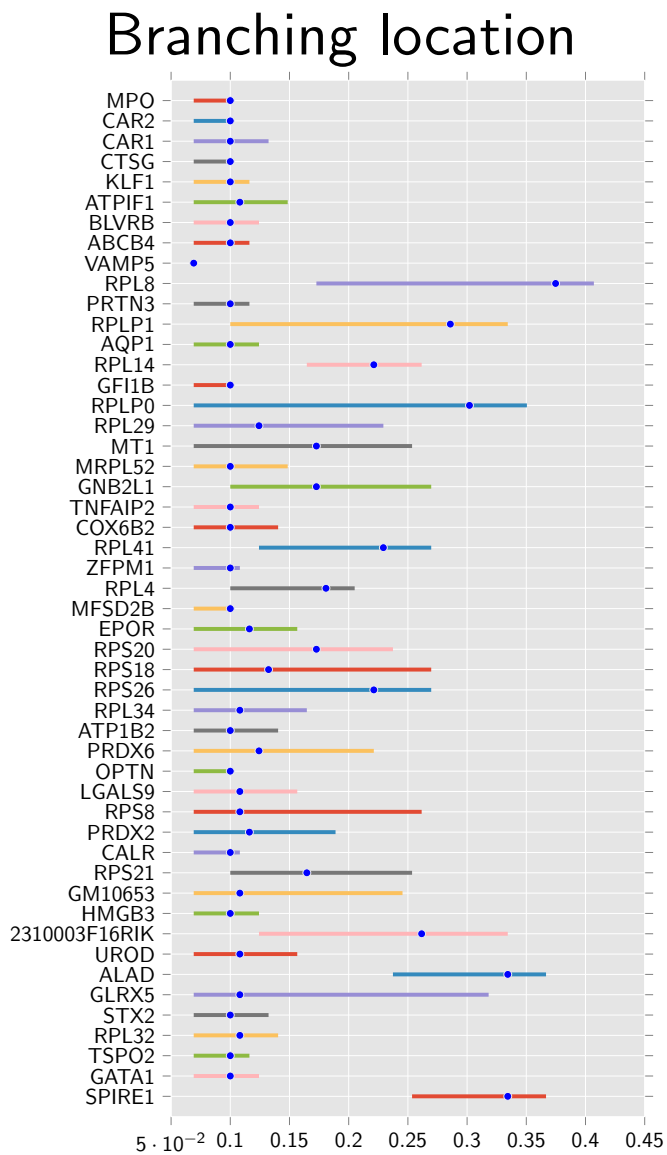
Monocle 2 projection and pseudo-time inference



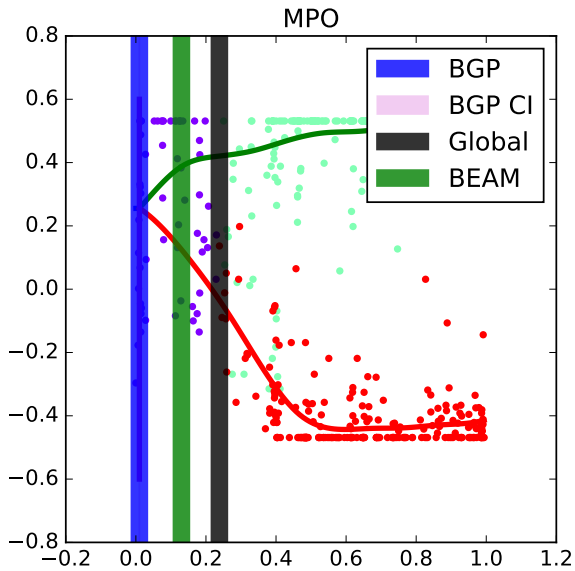
Trapnell et al. "Monocle: Cell counting, differential expression, and trajectory analysis for single-cell RNA-Seq experiments." (2016).

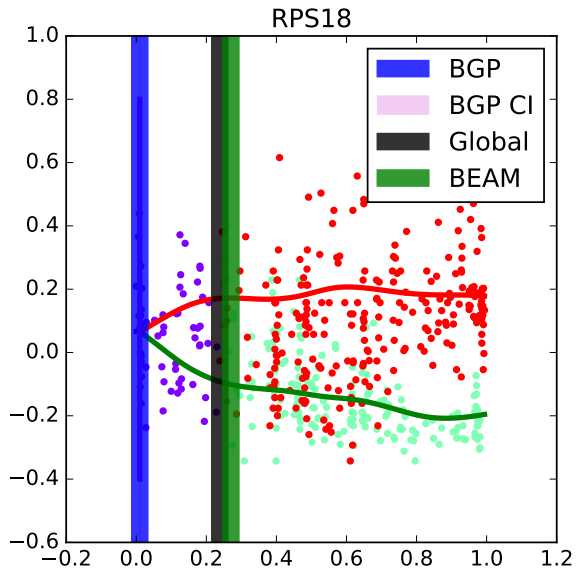
Single-cell: branching probability and location

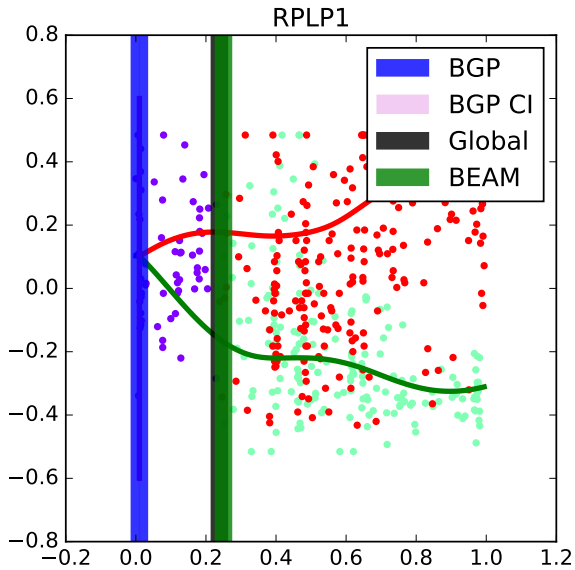


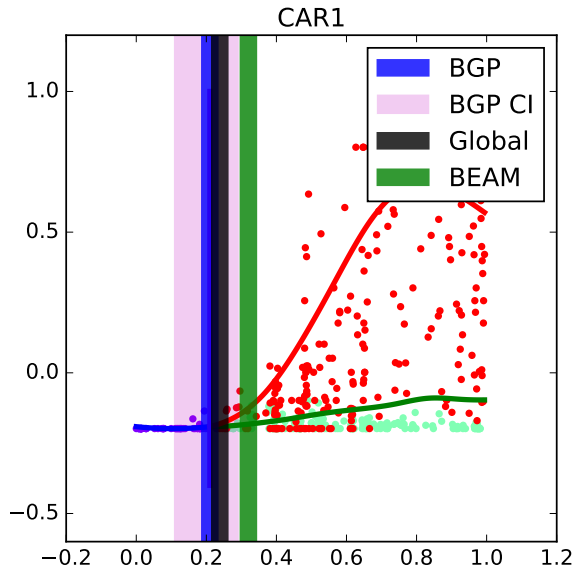


Single-cell: Example model fits









Part 4. Dimensionality reduction and pseudotime inference

RNA-Seq experiments can measure gene expression in single cells

Experiments are destructive – can't follow a cell through time

We can *infer* time in some dynamic process in the cell

Identifies a *pseudotemporal* ordering of cells

GPLVM for pseudotime inference with capture times

DeLorean package (Reid & Wernich 2016) uses Bayesian GPLVM for pseudotime inference with capture times τ_c

$$y_g(t) \sim \mathcal{GP}(0, k_t) \,\forall\, g \qquad t \sim \mathcal{N}(\tau_c, \sigma^2)$$

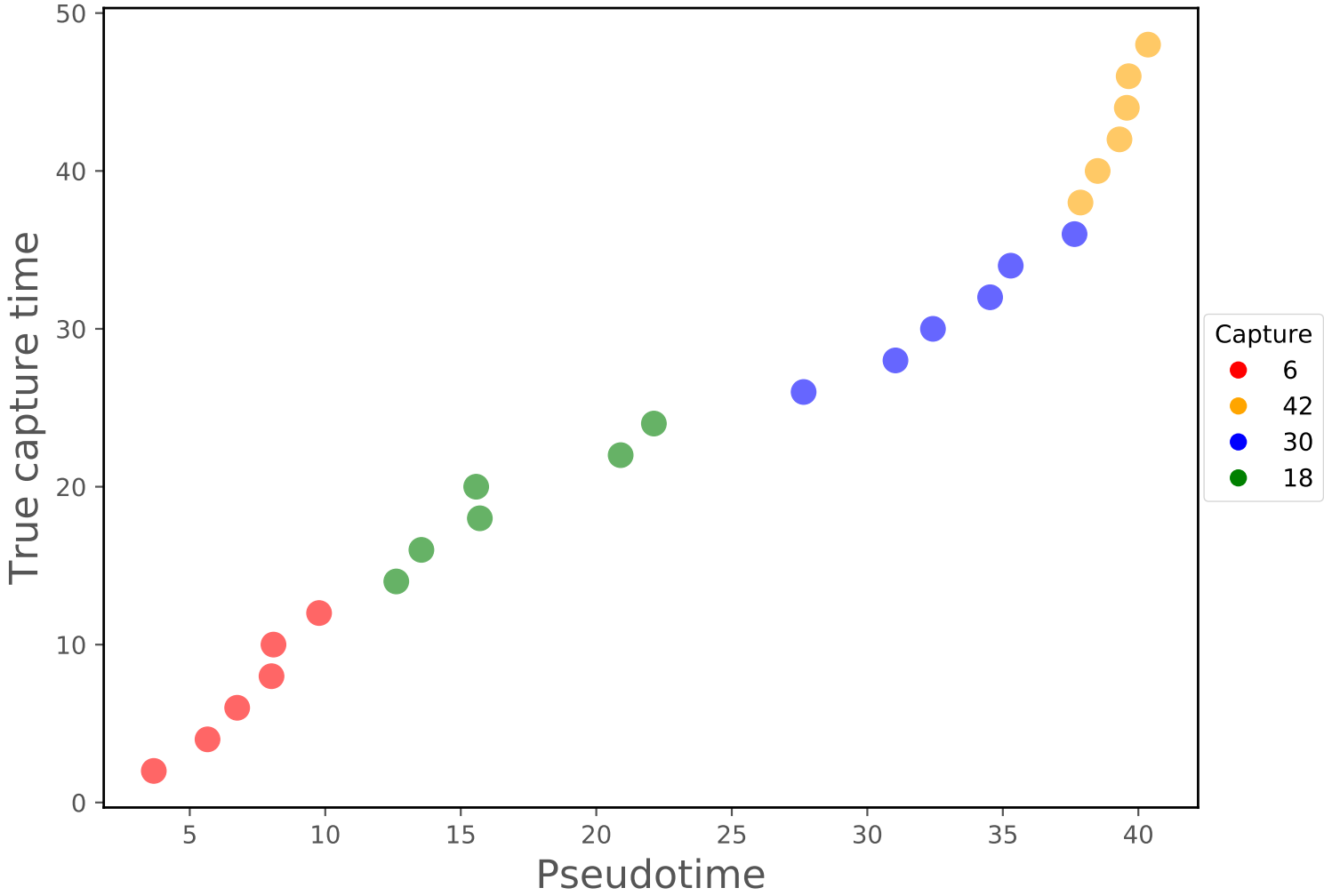
for gene g and inferred pseudotime t. We learn the regression function for each gene and cellular time t together.

We implemented a similar model using GPflow and sparse variational inference in the GrandPrix package,

$$y_g(t, x) \sim \mathcal{GP}(0, k_{xt}) \,\forall\, g \qquad t \sim \mathcal{N}(\tau_c, \sigma^2)$$

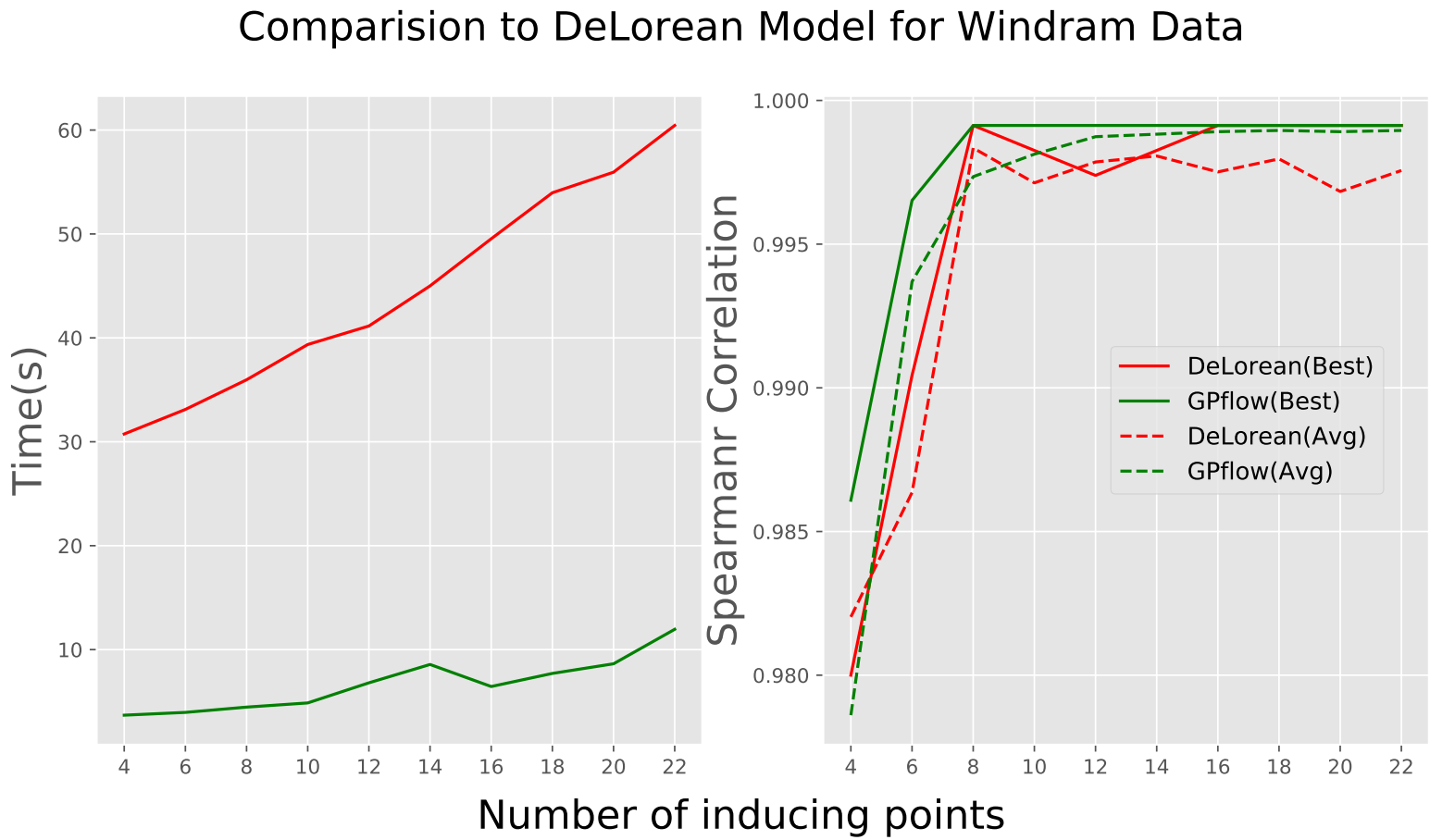
allowing for other sources of variation $x \sim \mathcal{N}(0, \sigma_x^2)$, e.g. branching

Benchmarking on time series data



Reid & Wernich benchmarked using time-series with hidden times

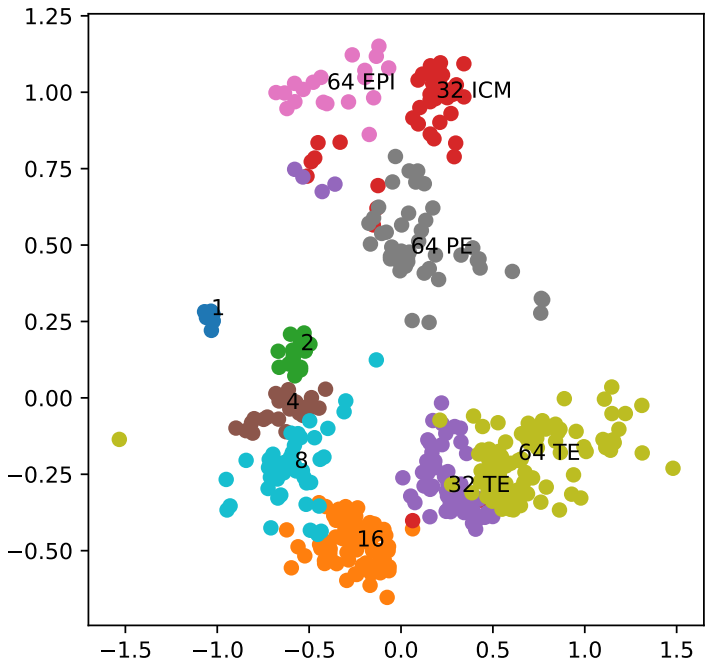
Benchmarking on time series data



Comparison using CPUs - GPUs give $\sim$ 10-fold further speed-up

New extension: pseudotime with branching

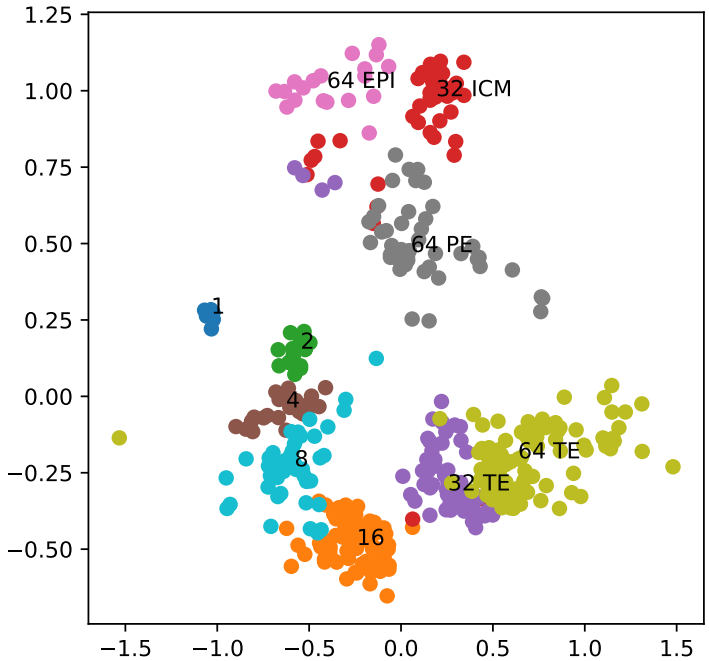
- Single cell qPCR data of early development (Guo et al. 2010)
- Gene expression of 48 genes measured across 437 cells
- Three cell states in the 64 cell stage: trophectoderm (TE), epiblast (EPI), and primitive endoderm (PE).
- Capture time helps disambiguate pseudotime from branching



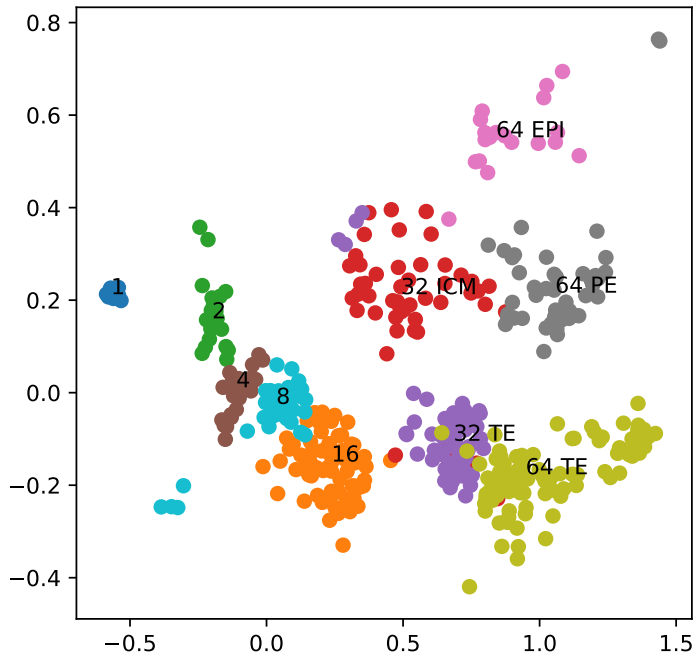
(a) Standard GPLVM

New extension: pseudotime with branching

- Single cell qPCR data of early development (Guo et al. 2010)
- Three cell states in the 64 cell stage: trophectoderm (TE), epiblast (EPI), and primitive endoderm (PE).
- Capture time helps disambiguate pseudotime from branching

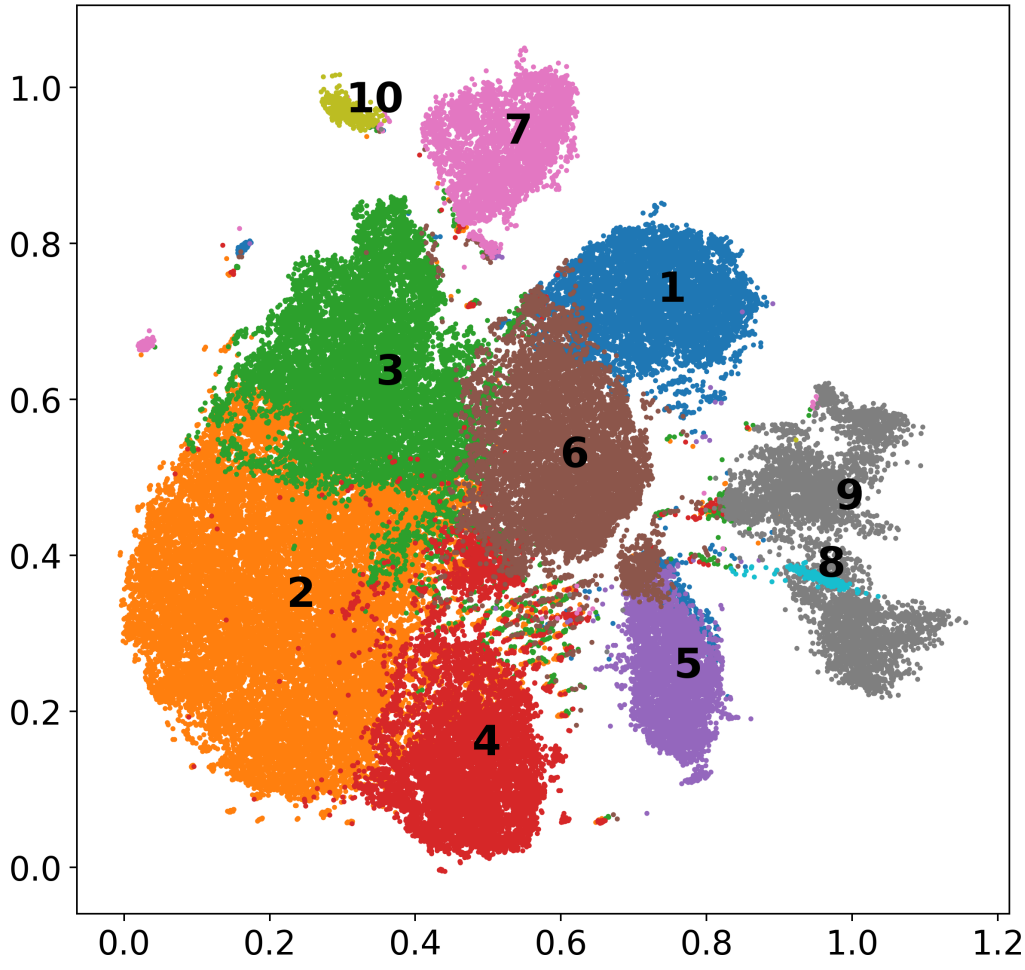


(a) Standard GPLVM



(b) GPLVM with informative prior

Scaling up to drop-seq data



Less than 10 mins for 68000 PBMCs and 1000 genes

Conclusions & Ongoing work

Summary

- Gaussian processes are flexible tools for modelling data
- Provide natural models of hierarchical relationships
- Provide a natural way to model branching time-series
- Speed-ups are important for single-cell applications

Next steps

- Extend to arbitrary number of branches
- Simultaneous inference of branching and pseudotime
- Non-Gaussian likelihoods, esp. for drop-seq data

Funding: MRC, Wellcome Trust Investigator Award

Collaborators: Magnus Rattray, Neil Lawrence, James Hensman, Sumon Ahmed, Jing Yang