Parametric and Non-parametric Gradient Matching for Network Inference

Leander Dony[1], Fei He[1,*], and Michael PH Stumpf[1,*]

[1]Centre for Integrative Systems Biology and Bioinformatics, Department of Life Sciences, Imperial College London, London SW7 2AZ, UK

Abstract

Reverse engineering of gene regulatory networks from time series gene-expression data is a challenging problem, not only because of the vast sets of candidate interactions but also due to the stochastic nature of gene expression. To avoid the computational cost of large-scale simulations, a two-step Gaussian process interpolation based gradient matching approach has been proposed to solve differential equations approximately. Based on this gradient matching approach, we evaluate the fits of parametric as well as non-parametric candidate models to the data under various settings for different inference objectives. We also use model averaging, based on the Bayesian Information Criterion (BIC), in order to combine the different inferences. We found that parametric methods can provide comparable, and often improved inference compared to non-parametric methods; the latter, however, require no kinetic information and are computationally more efficient.

The code used in this work is available at `https://github.com/ld2113/Final-Project`.

*Corresponding authors, email: fei.he@imperial.ac.uk, m.stumpf@imperial.ac.uk

1 Introduction

Gene expression is known to be subject to sophisticated and fine-grained regulation. Besides underlying the developmental processes and morphogenesis of every multicellular organism, gene regulation represents an integral component of cellular operation by allowing for adaptation to new environments through protein expression on demand [1, 2, 3, 4].

While the basic principles of gene regulation have been discovered as early as 1961 [5], understanding the structure and dynamics of complex gene regulatory networks (GRN) remains an open challenge. In this context experimental analysis has been complemented by a rich and growing literature on network reconstruction or inference, where statistical tools are used to "learn" or reconstruct interactions between sets of genes from data [1, 6, 7, 8].

Given the vast range of network inference approaches studied within and outside the life sciences, we necessarily have to limit our analysis, and we focus on a small but diverse set of inference approaches. Further details on other methods can be found in references [9, 10].

Gene regulatory interactions within a group of genes can be visualised in various ways. Usually, genes and their interactions are represented as nodes and edges of a graph respectively. Depending on the aim of the study and the employed method, the graph can be undirected (Figure 1A); directed (Figure 1B); or contain further information about interaction types (Figure 1C).

When inferring candidate regulatory networks from time-course data (e.g. time-resolved mRNA concentration measurements), different mathematical representations of the individual interactions can be used. The application of ordinary differential equation (ODE) models in this context has the advantage that each individual term in the final ODE model can provide direct mechanistic insight (such as presence of activation or repression) [11, 12]. Following [13] the ODEs of a GRN can e.g. be expressed as,

$$\begin{aligned} \dot{x}_n(t) &= s_n + \beta_n \cdot f_n(\mathbf{x}(t), \boldsymbol{\theta}_n, t) - \gamma_n \cdot x_n(t) \\ &= f(\mathbf{x}(t), \boldsymbol{\alpha}_n, t) \end{aligned} \tag{1}$$

Here, $x_n(t)$ denotes the concentration of n^{th} mRNA at time t, s_n is the basal transcription

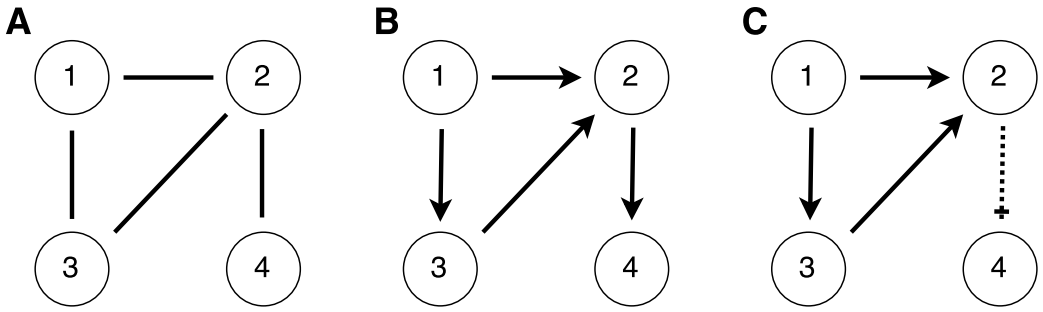

Figure 1: Gene regulatory network (GRN) schematics with four genes and four interactions. Three representations of the same GRN are shown. **A** Undirected graph showing interactions between genes: 1, 2; 1, 3; 2, 3; 2, 4. **B** Directed graph showing interactions between genes (parent node stated first): 1, 2; 1, 3; 3, 2; 2, 4. **C** Directed graph showing interactions between genes: 1 activates 2; 1 activates 3; 3 activates 2; 2 represses 4.

---

rate, γ_n is the mRNA decay rate, $\mathbf{x}$ is a vector of concentrations of all the parent mRNAs that regulate the n^{th} mRNA, the regulation function f_n describes the regulatory interactions among genes such as activation or repression that are normally quantified by Hill kinetics, with β_n the strength or sensitivity of gene regulation, and the parameter vector $\boldsymbol{\theta}_n$ contains regulatory kinetic parameters. The right-hand-side of the n^{th} ODE can be summarized in a single nonlinear function f with $\boldsymbol{\alpha}_n$ including all the kinetic parameters. Some approaches such as non-parametric Bayesian inference methods provide less mechanistic information but they may nevertheless provide realistic representations of complex regulatory interactions between genes, which a simple ODE system might not be able to capture [14], especially when accurate kinetic information is unavailable.

Parameter and structure inference of a mathematical model expressed as coupled ODEs (1) is a challenging problem, as repeatedly solving the ODEs by numerical integration is required which is computationally costly. Such costs quickly increase as the number of genes in the network increases. A two-step gradient matching approach has been proposed in the machine learning literature [15, 16, 17] to reduce the computational cost: in the first step, the time series data are interpolated, and in the second step, the parameters of ODEs are optimized by minimizing the difference between interpolated derivatives and the right-hand-side of ODEs. Thus the ODEs do not need to be solved explicitly. Previous work in the field of automatic network reconstruction has proposed a gradient matching approach to triaging different network topologies [11, 18]. Additionally, gradient matching enables treating each

species in the network independently.

Thus gradient matching for automatic ODE network reconstruction combined with Gaussian process (GP) regression could be a promising avenue for inferring GRNs. But still, some problems remain: model identifiability, as too many models provide a good fit to the data; reliably fitting HGPs to noisy data; and potentially limiting model assumptions, e.g. by considering only a limited range of interaction types.

In this work, we investigate and attempt to address those issues and furthermore evaluate inference performance of gradient matching approach under different conditions. We structure our work by comparing the inference performance of parametric and non-parametric inference methods as described in Figure 2.

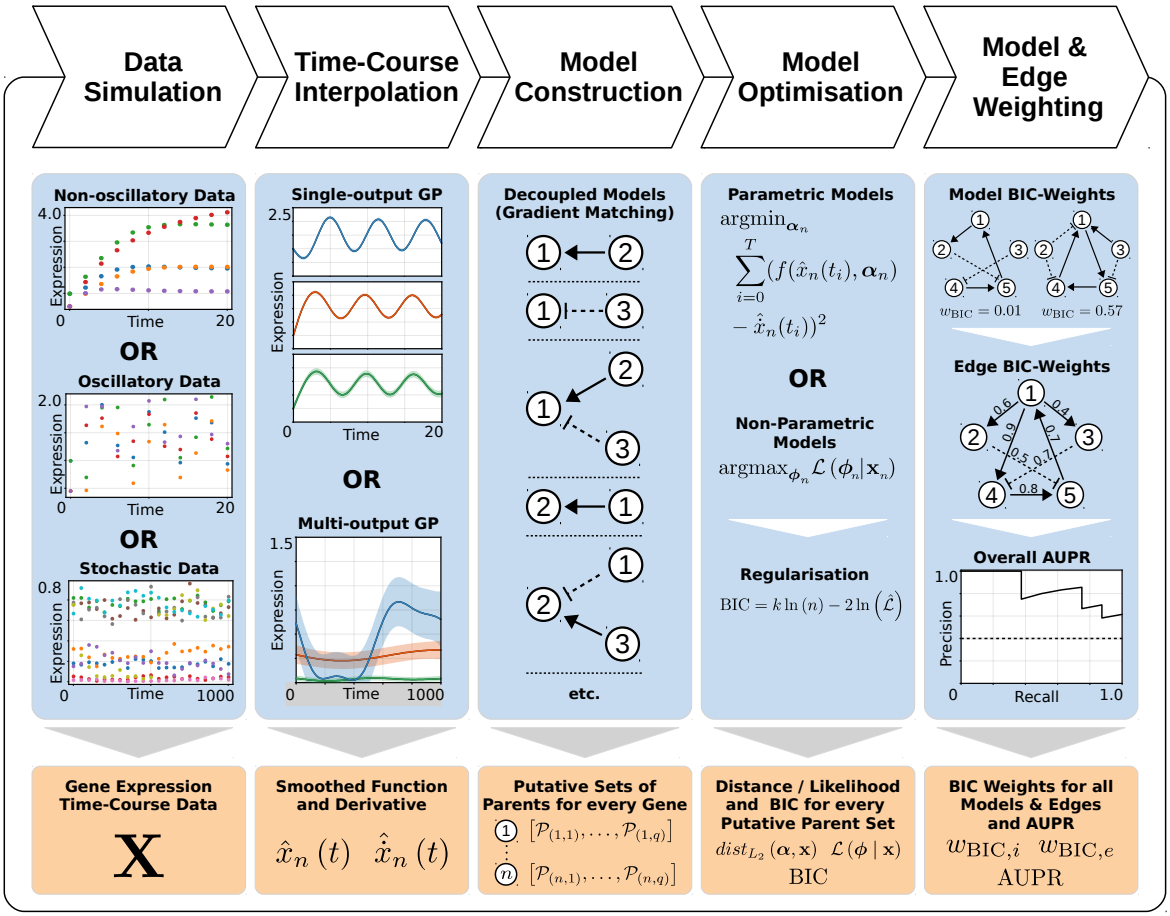

Figure 2: Pipeline outline schematic. This figure illustrates the five main steps in the network inference pipeline developed in this project. All numbers and schematics are shown purely for illustration and do not reflect actual results. Abbreviations used here are: **GP** – Gaussian Process; **BIC** – Bayesian Information Criterion; **AUPR** – Area under the precision-recall curve.

2 Methods

This section outlines the different approaches taken to reconstruct GRN. Details on the software and algorithms employed can be found in appendix D.

2.1 Gene Expression Data

To compare different network inference approaches and settings, we simulate deterministic gene expression data from a relative small 5-gene regulatory network. We then repeat the analysis using more realistic stochastically simulated data generated from a 10-gene regulatory network in *Saccharomyces cerevisiae*.

2.1.1 Deterministic ODE Model Simulation

We use deterministically simulated gene expression data based on the *in vivo* benchmarking of reverse-engineering and modelling approaches (IRMA) network [19]. The IRMA network is a quasi-isolated synthetic five-gene network, constructed in *Saccharomyces cerevisiae* (Figure 3A). We refer to this dataset as 'non-oscillatory data'.

To ensure comparability to previous work with this model [11, 18], we use the same model parameters and also create a second subset with one edge removed (Figure 3B) and regulatory interactions modelled as previously [11, 18, 20]. We refer to this dataset as 'oscillatory data'. For completeness we provide the structure of the ODE systems as well as the parameters and settings used for simulation once again in appendix A.1.

2.1.2 *In silico* Gene Expression Data

In order to evaluate the performance of different inference methods with more realistic stochastically simulated gene expression data, we use `GeneNetWaver` [21] to generate realistic gene expression profiles from an *in silico* ten gene network (Figure 3C) from *Saccharomyces cerevisiae* (as previously used in the DREAM3 and DREAM5 challenge [22]). The dataset

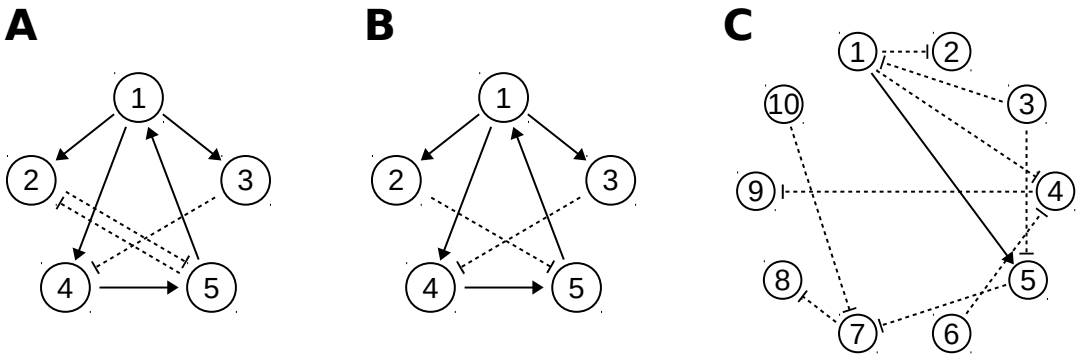

Figure 3: Schematics of gene regulatory networks used in this work. A Five-gene network with eight interactions used to simulate the 'non-oscillatory noise-free data'. **B** Five-gene network with seven interactions used to simulate the 'oscillatory noise-free data'. **C** Ten-gene network with ten interactions used by `GeneNetWeaver` to simulate the 'realistic *in silico* data'.

---

we used is referred to as `InSilicoSize10-Yeast1_dream4` in `GeneNetWeaver`. We obtain data for the same 20 time points for every gene.

`GeneNetWaver` [21] simulates realistic noisy gene expression data by introducing process noise (through stochastic differential equations) as well as observational noise to the underlying gene expression profiles.

2.2 Data Smoothing with Gaussian Processes

For smoothing and interpolation of the potentially noisy gene expression data, we use Gaussian process (GP) regression. This also allows us to obtain the rate of change in the expression via the GP derivative, which is analytically obtainable. In this section we only provide a very brief introduction to the theoretical foundations of GPs and mainly focus on outlining our choices and settings used in the GP framework. For more detail we refer to [23, 24, 25].

2.2.1 Gaussian Process Regression

A GP is defined by a mean m and covariance function k, so that we can write $f(t) \sim \mathcal{GP}(m, k)$ for any suitable function f. Any finite collection of values from $f(t)$ are hence distributed according to a multivariate Gaussian distribution and so we can write $[f(t_1), \ldots, f(t_D)] \sim$

$\mathcal{N}(\mathbf{m}, K)$. $\mathbf{m}$ describes the vector of n mean values and $K = k(t, t')$ is the covariance matrix, where the value of each element is defined by the GP covariance function.

We use a zero mean function and employ the common squared exponential covariance function [23], which defines the covariance between two observations at time points t and t' as

$$k(t, t') = \sigma_f^2 \exp\left(\frac{-(t-t')^2}{2l^2}\right) + \sigma_n^2 \delta(t, t') \tag{2}$$

with σ_f^2 controlling the variance ('amplitude') of the the GP, and the length-scale l controlling how many data points around the current one are taken into account when fitting the GP. σ_n^2 denotes the variance of the observational noise and we can write $x(t) \sim \mathcal{N}\left(f(t), \sigma_n^2\right)$. $\delta(t, t')$ denotes the Kronecker delta function, which is 1 if $t = t'$ and zero otherwise.

We optimise the hyperparameters $\boldsymbol{\phi} = \{\sigma_f, \sigma_n, l\}$ by maximising,

$$\ln p(\mathbf{x} \mid \mathbf{t}, \boldsymbol{\phi}) = -\frac{1}{2}\mathbf{x}^\top K^{-1}\mathbf{x} - \frac{1}{2}\ln|K| - \frac{D}{2}\ln 2\pi \tag{3}$$

where K corresponds to the covariance matrix and D denotes the number of observations in vector $\mathbf{x}$. $\mathbf{t}$, $\mathbf{x} \in \mathbf{R}^D$.

We obtain predictions $\mathbf{x}_*$ at time points $\mathbf{t}_* = [t_1^*, t_2^*, ..., t_S^*]$ from the GP model, since the joint (prior) probability distribution of the training output $\mathbf{x}$ and testing output $\mathbf{x}_*$ is again multivariate Gaussian,

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{x}_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K & K_*^\top \\ K_* & K_{**} \end{bmatrix}\right), \tag{4}$$

where

$$K = k(t, t'), \quad K_* = k\left(t, t_*'\right), \quad K_*^\top = k\left(t_*, t'\right), \quad K_{**} = k\left(t_*, t_*'\right).$$

The posterior distribution of the output at $\mathbf{t}_*$ can be calculated as,

$$\mathbf{x}_* | \mathbf{x} \sim \mathcal{N}\left(K_* K^{-1}\mathbf{x}, K_{**} - K_* K^{-1} K_*^\top\right). \tag{5}$$

2.2.2 Gaussian Process Derivatives

We can also directly obtain the derivative of the GP, representing the rate of change in mRNA concentration $\dot{\mathbf{x}}_*$, as the derivative of a GP is again a GP [24, 26],

$$\begin{aligned} \frac{d\mathbf{x}_*}{dt} &= L_* K^{-1} \mathbf{x}, \\ [L_*]_{ij} &= \frac{d}{dt_j} k\left(t_i, t_j^*\right) = \frac{(t_i - t_j^*)}{l^2} [K_*]_{ij} \end{aligned} \tag{6}$$

The derivatives obtained here will also be used for gradient matching inference algorithm to be discussed next.

2.2.3 Multiple Output Gaussian Processes

Standard GP regression allows us to make predictions on the expression level of a single gene. To improve the GP fitting to multiple genes, intrinsic coregionalisation for multi-output GP regression [27] ia employed. This is a form of a multiple output GP [28] which takes into account correlation between the expression of all genes in the network through a correlated noise process. Considering a system with N outputs, the overall covariance (or kernel) matrix K of the multi-output GP takes the form,

$$K(X, X) = B \otimes k(X, X), \tag{7}$$

where $B \in \mathbf{R}^{N \times N}$ is the coregionalisation matrix, $X = \{\mathbf{x}_i\}_{i=1}^N \in \mathbf{R}^{ND}$ is the input vector that contains observations for all the N outputs, and $\otimes$ denotes the Kronecker product. If $B = I_N$, then all outputs are uncorrelated. The hyperparameters in the covariance function $k(X, X)$ and B can be estimated jointly via the eigen-decomposition of the matrix B and maximum likelihood estimation [29].

We obtain the smoothed mRNA concentration values from the mean function of the GP. We can approximate the derivative at each point numerically,

$$\frac{dx}{dt} \approx \frac{x(t + \delta) - x(t)}{\delta}. \tag{8}$$

Here we use $\delta = 10^{-4}$.

2.3 Model Construction and Optimisation through Gradient Matching

We use a gradient-matching parameter optimisation approach to evaluate the goodness of fit of our model to the data [11, 14, 18]. Instead of solving the ODE systems, we directly compute the gradient of the gene expression data using GP regression (see Section 2.2) and then optimise the parameters of the ODE system.

As gradient matching can be carried out for each equation of the ODE system independently, the number of possible network topologies we have to consider reduces drastically. For the five gene network with two alternative interaction types and no self-interactions, we only have to consider $N \cdot \sum_{i=0}^{N-1} \left(\binom{N-1}{i} \cdot F^i \right) = 405$ topologies, given the decoupled system (opposed to $N = 3.5 \cdot 10^9$ fully coupled models). We can further limit the number of topologies by restricting the number of maximum parents per gene (e.g. the maximum in-degree of every gene in the network). A reasonable assumption would be a maximum of $M = 2$ parents per gene, which would further reduce the space of candidate topologies to $N \cdot \sum_{i=0}^{M} \left(\binom{N-1}{i} \cdot F^i \right) = 165$.

2.3.1 ODE Models

As during data simulation (Section 2.1.1) we use two different approaches to model activation and repression during network inference. The parameters and constraints used for model optimisation are provided in the Appendix A.2.

For the n^{th} ODE we minimize the L_2 (squared) distance between the constructed parametric function $f(\hat{x}_n(t), \boldsymbol{\alpha}_n)$ (with parameter vector $\boldsymbol{\alpha}_n$) and the associated derivative calculated from the GP regression $\hat{\dot{x}}_n(t)$ for all S time points $[t_0, \ldots, t_S]$:

$$\text{dist}_{L_2,n} = \sum_{i=0}^{S} (f(\hat{x}_n(t_i), \boldsymbol{\alpha}_n) - \hat{\dot{x}}_n(t_i))^2 \tag{9}$$

2.3.2 Non-Parametric Models

We also consider a fully non-parametric, GP-based gradient matching inference method adapted from [14]. This is particularly useful when the detailed reaction kinetics (i.e. ODEs) are unknown and when we are more interested to infer the network interactions instead of the kinetics or reaction types (i.e. activation or repression). Similar to the decoupled ODE system described in the previous section, gradient matching approach can also be integrated with non-parametric GP regression. This allows for treating each gene n conditionally independent of all other genes given its parents $\mathcal{P}_n$. We model each gene using the relationship:

$$\dot{x}_n(t) = f(\{\mathbf{x}_q(t) \mid q \in \mathcal{P}_n\}, \boldsymbol{\phi}_n) \tag{10}$$

where $f(\{x_q(t) \mid q \in \mathcal{P}_n\}, \boldsymbol{\phi}_n) \sim \mathcal{GP}(0, k)$ is a single-output-multiple-input GP with $\boldsymbol{\phi}_n$ denoting the vector of hyper-parameters for the squared exponential covariance function $k(t, t')$ (equation 2) for gene n. The derivative of the n^{th} gene expression $\dot{x}_n(t)$ can again be obtained from the derivative GP process. Optimisation of each putative GP model is via optimising the hyper-parameters of the covariance function by maximizing the likelihood function.

As this is a purely data-driven approach, basal transcription and degradation are not treated separately as in the ODE approach. Because the degradation of mRNA is usually modelled as a first order reaction, we include gene self-interaction in every putative network. This does not affect the total number of candidate topologies. Furthermore, as this approach is unable to distinguish alternative regulatory types (activation or repression) between genes so that the number of possible network topologies is reduced to $N \cdot \sum_{i=0}^{M} \binom{N-1}{i} = 55$ (with $M = 2$ and $N = 5$).

2.4 Model Selection and Edge Weighting

Following model optimisation, we obtain the final distance or likelihood of each gene with respect to their possible parents. For the ODE-based inference approach (Section 2.3.1) we

have,

$$\mathrm{BIC} = \ln(S) \cdot G + S \cdot \ln\left(\frac{\mathrm{dist}_{L_2}}{S}\right) \tag{11}$$

where S denotes the number of data points (sample size), G the number of free parameters and $dist_{L_2}$ the L_2 distance defined in equation 9. Alternatively, for the non-parametric inference approach (section 2.3.2) we obtain,

$$\mathrm{BIC} = \ln(S) \cdot G - 2 \cdot \ln\left(\mathcal{L}\left(\hat{\phi}_{MLE} \mid \mathbf{x}\right)\right) \tag{12}$$

S and G are defined as before and $\mathcal{L}\left(\hat{\phi}_{MLE} \mid \mathbf{x}\right)$ denotes the maximum likelihood of the model with optimised hyperparameters $\hat{\phi}_{MLE}$ given gene expression data $\mathbf{x}$. We use the BIC for weighting candidate models rather than the commonly used Akaike information criterion (AIC), as it is asymptotically valid for large sample sizes [30] whereas AIC tends to prefer overly complicated models in this case.

We then calculate the Schwarz weight [31] for each model $w_i\,(\mathrm{BIC})$ in the set of models j,

$$w_i\,(\mathrm{BIC}) = \frac{\exp\left(\frac{-\Delta_i(\mathrm{BIC})}{2}\right)}{\sum_j \exp\left(\frac{-\Delta_j(\mathrm{BIC})}{2}\right)} \tag{13}$$

such that $\sum_i w_i\,(\mathrm{BIC}) = 1$. $\Delta_i\,(\mathrm{BIC}) = \mathrm{BIC}_i - \mathrm{BIC}_{min}$ denotes the difference between the BIC of model i (BIC_i) and the lowest BIC across all models considered (BIC_{min}).

Once we have weighted all models across all genes in the network, we can calculate the weight w_e associated with every edge e in the GRN. This is done for each edge by summing the Schwarz weight of every model that contains the edge in question,

$$w_e = \sum_i w_i I_e\,(i) \tag{14}$$

where $I_e\,(i)$ denotes the indicator function which is 1 if edge e is present in model i and 0 otherwise.

2.5 Performance Evaluation

To evaluate the overall performance of the GRN inference, we use the BIC weights of every edge in the network to calculate the Area Under the Precision-Recall (AUPR) curve [32]. The detailed explanations and definitions of this AUPR approach are provided in the Appendix A.3.

Considering the sparsity of large GRN, we use the AUPR instead of the Area Under the Receiver Operating Characteristic (AUROC) curve [33] to evaluate performance.

3 Results

3.1 Deterministically Simulated Gene Expression Data

For the deterministically simulated gene expression data (see Section 2.1.1), we compare three main approaches to network inference (Table 1: 'Inference method'). All three methods are combined with gradient matching as described in Section 2.3. For each inference approach, we evaluate a range of different settings (Table 1) using the AUPR (as explained in Section 2.5). For the detailed model and parameter settings, please see Appendices A.1 and A.2. All data presented in this section represent the mean of five independent repeats.

Table 1: Employed settings for different network inference approaches.

Parameter	**Settings**
Inference method	Gaussian Process only (non-parametric), ODE constrained parameters, ODE unconstrained parameters
Input data	Non-oscillatory data (deterministic, 5 genes, 8 interactions), Oscillatory data (deterministic, 5 genes, 7 interactions), Realistic *in silico* data (stochastic, 10 genes, 10 interaction)
Data interpolation	Independent single-output GPs, Multiple-output GP
Number of datapoints	21, 41
Max. num. of parents	2, 3
Fixed GP length-scale (real. *in silico* data only)	50, 100, 150, 200

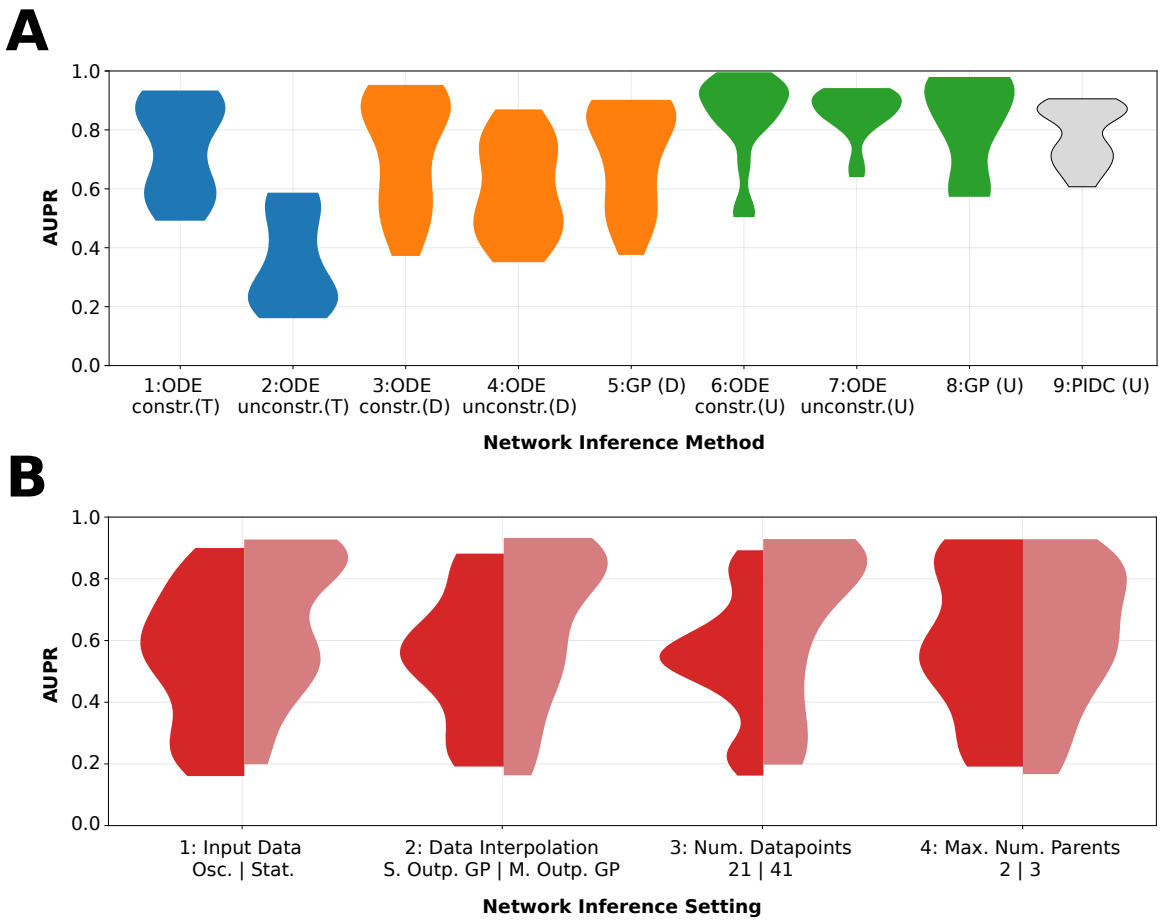

Figure 4: Performance comparison of network inference approaches using noise-free data. A: This subfigure displays the distribution of obtained performance (AUPR) for the three different classes of network inference methods, over all model settings listed in Table 1. There are four different network inference aims shown in four different shades. The blue distributions relate to the performance of the constrained and unconstrained ODE methods at inferring a directed GRN including information about interaction types (activation/repression) **(T)**. The orange distributions depict the performance of the two ODE-based methods and the GP-based method at predicting a directed GRN without type information **(D)**. The green distributions show the performance of the same three methods at inferring an undirected GRN **(U)**. The performance of a recently developed algorithm [8] based on partial information decomposition for the same settings and data is shown as the last distribution in grey ("PIDC"). *Baseline (random) performance is 0.2 for plots 1-2, 0.4 for plots 3-5, 0.7 for plots 6-9.* **B:** This subfigure shows the impact of different settings choices on network inference performance. Summing the two halves of each of the four asymmetric distributions in the figure gives rise to the same distribution of model performance (constituted by the three approaches discussed earlier - represented distributions 1, 2 and 5 in A).

3.1.1 Comparing Parametric and Non-Parametric Inference

Figure 4A contrasts the performance the three inference approaches across all settings and for three different inference aims, respectively. Only the parametric ODE-based methods allow for distinction between activating and repressing regulatory interactions between genes. From Figure 4A, we can however clearly see that this type of inference is successful only if the detailed kinetic information about the GRN is available prior to inference: the unconstrained ODE-based modelling of interactions shows a significant drop in performance over the tested settings compared to the constrained approach where basal transcription and degradation rates are known and ODE parameter ranges can be constrained *a priori* (see Table 2 in Appendix A.1 for parameters).

If we are only interested in the directionality of interactions and not their specific type, the three orange distributions in Figure 4A show that constraining the parameters of the ODE-based approach (and assuming known basal transcription and degradation rate) is no longer important for achieving good inference performance. The GP-based approach achieves on average higher performance on the simulated datasets used here. This is surprising, since gene interactions used in generating the data are of the same functional form assumed in the ODE inference.

The same trend (with slightly higher overall performance) can be seen when we are only predicting undirected edges. Interestingly, despite higher overall performance, constraining the ODE parameters can lead to worse performance under certain inference settings for this task (compare plot 6 and 7 in Figure 4A). All three approaches generally perform better on this simple noise-free five-gene networks than the PIDC approach [8].

Below, we analyse the impact of individual factors on inference quality, such as the data interpolation method, on the overall performance of the discussed methods.

3.1.2 Input Data

The distributions separated by the two input data types (plot 1, Figure 4B) show a slight performance increase for the non-oscillatory dataset over the oscillatory one. This should

be taken into account during experimental design in order to produce data which bears maximum information about the underlying regulatory network [34].

3.1.3 Data Interpolation

Despite the deterministic nature of the data we use for evaluation in this section, we find a pronounced difference in performance depending on the method used for interpolating the input data. By taking into account the correlation between the different gene expression time-courses, interpolation with a multiple output GP is able to achieve significantly better results compared to using independent GPs.

When interpolating oscillatory data using single output GPs, we observe that for low number of data points, the GP hyperparameters are optimised so that the oscillatory behaviour is no longer traced by the GP mean, but rather interpreted as noise (figure 14A in Appendix C). This was also observed in previous work [11]. As shown in Figure 14B (appendix C)this problem can be overcome by using multiple output GP regression, where the oscillatory behaviour correctly traced because trajectories of all genes are taken into account when optimising hyperparameters [35, 36].

3.1.4 Number of Data Points

Plot 3 of Figure 4B demonstrates increased performance as more time points are used. While this is unsurprising for noise-free data, we will re-evaluate this observation for stochastic data in Section 3.2.2.

3.1.5 Maximum Number of Parents Considered

In figure 4B we can see that the maximum number of parents considered per gene does not markedly affect performance. From this we conclude that for deterministic data, these network inference methods may be robust even for large putative network spaces. The main reason for this robustness is the weighting of the models using the BIC as this metric penalises model complexity explicitly.

3.2 Noisy *in silico* Gene Expression Data

Gene expression is a stochastic process and we apply the same inference procedures to stochastically simulated *in silico* gene expression data (but for 10 instead of 5 genes; see Section 2.1.2 for details).

3.2.1 Comparing Parametric and Non-Parametric Inference

The most notable difference between the results for the noise-free and noisy gene expression data is the absolute decline in performance, which is not unexpected. Despite this difference, we nevertheless observe similar trends as for the noise-free data. The unconstrained ODE-based modelling (plot 2, Figure 5A) again provides comparable performing result to the non-parametric GP-only modelling approach (plot 3, Figure 5A) when interaction types are not of interest.

When trying to infer only the existence of (undirected) edges between genes, we observe that the unconstrained ODE-based model performs slightly better than the GP-based approach; and both approaches perform better than PIDC.

The pronounced narrowing of distributions towards higher AUPR across different approaches indicates that unlike inference based on noise-free data, both ODE and GP-based methods only produce meaningful results (i.e. significantly better than random performance) for a very narrow range of scenarios.

3.2.2 Model Settings

Contrasting the performance for noise-less and noisy data shows not just lower absolute performance for each method for noisy data, but also different trends of their behaviour (Figure 5).

Interestingly, we can see from plot 1 of Figure 5B that in case of stochastic data, all well-performing inference approaches use single output GP interpolation of the data. This could be explained by the large number of free parameters in multiple output GP optimisation. For

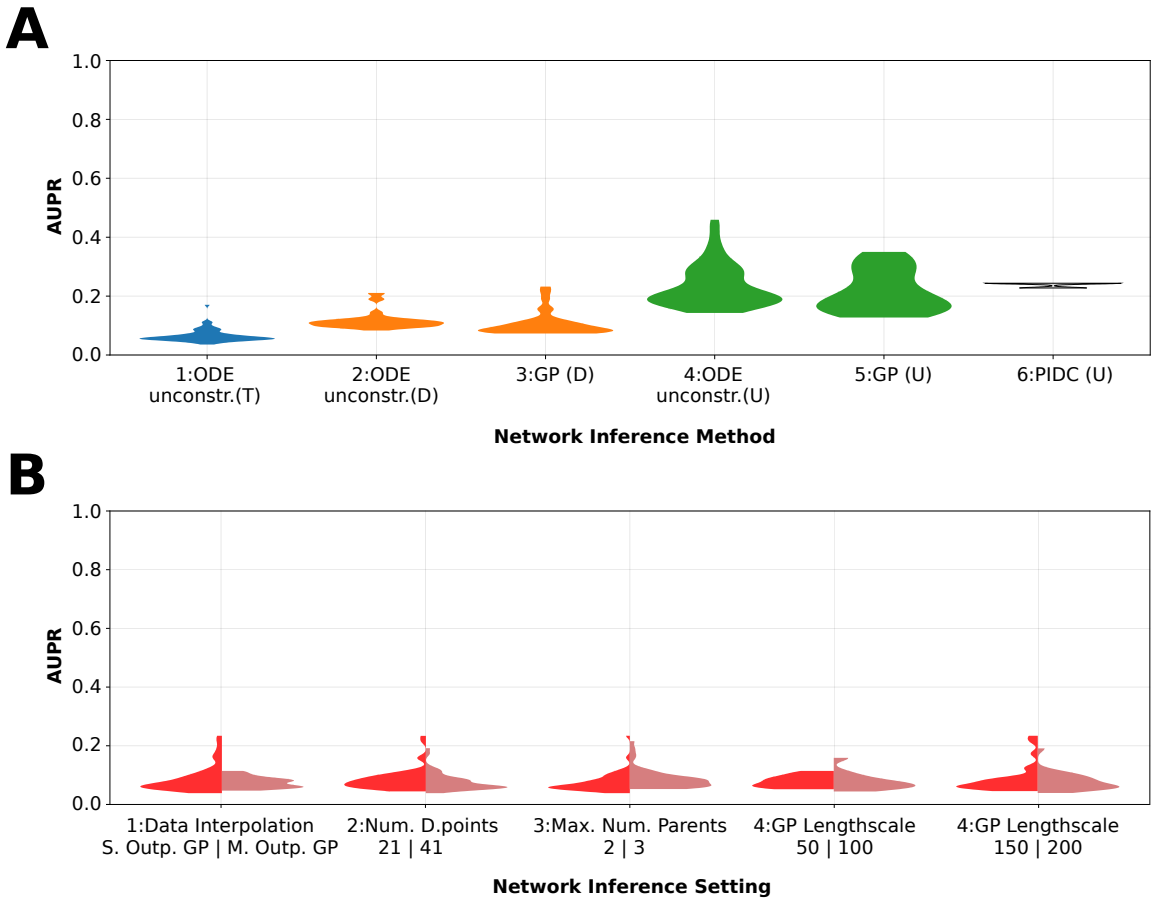

Figure 5: Performance comparison of network inference approaches using realistic simulated *in silico* data. A: This subfigure displays the distribution of obtained performance (AUPR) for the three different classes of network inference methods, over all model settings listed in Table 1. There are four different network inference aims shown in four different shades. The blue distribution relates to the performance of the unconstrained ODE method at inferring a directed GRN including information about interaction types (activation/repression) **(T)**. The orange distributions depict the performance of the ODE-based method and the GP-based method at predicting a directed GRN without type information **(D)**. The green distributions show the performance of the same three methods at inferring an undirected GRN **(U)**. The performance of a recently developed algorithm [8] based on partial information decomposition for the same settings and data is shown as the last distribution in grey ("PIDC"). Baseline (random) performance is 0.06 for plot 1, 0.1 for plots 2-3, 0.2 for plots 4-6. **B:** This subfigure shows the impact of different settings choices on network inference performance. Summing the two halves of each of the first three asymmetric distributions in the figure gives rise to the same distribution of model performance (constituted by the two main approaches discussed earlier (GP and ODE) - represented by distributions 1 and 5 in Figure 5A). The same is true for the sum of the last two distributions in the figure.

a ten-gene network, moving from ten independent single output GPs to one 10-output GP means solving a 32-parameter optimisation problem (31 for fixed length-scale) in contrast to solving ten 3-parameter problems. As finding the optimal solution in such a high-dimensional parameter space is extremely difficult, this may be the leading cause for this observation. We further substantiated this by interpolating gene expression data from a smaller GRN using single- and multiple output GP regression and comparing network inference results. An additional reason for the reduced performance could be the limitation to a single length-scale hyperparameter for multiple output GP, while single output GPs can have a different length-scale and variance for every gene they fit. This allows for more flexibility during interpolation. Multiple-output GP methods which allow for varying length-scales are available [37], however, but this further increases the number of free hyperparameters to be optimised.

We also see from plot 2 of Figure 5B that increasing the number of data points taken from the interpolated data no longer improves performance. While this might seem counter-intuitive at first, the inability of the GP to interpolate the true underlying gene expression dynamics renders the benefit of more data points futile; it appears that GPs can overfit the noise in the data (unless the GP hyperparameters are specifically constrained); using fewer time points can partially compensate for such overfitting. On closer inspection we find that this effect is particularly pronounced for the derivatives obtained from the GPs that play a major role in the inference.

Again changing the maximum number of parents allowed for a gene appears to have no effect (plot 3, Figure 5B). The rightmost two plots of Figure 5B show clear evidence for the importance of the right choice of length-scale during data interpolation (only at a length-scale of 150 can an inference performance of AUPR > 0.2 be achieved for this example).

4 Discussion

In this work, we compare the performance of different network inference methods, especially parametric and non-parametric gradient matching methods, under different settings and scenarios in order to gain an understanding of the strengths, weaknesses and impact of

different modelling choices.

When inferring GRNs from limited and inherently noisy gene expression data, there are usually a large number of potential models that can match the data [18]. By computing weights for each model and consequently each interaction in the network, we are able to obtain useful inferences by pooling over different methods.

We find that the simple non-parametric inference approach achieves slightly lower performance than the unconstrained ODE method despite the absence of mechanistic knowledge about the underlying regulatory processes. It was however shown in previous studies, that a more advanced non-parametric approach which combines Bayesian linear regression and GPs is able to achieve higher performance [14] assuming that some of the parameters are known. In our work, we show that knowledge of such parameters prior to network inference can strongly increase performance and even allows us to infer mechanistic aspects of interactions from data.

When inferring networks from gene expression data, the ability of the GP to reconstruct the underlying time-courses from noisy data is a critical factor. Especially the gradient obtained from the GP for the gradient matching procedure is particularly sensitive to poor fits. In order to alleviate this, previous work [11, 38] has suggested employing adaptive GPs which can improve performance by taking into account the structure of the ODE model (in case of parametric modelling) during GP fitting.

We believe that this approach is still worth pursuing further. Another promising avenue we see for future work is the combination of parametric and non-parametric methods. A possible approach would be to use the computationally cheaper GP-based method to sufficiently narrow the space of possible networks. We could then use ODE-based network inference to confirm interactions as well as obtain mechanistic information for the predicted edges in the GRN. If the space of putative networks is small enough following the non-parametric step, we could even avoid decoupling the network which would further increase inference performance.

5 Conclusion

In this work, we have carried out a comprehensive comparison of a range of parametric and non-parametric gradient-matching-based approaches on gene regulatory network inference from gene expression data.

We found that applying parametric ODE-based approaches on deterministic gene expression data showed that mechanistic information (such as the type of interaction) can be recovered during inference if enough knowledge about the network (e.g. parameter ranges) is present. For directed and undirected network inference, parametric ODE method can provide comparable or even better inference performance compared to non-parametric GP-based method, the latter approach however requires little mechanistic or kinetic regulatory information and computationally more efficient, which can be crucial for large-scale network inference problems. When applying to larger network or stochastic data, overall lower inference performance is observed for all methods, while consistent comparable performance between parametric and non-parametric methods is still obtained.

Several promising avenues to improve inference performance emerge from this analysis: in particular there is potential for the use of multiple output Gaussian Processes for data interpolation in cases of small networks. When applying the same methods to more complex stochastic networks these may, however, become less reliable.

A central result has been that Bayesian model averaging has real potential to increase the quality of network inference. We believe that combining the strengths of several existing approaches will ultimately be required to make significant further progress in solving this challenging problem.

References

[1] T. I. Lee, N. J. Rinaldi, F. Robert, D. T. Odom, Z. Bar-Joseph, G. K. Gerber, N. M. Hannett, C. T. Harbison, C. M. Thompson, I. Simon, J. Zeitlinger, E. G. Jennings, H. L. Murray, D. B. Gordon, B. Ren, J. J. Wyrick, J.-B. Tagne, T. L. Volkert, E. Fraenkel, D. K. Gifford, and R. A. Young, "Transcriptional regulatory networks in saccharomyces cerevisiae," *Science*, vol. 298, no. 5594, pp. 799–804, 2002.

[2] D. J. Kvitek and G. Sherlock, "Whole genome, whole population sequencing reveals that loss of signaling networks is the major adaptive strategy in a constant environment," *PLoS Genetics*, vol. 9, no. 11, p. e1003972, 2013.

[3] B. Göttgens, "Regulatory network control of blood stem cells," *Blood*, vol. 125, no. 17, pp. 2614–2620, 2015.

[4] N. Moris, C. Pina, and A. M. Arias, "Transition states and cell fate decisions in epigenetic landscapes," *Nature Reviews Genetics*, vol. 17, no. 11, pp. 693–703, 2016.

[5] F. Jacob and J. Monod, "Genetic regulatory mechanisms in the synthesis of proteins," *Journal of Molecular Biology*, vol. 3, no. 3, pp. 318–356, 1961.

[6] T. Thorne, P. Fratta, M. G. Hanna, A. Cortese, V. Plagnol, E. M. Fisher, and M. P. H. Stumpf, "Graphical modelling of molecular networks underlying sporadic inclusion body myositis," *Molecular BioSystems*, vol. 9, no. 7, pp. 1736–1742, 2013.

[7] C. Siegenthaler and R. Gunawan, "Assessment of network inference methods: How to cope with an underdetermined problem," *PLoS ONE*, vol. 9, no. 3, p. e90481, 2014.

[8] T. E. Chan, M. P. H. Stumpf, and A. C. Babtie, "Network inference from single-cell data using multivariate information measures," *bioRxiv*, p. 082099, 2017.

[9] C. A. Penfold and D. L. Wild, "How to infer gene networks from expression profiles, revisited," *Interface Focus*, vol. 1, no. 6, pp. 857–870, 2011.

[10] T. Äijö and R. Bonneau, "Biophysically motivated regulatory network inference: Progress and prospects," *Human Heredity*, no. 81, pp. 62–77, 2016.

[11] A. C. Babtie, P. Kirk, and M. P. H. Stumpf, "Biological network inference using gaussian

process regression," in *MSc Bioinformatics and Theoretical Systems Biology Project Reports*, Imperial College London, 2013.

[12] M. Sunnåker, E. Zamora-Sillero, R. Dechant, C. Ludwig, A. G. Busetto, A. Wagner, and J. Stelling, "Automatic generation of predictive dynamic models reveals nuclear phosphorylation as the key msn2 control mechanism," *Science Signaling*, vol. 6, no. 277, p. ra41, 2013.

[13] M. Barenco, D. Tomescu, D. Brewer, R. Callard, J. Stark, and M. Hubank, "Ranked prediction of p53 targets using hidden variable dynamic modeling," *Genome biology*, vol. 7, no. 3, p. R25, 2006.

[14] T. Äijö and H. Lähdesmäki, "Learning gene regulatory networks from gene expression measurements using non-parametric molecular kinetics," *Bioinformatics*, vol. 25, no. 22, pp. 2937–2944, 2009.

[15] B. Calderhead, M. Girolami, and N. D. Lawrence, "Accelerating bayesian inference over nonlinear differential equations with gaussian processes," in *Advances in neural information processing systems*, pp. 217–224, 2009.

[16] M. Brown, F. He, C. Zhan, and L. F. Yeung, "Nonparametric collocation ode parameter estimation: application in biochemical pathway modelling," in *UKACC international conference on control*, 2008.

[17] H. Liang and H. Wu, "Parameter estimation for differential equation models using a framework of measurement error in regression models," *Journal of the American Statistical Association*, vol. 103, no. 484, pp. 1570–1583, 2008.

[18] A. C. Babtie, P. Kirk, and M. P. H. Stumpf, "Topological sensitivity analysis for systems biology," *Proceedings of the National Academy of Sciences*, vol. 111, no. 52, pp. 18507–18512, 2014.

[19] I. Cantone, L. Marucci, F. Iorio, M. A. Ricci, V. Belcastro, M. Bansal, S. Santini, M. di Bernardo, D. di Bernardo, and M. P. Cosma, "A yeast synthetic network for in vivo assessment of reverse-engineering and modeling approaches," *Cell*, vol. 137, no. 1, pp. 172 – 181, 2009.

[20] J. Mazur, D. Ritter, G. Reinelt, and L. Kaderali, "Reconstructing nonlinear dynamic models of gene regulation using stochastic sampling," *BMC Bioinformatics*, vol. 10, p. 448, 2009.

[21] T. Schaffter, D. Marbach, and D. Floreano, "Genenetweaver: in silico benchmark generation and performance profiling of network inference methods," *Bioinformatics*, vol. 27, no. 16, pp. 2263–2270, 2011.

[22] D. Marbach, J. C. Costello, R. Kuffner, N. M. Vega, R. J. Prill, D. M. Camacho, K. R. Allison, M. Kellis, J. J. Collins, and G. Stolovitzky, "Wisdom of crowds for robust gene network inference," *Nature Methods*, vol. 9, no. 8, pp. 796–804, 2012.

[23] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*, vol. 1. MIT press Cambridge, 2006.

[24] P. Kirk, *Inferential stability in systems biology*. PhD thesis, Imperial College London, 2011.

[25] M. Ebden, "Gaussian processes: A quick introduction," *arXiv preprint*, p. 1505.02965, 2008.

[26] E. Solak, R. Murray-Smith, W. E. Leithead, D. J. Leith, and C. E. Rasmussen, "Derivative observations in gaussian process models of dynamic systems," in *Advances in neural information processing systems*, pp. 1057–1064, 2003.

[27] P. Goovaerts, *Geostatistics For Natural Resources Evaluation*. Oxford University Press, 1997.

[28] H. Wackernagel, *Multivariate Geostatistics: an Introduction with Applications*, vol. 3. Springer-Verlag Berlin Heidelberg, 2003.

[29] M. A. Alvarez, L. Rosasco, N. D. Lawrence, *et al.*, "Kernels for vector-valued functions: A review," *Foundations and Trends® in Machine Learning*, vol. 4, no. 3, pp. 195–266, 2012.

[30] P. Kirk, T. Thorne, and M. P. H. Stumpf, "Model selection in systems and synthetic biology," *Current Opinion in Biotechnology*, vol. 24, no. 4, pp. 767 – 774, 2013.

[31] G. Schwarz, "Estimating the dimension of a model," *The annals of statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[32] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *Proceedings of the 23rd international conference on Machine learning*, pp. 233–240, ACM, 2006.

[33] T. Fawcett, "An introduction to roc analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

[34] J. Liepe, S. Filippi, M. Komorowski, and M. P. H. Stumpf, "Maximizing the information content of experiments in systems biology," *PLoS Computational Biology*, vol. 9, no. 1, p. e1002888, 2013.

[35] J. Zurauskiene, P. D. W. Kirk, T. W. Thorne, and M. P. H. Stumpf, "Bayesian non-parametric approaches to reconstructing oscillatory systems and the Nyquist limit," *Physica A: Statistical Mechanics and its Applications*, vol. 407, pp. 33–42, 2014.

[36] J. Zurauskiene, P. Kirk, P. D. W. Kirk, T. W. Thorne, J. Pinney, M. P. H. Stumpf, and M. P. H. Stumpf, "Derivative processes for modelling metabolic fluxes.," *Bioinformatics (Oxford, England)*, vol. 30, pp. 1892–1898, July 2014.

[37] N. D. Lawrence, "Fitting covariance and multioutput gaussian processes," in *Gaussian Process Summer School*, 2015. URL: `http://gpss.cc/gpss15/talks/gp_gpss15_session3.pdf`.

[38] B. Macdonald and D. Husmeier, "Computational inference in systems biology," in *Bioinformatics and Biomedical Engineering: International Conference*, pp. 276–288, 2015.

[39] P. Meyer, T. Cokelaer, D. Chandran, K. H. Kim, P.-R. Loh, G. Tucker, M. Lipson, B. Berger, C. Kreutz, A. Raue, B. Steiert, J. Timmer, E. Bilal, H. M. Sauro, G. Stolovitzky, and J. Saez-Rodriguez, "Network topology and parameter estimation: from experimental design methods to gene regulatory network kinetics using a community based approach," *BMC Systems Biology*, vol. 8, p. 13, 2014.

[40] A. V. Hill, "The possible effects of the aggregation of the molecules of haemoglobin on its dissociation curves," *Journal of Physiology*, vol. 40, pp. 4–7, 1910.

[41] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A fresh approach to numerical computing," *SIAM Review*, vol. 59, no. 1, pp. 65–98, 2017.

[42] C. Rackauckas and Q. Nie, "DifferentialEquations.jl – a performant and feature-rich ecosystem for solving differential equations in julia," *Journal of Open Research Software*, vol. 5, no. 1, p. 15, 2017.

[43] J. A. Nelder and R. Mead, "A simplex method for function minimization," *The Computer Journal*, vol. 7, no. 4, pp. 308–313, 1965.

A Appendix: Methodology - Further Details

A.1 Parameters, Functional Forms and Data for ODE Model Simulation

For simulation of the non-oscillatory data, we model the expression of each gene in the network shown in (Figure 3A) as ODEs of the form (as used in the DREAM7 (Dialogue on Reverse Engineering Assessment and Methods) challenge [39]):

$$\dot{x}_n(t) = s_n - \gamma_n \cdot x_n + \beta_n \cdot f_n^+(\{\mathbf{x}_q(t) \mid q \in \mathcal{P}_n^+\}, \boldsymbol{\theta}_n^+, \mathbf{m}_n^+) \cdot f_n^-(\{\mathbf{x}_q(t) \mid q \in \mathcal{P}_n^-\}, \boldsymbol{\theta}_n^-, \mathbf{m}_n^-) \quad (15)$$

with the following two expressions defining the activating (f_n^+) and repressing (f_n^-) regulation of gene n respectively:

$$f_n^+(\mathbf{x}_n(t), \boldsymbol{\theta}_n^+, \mathbf{m}_n^+) = \sum_{q \in \mathcal{P}_n^+} \left(\frac{x_q(t)^{m_{nq}}}{\theta_{nq}^{m_{nq}} + x_q(t)^{m_{nq}}} \right) \quad (16)$$

$$f_n^-(\mathbf{x}_n(t), \boldsymbol{\theta}_n^-, \mathbf{m}_n^-) = \prod_{q \in \mathcal{P}_n^-} \left(\frac{1}{1 + \left(\frac{x_q(t)}{\theta_{nq}} \right)^{m_{nq}}} \right) \quad (17)$$

Here, $\dot{x}_n(t)$ denotes the rate of change of the mRNA concentration at time t, $x_n(t)$ the mRNA concentration, s_n the basal transcription rate, γ_n the mRNA degradation rate, β_n the strength of gene regulation, all with respect to gene $n \in [1, \ldots, i]$ where i denotes the number of genes in the network. $\mathcal{P}_n^+$ and $\mathcal{P}_n^-$ denote the subsets of parent genes which have an activating or repressing effect on the expression of gene n respectively. θ_{nq} and m_{nq} are the commonly used hill parameters for the regulation of gene n by its parent q [40].

For simulation of the oscillatory data, we model the expression of each gene in the network shown in (Figure 3B) as ODEs of the form (parameter notation as in equation 15):

$$\dot{x}_n(t) = s_n - \gamma_n \cdot x_n + \sum_{q \in \mathcal{P}_n} \beta_{nq} \cdot f_{nq}(\mathbf{x}_q(t), \boldsymbol{\theta}_{nq}, \mathbf{m}_{nq}) \quad (18)$$

with the following two terms representing the interaction term f_{nq} of the parent $q \in \mathcal{P}_n$ with the gene n.

In case of an activating interaction between the parent gene $q \in \mathcal{P}_n$ and gene n:

$$f_n(\mathbf{x}_n(t), \boldsymbol{\theta}_n, \mathbf{m}_n) = \left(\frac{x_q(t)^{m_{nq}}}{\theta_{nq}^{m_{nq}} + x_q(t)^{m_{nq}}} \right) \tag{19}$$

In case of an repressing interaction between the parent gene $q \in \mathcal{P}_n$ and gene n:

$$f_n(\mathbf{x}_n(t), \boldsymbol{\theta}_n, \mathbf{m}_n) = \left(\frac{1}{1 + \left(\frac{x_q(t)}{\theta_{nq}} \right)^{m_{nq}}} \right) \tag{20}$$

Data simulated from the networks shown in Figure 3 was simulated over the timespan $(0, 20)$ with a time-step of either 0.5 or 1.0 for deterministic data and 25 or 50 for stochastic data. Starting concentrations for the five genes $[x_1(0), \ldots, x_5(0)]$ were set as $[1.0, 0.5, 1.0, 0.5, 0.5]$ in all cases.

The following ODE systems were used for simulation of the oscillatory and non-oscillatory noise-free data:

Non-oscillatory data

$$\begin{aligned}
\dot{x}_1(t) &= 0.1 - 0.4 \cdot x_1(t) + 2.0 \cdot \frac{x_5(t)^2}{1.5^2 + x_5(t)^2} \\
\dot{x}_2(t) &= 0.2 - 0.4 \cdot x_2(t) + 1.5 \cdot \frac{x_1(t)^2}{1.5^2 + x_1(t)^2} \cdot \left(1 + \left(\frac{x_5(t)}{2} \right)^1 \right)^{-1} \\
\dot{x}_3(t) &= 0.2 - 0.4 \cdot x_3(t) + 2.0 \cdot \frac{x_1(t)^2}{1.5^2 + x_1(t)^2} \\
\dot{x}_4(t) &= 0.4 - 0.1 \cdot x_4(t) + 1.5 \cdot \frac{x_1(t)^2}{1.5^2 + x_1(t)^2} \cdot \left(1 + \left(\frac{x_3(t)}{1} \right)^2 \right)^{-1} \\
\dot{x}_5(t) &= 0.3 - 0.3 \cdot x_5(t) + 2.0 \cdot \frac{x_4(t)^2}{1.0^2 + x_4(t)^2} \cdot \left(1 + \left(\frac{x_2(t)}{0.5} \right)^3 \right)^{-1}
\end{aligned}$$

Oscillatory data

$$\dot{x}_1(t) = 0.2 - 0.9 \cdot x_1(t) + 2 \cdot \frac{x_5(t)^5}{1.5^5 + x_5(t)^5}$$

$$\dot{x}_2(t) = 0.2 - 0.9 \cdot x_2(t) + 2 \cdot \frac{x_1(t)^5}{1.5^5 + x_1(t)^5}$$

$$\dot{x}_3(t) = 0.2 - 0.7 \cdot x_3(t) + 2 \cdot \frac{x_1(t)^5}{1.5^5 + x_1(t)^5}$$

$$\dot{x}_4(t) = 0.2 - 1.5 \cdot x_4(t) + 2 \cdot \frac{x_1(t)^5}{1.5^5 + x_1(t)^5} + 2 \cdot \left(1 + \left(\frac{x_3(t)}{1.5}\right)^5\right)^{-1}$$

$$\dot{x}_5(t) = 0.2 - 1.5 \cdot x_5(t) + 2 \cdot \frac{x_4(t)^5}{1.5^5 + x_4(t)^5} + 2 \cdot \left(1 + \left(\frac{x_2(t)}{1.5}\right)^3\right)^{-1}$$

Raw Gene-Expression Time-Course Data Plots

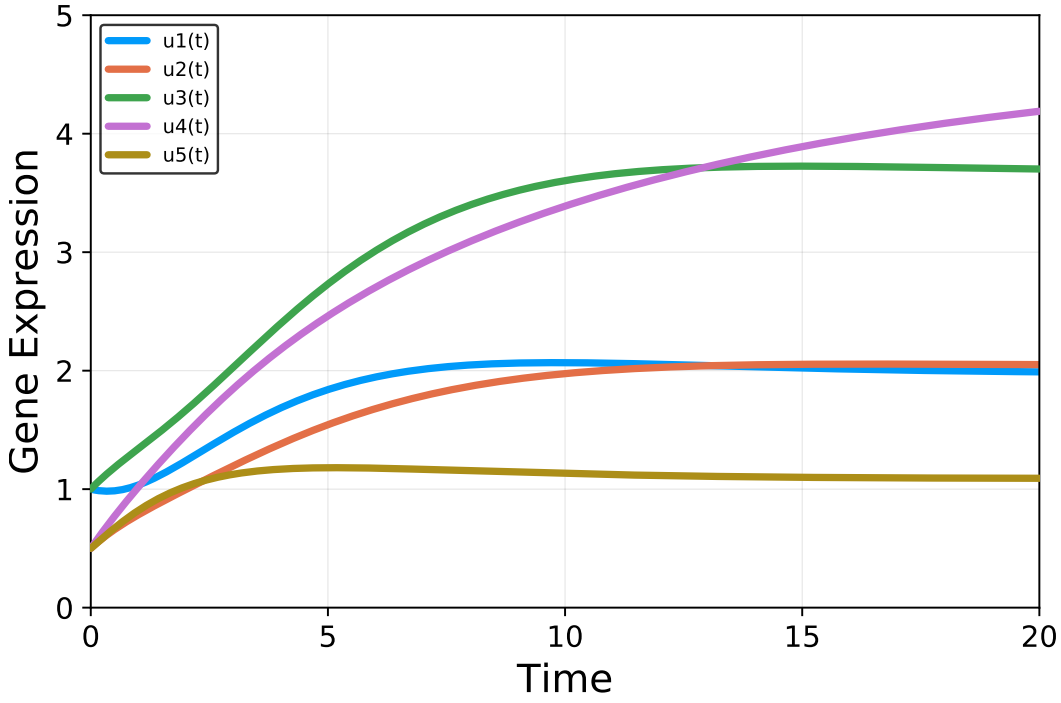

Figure 6: Solution to ODE system used for simulation of the non-oscillatory time-course data.

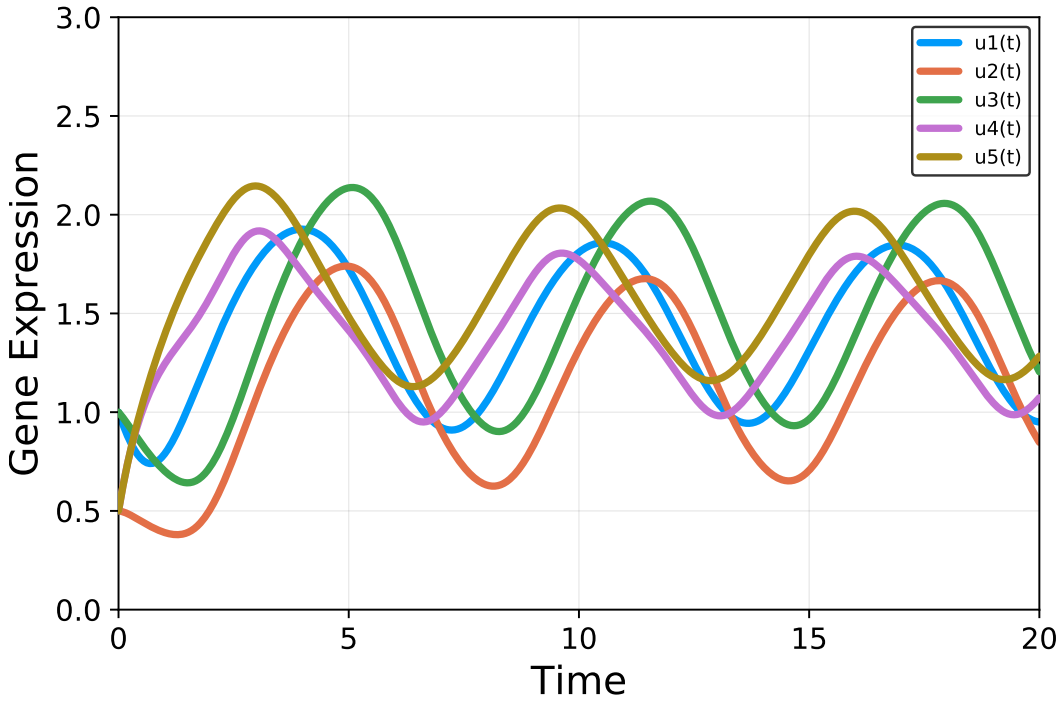

Figure 7: Solution to ODE system used for simulation of the oscillatory time-course data.

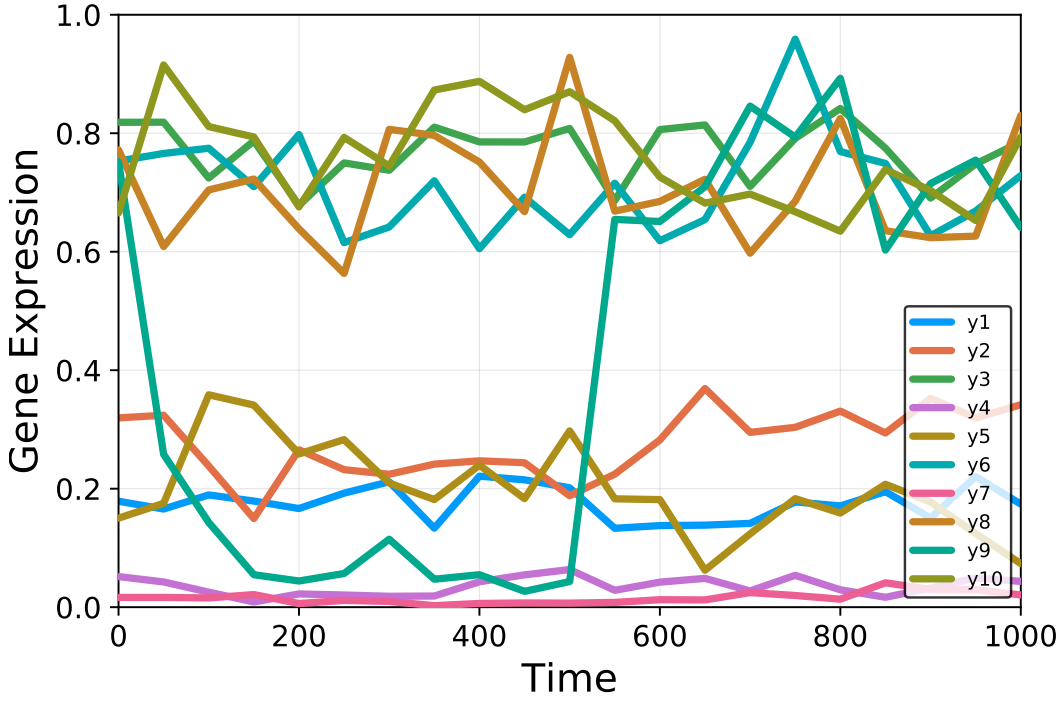

Figure 8: Realistic *in silico* time-course data simulated by `GeneNetWeaver`.

A.2 Parameters for Model Optimisation

Parameter optimisation of each putative decoupled model was done using either known bounds and starting values for each parameter or completely unconstrained optimisation. For constrained optimisation, basal transcription and degradation rates were assumed to be known, while they were included as free parameters for the unconstrained optimisation.

For the guidelines used to simulate the non-oscillatory data (equations 15, 16, 17 in Appendix A.1 and Figure 3A), this results in either $3 + 2 \cdot r$ or $1 + 2 \cdot r$ free parameters to be optimised for each decoupled gene, depending on whether basal transcription and degradation rates are known and where r is the number of parents for each gene. For the guidelines used to simulate the oscillatory data (equations 18, 19, 20 in Appendix A.1 and figure 3B), the number of free parameters per decoupled ODE system is $2 + 3 \cdot r$ or $3 \cdot r$ respectively.

Parameter constraints and starting values for the remaining parameters optimised during constrained optimisation are displayed in Table 2. In case of unconstrained ODE optimisation, all free parameters were initialised at 1.0.

For non-parametric network inference, models were optimised using the likelihood (equation 3) as the maximisation objective and starting values 1.0 for all free hyperparameters. Under certain circumstances (as mentioned in the main text) the length-scale hyperparameter of the GP was fixed at 50, 100, 150 or 200 before running the optimisation in order to achieve a satisfactory fit to the data.

Table 2: Constraints and and initial parameter values used for constrained optimisation of putative ODE models.

Data class	**Param.**	**Initial value**	**Lower bound**	**Upper b.**	**True value(s)**
'Stat. data'	β	1.0	0.0	5.0	2.0, 1.5
	m	2.0	0.1	5.0	1.0, 2.0, 3.0
	θ	1.0	0.0	4.0	0.5, 1.0, 1.5, 2.0
'Osc. data'	β	1.0	0.5	4.0	2.0
	m	1.0	0.7	5.0	5.0
	θ	1.0	0.2	3.0	1.5

A.3 Performance Evaluation using Precision-Recall curves

In this section, we explain and define how we evaluated performance of our inference method using precision-recall curves.

To obtain the AUPR, the vector of edge weights was first sorted. Each value in the vector was then iteratively chosen as the cut-off point between a positive and negative prediction (i.e. all larger weights are considered positive predictions and vice versa). It was therefore possible to calculate the precision and recall for each of these thresholds. A graph was plotted using the obtained precision (y-axis) and recall (x-axis) values. The area under the curve arising from this plot is the AUPR (see Figure 9).

The procedure for obtaining the AUROC is analogous, with the only difference that the false positive rate is used instead of the precision.

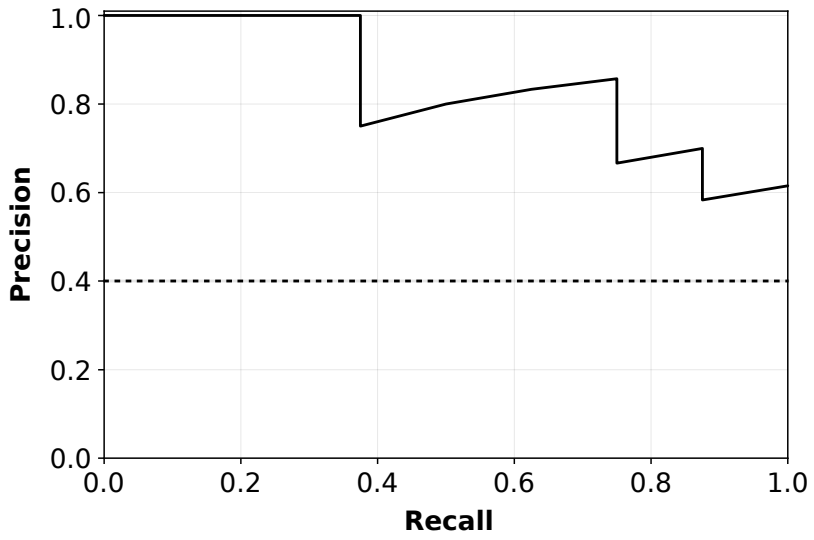

Figure 9: Example Precision-Recall curve. PR curve with AUPR of 0.84 with the baseline (random performance) shown as a dashed line.

Definitions

TP: True positive (a predicted edge which is actually present)

FP: False positive (a predicted edge which is not actually present)

TN: True Negative (an unpredicted edge which is actually not present)

FN: False Negative (an unpredicted edge which is actually present)

Precision (Positive Predictive Value) - the proportion of true edges that were predicted correctly:

$$\mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{21}$$

Recall (True Positive Rate) - the proportion of predicted edges that are actually present:

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{22}$$

False Positive Rate - the proportion of predicted edges that are not actually present:

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \tag{23}$$

B Appendix: Supplementary Figures for Settings Evaluation

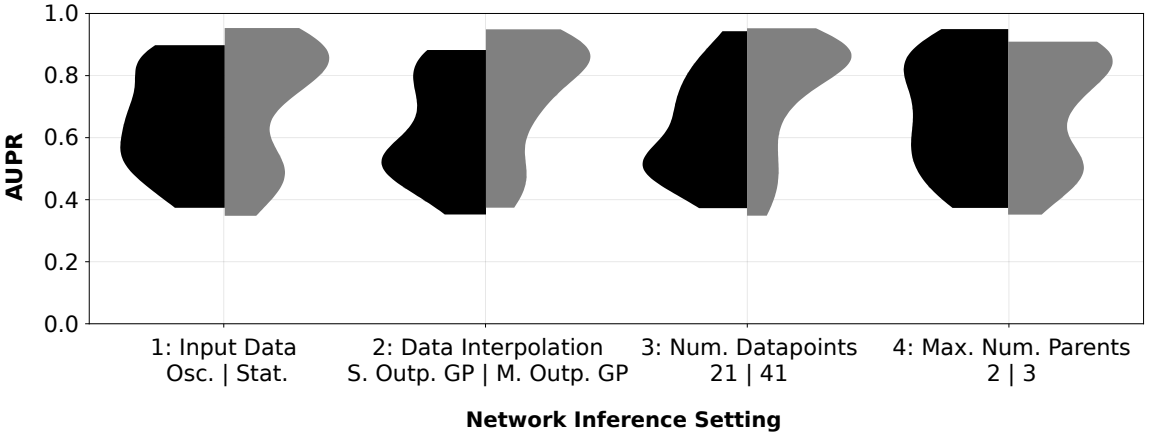

Figure 10: Performance of all evaluated inference approaches by model choices for noise-free data and inferring directed edges. This figure shows the impact of different settings choices on network inference performance. Summing the two halves of each of the four asymmetric distributions in the figure gives rise to the same distribution of model performance (constituted by distributions 1, 2 and 5 in Figure 4).

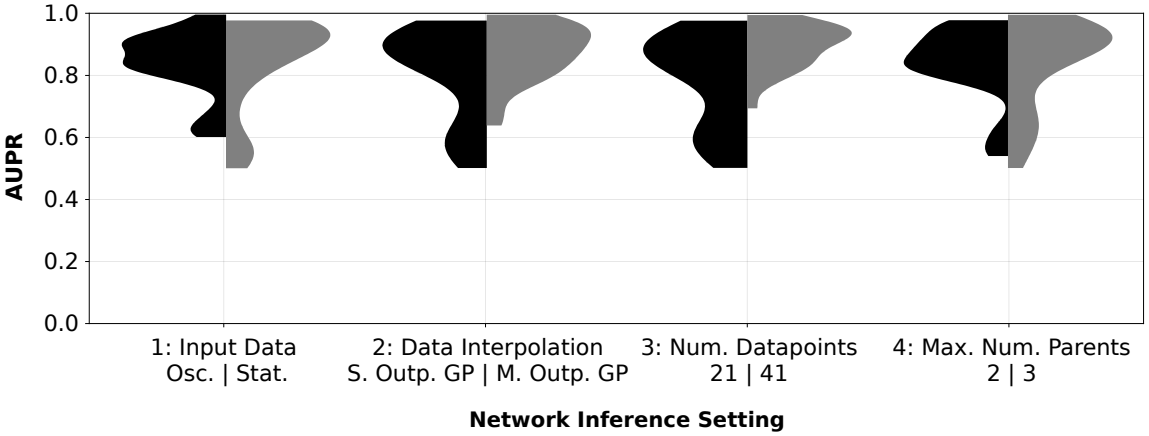

Figure 11: Performance of all evaluated inference approaches by model choices for noise-free data and inferring undirected edges. This figure shows the impact of different settings choices on network inference performance. Summing the two halves of each of the four asymmetric distributions in the figure gives rise to the same distribution of model performance (constituted by distributions 1, 2 and 5 in Figure 4).

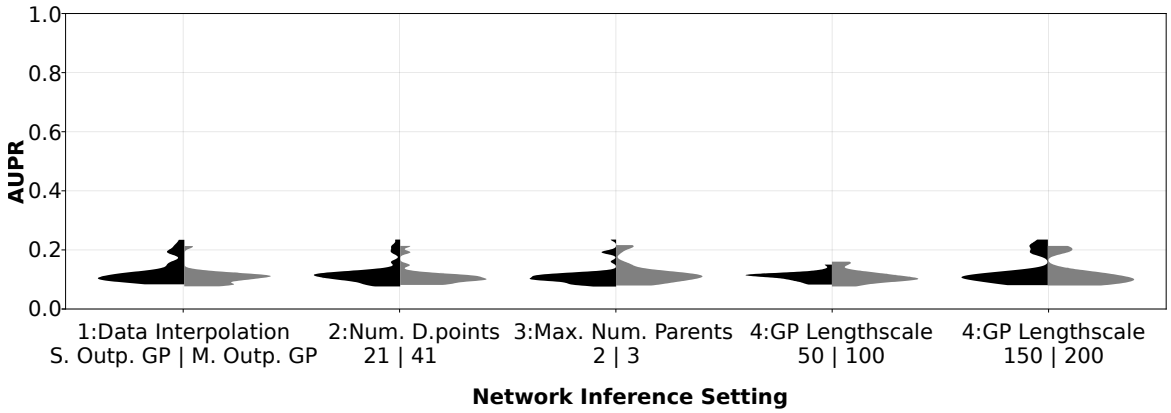

Figure 12: Performance of all evaluated inference approaches by model choices for stochastic data and inferring directed edges. This figure shows the impact of different settings choices on network inference performance. Summing the two halves of each of the first three asymmetric distributions in the figure gives rise to the same distribution of model performance (constituted by distributions 1 and 5 in Figure 5). The same is true for the sum of the last two distributions in the figure ("GP Lengthscale").

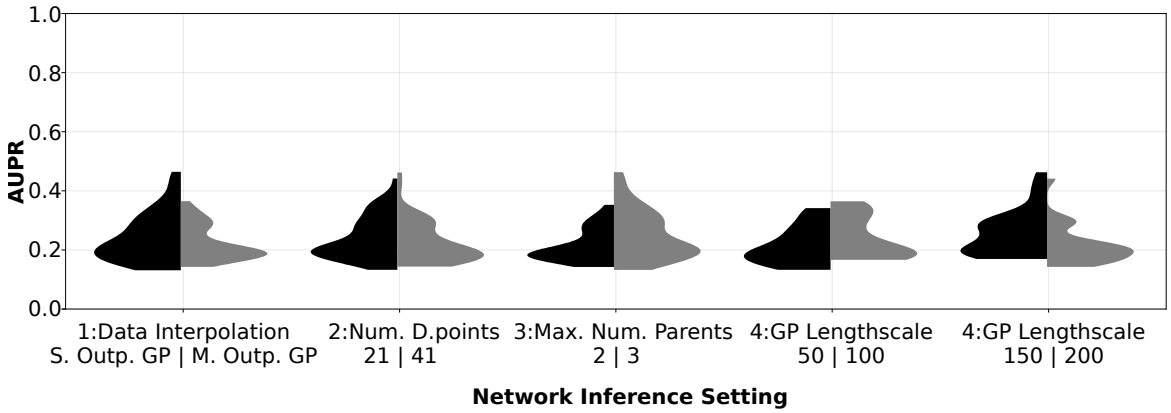

Figure 13: Performance of all evaluated inference approaches by model choices for stochastic data and inferring undirected edges. This figure shows the impact of different settings choices on network inference performance. Summing the two halves of each of the first three asymmetric distributions in the figure gives rise to the same distribution of model performance (constituted by distributions 1 and 5 in Figure 5). The same is true for the sum of the last two distributions in the figure ("GP Lengthscale").

C Appendix: Supplementary Figures for fitting GPs

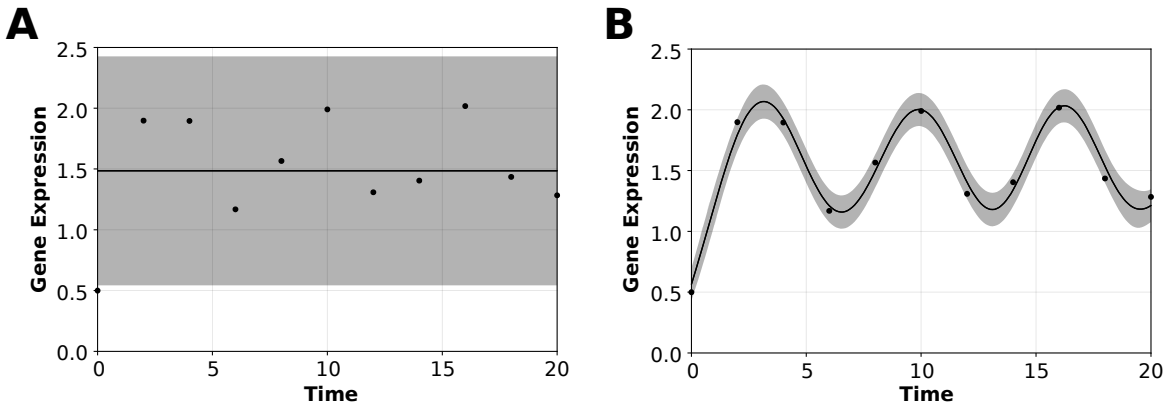

Figure 14: Comparison of single output GP and multiple output GP fits to data. Both plots show time-course data for the expression of gene 5 from the non-oscillatory (noise-free) data indicated by black dots. The black lines refer to the GP mean function with 95% confidence intervals shaded in grey. **A** Data interpolated using a single output GP. **B** Data interpolated using a multiple output GP, which takes into account correlations between the expression time-courses of all genes in the network.

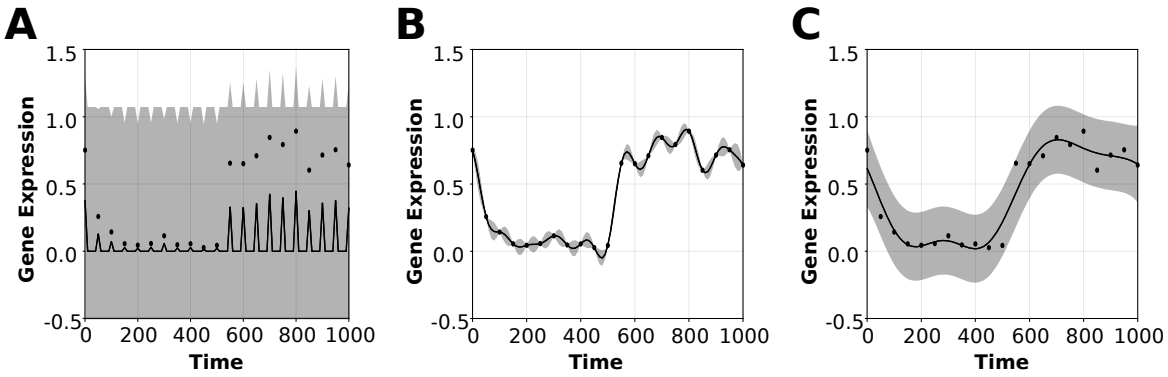

Figure 15: Comparison of GP fit with different length-scale hyperparameters. All three plots show the time-course data for the expression of gene 9 from the GNW dataset indicated by black dots. The black lines refer to the GP mean function with 95% confidence intervals shaded in grey. The three plots show different length-scale hyperparameter settings: **A** unconstrained hyperparameter optimisation, **B** length-scale fixed to 50, **C** length-scale fixed to 150.

D Appendix: Software Tools

All code for this project was written in the `Julia` programming language, version *0.6* [41].

D.1 Data Simulation

Simulations of the mRNA expression data described by the ODE systems shown is Section A.1 were carried out using the Runge-Kutta Order 4 (`RK4()`) solver from `Differential Equations.jl`[1] [42].

D.2 Gaussian Process Regression

Gaussian Process regression was done using a squared exponential kernel (`RBF()`) the `GPy`[2] `Python` package called in `Julia` through `PyCall.jl`[3]. GP hyperparameters were optimised using the `optimize_restarts` attribute of the model with 3, 5 or 15 restarts. In certain cases (stochastic *in silico* gene expression data), the length-scale parameter of the RBF kernel was fixed prior to optimisation.

D.3 Parameter Optimisation

ODE parameter optimisation was done using the Nelder Mead [43] (`NelderMead()`) algorithm from `Optim.jl`[4]. In case of constrained parameter optimisation, this was carried out using the `Fminbox{NelderMead}()` method from the same package.

GP hyperparameter optimisation for the non-parametric inference approach was done using `GPy` as described in the previous section.

[1] `https://github.com/JuliaDiffEq`
[2] `https://github.com/SheffieldML/GPy`
[3] `https://github.com/JuliaPy/PyCall.jl`
[4] `https://github.com/JuliaNLSolvers/Optim.jl`