Supplementary Information

C.J.Oates *et al.*

July 8, 2014

Contents

1 Truncated normal distributions 2

1.1 Definition 2

1.2 Sampling 2

2 A graphical model view 3

3 Markov chain Monte Carlo 3

4 Sensitivity to hyperparameter specification 6

5 ODE model of MAPK signaling for simulation 9

5.1 Dynamical system 9

5.2 Simulation regimes 9

5.3 Details of assessment 11

6 Implementation 12

6.1 LASSO 12

6.2 TSNI 12

6.3 DBN 12

6.4 TVDBN 12

6.5 GP 13

6.6 Computational times 13

7 *In silico* results 14

7.1 Network reconstruction 14

7.2 Parameter inference 14

7.3 The effect of nonlinearity 16

8 Prediction of signaling response 16

8.1 Data generation 17

8.2 Stationary benchmark 17

8.3 CheMA 17

8.4 Linear kinetics 17

8.5 Results 18

9 RPPA experimental protocol **18**

10 *In vitro* results **19**

1 Truncated normal distributions

We used truncated normal distributions as priors for kinetics parameters, as described in Main Text. Here, we describe truncated normal distributions and how we sampled from them.

1.1 Definition

A random variable $\boldsymbol{Y} \in \mathbb{R}^p$ has a truncated multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, denoted $\boldsymbol{Y} \sim \mathcal{N}_T(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, if $\boldsymbol{Y}$ has probability density function

$$p_{\boldsymbol{Y}}(\boldsymbol{y}) \propto \exp\left(-\frac{(\boldsymbol{y}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{y}-\boldsymbol{\mu})}{2}\right)\mathbb{I}(\boldsymbol{y} \geq \boldsymbol{0}). \tag{1}$$

where $\mathbb{I}$ here is the indicator function, such that $\mathbb{I}(\boldsymbol{y} \geq \boldsymbol{0}) = 1$ if $\boldsymbol{y} \geq \boldsymbol{0}$, otherwise $\mathbb{I}(\boldsymbol{y} \geq \boldsymbol{0}) = \boldsymbol{0}$. (The notation $\boldsymbol{y} \geq \boldsymbol{0}$ is taken to mean that $y_i \geq 0$ for all $i = 1, \ldots, p$.) The density $p_{\boldsymbol{Y}}$ is related to the standard normal probability density ϕ via $p_{\boldsymbol{Y}}(\boldsymbol{y}) = C^{-1}\phi(\boldsymbol{y})\mathbb{I}(\boldsymbol{y} \geq \boldsymbol{0})$, so evaluation of $p_{\boldsymbol{Y}}$ requires

$$C = \int_{\boldsymbol{y} \geq \boldsymbol{0}} \phi(\boldsymbol{y}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) d\boldsymbol{y} = \int_{\boldsymbol{z} \leq \boldsymbol{0}} \phi(\boldsymbol{z}; -\boldsymbol{\mu}, \boldsymbol{\Sigma}) d\boldsymbol{z} := \Phi(\boldsymbol{0}; -\boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{2}$$

where Φ is the normal cumulative distribution function.

1.2 Sampling

In general, sampling efficiently from truncated multivariate normal distributions is challenging. For example a rejection sampler based on an unconditioned normal density becomes inefficient when the measure of the target density's support is small. One approach is to construct a Gibbs sampler based on Eqn. 1 (see [9, 10]) but this is considerable effort for obtaining random samples for our purposes. However if the target distribution is non-degenerate (i.e. $\boldsymbol{\Sigma}$ is positive definite) then there exists a bijective mapping onto a product of standard truncated normal densities, which we exploit for sampling. Specifically, if $\boldsymbol{Y} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ then we can write $\boldsymbol{Y} = \boldsymbol{\mu} + \boldsymbol{A}\boldsymbol{Z}$ where $\boldsymbol{Z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ where $\boldsymbol{I}$ is the identity matrix and $\boldsymbol{A}$ arises from the Cholesky decomposition $\boldsymbol{\Sigma} = \boldsymbol{A}\boldsymbol{A}^T$. Positive definiteness ensures that the Cholesky decomposition exists and is unique. Moreover $\boldsymbol{A}$ is invertible, being lower triangular with strictly positive diagonal entries. Since $\boldsymbol{Y} \geq \boldsymbol{0}$ if and only if $\boldsymbol{Z} \geq -\boldsymbol{A}^{-1}\boldsymbol{\mu}$, we have the basis for efficient sampling (Algorithm 1). In the case that the target distribution approximates a point mass (this arises from conditioning on a rare event in the tails of a normal distribution), the algorithm uses numerical regularization.

Algorithm 1 Efficient sampling from the (non-degenerate) truncated multivariate normal $\boldsymbol{Y} \sim \mathcal{N}_T(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, with numerical regularization. Here U is the uniform distribution, p is the dimension of $\boldsymbol{Y}$ and ϵ is taken to be machine precision.

```
$\boldsymbol{A} \leftarrow$ Cholesky($\boldsymbol{\Sigma}$)
$\boldsymbol{b} \leftarrow -\boldsymbol{A}^{-1}\boldsymbol{\mu}$
for $i = 1$ to $p$ do
    $u \sim U[\Phi(b_i), 1]$
    if $u > 1 - \epsilon$ then
        $z_i \leftarrow b_i$
    else
        $z_i \leftarrow \Phi^{-1}(u)$
    end if
end for
$\boldsymbol{y} \leftarrow \boldsymbol{\mu} + \boldsymbol{A}\boldsymbol{z}$
```

2 A graphical model view

Figure 1(a) restates the reaction graph G as a probabilistic graphical model, where bounding boxes are used to indicate a set of variables. Nodes in the graphical model then correspond to the kinase and kinase inhibitor sets $\mathcal{E}_i$, $\mathcal{I}_{i,E}$, as shown in Figure 1(b).

3 Markov chain Monte Carlo

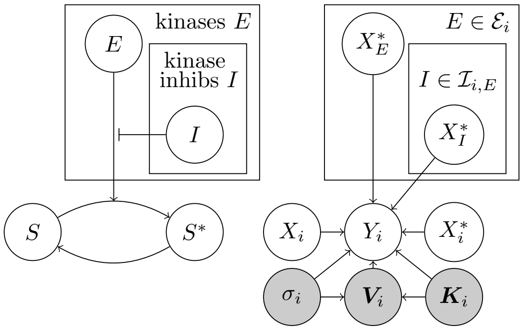
Here we describe the Markov chain Monte Carlo (MCMC) approach used to compute the marginal likelihood $p(\mathcal{D}|G)$. The methodology, due to [Chib and Jeliazkov, 2001], has been demonstrated to perform well against state-of-the-art methods for estimation of marginal likelihood [Friel and Wyse, 2012]. As in Main Text, we focus on a single substrate i and take both i and conditioning on a local reaction graph G_i to be implicit in what follows.

Partition the parameter vector $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$ where $\boldsymbol{\theta}_1 = \boldsymbol{K}$, $\boldsymbol{\theta}_2 = (\boldsymbol{V}, \sigma)$. As noted in Main Text, the conditional posterior density $p(\boldsymbol{\theta}_2|\boldsymbol{\theta}_1, \mathcal{D})$ is available in closed form, making it natural to implement a Gibbs sampler. Indeed, the conditional density $p(\boldsymbol{V}, \sigma|\boldsymbol{K}, \mathcal{D})$ is given in closed form as

$$p(\boldsymbol{V}, \sigma|\boldsymbol{K}, \mathcal{D}) \quad = \quad \mathcal{N}_T(\boldsymbol{V}; \boldsymbol{\mu}, \boldsymbol{\Sigma})\mathcal{IG}(\sigma; a, b), \tag{3}$$

where

$$\begin{aligned} \boldsymbol{\mu} &= \boldsymbol{1}/(n+1) + n/(n+1) \times (\boldsymbol{D}'\boldsymbol{D})^{-1}\boldsymbol{D}'\boldsymbol{z} & (4) \\ \boldsymbol{\Sigma} &= \sigma^2 n/(n+1) \times (\boldsymbol{D}'\boldsymbol{D})^{-1} & (5) \\ a &= (n-1)/2 & (6) \\ b &= (1/2)(\boldsymbol{1}'\boldsymbol{D}'\boldsymbol{D}\boldsymbol{1}/n + \boldsymbol{z}'\boldsymbol{z} - n/(n+1) \times \boldsymbol{z}'\boldsymbol{D}(\boldsymbol{D}'\boldsymbol{D})^{-1}\boldsymbol{D}'\boldsymbol{z}) & (7) \end{aligned}$$



(a) Reaction Graph G (b) Statistical Formulation

Figure 1: (a) A reaction graph G may be considered as a series of phosphorylation and dephosphorylation cycles, with the rate of phosphorylation depending on the concentrations of various kinases E and their inhibitors I. Bounding boxes are used as a shorthand to denote multiple kinases and inhibitors. (b) A graphical model for the kinetics $\boldsymbol{f}_G$ corresponding to G, with unknown parameters $\boldsymbol{\theta}$ in dark gray.

and $\mathcal{IG}(\bullet; a, b)$ is an inverse gamma density with shape and scale parameters a, b respectively. (Here $\boldsymbol{D} = \boldsymbol{D}_{G,S}(\boldsymbol{K})$ is the design matrix defined in the Main Text.) However the remaining conditional $p(\boldsymbol{\theta}_1|\boldsymbol{\theta}_2, \mathcal{D})$ is not available analytically and a Metropolis-Hastings step must be used to facilitate sampling from this distribution.

Denote a Metropolis-Hastings proposal as $q(\boldsymbol{\theta}_1, \boldsymbol{\theta}_1'|\boldsymbol{\theta}_2)$ so that the acceptance probability is

$$\alpha(\boldsymbol{\theta}_1, \boldsymbol{\theta}_1'|\boldsymbol{\theta}_2, \mathcal{D}) = \min\left\{1, \frac{p(\mathcal{D}|\boldsymbol{\theta}_1', \boldsymbol{\theta}_2)p(\boldsymbol{\theta}_1', \boldsymbol{\theta}_2)}{p(\mathcal{D}|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)p(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)} \frac{q(\boldsymbol{\theta}_1', \boldsymbol{\theta}_1|\boldsymbol{\theta}_2, \mathcal{D})}{q(\boldsymbol{\theta}_1, \boldsymbol{\theta}_1'|\boldsymbol{\theta}_2, \mathcal{D})}\right\}. \tag{8}$$

In practice the proposal density is taken to be $\mathcal{N}_T(\boldsymbol{\theta}_1, \lambda \boldsymbol{I})$ where λ is chosen to deliver an average acceptance probability of 30%. The Metropolis-within-Gibbs scheme with M iterations is summarized in Algorithm 2.

Following [Chib and Jeliazkov, 2001] we construct the identity

$$p(\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)p(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)}{p(\boldsymbol{\theta}_2|\boldsymbol{\theta}_1, \mathcal{D})p(\boldsymbol{\theta}_1|\mathcal{D})} \tag{9}$$

and seek an estimator $\hat{p}(\boldsymbol{\theta}_1|\mathcal{D})$ of the posterior ordinate $p(\boldsymbol{\theta}_1|\mathcal{D})$. Then an estimate for the marginal likelihood will be

$$\hat{p}(\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{\theta}_1^*, \boldsymbol{\theta}_2^*)p(\boldsymbol{\theta}_1^*, \boldsymbol{\theta}_2^*)}{p(\boldsymbol{\theta}_2^*|\boldsymbol{\theta}_1^*, \mathcal{D})\hat{p}(\boldsymbol{\theta}_1^*|\mathcal{D})}, \tag{10}$$

for some choice of $\boldsymbol{\theta}^*$. For minimizing estimator variance, [Chib and Jeliazkov, 2001] propose to take $\boldsymbol{\theta}^*$ to be the *maximum a posteriori* (MAP) estimate (or more

Algorithm 2 Parameter sampling scheme

```
$\boldsymbol{\theta}^{(0)} = (\boldsymbol{\theta}_1^{(0)}, \boldsymbol{\theta}_2^{(0)}) \leftarrow$ initial guess
for $i = 1$ to $M$ do
    $\boldsymbol{\theta}'_1 \sim q(\boldsymbol{\theta}_1^{(i-1)}, \boldsymbol{\theta}'_1 | \boldsymbol{\theta}_2^{(i-1)}, \mathcal{D})$
    $r \sim U[0, 1]$
    if $r < \alpha(\boldsymbol{\theta}_1^{(i-1)}, \boldsymbol{\theta}'_1 | \boldsymbol{\theta}_2^{(i-1)}, \mathcal{D})$ then
        $\boldsymbol{\theta}_1^{(i)} \leftarrow \boldsymbol{\theta}'_1$
    else
        $\boldsymbol{\theta}_1^{(i)} \leftarrow \boldsymbol{\theta}_1^{(i-1)}$
    end if
    $\boldsymbol{\theta}_2^{(i)} \sim p(\boldsymbol{\theta}_2 | \boldsymbol{\theta}_1^{(i)}, \mathcal{D})$
end for
```

conveniently the MAP estimator derived from the MCMC sample). In this application we found better performance to be achieved by taking $\boldsymbol{\theta}^*$ to be the arithmetic mean estimator; however in general the arithmetic mean may be unsuitable due to multi-modality or skew in the multidimensional likelihood.

An estimator is constructed based on the identity

$$p(\boldsymbol{\theta}_1^* | \mathcal{D}) = \frac{\mathbb{E}_{p(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 | \mathcal{D})}[\alpha(\boldsymbol{\theta}_1, \boldsymbol{\theta}_1^* | \boldsymbol{\theta}_2, \mathcal{D}) q(\boldsymbol{\theta}_1, \boldsymbol{\theta}_1^* | \boldsymbol{\theta}_2, \mathcal{D})]}{\mathbb{E}_{p(\boldsymbol{\theta}_2 | \boldsymbol{\theta}_1^*, \mathcal{D}) q(\boldsymbol{\theta}_1^*, \boldsymbol{\theta}_1 | \boldsymbol{\theta}_2, \mathcal{D})}[\alpha(\boldsymbol{\theta}_1^*, \boldsymbol{\theta}_1 | \boldsymbol{\theta}_2, \mathcal{D})]}. \tag{11}$$

Estimation of the numerator is directly facilitated by the MCMC output, whereas estimation of the denominator requires an additional Monte Carlo integration, summarized in Algorithm 3. In practice the length of this additional run is taken to be equal to the length M of the full run. For further details see [Chib and Jeliazkov, 2001].

Algorithm 3 Computation of the Chib and Jeliazkov denominator

```
for $i = 1$ to $M$ do
    $\boldsymbol{\theta}_2^{(i)} \sim p(\boldsymbol{\theta}_2 | \boldsymbol{\theta}_1^*, \mathcal{D})$
    $\boldsymbol{\theta}_1^{(i)} \sim q(\boldsymbol{\theta}_1^*, \boldsymbol{\theta}_1 | \boldsymbol{\theta}_2^{(i)}, \mathcal{D})$
end for
denominator $\leftarrow \frac{1}{M} \sum_{i=1}^{M} \alpha(\boldsymbol{\theta}_1^*, \boldsymbol{\theta}_1^{(i)} | \boldsymbol{\theta}_2^{(i)}, \mathcal{D})$
```

We used standard diagnostics to assess convergence of the MCMC sampler, including both “within-run” and “between-run” diagnostics, using parallel runs from dispersed initial conditions [Cowles and Carlin, 1996]. In general the Metropolis-within-Gibbs sampler provided satisfactory convergence of the posterior edge inclusion probabilities. In all experiments we used $M = 10,000$ Monte-Carlo iterations. An example of within-run convergence for the cancer cell line data is shown in Fig. 2.

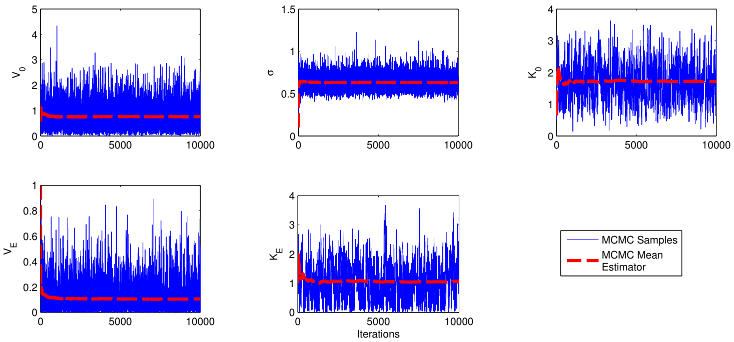


Figure 2: Within-run MCMC convergence diagnostics; cancer cell line data, typical trace plots for kinetic parameters $\boldsymbol{\theta}_S = (V_0, K_0, V_E, K_E, \sigma)$.

4 Sensitivity to hyperparameter specification

The analysis presented in the Main Text requires specification of hyperparameters $\boldsymbol{\mu_V}$, $\boldsymbol{\mu_K}$ and ν. To investigate sensitivity, we first considered one of the simulation regimes presented in the Main Text (specifically $n = 100$ and $\sigma = 0.1$, notation as in Main Text). For this regime we varied each of the three hyperparameters one at a time with the other two held at the values used in Main Text (the set of values used for results reported in Main Text were $\boldsymbol{\mu_V} = \boldsymbol{\mu_K} = \mathbf{1}$, $\nu = 0.5$). We are not directly concerned with identification of dynamical parameters, rather we investigated whether network inference performance (quantified, as in Main Text, by AUPR and AUROC) was highly dependent on the precise values used for these hyperparameters. Results are shown in SFig. 3. Both performance measures appear stable to changes in the hyperparameters.

Empirical results in Main Text demonstrate that the overall set-up, including prior specification, performs well across a range of regimes. Due to computational considerations, we did not carry out exhaustive exploration of hyperparameter values on full networks. Instead, we constructed a smaller toy model, and explored sensitivity more fully using that model. In addition to the hyperparameters considered above, we also considered the influence of the maximum in-degree constraints c_1, c_2. The following model was used

$$\boldsymbol{X} \sim \mathcal{N}_T(\mathbf{1}_{10\times 1}, \boldsymbol{I}_{10\times 10}) \tag{12}$$

$$Z_1 | \boldsymbol{X} \sim \mathcal{N}\left(f_{G,1}(\boldsymbol{X}, \boldsymbol{\theta}_1), \sigma^2 \boldsymbol{I}\right) \tag{13}$$

where we took

$$f_{G,1}(\boldsymbol{X}, \boldsymbol{\theta}_1) = -\frac{V_0 X_1^*}{X_1^* + K_0} + \frac{V_2 X_2^* X_1}{X_1 + K_1} + \frac{V_3 X_3^* X_1}{X_1 + K_3(1 + X_4^*/K_4)} \tag{14}$$

corresponding to two kinases X_2^* and X_3^*, the second of which is inhibited by X_4^*. All parameter values $\boldsymbol{\theta}_1$ were taken to be unity, in line with the observability

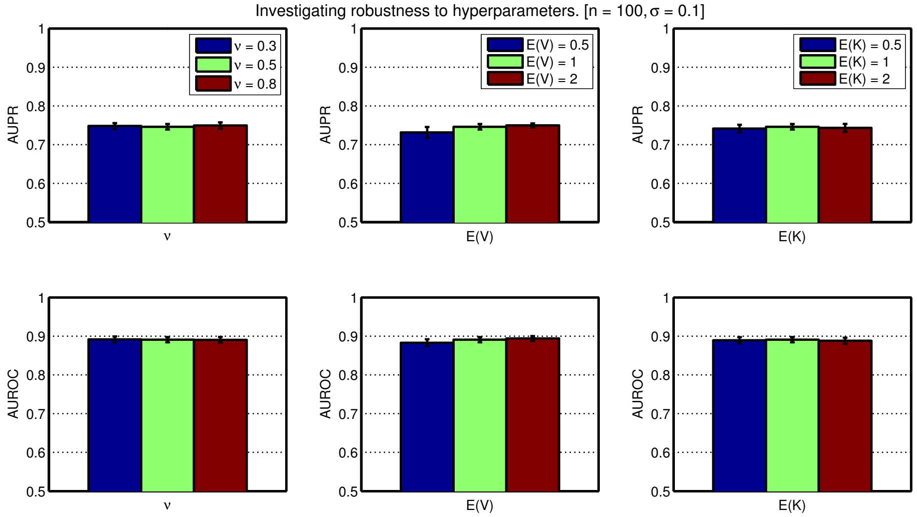
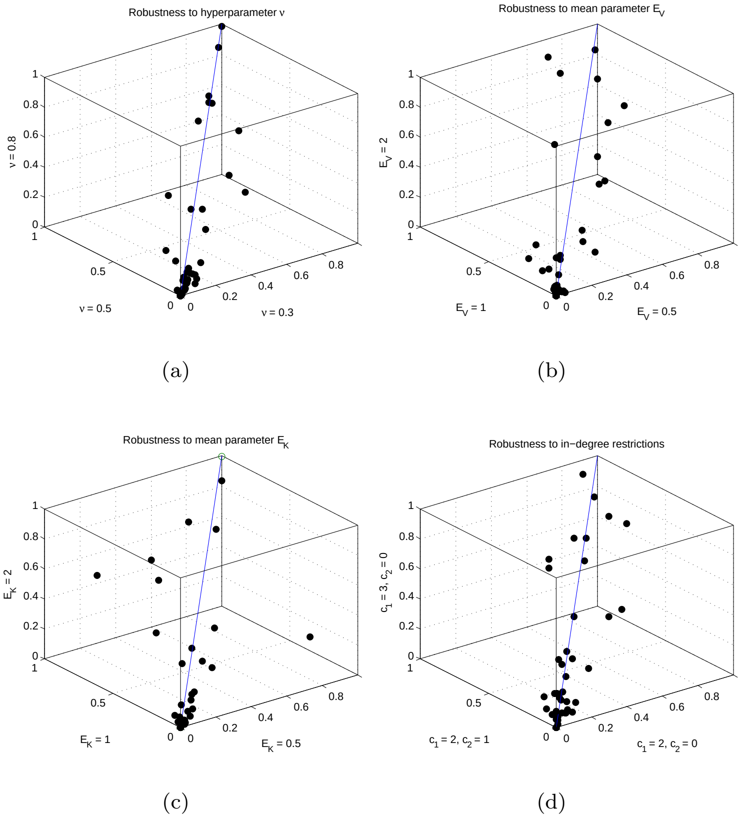


Figure 3: Sensitivity to hyperparameter specification. Network inference performance (quantified, as in Main Text, by AUPR and AUROC) for various hyperparameter values. [Here we present results over 5 independent datasets generated with $n = 100$, $\sigma = 0.1$. The 3 hyperparameters were varied one at a time, with the remaining 2 hyperparameters being set equal to the values used in Main Text, namely $\boldsymbol{\mu_V} = \boldsymbol{\mu_K} = \mathbf{1}$, $\nu = 0.5$.]



(a)

(b)

(c)

(d)

Figure 4: Sensitivity to hyperparameter specification, toy model. (a) prior variance $\mathrm{Var}(K) = \nu^2$, (b) prior mean $\boldsymbol{\mu_V}$ of $\boldsymbol{V}$, (c) prior mean $\boldsymbol{\mu_K}$ of $\boldsymbol{K}$, (d) in-degree restrictions c_1 and c_2,.

hypothesis (see Main Text). For all experiments using the toy model we used $N = 10,000$ MCMC iterations (this was sufficient for convergence).

We first considered $\boldsymbol{\mu_V}$ and $\boldsymbol{\mu_K}$, along with the variance ν^2 of Michaelis-Menten parameters $\boldsymbol{K}$. Fixing $c_1 = 2$, $c_2 = 0$ we computed posterior edge probabilities (PEPs) whilst varying these hyperparameters (SFigs. 4(a-c)). In general we found that PEPs are stable, suggesting that results reported are not highly sensitive to the precise values used.

To investigate sensitivity to the in-degree constraint, we compared results obtained on the toy model using $(c_1 = 2, c_2 = 0)$ with $(c_1 = 2,\ c_2 = 1)$ and $(c_1 = 3,\ c_2 = 0)$ (with $\nu = 0.5$ in all cases). Results are shown in SFig. 4(b), comparing PEPs obtained under the three (c_1, c_2) regimes; we find good agreement between the three regimes, suggesting that the restriction is not highly influential in this setting. Results using $c_1 = 3$ suggest that models allowing 3 kinases to jointly influence a substrate are not needed in situations where the true number of kinases is ≤ 2 (arguably a reasonable assumption for this paper). Results for $c_1 = 2$, $c_2 = 1$ showed that the inhibitor X_4^* was difficult or impossible to identify from data (SFig. 4(b)). This suggests that time course data obtained experimentally may not contain enough information to identify such "second order" inhibitory effects, in line with previous reports that Michaelis-Menten parameters K_i (and hence inhibitory interactions) are only "weakly identifiable" from time course data [4]. The ODE model of [12] does not include inhibitory effects of this kind. Combined with computational considerations, we decided to fix $c_2 = 0$ in all subsequent experiments. As demonstrated in results in the Main Text, CheMA performs well empirically with these restrictions.

5 ODE model of MAPK signaling for simulation

5.1 Dynamical system

The *in silico* model used for our investigation was published by [12], with the ODE formulation $\dot{\boldsymbol{X}} = \boldsymbol{f}_G(\boldsymbol{X}; \boldsymbol{\theta})$ reproduced in Figure 5. Parameter values $\boldsymbol{\theta}$ were chosen an in Section 4 in order to ensure signaling was identifiable in principle from the dynamics.

5.2 Simulation regimes

In order to accurately assess the impact of sample size upon performance, it is important that the amount of information in the simulated data increases with n. Given that the informative range of the dynamics is determined by the choice of parameters (approximately $0 \leq t \leq 20$), adding noise to deterministic data will not satisfy the above requirement, since additional data will merely replicate existing information. We therefore introduced intrinsic stochasticity into the data generating process, interpreting the Xu *et al.* model as the drift

$$\dot{unboundEGFR} = -p_5 \cdot EGF \cdot unboundEGFR + p_6 \cdot boundEGFR$$

$$\dot{removed\text{Raf-1}} = \frac{p_{23} \cdot PKA \cdot \text{Raf-1}}{p_{24} + \text{Raf-1}}$$

$$\dot{removedSOS} = \frac{p_1 \cdot ERKPP \cdot inactiveSOS}{p_2 + inactiveSOS} + \frac{p_1 \cdot ERKPP \cdot activeSOS}{p_2 + activeSOS}$$

$$\dot{inactiveSOS} = -\frac{p_3 \cdot boundEGFR \cdot inactiveSOS}{p_4 + inactiveSOS} + \frac{p_8 \cdot activeSOS}{p_7 + activeSOS} - \frac{p_1 \cdot ERKPP \cdot inactiveSOS}{p_2 + inactiveSOS}$$

$$\dot{inactiveRas} = -\frac{p_9 \cdot activeSOS \cdot inactiveRas}{p_{10} + inactiveRas} + \frac{p_{11} \cdot Gap \cdot activeRas}{p_{12} + activeRas}$$

$$\dot{inactiveRap1} = -\frac{p_{37} \cdot EPAC \cdot inactiveRap1}{p_{38} + inactiveRap1} + \frac{p_{39} \cdot Gap \cdot activeRap1}{p_{40} + activeRap1} - \frac{p_{50} \cdot activeC3G \cdot inactiveRap1}{p_{51} + inactiveRap1}$$

$$\dot{inactivePKA} = -\frac{p_{25} \cdot PKAA \cdot inactivePKA}{p_{26} + inactivePKA} - \frac{p_{27} \cdot Cilostamide \cdot inactivePKA}{p_{28} + inactivePKA} + \frac{p_{30} \cdot PKA}{p_{29} + PKA}$$

$$\dot{inactiveEPAC} = -\frac{p_{31} \cdot EPACA \cdot inactiveEPAC}{p_{32} + inactiveEPAC} - \frac{p_{33} \cdot Cilostamide \cdot inactiveEPAC}{p_{34} + inactiveEPAC} + \frac{p_{36} \cdot EPAC}{p_{35} + EPAC}$$

$$\dot{\text{Raf-1}PP} = \frac{p_{13} \cdot activeRas \cdot \text{Raf-1}}{p_{14} + \text{Raf-1}} - \frac{p_{16} \cdot \text{Raf-1}PP}{p_{15} + \text{Raf-1}PP}$$

$$\dot{\text{Raf-1}} = -\frac{p_{13} \cdot activeRas \cdot \text{Raf-1}}{p_{14} + \text{Raf-1}} + \frac{p_{16} \cdot \text{Raf-1}PP}{p_{15} + \text{Raf-1}PP}$$

$$\dot{boundEGFR} = p_5 \cdot EGF \cdot unboundEGFR - p_6 \cdot boundEGFR$$

$$\dot{activeSOS} = \frac{p_3 \cdot boundEGFR \cdot inactiveSOS}{p_4 + inactiveSOS} - \frac{p_8 \cdot activeSOS}{p_7 + activeSOS} - \frac{p_1 \cdot ERKPP \cdot activeSOS}{p_2 + activeSOS}$$

$$\dot{activeRas} = \frac{p_9 \cdot activeSOS \cdot inactiveRas}{p_{10} + inactiveRas} - \frac{p_{11} \cdot Gap \cdot activeRas}{p_{12} + activeRas}$$

$$\dot{activeRap1} = \frac{p_{37} \cdot EPAC \cdot inactiveRap1}{p_{38} + inactiveRap1} - \frac{p_{39} \cdot Gap \cdot activeRap1}{p_{40} + activeRap1} + \frac{p_{50} \cdot activeC3G \cdot inactiveRap1}{p_{51} + inactiveRap1}$$

$$\dot{PKA} = \frac{p_{25} \cdot PKAA \cdot inactivePKA}{p_{26} + inactivePKA} + \frac{p_{27} \cdot Cilostamide \cdot inactivePKA}{p_{28} + inactivePKA} - \frac{p_{30} \cdot PKA}{p_{29} + PKA}$$

$$\dot{MEKPP} = \frac{p_{17} \cdot \text{Raf-1}PP \cdot MEK}{p_{18} + MEK} - \frac{p_{20} \cdot MEKPP}{p_{19} + MEKPP} + \frac{p_{45} \cdot \text{B-Raf}PP \cdot MEK}{p_{46} + MEK}$$

$$\dot{MEK} = -\frac{p_{17} \cdot \text{Raf-1}PP \cdot MEK}{p_{18} + MEK} + \frac{p_{20} \cdot MEKPP}{p_{19} + MEKPP} - \frac{p_{45} \cdot \text{B-Raf}PP \cdot MEK}{p_{46} + MEK}$$

$$\dot{ERKPP} = \frac{p_{21} \cdot MEKPP \cdot ERK}{p_{22} + ERK} - \frac{p_{55} \cdot ERKPP}{p_{54} + ERKPP}$$

$$\dot{ERK} = -\frac{p_{21} \cdot MEKPP \cdot ERK}{p_{22} + ERK} + \frac{p_{55} \cdot ERKPP}{p_{54} + ERKPP}$$

$$\dot{EPAC} = \frac{p_{31} \cdot EPACA \cdot inactiveEPAC}{p_{32} + inactiveEPAC} + \frac{p_{33} \cdot Cilostamide \cdot inactiveEPAC}{p_{34} + inactiveEPAC} - \frac{p_{36} \cdot EPAC}{p_{35} + EPAC}$$

$$\dot{EGF} = -p_5 \cdot EGF \cdot unboundEGFR + p_6 \cdot boundEGFR$$

$$\dot{\text{B-Raf}PP} = \frac{p_{41} \cdot activeRap1 \cdot \text{B-Raf}}{p_{42} + \text{B-Raf}} - \frac{p_{44} \cdot \text{B-Raf}PP}{p_{43} + \text{B-Raf}PP} + \frac{p_{52} \cdot activeRas \cdot \text{B-Raf}}{p_{53} + \text{B-Raf}}$$

$$\dot{\text{B-Raf}} = -\frac{p_{41} \cdot activeRap1 \cdot \text{B-Raf}}{p_{42} + \text{B-Raf}} + \frac{p_{44} \cdot \text{B-Raf}PP}{p_{43} + \text{B-Raf}PP} - \frac{p_{52} \cdot activeRas \cdot \text{B-Raf}}{p_{53} + \text{B-Raf}}$$

$$\dot{activeC3G} = \frac{p_{47} \cdot boundEGFR \cdot inactiveC3G}{p_{48} + inactiveC3G} - p_{49} \cdot activeC3G$$

$$\dot{inactiveC3G} = -\frac{p_{47} \cdot boundEGFR \cdot inactiveC3G}{p_{48} + inactiveC3G} + p_{49} \cdot activeC3G$$

Figure 5: *In silico* ODE model of the EGFR/ERK signaling pathway due to [12].

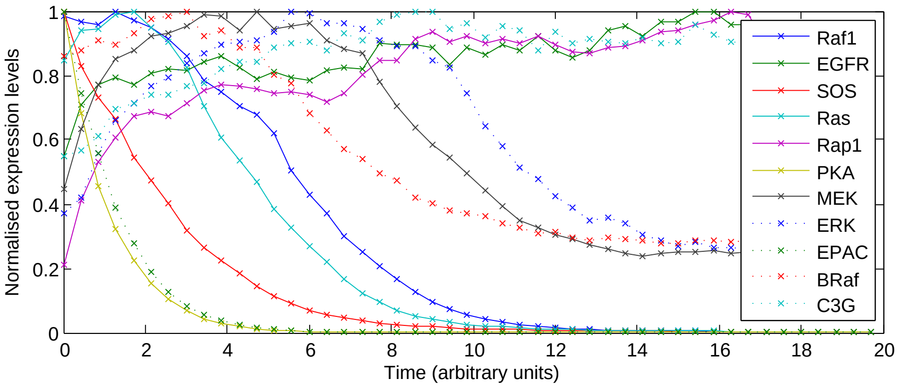


Figure 6: Typical simulated time course from the ODE model of [12]. [Initial conditions were drawn from a truncated standard Gaussian; four such time courses constitute a dataset. Here 100 evenly spaced samples are shown with intrinsic noise of magnitude $\sigma = 0.05$. Species expression is normalized to unit maximum to improve presentation.]

in a stochastic differential equation:

$$\boldsymbol{X}(0) = \boldsymbol{x}_0 \qquad (15)$$

$$d\boldsymbol{X} = \boldsymbol{f}_G(\boldsymbol{X}, \boldsymbol{\theta})dt + \sigma d\boldsymbol{B} \qquad (16)$$

where σ controls the magnitude of the stochastic fluctuations. Initial state $\boldsymbol{x}_0$ was drawn from the truncated standard normal distribution.

To generate time courses, we simulated solutions $\boldsymbol{X}(t)$ of this SDE for times $0 \leq t \leq 20$ and then selected $[n/4]$ evenly spaced samples; four such time courses constituted a dataset. Data regimes were characterized by total observation sample size $n \in \{25, 50, 100, 200\}$ and noise magnitude $\sigma \in \{0, 0.05, 0.1, 0.15, 0.2\}$. A time course with 100 evenly spaced samples is shown in SFig. 6. Simulated datasets differ in both the initial state $\boldsymbol{x}_0$ and the realization of the Brownian motion $\boldsymbol{B}$.

5.3 Details of assessment

Of the 25 state variables, 3 denote drug compounds; these were not considered for the purpose of network inference. The remaining 22 variables denote the active and inactive forms of 11 signaling proteins; Raf1, EGFR, SOS, Ras, Rap1, PKA, MEK, ERK, EPAC, BRaf and C3G. Network inference was therefore performed for these 11 proteins, in each of the experimental regimes, using each available method. Disregarding self-edges made a total of $(2^{10})^{10} \approx 10^{29}$ possible networks.

6 Implementation

All of the methods used in Main Text have a number of user-set parameters or configurations. We used default configurations for each method, as described below.

6.1 LASSO

We used the R package glmnet [6] to train an l1-regularised linear model (known as LASSO, for Least Absolute Shrinkage and Selection Operator) on the input data. The optimal setting of the regularisation parameter λ was determined for each dataset separately using cross-validation. For each node i in the network, we learn a regression model for observations $Y_i^*(t)$ with respect to the remaining nodes $Y_j^*(t-1)$ $(j \neq i)$ at time $t-1$. LASSO automatically sets the regression coefficients of some nodes to zero. We used the absolute values of the regression coefficients to give an indication of the strength of each edge in the network. We used the default settings of the glmnet, and the input data for each regression were standardised to mean 0 and variance 1.

6.2 TSNI

Time Series Network Inference (TSNI) [2] was run according to the recommended settings provided at http://dibernardo.tigem.it/wiki/index.php/Time_Series_Network_Identification_TSNI-integral. Since TSNI only accepts single time series, the resulting weighted adjacency matrices corresponding to separate time courses were subsequently averaged to obtain a single network estimate.

6.3 DBN

To learn dynamic Bayesian networks (DBNs) from the data, we used the model described in [7], which also corresponds to the model in [5] when one imposes the restriction of not allowing changepoints. For obtaining the results in this paper, we therefore used the R software package EDISON that implements the model in [5] and samples from it via reversible-jump MCMC. We fixed the changepoint settings so that no changepoints would be inferred during the network inference. The sampled networks were evaluated based on the marginal posterior probability of each edge. We used the default settings of the software package, except for the maximum number of iterations, which was set to 1e6. The data was standardised to mean 0 and variance 1. Note that alternative implementations of linear DBNs may enjoy computational advantages [7].

6.4 TVDBN

For inferring time-varying DBNs, we again used the R software package EDISON that implements the model in [5]. In this case, changepoints were allowed to be

Method:	CheMA	LASSO	TSNI	DBN	TVDBN	GP
Time (secs):	2×10^4	1	1	4×10^3	4×10^3	3×10^2

Table 1: Computational times (approximate) for inference of the Xu *et al.* network. [Implementational details for the various methods are contained in Section 6. Note that certain methods may enjoy more favourable computational implementations, e.g. [7] for linear DBNs.]

inferred during the reversible-jump MCMC, which potentially allows for modelling nonlinear effects. The sampled networks were evaluated based on the marginal posterior probability of each edge. We used the default settings of the software package, except for the maximum number of iterations, which was set to 1e6. The data was standardised to mean 0 and variance 1. We also used information sharing with a soft coupling of nodes, as described in [5], to regularise the number of changes at each changepoint.

6.5 GP

GP [1] was run in MATLAB R2012b using code generously supplied by Tarmo Äijö. On noise-free data ($\sigma = 0$) this code could encounter numerical loss of positive-definiteness so, when required, covariance matrices were regularized using Tikhonov regularization prior to Cholesky decomposition. GP was then run using the following settings; optimization iterations = 50, no delay terms, zero order model = used, maximum in-degree = 2, prior covariance = $0.01 \times \mathbf{I}$, prior mean = $\mathbf{0}$.

6.6 Computational times

Table 1 contains the approximate computational time requirements of the competing methodologies. It may be seen that the chemical kinetic approach is considerably more demanding compared with competing approaches, requiring at least 5 times more computation. Note that these time requirements are empirical and implementation-dependent; a formal time complexity analysis of the algorithms is beyond the scope of this paper.

For illustration of computation for larger networks, we ran CheMA using data obtained on breast cancer cell line AU565 (see Section 9) based on 27 phosphoproteins (network not shown, since its interpretation and assessment is beyond the scope of this paper). This required over 12 hours of computational time. This illustrates that in principle CheMA could be used for larger networks. However, there were fewer samples ($n = 24$) than protein species in the dataset, and only 2 targeted interventions, so caution would need to be exercised in interpreting the results.

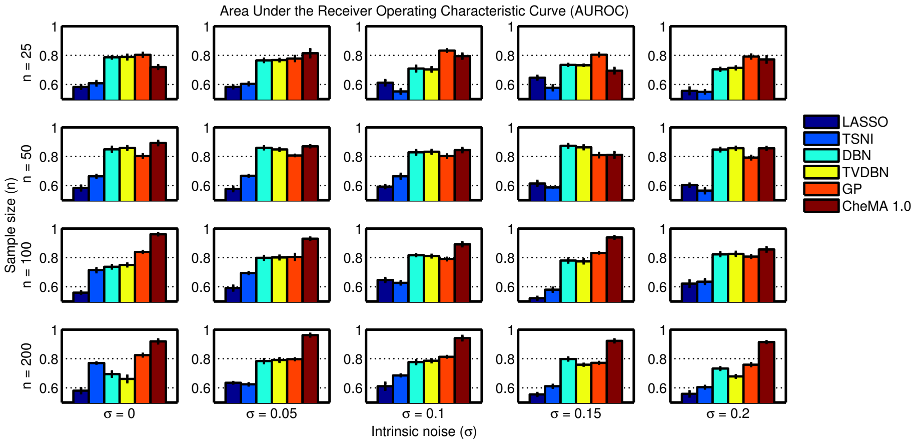


Figure 7: Average area under the ROC curve (AUROC; with respect to the true causal graph). [Network inference methods: (i) LASSO, ℓ_1-penalized regression, (ii) TSNI, ℓ_2-penalized regression, (iii) DBN, dynamic Bayesian networks, (iv) TVDBN, time-varying DBNs, (v) GP, nonparametric regression, (vi) CheMA, the proposed approach. For each panel we averaged AUROC over 5 independent datasets. Subplots correspond to particular sample size n and noise level σ.]

7 *In silico* results

7.1 Network reconstruction

SFig. 7 shows AUROC scores for varying sample size n and intrinsic noise σ. We see that CheMA offers superior performance at large sample sizes ($n = 100, 200$), whereas at low sample sizes GP, DBN and TVDBN may confer an advantage. The linear and piecewise linear DBNs displayed reduced performance at large sample size, in line with inconsistency arising from model misspecification. SFig. 8 summarizes both AUPR and AUROC scores by averaging over σ for fixed n and *vice versa*.

7.2 Parameter inference

The present work does not focus on identification of dynamical parameters, but rather on network inference and dynamical prediction. Nevertheless it is interesting to consider behaviour with respect to parameter inference. SFig. 9, which is reproduced in the Main Text, displays posterior probability distributions over parameters $\boldsymbol{\theta}$ for the toy model of Eqn. 14 (assuming known true graph G, else the parameters are not well defined) for varying sample size n. Results show that, whilst maximum reaction rates V_0, V_2, V_3 could be estimated from data, Michaelis-Menten parameters K_0, K_1, K_2, K_3, K_4 were much more difficult to infer. Estimation for the noise parameter σ demonstrated bias toward lower values. In general, inference at the smaller sample size was much less successful.

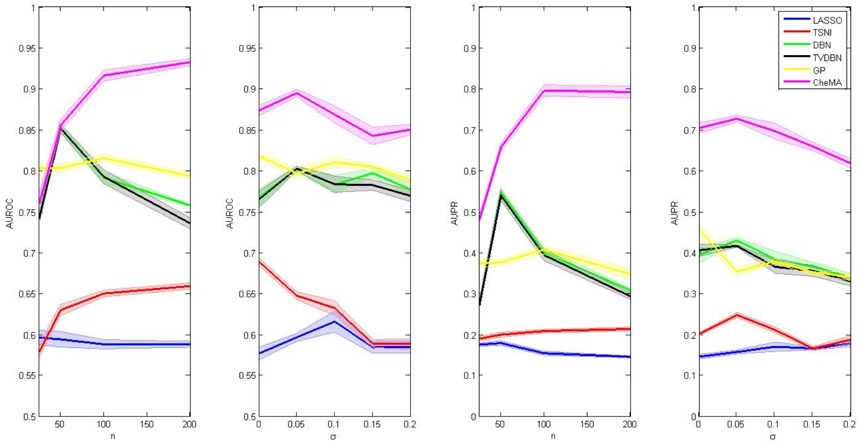


Figure 8: An overview of network inference performance. [Here we average the AUROC and AUPR results from Main Text Fig. 2 and SFig. 7 over varying stochasticity σ for fixed sample size n and *vice versa*. Network inference methods: (i) LASSO, ℓ_1-penalized regression, (ii) TSNI, ℓ_2-penalized regression, (iii) DBN, dynamic Bayesian networks, (iv) TVDBN, time-varying DBNs, (v) GP, nonparametric regression, (vi) CheMA, the proposed approach. Shaded regions display standard error, computed over 25 independent datasets.]

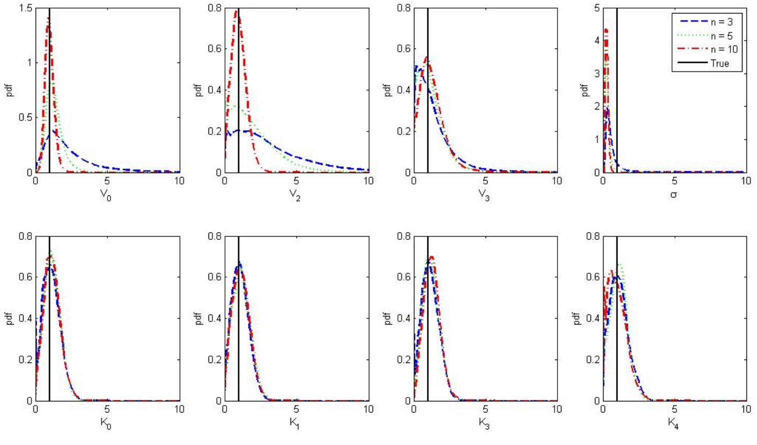


Figure 9: (Marginal) parameter posterior distributions for increasing sample size n. [For the Zellner g-prior, the $n = 3$ case is the closest well-defined analogue to a prior which we can plot.]

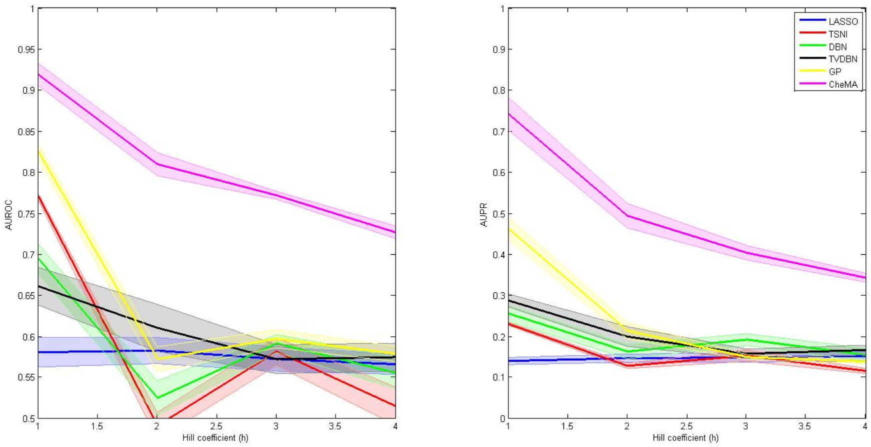


Figure 10: Impact of nonlinearity. [Here $n = 200$, $\sigma = 0$. Shaded regions display standard error computed over 5 independent datasets.]

7.3 The effect of nonlinearity

We varied the amount of nonlinearity in the data-generating model by introducing Hill coefficients h into all Michaelis-Menten functionals

$$\frac{V[S]}{[S]+K} \mapsto \frac{V[S]^h}{[S]^h + K^h} := g([S]). \tag{17}$$

Here $h > 1$ implies positively cooperative binding: Once one substrate molecule is bound to the enzyme, its affinity for other substrate molecules increases. As $h \to \infty$, the function g approaches a step function $g(x) = V \times 1\{x \geq 1\}$, grossly invalidating the assumption of the linear model.

We simulated additional data under positively cooperative binding ($h = 2, 3, 4$, $n = 200$, $\sigma = 0$) in order to quantify the impact of nonlinearity upon inference. Results (SFig. 10), which are averaged over 5 independent datasets, showed that all methods' performance decreased in the highly nonlinear regimes.

8 Prediction of signaling response

For the prediction problem we are given training data $\mathcal{D}$ and an initial condition $\boldsymbol{x}_0$, from which the goal is to predict the entire time course $\boldsymbol{x}(t)$. Below we describe how these data were generated and how training data were used. The quality of a prediction was assessed by mean square error (MSE) with respect to the test data. All protein species were normalized by their maximum value in the training data $\mathcal{D}$. The network inference algorithms used in Section 7 have not been modified for prediction; we therefore considered simple stationary and linear benchmark predictors (described in the Main Text).

8.1 Data generation

Training data $\mathcal{D}$ were generated as described in Section 5. For test data, one randomly chosen protein X_i was selected as the target of an intervention. One time course $\boldsymbol{x}(t)$ was generated under this intervention by forcing terms X_i^* corresponding to the target(s) of intervention to equal zero in the drift $\boldsymbol{f}_G$ of Eqn. 7.

8.2 Stationary benchmark

The benchmark mean square error was computed by predicting $\boldsymbol{x}(t) = \boldsymbol{x}_0$ for all t.

8.3 CheMA

Our approach returns samples from the joint posterior distribution $p(G, \boldsymbol{\theta}|\mathcal{D})$ over reaction graphs G and parameters $\boldsymbol{\theta}$. In order to facilitate prediction of $\boldsymbol{x}(t)$, we perform model-averaging as described in Algorithm 4. For the experiments reported in the Main Text we used $I = 1,000$ samples to construct an averaged prediction. Note that, since we do not model genetic variation, prediction is conditional upon the noisy measurements of unphosphorylated protein expression in $\boldsymbol{x}$; linear interpolation of noisy data is used to approximate unphosphorylated protein concentrations at any given time.

Algorithm 4 CheMA prediction

```
for $i = 1$ to $I$ do
    $G^{(i)} \sim p(G|\mathcal{D})$
    $\boldsymbol{\theta}^{(i)} \sim p(\boldsymbol{\theta}|G^{(i)}, \mathcal{D})$
    Numerically solve the ODE $\dot{\boldsymbol{X}} = \boldsymbol{f}_{G^{(i)}}(\boldsymbol{X}, \boldsymbol{\theta}^{(i)})$ from the initial condition $\boldsymbol{X}(0) = \boldsymbol{x}_0$. Denote the solution by $\boldsymbol{X}^{(i)}$.
end for
Predict $\boldsymbol{x}(t) \approx \frac{1}{I}\sum_{i=1}^{I} \boldsymbol{X}^{(i)}(t)$.
```

8.4 Linear kinetics

For an unbiased assessment of the importance of nonlinearity in inference, the same approach to prediction was employed based on the linear model $f_{G,S}(\boldsymbol{X}, \boldsymbol{\theta}) = \beta_{0,S} + \sum_{E \in \mathcal{E}_S} \beta_{E,S} X_E^*$ where, following [7], the parameters $\boldsymbol{\beta}_S$ and σ_S for a given target S are assigned (untruncated) Zellner prior distributions with zero mean. Models G involving kinase inhibition were excluded from inference (inhibitory effects are accommodated by allowing coefficients to become negative). We believe this to be the closest (reasonable) linear approximation to the chemical kinetic framework described above.

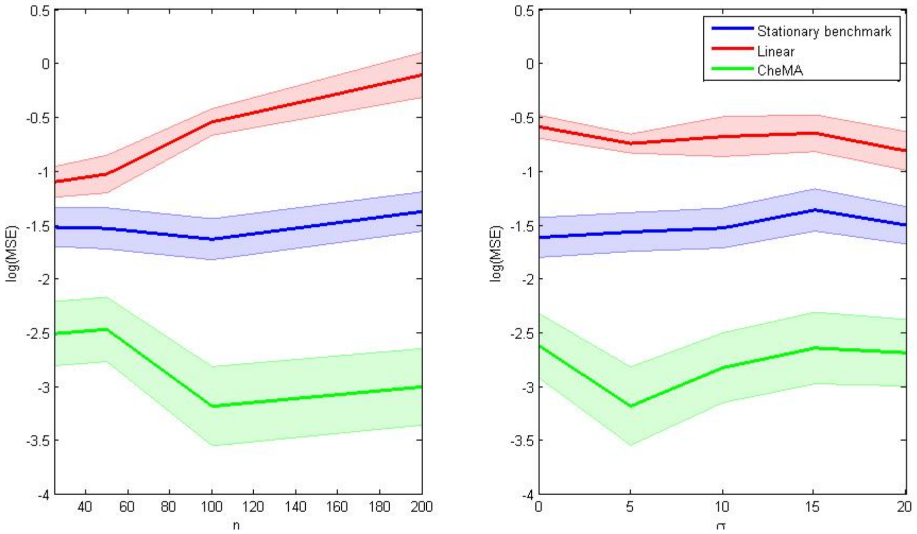


Figure 11: Assessment of predictive performance over varying sample size n and noise level σ; average normalized mean square error. [Shaded regions display standard error, computed over 15 independent datasets.]

8.5 Results

We found that linear methods do not deliver stable prediction over longer periods of time (Fig. 3(a) Main Text). We therefore computed MSE relative to the test data over the initial duration $0 \leq t \leq 5$, representing the first 25% of the held-out time course. SFig. 11 displays average normalized mean square error for the above predictions, varying both the size n of the test sample and the noise level σ. We see that in all regimes the CheMA predictions significantly outperform both the linear and stationary benchmark predictions. Interestingly, linear predictions performed increasingly badly for increasing sample size n; this may be due to model misspecification.

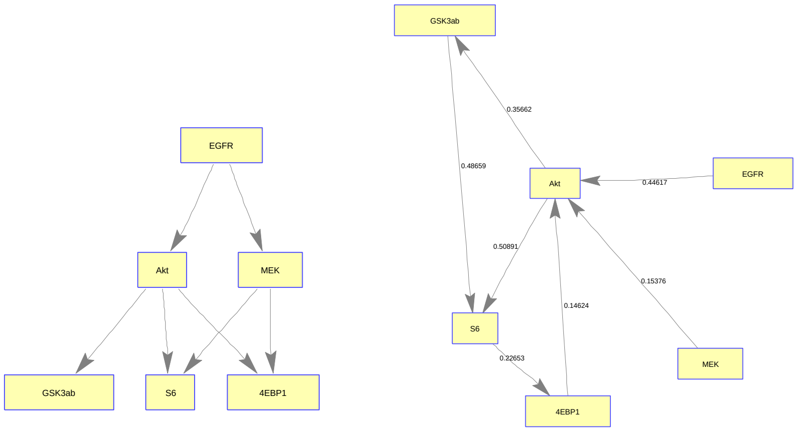
9 RPPA experimental protocol

Cells were plated into 10 cm^2 dishes at a density of $1 - 2 \times 10^6$ cells. After 24 hours, cells were treated with 250 nM lapatinib or 250 nM AKTi (GSK690693). DMSO served as a control. Cells were grown in 10% FBS and harvested in RPPA lysis buffer at 30 min, 1h, 2h, 4h, 8h, 24h, 48h, and 72h post-treatment. Cell lysates were quantitated, diluted, arrayed, and probed as described previously [11]. Imaging and quantitation of signal intensity was done as described previously [11]. Pre-treatment allowed for protein phosphorylation levels to respond to kinase inhibition treatment. In this way, the initial time point contains considerable information concerning the effect of treatment.

The particular protein species analyzed were 4EBP1(pT37), AKT(pS473), EGFR(pY1173), GSK3ab(pS21), MEK1/2(pS217), S6(pS240).

10 *In vitro* results

From literature we obtained a canonical protein signaling network (SFig. 12a). Many of the networks inferred by CheMA shared topology with the literature network (SFig. 12b). However it is not possible to validate inferred line-specific topology without extensive biochemistry. We therefore focused on the predictive power of CheMA, comparing this to the predictive power afforded by the literature network coupled with kinetic equations as described in the Main Text. SFig. 13 displays typical predictions produced by both approaches. MSE was calculated over all proteins and all time points in the test data, with protein-specific normalization performed as in Section 8.



(a) (b)

Figure 12: (a) Protein signaling network derived from literature. (b) Inferred topology for cell line HCC 70. [Edge weights correspond to posterior probabilities. Only the most probable edges are displayed.]

References

[1] Äijö T, Lähdesmäki H (2010) Learning gene regulatory networks from gene expression measurements using non-parametric molecular kinetics. *Bioinformatics* **25**(22):2937-2944.

[2] Bansal M, di Bernardo D (2007) Inference of gene networks from temporal gene expression profiles. *IET Systems Biology* **1**(5):306-312.

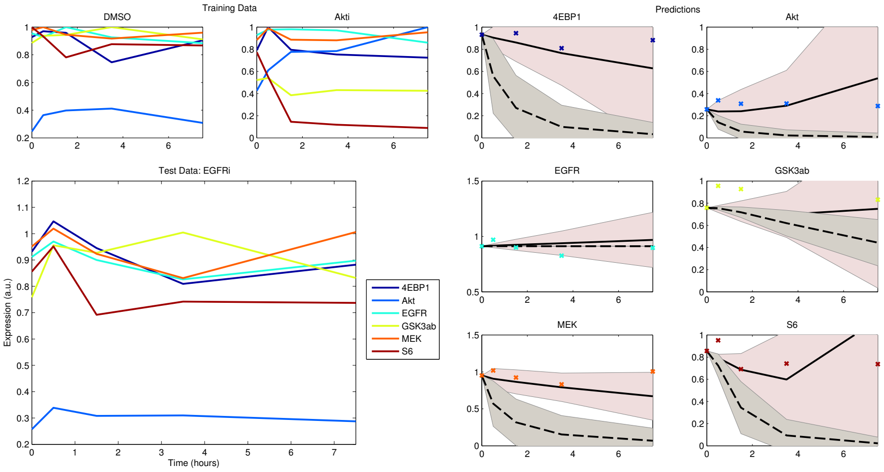


Figure 13: True and predicted *in vitro* trajectories for cell line HCC 70. [CheMA predictions (solid) with $1 \times \sigma$ confidence region (pink) and predictions based on the literature signaling topology (dashed) with $1 \times \sigma$ confidence region (gray).]

[3] Bender *et al.* (2011) Inferring signalling networks from longitudinal data using sampling based approaches in the R package "ddepn". *BMC Bioinformatics* **12**:291.

[4] Calderhead B, Girolami M (2011) Statistical analysis of nonlinear dynamical systems using differential geometric sampling methods. *J Roy Soc Interface Focus*, **1**(6):821-835.

[Chib and Jeliazkov, 2001] Chib, S., and Jeliazkov, I. (2001). Marginal Likelihood From the Metropolis-Hastings Output. *J Am Stat Assoc*, 96(453), 270-281.

[Cowles and Carlin, 1996] Cowles, M. K., and Carlin, B. P. (1996). Markov Chain Monte Carlo Convergence Diagnostics: A Comparative Review. *J Am Stat Assoc*, 91(434), 883-904.

[5] Dondelinger F, Lèbre S, Husmeier D (2012) Non-homogeneous dynamic Bayesian networks with Bayesian regularization for inferring gene regulatory networks with gradually time-varying structure. *Mach Learn* **90**(2):191-230.

[6] Friedman J, Hastie T, Tibshirani R. (2010) Regularization paths for generalized linear models via coordinate descent. *J Stat Softw* **33**(1):1.

[Friel and Wyse, 2012] Friel, N., and Wyse, J. (2012) Estimating the Statistical Evidence - A Review. *Stat Neerl*, 66, 288-308.

[7] Hill *et al.* (2012) Bayesian inference of signaling network topology in a cancer cell line. *Bioinformatics* **28**(21):2804-2810.

[8] Roberts GO, Rosenthal JS (2006) Harris Recurrence of Metropolis-within-Gibbs and Trans-Dimensional Markov Chains, *Ann Appl Probab* **16**(4):2123-2139.

[9] Rodriguez-Yam G, Davis RA, Scharf LL (2002) A Bayesian model and Gibbs sampler for hyperspectral imaging. *Sensor Array and Multichannel Signal Processing Workshop Proceedings*: 105-109.

[10] Rodriguez-Yam G, Davis RA, Scharf LL (2004) Efficient Gibbs Sampling of Truncated Multivariate Normal with Application to Constrained Linear Regression, unpublished manuscript.

[11] Tibes *et al.* (2006) Reverse phase protein array: validation of a novel proteomic technology and utility for analysis of primary leukemia specimens and hematopoietic stem cells. *Mol Cancer Ther* bf 5(10):2512-2521.

[12] Xu *et al* (2010) Inferring signaling pathway topologies from multiple perturbation measurements of specific biochemical species, *Science Signaling* **3**(113):ra20.