Supplement: Efficient Amortised Bayesian Inference for Hierarchical and Nonlinear Dynamical Systems

Geoffrey Roeder[1,2], Paul K Grant[1], Andrew Phillips[1], Neil Dalchau[1], and Edward Meeds[1]

[1]Microsoft Research, Cambridge, United Kingdom
[2]Princeton University, Princeton, United States of America

S1 Extended Prior Work

This section presents an extended version of the prior works in the body of the text.

Structure in Variational Distributions. Independently, Ainsworth et al. (2018) developed a methodology for leveraging block factorization in factor analysis. Hierarchical models in variational inference have previous been associated with hierarchical structure within the variational distribution, as in (Ranganath et al., 2016). For nonlinear mixed-effects models, we are instead interested in hierarchical structure within the dependent variable rather than the approximate posterior. Hence, we can keep the computationally-convenient, fully factorized approximate posterior, and induce conditioning through a block-factorization. We therefore expand the formulation of NLME ODE models proposed in Karlsson et al. (2015) to consider a more general hierarchical structure.

ODE Parameter Inference *Markov-Chain Monte Carlo (MCMC)* methods have been considered the gold standard for inference in ODEs (e.g., Xun et al. 2013). This is largely because MCMC permits sampling exactly from the true posterior, as guaranteed by asymptotic analysis of the ergodic sampling chain. However, MCMC inference requires expensive numerical integration at each step. For a latent sequential process, approximating an integral can be prohibitively expensive. Moreover, MCMC does not typically converge quickly when there is multimodality in the true posterior due to the accept/reject sampling. Our case study was chosen to exhibit this problem: Dalchau et al. (2019) applied MCMC inference to the synthetic biological problem described in sec. 4, and report chain convergence times of approximately 24 hours for relatively small datasets.

Likelihood-free methods (a.k.a. Approximate Bayesian Computation (ABC)) are also commonly used to learn parameters of dynamical systems, avoiding the need to run a chain to convergence (Gorbach et al., 2017). ABC leverages fast model simulations and computes approximate likelihoods through comparing summary statistics. This also avoids the need to compute a potentially costly or intractable likelihood functions. Sequential Monte Carlo (SMC) ABC (Sisson et al., 2007) simulates a discretised dynamical system with Runge-Kutta methods Toni et al. (2009). We also apply a Runge-Kutta simulation, but use an explicit likelihood and variational inference without SMC to take advantage of fast, gradient-based optimization.

Gradient matching is a learning algorithm that avoids numerical integration through Gaussian Process regression to the state variables of the dynamical system. Gorbach et al. (2017) introduce a gradient matching algorithm that applies mean-field variational inference to discover moments of the population distribution in order to fit a GP that matches them. GP regression is very effective for small to medium sized models with little data. However, GPs do not scale adequately to massive datasets without sparse inducing points or additional structural assumptions. It is not clear how to extend GP regression to the kinds of conditional distributions required for hierarchical modelling as we investigate here.

Variational Inference for Dynamical Systems A number previous works have applied variational inference to learn the parameters of non-hierarchical dynamical models, mostly in the Kalman filter family. We briefly summarize and indicate key differences with our work.

State-space models have been learned through variational inference, as in Archer et al. (2015). The goal of Archer et al. (2015) method is to reduce the dimension of the input space down to an interpretable

two-dimensional random variable. Such compression induces a loss of information, which is not a goal of our method. Their method generalizes linear dynamical systems, and applies variational inference to learn a suitable approximate posterior distribution. Methodologically, our paper deals with dynamical systems that have hierarchical latent structure with highly nonlinear latent transitions. Moreover, our model does not require full state observability, e.g., we model hidden latent processes that are captured implicitly in the equations but do not explicitly appear as terms in the final observation process.

Similarly, Krishnan et al. (2015) *learn nonlinear Kalman filters* through stochastic variational inference. Krishnan et al. (2015) explicitly generalise linear dynamical systems, discovering arbitrarily complex transition dynamics and emission distributions. A key difference is that the mean and covariance functions for their (tridiagonal) variational distribution are recurrent neural networks. The use of an RNN to generate the parameters of a highly nonlinear Kalman filter is similar to but more constrained than our black-box model (since we make no restrictions on the transitions).

Recently, Ryder et al. (2018) explores variational inference for *stochastic differential equations*. Methodologically, our paper is similar to Ryder et al. (2018) in that an ordinary differential equation is the limit of a stochastic differential equation as the diffusion term approaches zero. The approximate posterior in Ryder et al. (2018) factorizes into a component that determines the parameters for the SDE drift and diffusion matrices, and a component that describes the evolution of the latent process. The component that describes the latent process evolution is autoregressive, requiring a sequential evaluation for the log-density that includes a log-det Jacobian term to account for how the probability mass changes step to step. Our method uses much less computation by simplifying the approximate posterior, supporting our goal of fast iteration among different candidate models of a data-generating process. Moreover, we apply our approximate posterior not to model the evolution of the latent process, but to identify the parameters of the dynamical system.

Recently (Chen et al., 2018) proposed *neural ordinary differential equations*, a new perspective on residual networks, recurrent neural network decoders, normalizing flows and other functions approximators that exhibit repeated composition. They observe that since the outputs of such recursive functions are functionally identical to discretised ODEs, then some continuous-time differential function at some initial condition can uniquely generate them. They learn a variational distribution over the initial conditions for such models, and then use an ODE solver to simulate the system. They instead apply the adjoint method to compute the gradients Pontryagin (2018), helping alleviate memory problems with large models.

Despite surface similarities, the modeling regime of (Chen et al., 2018) is markedly different from ours. We learn a block-conditional variational distribution over the parameters of a system of ODEs (in the white box case), or over a hierarchical factorization of the latent variables (in the black box case). The variational distribution our method discovers represents the parameters of this system of ODEs, rather than the latent state itself. By contrast, Chen et al. (2018)'s variational distribution is over the initial state of the latent time series. They simulate by generating the solution using any off-the-shelf ODE solver. In our case, the initial state is irrelevant, as we are leveraging probabilistic structure in the variational distribution to capture data-generating process more efficiently, by sharing statistical strength where relevant. In particular, each dimension of the variational distribution we learn has a specified interpretation according to either a mechanistic model of the data-generating process, or a hierarchical assumption about variability, in the black-box case.

By 'wiring up' interactions in cells, a synthetic biologist can build information processing systems that function like Boolean logic gates (Nielsen et al., 2016), like analog electrical circuits (Daniel et al., 2013), or that mimic the behavior of natural biological systems (Grant et al., 2016). While a design environment exists for scalably building transcriptional logic circuits with predictable Boolean behaviors (Nielsen et al., 2016), existing approaches for constructing circuits with dynamical behaviors are still in their infancy (Dalchau et al., 2019). The quantitative behavior over time of biological circuits cannot currently be predicted from DNA sequence alone, so experiments must be performed to measure key properties to understand the dynamics of these systems and to allow rational design decisions about future circuits.

S2 Extended Case Study Description

The synthetic genetic circuits we have used in this work are built from gene cassettes comprised of DNA sequences encoding a promoter, ribosome binding site, coding region, and terminator. These cassettes are assembled into plasmids that are used to transform *E. coli* cells. We refer to a collection of cassettes that implement a particular design as a *device*. The

devices used in this paper are all *double receiver* devices that respond to 3-oxo-C6-homoserine lactone (C6) by producing cyan fluorescent protein (CFP) and to 3-oxo-C12-homoserine lactone (C12) by producing yellow fluorescent protein (YFP) (Grant et al., 2016). These devices are built of 5 cassettes. The first 4 cassettes are arranged on a plasmid, which includes one cassette each for producing luxR and lasR proteins (R and S in the main text), the receiver proteins that bind C6 and C12, respectively, a CFP cassette activated by C6-bound luxR, and a YFP cassette activated by C12-bound lasR. The fifth cassette is chromosomally integrated, and constitutively expresses RFP. The devices vary in the strength of the ribosome binding sites in the luxR and lasR cassettes, creating devices that vary in the amount of each of those proteins expressed and therefore their sensitivity to C6 and C12.

S2.1 White-box (mechanistic)

Our general approach for constructing prescribed (white-box) models of biological circuits combines a population-level model for cell culture growth with more detailed models for the concentrations of intra- and intercellular molecules, and resembles the approach commonly used in the synthetic biology literature (Balagaddé et al., 2008; Daniel et al., 2013; Chen et al., 2015; Dalchau et al., 2019). Cell growth models are generally described by the product of the current cell density $c(t)$ and the *specific growth rate* $\gamma(c(t))$, which describes both the per capita growth rate and the decrease in intracellular concentrations due to an increased volume. As explained in the main text, we used a smoothed version of the lag-logistic model for cell growth here.

To model the cellular biochemistry, we translate chemical reaction networks to ODEs using mass action kinetics, which assumes that reactions fire at a rate proportional to the concentration of the reactants. Translating chemical reactions in this way in general leads to a large number of equations, because all mRNAs, proteins, small molecules and complexes between each produce their own equation. As such, model reductions are commonly applied to reduce the number of dependent variables, but result in more complex nonlinearities. Following this approach, the white-box model we consider here (Section 4) was derived in detail previously (Grant et al., 2016; Dalchau et al., 2019). It describes the time-evolution of the response of double receiver devices to HSL signals C_6 and C_{12}, vector $\mathbf{u}$ in (2). The latent variables $\mathbf{x}$ in (2) are the culture density c, the intracellular concentrations of each expressed protein (luxR, lasR, RFP, CFP, YFP) and variables for autofluorescence, which we model as concentration of intracellular material fluorescent at 480 nm (F_{480}) and 530 nm (F_{530}).

As there are no mRNA species, the variables a_k describe a lumped maximal rate of transcription and translation. The variables d_k describe the intracellular degradation rates of each protein.

The *response* functions f_{76} and f_{81} describe the inducibility of CFP and YFP to complexes involving the HSL signals and the receiver proteins luxR and lasR. The response functions were derived from chemical reactions, making the assumption that signal-receiver binding and unbinding is faster than protein synthesis and degradation (Dalchau et al., 2019). This results in very complex functions, but they are still interpretable as exhibiting saturation behaviour, which occurs as either receiver proteins or promoters become limiting. We define $B_R^{(k)}$ and $B_S^{(k)}$ as the fractions of luxR and lasR proteins bound by an HSL signal

$$B_R = \frac{(K_{R6}.C_6)^{n_R} + (K_{R12}.C_{12})^{n_R}}{(1 + K_{R6}.C_6 + K_{R12}.C_{12})^{n_R}} \tag{S1a}$$

$$B_S = \frac{(K_{S6}.C_6)^{n_S} + (K_{S12}.C_{12})^{n_S}}{(1 + K_{S6}.C_6 + K_{S12}.C_{12})^{n_S}}, \tag{S1b}$$

These functions are bounded above by 1, which occurs when luxR/lasR become limiting. The CFP or YFP genes are transcribed more efficiently when their promoters are bound by one of the receiver-signal complexes. As such, an additional saturation can be observed within the derived forms

$$f_j(R, S, C_6, C_{12}) = \frac{\epsilon^{(j)} + K_{GR}^{(j)} R^2 B_R + K_{GS}^{(j)} S^2 B_S}{1 + K_{GR}^{(j)} R^2 B_R + K_{GS}^{(i)} S^2 B_S} \tag{S2}$$

where $j \in \{76, 81\}$. Here, the parameters $K_{GR}^{(j)}$ and $K_{GS}^{(j)}$ are the affinity constants of receiver-signal complexes for each promoter j, and $\epsilon^{(j)}$ is the leaky rate of transcription/translation in the absence of an activating complex (such as when there is no HSL).

The specific growth rate $\gamma(c_i)$ describes the per-capita cellular growth rate of culture i. Cultures are subscripted in this way because their growth parameters will be local to the culture, which enables implicit accounting for feedback from circuit activity or extrinsic factors that vary in different experiments. As in (Dalchau et al., 2019), we use a lag-logistic growth model which explicitly quantifies a *lag* phase of bacterial growth before an *exponential* phase and then *stationary* phase. This is usually formulated as

$$\gamma(c_i) = \begin{cases} r_i.(1 - \frac{c}{K_i}), & t > t_{\text{lag},i} \\ 0, & t < t_{\text{lag},i} \end{cases}$$

but to ensure that the right-hand sides of f are differentiable, we replace the switch at $t_{\text{lag},i}$ with a steep sigmoid (Equation 9).

Finally, we remind the reader that we consider the application of this model to multiple devices, in which different ribosome-binding site sequences have been used for the luxR and lasR genes (Section 4). In this mechanistic model, it is the parameters a_R and a_S that are allowed to be device-conditioned. Therefore, using the one-hot mapping strategy for specifying the rbs elements in each device, the model can take on device-specific quantities for luxR and lasR synthesis.

S3 Comparison with MCMC

To determine how VI-HDS compares with a simple approach to Bayesian inference, we sought to approximate the inference problem using Markov chain Monte Carlo (MCMC). In principle, MCMC methods can provide an exact characterisation of the posterior, but often many samples are necessary for convergence. For comparison with our VI-HDS results, we generated MCMC chains for the white-box model. Due to the presence of 4 *individual* parameters in the model (r, K, t_{lag}, r_c), and $N = 312$ data-points, there were more than 4×312 parameters to be sampled. With so many parameters, we found that even 1 million burn-in samples was insufficient to reach convergence (Figure S1). Furthermore, local optima of the likelihood function were difficult to avoid.

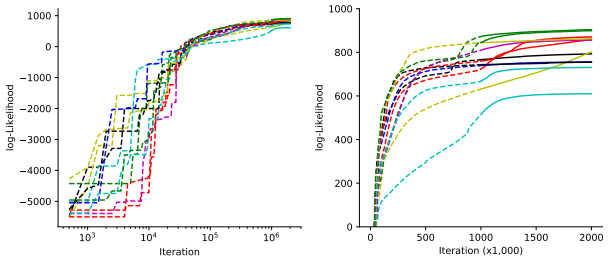

Figure S1: Convergence of Markov chain Monte Carlo

The solutions found by MCMC after 2 million samples were not as convincing as those found after 500 epochs of VI-HDS. We found that the RFP signal was poorly reconstructed, with the model showing faster dynamics than the data, and the posterior predictive distribution having higher variance than for other signals (Figure S2). With even more MCMC samples, perhaps a better parameter regime could be found.

In summary, the shortcomings of MCMC are clearly visible. A sequence of 1-2 million likelihood evaluations is the absolute minimum required to find a reasonable solution to the inference problem, whereas

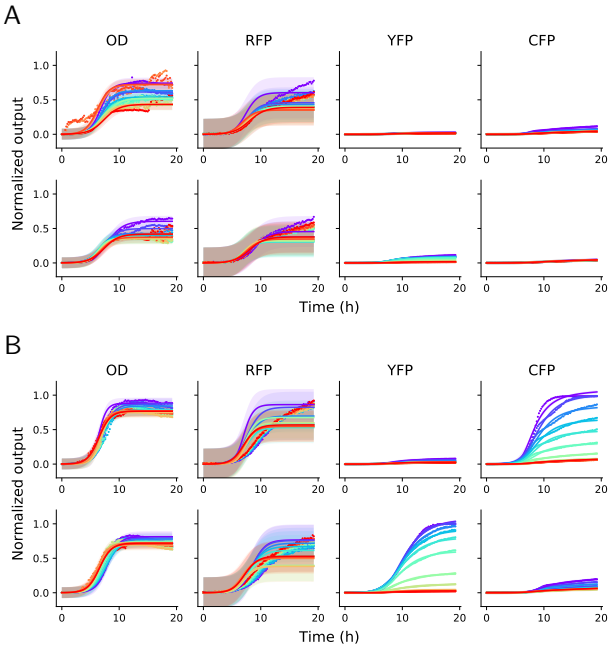

Figure S2: Summary plots for the white-box model inferred using MCMC. The posterior samples from the MCMC chain with highest log-likelihood was used to produce a posterior predictive distribution. Time-series are partitioned by treatment (C_6 in the top row of each panel, C_{12} in the bottom row), with a color scale indicating the concentration (warm colors indicate higher concentration). Devices shown are (**A**) Pcat-Pcat and (**B**) R33-S175.

in VI-HDS, a sequence of 500 epochs is sufficient. As each epoch incorporates 100 importance samples, there are essentially 50,000 likelihood function evaluations, which is 40 times fewer than was required of MCMC.

S4 Supplementary Figures

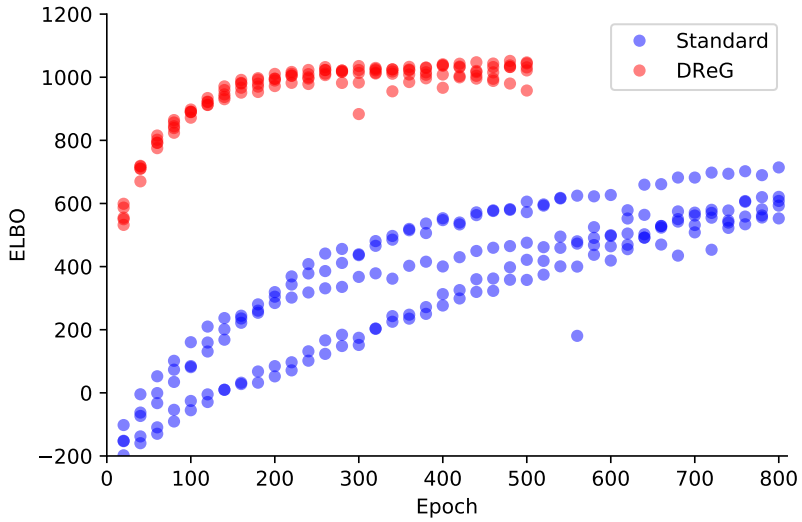

Figure S3: The convergence of the ELBO is improved by the DReG estimator. Shown are 5 independent evaluations of the VI method applied to the prescribed *constant* model, using the standard gradient estimator (blue) and the DReG estimator (red). Each reported ELBO score is the average of a 4-fold cross-validation at a given epoch.

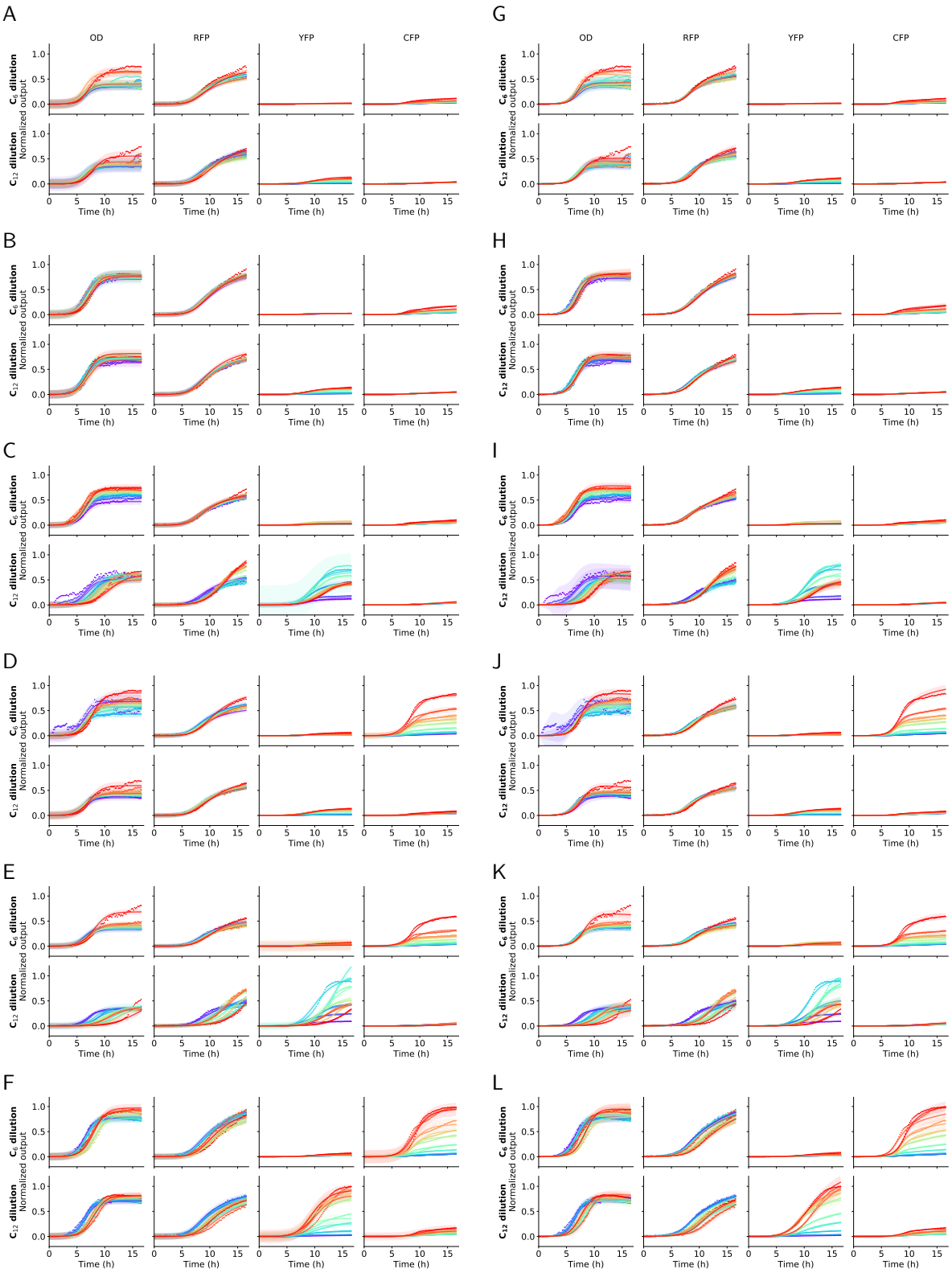

Figure S4: Summary plots for the white-box model. Time-series are partitioned by device and by treatment (C_6 or C_{12}), with a color scale indicating the concentration (warm colors indicate higher concentration). Devices shown are (**A**) Pcat-Pcat, (**B**) RS100-S32, (**C**) RS100-S34, (**D**) R33-S32, (**E**) R33-S34 and (**F**) R33-S175.

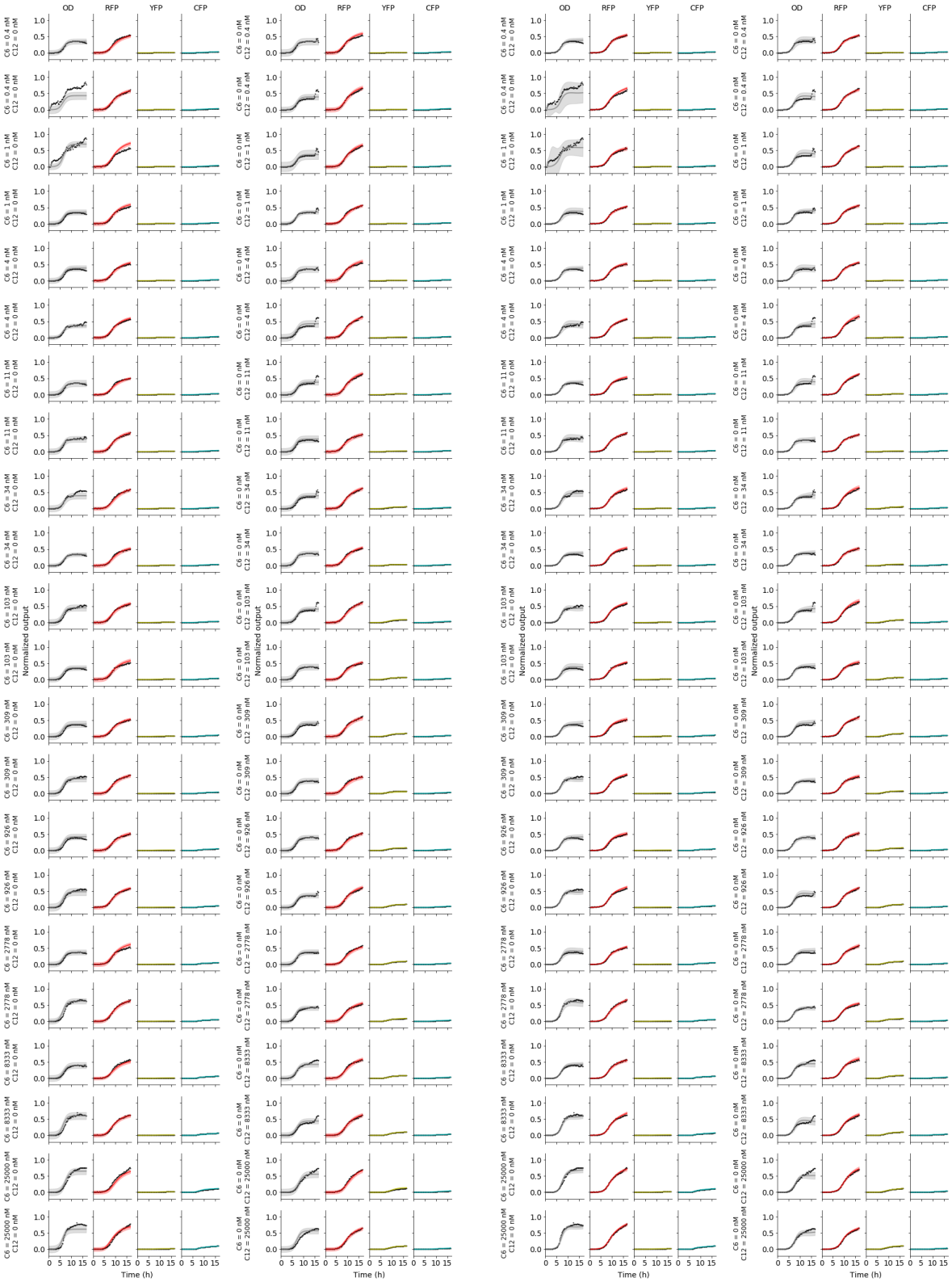

Figure S5: Posterior predictive distribution for 4-fold cross-validation experiment: Pcat-Pcat device. The left half shows the white-box model, and the right half shows the black-box model comparisons. Within each half, the left batch of four columns are C_6 treatments, and the right batch of four columns are C_{12} treatments, as detailed to the left of each row.

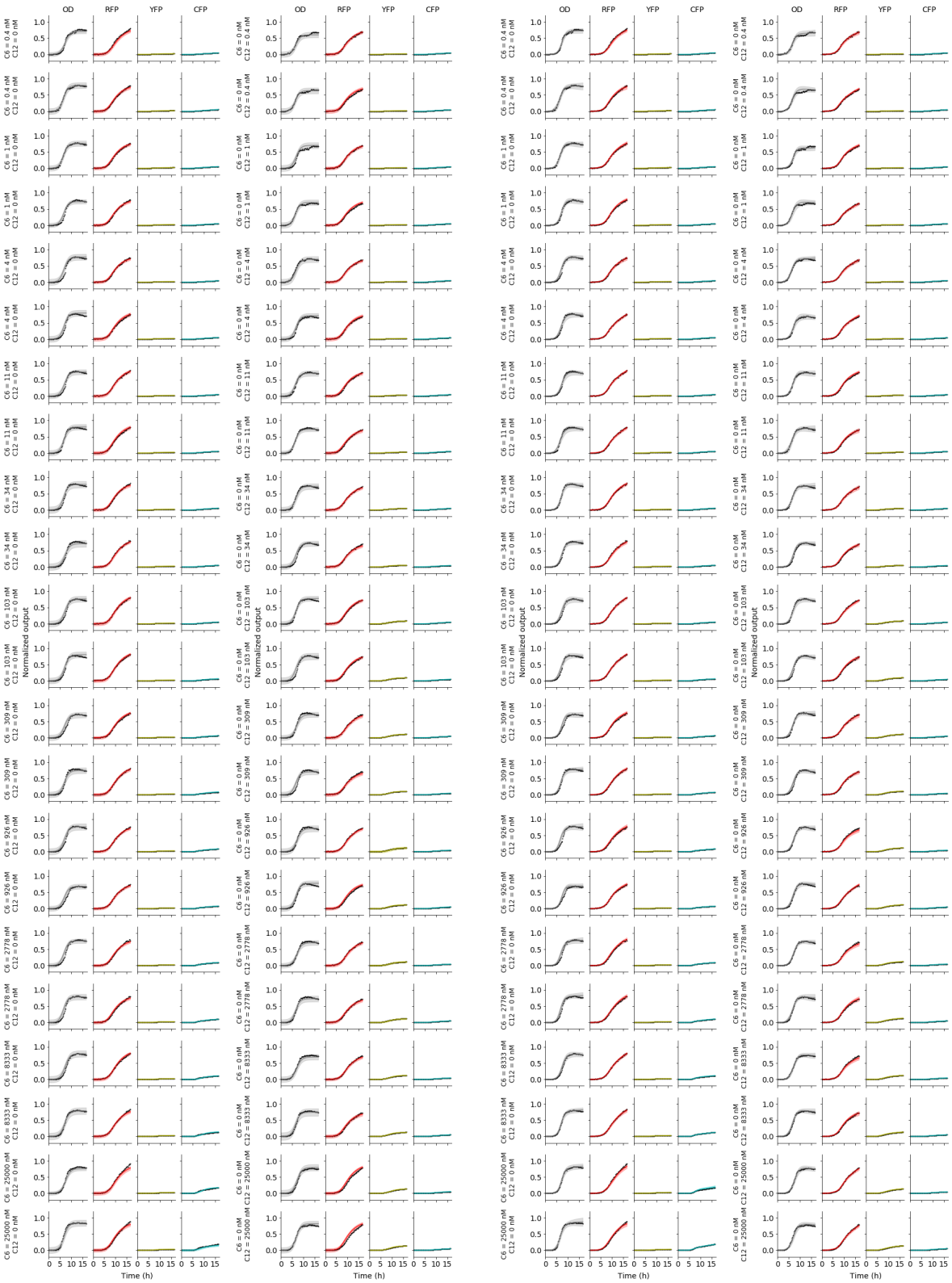

Figure S6: Posterior predictive distribution for 4-fold cross-validation experiment: RS100-S32 device. The left half shows the white-box model, and the right half shows the black-box model comparisons. Within each half, the left batch of four columns are C_6 treatments, and the right batch of four columns are C_{12} treatments, as detailed to the left of each row.

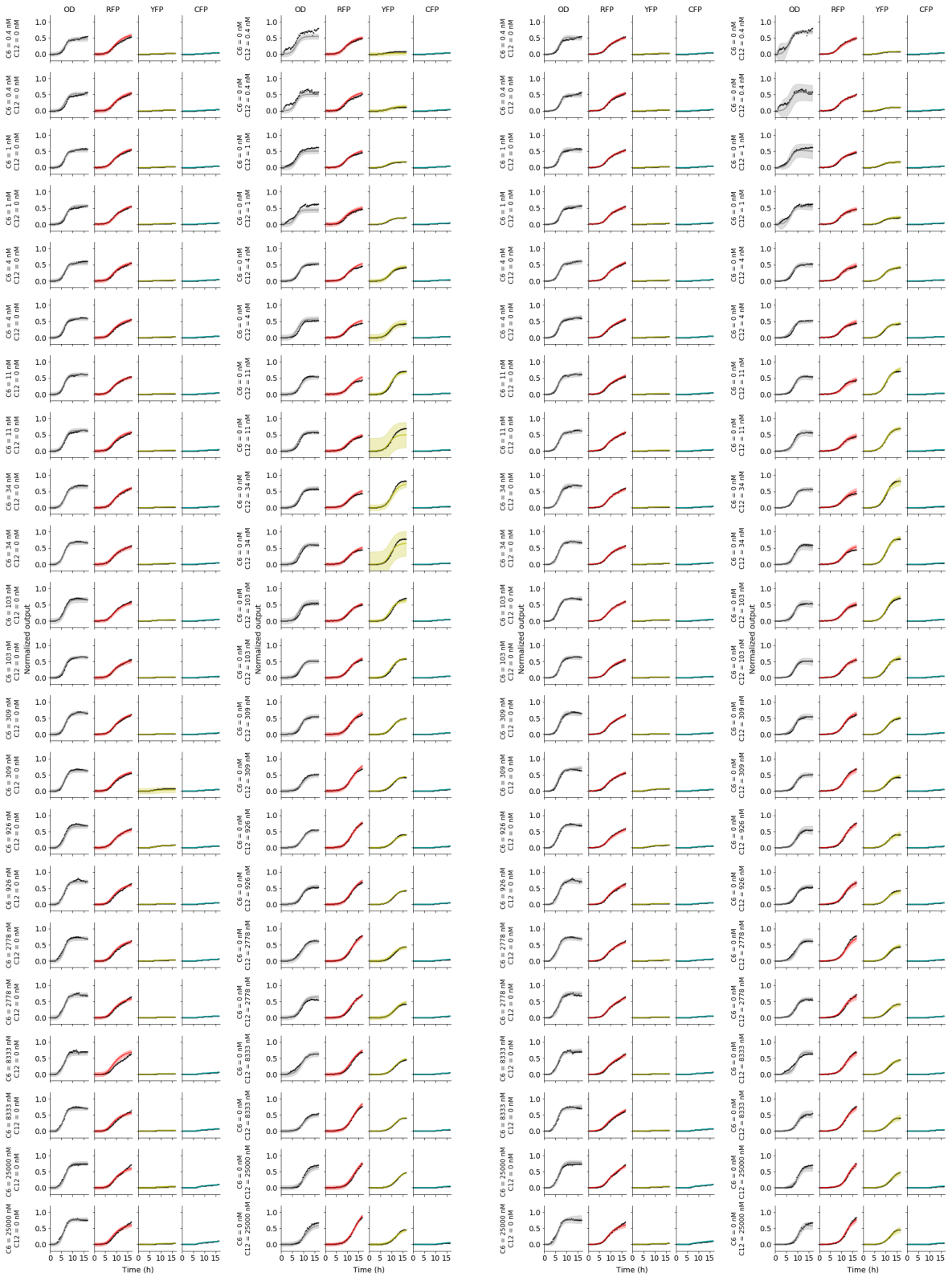

Figure S7: Posterior predictive distribution for 4-fold cross-validation experiment: RS100-S34 device. The left half shows the white-box model, and the right half shows the black-box model comparisons. Within each half, the left batch of four columns are C_6 treatments, and the right batch of four columns are C_{12} treatments, as detailed to the left of each row.

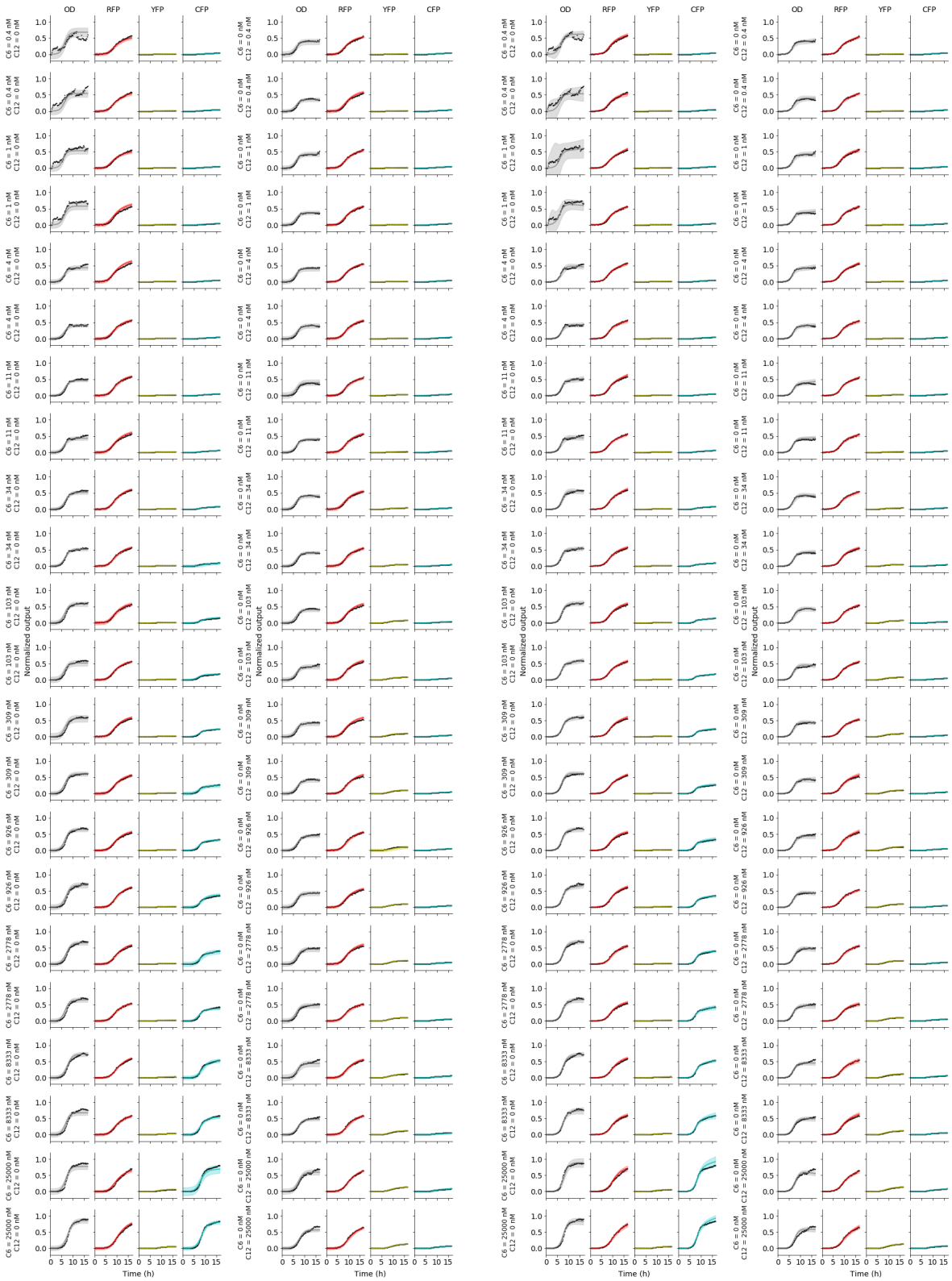

Figure S8: Posterior predictive distribution for 4-fold cross-validation experiment: R33-S32 device. The left half shows the white-box model, and the right half shows the black-box model comparisons. Within each half, the left batch of four columns are C_6 treatments, and the right batch of four columns are C_{12} treatments, as detailed to the left of each row.

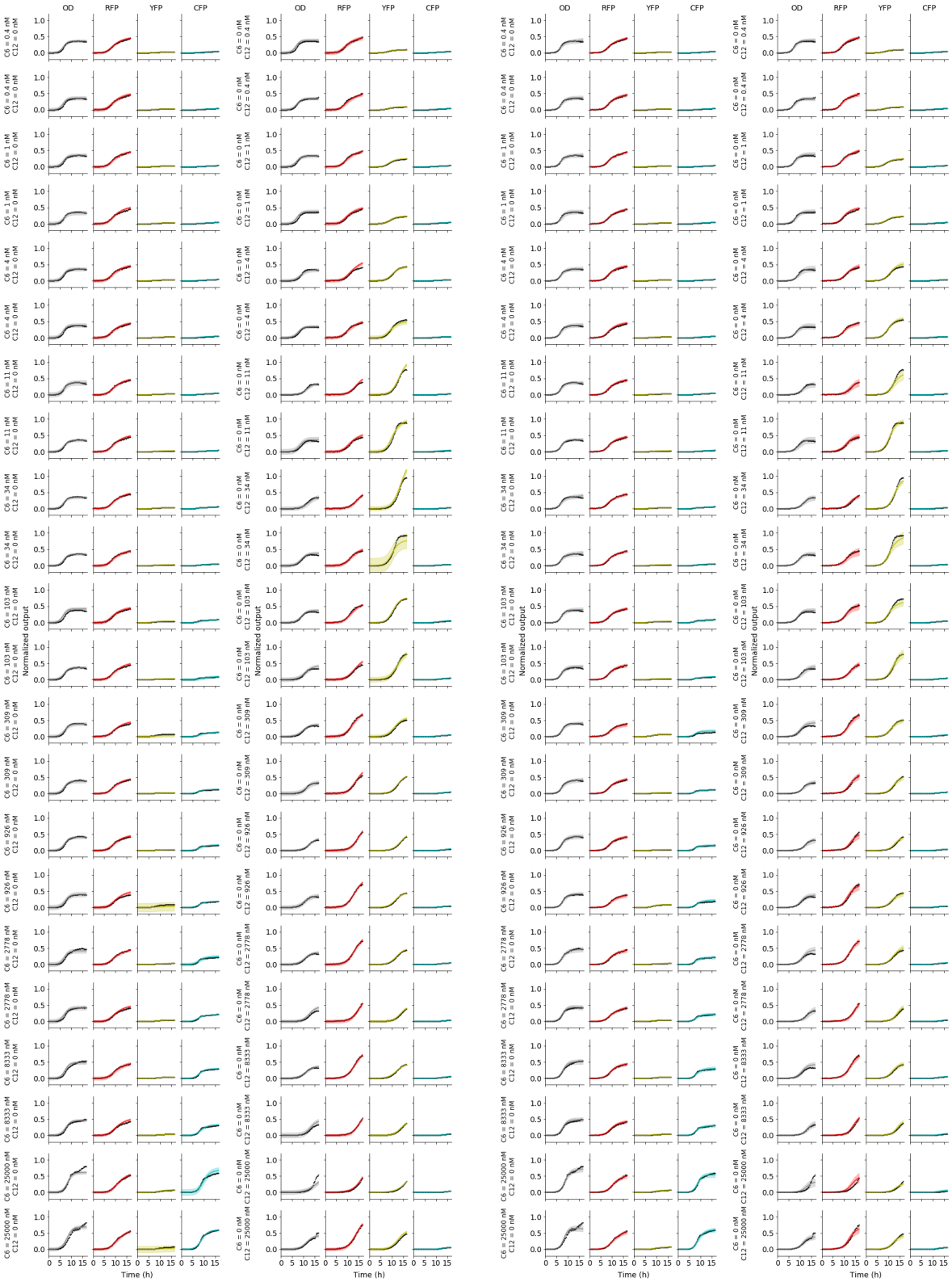

Figure S9: Posterior predictive distribution for 4-fold cross-validation experiment: R33-S34 device. The left half shows the white-box model, and the right half shows the black-box model comparisons. Within each half, the left batch of four columns are C_6 treatments, and the right batch of four columns are C_{12} treatments, as detailed to the left of each row.

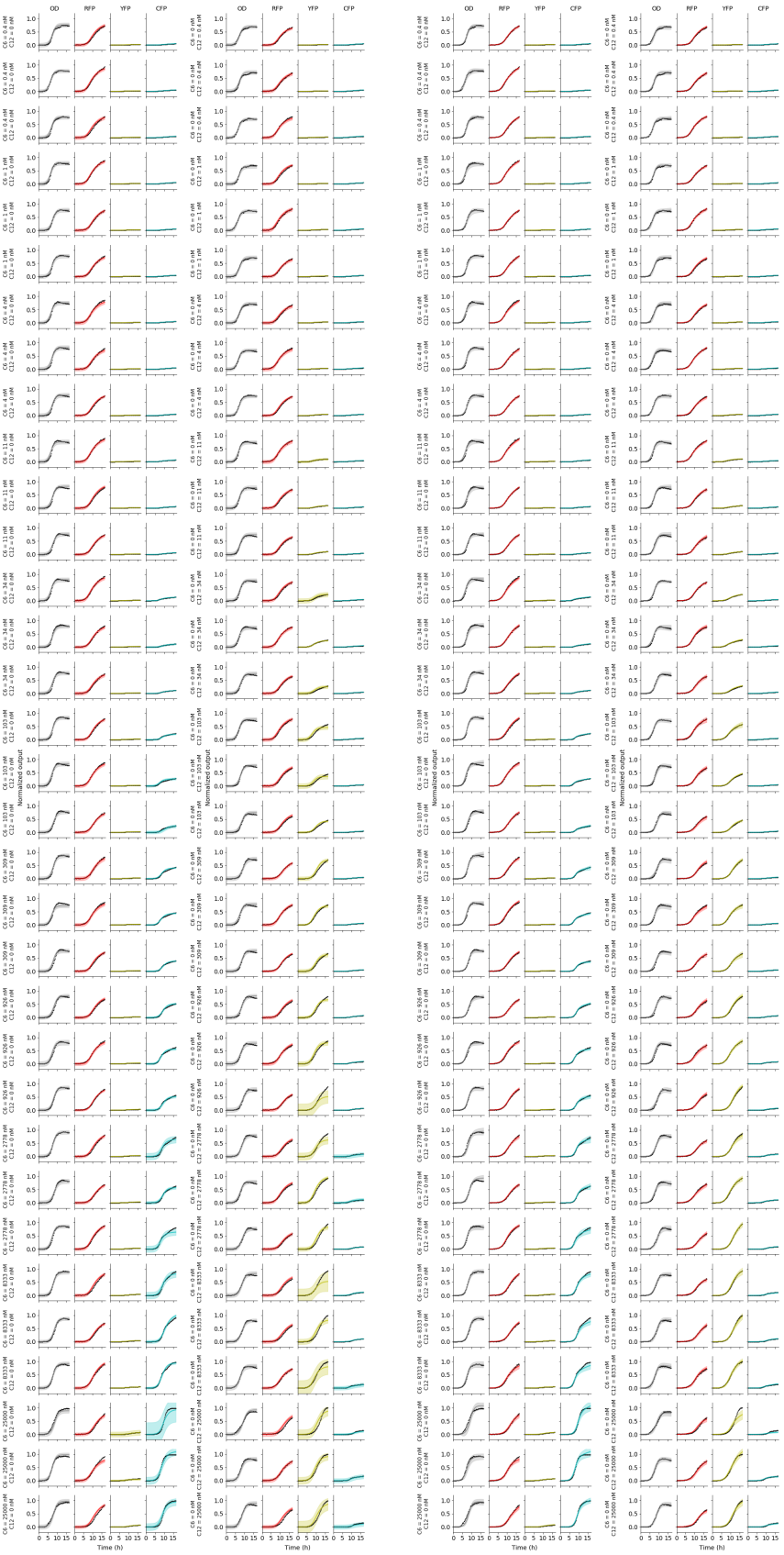

Figure S10: Posterior predictive distribution for 4-fold cross-validation experiment: R33-S175 device. The left half shows the white-box model, and the right half shows the black-box model comparisons. Within each half, the left batch of four columns are C_6 treatments, and the right batch of four columns are C_{12} treatments, as detailed to the left of each row.

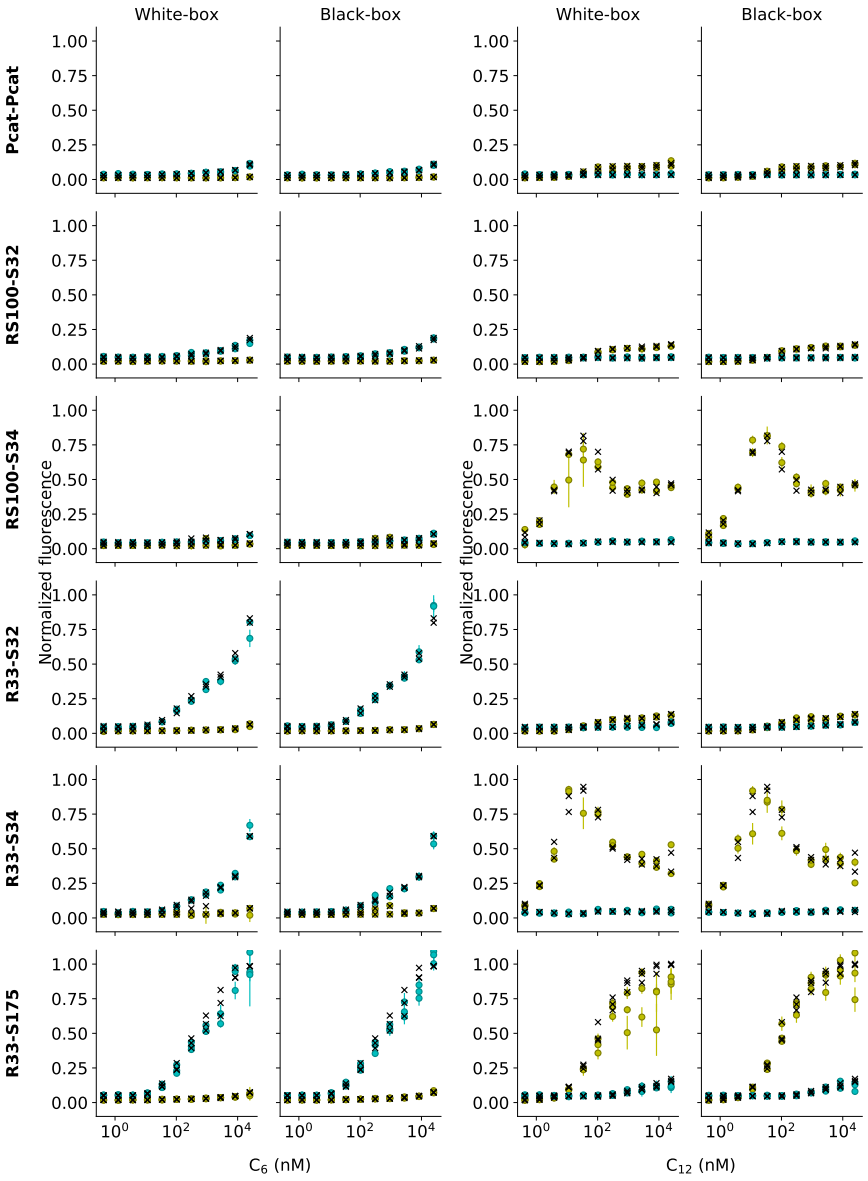

Figure S11: Treatment response plots

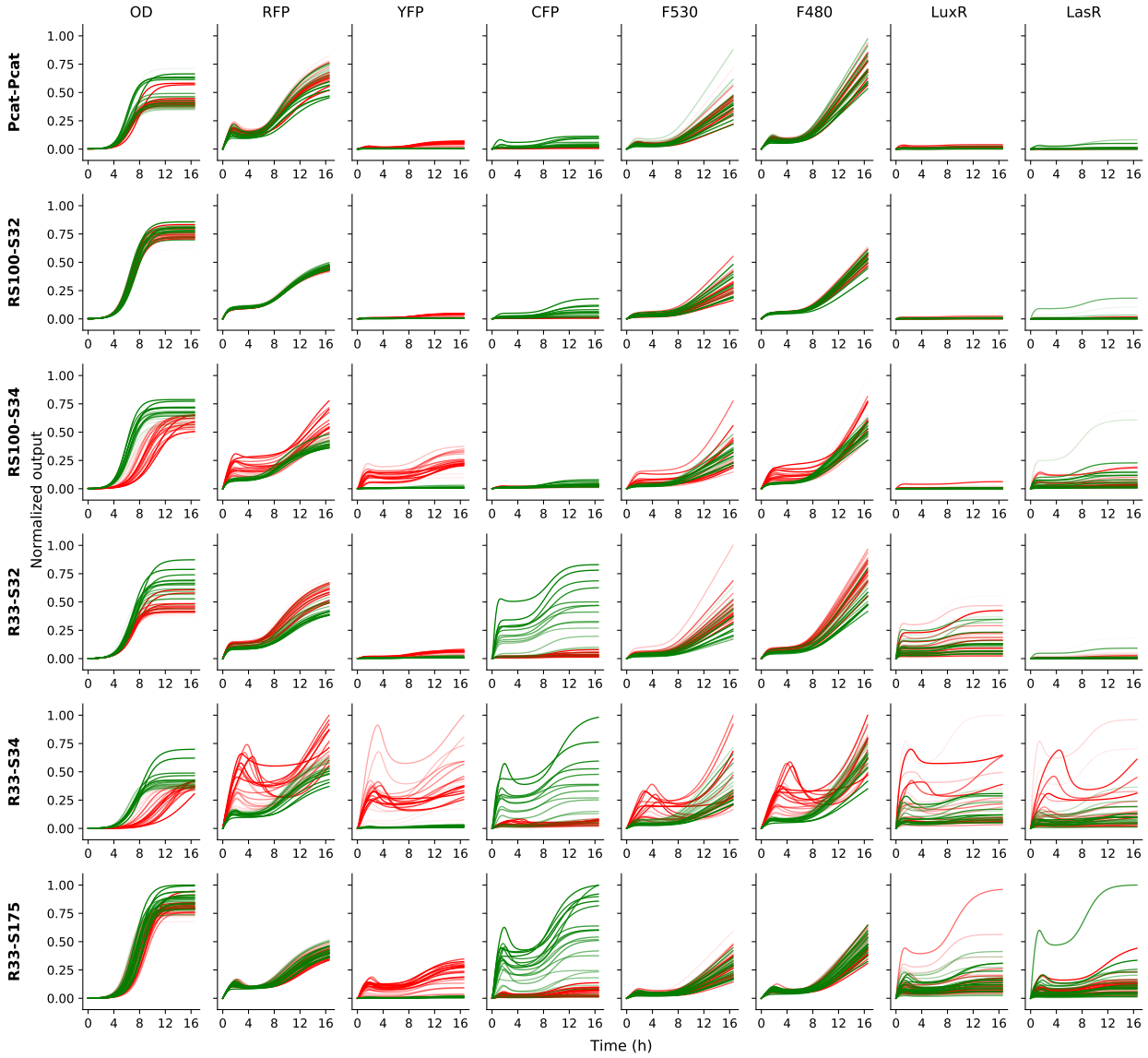

Figure S12: Predicted concentration dynamics of the hidden species (**x**) in the white-box model. In each panel, the traces are coloured according to whether they correspond to cultures treated with C_6 (green) or C_{12} (red).

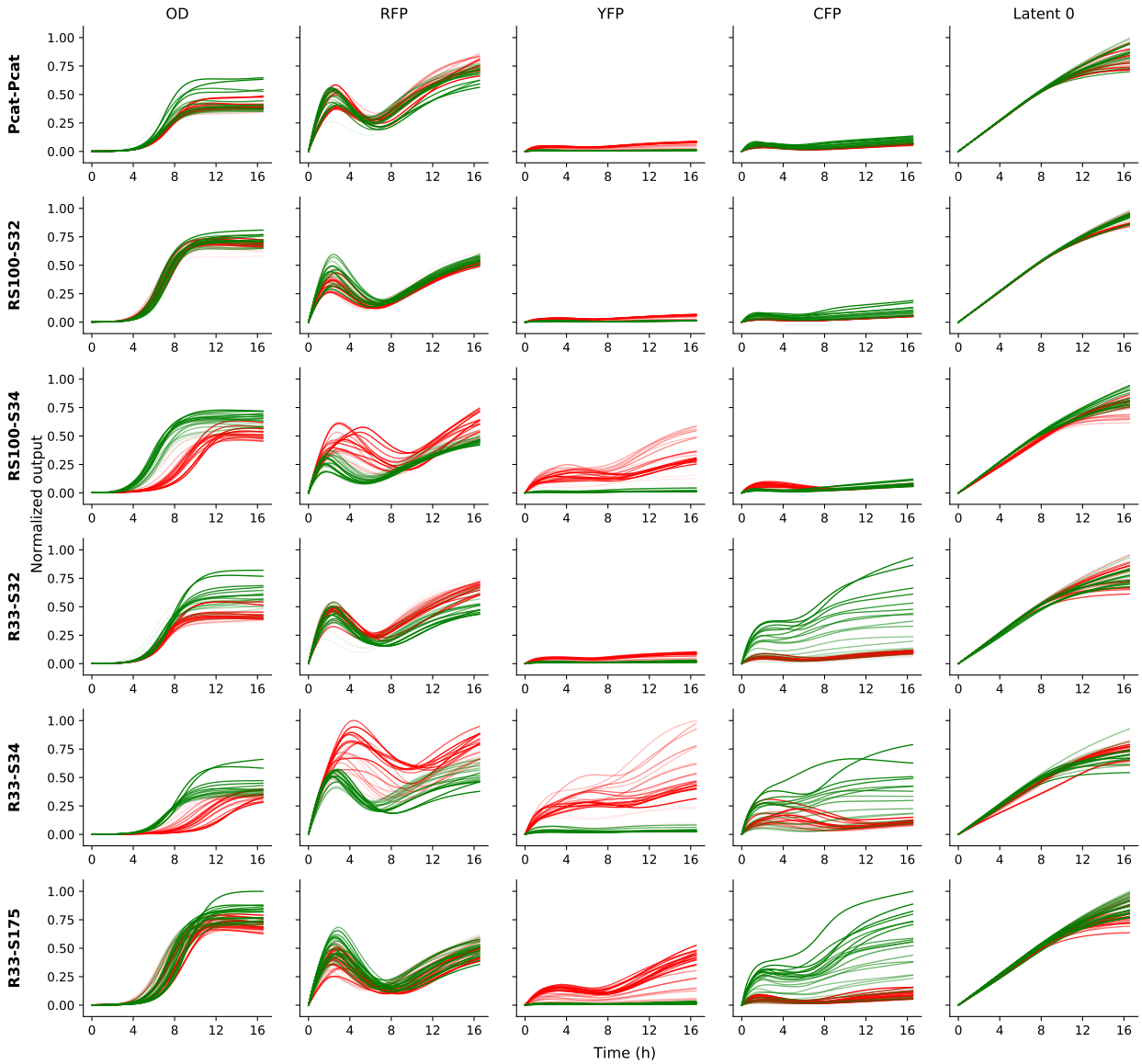

Figure S13: Predicted concentration dynamics of the hidden species (**x**) in the black-box model. In each panel, the traces are coloured according to whether they correspond to cultures treated with C_6 (green) or C_{12} (red).

References

Ainsworth, S., Foti, N., Lee, A. K., and Fox, E. Interpretable vaes for nonlinear group factor analysis. *arXiv preprint arXiv:1802.06765*, 2018.

Archer, E., Park, I. M., Buesing, L., Cunningham, J., and Paninski, L. Black box variational inference for state space models. *arXiv preprint arXiv:1511.07367*, 2015.

Balagaddé, F. K., Song, H., Ozaki, J., Collins, C. H., Barnet, M., Arnold, F. H., Quake, S. R., and You, L. A synthetic escherichia coli predator–prey ecosystem. *Molecular systems biology*, 4(1):187, 2008.

Chen, T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, 2018.

Chen, Y., Kim, J. K., Hirning, A. J., Josić, K., and Bennett, M. R. Emergent genetic oscillations in a synthetic microbial consortium. *Science*, 349(6251): 986–989, 2015.

Dalchau, N., Grant, P. K., Vaidyanathan, P., Gravill, C., Spaccasassi, C., and Phillips, A. Scalable dynamic characterization of synthetic gene circuits. *bioRxiv*, pp. 635672, 2019.

Daniel, R., Rubens, J. R., Sarpeshkar, R., and Lu, T. K. Synthetic analog computation in living cells. *Nature*, 497(7451):619–623, May 2013.

Gorbach, N. S., Bauer, S., and Buhmann, J. M. Scalable variational inference for dynamical systems. In *Advances in Neural Information Processing Systems*, pp. 4806–4815, 2017.

Grant, P. K., Dalchau, N., Brown, J. R., Federici, F., Rudge, T. J., Yordanov, B., Patange, O., Phillips, A., and Haseloff, J. Orthogonal intercellular signaling for programmed spatial behavior. *Mol. Syst. Biol.*, 12(1):849, Jan 2016.

Karlsson, M., Janzén, D. L., Durrieu, L., Colman-Lerner, A., Kjellsson, M. C., and Cedersund, G. Nonlinear mixed-effects modelling for single cell estimation: when, why, and how to use it. *BMC systems biology*, 9(1):52, 2015.

Krishnan, R. G., Shalit, U., and Sontag, D. Deep kalman filters. *arXiv preprint arXiv:1511.05121*, 2015.

Nielsen, A. A., Der, B. S., Shin, J., Vaidyanathan, P., Paralanov, V., Strychalski, E. A., Ross, D., Densmore, D., and Voigt, C. A. Genetic circuit design automation. *Science*, 352(6281):aac7341, Apr 2016.

Pontryagin, L. S. *Mathematical theory of optimal processes*. Routledge, 2018.

Ranganath, R., Tran, D., and Blei, D. Hierarchical variational models. In *International Conference on Machine Learning*, pp. 324–333, 2016.

Ryder, T., Golightly, A., McGough, A. S., and Prangle, D. Black-box variational inference for stochastic differential equations. *arXiv preprint arXiv:1802.03335*, 2018.

Sisson, S. A., Fan, Y., and Tanaka, M. M. Sequential monte carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 104(6):1760–1765, 2007.

Toni, T., Welch, D., Strelkowa, N., Ipsen, A., and Stumpf, M. P. Approximate bayesian computation scheme for parameter inference and model selection in dynamical systems. *Journal of the Royal Society Interface*, 6(31):187–202, 2009.

Xun, X., Cao, J., Mallick, B., Maity, A., and Carroll, R. J. Parameter estimation of partial differential equation models. *Journal of the American Statistical Association*, 108(503):1009–1020, 2013.