Contents

1 Background

A circuit must be robust to a fluctuating cellular environment and its response and sensitivity must be able to be fine tuned in order to orchestrate a network of circuits that function together. A robust circuit can tolerate the compound stochasticity that a chain of circuits brings, and fine tuning of its response and sensitivity enables the researcher to make it sensitive to an upstream signal as well as influence a downstream subsystem. Parts can be fine tuned by developing component libraries (Lu, Khalil, & Collins 2009), but this will be of little use if the required parameter ranges for parts to make a functional complex network are unknown, and will only perpetuate the cycles of trial-and-error. A computational method to find the range of parameter values that will produce the behaviour of choice is crucial to the design process by enabling the informed selection of appropriate parts from the libraries. For example, if it is known that gene expression must be low for a given stability, one can select a weak promoter or a low copy plasmid for the desired construct.

Both analytical and computational approaches have been deployed for the study of the toggle switch. Analytical approaches are limited to simpler models and thus require a number of assumptions to be made. The system under consideration has to be reduced to very few equations and parameters in order to make the system solvable. This requires assumptions to be made about the system that cannot always be justified, such as the quasi-steady state approximation (QSSA). The QSSA assumes that the binding/unbinding processes are much faster than any other process (Loinger et al. 2007), thus the bound intermediate is assumed to always be in steady state. The QSSA assumption is met *in vitro* but often does not hold *in vivo* and its misuse can lead to large errors and incorrectly estimated parameters (Pedersen, Bersani, & Bersani 2007). Moreover, it is generally not possible to solve even simple stochastic models analytically, and these methods are restricted to deterministic models. The computational and graph-theoretic approaches developed for the study of multistationarity generally focus on deciding on whether a given system is incapable of producing multiple steady states (Conradi et al. 2007; Banaji

& Craciun 2010; Feliu & Wiuf 2013). For example, Feliu & Wiuf (2013) developed an approach using chemical reaction theory and generalised mass action modelling (Feliu & Wiuf 2013). No approach exists that can handle both deterministic and stochastic systems in an integrated manner.

For this purpose, I developed a computational framework based on sequential Monte Carlo that takes a model and determines whether it is capable of producing a given number of (stable) steady states and the parameter space that gives rise to the behaviour. Uniquely, this can be done for both deterministic and stochastic models, and also complex models with many parameters, thus removing the need for simplifying assumptions. This framework can be used for comparing the conclusions drawn by various modelling approaches and thus provides a way to investigate appropriate abstractions. I have made this framework into a python package, called Stability Finder.

I use this methodology to investigate genetic toggle switches and uncover the design principles behind making a bistable switch, as well as those necessary to make a tristable and a quadristable switch (4 steady states). I also demonstrate the ability of Stability Finder to examine more complex systems and examine the design principles of a three gene switch. The examples I used demonstrate that Stability Finder will be a valuable tool in the future design and construction of novel gene networks.

1.0.1 ABC algorithms

Stability Finder is based on a statistical inference method which combines Approximate Bayesian Computation (ABC) with Sequential Monte Carlo (SMC) (Toni et al. 2009). This simulation-based method uses an iterative process to arrive at a distribution of parameter values that can give rise to observed data or a desired system behaviour (Barnes et al. 2011).

ABC methods are used for inferring the posterior distribution in cases where it is too computationally expensive to evaluate the likelihood function. Instead of calculating the likelihood, ABC methods simulate the data and then compare the simulated and observed data through a distance function (Toni et al. 2009). Given the prior distribution $\pi(\theta)$ we can approximate the posterior distribution, $\pi(\theta \mid x) \propto f(x \mid \theta)\pi(\theta)$, where $f(x \mid \theta)$ is the likelihood of a parameter, θ, given the data, x. There are a number of different variations of the ABC algorithm depending on how the the approximate posterior distribution is sampled.

The simplest ABC algorithm is the ABC rejection sampler (Pritchard et al. 1999). In this method, parameters are sampled from the prior and data simulated through the data generating model. For each simulated data set, a distance from that of the desired behaviour is calculated, and if greater than a threshold, ε, the sample is rejected, otherwise it is accepted.

Algorithm 1 ABC rejection algorithm

1: Sample a parameter vector θ from prior $\pi(\theta)$
2: Simulate the model given θ
3: Compare the simulated data with the desired data, using a distance function d and tolerance ε. if $d \leq \varepsilon$, accept θ

The main disadvantage of this method is that if the prior distribution is very different from the posterior, the acceptance rate is very low (Toni et al. 2009). An alternative method is the ABC Markov Chain Monte Carlo (MCMC) developed by Marjoram et al. (2003). The disadvantage of this method is that if it gets stuck in an area of low probability it can be very slow to converge (Sisson, Fan, & Tanaka n.d.).

The method used here is based on Sequential Monte Carlo, which avoids both issues faced by the rejection and MCMC methods. It propagates the prior through a series of intermediate distributions in order to arrive at an approximation of the posterior. The tolerance, ε, for the distance of the simulated data to the desired data is made smaller at each iteration. When ε is sufficiently small, the result will approximate the posterior distribution (Toni et al. 2009).

ABC SMC can identify the parameter values within a predefined range of values that can achieve the desired behaviour. It works by first sampling at random from the initial range set by the user, i.e. form the prior distribution of values. Each sample from the priors is called a particle. It then simulates the model given those values and compares that to the target behaviour. If the distance between the simulation and the target behaviour is greater than a predefined threshold distance ε, then the parameter values that produced that simulation are rejected. This is repeated for a predefined number of samples which are collectively referred to as a population. Each particle in a population has a weight associated with it, which represents the probability of it producing the desired behaviour. At subsequent iterations the new samples are obtained from the previous populations and the ε is set to smaller value, thus eventually reaching the desired behaviour. The algorithm proceeds as follows:

Algorithm 2 ABC SMC algorithm

1: Select ε and set population t = 0
2: Sample particles (θ). If t = 0, sample from prior distributions (P). If t > 0, sample particles from previous population.
3: If t > 0: Perturb each particle by ± half the range of the previous population (j) to obtain new perturbed population (i).
4: Simulate each particle to obtain time course.
5: Reject particles if d > ε.
6: Calculate the weight for each accepted particle. At the first population assign a weight equal to 1 for all particles. In subsequent populations the weight of a particle is equal to the probability of observing that particle divided by the sum of the probabilities of the particle arising from each of the particles in the previous population:
7: $w_t^{(i)} = \begin{cases} 1, & \text{if } n = 0 \\ \frac{P(\theta_t^{(i)})}{\sum_{j=1}^{N} w_{t-1}^{(j)} K_t(\theta_{t-1}^{(j)}, \theta_t^{(i)})}, & \text{if } n>0. \end{cases}$

This algorithm is implemented on a simple example for illustration. A simple model was used, consisting of one species, A converting to another, B. The model is described by two differential equations, where A is the reactant and B the product, produced at a rate p.

$$\frac{d[B]}{dt} = p[A] \tag{1.1}$$

$$\frac{d[A]}{dt} = -p[A] \tag{1.2}$$

The priors were set to $p \sim U(0, 10)$. Initial conditions for A and B were set to 1 and 0 respectively. The data to which the model was compared to was generated by simulating the same model with the parameter set to 1, as shown in Figure 1.1.

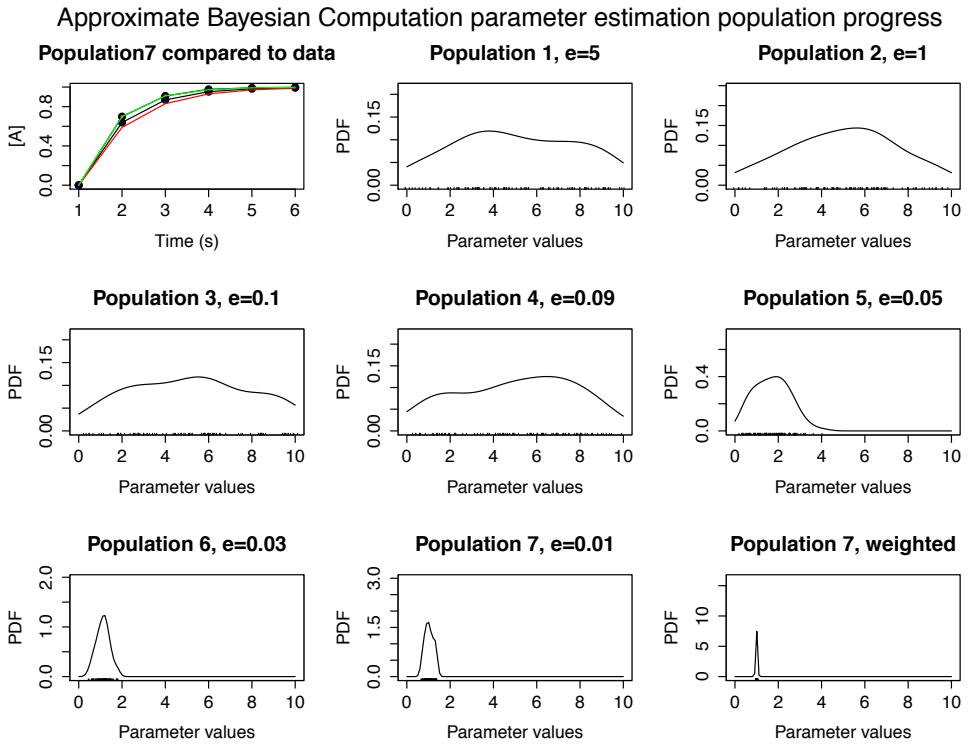

Figure 1.1 ABC SMC parameter inference. The posterior parameter is equal to 1 and its time course shown in red in the top left panel. The blue time course is that of the final population, green is the upper quartile and red is the lower quartile range of values. The progress of the selection process can be seen the εschedule proceeds from the top left to the bottom right. The bottom far right panel is a density plot of ε= 0.01 with their weights taken into account.

Figure 1.1 demonstrates, using a simple example, that ABC SMC is capable of fitting a model to the data. During the course of 7 populations, the accepted distance ε of the simulated particles to the data is incrementally decreased. This leads to a final population where the distance of the data to the particles is very small, and there is a good agreement between the two. The algorithm concludes with a set of parameter values that produced this behaviour, which approximate the posterior distribution. The posterior distribution found in this model is in good agreement with the parameter value used to generate the data. This example successfully demonstrates the effectiveness of the ABC SMC algorithm in fitting models to data.

1.0.2 Robustness

During this thesis I define robustness as the ability of a system to retain its function despite parameter perturbations (Stelling et al. 2004). The robustness of biological systems has been studied extensively (Barkai & Leibler 1997; Stelling et al. 2004; Prill, Iglesias, & Levchenko 2005; Kim et al. 2006; Kitano 2007; Hafner et al. 2009; Shinar & Feinberg 2010; Zamora-Sillero et al. 2011; Woods et al. 2015). and it is well known that feedback loops can increase the robustness of a system (**Doyle:2005ul**; Becskei & Serrano 2000).

The robustness of a model can be calculated by dividing the volume of its functional region by the volume of its priors. This is a measure of the volume of the posterior distribution is compared to the priors. It comes from Bayes' rule that:

$$f(\theta|x) = \frac{f(\theta)f(x|\theta)}{\int p(x|\theta)p(\theta)d\theta} \tag{1.3}$$

where $p(x|\theta)$ is the likelihood, $p(\theta)$ is the prior, and $\int p(x|\theta)p(\theta)d\theta$ is the evidence. The evidence is the normalisation added so that the distribution integrates to 1. For a given model design D and objective O we define the functional region F as the region within the prior where O is satisfied. So within the prior we can assign 1 to any region that falls within F and 0 to any region outside that.

$$p(O|D_1) = \int p(O|\theta, D_1)p(\theta|D_1)d\theta, \tag{1.4}$$

For a design with three parameters this becomes:

$$p(O|D_1) = \iiint_{\underline{\Theta}} p(O|\underline{\Theta})p(\underline{\Theta}|D_1)d\underline{\Theta}, \tag{1.5}$$

where $\underline{\Theta}$ is a vector containing the three parameters = $\theta_1, \theta_2, \theta_3$. To calculate the robustness, or model evidence, we integrate this with respect to $\underline{\Theta}$. We assume all parameters $\theta_1, \theta_2, \theta_3$ are uniform, $p(\underline{\Theta}|D_1) \sim U(a, b)$. If we assume a = 0 this integral becomes:

$$p(O|D_1) = \iiint_{\underline{\Theta}} p(O|\underline{\Theta}) \frac{1}{b_1}\frac{1}{b_2}\frac{1}{b_3} d\underline{\Theta} \text{ , and} \tag{1.6}$$

$$p(O|D_1) = \frac{1}{b_1}\frac{1}{b_2}\frac{1}{b_3} \iiint_{\underline{\Theta}} p(O|\underline{\Theta}) d\underline{\Theta} \tag{1.7}$$

since $\frac{1}{b_1}\frac{1}{b_2}\frac{1}{b_3}$ is a constant. Then assuming that the likelihood is uniform Equation1.7 becomes:

$$p(O|D_1) = \frac{1}{b_1}\frac{1}{b_2}\frac{1}{b_3}\Big[\iiint_{\underline{\Theta}_F} 1 d\underline{\Theta} + \cancelto{0}{\iiint_{\underline{\Theta}_{\not F}} 0 d\underline{\Theta}} \Big] \tag{1.8}$$

$$\tag{1.9}$$

since we assign 1 to any region within F and 0 to any region outside it. This becomes:

$$p(O|D_1) = \overbrace{\frac{1}{b_1}\frac{1}{b_2}\frac{1}{b_3}}^{|P|} \underbrace{\iiint_{\underline{\Theta}_F} 1 d\underline{\Theta}}_{|F|}, \tag{1.10}$$

$$\therefore p(O|D_1) = \frac{|F|}{|P|}, \tag{1.11}$$

where |P| is the volume of the prior P and |F| the volume of the functional region F. Therefore, in the case where both the prior and the likelihood are uniform, the robustness R of the design is the ratio of the volumes of the two.

If on the other hand we assume the likelihood is multivariate normal, with priors remaining uniform, Equation1.7 becomes:

$$p(O|D_1) = \frac{1}{|P|} \iiint_{\underline{\Theta}} f(\underline{\Theta}; \mu, \Sigma) d\underline{\Theta} \tag{1.12}$$

$$\therefore p(O|D_1) = \frac{1}{|P|} \underbrace{\times (2\pi)^{\frac{k}{2}} \times |\Sigma|^{\frac{1}{2}}}_{|F|} \tag{1.13}$$

$$\therefore p(O|D_1) = \frac{|F|}{|P|}, \tag{1.14}$$

We can use the Bayes' factor in order to compare the robustness between two model designs. The Bayes' factor is defined as follows:

$$B_{ab} = \frac{\int p(x|\theta, D_a)p(\theta, D_a)d\theta}{\int p(x|\theta, D_b)p(\theta, D_b)d\theta} \tag{1.15}$$

$$\therefore B_{ab} = \frac{|Fa|}{|Pa|} / \frac{|Fb|}{|Pb|} \tag{1.16}$$

Therefore, we can use the ratio of the two robustness measures to calculate the Bayes' factor. If two models have a different number of parameters, the robustness of the system will only increase if |F| increases by more than the proportion by which |P| increased (Woods et al. 2015). A model will be penalised for an additional if it does not increase the volume of the functional region by more than the volume that the added parameter added to the prior. This is true for nested models, where one model is wholly contained in the other.

1.1 Contents of this thesis

Bibliography

Banaji, M. & Craciun, G. (2010). 'Graph-theoretic criteria for injectivity and unique equilibria in general chemical reaction systems'. *Advances in Applied Mathematics* **44**(2), 168–184.

Barkai, N. & Leibler, S. (1997). 'Robustness in simple biochemical networks.' *Nature* **387**(6636), 913–917.

Barnes, C. P., Silk, D., Sheng, X., & Stumpf, M. P. H. (2011). 'Bayesian design of synthetic biological systems.' *Proceedings of the National Academy of Sciences of the United States of America* **108**(37), 15190–15195.

Becskei, A. & Serrano, L. (2000). 'Engineering stability in gene networks by autoregulation.' *Nature* **405**(6786), 590–593.

Conradi, C., Flockerzi, D., Raisch, J., & Stelling, J. (2007). 'Subnetwork analysis reveals dynamic features of complex (bio)chemical networks'. *PNAS* **104**(49), 19175–19180.

Feliu, E. & Wiuf, C. (2013). 'A computational method to preclude multistationarity in networks of interacting species.' *Bioinformatics (Oxford, England)* **29**(18), 2327–2334.

Hafner, M., Koeppl, H., Hasler, M., & Wagner, A. (2009). ''Glocal' Robustness Analysis and Model Discrimination for Circadian Oscillators'. *PLoS Computational Biology* **5**(10), e1000534.

Kim, J., Bates, D. G., Postlewaite, I., Ma, L., & Iglesias, P. A. (2006). 'Robustness analysis of biochemical network models'. *Systems biology*.

Kitano, H. (2007). 'Towards a theory of biological robustness'. *Molecular systems biology* **3**.

Loinger, A., Lipshtat, A., Balaban, N. Q., & Biham, O. (2007). 'Stochastic simulations of genetic switch systems'. *Physical Review E*.

Lu, T. K., Khalil, A. S., & Collins, J. J. (2009). 'Next-generation synthetic gene networks'. *Nature Biotechnology* **27**(12), 1139–1150.

Marjoram, P., Molitor, J., Plagnol, V., & Tavare, S. (2003). 'Markov chain Monte Carlo without likelihoods.' *Proceedings of the National Academy of Sciences of the United States of America* **100**(26), 15324–15328.

Pedersen, M. G., Bersani, A. M., & Bersani, E. (2007). 'Quasi steady-state approximations in complex intracellular signal transduction networks – a word of caution'. *Journal of Mathematical Chemistry* **43**(4), 1318–1344.

Prill, R. J., Iglesias, P. A., & Levchenko, A. (2005). 'Dynamic properties of network motifs contribute to biological network organization.' *PLoS Biology* **3**(11), e343–e343.

Pritchard, J. K., Seielstad, M. T., Perez-Lezaun, A., & Feldman, M. W. (1999). 'Population growth of human Y chromosomes: A study of Y chromosome microsatellites'. *Molecular Biology and Evolution* **16**(12), 1791–1798.

Shinar, G. & Feinberg, M. (2010). 'Structural Sources of Robustness in Biochemical Reaction Networks'. *Science* **327**(5971), 1389–1391.

Sisson, S. A., Fan, Y., & Tanaka, M. M. 'Sequential Monte Carlo without likelihoods'.

Stelling, J., Sauer, U., Szallasi, Z., Doyle, F. J., & DOYLE, J. (2004). 'Robustness of cellular functions'. *Cell* **118**(6), 675–685.

Toni, T., Welch, D., Strelkowa, N., Ipsen, A., & Stumpf, M. P. H. (2009). 'Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems.' *Journal of the Royal Society, Interface / the Royal Society* **6**(31), 187–202.

Woods, M., Leon, M., Perez-Carrasco, R., & Barnes, C. P. (2015). 'A statistical approach reveals designs for the most robust stochastic gene oscillators'. *bioRxiv*, 025056.

Zamora-Sillero, E., Hafner, M., Ibig, A., Stelling, J., & Wagner, A. (2011). 'Efficient characterization of high-dimensional parameter spaces for systems biology.' *BMC systems biology* **5**, 142.