

# Tutorial on (Computational Systems) Biological Models

**Darren Wilkinson**

School of Mathematics & Statistics

and

Centre for Integrated Systems Biology of Ageing and Nutrition  
(CISBAN)

Newcastle University, UK

SAMSI Kickoff Workshop, RTP, NC, USA, September 10–14  
2006

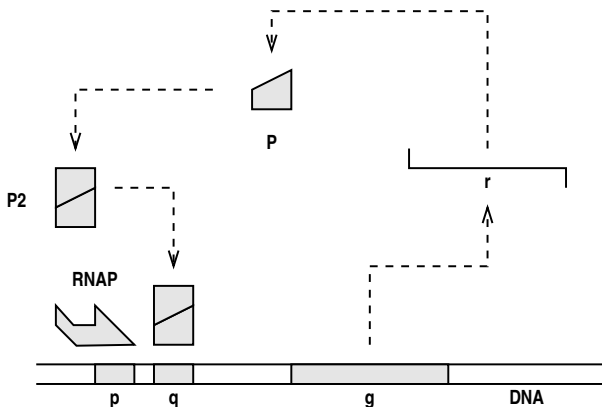
# Computational Systems Biology (CSB)

- Much of CSB is concerned with building models of complex biological pathways, then validating and analysing those models using a variety of methods, including time-course simulation
- Most CSB researchers work with continuous deterministic models (coupled ODE and DAE systems)
- There is increasing evidence that much intra-cellular behaviour (including gene expression) is intrinsically stochastic, and that systems cannot be properly understood unless stochastic effects are incorporated into the models
- Stochastic models are harder to build, estimate, validate, analyse and simulate than deterministic models...

# Modelling

- Start with some kind of picture or diagram for a mechanism
- Turn it into a set of (pseudo-)biochemical reactions
- Specify the rate laws and rate parameters of the reactions
- Run some stochastic or deterministic computer simulator of the system dynamics
- Study the dynamics in a variety of ways to gain insight into the system

## Example — genetic auto-regulation



# Biochemical reactions

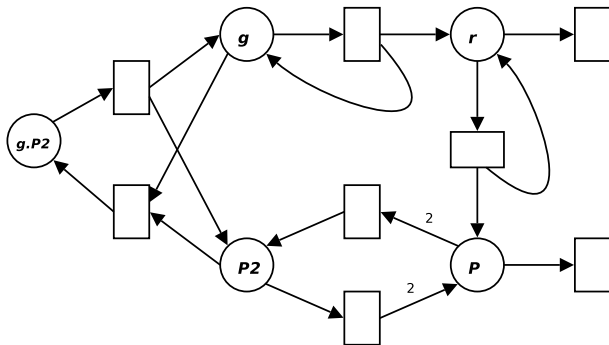
Simplified view:

## Reactions

$g + P_2 \longleftrightarrow g \cdot P_2$	Repression
$g \longrightarrow g + r$	Transcription
$r \longrightarrow r + P$	Translation
$2P \longleftrightarrow P_2$	Dimerisation
$r \longrightarrow \emptyset$	mRNA degradation
$P \longrightarrow \emptyset$	Protein degradation

But these aren't as nice to look at as the picture...

# Petri net representation



Simple bipartite digraph representation of the reaction network —  
 useful both for visualisation and computational analysis

# Matrix representation of the Petri net

Species	Reactants ( <i>Pre</i> )					Products ( <i>Post</i> )				
	$g \cdot P_2$	$g$	$r$	$P$	$P_2$	$g \cdot P_2$	$g$	$r$	$P$	$P_2$
Repression		1			1	1				
Reverse repression	1						1			1
Transcription		1					1	1		
Translation			1					1	1	
Dimerisation				2						1
Dissociation					1				2	
mRNA degradation			1							
Protein degradation				1						

But still need rate laws and reaction rates...

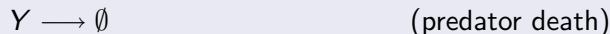
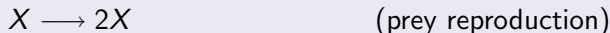
# Mass-action stochastic kinetics

Stochastic molecular approach:

- Statistical mechanics arguments lead to a Markov jump process in continuous time whose instantaneous reaction rates are directly proportional to the number of molecules of each reacting species
- Such dynamics can be simulated (exactly) on a computer using standard discrete-event simulation techniques
- Standard implementation of this strategy is known as the “Gillespie algorithm” (just discrete event simulation), but there are several exact and approximate variants of this basic approach

# Lotka-Volterra system

## Reactions



- $X$  – Prey,  $Y$  – Predator
- We can re-write this using matrix notation for the corresponding Petri net

# Forming the matrix representation

## The L-V system in tabular form

	Rate Law $h(\cdot, c)$	LHS		RHS		Net-effect	
		X	Y	X	Y	X	Y
$R_1$	$c_1x$	1	0	2	0	1	0
$R_2$	$c_2xy$	1	1	0	2	-1	1
$R_3$	$c_3y$	0	1	0	0	0	-1

Call the  $3 \times 2$  net-effect (or **reaction**) matrix  $A$ . The matrix  $S = A'$  is the **stoichiometry matrix** of the system. Typically both are **sparse**. The SVD of  $S$  (or  $A$ ) is of interest for structural analysis of the system dynamics...

# Petri net invariants

- A  $P$ -invariant is a non-zero solution to  $Ay = 0$  (ie.  $y$  is in the null-space of  $A$ )
  - $P$ -invariants correspond to **conservation laws** in the network, and lead to rank-degeneracy of  $A$
- A  $T$ -invariant is a non-zero, non-negative (integer-valued) solution to  $Sx = 0$  (ie.  $x$  is in the null-space of  $S$ )
  - $T$  invariants correspond to sequences of reaction events that return the system to its original state
- The SVD of  $S$  (or  $A$ ) characterises the null-space of  $S$  and  $A$
- The Lotka-Volterra model is of full rank (so no  $P$ -invariants), and has one  $T$ -invariant,  $x = (1, 1, 1)'$

# The Gillespie algorithm

- 1 Initialise the system at  $t = 0$  with rate constants  $c_1, c_2, \dots, c_v$  and initial numbers of molecules for each species,  $x_1, x_2, \dots, x_u$ .
- 2 For each  $i = 1, 2, \dots, v$ , calculate  $h_i(x, c_i)$  based on the current state,  $x$ .
- 3 Calculate  $h_0(x, c) \equiv \sum_{i=1}^v h_i(x, c_i)$ , the combined reaction hazard.
- 4 Simulate time to next event,  $t'$ , as an  $\text{Exp}(h_0(x, c))$  random quantity, and put  $t := t + t'$ .
- 5 Simulate the reaction index,  $j$ , as a discrete random quantity with probabilities  $h_i(x, c_i) / h_0(x, c)$ ,  $i = 1, 2, \dots, v$ .
- 6 Update  $x$  according to reaction  $j$ . That is, put  $x := x + S^{(j)}$ , where  $S^{(j)}$  denotes the  $j$ th column of the stoichiometry matrix  $S$ .
- 7 Output  $x$  and  $t$ .
- 8 If  $t < T_{max}$ , return to step 2.

# The continuous deterministic approximation

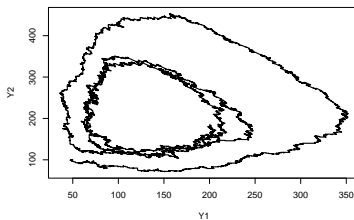
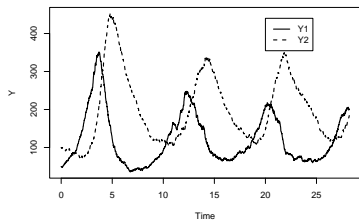
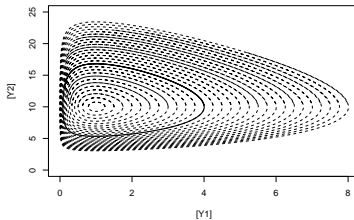
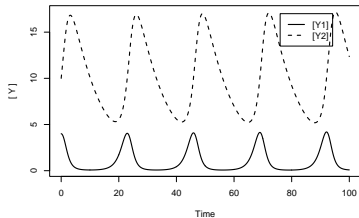
- If the discreteness and stochasticity are ignored, then by considering the reaction fluxes it is straightforward to deduce the mass-action ordinary differential equation (ODE) system:

## ODE Model

$$\frac{dX_t}{dt} = Sh(X_t, c)$$

- Analytic solutions are rarely available, but good numerical solvers can generate time course behaviour
- Slight complications due to rank-degeneracy of  $S$
- Also spatial versions — reaction-diffusion kinetics — PDE models — computationally intensive (slow)

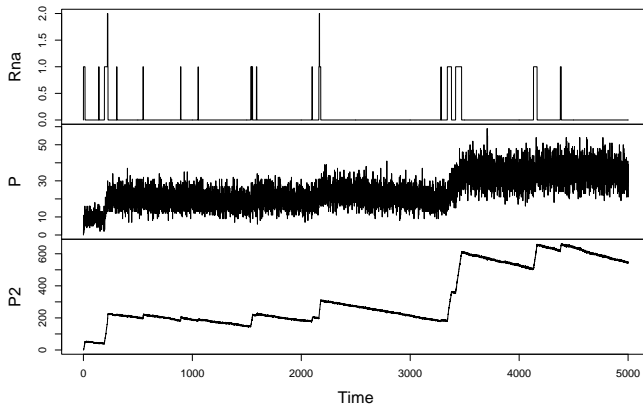
# The Lotka-Volterra model



## Key differences

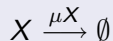
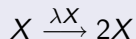
- Deterministic solution is exactly periodic with perfectly repeating oscillations, carrying on indefinitely
- Stochastic solution oscillates, but in a random, unpredictable way (wandering from orbit to orbit in phase space)
- Stochastic solution **will** end in disaster! Either prey or predator numbers will hit zero...
- Either way, predators will end up extinct, so **expected** number of predators will tend to zero — **qualitatively different** to the deterministic solution
- So, in general the deterministic solution does not provide reliable information about either the stochastic process or its average behaviour

# Simulated realisation of the auto-regulatory network



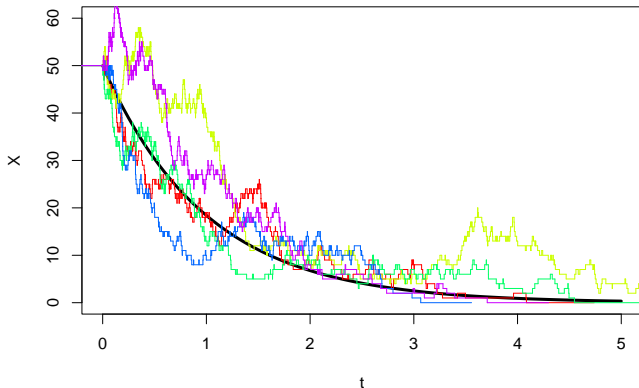
# Simple example: linear birth-death process

## Birth-death reactions



- Deterministic solution:  $X_t = X_0 \exp\{(\lambda - \mu)t\}$
- This is a function of  $(\lambda - \mu)$  only!
- Stochastic solution is more interesting, and depends on both  $\lambda$  and  $\mu$ ...

# Birth-death realisations

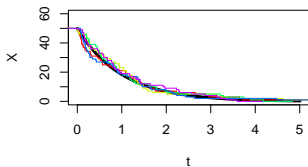


## Issues with the birth-death process

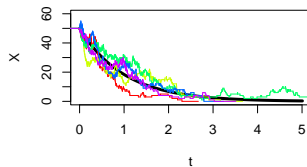
- Stochastic variation: random distribution at each time point, correlations between time points, random time to extinction, etc.
- Parameter identification: if a deterministic model is fitted, one can only **ever** identify  $(\lambda - \mu)$  — never  $\lambda$  and  $\mu$  separately
- Information about **both**  $\lambda$  and  $\mu$  in the data...
- Need **both**  $\lambda$  and  $\mu$  for reliable stochastic simulation
- Deterministic parameter fitting algorithms are likely to lead to an underestimate of true intrinsic noise levels

# Birth-death realisations

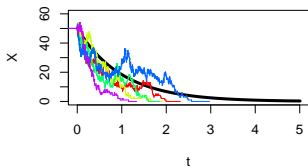
$\lambda=0, \mu=1$



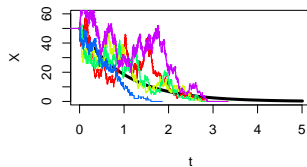
$\lambda=3, \mu=4$



$\lambda=7, \mu=8$



$\lambda=10, \mu=11$



## Fully Bayesian inference

- In principle it is possible to carry out rigorous statistical inference for the parameters of the stochastic process model
- Fairly detailed experimental data are required — eg. quantitative single-cell time-course data derived from live-cell imaging
- The standard procedure uses GFP labelling of key reporter proteins together with time-lapse confocal microscopy, but other approaches are also possible
- Techniques for exact inference for the true discrete model (Boys, Wilkinson, Kirkwood 2004) do not scale well to problems of realistic size and complexity, due to the difficulty of efficiently exploring large complex integer lattice state spaces

## Diffusion approximations for inference

Joint work with PhD student: Andrew Golightly

Bayesian sequential inference for nonlinear multivariate diffusions

- True process is discrete and stochastic — stochasticity is vital — what about discreteness?
- Apply the Fokker-Planck equation to the Master equation for the true process to obtain an SDE known as the Chemical Langevin Equation (CLE)
- The approximation isn't entirely satisfactory for simulation, but seems OK for inference...

# The stochastic-kinetic diffusion approximation

## Chemical Langevin Equation (Itô SDE)

$$dX_t = Sh(X_t, c)dt + [S \text{diag}\{h(X_t, c)\}S']^{1/2} dW_t$$

- Fairly general class of non-linear multivariate SDEs
- The stoichiometry matrix  $S = (Post - Pre)'$  is typically rank-degenerate, which complicates things slightly
- $S$  is known and  $X$  (or a subset) is observed at discrete times (subject to error)
- Inference is for  $c$  (the vector of rate constants parameterising the reaction rate vector,  $h(\cdot, \cdot)$ )

## Inference for diffusions

Inference for very general non-linear multivariate diffusion processes observed partially and discretely (and with error)

- Idea: Use an MCMC algorithm which “fills-in” the missing diffusion bridges between successive observations
- Use an Euler approximation to the true diffusion, but on a much finer scale than the data
- There are pathological mixing/convergence problems for regular MCMC schemes as the discretisation gets finer (essentially, there is an infinite amount of information about the parameters in the augmentation)

## Likelihood concepts

- Putting  $\mu(x, c) = Sh(x, c)$  and  $\beta(x, c) = S \text{diag}\{h(x, c)\}S'$ :

$$dX_t = \mu(X_t, c)dt + \sqrt{\beta(X_t, c)}dW_t$$

- If we choose a small enough  $\Delta t$ , we get the Euler-Maruyama approximation

$$X_{t+\Delta t} | X_t, c \sim N(X_t + \mu(X_t, c)\Delta t, \beta(X_t, c)\Delta t)$$

- Perfect observation of the system state on this time grid leads to the “complete”-data likelihood

$$L(c; x) \propto \left\{ \prod_{i=0}^{n-1} |\beta(x_{i\Delta t}, c)|^{-1/2} \right\} \times \exp \left\{ -\frac{1}{2} \sum_{i=0}^{n-1} \left( \frac{\Delta x_{i\Delta t}}{\Delta t} - \mu(x_{i\Delta t}, c) \right)' \beta(x_{i\Delta t}, c)^{-1} \left( \frac{\Delta x_{i\Delta t}}{\Delta t} - \mu(x_{i\Delta t}, c) \right) \Delta t \right\}$$

## Likelihood problems

- Unfortunately the likelihood has no limit as  $\Delta t \rightarrow 0$
- **If** the diffusion term  $\beta(x, c)$  were independent of  $c$ , then it **would** be possible to discard some terms and then get a nice limit (exponential of the sum of a regular integral and an Itô stochastic integral), but it isn't...
- This is at the root of all of the computational problems concerning inference for diffusions
- More formally, the issue is whether or not the relevant Radon-Nikodym derivatives exist...

# Gibbs sampler

- Impute  $m$  missing time points between every observation
- Choose  $m$  large enough to make Euler-Maruyama valid
- Cycle through parameters and all missing observations updating with a Metropolis proposal
- Easy to extend to partial observation and measurements observed with error
- **Problem:** mixing is **very** poor!

## Sequential MCMC particle filters

- As each new observation arrives, can construct an MCMC scheme which operates on a sample from the (current) prior to give a sample from the posterior given the new observation
- This posterior sample becomes the prior sample for the next observation
- Uses a clever technique due to Durham and Gallant for approximate simulation of non-linear diffusion bridges, which is then corrected with a Metropolis-Hastings step
- Easy to use with multiple partial data sets, and data subject to measurement error
- Overcomes the dependence problems!

## An efficient global MCMC scheme?

- Rather than use a sequential sampling procedure, try to “fix” the obvious block Gibbs sampling approach
- Would like to reparameterise the diffusion process so that the diffusion coefficient doesn't depend on the model parameters — a partially non-centred parameterisation
- Unfortunately it is generally impossible to construct such a transformation for nonlinear multivariate diffusions like the CLE

# The innovation scheme

- The *innovation scheme* (Chib, Pitt & Shephard, 2004) uses a numerical scheme (such as the Euler-Maruyama method) to numerically map between the observed sample paths and the driving Wiener process
- Then can operate a Gibbs sampler on  $(c, \mathbf{w})$  rather than  $(c, \mathbf{x})$
- This only works for indirectly observed diffusions (otherwise problems hitting data points), so requires adaptation for our scenario
- Key idea is to map between the sample paths and the Wiener process driving the modified diffusion bridge process — then sure to hit the data points

## Comparison of innovation scheme with sequential filter

- Both schemes work well and give consistent results, and neither break down for fine discretisations
- For relatively small data sets (less than 100 observations), the sequential filter seems more computationally efficient than the innovation scheme
- For large data sets, innovation scheme appears to be more reliable than the sequential scheme (degeneracy issues)
- Innovation scheme more robust to outliers
- Sequential scheme much more convenient to use in the context of multiple data sets

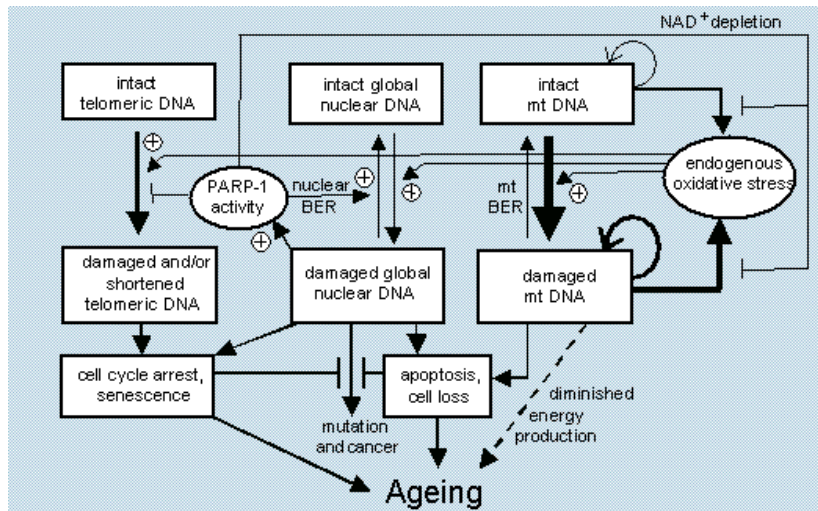
## Model-based inference (a summary)

- It is just about possible to do model-based inference for small simple models with high-quality data
- Very computationally intensive, and serious scalability issues
- Very hard to tailor methods to non-standard (more realistic) model variants (non-Markov, triggered events, external interventions, etc.)
- Very hard to tailor to non-standard data, integrate data from disparate sources, etc.
- Not realistic for any model that takes more than a couple of seconds to forward simulate (as an MCMC iterate takes at least as long as a forward simulation)

## Mechanisms of ageing

- Ageing is caused by the gradual accumulation of unrepaired molecular damage, leading to an increasing fraction of damaged cells and eventually to functional impairment of tissues and organs
- One major cause of molecular damage is highly reactive oxygen species (ROS)
- Molecular damage may trigger cellular response programmes, so that the ageing process may also be seen to be governed by genetically determined pathways
- Many of the (random) damage and (imperfect) repair mechanisms important for understanding cellular ageing are intrinsically stochastic

# Network theory of ageing



# Modelling large biological systems

BBSRC/MRC/DTI Grant (+ Unilever)

**BASIS — Biology of Ageing e-Science Integration and Simulation** (4/02–3/06) — Kirkwood, Wilkinson, Boys, Gillespie, Proctor, Shanley

- Modelling large complex systems with many interacting components
- SBML model database (SBML encoded for discrete stochastic simulation)
- Discrete stochastic simulation service (and results database)
- Distributed computing infrastructure for routine use (web portal and web-service interface for GRID computing)

# SBML — The Systems Biology Markup Language

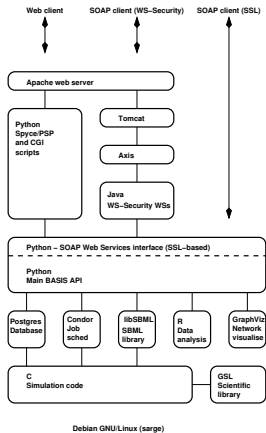
- SBML is an XML-based language for encoding and exchanging quantitative biochemical network models
- Encodes species, initial amounts, reactions, rate laws, etc.
- Original specification (Level 1) aimed mainly at continuous deterministic models
- Current specification (Level 2) perfectly capable of encoding discrete stochastic models in an unambiguous way
- Many tools for working with SBML models (model builders, deterministic and stochastic simulators, etc.)
- Issues with testing correctness of stochastic simulators, and correctly encoding discrete stochastic models using off-the-shelf model-building tools

# Computer model technology

- BASIS features — service-oriented architecture (SOA)
  - Controls access to models, data and computational resources
  - Represents and encodes complex models using XML technology (SBML in this case)
  - Simulation engine that can handle a broad class of models without recompilation
  - Databases for models and simulation output
  - Web interface for human-interaction
  - SOAP web-services API for programatical access
- Shouldn't all complex computer models be made this way?!
- Do we need standards for a complex computer models API?

# BASIS Software — [www.basis.ncl.ac.uk](http://www.basis.ncl.ac.uk)

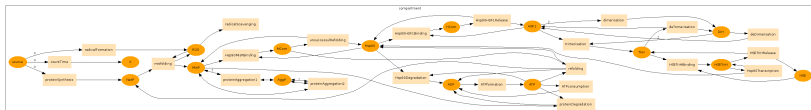
## UK e-Science GRID Pilot Project



Software architecture used to implement the BASIS system

## Example: Chaperones and their role in ageing

- C. J. Proctor, C. Soti, R. J. Boys, C. S. Gillespie, D. P. Shanley, D. J. Wilkinson, T. B. L. Kirkwood (2005) **Modelling the actions of chaperones and their role in ageing**, *Mechanisms of Ageing and Development*, **126**(1):119-131.
- Several versions of this model in the BASIS public model repository, each with a unique ID — each can be copied, modified and simulated
- eg. `urn:basis.ncl:model:518`



## Calibration of large simulators

### BBSRC Grant

**CaliBayes — Integration of GRID-based post-genomic data resources through Bayesian calibration of biological simulators** (1/05–12/07) — Wilkinson, Kirkwood, Boys, Henderson, Wolski — [www.calibayes.ncl.ac.uk](http://www.calibayes.ncl.ac.uk)

The primary aim of this project is to capitalise on the development of GRID-based modelling and simulation resources such as BASIS by building a higher level computational GRID facility designed for integration of multiple post-genomic data resources

# Outline

- Large, complex biological simulation models typically contain many parameters whose values are uncertain
- There are increasing amounts of post-genomic data being made available on-line, that can in principle be used to calibrate simulation models
- Full likelihood-based direct statistical inference techniques are likely to be impractical in this context — and in any case, it is interesting to consider the problem of calibrating a model given only the ability to **forward-simulate** from it

# The problem

- Traditional optimisation techniques (eg. steepest descent) are very wasteful of information as they use only the last one or two function evaluations to form the next “guess” at the optimum parameter set. They are also “myopic” in terms of their search strategy
- Some modern optimisation techniques (eg. simulated annealing, genetic algorithms, etc.) are impractical if function evaluations are expensive (eg. the simulator takes a long time to run), as they require vast numbers of runs
- Finding an “optimum” parameter combination is only really one small part of the problem anyway...

# MCMC-based fully Bayesian inference for *fast* computer models

- Before worrying about the issues associated with **slow** simulators, it is worth thinking about the issues involved in calibrating **fast deterministic** and **stochastic** simulators, based only on the ability to **forward-simulate** from the model
- In this case it is often possible to construct MCMC algorithms for fully Bayesian inference using the ideas of **likelihood-free MCMC** (Marjoram et al 2003)

# Parameter inference using forward simulation

- Reconsider the problem of constructing an MCMC scheme for  $\pi(c|\mathcal{D})$ .
- Simple MCMC scheme:
  - Propose  $c^* \sim f(c^*|c)$
  - Accept with probability  $\min\{1, A\}$ , where

$$A = \frac{\pi(c^*)}{\pi(c)} \times \frac{f(c|c^*)}{f(c^*|c)} \times \frac{\pi(\mathcal{D}|c^*)}{\pi(\mathcal{D}|c)}$$

- $\pi(\mathcal{D}|c)$  is the “marginal likelihood”
- Assume that  $c \perp\!\!\!\perp \mathcal{D}|\mathbf{x}$ , where  $\mathbf{x}$  is the simulator output

# Calibrating deterministic simulators

- Simulator output  $\mathbf{x} = g(c)$
- Then  $\pi(\mathcal{D}|c) = \pi(\mathcal{D}|c, g(c)) = \pi(\mathcal{D}|c, \mathbf{x}) = \pi(\mathcal{D}|\mathbf{x})$
- Use a simple measurement error model (maybe  $t$ -errors)

$$\pi(\mathcal{D}|\mathbf{x}) = \prod f(d_i|x_i)$$

- If the simulator is **slow**, could use an **emulator**,  $\hat{g}(\cdot)$  (and include extra error component)

# Calibrating stochastic simulators

- Can't get at the marginal likelihood directly, so make the target  $\pi(c, \mathbf{x}|\mathcal{D})$ , where  $\mathbf{x}$  is the “true” simulator output which led to the observed data...
- Clear that we can marginalise out  $\mathbf{x}$  if necessary, but typically of inferential interest anyway
- Propose  $(c^*, \mathbf{x}^*) \sim f(c^*|c)\pi(\mathbf{x}^*|c^*)$ , so that  $\mathbf{x}^*$  is a forward simulation from the (stochastic) model based on the proposed new  $c^*$

$$A = \frac{\pi(c^*)}{\pi(c)} \times \frac{f(c|c^*)}{f(c^*|c)} \times \frac{\pi(\mathcal{D}|c^*, \mathbf{x}^*)}{\pi(\mathcal{D}|c, \mathbf{x})}$$

- Again  $\pi(\mathcal{D}|c, \mathbf{x}) = \pi(\mathcal{D}|\mathbf{x})$  is a simple measurement error model...

## A sequential approach

- Crucially, because the proposal exploits a forward simulation, the acceptance probability does not depend on the likelihood of the simulator output — vital for complex stochastic models
- If sampling from  $\pi(\mathbf{x}|c)$  is **slow**, can use a fast **stochastic emulator**,  $\hat{\pi}(\mathbf{x}|c)$ ...
- **Problem:** If  $|\mathcal{D}|$  is large, the MCMC scheme will mix very poorly (very low acceptance rates)
- **Solution:** Adopt a sequential approach (as for the diffusion model)
- N.B. In the case of a diffusion model, have the same algorithm as before, but using an (inefficient) forward-sample rather than an (efficient) diffusion bridge proposal

## Building emulators for slow simulators





- Use Gaussian process regression to build an emulator of a slow deterministic simulator
- Obtain runs on a carefully constructed set of design points (eg. a Latin hypercube) — easy to exploit parallel computing hardware here
- For a stochastic simulator, many approaches are possible
  - (Mixtures of) Dirichlet processes (and related constructs) are potentially quite flexible
  - Can also model output parametrically (say, Gaussian), with parameters modelled by (independent) Gaussian processes

# Why are Systems Biology models interesting examples of computer models?

- Models
  - Diverse class of models: **fast/slow**, **spatial/non-spatial**, **deterministic/stochastic**, **discrete/continuous time/states** — even modelling the same biological process!
  - Many parameters
  - Structural uncertainty
  - Genuine interest in the (posterior distribution of the) parameters — not just in prediction
- Data
  - High-dimensional
  - Diverse: high-resolution time-course data, coarse population averaged data, endpoint data, **distributional data**, individual specific parameters/data, covariates
  - Multiple distinct sources of data for a given model

## Interesting methodological problems

- Calibration of fast and slow **stochastic** simulators, using individual, averaged and distributional data
- Dealing with **heterogeneity** — cell–cell, tissue–tissue, or organism–organism
- **Emulation** of slow stochastic simulators — good models and fitting procedures
- Experimental **design** for stochastic computer models — trade offs between repetition and space-filling, etc.
- Utilising fast stochastic or deterministic **approximate** simulators for a slow stochastic simulator

-  Gillespie, C. S., Wilkinson, D. J., Shanley, D. P., Proctor, C. J., Boys, R. J., Kirkwood, T. B. L. (2006). BASIS: An internet resource for network modelling, *Journal of Integrative Bioinformatics* 3(2):26.
-  Golightly, A. and D. J. Wilkinson (2005). Bayesian inference for stochastic kinetic models using a diffusion approximation. *Biometrics* 61(3), 781–788.
-  Golightly, A. and D. J. Wilkinson (2006). Bayesian sequential inference for stochastic kinetic biochemical network models. *Journal of Computational Biology*. 13(3), 838–851.
-  Wilkinson, D. J. (2006). *Stochastic Modelling for Systems Biology*. Chapman & Hall/CRC Press.

## Contact details...

email: [d.j.wilkinson@ncl.ac.uk](mailto:d.j.wilkinson@ncl.ac.uk)

www: <http://www.staff.ncl.ac.uk/d.j.wilkinson/>