Western Swiss Doctoral School in
Statistics and Probability

Statistical computing for systems biology

CUSO, Villars-sur-Ollon, Switzerland
September 2019

Darren Wilkinson
Newcastle University, UK
and
The Alan Turing Institute, UK

`darrenjw.github.io`

Lecture 1: Markov processes and stochastic kinetic models

Introduction

Markov processes form a rich class of stochastic models suitable for modelling in a range of application areas. They are particularly appropriate for modelling stochastic processes that arise naturally in the physical and life sciences, and biological modelling in particular. We therefore need a basic grounding in the theory and practice of modelling with Markov processes before we can proceed further.

A stochastic process is a random variable which evolves through time. The state may be continuous or discrete, and it can evolve through time in a discrete or continuous way. A Markov process is a stochastic process which possesses the property that the future behaviour depends only on the current state of the system. Put another way, given information about the current state of the system, information about the past behaviour of the system is of no help in predicting the time-evolution of the process.

Finite discrete time Markov chains

Introduction

The set $\{\theta^{(t)} | t = 0, 1, 2, \ldots\}$ is a **discrete time stochastic process**. The **state-space** S is such that $\theta^{(t)} \in S, \ \forall t$ and may be discrete or continuous.

A (first-order) **Markov chain** is a stochastic process with the property that the future states are independent of the past states given the present state. Formally, for $A \subseteq S$, $t = 0, 1, 2, \ldots$, we have

$$\begin{aligned} \mathsf{P}\Big(\theta^{(t+1)} \in A | \theta^{(t)} = x, \theta^{(t-1)} = x_{t-1}, \ldots, \theta^{(0)} = x_0\Big) \\ = \mathsf{P}\Big(\theta^{(t+1)} \in A | \theta^{(t)} = x\Big), \qquad \forall x, x_{t-1}, \ldots, x_0 \in S. \end{aligned}$$

The past states provide no information about the future state if the present state is known. The behaviour of the chain is therefore determined by $\mathsf{P}(\theta^{(t+1)} \in A | \theta^{(t)} = x)$. In general, this depends on $t, \ A$ and x. However, if there is no t dependence, so that

$$\mathsf{P}\Big(\theta^{(t+1)} \in A | \theta^{(t)} = x\Big) = \mathsf{P}(x, A), \quad \forall t,$$

then the Markov chain is said to be (time) **homogeneous**, and the **transition kernel**, $\mathsf{P}(x, A)$ determines the behaviour of the chain. Note that $\forall x \in S, \ \mathsf{P}(x, \cdot)$ is a probability measure over S.

Notation

When dealing with discrete state-spaces, it is easier to write

$$\mathsf{P}(x, \{y\}) = \mathsf{P}(x, y) = \mathsf{P}\left(\theta^{(t+1)} = y | \theta^{(t)} = x\right).$$

In the case of a finite discrete state-space, $S = \{x_1, \ldots, x_r\}$, we can write $\mathsf{P}(\cdot, \cdot)$ as a matrix

$$P = \begin{pmatrix} \mathsf{P}(x_1, x_1) & \cdots & \mathsf{P}(x_1, x_r) \\ \vdots & \ddots & \vdots \\ \mathsf{P}(x_r, x_1) & \cdots & \mathsf{P}(x_r, x_r) \end{pmatrix}.$$

The matrix P is a **stochastic matrix**.

Stationary distributions

A distribution π is said to be a **stationary distribution** of the homogeneous Markov chain governed by the transition matrix P if

$$\pi = \pi\, P. \tag{1}$$

Note that π is a row eigenvector of the transition matrix, with corresponding eigenvalue equal to 1. It is also a fixed point of the linear map induced by P. The stationary distribution is so-called because if at some time n, we have $\pi^{(n)} = \pi$, then $\pi^{(n+1)} = \pi^{(n)} P = \pi\, P = \pi$, and similarly $\pi^{(n+k)} = \pi,\ \forall k \geq 0$. That is, if a chain has a stationary distribution, it retains that distribution for all future time. Note that

$$\begin{aligned} \pi = \pi\, P &\iff \pi - \pi\, P = 0 \\ &\iff \pi(I - P) = 0, \end{aligned}$$

where I is the $r \times r$ identity matrix. Hence the stationary distribution of the chain may be found by solving

$$\pi(I - P) = 0. \tag{2}$$

Note that the trivial solution $\pi = 0$ is not of interest here, as it does not correspond to a probability distribution (its elements do not sum to 1). However, there are always infinitely many solutions to (2), so proper solutions can be found by finding a positive solution and then imposing the unit-sum constraint. In the case of a unique stationary distribution (just one eigenvalue of P equal to 1), then there will be a one-dimensional set of solutions to (2), and the unique stationary distribution will be the single solution with positive elements summing to 1.

In the case of a unique stationary distribution it is natural to wonder whether $\pi^{(n)}$ will converge to this stationary distribution as $n \longrightarrow \infty$ irrespective of $\pi^{(0)}$, in which case it is known as the (unique) **equilibrium distribution** of the chain. In the case of a finite state space, this question is relatively straightforward to resolve. However, in the general state space case that we will be most interested in ultimately, this turns out to be a very delicate question to answer. We will assume that in most cases of practical interest that we do have convergence, but it is important to be aware that this is **not** true in general.

Markov chains with continuous state-space

Many Markovian models naturally posess a discrete state space, but sometimes it is helpful to regard the state of certain quantities as continuous. In this case, we have to understand how the concept of a discrete state Markov chain extends to the continuous state case. In fact, this extension is exactly analogous to the generalisation of discrete random quantities to that of continuous random quantities.

Here we are still working with discrete time, but we are allowing the state-space S of the Markov chain to be continuous (e.g. $S \subseteq \mathbb{R}$).

Transition kernels

Again, for a homogeneous chain, we can define

$$\mathsf{P}(x, A) = \mathsf{P}\Big(\theta^{(t+1)} \in A|\theta^{(t)} = x\Big) .$$

For continuous state-spaces we always have $\mathsf{P}(x, \{y\}) = 0$, so in this case we define $\mathsf{P}(x, y)$ by

$$\begin{aligned} \mathsf{P}(x, y) &= \mathsf{P}\Big(\theta^{(t+1)} \leq y|\theta^{(t)} = x\Big) \\ &= \mathsf{P}\Big(\theta^{(1)} \leq y|\theta^{(0)} = x\Big) , \ \forall x, y \in S, \end{aligned}$$

the conditional cumulative distribution function (CDF). This is the distributional form of the transition kernel for continuous state space Markov chains, but we can also define the corresponding conditional density

$$p(x, y) = \frac{\partial}{\partial y} \mathsf{P}(x, y), \quad x, y, \in S.$$

We can use this to define the density form of the **transition kernel** of the chain. Note that $p(x, y)$ is just the conditional density for the next state (with variable y) given that the current state is x, so it could also be written $p(y|x)$. The density form of the kernel can be used more conveniently than the CDF form for vector Markov chains, where the state-space is multidimensional (say $S \subseteq \mathbb{R}^n$).

Stochastic simulation and analysis

Simulation of Markov chains with a continuous state in discrete time is easy provided that methods are available for simulating from the initial distribution, $\pi^{(0)}(x)$, and from the conditional distribution represented by the transition kernel, $p(x, y)$.

1. First $\theta^{(0)}$ is sampled from $\pi^{(0)}(\cdot)$
2. We can then simulate $\theta^{(1)}$ from $p(\theta^{(0)}, \cdot)$
3. In general, once we have simulated a realisation of $\theta^{(t)}$, we can simulate $\theta^{(t+1)}$ from $p(\theta^{(t)}, \cdot)$

Markov chains in continuous time

We have now looked at Markov chains in discrete time with both discrete and continuous state-spaces. However, many processes evolve continuously in time, and so we now turn our attention to the continuous time case. We begin by studying chains with a finite number of states, but relax this assumption in due course.

Before we begin we should try to be explicit about what exactly a Markov process is in the continuous time case. Intuitively it is a straightforward extension of the discrete time definition. In continuous time, we can write this as

$$\begin{aligned}\mathrm{P}(X(t+dt)=x|\{X(t)=x(t)|t\in[0,t]\})\\ =\mathrm{P}(X(t+dt)=x|X(t)=x(t))\,, \quad \forall t\in[0,\infty),\ x\in S.\end{aligned}$$

Again, this expresses the idea that the future behaviour of the process depends on the past behaviour of the process only via the current state.

Finite state-space

Consider first a process which can take on one of r states, which we label $S = \{1, 2, \dots, r\}$. If at time t the process is in state $x \in S$, its future behaviour can be characterised by the transition kernel

$$p(x, t, x', t') \equiv \mathsf{P}\Big(X(t + t') = x' | X(t) = x\Big)\,.$$

If this function does not depend explicitly on t, the process is said to be **homogeneous**, and the kernel can be written $p(x, x', t')$. For each value of t', this kernel can be expressed as an $r \times r$ transition matrix, $P(t')$. It is clear that $P(0) = I$, the $r \times r$ identity matrix, as no transitions will take place in a time interval of length zero. Also note that since $P(\cdot)$ is a transition matrix for each value of t, we can multiply these matrices together to give combined transition matrices in the usual way. In particular, we have $P(t + t') = P(t)P(t') = P(t')P(t)$, just as in the discrete time case.

Now define the **transition rate matrix**, Q, to be the derivative of $P(t')$ at $t' = 0$. Then

$$\begin{aligned} Q &= \left.\frac{d}{dt'}P(t')\right|_{t'=0} \\ &= \lim_{\delta t \to 0} \frac{P(\delta t) - P(0)}{\delta t} \\ &= \lim_{\delta t \to 0} \frac{P(\delta t) - I}{\delta t}. \end{aligned}$$

The elements of the Q matrix give the “hazards” of moving to different states. Re-arranging gives the infinitesimal transition matrix

$$P(dt) = I + Q\,dt.$$

Note that for $P(dt)$ to be a stochastic matrix (with non-negative elements and rows summing to 1), the above implies several constraints which must be satisfied by the rate matrix Q. Since the off-diagonal elements of I are zero, the off-diagonal elements of $P(dt)$ and $Q\,dt$ must be the same, and so the off-diagonal elements of Q must be non-negative. Also, since the diagonal elements of $P(dt)$ are bounded above by one, the diagonal elements of Q must be non-positive. Finally, since the rows of $P(dt)$ and I both sum to 1, the rows of Q must each sum to zero. These properties must be satisfied by **any** rate matrix Q.

The above rearrangement gives us a way of computing the stationary distribution of the Markov chain, as a probability row vector π will be stationary only if

$$\begin{aligned} \pi\, P(dt) &= \pi \\ \Rightarrow \pi(I + Qdt) &= \pi \\ \Rightarrow \pi\, Q &= 0. \end{aligned}$$

Solving this last equation (subject to the constraint that the elements of π sum to 1), will give a stationary distribution for the system.

If $P(t)$ is required for finite t, it may be computed by solving a matrix differential equation. This can be derived by considering the derivative of $P(t)$ for arbitrary times t.

$$\begin{aligned}\frac{d}{dt}P(t) &= \frac{P(t+dt)-P(t)}{dt}\\ &= \frac{P(dt)P(t)-P(t)}{dt}\\ &= \frac{P(dt)-I}{dt}P(t)\\ &= Q\,P(t).\end{aligned}$$

Therefore, $P(t)$ is a solution to the matrix differential equation

$$\frac{d}{dt}P(t) = QP(t),$$

subject to the initial condition $P(0) = I$. This differential equation has solution

$$P(t) = \exp\{Q\,t\},$$

where $\exp\{\cdot\}$ denotes the matrix exponential function (Golub & Van Loan 1996).

Example

Consider a very simple model for the activation of a single prokaryotic gene. In this model, the gene will be activated unless a repressor protein is bound to its regulatory region. We will consider just two states in our system: state 0 (inactive), and state 1 (active). In the inactive state (0), we will assume a constant hazard of $\alpha > 0$ for activation. In the active state, we will assume a constant hazard of $\beta > 0$ for inactivation. Given that the rows of Q must sum to zero, it is now completely specified as

$$Q = \begin{pmatrix} -\alpha & \alpha \\ \beta & -\beta \end{pmatrix}.$$

Solving $\pi Q = 0$ gives the stationary distribution

$$\pi = \left(\frac{\beta}{\alpha+\beta} \quad , \quad \frac{\alpha}{\alpha+\beta} \right).$$

We can also compute the infinitesimal transition matrix

$$P(dt) = I + Q\,dt = \begin{pmatrix} 1-\alpha\,dt & \alpha\,dt \\ \beta\,dt & 1-\beta\,dt \end{pmatrix}.$$

A simulated realisation of this process is shown in Figure 1. Note that this process is sometimes known as the **telegraph process** and can also form the basis for a very simple model of ion channels.

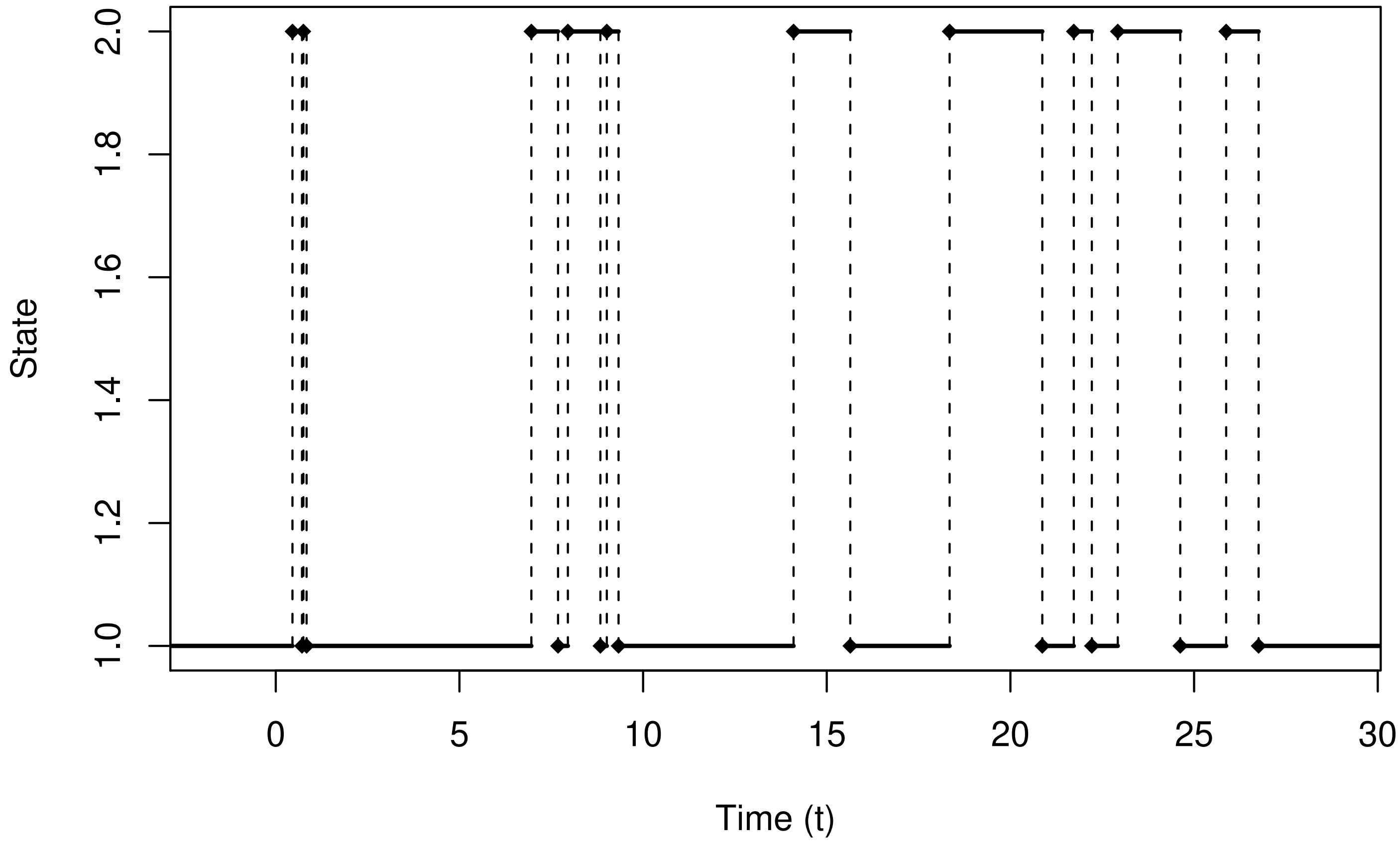

Figure 1: A simulated realisation of the simple gene activation process with $\alpha = 0.5$ and $\beta = 1$.

Stochastic simulation

There are three straightforward approaches one can take to simulating this process on a computer. The first is based on a fine time-discretisation of the process, similar in spirit to the first-order Euler method for integrating ordinary differential equations. Given the definition of the infinitesimal transition matrix

$$P(dt) = I + Q\,dt,$$

for small time steps Δt we will have

$$P(\Delta t) \simeq I + Q\,\Delta t.$$

$P(\Delta t)$ can then be regarded as the transition matrix of a discrete time Markov chain, and a simulated sequence of states at times $0,\ \Delta t,\ 2\,\Delta t,\ 3\,\Delta t, \ldots$ may be generated in the usual way.

The above method can be easily improved by replacing the above approximation for $P(\Delta t)$ by its **exact** value

$$P(\Delta t) = \text{exp}\{Q\,\Delta t\},$$

provided that a method for computing the matrix exponential is available. Then it does not matter how small Δt is chosen to be, provided it is small enough to clearly show the behaviour of the process and not so large that interesting transitions are “missed”.

A third approach to simulation may be taken by simulating each transition event and its corresponding time sequentially, rather than simply looking at the processes only at times on a given lattice. Like the previous method, this gives an exact realisation of the process and offers the additional advantage that recording every event ensures none will be “missed”. Such an approach is known as **discrete event simulation**, and in the statistics literature the technique dates back to at least Kendall (1950). If the process is currently in state x, then the xth row of Q gives the hazards for the transition to other states. As the row sums to zero, $-q_{xx}$ gives the combined hazard for moving away from the current state — a discrete transition event (note that q_{xx} is non-positive). So, the time to a transition event is exponential with rate $-q_{xx}$. When that transition event occurs, the new state will be random with probabilities proportional to the xth row of Q (with q_{xx} omitted). The above intuitive explanation can be formalised as follows.

To understand how to simulate the process we must consider being in state i at time t, and think about the probability that the next event will be in the time interval $(t + t', t + t' + dt]$, **and** will consist of a move to state j. Let this probability divided by dt be denoted by $f(t', j|t, i)$, so that the probability is $f(t', j|t, i)dt$. It is clear that as the Markov process is homogeneous, there will be no explicit dependence on t in this probability, but we will include it to be clear about exactly what we mean. Then

$$f(t', j|t, i)dt = \mathrm{P}\Big(\text{Next event in } (t + t', t + t' + dt]|t, i\Big) \\ \times \mathrm{P}\Big(j|\text{Next event in } (t + t', t + t' + dt], t, i\Big).$$

Thinking about the first term, we know that the hazards for the individual transitions are given by the off-diagonal elements of the ith row of Q. The combined hazard is the sum of these off-diagonal elements, which is $-q_{ii}$ (as the row sums to zero). Combined hazards can always be computed as the sum of hazards in this way because the probability that two events occur in the interval $(t, t + dt]$ is of order dt^2 and can therefore be neglected. Now we know from properties of the exponential distribution that the time to an event of constant hazard is $Exp(-q_{ii})$, and so the first term must be $-q_{ii}e^{q_{ii}t'}\,dt$.

The second term is

$$\begin{aligned}\mathsf{P}\Big(X(t+t'+dt)=j|[X(t+t')=i]\cap[X(t+t'+dt)\neq i]\Big)\\ = \frac{\mathsf{P}(X(t+t'+dt)=j|X(t+t')=i)}{\mathsf{P}(X(t+t'+dt)\neq i|X(t+t')=i)} = \frac{q_{ij}dt}{\sum_{k\neq i} q_{ik}dt} = \frac{q_{ij}}{-q_{ii}}.\end{aligned}$$

Taking the two terms together we have

$$f(t',j|t,i) = -q_{ii}e^{q_{ii}t'} \times \frac{q_{ij}}{-q_{ii}}.$$

The fact that this function factorises into the form of a probability density for the time to the next event and a probability mass function for the type of that event means that we can simulate the next event with the generation of two random variables. Note also that there is no j dependence in the PDF for t' and no t' dependence in the PMF for j, so the two random variables are independent of one another and hence can be simulated independently.

It is the consideration of $f(t', j|t, i)$ that leads to the standard discrete event simulation algorithm which could be stated as follows:

1. Initialise the process at $t = 0$ with initial state i;

2. Call the current state i. Simulate the time to the next event, t', as an $Exp(-q_{ii})$ random quantity;

3. Put $t := t + t'$;

4. Simulate new state j as a discrete random quantity with PMF $-q_{ik}/q_{ii},\ k \neq i$;

5. Output the time t and state j;

6. If $t < T_{max}$, return to step 2.

This particular discrete event simulation technique is known as the **direct method**. A simple R function to implement this algorithm is given in Figure 2. The function returns a step-function object, which is easy to plot. Using this function, a plot similar to Figure 1 can be obtained with the following command:

```
plot(rcfmc(20,matrix(c(-0.5,0.5,1,-1),ncol=2,byrow=TRUE),c(1,0)))
```

```
rcfmc <- function(n,Q,pi0)
{
        xvec = vector("numeric",n+1)
        tvec = vector("numeric",n)
        r = length(pi0)
        x = sample(r,1,prob=pi0)
        t = 0
        xvec[1] = x
        for (i in 1:n) {
                t = t+rexp(1,-Q[x,x])
                weights = Q[x,]
                weights[x] = 0
                x = sample(r,1,prob=weights)
                xvec[i+1] = x
                tvec[i] = t
        }
        stepfun(tvec,xvec)
}
```

Figure 2: An R function to simulate a sample path with `n` events from a continuous time Markov chain with transition rate matrix `Q` and initial distribution `pi0`.

All of these simulation methods give a single realisation of the Markov process. Now obviously, just as one would not study a normal distribution by looking at a single simulated value, the same is true with Markov processes. Many realisations must be simulated in order to get a feel for the **distribution** of values at different times. In the case of a finite number of states, this distribution is relatively straightforward to compute directly without any simulation at all, but for the stochastic kinetic models we will consider later, simulation is likely to be the only tool we have available to us for gaining insight into the behaviour of the process.

Countable state-space

Before moving on to thinking about continuous state-spaces, it is worth spending a little time looking at the case of a countably infinite state-space. Rather than attempting to present the theory in generality, we will concentrate on a simple example, which illustrates many of the interesting features. The model is known as the **immigration-death process**. In this model, individuals arrive into the population with constant hazard λ, and each individual dies independently with constant hazard μ. Consequently, the population of individuals increases by one when an immigration event occurs and decreases by one when a death event occurs. There is no reproduction in this model. The key transition equations are:

$$\begin{aligned} \mathrm{P}(X(t+dt)=x+1|X(t)=x) &= \lambda\, dt \\ \mathrm{P}(X(t+dt)=x-1|X(t)=x) &= x\mu\, dt \\ \mathrm{P}(X(t+dt)=x|X(t)=x) &= 1-(\lambda+x\mu)dt \\ \mathrm{P}(X(t+dt)=y|X(t)=x) &= 0,\ \forall y\notin\{x-1,x,x+1\}. \end{aligned}$$

These equations clearly define a homogeneous Markov process, but with infinite state-space $S=0,1,2,\ldots$. We therefore cannot easily write down a set of matrix equations for the process, as the matrices are infinite dimensional, but this does not prevent us from working with the process or from simulating it on a computer.

First let's think about understanding this process theoretically. Although the Q matrix is infinite in extent, we can write its general form as follows:

$$Q = \begin{pmatrix} -\lambda & \lambda & 0 & 0 & 0 & \cdots \\ \mu & -\lambda-\mu & \lambda & 0 & 0 & \cdots \\ 0 & 2\mu & -\lambda-2\mu & \lambda & 0 & \\ 0 & 0 & 3\mu & -\lambda-3\mu & \lambda & \ddots \\ \vdots & & \ddots & \ddots & \ddots & \ddots \end{pmatrix}.$$

Then for an infinite dimensional $\pi = (\pi_0, \pi_1, \pi_2, \ldots)$ we can solve $\pi Q = 0$ to get the stationary distribution one equation at a time, expressing each π_k in terms of π_0 to find the general form

$$\pi_k = \frac{\lambda^k}{k!\mu^k}\pi_0, \quad k = 1, 2, \ldots.$$

But these are terms in the expansion of $\pi_0 e^{\lambda/\mu}$, and so imposing the unit-sum constraint we get $\pi_0 = e^{-\lambda/\mu}$ giving the general solution

$$\pi_k = \frac{(\lambda/\mu)^k e^{-\lambda/\mu}}{k!}, \quad k = 0, 1, 2, \ldots.$$

This is easily recognised as the PMF of a Poisson random quantity with mean λ/μ. Hence, the stationary distribution of this process is Poisson with mean λ/μ.

We can also simulate realisations of this process on a computer. Here it is easiest to use the technique of discrete event simulation. If the current state of the process is x, the combined hazard for moving away from the current state is $\lambda + x\mu$, and so the time to the next event is an exponentially distributed random quantity with rate $\lambda + x\mu$. When that event occurs, the process will move up or down with probabilities proportional to their respective hazards, λ and $x\mu$. That is, the state will increase by 1 with probability $\lambda/(\lambda + x\mu)$ and decrease by 1 with probability $x\mu/(\lambda + x\mu)$. This sequence can be easily simulated on a computer to give a set of states and event times which can be plotted, summarised, etc. A simulated realisation of this immigration-death process is shown in Figure 3. An R function to simulate the process is given in Figure 4.

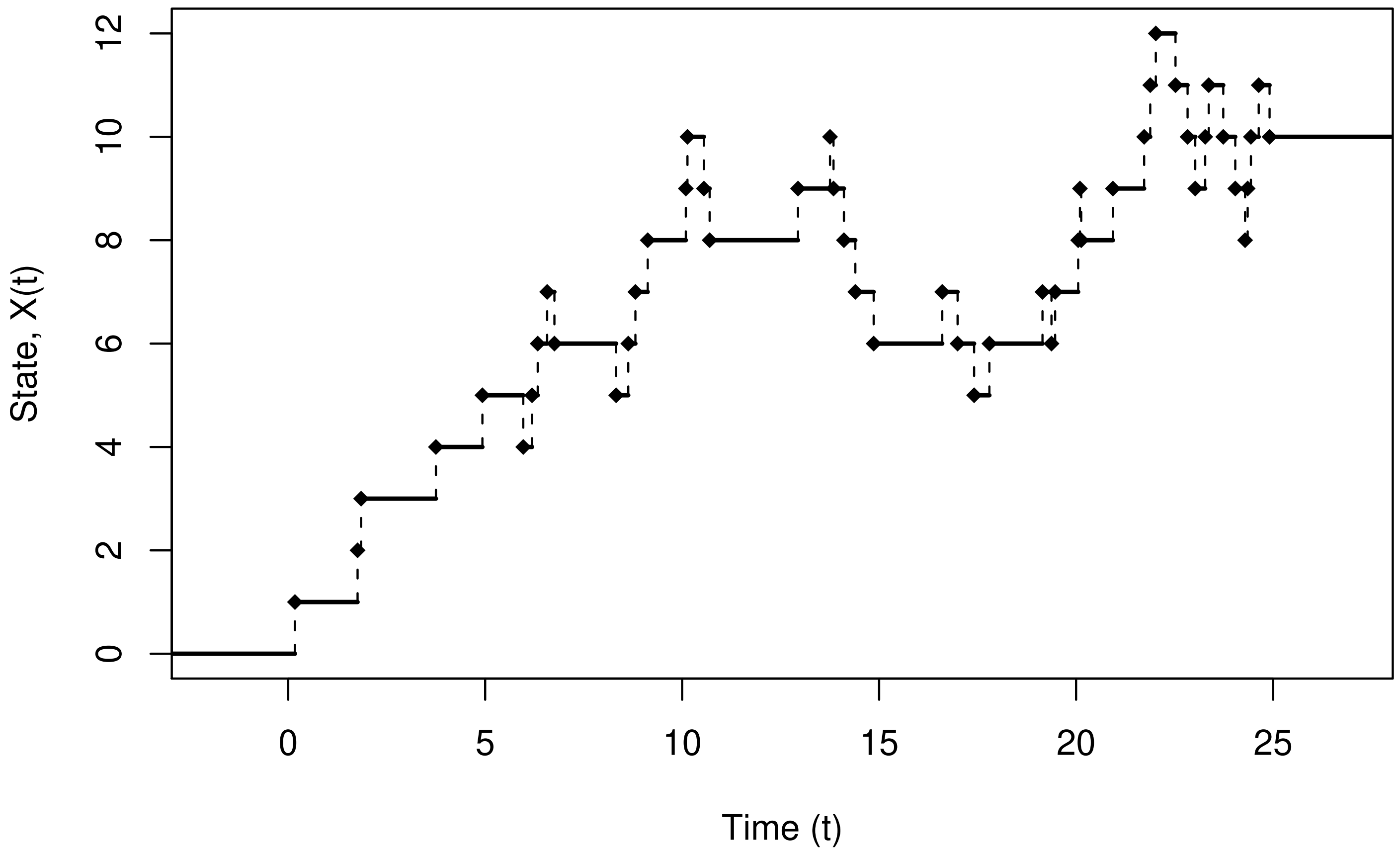

Figure 3: A single realisation of the immigration-death process with parameters $\lambda = 1$ and $\mu = 0.1$, initialised at $X(0) = 0$. Note that the stationary distribution of this process is Poisson with mean 10.

```
imdeath <- function(n=20, x0=0, lambda=1, mu=0.1)
{
        xvec = vector("numeric",n+1)
        tvec = vector("numeric",n)
        t = 0
        x = x0
        xvec[1] <- x
        for (i in 1:n) {
                t = t+rexp(1,lambda+x*mu)
                if ( runif(1,0,1) < lambda/(lambda+x*mu) )
                        x <- x+1
                else
                        x <- x-1
                xvec[i+1] <- x
                tvec[i] <- t
        }
        stepfun(tvec, xvec)
}
```

Figure 4: R function for discrete-event simulation of the immigration-death process.

Diffusion processes

Introduction

Markov processes with continuous states that evolve continuously in time are often termed **diffusion processes**. A rigorous formal discussion of the theory of such processes is beyond the scope of this course. Nevertheless, it is useful to provide a non-technical introduction at this point, as these processes provide an excellent modelling tool for many applications.

A d-dimensional Itô diffusion process Y is governed by a stochastic differential equation (SDE) of the form

$$dY_t = \mu(Y_t)dt + \Psi(Y_t)dW_t, \tag{3}$$

where $\mu : \mathbb{R}^d \to \mathbb{R}^d$ is a d-dimensional drift vector and $\Psi : \mathbb{R}^d \to \mathbb{R}^d \times \mathbb{R}^d$ is a $(d \times d)$-dimensional diffusion matrix. The SDE can be thought of as a recipe for constructing a realisation of Y from a realisation of a d-dimensional Brownian motion (or Wiener process), W. A d-dimensional Brownian motion has d independent components, each of which is a univariate Brownian motion, B. A univariate Brownian motion B is a process defined for $t \geq 0$ in the following way.

1. $B_0 = 0$,

2. $B_t - B_s \sim N(0, t - s), \ \forall t > s$,

3. The increment $B_t - B_s$ is independent of the increment $B_{t'} - B_{s'}, \ \forall t > s \geq t' > s'$.

It is clear from property 2 that $B_t \sim N(0, t)$ (and so $\mathrm{E}(B_t) = 0$ and $\mathrm{Var}(B_t) = t$). It is also clear that if for some small time increment Δt we define the process increment $\Delta B_t = B_{t+\Delta t} - B_t$, we then have $\Delta B_t \sim N(0, \Delta t), \ \forall t$, and since we know that the increments are independent of one another, this provides a mechanism for simulating the process on a regular grid of time points.

If we define the increment in the diffusion process Y (and the multivariate Brownian motion W) similarly, then we can interpret the SDE (3) as the limit of the difference equation

$$\Delta Y_t = \mu(Y_t)\Delta t + \Psi(Y_t)\Delta W_t, \tag{4}$$

as Δt gets infinitesimally small. For finite Δt, (4) is known as the **Euler approximation** (or, more correctly, as the **Euler–Maruyama approximation**) of the SDE, and it provides a simple mechanism for approximate simulation of the process Y on a regular grid of time points.

In the case $d = 1$ we have a univariate diffusion process, and it is clear that then the increments of the process are approximately distributed as

$$\Delta Y_t \sim N(\mu(Y_t)\Delta t, \Psi(Y_t)^2\Delta t).$$

An R function to simulate a univariate diffusion using an Euler approximation is given in Figure 5. Note that more efficient simulation strategies are possible; see Kloeden & Platen (1992) for further details.

```
rdiff <- function(afun, bfun, x0 = 0, t = 50, dt = 0.01, ...)
{
            n <- t/dt
            xvec <- vector("numeric", n)
            x <- x0
            sdt <- sqrt(dt)
            for (i in 1:n) {
                    t <- i*dt
                    x <- x + afun(x,...)*dt +
                                   bfun(x,...)*rnorm(1,0,sdt)
                    xvec[i] <- x
            }
            ts(xvec, deltat = dt)
}
```

Figure 5: R function for simulation of a diffusion process using the Euler method.

We can approximate a discrete Markov process using a diffusion by choosing the functions $\mu(\cdot)$ and $\Psi(\cdot)$ so that the mean and variance of the increments match. This is best illustrated by example.

Example — diffusion approximation of the immigration-death process

Suppose we have an immigration-death process with immigration rate λ and death rate μ, and that at time t the current state of the system is x. Then at time $t + dt$, the state of the system is a discrete random quantity with PMF,

$$\begin{aligned} \mathsf{P}\Big(X_{t+dt} = x - 1\Big) &= x\mu\, dt, \\ \mathsf{P}\Big(X_{t+dt} = x\Big) &= 1 - (\lambda + x\mu)dt, \\ \mathsf{P}\Big(X_{t+dt} = x + 1\Big) &= \lambda\, dt. \end{aligned}$$

So the increment of the process has PMF

$$\mathsf{P}(dX_t = -1) = x\mu\, dt,\ \mathsf{P}(dX_t = 0) = 1{-}(\lambda{+}x\mu)dt,\ \mathsf{P}(dX_t = 1) = \lambda\, dt.$$

From this PMF we can calculate the expectation and variance as

$$\mathsf{E}(dX_t) = (\lambda - \mu x)dt, \quad \mathsf{Var}(dX_t) = (\lambda + \mu x)dt.$$

We therefore set $\mu(x) = \lambda - \mu x$ and $\Psi(x)^2 = \lambda + \mu x$ to get the diffusion approximation

$$dX_t = (\lambda - \mu x)dt + \sqrt{\lambda + \mu x}\, dW_t.$$

We can use our code for simulating diffusion processes to get sample paths like that shown in Figure 6 using the R code shown in Figure 7.

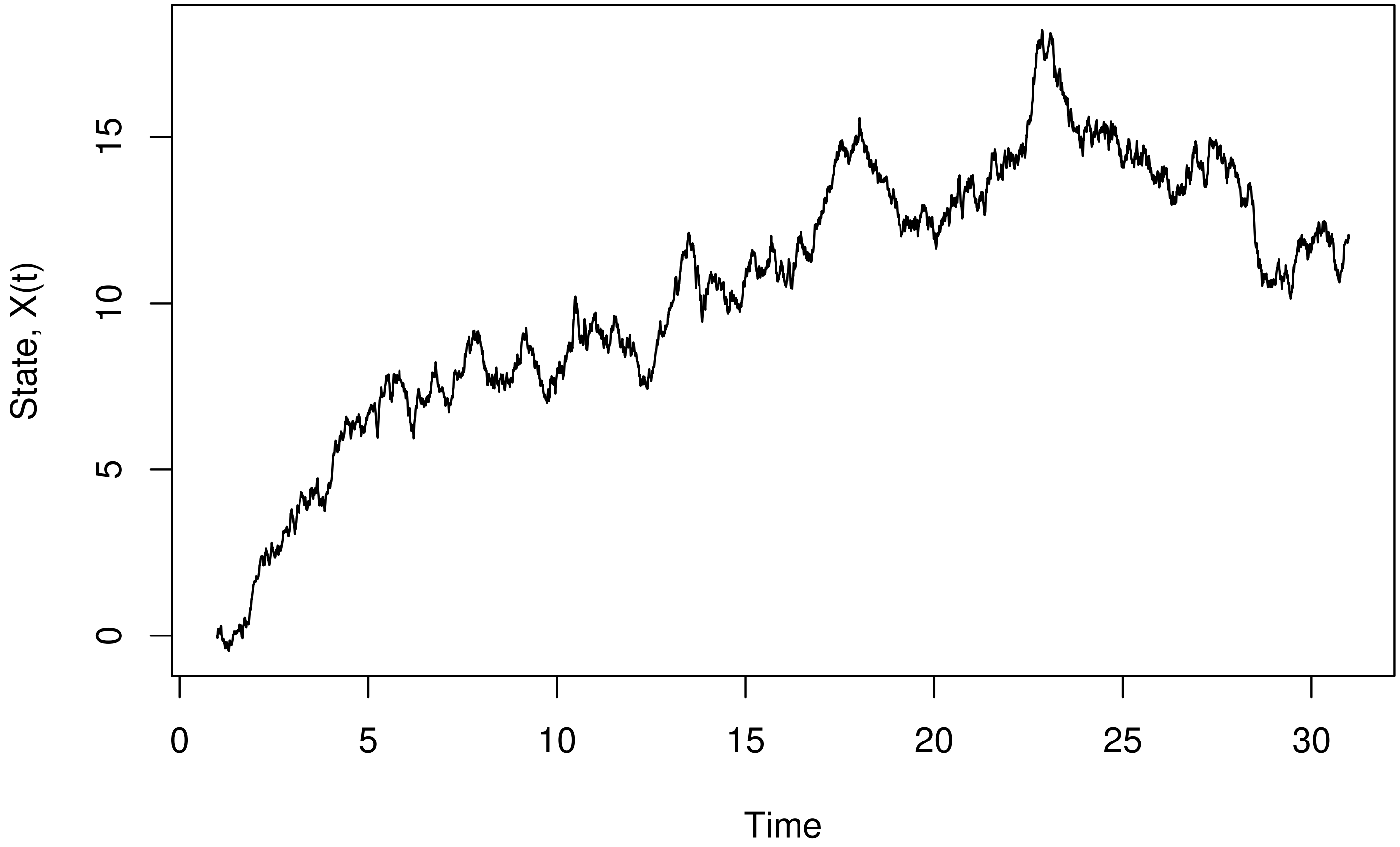

Figure 6: A single realisation of the diffusion approximation to the immigration-death process with parameters $\lambda = 1$ and $\mu = 0.1$, initialised at $X(0) = 0$. Note that this realisation appears to dip below zero near the time origin.

```r
afun <- function(x,lambda,mu)
{
        lambda-mu*x
}
bfun <- function(x,lambda,mu)
{
        sqrt(lambda+mu*x)
}
plot(rdiff(afun,bfun,lambda=1,mu=0.1,t=30))
```

Figure 7: R code for simulating the diffusion approximation to the immigration-death process.

Chemical reactions

A general chemical reaction takes the form

$$m_1 R_1 + m_2 R_2 + \cdots + m_r R_r \longrightarrow n_1 P_1 + n_2 P_2 + \cdots + n_p P_p,$$

where r is the number of reactants and p is the number of products. R_i represents the ith reactant molecule and P_j is the jth product molecule. m_i is the number of molecules of R_i consumed in a single reaction step, and n_j is the number of molecules of P_j produced in a single reaction step. The coefficients m_i and n_j are known as **stoichiometries**. The stoichiometries are usually (though not always) assumed to be integers, and in this case it is assumed that there is no common factor of the stoichiometries. That is, it is assumed that there is no integer greater than one which exactly divides each stoichiometry on both the left and right sides. There is no assumption that the R_i and P_j are distinct, and it is perfectly reasonable for a given molecule to be both consumed and produced by a single reaction. The reaction equation describes precisely which chemical species react together, and in what proportions, along with what is produced.

In order to make things more concrete, consider the dimerisation of a protein P. This is normally written

$$2P \longrightarrow P_2,$$

as two molecules of P react together to produce a single molecule of P_2. Here P has a stoichiometry of 2 and P_2 has a stoichiometry of 1. Stoichiometries of 1 are not usually written explicitly. Similarly, the reaction for the dissociation of the dimer would be written

$$P_2 \longrightarrow 2P.$$

A reaction that can happen in both directions is known as **reversible**. Reversible reactions are quite common in biology and tend not to be written as two separate reactions. They can be written with a double-headed arrow such as

$$2P \rightleftharpoons P_2 \quad \text{or} \quad 2P \longleftrightarrow P_2 \quad \text{or} \quad 2P \Longleftrightarrow P_2.$$

If one direction predominates over the other, this is sometimes emphasised in the notation. So if the above protein prefers the dimerised state, this may be written something like

$$2P \underset{\leftharpoondown}{\rightharpoonup} P_2 \quad \text{or} \quad 2P \underset{\leftarrow}{\longrightarrow} P_2.$$

Modelling genetic and biochemical networks

Transcription (prokaryotes)

Transcription is a key cellular process, and control of transcription is a fundamental regulation mechanism. As a result, virtually any model of genetic regulation is likely to require some modelling of the transcription process. This process is much simpler in prokaryotic organisms, so it will be helpful to consider this in the first instance. Here, typically, a promoter region exists just upstream of the gene of interest. RNA-polymerase (RNAP) is able to bind to this promoter region and initiate the transcription process, which ultimately results in the production of an mRNA transcript and the release of RNAP back into the cell. The transcription process itself is complex, but whether it will be necessary to model this explicitly will depend very much on the modelling goals. If the modeller is primarily interested in control and the downstream effects of the transcription process, it may not be necessary to model transcription itself in detail.

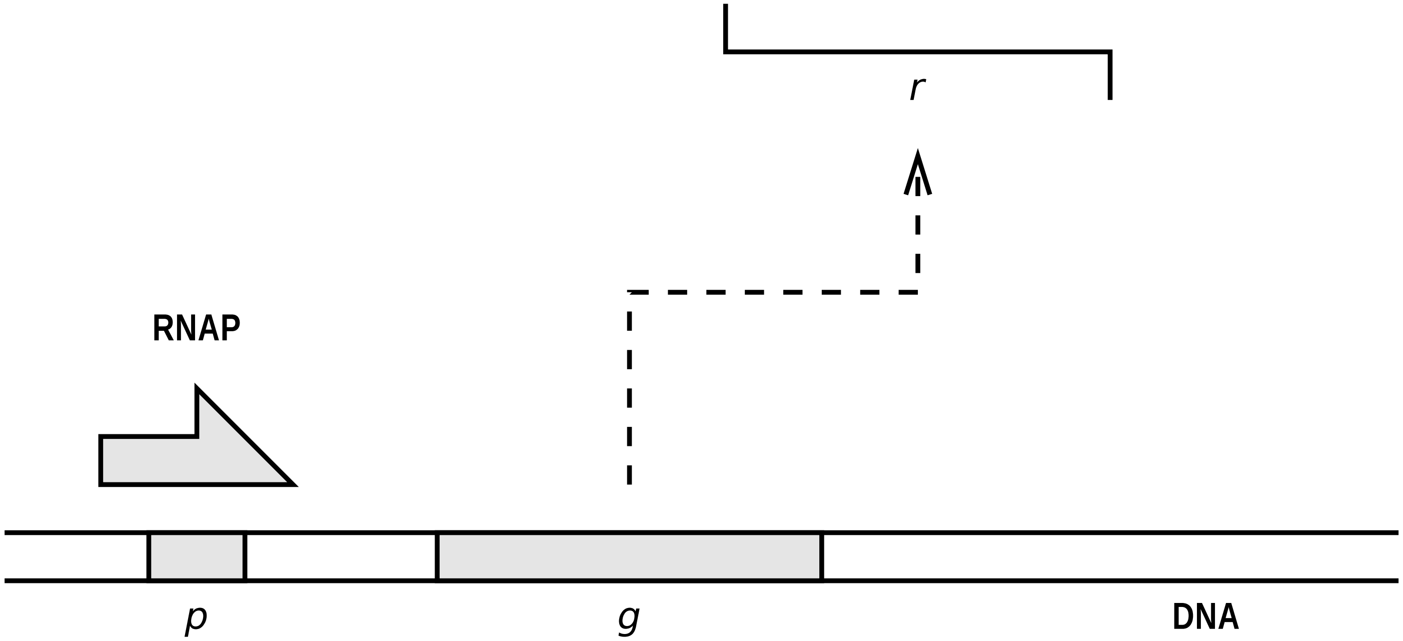

Figure 8: Transcription of a single prokaryotic gene.

The process is illustrated diagrammatically in Figure 8. Here, g is the gene of interest, p is the upstream promoter region, and r is the mRNA transcript of g. A very simple representation of this process as a system of coupled chemical reactions can be written as follows:

$$\begin{aligned} p + \text{RNAP} &\longrightarrow p \cdot \text{RNAP} \\ p \cdot \text{RNAP} &\longrightarrow p + \text{RNAP} + r. \end{aligned}$$

As discussed, the second reaction is really the end result of a very large number of reactions. It is also worth emphasising that the reactions do not represent a closed system, as r appears to be produced out of nothing. In reality, it is created from other chemical species within the cell, but we have chosen here not to model at such a fine level of detail. One detail not included here that may be worth considering is the reversible nature of the binding of RNAP to the promoter region.

It is also worth noting that these two reactions form a simple linear chain, whereby the product of the first reaction is the reactant for the second. Indeed, we could write the pair of reactions as

$$p + \text{RNAP} \longrightarrow p \cdot \text{RNAP} \longrightarrow p + \text{RNAP} + r.$$

It is therefore tempting to summarise this chain of reactions by the single reaction

$$p + \text{RNAP} \longrightarrow p + \text{RNAP} + r,$$

and this is indeed possible, but is likely to be inadequate for any model of regulation or control where the intermediary compound $p \cdot \text{RNAP}$ is important, such as any model for competitive binding of RNAP and a repressor in the promoter region.

Modelling higher-level systems

We have concentrated so far on fairly low-level biochemical models where the concept of modelling with “chemical reactions” is perhaps most natural. However, it is important to recognise that we use the notation of chemical reactions simply to describe things that combine and the things that they produce, and that this framework can be used to model higher-level phenomena in a similar way.

Consider the Lotka–Volterra predator prey model for two interacting species:

$$
\begin{aligned}
Y_1 &\longrightarrow 2Y_1 \\
Y_1 + Y_2 &\longrightarrow 2Y_2 \\
Y_2 &\longrightarrow \emptyset.
\end{aligned}
$$

This is not a real reaction system in the strictest sense, but it is interesting and useful, as it is the simplest model exhibiting the kind of non-linear auto-regulatory feedback behaviour that is very common in biological systems. Also, as it only involves two species and three reactions, it is relatively easy to work with without getting lost in detail. Here, Y_1 represents a "prey" species (such as rabbits) and Y_2 represents a "predator" species (such as foxes). The first reaction is a simple representation of prey reproduction. The second reaction is an attempt to capture predator-prey interaction (consumption of prey by predator, in turn influencing predator reproduction rate). The third reaction represents death of predators due to natural causes.

Another widely studied individual level model is the so-called SIR model for disease epidemiology, where the initials stand for **Susceptible**, **Infected**, and **Recovered**. The idea is that individuals are initially susceptible to catching a disease from an infected person. Should they contract the disease, they will make the transition from susceptible to infected, where they will have the possibility of infecting susceptibles. Eventually the infected individual will make the transition to the "recovered" category, when they will no longer be able to infect susceptibles, but will have immunity to the disease, and hence will not be themselves any longer susceptible to infection. Of course for some diseases, this "recovered" category will include individuals who are in fact dead! In this case, the "R" category is sometimes used to stand for **Removed**. The simplest variant of this model is often summarised with just two reactions as

$$S \longrightarrow I \longrightarrow R.$$

However, it should more correctly be written as the pair of reactions

$$\begin{aligned} S + I &\longrightarrow 2I \\ I &\longrightarrow R. \end{aligned}$$

There are obviously many variants on this basic model. For example, some individuals may develop immunity without ever becoming infectious ($S \longrightarrow R$) and some recovered individuals may lose their immunity ($R \longrightarrow S$), etc. Another commonly studied variant is the SEIR model which introduces an additional category, **Exposed**, representing individuals who have been infected with the disease but are not yet themselves infectious, and this is often summarised as

$$S \longrightarrow E \longrightarrow I \longrightarrow R.$$

Classical continuous deterministic chemical kinetics

Introduction

Chemical kinetic modelling is concerned with understanding the time-evolution of a reaction system specified by a given set of coupled chemical reactions. In particular, it is concerned with system behaviour away from equilibrium. In order to introduce the concepts it is helpful to use a very simple model system. Consider the “Lotka–Volterra” (LV) system introduced previously:

$$\begin{aligned} Y_1 &\longrightarrow 2Y_1 \\ Y_1 + Y_2 &\longrightarrow 2Y_2 \\ Y_2 &\longrightarrow \emptyset. \end{aligned}$$

Although the reaction equations capture the key interactions between the competing species, on their own they are not enough to determine the full dynamic behaviour of the system. For that, we need to know the **rates** at which each of the reactions occurs (together with some suitable initial conditions).

Mass-action kinetics

The above model encourages us to think about the number of prey (Y_1) and predators (Y_2) as integers, which can change only by discrete (integer) amounts when a reaction **event** occurs. This picture is entirely correct, and we will study the implications of such an interpretation later. However, we will introduce the study of kinetics by thinking about a more classical chemical reaction setting of macroscopic amounts of chemicals reacting in a “beaker of water”. There, the amount of each chemical is generally regarded as a **concentration**, measured in (say) moles per litre, M, which can vary continuously as the reaction progresses. Conventionally, the concentration of a chemical species X is denoted $[X]$.

It is generally the case that the instantaneous rate of a reaction is directly proportional to the concentration (in turn directly proportional to mass) of each reactant raised to the power of its stoichiometry. We will see the reason behind this when we study stochastic kinetics later, but for now we will accept it as an empirical law. This kinetic “law” is known as **mass-action kinetics**. So, for the LV system, the second reaction will proceed at a rate proportional to $[Y_1][Y_2]$. Consequently, due to the effect of this reaction, $[Y_1]$ will decrease at instantaneous rate $k_2[Y_1][Y_2]$ (where k_2 is the constant of proportionality for this reaction), and $[Y_2]$ will increase at the same rate (since the overall effect of the reaction is to decrease $[Y_1]$ at the same rate $[Y_2]$ increases). $k_2[Y_1][Y_2]$ is known as the **rate law** of the reaction, and k_2 is the **rate constant**. Considering all three reactions, we can write down a set of ordinary differential equations (ODEs) for the system:

$$\frac{d[Y_1]}{dt} = k_1[Y_1] - k_2[Y_1][Y_2]$$

$$\frac{d[Y_2]}{dt} = k_2[Y_1][Y_2] - k_3[Y_2].$$

The three rate constants, $k_1,\ k_2$, and k_3 (measured in appropriate units) must be specified, as well as the initial concentrations of each species. Once this has been done, the entire dynamics of the system are completely determined and can be revealed by “solving” the set of ODEs, either analytically (in the rare cases where this is possible) or numerically using a computer.

It is instructive to rewrite the above ODE system in matrix form as

$$\frac{d}{dt}\binom{[Y_1]}{[Y_2]} = \left(\begin{array}{rrr} 1 & -1 & 0 \\ 0 & 1 & -1 \end{array}\right) \begin{pmatrix} k_1[Y_1] \\ k_2[Y_1][Y_2] \\ k_3[Y_2] \end{pmatrix},$$

where the 2×3 matrix is just the stoichiometry matrix, S, of the reaction system (the change in each species caused by each reaction). This leads to a general strategy for constructing ODE models from the reaction network representation.

If we write $r([Y]) = (r_1([Y]), r_2([Y]), \ldots, r_v([Y]))^{\mathsf{T}}$ for the vector of rate laws of the v different chemical reactions, then the ODE model may be derived from the state updating equation as

$$\frac{d}{dt}[Y] = S\, r([Y]).$$

We will see later that this ODE model may be regarded as a continuous deterministic approximation of the natural discrete stochastic Markov process model description given by the theory of stochastic chemical kinetics.

The dynamics for a particular combination of rate parameters and initial conditions are shown in Figure 9. An alternative way of displaying these dynamics is as an “orbit” in “phase-space” (where the value of one variable is plotted against the others, and time is not shown directly). Figure 10 shows the dynamics in this way.

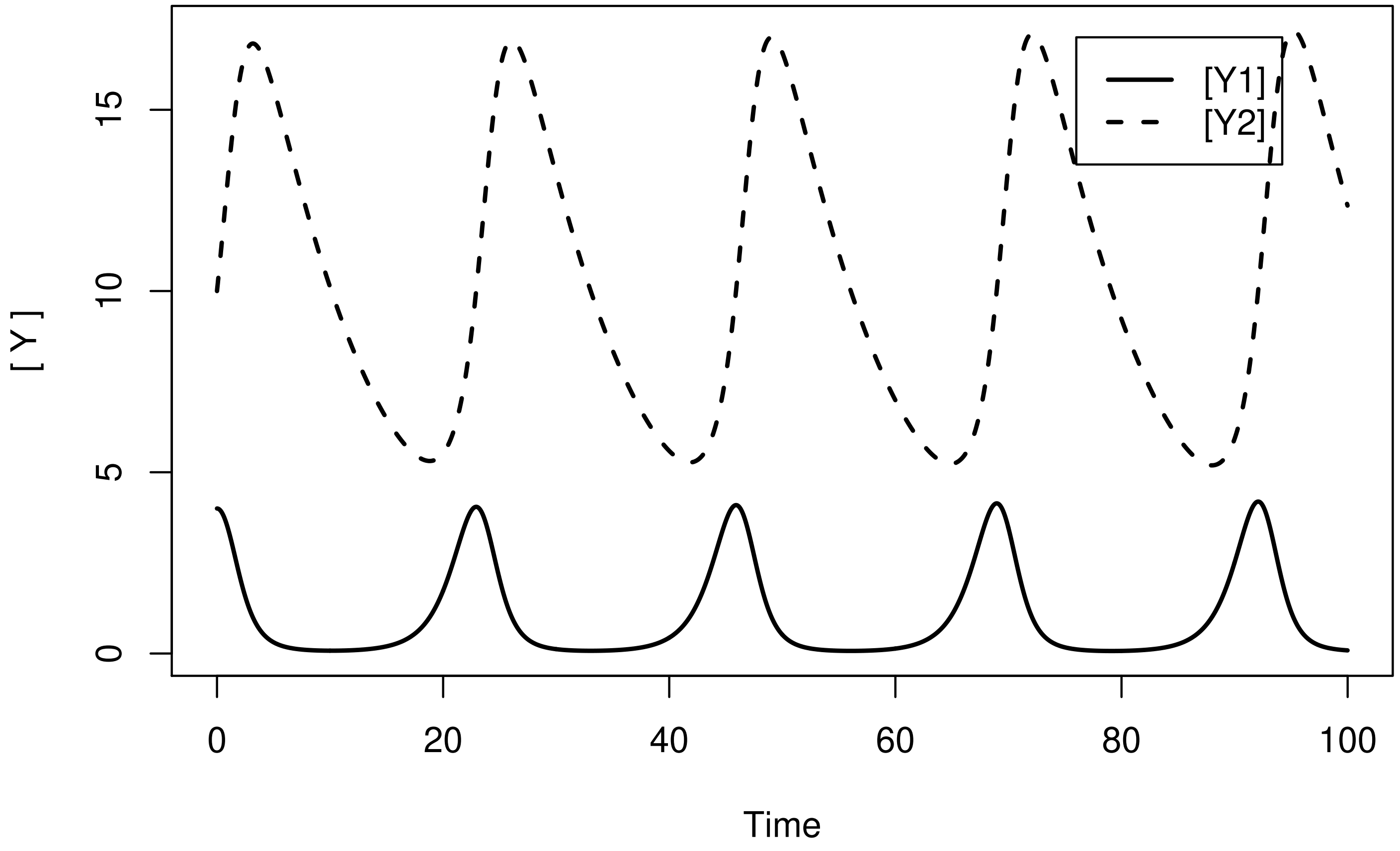

Figure 9: Lotka–Volterra dynamics for
$[Y_1](0) = 4,\ [Y_2](0) = 10,\ k_1 = 1,\ k_2 = 0.1,\ k_3 = 0.1.$

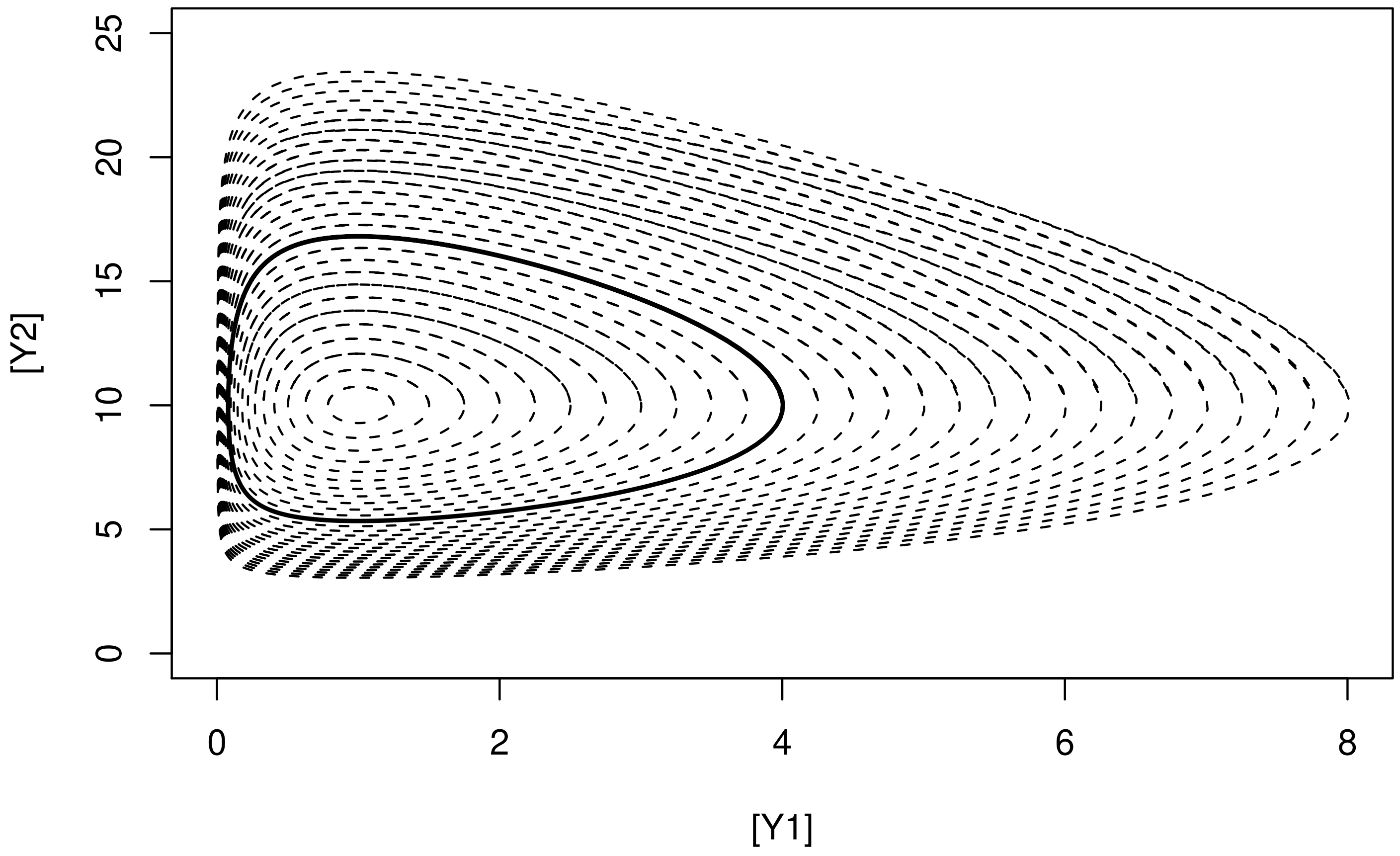

Figure 10: Lotka–Volterra dynamics in phase-space for rate parameters $k_1 = 1,\ k_2 = 0.1,\ k_3 = 0.1$. The dynamics for the initial condition $[Y_1](0) = 4,\ [Y_2](0) = 10$ are shown as the bold orbit. Orbits for other initial conditions are shown as dotted curves.

Molecular approach to kinetics

The deterministic approach to kinetics fails to capture the discrete and stochastic nature of chemical kinetics at low concentrations. As many intra-cellular processes involve reactions at extremely low concentrations, such discrete stochastic effects are often relevant for systems biology models. We are now in a position to see how chemical kinetics can be modelled in this way.

Consider a bi-molecular reaction of the form

$$X + Y \longrightarrow ?$$

(the RHS is not important). What this reaction really means is that a molecule of X is able to react with a molecule of Y if the pair happen to collide with one another (with sufficient energy), while moving around randomly, driven by Brownian motion. Considering a single pair of such molecules in a container of volume V, it is possible to use statistical mechanical arguments to understand the hazard of molecules colliding.

Under fairly weak assumptions regarding the container and its contents (essentially that it is small or well stirred, and in thermal equilibrium), it can be rigorously demonstrated that the collision hazard is **constant**, provided the volume is fixed and the temperature is constant. A comprehensive treatment of this issue is given in Gillespie (1992), to which the reader is referred for further details. However, the essence of the argument is that as the molecules are uniformly distributed throughout the volume and this distribution does not depend on time, then the probability that the molecules are within reaction distance is also independent of time. In the case of time-varying V (which can be quite relevant in the biological context), the hazard is inversely proportional to V. Again, for the careful statistical mechanical argument see Gillespie (1992), but an intuitive explanation can be given as follows. Let the molecules position in space be denoted by P_1 and P_2, respectively. Then P_1 and P_2 are uniformly and independently distributed over the volume V. This means that for a region of space d with volume v' we have

$$\mathsf{P}(P_i \in d) = \frac{v'}{V}, \quad i = 1, 2.$$

Now if we are interested in the probability that X and Y are within a reacting distance (r) of one another at any given instant of time (assuming that r is very small relative to the dimensions of the container, so that boundary effects can be ignored), this probability can be computed as

$$\mathsf{P}(|P_1 - P_2| < r) = \mathsf{E}(\mathsf{P}(|P_1 - P_2| < r|P_2))$$

but the conditional probability will be the same for any P_2 away from the boundary, rendering the expectation redundant, and reducing the expression to

$$\begin{aligned} &= \mathsf{P}(|P_1 - p| < r) \quad \text{(for any } p \text{ away from the boundary)} \\ &= \mathsf{P}(P_1 \in d) \qquad\qquad \text{(where } d \text{ is a sphere of radius } r\text{)} \\ &= \frac{4\pi r^3}{3V}. \end{aligned}$$

This probability is inversely proportional to V. Then conditional on the molecules being within reaction distance, they will not necessarily react, but will do so with a probability independent of V (as other important variables, such as the velocity distributions, are independent of V), thus preserving the inverse dependence on V in the combined probability of being within reaction distance and reacting.

Mass-action stochastic kinetics

We will consider a system of reactions involving u species $\mathcal{X}_1,\ \mathcal{X}_2, \ldots, \mathcal{X}_u$ and v reactions, $\mathcal{R}_1,\ \mathcal{R}_2, \ldots, \mathcal{R}_v$. Typically (but not always) there will be more reactions than species, $v > u$. We will assume that the qualitative structure of the reaction network can be encoded in the form of a Petri net $N = (P, T, Pre, Post, M)$, where $P = (\mathcal{X}_1,\ \mathcal{X}_2,\ \ldots,\ \mathcal{X}_u)^\mathsf{T}$ and $T = (\mathcal{R}_1,\ \mathcal{R}_2,\ \ldots,\ \mathcal{R}_v)^\mathsf{T}$. Denote the number of molecules of $\mathcal{X}_i$ at time t by X_{it}, and put $X_t = (X_{1t}, X_{2t}, \ldots, X_{ut})^\mathsf{T}$ for the state of the system at time t. Let R_{it} denote the number reactions of type $\mathcal{R}_i$ in the time window $(0, t]$, and then define $R_t = (R_{1t}, \ldots, R_{vt})^\mathsf{T}$. From the state updating equation we have

$$X_t - X_0 = SR_t, \tag{5}$$

where

$$S = (Post - Pre)^\mathsf{T}$$

is the $u \times v$ stoichiometry matrix of the reaction network.

In addition, each reaction, $\mathcal{R}_i$, will have a **stochastic rate constant**, c_i, and an associated **rate law** (or **hazard** function), $h_i(x, c_i)$, where $x = (x_1, x_2, \ldots, x_u)^{\mathsf{T}}$ is the current state (or marking) of the system (and so at time t this will be $h_i(X_t, c_i)$). The form of $h_i(x, c_i)$ (and the interpretation of the rate constant c_i) is determined by the order of reaction $\mathcal{R}_i$. In all cases the hazard function has the same interpretation, namely that conditional on the state being x at time t, the probability that an $\mathcal{R}_i$ reaction (or transition) will occur in the time interval $(t, t + dt]$ is given by $h_i(x, c_i)\, dt$. Thus, **in the absence of any other reactions taking place**, the time to such a reaction event **would be** an $Exp(h_i(x, c_i))$ random quantity. Note, however, that since the hazard depends on the state x, and other reactions could change the state, the actual time until an $\mathcal{R}_i$ reaction will typically not be exponential.

Zeroth-order reactions

First consider a reaction of the form

$$\mathcal{R}_i : \qquad \emptyset \xrightarrow{c_i} X.$$

Although in practice things are not created from nothing, it can sometimes be useful to model a constant rate of production of a chemical species (or influx from another compartment) via a zeroth-order reaction. In this case, c_i is the hazard of a reaction of this type occurring, and so

$$h_i(x, c_i) = c_i.$$

First-order reactions

Consider the first-order reaction

$$\mathcal{R}_i: \qquad \mathcal{X}_j \xrightarrow{c_i} ?.$$

Here c_i represents the hazard that a particular molecule of $\mathcal{X}_j$ will undergo the reaction. However, there are x_j molecules of $\mathcal{X}_j$, each of which having a hazard of c_i of reacting. This gives a combined hazard of

$$h_i(x, c_i) = c_i x_j$$

for a reaction of this type. Note that first-order reactions of this nature are intended to capture the spontaneous change of a molecule into one or more other molecules, such as radioactive decay, or the spontaneous dissociation of a complex molecule into simpler molecules. It is not intended to model the conversion of one molecule into another in the presence of a catalyst, as this is really a second-order reaction. However, in the presence of a large pool of catalyst that can be considered not to vary in level during the time evolution of the reaction network, a first-order reaction may provide a reasonable approximation.

Second-order reactions

For second-order reactions of the form

$$\mathcal{R}_i : \qquad \mathcal{X}_j + \mathcal{X}_k \xrightarrow{c_i} ?,$$

c_i represents the hazard that a particular pair of molecules of type $\mathcal{X}_j$ and $\mathcal{X}_k$ will react. But since there are x_j molecules of $\mathcal{X}_j$ and x_k molecules of $\mathcal{X}_k$, there are $x_j\,x_k$ different pairs of molecules of this type, and so this gives a combined hazard of

$$h_i(x, c_i) = c_i x_j\,x_k$$

for this type of reaction.

There is another type of second-order reaction which needs to be considered:

$$\mathcal{R}_i : \qquad 2\mathcal{X}_j \xrightarrow{c_i} ?.$$

Again c_i represents the hazard of a particular pair of molecules reacting. But here there are only $x_j(x_j - 1)/2$ pairs of molecules of type $\mathcal{X}_j$, and so

$$h_i(x, c_i) = c_i \frac{x_j(x_j - 1)}{2}.$$

Note that this does not match exactly the form of the corresponding deterministic mass-action rate law.

The Gillespie algorithm

The discussion in the previous sections shows that the time-evolution of a reaction system can be regarded as a stochastic process. Further, due to the fact that the reaction hazards depend only on the current state of the system (the number of molecules of each type), it is clear that the time-evolution of the state of the reaction system can be regarded as a continuous time Markov process with a discrete state space. Detailed mathematical analysis of such systems is usually intractable, but stochastic simulation of the time-evolution of the system is quite straightforward.

In a given reaction system with v reactions, we know that the hazard for a type i reaction is $h_i(x, c_i)$, so the hazard for a reaction of some type occurring is

$$h_0(x, c) \equiv \sum_{i=1}^{v} h_i(x, c_i).$$

We now follow the discrete event stochastic simulation procedure (Kendall 1950) discussed previously to update the state of the process. It is clear that the time to the next reaction is $Exp(h_0(x, c))$, and also that this reaction will be a random type, picked with probabilities proportional to the $h_i(x, c_i)$, independent of the time to the next event. That is, the reaction type will be i with probability $h_i(x, c_i)/h_0(x, c)$. Using the time to the next event and the event type, the state of the system can be updated, and simulation can continue. In the context of chemical kinetics, this standard discrete event simulation procedure is known as “the Gillespie algorithm” (or “Gillespie’s direct method”), after Gillespie (1977). The algorithm can be summarised as follows:

The Gillespie algorithm

1. Initialise the system at $t = 0$ with rate constants $c_1, c_2, \ldots, c_v$ and initial numbers of molecules for each species, $x_1, x_2, \ldots, x_u$.

2. For each $i = 1, 2, \ldots, v$, 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 $Exp(h_0(x, c))$ random quantity.

5. Put $t := t + t'$.

6. Simulate the reaction index, j, as a discrete random quantity with probabilities $h_i(x, c_i)\ /\ h_0(x, c)$, $i = 1, 2, \ldots, v$.

7. Update x according to reaction j. That is, put $x := x + S^{(j)}$, where $S^{(j)}$ denotes the jth column of the stoichiometry matrix S.

8. Output x and t.

9. If $t < T_{max}$, return to step 2.

We will examine different ways of turning this algorithm into R code in the following sections. Note that Step 6 is usually executed via some kind of “lookup method”. Efficient implementation of this step is crucial to obtaining a simulation algorithm which performs well.

Analysis of simulation output

The `smfsb` R package

Installation

We will use the package `smfsb`, which accompanies Wilkinson (2018), and is available from CRAN. It should therefore install using

```
install.packages("smfsb")
```

from any machine with an internet connection.

Once installed, the package can be loaded ready for use with

```
library(smfsb)
```

Accessing documentation

I have tried to ensure that the package and all associated functions and datasets are properly documented with runnable examples. So,

```
help(package="smfsb")
```

will give a brief overview of the package and a complete list of all functions. The list of vignettes associated with the package can be obtained with

```
vignette(package="smfsb")
```

Only one is available, and can be accessed from the R command line with

```
vignette("smfsb",package="smfsb")
```

Help on functions can be obtained using the usual R mechanisms. For example, help on the function `StepGillespie` can be obtained with

```
?StepGillespie
```

and the associated example can be run with

```
example(StepGillespie)
```

A list of demos associated with the package can be obtained with

```
demo(package="smfsb")
```

Simulation of stochastic kinetic models

Example: the Lotka Volterra model

The main purpose of the `smfsb` package is to provide a collection of tools for building and simulating stochastic kinetic models. This can be illustrated using a simple Lotka–Volterra predator–prey system.

Later we will explore additional functions for Bayesian parameter inference.

First, consider the prey, X_1 and the predator X_2 as a stochastic network as

$$\begin{array}{rcl} X_1 & \longrightarrow & 2X_1 \\ X_1 + X_2 & \longrightarrow & 2X_2 \\ X_2 & \longrightarrow & \emptyset. \end{array}$$

The first “reaction” represents predator reproduction, the second predator–prey interaction and the third predator death. We can write this in tabular form as

Pre		Post		Hazard
X_1	X_2	X_1	X_2	$h(\cdot)$
1	0	2	0	$\theta_1 x_1$
1	1	0	2	$\theta_2 x_1 x_2$
0	1	0	0	$\theta_3 x_2$

This can be encoded in R as a stochastic Petri net (SPN) using

```
# SPN for the Lotka-Volterra system
LV=list()
LV$Pre=matrix(c(1,0,1,1,0,1),ncol=2,byrow=TRUE)
LV$Post=matrix(c(2,0,0,2,0,0),ncol=2,byrow=TRUE)
LV$h=function(x,t,th=c(th1=1,th2=0.005,th3=0.6))
{
 with(as.list(c(x,th)),{
         return(c(th1*x1, th2*x1*x2, th3*x2 ))
        })
}
```

which could be created directly by executing

```
data(spnModels)
```

since this model is included in the package.

Functions for simulating from the transition kernel of the Markov process defined by the SPN can be created easily by passing the SPN object into the appropriate constructor. For example, if simulation using the Gillespie algorithm is required, a simulation function can be created with

```
stepLV=StepGillespie(LV)
```

This function can then be used to advance the state of the process. For example, to simulate the state of the process at time 1, given an initial condition of $X_1 = 50$, $X_2 = 100$ at time 0, use

```
stepLV(c(x1=50,x2=100),0,1)
```

Alternatively, to simulate a realisation of the process on a regular time grid over the interval $[0, 100]$ in steps of 0.1 time units, use

```
out = simTs(c(x1=50,x2=100),0,100,0.1,stepLV)
```

This returns an R time series object which can be plotted directly. See the help and runnable example for the function `StepGillespie` for further details.

Note that in addition to the function `simTs()` which simulates a realisation of the process on a regular time grid, there is also a function `simTimes()` which simulates a realisation on an arbitrary user-specified set of times, and a function `simSample()` which simulates many realisations from the transition kernel, providing an empirical representation of the transition density. See the help and runnable examples of these functions for further details.

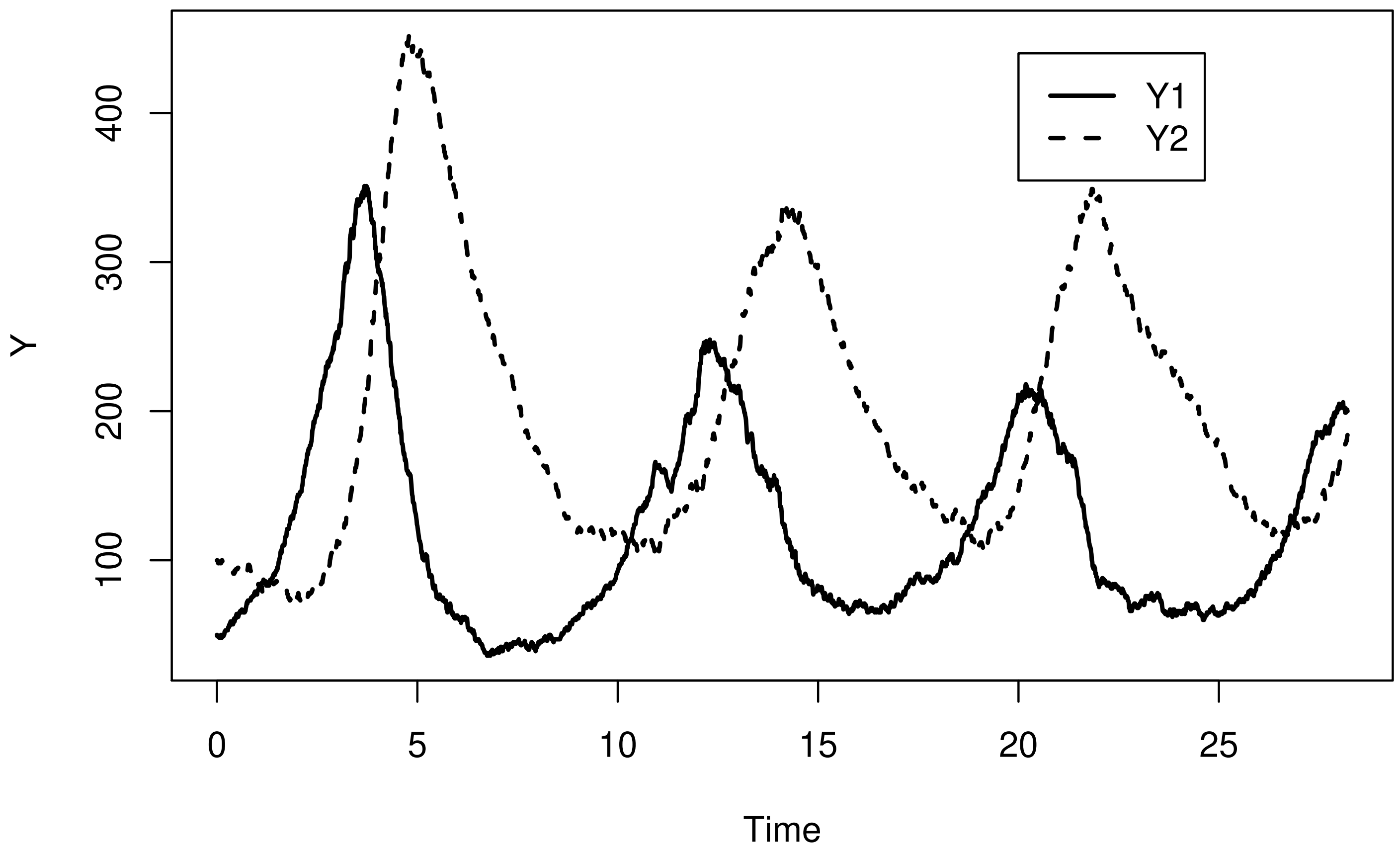

Figure 11: A single realisation of a stochastic LV process. The state of the system is initialised to 50 prey and 100 predators, and the stochastic rate constants are $c = (1, 0.005, 0.6)^{\mathsf{T}}$.

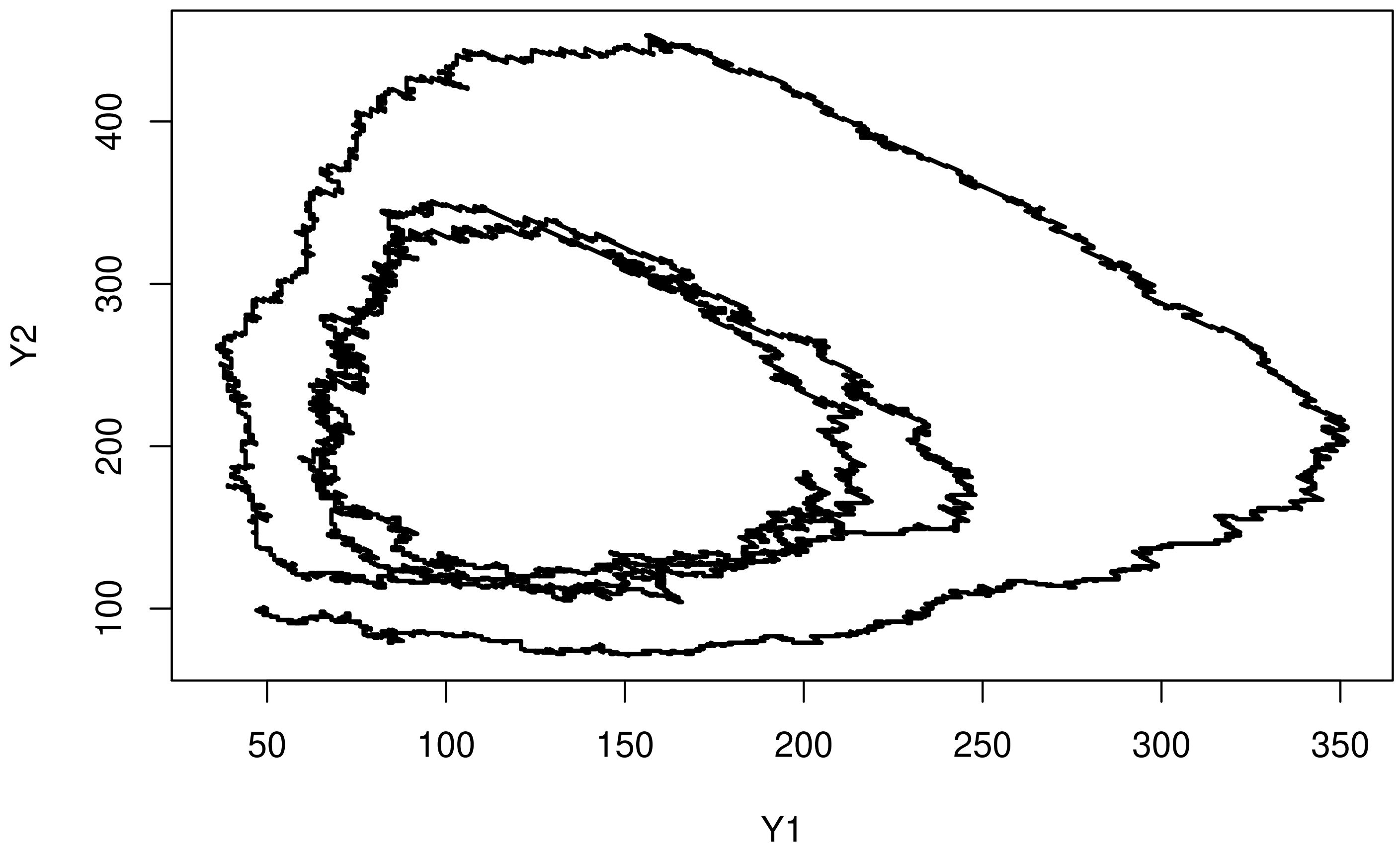

Figure 12: A single realisation of a stochastic LV process in phase-space. The state of the system is initialised to 50 prey and 100 predators, and the stochastic rate constants are $c = (1, 0.005, 0.6)^{\mathsf{T}}$.

Stochastic modelling for systems biology, third edition

Darren J Wilkinson

Chapman & Hall / CRC Press

2018

https://github.com/darrenjw/smfsb/

*

References

Gillespie, D. T. (1977), ‘Exact stochastic simulation of coupled chemical reactions’, *Journal of Physical Chemistry* **81**, 2340–2361.

Gillespie, D. T. (1992), ‘A rigorous derivation of the chemical master equation’, *Physica A* **188**, 404–425.

Golub, G. H. & Van Loan, C. F. (1996), *Matrix Computations*, third edn, Johns Hopkins University Press, Baltimore, Maryland.

Kendall, D. G. (1950), ‘An artificial realization of a simple “birth-and-death” process’, *Journal of the Royal Statistical Society. Series B (Methodological)* **12**(1).

Kloeden, P. E. & Platen, E. (1992), *Numerical Solution of Stochastic Differential Equations*, Springer, New York.

Wilkinson, D. J. (2018), *Stochastic Modelling for Systems Biology, third edition*, Chapman & Hall/CRC Press, Boca Raton, Florida.