Sgouralis research group

We consider an *initial value problem* (IVP) for the nutrient density $N(t)$ and cell population $P(t)$. According to the given ODE, our IVP is

$$\frac{dN}{dt} = -\frac{\mu_{\max} N}{N + K_s} P \bar{Q}_n, \qquad N(0) = N_0,$$
$$\frac{dP}{dt} = +\frac{\mu_{\max} N}{N + K_s} P, \qquad P(0) = P_0.$$

To avoid exceedingly small values and reduce numerical round off, here we use $\bar{Q}_n = Q_n \times 10^6$ instead of Q_n.

Our *unknowns* are the parameters $\mu_{\max}, K_s, \bar{Q}_n$ and the initial condition N_0. On them we apply *Bayesian priors;* these are

$$\mu_{\max} \sim \text{Gamma}\left(\phi_{\mu_{\max}}, \frac{\psi_{\mu_{\max}}}{\phi_{\mu_{\max}}}\right), \qquad K_s \sim \text{Gamma}\left(\phi_{K_s}, \frac{\psi_{K_s}}{\phi_{K_s}}\right),$$
$$\bar{Q}_n \sim \text{Gamma}\left(\phi_{\bar{Q}_n}, \frac{\psi_{\bar{Q}_n}}{\phi_{\bar{Q}_n}}\right), \qquad N_0 \sim \text{Gamma}\left(\phi_{N_0}, \frac{\psi_{N_0}}{\phi_{N_0}}\right).$$

Our priors exclude negative values for the unknowns that are nonphysical and allow for modeling flexibility as they assume two hyperparameters each. Our Gamma(A, B) priors have means equal to AB and variances equal to AB^2, therefore with our Gamma parametrization this means that ψ is the mean and ψ^2/ϕ is the variance, so we can control the standard deviation by using ϕ once ψ is set. Accordingly, we set up our *hyperparameters* like this

$\psi_{\mu_{\max}} = 0.6$	$\psi_{K_s} = 0.09$	$\psi_{\bar{Q}_n} = 6.7 \times 10^{-4}$	$\psi_{N_0} = 600$
$\phi_{\mu_{\max}} = 900$	$\phi_{K_s} = 11.11$	$\phi_{\bar{Q}_n} = 900$	$\phi_{N_0} = 900$

Under these choices, our prior means agree with the provided values and our standard deviations agree with the provided percentage of uncertainty.

To compare the solution of our IVP with the data provided, we apply a *likelihood;* this takes the form

$$z_n | \mu_{\max}, K_s, \bar{Q}_n, N_0, \tau \sim \text{LogNormal}\left(P(t_n), \tau\right), \qquad n = 1 : N.$$

Here, z_n denotes the cell counts at time t_n which form our data, and $P(t_n)$ is the solution of our IVP under parameter values $\mu_{\max}, K_s, \bar{Q}_n$ and N_0. Accordingly, $P(t_n)$ is an implicit function of the parameters. In addition, τ is a *precision* parameter which models how much a given measurement z_n may vary from the IVP's solution due to noise. Because τ is unknown, we also apply a Bayesian prior; this is

$$\tau \sim \text{Gamma}\left(\phi_\tau, \frac{\psi_\tau}{\phi_\tau}\right).$$

This is a conditionally conjugate prior. We configure the hyper-parameters by setting $\phi_\tau = 2$ to model a weakly informative prior and setting $\psi_\tau = 1$ arbitrarily.

Under our Bayesian model, we seek to *infer* the values of the unknowns given the data. This determines our *target distribution* which is given by $p(\mu_{\max}, K_s, \bar{Q}_n, N_0 | z_{1:N})$. In turn, this is determined after completing with

τ and the result is

$$
\begin{aligned}
p(\mu_{\max}, K_s, \bar{Q}_n, N_0 | z_{1:N}) \propto & \left(\frac{\mu_{\max}}{\psi_{\mu_{\max}}}\right)^{\phi_{\mu_{\max}}-1} \exp\left(-\phi_{\mu_{\max}}\frac{\mu_{\max}}{\psi_{\mu_{\max}}}\right) && \text{prior of } \mu_{\max} \\
& \times \left(\frac{K_s}{\psi_{K_s}}\right)^{\phi_{K_s}-1} \exp\left(-\phi_{K_s}\frac{K_s}{\psi_{K_s}}\right) && \text{prior of } K_s \\
& \times \left(\frac{\bar{Q}_n}{\psi_{\bar{Q}_n}}\right)^{\phi_{\bar{Q}_n}-1} \exp\left(-\phi_{\bar{Q}_n}\frac{\bar{Q}_n}{\psi_{\bar{Q}_n}}\right) && \text{prior of } \bar{Q}_n \\
& \times \left(\frac{N_0}{\psi_{N_0}}\right)^{\phi_{N_0}-1} \exp\left(-\phi_{N_0}\frac{N_0}{\psi_{N_0}}\right) && \text{prior of } N_0 \\
& \times \left(\frac{\phi_\tau}{\psi_\tau} + \frac{1}{2}\sum_{n=1}^{N}\left(\log\frac{z_n}{P(t_n)}\right)^2\right)^{-\phi_\tau - N/2} && \text{likelihood after marginalization of } \tau
\end{aligned}
$$

We provide a detailed derivation at the end of the writeup.

To characterize our target distribution, we develop an *MCMC sampling scheme.* Our scheme proceeds like this

- Setup MCMC sampler
  - set Bayesian hyper-parameters
  - choose sampler settings such as the number of samples to generate and proposal configurations
  - preallocate storage space
- Initialize MCMC iterations
  - generate a value for $\mu_{\max}$ by sampling from the prior
  - generate a value for K_s by sampling from the prior
  - generate a value for $\bar{Q}_n$ by sampling from the prior
  - generate a value for N_0 by sampling from the prior
- Execute MCMC iterations until the requested number of samples is produced
  - repeat 4 times
    - randomly pick a parameter from $\mu_{\max}, K_s, \bar{Q}_n, N_0$ to update
    - generate a proposal for the chosen parameter
    - compute the acceptance ratio
    - perform the acceptance test
  - save current values of $\mu_{\max}, K_s, \bar{Q}_n, N_0$
- Save MCMC samples for post-processing

Overall, our sampling scheme adopts a *Metropolis-Hastings within Gibbs* strategy. Our Metropolis-Hastings proposals are implemented via *multiplicative random walks.* These are obtained as follows.

Suppose, in an iteration $\mu_{\max}$ is chosen, multiplicative random walks generate proposals like this

- Generate a standard uniform random variate v
- Generate gamma random variates $g_1 \sim \text{Gamma}(1,1)$ and $g_2 \sim \text{Gamma}(\rho_{\mu_{\max}}, 1)$ where $\rho_{\mu_{\max}}$ is the proposal's configuration
- If $v < 1/2$
  - $\mu_{\max}^{\text{prop}} = \mu_{\max}^{\text{old}} \times (1 + g_1/g_2)$

  else
  - $\mu_{\max}^{\text{prop}} = \mu_{\max}^{\text{old}} \ / \ (1 + g_1/g_2)$

The acceptance ratio for multiplicative random walks is given by

$$a_{\mathrm{acc}} = \frac{\mu_{\mathrm{max}}^{\mathrm{prop}}}{\mu_{\mathrm{max}}^{\mathrm{old}}} \times \frac{p(\mu_{\mathrm{max}}^{\mathrm{prop}}, K_s, \bar{Q}_n, N_0 | z_{1:N})}{p(\mu_{\mathrm{max}}^{\mathrm{old}}, K_s, \bar{Q}_n, N_0 | z_{1:N})}.$$

For the other parameters, generation of the proposals and acceptance ratios are similar. For all proposals, we choose the configurations $\rho_{\mu_{\mathrm{max}}}, \rho_{K_s}, \rho_{\bar{Q}_n}, \rho_{N_0}$ heuristically.

In detail, our target is *derived* as follows

$$
\begin{aligned}
p(\mu_{\max}, K_s, \bar{Q}_n, N_0 | z_{1:N}) &\propto p(\mu_{\max}, K_s, \bar{Q}_n, N_0) p(z_{1:N} | \mu_{\max}, K_s, \bar{Q}_n, N_0) \\
&= p(\mu_{\max}, K_s, \bar{Q}_n, N_0) \int_\tau d\tau \, p(\tau, z_{1:N} | \mu_{\max}, K_s, \bar{Q}_n) \\
&= p(\mu_{\max}, K_s, \bar{Q}_n, N_0) \int_\tau d\tau \, p(\tau | \mu_{\max}, K_s, \bar{Q}_n) p(z_{1:N} | \mu_{\max}, K_s, \bar{Q}_n, \tau) \\
&= p(\mu_{\max}) p(K_s) p(\bar{Q}_n) p(N_0) \int_\tau d\tau \, p(\tau) \prod_{n=1}^{N} p(z_n | \mu_{\max}, K_s, \bar{Q}_n, \tau) \\
&\propto \underbrace{p(\mu_{\max}) p(K_s) p(\bar{Q}_n) p(N_0)}_{\text{priors}} \underbrace{\left(\frac{\phi_\tau}{\psi_\tau} + \frac{1}{2} \sum_{n=1}^{N} \left(\log \frac{z_n}{P(t_n)} \right)^2 \right)^{-\phi_\tau - N/2}}_{\text{likelihood after marginalization of } \tau} .
\end{aligned}
$$

The factors involving the priors are

$$
\begin{aligned}
p\left(\mu_{\max}\right) &\propto \left(\frac{\mu_{\max}}{\psi_{\mu_{\max}}} \right)^{\phi_{\mu_{\max}} - 1} \exp\left(-\phi_{\mu_{\max}} \frac{\mu_{\max}}{\psi_{\mu_{\max}}} \right), & p\left(K_s\right) &\propto \left(\frac{K_s}{\psi_{K_s}} \right)^{\phi_{K_s} - 1} \exp\left(-\phi_{K_s} \frac{K_s}{\psi_{K_s}} \right), \\
p\left(\bar{Q}_n\right) &\propto \left(\frac{\bar{Q}_n}{\psi_{\bar{Q}_n}} \right)^{\phi_{\bar{Q}_n} - 1} \exp\left(-\phi_{\bar{Q}_n} \frac{\bar{Q}_n}{\psi_{\bar{Q}_n}} \right), & p\left(N_0\right) &\propto \left(\frac{N_0}{\psi_{N_0}} \right)^{\phi_{N_0} - 1} \exp\left(-\phi_{N_0} \frac{N_0}{\psi_{N_0}} \right).
\end{aligned}
$$

Throughout this writeup, we use the *conventions*

$$
\text{Gamma}(x; \kappa, \theta) = \frac{x^{\kappa-1} e^{-x/\theta}}{\Gamma(\kappa) \theta^\kappa}, \qquad \text{LogNormal}(z; y, h) = \frac{1}{z} \sqrt{\frac{h}{2\pi}} \exp\left(-\frac{h}{2} \left(\log \frac{z}{y} \right)^2 \right).
$$

1. *What programming language did you use?*

   MATLAB

2. *What package / language did you use for your ODE solver?*

   `ode15s`

3. *What package / language did you use for your probabilistic commands?*

   We use the built in primitive samplers `rand` for uniform and `randg` for Gamma with unit scale, and build up from these for more complicated probabilistic commands.

4. *Did you specify an error model or was the error fitted? If you provided an error model, please provide details for how this was determined.*

   We assume a multiplicative noise model. Reasons for choosing the LogNormal include the desire for a positive distribution that scales with the mean, as well as some balance between flexibility and parsimony since LogNormal has location and scale parameters. Also, importantly, LogNormal gives us model conjugacy which allows us to leverage analytic expressions for improved algorithmic efficiency. The Gamma prior for τ is chosen for the same positivity/flexibility/parsimony/conjugacy concerns. We heuristically set the prior for the noise parameter τ to be wide.

5. *Did you test for convergence of your computations? If so, how?*

   We check convergence heuristically. We do live visualizations and look for when the sampler seems to have settled into a mode. We do many runs with randomized MCMC initializations. This allows us to get a feel for the dominant modes of the target distribution, to the point where we can spot aberrant behavior.

6. *Did you use any post-processing? Burn-in removal, thinning or similar?*

   We perform no thinning. We remove burn-in with a heuristic first 50% of the generated MCMC rule.