Transitioning from JAGS/BUGS to NIMBLE

L.C. Ponisio, Perry de Valpine, Nicholas L. Michaud

December 7, 2017

1 Steps

1. Wrap your JAGS/BUGS model code in `nimbleCode({})`, directly in R. This replaces the step of generating a file containing the model code, as is often done via `cat` or `sink` prior to calling BUGS/JAGS via `R2WinBUGS`, `R2jags`, `rjags` or `jagsUI`. (Alternatively, one can read JAGS- and BUGS-formatted code and data files using `readBUGSmodel`.)

2. Provide information about empty indices. The easiest solution is to write the index range explicitly. For example, `Omega[,]` will not work in NIMBLE, but `Omega[1:nsite,1:nspecies]` will. (Alternatively, one can use the `dimensions` argument to `nimbleModel`.)

3. Split the `data` for BUGS/JAGS into `data` and `constants` for NIMBLE. Constants are necessary to define the model, such as `nsite` in `for(i in 1:nsite) {...}`. Data are observed values of some variables, which in NIMBLE can be changed and do not need to be provided when a model is created via `nimbleModel`. (Alternatively, one can provide a list of both constants and data for the `constants` argument to `nimbleModel`, and NIMBLE will determine which is which.)

4. Convert `inits` and `monitors` inputs from functions to named lists.

5. To run an MCMC on your model, you have two choices:

   - Use `nimbleMCMC()` much like a call to `jags()` from the R2jags package. This will take all steps to set up and run an MCMC using NIMBLE's default configuration (Figure 1).
   - To use NIMBLE's full flexibiilty, build the model, configure and build the MCMC, and compile both the model and MCMC before running the MCMC. Each of these pieces can be customized and/or explored individually (Figure 1).

2 Differences

1. NIMBLE models allow alternative parameterizations for distributions (e.g. `sd` instead of `tau`), vectorized math, and user-defined functions and distributions.

2. NIMBLE allows an MCMC configuration to be customized by changing the the samplers to be used.

3. NIMBLE allows new samplers and other kinds of algorithms to be programmed from R.

4. `burnin` in NIMBLE is post-thinning – whereas it is pre-thinning in BUGS/JAGS – when using the `runMCMC` function.

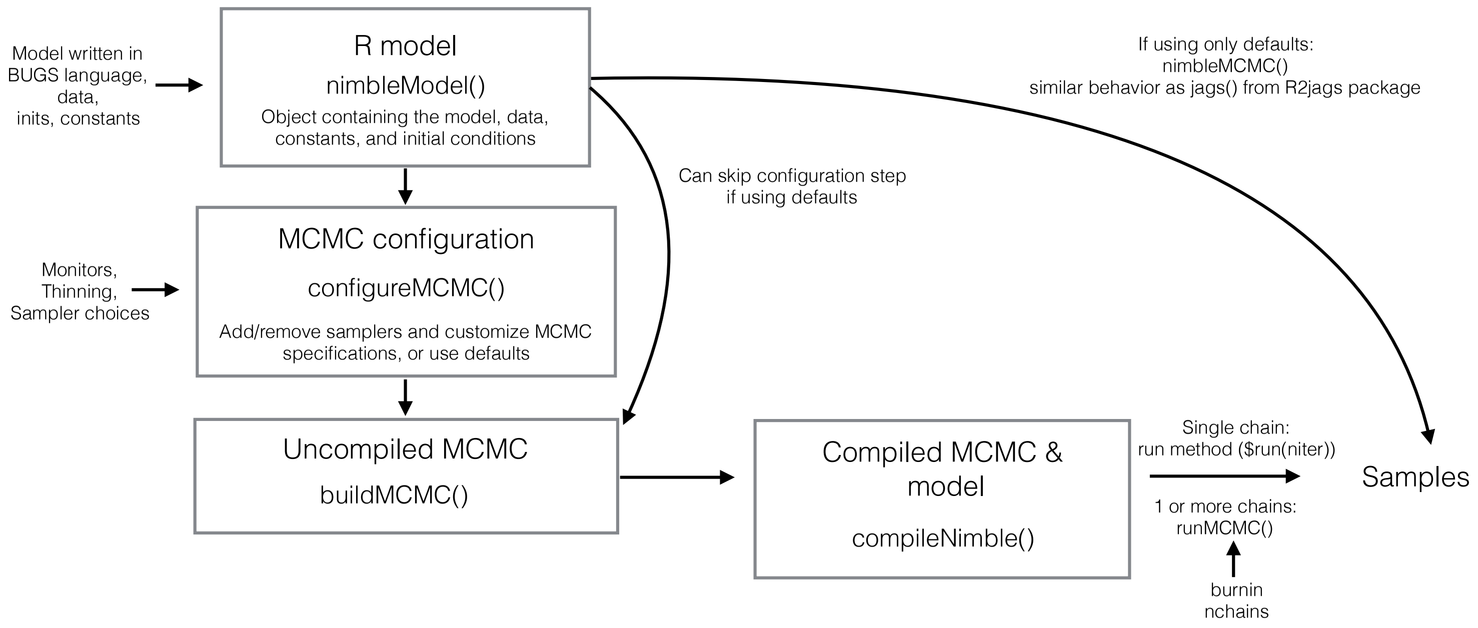

Figure 1: Workflow options within NIMBLE.