Occupancy models with NIMBLE

1 Goals

1. Showcase the functionality/efficiency of NIMBLE using occupancy modeling
2. provide code online and as supplement for common occupancy models (and a tutorial?)
3. target journal: *Methods in Ecology and Evolution*

2 Models

1. Multi-season occupancy model
   (a) vectorize Bernoulli call ✓
   (b) write a user-defined MCMC for the Zs, slice sampler for other parameters ✓
   (c) custom z sampelr and auto blocking for other paramters ✓
   (d) add reflective samplers for bounded parameters, and custom z sampler ✓
   (e) write function for latent state (Zs) likelihood ✓
   (f) write function for latent state (Zs) likelihood with auto blocking ✓
   (g) sample uniforms on logit scale?
   (h) cross level sampling?
2. Multi-species occupancy model (from Zipkin et al.)
   (a) write function for latent state (Zs) likelihood ✓
   (b) block samplers for species random effect for each type ✓
   (c) block samplers for species random effect for each species across types ✓
   (d) for all above, compute "derived quantities" as posterior sample of probabilities vs. posterior sample of binomial samples of the posterior probabilities ✓(now removed from efficiency calculation)
3. Spatial occupancy model (CAR model or distance decay)
   (a) have to implement the model
   (b) auto-blocking of parameters related to decay with distance

3 Comparisons

1. original single species, multi season model to 1) vectorized Bernoulli draw, 2) custom z sampler with slice sampler on other parameters, 3) custom z sampler with auto blocker on other paramters, 4) custom z sampler with reflective sampler on other parameters, 5) explicit likelihood for latent state, 6) explicit likelihood for latent state and auto blocking

2. original multi species, single season model to 1) explicit likelihood for latent state with vectorized calculations and expected values of derived quantities, 2) all the same but with block sampling of the species random effects

4 Results

4.1 Single species, multi season

1. JAGS and NIMBLE tie on the multi season model when the slice sampler and custom z sampler are used (same samplers as JAGS)

2. For the run with custom z on the zs and slice samplers on the uniform(0,1)s, copying takes about 10% of total time.

3. For the run with custom z on the zs and default RW samplers on the uniform(0,1)s, copying takes a bit over 20% of total time.

4. So not trivial, but also not something that if improved would cause a huge jump in performance. Improvements from algorithmic insights have potential for big jumps.

5 Considerations

1. Daniel just modified MCMCsuite() to allow a skip check argument to allow user-defined functions with dimension "mismatches" to pass the check ✓

2. try MCMCsuite() with user function with and without check

3. compareMCMC() is a new function, see Perry tutorial for an example (make sure summary=FALSE to allow for further comparisons)

6 Misc. notes

1. copy issue in nimble: In Jags each step always updates, whereas in NIMBLE certain proposals can be rejected. This enables NIMBLE to be more flexible about MCMC samplers (for example, Metropolis Hastings) but means NIMBLE has to copy more. If the proposal is rejected the original is copied to the next step. If the proposal is accepted, the new value because the original for the next step.

2. thinning may "hurt" samples that are mixing well. In contrast, a poorly mixing sampler will be "helped" because tinning helps is "mix better"

3. reflective samplers and bounded parameters: Reflective samplers might be most helpful when the posterior is skewed. In that case, the sampler would reject more often near the boundaries in the non-reflective case, so it may take longer to converge on the correct, skewed posterior. In constrast, when the posterior is not skewed, it may not matter as much because the sampler will spend less time at the doundaries.

7 Future directions

1. "parametric bootstrapping" for model evaluation, calibrated pvalues

2. avoiding the data augmentation BUGS work around paper