Vignette

Assessing Ecosystem Models

11/9/2021

Vignette

This document is designed to show users how to change MCMC and data generation inputs for our simulation study on `DALECev`.

The folder layouts are as follows:

- `src` contains relevant `C++` code for the MCMC implementation
- `setup/data_sim/` contains the scripts used to generate synthetic data for our simulation studies, as well as `R` scripts containing useful functions for the data synthesis
- `setup/MCMC/` contains useful functions as well the scripts used to setup the MCMC, including everything from allocating space for the posterior samples to generating initial starting values for the first MCMC iteration.

There are a number of simulations contained in this repository. Below is a list of them with short explanations:

- Daily time resolution model with daily observation data (`daily_timestep_daily_obs.R`)
- Daily time resolution model with monthly observation data (`daily_timestep_monthly_obs.R`)
- Daily time resolution model with monthly observation data (`daily_timestep_yearly_obs.R`)
- Monthly time resolution model with monthly observation data (`monthly_timestep_monthly_obs.R`)
- Monthly time resolution model with yearly observation data (`monthly_timestep_yearly_obs.R`)
- Monthly time resolution model with yearly observation data and NEON fluxes (`dc_neon.R`)

Each of these files will first run a script to generate data from the model, then run a script to setup the MCMC, then perform the MCMC. The code has been modified to allow users to change the following settings:

MCMC settings:

- `chain_length` - integer, number of MCMC iterations
- `burn` - integer, number of iterations for burn-in
- `observation_index` - vector, times (days or months, depending on model) where data should be observed (currently available only for `monthly_timestep_yearly_obs.R`, `daily_timestep_monthly_obs.R`, `daily_timestep_yearly_obs.R`, `dc_neon.R`)
- `stock_inits` - matrix, initial latent state estimates
- `init_type` - character, with options `gp` and `pwl`, for initialization routine to be used for latent states
- `par_inits` - vector of initial parameter estimates ($p_{1:11}, \phi_{1:5}$)
- `sd` - vector of proposal standard deviations for $p_{1:11}$
- `update` - logical, whether update plots and summaries should be printed
- `block` - list for parameters that should be updated en bloc after burn-in
- `r` - integer, currently only available for monthly timestep models. number of data cloning replicates to use (default = 1).
- `neon_test` - available only for `dc_neon.R`. specifies which flux data case to run (1 = GPP, 2 = NEE)
- `covadj` - vector, inverse multiplicative covariance adjustment for blocks during block sampling. used mainly to adjust acceptance percentage

Data generation settings:

- `p_e` - vector of 11 process parameters $p_{1:11}$ to be used for data generation
- `init_var` - vector of initial variances for $C_f, C_w, C_r, C_{lit}, C_{som}$
- `init_mean` - vector of initial means for $C_f, C_w, C_r, C_{lit}, C_{som}$
- `var_add` - vector of process variances for $C_f, C_w, C_r, C_{lit}, C_{som}$
- `var_obs` - vector of observation variances for $C_f, C_w, C_r, C_{lit}, C_{som}$

Most of these settings should have checks to ensure that things are the proper classes and dimensions. If you notice anything out of the ordinary, please feel free to reach out and I will do my best to fix it.

If no user changes are detected, the scripts will print a message to indicate that they are using the default values. Now, let's take a look to see how to change some of these inputs.

```r
## Lets start by running "monthly_timestep_yearly_obs.R" with the default settings,
## except no update and for only 100 iterations.

rm(list = ls())
## Before running any of these scripts, I suggest that you clear your
## workspace. Some settings may not work properly if this is not done.

chain_length <- 100
update = FALSE

source('./monthly_timestep_yearly_obs.R')
```

```
## No changes detected in process parameters, using default parameter set
## No changes detected in IC precisions, using default precision set
## No changes detected in IC means, using default initial means
## No changes detected in process precisions, using default precision set
## No changes detected in observation precisions, using default precision set
## No r value supplied for datacloning, using default (r=1)
## No changes detected in burnin period, using default of 2000
## No changes detected in observation index, using yearly observation data
## Initializing latent states using default, piece wise linear interpolation
## No changes detected to parameter initializations, using default
## No changes detected in proposal standard deviations, using default values
## No changes detected in block updates, using default blocks
```

We see that messages are printed to inform you about defaults being used. Additional settings may be changed by simply specifying variables from the MCMC or data generation settings. For example,

```r
rm(list = ls())

chain_length = 100
update = FALSE

## change r value this time
r = 5

## change burnin period
burn = 1000

## change proposal standard deviation
sd = rep(1, 16)

source('./monthly_timestep_yearly_obs.R')
```

```
## No changes detected in process parameters, using default parameter set
## No changes detected in IC precisions, using default precision set
## No changes detected in IC means, using default initial means
## No changes detected in process precisions, using default precision set
## No changes detected in observation precisions, using default precision set
## No changes detected in observation index, using yearly observation data
## Initializing latent states using default, piece wise linear interpolation
## No changes detected to parameter initializations, using default
## No changes detected in block updates, using default blocks
```

“‘