General strategy for implementing the hierarchical Bayesian non-steady state diffusion (BHD) model

The zip file (Code C1) contains the following files:

- ModelScript_HBD_SI (.odc or .txt): An OpenBUGS model script file that points OpenBUGS to the correct model file, loads the correct data, loads initial values for each MCMC chain, compiles the model, sets nodes (parameters) to monitor, and produces convergence plots and posterior statistics.
- ModelCode_HBD_SI (.odc or .txt): The actual OpenBUGS code that specifies the statistical and process models for the HBD model, with annotation.
- Datasets required by the HBD model: Dataset1_HBD.txt, Dataset2a_HBD.txt, Dataset2b_HBD.txt, Dataset3_HBD.txt, and Dataset4_HBD.txt
- Starting values for three MCMC chains: Initials_HBDa_S1.txt, Initials_HBDb_S1.txt, Initials_HBDc_S1.txt

Running the example code and data provided:

- One can simply evaluate the script files provided in order to run the models presented in the manuscript, using the same data used in the manuscript.

Strategies for implementing your own models:

- Make appropriate modifications to the code to reflect the experimental design.
- Try to follow similar formatting of data. Note that all categorical variables (e.g., treatment levels, dates, etc.) are coded as integer values, beginning at 1, which often serve as "indexing" variables in the model code.
- The trickiest part of implementing the models may be coming up with reasonable initial or starting values. We summarize our approach to generating the initial values used in our HBD model (see below).

Strategies for generating initials values for each chain.

- Since the HBL model tends to run much faster and is easier to troubleshoot than the HBD model, we suggest that one first implement the HBL model. Once the HBL model has converged, then use "save state" to save the parameter values generated at the last MCMC iteration, and the files produced can be used to initialize *some* of the nodes in the HBD model (the "save state" function is provided in the script files).
- In our HBD model, the following parameters (nodes) require initial values (below); ones preceded by an asterisk (*) will not be produced by the HBL save state procedure. Please see one of the initials files (e.g., Initials_HBDa.txt) for proper formatting (list syntax as used by R) for scalars, vectors, and arrays (structures). Note that we just need to get rough, ball-park estimates of the potential range of values for each node, we do not have to conduct formal, rigorous statistical analyses to generate these initials.
    - Ca.ppm
    - CorrCO2.rep
    - *Fmoles
    - *SWC100

    - *b
    - const
    - eps.C
    - mu.Ca.ppm
    - mu.Fmoles
    - tau.CO2
    - tau.Ca
    - tau.eps
    - *tau.F
- **Ca.ppm**: Use save states from HBL model, or follow procedure in Read_me_HBL.
- **CorrCO2.rep**: Use save states from HBL model, or follow procedure in Read_me_HBL.
- **Fmoles**: Three reasonable ways of getting starting values: (1) Convert the Slope values generated by the HBL save state to appropriate Fmoles units by the following conversion: Fmoles = Slope*((Press/(R*Temp))*(V/A)*1000), as done inside the HBL model code. (2) Generate initials for Slope as described in Read_me_HBL, then convert to Fmoles as done for (1). Or, (3) during an active run of the HBL model, go to Info -> Node Info, type Fmoles in the "node" cell, and click "values." Use these values (one set for each of 3 chains) as the initial values for the HBD model.
- **const**: Use save states from HBL model, or follow procedure in Read_me_HBL.
- **eps.C**: Use save states from HBL model, or follow procedure in Read_me_HBL.
- **mu.Ca.ppm**: Use save states from HBL model, or follow procedure in Read_me_HBL.
- **mu.Fmoles**: Three reasonable ways of getting starting values: (1) Convert the mu.slope values generated by the HBL save state to appropriate mu.Fmoles units by the following conversion: mu.Fmoles = mu.slope*((Press/(R*Temp))*(V/A)*1000), as done inside the HBL model code. (2) Generate initials for mu.slope as described in Read_me_HBL, then convert to mu.Fmoles as done for (1). Or, (3) during an active run of the HBL model, go to Info -> Node Info, type mu.Fmoles in the "node" cell, and click "values." Use these values (one set for each of 3 chains) as the initial values for the HBD model.
- **tau.CO2**: This is the precision (1/variance) associated with the residual observation error, which varies by global change treatment level. A decent estimate can be obtained by computing the sample variance for the initial (time zero) [CO2] values grouped by treatment level. Invert this to get the precision, and add some noise (within 1 order of magnitude, favoring smaller values) to produce initials for each chain.
- **tau.Ca**: Use save states from HBL model, or follow procedure in Read_me_HBL.
- **tau.eps**: Use save states from HBL model, or follow procedure in Read_me_HBL.
- **tau.F**: Three reasonable ways of getting starting values: (1) Convert the tau.S values generated by the HBL save state to appropriate tau.F units by the following conversion: tau.F = tau.S*(1/(((Press/(R*Temp))*(V/A)*1000))^2). (2) Generate initials for sig.S (*standard deviation*) as described in Read_me_HBL, then convert to sig.F by sig.F = sig.S*((Press/(R*Temp))*(V/A)*1000). Or, (3) during an active run of the HBL model, go to Info -> Node Info, type tau.F in the "node" cell, and click "values." Use these values (one set for each of 3 chains) as the initial values for the HBD model.
- **SWC100** and **b**: These are soil water retention parameters that have been given informative priors. Let OpenBUGS or JAGS generate values, or provide values within the prior range (e.g., around the prior mean).