## WHAT IS THE Hmsc PIPELINE?

"Hmsc pipeline" is a set of R-scripts that implement the most typical steps for applying HMSC to a case study: defining model(s), fitting the model(s), checking MCMC convergence, evaluating model fit, exploring parameter values, and making predictions over environmental gradients. The pipeline has been developed to apply generally to Hmsc models, with the aim that users can easily apply the pipeline with minimal (or even no) modification. At the same time, this means that the pipeline is not likely to perform all the analyses of interest for any given case study; for the analyses missing from the Hmsc pipeline the user needs to modify the script included in the pipeline, or write completely new scripts.

**OVERVIEW OF THE Hmsc PIPELINE.** The Hmsc pipeline consists of the following scripts:

- S1_define_models.R
- S2_fit_models.R
- S3_evaluate_convergence.R
- S4_compute_model_fit.R
- S5_show_model_fit.R
- S6_show_parameter_estimates.R
- S7_make_predictions.R

Out of these scripts, the first one (S1_define_models.R) is specific for each case study, and thus the user needs to develop this script him/herself, using the overall template given and utilizing provided examples. The remaining scripts can be run as such, or with user-made modifications if the default choices made in the scripts do not lead to the desired type of output.

## ORGANIZING THE FILES AND FOLDERS

Let us call the base folder of your case study as "base". Create subfolder "base/data". When starting the work, place the original data of your case study in the folder "data". The scripts will create folders "models" and "results", alternatively you can also generate them as originally empty folders. Copy the provided scripts S1-S7 of the pipeline into the base folder and then modify them so that they implement the analyses of your case study. In this way, you will have the pipeline specifically for your case study immediately available also for publishing your work in a reproducible manner. When running any of the scripts, make sure that your working folder is set the base folder, e.g., by adding "setwd(…)" as the first line of each script.

## INPUT-OUTPUT STRUCTURE AND DEPENDENCIES AMONG THE SCRIPTS

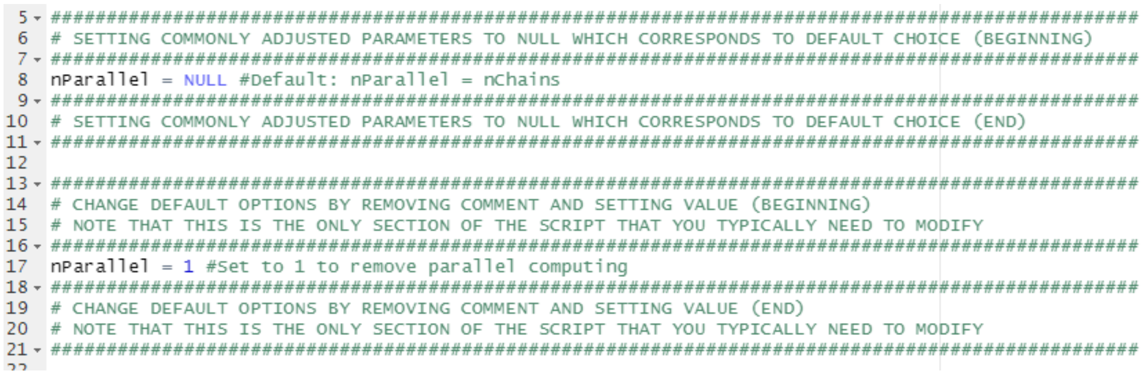
- S1_define_models.R
    - *INPUT.* Original datafiles of the case study, placed in the data folder.
    - *OUTPUT.* ***Unfitted models***, i.e., the list of Hmsc model(s) that have been defined but not fitted yet, stored in the file "models/unfitted_models.RData".
- S2_fit_models.R (you must run S1 before running this one)
    - *INPUT.* ***Unfitted models***.
    - *OUTPUT.* ***Fitted models***, with fitting done for multiple RUNs: first short MCMC chains (to provide some results fast), and then with increasingly long MCMC chains (until MCMC convergence or computational limit is reached): the files "models/models_thin_1_samples_5_chains_4.Rdata" (RUN 0), "models/models_thin_1_samples_250_chains_4.Rdata" (RUN 1), "models/models_thin_10_samples_250_chains_4.Rdata" (RUN 2),

    "models/models_thin_100_samples_250_chains_4.Rdata" (RUN 3), and so on.

- S3_evaluate_convergence.R (you must run S2 before running this one):
    - *INPUT.* ***Fitted models***
    - *OUTPUT.* MCMC convergence statistics for selected model parameters, illustrated (for all RUNs performed thus far in S3) in the file "results/MCMC_convergence.pdf", and the text file "results/MCMC_convergence.txt".
- S4_compute_model fit.R (you must run S2 before running this one)
    - *INPUT.* ***Fitted models***.
    - *OUTPUT.* ***Model fits*** computed by the cross-validation, with fitting (which is part of cross-validation) done for multiple RUNs: first short MCMC chains (to provide some results fast), and then with increasingly long MCMC chains (up to the longest run performed in S2). The results are stored in the files "models/MF_thin_1_samples_5_chains_4.Rdata" (RUN 0), "models/MF_thin_1_samples_250_chains_4.Rdata" (RUN 1), "models/MF_thin_10_samples_250_chains_4.Rdata" (RUN 2), "models/MF_thin_100_samples_250_chains_4.Rdata" (RUN 3), and so on.
- S5_show_model_fit.R (you must run S4 before running this one)
    - INPUT. ***Model fits***.
    - *OUTPUT.* Model fits illustrated (for highest RUN of S4) in the file "results/model_fit.pdf".
- S6_show_parameter_estimates.R (you must run S2 before running this one)
    - INPUT. the ***Fitted models***.
    - OUTPUT. Parameter estimates illustrated (for highest RUN of S2) in the file "results/parameter_estimates.pdf", the text file "results/parameter_estimates.txt", as well as given numerically in multiple csv files (one per parameter type) named "results/parameter_estimates_[parameter_name].csv".
- S7_make_predictions.R (you must run S2 before running this one)
    - INPUT. the ***Fitted models***.
    - OUTPUT. Predictions over environmental gradients (for highest RUN of S2) in the file "results/predictions.pdf".

**MORE SPECIFIC INFORMATION ABOUT THE SCRIPTS, INCLUDING TYPICAL USER-MODIFICATIONS**

You can run all scripts S2-S7 with default choices simply by pressing control-shift-enter (or copy pasting the hole script into the console). We recommend doing so at the first stage, and tailoring the scripts for your purposes only at a later stage. While you can of course modify any parts of the scripts, the most typically needed modifications can be easily made in the beginnings of the script. The parameters guiding these choices are set to NULL in the beginning of each script. If the value is kept as NULL, the script will apply the default choice. If the value is modified by the user, the user-set value will be applied. For example, for script S2 the default choice is to apply parallel computing over the MCMC chains (nParallel=nChains). In the code below, the user has removed the commenting symbol # from "#nParallel = 1" and thus chosen not to run the chains in parallel, possibly because the computational resources do not support parallel computing, or because the user wishes to track how the MCMC chains are proceeding (there is no output to the console when applying parallel computing).



```
##################################################################################################
# SETTING COMMONLY ADJUSTED PARAMETERS TO NULL WHICH CORRESPONDS TO DEFAULT CHOICE (BEGINNING)
##################################################################################################
nParallel = NULL #Default: nParallel = nChains
##################################################################################################
# SETTING COMMONLY ADJUSTED PARAMETERS TO NULL WHICH CORRESPONDS TO DEFAULT CHOICE (END)
##################################################################################################

##################################################################################################
# CHANGE DEFAULT OPTIONS BY REMOVING COMMENT AND SETTING VALUE (BEGINNING)
# NOTE THAT THIS IS THE ONLY SECTION OF THE SCRIPT THAT YOU TYPICALLY NEED TO MODIFY
##################################################################################################
nParallel = 1 #Set to 1 to remove parallel computing
##################################################################################################
# CHANGE DEFAULT OPTIONS BY REMOVING COMMENT AND SETTING VALUE (END)
# NOTE THAT THIS IS THE ONLY SECTION OF THE SCRIPT THAT YOU TYPICALLY NEED TO MODIFY
##################################################################################################
```

*Shown is part of the beginning of the script S2 illustrating how user can modify some choices made in the scripts by replacing NULL by some other value.*

## RECOMMENDATIONS ON HOW TO APPLY THE PIPELINE

A typical “user mistake” is to try to define the final model immediately, and to fit that model until MCMC convergence before examining preliminary results. Doing so easily leads to a situation that after considerable running time (hours, days, or even weeks) the user finally explores the results, just to realize that the results are not as expected and thus something was wrong in how the model was defined (say; a continuous covariate was accidentally treated as a factor). Also, even if the final run might take weeks, usually the preliminary results can be obtained already after some minutes/hours, making it possible to develop the rest of the pipeline (e.g., case-specific components) and even write the results section of your manuscript before waiting for long computations. Those results can then be updated once the longer runs have been finished. We thus recommend the following.

- Make first a highly simplified version of your model, without worrying too much about the details (exactly which covariates to select, whether a phylogenetic tree or traits are included or not, etc.). Run that model through the entire pipeline (with small number of MCMC iterations) to see that you get roughly the output that you expected. After that it is time to think more deeply about the final model structure.
- The model fitting (and cross-validation) is organized through the sequence of RUNs with increasing number of MCMC samples. The chains are thinned so that (after RUN 0 which is mainly to check that no errors appear) each RUN produces 250 samples for four chains, and thus 1000 posterior samples in total. The sole reason why thinning is done is to keep the file sizes of the fitted model objects in check, as without thinning they could be of the order of gigabytes for large models. For usual requirements inference, 1000 posterior samples are well enough, as long as those samples are (essentially) independent, for which reason a high value of thinning may be needed.
- A typical question is that “why to run first shorter MCMC chains if in the end a long one is needed”. The answer is that running first the short ones gives you immediate results, and you do not need to decide beforehand “how long chain I will eventually need”, as the MCMC convergence results will show that *post hoc.* Note that by running the shorter chains first does not take essentially any more time, as in our pipeline the next longer run always takes ca. 10 times more time than the previous one. Hence running all the RUNs in our pipeline takes only ca. 11% more time than running the final RUN alone. A further argument is that by running through the sequence of RUNs you will not be in the dark about whether your script will finish in hours,

days, or weeks: the first RUNs will be fast, and you can always predict that the next RUN takes ca. 10 times more time than the previous RUN (except RUN 1 takes ca. 50 times longer than RUN 0).

- All scripts have been written so that they take well defined input and yield well defined output. No “hidden variables” remain in the memory. Thus, you can close R and “start fresh” for running any part of the pipeline.
- The model fitting (including cross-validation) scripts are written so that they examine from the “models” folder which is the highest RUN completed thus far, and then start from the next RUN. Thus, if you interrupt these scripts and restart, the scripts will not start from the beginning of the RUNs but jump directly to the RUN not completed yet (and restart that one). The outputs will be generated for the highest RUN performed, except for MCMC convergence which is generated for all RUNs so you can examine at which rate MCMC convergence is achieved.
- We recommend that in the end of Script S1_define_models.R you run a very short MCMC to see if model fitting goes without error messages before setting up the computations to e.g. a cluster. Use e.g. these lines: for(i in 1:length(models)){sampleMcmc(models[[i]],samples=2)}.
- Even if you have a separate computational cluster, we recommend that you run the first RUNs in your laptop/desktop where you develop the model (i.e., where you write script S1), as that allows you to initially fast move through the pipeline and e.g., pick up any obvious errors. Even RUN 0 is sufficient for this, but it is beneficial to go through also higher RUNs, until the computations start taking substantial time and distract your working (say, hours).
- If you have a dedicated cluster, we recommend that you perform the model fittings (S2 and S4) there but apply the rest of the pipeline (i.e., the computationally non-intensive parts) in your laptop/desktop. Set up the model fitting (as is the default for S2 and S4) for runs R0, R1, … until a very high thinning such as thin=10,000 (RUN 5), even if you do not know *a priori* whether RUN 5 will be computation ally feasible. As the RUNs proceed, copy the ***fitted models*** and ***model fits*** to your local computer and run the rest of the pipeline. If you pick up any problems, refine the model and re-start as needed. If you do not pick up any problems, you may work on the analyses based on the highest RUN finished thus far and let the model fitting continue in the background. Interrupt the model fittings (S2 and S4) after MCMC convergence has been reached or after you have reached your computational limit.