# Guide to Running the Simulation

This simulation consists of an R script that is designed to run on a cluster. Parallelization occurs across the range of simulated years, which is 302 years across the combined events (151 years each for the LST and the YDB). Optimally, one year is assigned per available core, plus one additional core for the master process. If this is accomplished, execution time is mainly limited by the number of simulation iterations. The results in this paper were obtained by executing the R script on the ManeFrame II cluster at Southern Methodist University (SMU), utilizing one core per simulated year. Under these circumstances, a 10,000-iteration simulation completes in about 36 hours, a 1000-iteration simulation completes in just under four hours, and 100 iterations completes in about 0.5 hours. Simulations under 1000 iterations generally provide noisy output distributions of $MPMD_E$ and $^{14}C_E^g$. If cores are limited and multiple years are run per core, execution time will increase dramatically depending on the core that is assigned the most years over which to simulate (henceforth, maximum-core-years). Estimated execution time is roughly maximum-core-years multiplied by the above specified time estimates. For example, if maximum-core-years is three for a 10,000-iteration simulation on the ManeFrame II cluster, the expected time is 3*36 hours.

### 1.1. Requirements

RStan: The LBM is fit via Hamiltonian Monte Carlo simulation in Stan (1). Visit https://www.mc-stan.org for installation details.

R packages: *ggplot2* (2), *parallel* (3), *reshape2* (4), *rcarbon* (5), *matrixStats* (6), *patchwork* (7), and *rethinking* (8). The first five packages are available in the CRAN.
*patchwork* is available at: https://github.com/thomasp85/patchwork
*rethinking* is available at: https://github.com/rmcelreath/rethinking

Data files: *IRI.csv* (table of reported $^{14}$C measurements from the Fifth International Radiocarbon Intercomparison), *RCmeasurements.csv* (table of dates reported for the Laacher See Tephra and Younger Dryas Boundary).

The script must be executed on a cluster with cores distributed across nodes. The main file output (*SimDat#.RData*) for the 10,000-iteration simulation is 587 MB, and there are minor memory spikes during the simulation (intermediate R objects are created during the simulation that are not included in the main output file).

### 1.2. Running the R Script

Place the R script, *IRI.csv*, and *RCmeasurements.csv* in your working directory. Ensure that RStan and all required packages are installed. Follow these steps:

1. Adjust user-arguments for simulation. Navigate to the 'USER ARGUMENTS' block of code in the R script to adjust the simulation as appropriate (code lines 220-305). This block contains variables that specify the number of nodes to be used, the number of cores per node, the number of simulation iterations, plotting options, the ranges of calendar years over which to simulate each event, OWM parameter values, and other variables of interest. Inline code comments further detail each variable.
2. Source the R script. This will load the csv data files and prepare parameters for the simulation. On the first run, the script fits the LBM, which may take 5-15 minutes. After the model is fitted, posterior parameter values are exported in RData file *IRI#.RData* (where # is the number of user-specified iterations for the simulation). If you leave this file in your

working directory and you plan to run the simulation again in the future, the script will read *IRI#.RData* into the simulation rather than refit the model, saving run time. Every time a simulation is run with a new number of iterations (i.e., new # values), the model will be refitted. As such, users can store multiple *IRI#.RData* files for running the simulation with different numbers of iterations. The correct file will be read for each simulation if it has already been generated in the working directory.

3. Monitor the working directory for intermediate output files. In addition to *IRI#.RData*, the script will also output *SimDatIntermediate.RData*. This contains simulation parameters to be read by each node. If you wish to delete this file, do not so until the simulation has initiated on every node (i.e., every node has imported the simulation parameters from *SimDatIntermediate.RData*). Immediately following the creation of *SimDatIntermediate .RData*, the main R script will create daughter R scripts with the filename *NodeSim#.R*. The number of these scripts that is created corresponds to the number of user-specified nodes. After these scripts appear in the working directory, move to Step 4.

4. Submit daughter scripts to the cluster. The main R script also outputs an sbatch array submit script that can be executed to request nodes for every daughter R script (*nodesim.sh*). This is formatted to run on a partition of the ManeFrame II cluster at SMU, but it can be easily edited to run on other clusters using a Slurm workload manager. Alternatively, you may submit the daughter scripts to nodes using your own method.

5. Wait for results. After you submit the daughter scripts, the main R script waits for them to complete (it scans the working directory every 30 seconds for output from the daughter scripts). When a daughter script completes, it outputs *NodeDat#.RData*. After all daughter output files are present in the working directory, the main script automatically imports them, aggregates the results, and creates one output RData file (*SimDat#.RData*, where # is the number of user specified simulation iterations).

### 1.3. Results and Output Files

Simulation results are contained in objects stored in *SimDat#.RData.* Comments in the script describe these objects. This file is automatically written to the working directory after the simulation completes. *SimDat#.RData* can be opened and explored in an interactive R session on a personal computer**.** Although the intermediate output files may be of interest (*IRI#.RData, SimDatIntermediate.RData, NodeDat#.RData, NodeSim#.R, nodesim.sh*), they do not contain the primary results and may be deleted after the simulation is completed.

## Citations

1. Stan Development Team (2018) *RStan 2.17.2: The R interface for Stan* Available at: http://mc-stan.org.

2. Wickham H (2016) *ggplot2: Elegant Graphics for Data Analysis* (Springer-Verlag, New York).

3. R Core Team (2018) *R: A language and environment for statistical computing.* (R Foundation for Statistical Computing, Vienna, Austria) Available at: URL http://www.R-project.org/.

4. Wickham H (2007) Reshaping data with the reshape package. *Journal of Statistical Software* 21(12):1–20.

5. Bevan A, Crema ER (2018) *rcarbon v1.2.0: Methods for calibrating and analyzing radiocarbon dates* Available at: https://CRAN.R-project.org/package=rcarbon.

6. Bengtsson H, et al. (2018) *matrixStats: Functions that Apply to Rows and Columns of Matrices (and to Vectors)* Available at: https://cran.rstudio.com/web/packages/matrixStats/index.html.

7. Pedersen TL (2018) *patchwork: The Composer of ggplots*.

8. McElreath R (2016) *rethinking v1.59: Statistical Rethinking book package* Available at: https://www.github.com/user/rmcelreath/rethinking.