

## Appendix 2: Code, data, and information for model integration example 2

### 1 Building and running the model

Building the model requires an installation of R as well as a C++ compiler (see **Dependencies** below for information on all software required to fully build and run the model). For users familiar with Gnu Make, a makefile is provided as a guide. To perform the compilation of the Metropolis sampler (and thus to check for errors in the build process that would prevent the model executing properly), use `make model`. Once the model is built, there are five primary steps to run the model. To run them all in series, use `make ex2.pdf`. The exact procedures for each step are documented in detail in the source files, and summarized below.

#### 1.1 Setup

```
Rscript 1-setup_naive_model.r or  
make dat/naive_model.rdata
```

This step formats the raw data, builds the naive model (which also provides the structure for the metamodel), and sets up the data to be used as input for running the integrated model. We divided the dataset into calibration and validation subsets. To avoid over representing absences in the calibration set, we initially identified the ratio of presences to absences in the original data (approximately 1:13), then randomly chose 1/3 of the presences for validation and 2/3 for calibration, and finally chose absences for the calibration set to maintain the original presence:absence ratio. For validation, we used a number of absences equal to the number of presences in the validation set, selected from the remaining absences.

To select the form of the naive model (and thus the metamodel), we used stepwise regression with BIC as the evaluation criteria. Exploratory analysis revealed that there were collinearities within the predictor variables. Thus, we only included three variables in the stepwise regression: number of degree days (`ddeg`), mean summer precipitation (`sum_prctp`), and the ratio of mean annual precipitation to potential evapotranspiration (`pToPET`). The search scope included all models with linear, quadratic, and cubic terms for all three variables.

#### 1.2 Integration

```
./bin/integrated_model >results/integratedModel.csv or  
make results/integratedModel.csv
```

This step starts the Metropolis sampler to estimate the parameters of the integrated model. Specific details can be found in the C++ files in the src directory of this appendix. The model first reads data from files produced in the previous step, then performs some automatic adaptation to improve the efficiency of the sampler, and finally runs for (by default) 50,000,000 iterations.

#### 1.3 Discard burn-in and thin samples

```
Rscript 3-process_integrated.r or  
make results/integratedModel.rdata
```

We used preliminary runs with multiple chains to determine for how long the MCMC should be run before the chain converged on its stationary distribution and to choose a thinning interval. We discarded the first 500,000 samples as not representative of the posterior distribution of the parameters, and chose a thinning interval of 50 to reduce autocorrelation in the posterior samples.

## 1.4 Compute posterior distribution of predictions

```
./bin/stats >results/integratedStats.csv or  
make results/integratedStats.csv
```

This is a C++ program that steps through each (thinned) posterior sample and predicts the present and future probability of presence at every point in the original (i.e., calibration+validation) dataset. The output is a table, where each row is a data point and the columns are the mean probability of presence, the standard error, the 5% quantile, and the 95% quantile for present and future climate.

## 1.5 Build the figure

```
Rscript 5-make_figure.r or  
make ex2.pdf
```

Using the predicted probabilities from Phenofit, the naive model, and the integrated model, this script will produce Figure 5 as found in the manuscript.

## 2 Dependencies

Much of the data preparation and analysis uses R. In addition to base R, the following packages are required:

- sp
- fields
- rgdal
- glm2
- coda

To fully build and run this example, a C++ compiler that is compliant with C++11 is required. We recommend an openMP-compliant compiler for best performance (GCC 4.4 or greater will work). The makefile includes make procedures for building the project and includes a node for building the entire model: use `make model` at the command line to invoke it. However, compilation procedures vary greatly from platform to platform. The makefile should be considered a guide for compiling the program, but it will likely have to be edited before it functions properly. Finally, this project uses the Gnu Scientific Libraries (GSL); a copy of this is required to compile the project.