

GUIDE TO THE CODE

PREDICTING HOW MANY ANIMALS WILL BE WHERE: HOW TO BUILD, CALIBRATE AND EVALUATE INDIVIDUAL-BASED MODELS

Elske van der Vaart, Alice Johnston & Richard Sibly
Ecological Modelling (2015)

This guide offers a brief introduction to the code that accompanies the above paper. The paper summarizes earlier work and then continues with a further example; for the earlier code [1], see the **figshare** repository <http://dx.doi.org/10.6084/m9.figshare.1412720>.

All the new code – relating to the population level results in §3.3 – can be found in a new **figshare** repository: <http://dx.doi.org/10.6084/m9.figshare.1494754>. If not, check there for possible updates! In the following, familiarity with the paper itself is assumed.

There are two kinds of code in the repository: NetLogo code (.nlogo) and *R* code (.R). Both NetLogo and *R* are programming environments that are freely available for all operating systems; NetLogo is for IBMs, while *R* is for statistics. Accordingly, the paper's earthworm IBM is written in NetLogo, while its ABC calculations are done in *R*.

This guide will explain how to look at the earthworm IBM, and how to re-create the ABC results given in the paper. Any questions, e-mail <elskevdv@gmail.com>.

- Elske van der Vaart, 3 August 2015

LOOKING AT THE EARTHWORM IBM

As stated, the earthworm IBM is written in NetLogo, which can be downloaded from here: <http://ccl.northwestern.edu/netlogo/>

Working with NetLogo models is very easy; NetLogo itself can be installed with a graphical installer, and then individual models can be opened from a drop-down menu. Each model has a visual interface and very human-readable code that can be accessed from another tab; for the earthworm IBM, the visual interface already offers some parameters to play with and some graphs to assess the model's performance as it's running.

This NetLogo model, `monroy.nlogo`, was developed by Alice Johnston and colleagues, and is described in detail in [2]; we have re-coded it slightly. To look at it, simply open it from NetLogo's drop-down menu. Press "Setup", then "Set Basic Interface", "Set ABC Parameters" and then "Go to End of Data" in the 'Interface' tab to watch it run. The graphs will display the model's output in terms of numbers of worms in grey and the empirical data in black; they should roughly match the top row of Figure 4. [Note that, if the model is run with "Set Basic Parameters", the output doesn't quite match [2]; this is due to a fixed coding issue.]

The model code itself is under the 'Code' tab.

RUNNING THE ABC ANALYSES IN R

To reproduce the paper's new ABC results, first download *R*: <http://www.r-project.org>. *R* can be installed with a graphical installer, and then from within *R*, it's necessary to also download the *R* **package** *car*. This should be possible from a drop-down menu, but as the process differs across versions and operating systems, I'm refraining from giving specific instructions. Make sure to download all dependencies as well.

the simulation results

It will also be necessary to download the new simulation results, available from a separate **figshare** repository due to licensing issues: <http://dx.doi.org/10.6084/m9.figshare.1494757>. The files in this repository are the following:

- `monroy_data.rds` – a file with all of the empirical data as plotted in Figure 4;
- `priors_monroy.rds` and `results_monroy.rds` and `results_full_2.rds` – all one million priors and results stored in *R*'s file format; each row in *priors* contains the parameter settings for the corresponding row in *results*

organising all files correctly

To actually run the code, open the file 'Commands.R' in *R* (this can again be done from a drop-down menu). In theory, setting two paths in this file and then executing it (for instance, by copying and pasting it into *R*'s console) will produce all of the results figures and tables in the paper. However, it does assume a certain file structure, which looks like this:

- Main folder: name of your choice, e.g., 'levels'
 - Subfolder: 'results' [all files from the 'Runs' **figshare**]
 - `monroy_data.rds`
 - `priors_monroy.rds`
 - `results_monroy.rds`
 - Subfolder: 'src'
 - Subfolder: 'models' [the .nlogo file from the 'Code' **figshare**]
 - `monroy.nlogo`
 - Subfolder: 'R' [all .R files from the 'Code' **figshare**]
 - `ABCOBJECT.R`
 - `Commands.R`
 - `ParameterEstimation.R`
 - `PosteriorChecking.R`

actually running the code – Commands.R

The code in 'Commands.R' is the 'skeleton' for what's in the paper: It executes all of the necessary commands, but most of the 'real work' occurs in other scripts (i.e., all of the other *R* scripts in the 'R' folder described above). These scripts have sensible names and are organised into functions, with some annotation; please look at them to truly see what's happening. The only slightly opaque script is "ABCOBJECT.R", which contains functions to make sure objects of class "abcObject" are printed to the console in a pretty fashion.

Some of these functions use code from the very, very useful *R* package `abc` [3]; the reason we don't use this package directly is because it normalises simulation results by a metric that's unsuitable in our case, as described in our first ABC paper, [1].

Now, to run the 'Commands.R' script up until the 'Doing Posterior Checking' section, set the 'f.path' variable to the location of the 'main folder' (e.g., 'levels') as described above.

The posterior check as performed in the paper – Figure 4 – has its own section in 'Commands.R' because it is a bit trickier to run. It will require downloading NetLogo (as described under 'Looking at the Earthworm IBMs') and the *R* package *RNetLogo* [4, 5], as well as setting the 'nl.path' variables to the correct location. In theory this should be easy – *RNetLogo* can be downloaded from within *R* just like any other package – but in practice it

may prove difficult. *RNetLogo* relies on the *R* package *rJava* which in turn requires a Java installation; that's a lot of opportunities for things to go wrong.

Re-doing the full posterior check as shown in Figure 4 also takes quite a bit of time; that's why the *R* code provided takes only 5 draws from the model's posteriors instead of 100. Adjust to 100 to actually re-create the figure, but be prepared to wait for about an hour. In addition, because both the draws and the model's output are stochastic, the re-drawn figure will not be exactly identical to the one shown in the paper. Alternatively, set 'rerun' to FALSE; this will plot 100 random draws from the model's accepted runs, without re-running the model. This should also work if installing *rNetLogo* fails.

troubleshooting rNetLogo

One problem we have encountered is when *R* and *Java* aren't installed using the same (32 or 64) bit version; then *R* throws an error which can be fixed by re-installing the correct *Java* version. Another issue we have encountered is that the latest version of *RNetLogo* (1.0 – 1) does not appear to work with Mac OS X; *RNetLogo* 1.0 – 1 requires *Java* 7, but *R* insists on using *Java* 6. (See <http://stackoverflow.com/questions/26745364/getting-r-to-use-newer-versions-of-java> for more information.) Currently we are aware of no stable way to fix this other than rolling back to *RNetLogo* 1.0 – 0.

The *RNetLogo* package itself is very well documented and its manual, vignette & associated papers [4, 5] should solve many problems. (It's a lovely package – I recommend it strongly to anyone working with both *R* and *NetLogo* for whatever reason!) However, feel free to contact me if this step (or indeed, any step) proves difficult.

REFERENCES

1. van der Vaart, E., et al., *Calibration and evaluation of individual-based models using Approximate Bayesian Computation*. Ecological Modelling, 2015.
2. Johnston, A.S.A., et al., *An energy budget agent-based model of earthworm populations and its application to study the effects of pesticides*. Ecological Modelling, 2014. **280**: p. 5-17.
3. Csillery, K., O. François, and M.G.B. Blum, *abc: An R package for approximate Bayesian computation (ABC)*. Methods in Ecology and Evolution, 2012. **3**: p. 475-479.
4. Thiele, J.C., W. Kurth, and V. Grimm, *RNetLogo: An R package for running and exploring individual-based models implemented in NetLogo*. Methods in Ecology and Evolution, 2012. **3**: p. 480-483.
5. Thiele, J.C., W. Kurth, and V. Grimm, *Facilitating parameter estimation and sensitivity analysis of agent-based models: A cookbook using NetLogo and R*. Journal of Artificial Societies and Social Simulation, 2014. **17**: p. 11.