MAGYARORSZÁG KORMÁNYA

Európai Unió
Európai Regionális
Fejlesztési Alap

BEFEKTETÉS A JÖVŐBE

SZÉCHENYI 2020

AgroMo Technical Note vol. 2: Sensitivity analysis of Biome-BGCMuSo using the RBBGCMuSo package

by Roland HOLLÓS, Dóra HIDY, Zoltán BARCZA

AgroMo

Martonvásár
2019

Sensitivity analysis of Biome-BGCMuSo using the RBBGCMuso package

Roland HOLLÓS, Zoltán BARCZA

January 7, 2020

Installation of required software

Before start using the RBBGCMuso package, you have to install it on your computer under the R environment. Note that we recommend using the latest R package, and we also suggest to use the latest RStudio software.

At this stage of development, you have to have Rtools and devtools installed on your Windows/Linux box. Once the RBBGCMuso package will be finished and documented, it will be available throught CRAN (which is the official repository of R), which means that devtools and Rtools will not be needed.

In order to install Rtools, please visit this link: https://cran.r-project.org/bin/windows/Rtools/, then download the installation file and install it.

The installation of the devtools package is much easier:

```r
install.packages("devtools")
```

Installation of RBBGCMuso

After you installed devtools and Rtools, we propose to install the most recent RBBGCMuso package and load it using the following commands in R:

```r
devtools::install_github("hollorol/RBBGCMuso/RBBGCMuso",upgrade="never")
library(RBBGCMuso)
```

Preparations

Sensitivity analysis aims to select the most influential ecophysiological parameters of the model. This is needed because model calibration (or in other words, parameter optimization) is a computationally demanding procedure, and reducing the number of parameters is highly needed for optimal run time.

First prepare a directory (folder) where you have a complete input set (spinup.ini, normal.ini, soil, meteorology, EPC, ancillary CO2 and Ndep files…) to run Biome-BGCMuSo together with the model executable (you should call the executable as “muso.exe” under MS Windows, and simply “muso” under Linux).

You might want to download the sample input dataset from the website of the model, which is a grassland simulation prepared for the HU-He2 eddy covariance site: http://agromo.agrar.mta.hu/bbgc/files/HU-He2-muso6.zip. If you use this sample dataset, please do not forget to copy the model executable to this directory. Windows execuatable is available at the website of the model: http://agromo.agrar.mta.hu/bbgc/download.html.

Alternatively, you can easily get a sample dataset (including model executable) with the application of the copyMusoExampleTo command in RBBGCMuso. Here we refer to the RBBGCMuso GitHub page for further details (https://github.com/hollorol/RBBGCMuso).

In this example we assume that the directory is C:\biome-bgcmuso\sensitivity\.

We recommend to set the working directory of R to the one where you have the model input set. This can be done by the command:

```
setwd("C:/biome-bgcmuso/sensitivity/")
```

(Note that in R the Windows-style path “C:\something\in” should be written like this: “C:\\something\\in” or “C:/something/in”.)

Alternatively, this is set automatically if you start R from the model directory (this is typical under Linux).

Note that the prerequisite of the Monte-Carlo and sensitivity run is the construction of one operating BBGCMuSo simulation than can provide meaningful results!

The typical method to run the model (without RBBGCMuso) is to issue this command within the terminal (Linux), or within the DOS-like Command Prompt under Windows:

muso.exe s.ini

In case of the example provided above, we propose to run the model first using RBBGCMuso to check if there are any issues:

```
runMuso(skipSpinup = FALSE)
```

The plotMuSo command can be used to see the results of the test run:

```
plotMuso()
```

We propose to use the following filename convention. Please name your spinup INI file as something_s.ini, and the normal INI file as something_n.ini where something is arbitrary (note the _s and _n convention). It is not obligatory, but if you do not follow this convention then you have to generate the settings variable manually with the setupMuso command (part of RBBGCMuso). However, if you do follow this convention, then RBBGCMuso will automatically recognize your spinup and normal INI file name and content, so the work is much easier. (See help on this with the ?setupMuso command in R.) The existence of the directory containing the model executable and the complete set of input data is obligatory.

Before starting the sensitivity analysis you have to specify the ecophysiological parameters (stored in the EPC file of the model) you are interested in. This list has to be stored in an external file. You should specify the name, the index (in other words, the physical line number of the EPC file), and the range defined by its minimum and maximum values of the parameter. The header names are: NAME, INDEX, MIN, MAX (all uppercase). Note that NAME can be anything that might help you to identify the parameter. We suggest to use the csv format for these parameter tables, and we also strongly recommend to use comma as separator of the content, and not semicolon. The csv file can be created using e.g. Notepad or EditPad Lite under Windows, or any other text editor. The content within the text editor should look like this:

```
NAME,INDEX,MIN,MAX
CN_leaf,26,14.3,58.8
FLNR,56,0.1,0.4
MSTOMACOND,58,0.001,0.015
ROOTDEPTH,64,0.5,3
SENESCENCABG,101,0,0.2
PHENOPHLENGT3,127.62,50,600
PHENOPHLENGT4,127.63,100,600
PHENOPHLENGT5,127.64,10,200
PHENOPHLENGT6,127.65,400,1000
```

Note that parameter selection is not straightforward in case of multiple columns (see the end of the EPC file). The logic is that fractional part of a number is used to select the appropriate parameter from multiple columns. For example “emergence,127.62,0,1000” means that in the 127th line of the EPC file there are 7 columns (numbering starts from 0, so it is set to 6), and we would like to adjust the 3nd column (marked by 2), which ends up with 127.62.

After you have saved the parameter table to a file (in this example we store the content of the above table in a file called "parameters.csv"), you have to read it into a suitable variable within R using the command: parameters <- read.csv("/path/to/your/parameters.csv"). Note that here we assumed that parameters.csv is located in the same directory as the model executable, so in our example it is as follows:

```
parameters <- read.csv("parameters.csv")
```

Running the sensitivity analysis 1 (detailed description)

Sensitivity analysis uses the musoMonte command that is part of RBBGCMuso. musoMonte is a very flexible tool but its execution needs a couple of settings. Below we provide details about those settings noting that we will mostly use its default settings. It means that the execution is simple (see below). In any case, please go through the text below.

In this example we will run RBBGCMuso sensitivity package (that is based on the monteCarlo tool) with 1000 iterations, so musoMonte parameter should be set to iterations = 1000. The model directory will be the current working directory, so inputDir = "./". Within this directory we need another directory. This is the directory where we would like to store all outputs, and where the input files and model executable are stored. If this directory does not exist, musoMonte will create is. Now let us set outLoc = "./calib", which means that all output data will be stored in C:\biome-bgcmuso\sensitivity\calib\.

Assume that we want to call every output file as "mont-n.csv", where n is the current number of iterations, so we set it using this command: pretag= "mont-".

Assume that we want to create csv files for the output, so we set outputType = "moreCsv". Assume that we would like to use the 8th variable of the daily outputs (this is set by the User in the DAILY_OUTPUT section of the INI file), so we set varIndex = 8. Then use the "mean" function of R as the one which makes the post-processing of the daily output, so we set fun=mean. This step is needed in the sensitivity analysis as the model output variabilty will be represented by a single scalar that is based on the daily results of the model. We can e.g. use maximum or mean LAI here during the entire simulation. Now we would like to ask musoMonte to automatically generate settings variable, so we set setting = NULL. After all, this is the complete command line for musoMonte (prerequisite of the sensitivity run):

musoMonte(settings = NULL, parameters = parameters, inputDir = "./", outLoc = "./calib", iterations = 1000, preTag = "mont-", outputType = "moreCsv", fun=mean, varIndex = 8)

It might look quite complex, but this can be really easy if we do some preparations. For example, we should set the working directory to inputDir, but we can ignore the specification of outLoc which means that the above (default) setting will be used. You only need to specify those variables which are different from the default (the above list contains the default values except the varIndex, which is 1 by default). So the following line is a simple way to initiate the Monte Carlo run:

```
musoMonte(parameters=parameters, iterations = 1000, varIndex = 2)
```

Once we are done with the Monte Carlo simulation (which can be rather time consuming), initiation of the sensitivity run is very simple. An example line can be seen below, where we use the previously created "preservedEpc.csv" file which was created during the Monte Carlo run (by default).

```
musoSensi(parameters = parameters, monteCarloFile="./preservedEpc.csv")
```

If everything went well, we will get the results of the sensitivity analysis in a file called "sensitivity.csv". This file will contain the relative importance of the predefined parameters in terms of importance that is associated with the variability of the selected output (in this example this output variable was the 8th daily output variable).

You may visualize the results using e.g. MS Excel. Note that the results are automatically visualized in a newly created file called sensitivity.png.

Running the sensitivity analysis 2 (simple solution)

There is a very simple way to initiate the sensitivity analysis using RBBGCMuso which uses the default settings for all variables described above. For everyday application this method is the simplest which does not need decisions from the user (except the number of model runs and selection of output variable).

If the parameters.csv file is ready to use and is located next to the model executable (note that within this context the filename is fixed and must be parameters.csv), the sensitivity run is initated by simply issuing this command:

```
musoSensi(iterations = 10000, varIndex = 2)
```

This example executes 10 000 Monte Carlo simulations, and uses the 2nd variable of the daily outputs (this is set by the User in the DAILY_OUTPUT section of the INI file) as the target output variable for the sensitivity analysis. By default, the non-informative simulations (where all daily output is zero or the model fails to run) are excluded from the analysis. The result will be stored in the sensitivity.csv. Additionally, the results are automatically plotted in the sensitivity.png file, and also on the screen in RStudio.

Please let us know if you encounter any problems!

Roland HOLLÓS: hollorol@gmail.com Zoltán BARCZA: zoltan.barcza@ttk.elte.hu