Description of the Computer Program of the Monodominance Model

Pia Backmann

March 9, 2016

1 Overview over provided documents

In the **Monodominance_Modell** folder, you find the following sub-folders:

- **The main program**
  is situated in the root folder. For more information, see section "Implementation".
- **Data**
  Provides the structure in which the data produced by the main program is stored. Please do not remove this folder.
- **Example_DATA**
  Contains an example dataset needed to test the R-scripts.
- **R_scripts**
  Contains the most essential scripts to analyse the produced data.
- **Pictures**
  Here the produced pictures are stored.
- **README**
  Contains this description and the Model description protocol ODD (for further information, see: Grimm, V., Berger, U., Bastiansen, F., Eliassen, S., Ginot, V., Giske, J. et al. (2006). A standard protocol for describing individual-based and agent-based models. Ecological Modelling, 198, 115-126. Grimm, V., Berger, U., DeAngelis, D.L., Polhill, J.G., Giske, J., Railsback, S.F. (2010). The ODD protocol A review and first update. Ecological Modelling, 221, 2760-2768).

2 Implementation

The monodominance model has been implemented in C(++) using the following libraries:

- The standard library (stdlib.h)
- The standard input-output library (stdio.h)

- The math-library (math.h)
- The string library (string.h).

and the standard Gnu C++ compiler (under Linux Ubuntu).
The model and as well many important algorithms (e.g. the cluster detecting algorithm or the calculation of the radius of gyration) have been implemented from scratch.

The Monodominance program is divided into the following source files:

- **mainwindow.h**
  contains all the function headers and structs

- **mono_basic.cpp**
  Contains all the basic functions for simple calculations

- **mono_cluster.cpp**
  contains inter alia the cluster algorithm counting the number of monodominant clusters and calculates the size of the biggest cluster

- **mono_inits.cpp**
  contains the initializations of all structs and of the world grid

- **mono_kills.cpp**
  contains the kills of all structs to free the computer memory after a struct is used

- **mono_main.cpp**
  the main routine of the Monodominance program

- **mono_routinen.cpp**
  contains more complex and specific calculations

3 Parametrisation

Here a complete list of the parameters that can be modified when running the program:

Parameter name	Description	Default value	
–mort_rate	Annual mortality rate of all tree species	1.5% per year	
–seed	for random number generator	1	
–init_option	which initialization scheme to chose: 1: random, 2: grid, 3: knight's move, -1: streets	1 (random)	
–density_radius		5	
		identical species	**monodom. species**
–num_species	the number of all species (*num_species - 1* idential plus one candidate for Monodominance)	7	1
–radius	dispersal range of the candidate monodom. species	20	1
–seed_mass	seed mass per grid cell of the candidate monodom. species	20	9450

Table 1: The range and default values of species attributes and simulation parameters of the monodominance model.

4 Run the program under Linux

Before compiling, please change the path given in the file to the location where you stored the Monodominance Program.

Open the makefile **Makefile** and change the BASE_DIR line into you own path (see Fig. 1).

Figure 1: Where to change the path in the Makefile.

To compile the Monodominance program, open a console (shell) and go into the "Monodominance_Modell" folder and type:

make

This runs the Makefile which then builds the Monodominance software from all its source files (the "-.cpp" - files in the folder).

To run the compiled program, please type:

./Monodominance

into the console. Like this, the program is run with default parametrisation. The parameters can be addressed, by adding the specific options:

./Monodominance –size 512 – num_species 8 –mort_rate 0.015 –seed_mass 9450 –seed 1 –radius 20

We as well provided an exemplar of a shell script Mono_shell.sh which runs the simulation automatically for different parameter combinations and - to obtain more statistical relevance - iterates over 1000 seeds. All variable settings can be changed here. The shell script can be made executable by typing into the shell:

chmod u+x Mono_shell.sh

and it can be run by typing:

./Mono_shell.sh

5 Run the program under Windows

Please download a C++ compiler like Borland C++, Visual C++, or freeware like GCC (`http://gcc.gnu.org/`), Visual Studio Express (`http://www.visualstudio.com/`) or MinGW (`http://www.mingw.org/` and `http://mingw-w64.org/`) and follow the instructions for installation and compilation of the code.

6 Processing the output files

The output of the program is written into the Files of the delivered Monodominance_Modell/Data folder. Use the R-scripts provided in the folder Monodominance_Modell/R-scripts to obtain figures showing the current results.

The following R-scripts are necessary:

- **Multiple_Weltplot_FINAL.R**
  This script produces a multipicture of a sequence of plots of the simulated forest in steps of 500 simulation years. Important remark: it works only in combination with the **"Picture_world.R"** function - so this file has to stay in the same folder.
- **Abundance_boxplot_mortality.R**
  It produces a boxplot of the relative abundance of the monodominant species for different seed masses and mortalities.
- **Mortality.R**
  Plots the mean of relative size of the biggest monodominant cluster against the annual background mortality.
- **Binder_Transformation_BiggestCluster.R**
  Calculates the mean of the size of the biggest cluster (for each value of the seed mass SM). Plot the mean biggest cluster size against the seed masss (main plot) and the binder cumulant of the mean biggest cluster size (subplot), both against the seed mass (SM) for different system sizes.
- **Cluster_Bild_FINAL.R**
  Plots all clusters of the monodominant species (each cluster is to another color which is randomly chosen. Plots a sequence of plots at different time steps. To ensure that for two consecutive pictures the same color codes are used for the same clusters, a special algorithm is used (to at least detect the four biggest clusters).
  !!! Attention: Produces a large number of pictures !!!
- **Korrelation_length.R**
  Plots the correlation length (radius of gyration of the biggest cluster) against the seed mass for different system sizes.
- **Percolation_probability.R**
  Plots the percolation probability against the seed mass for different system sizes.
- **Init_rel_abundance.R**
  Creates a boxplot of the relative abundance of the monodominance species at the end of the simulations for 4 different initialization schemes.

- **Statistics_init.R**
  Plots the relative abundances for the four initialization schemes against the seedmasses chosen (line plot).

- **Local_density_map.R**
  Creates a map of the local

To test, whether the scripts are running correctly on your computer (after you changed all pathways to your current path), one dataset which can be used to test the R-scripts is provided in the folder Monodominance_Modell/Data/Example_DATA. If you want to check the results obtained by your own runs, please change the folder in the R-scripts.

The parameters, which have to be changed in the R scripts are adapted to this example-dataset (and need to be chaged manually, when you are running different simulations).