University of Oxford & Institute of Zoology, CAS

EcoEvo Simulator: EES

Overview document

Erin E. Saupe & Huijie Qiao
2018-11-27

1. Generate the configuration files (in JSON format) and run commands for the simulations.

   The script was written in Java. All scripts in Java can be downloaded via a subversion repository (svn://mmweb.animal.net.cn/NicheBreadth/trunk). The function for step No.1 can be found in the following location.

```
package org.ioz.niche.breadth.r2016_r12
class GenerateJson
public void generateSpeciesJson()
public void generateScenarioJson()
public void generateCommand()
```

Json format of the configuration of virtual species:

```
{
    "id" : 1,
    //should be the same length as the environmental layers defined in the
scenario configuration file
    "niche_breadth" : [[v1_min, v1_max], [v2_min, v2_max], […], [vn_min,
vn_max]],
    //
    "dispersal_ability" : [0.3,0.7,0.9,0,1.0],
    "dispersal_speed" : 100,
    "dispersal_method" : 2,
    "number_of_path" : -1,
    "speciation_years" : 10000,
    "species_extinction_threshold" : 0,
    "species_extinction_time_steps" : 1,
    "species_extinction_threshold_percentage" : 0.8,
    "group_extinction_threshold" : 0,
    "initial_seeds" : [[124.5, 54.5]]
}
```

Json format of the configuration of virtual scenario:

```
{
    "environments" : ["path_to_env_layer_1", "path_to_env_layer_2", "…",
"path_to_env_layer_n"],
    "total_years" : 120000,
    "mask" : "path_to_mask_layer",
    //connect the species configuration via the “species_id”.
    "species" : ["species_id"],
    "burn_in_year" : 0
}
```

Command to launch the simulation application:

> ./NB_Icesheet base_folder scenario_configuration Results_folder 800 0 0

2. Launch simulations by running the commands generated in step No.1.

   The simulation application was written in C++. The source code is open access and can be used under the GNU Public License. The latest source code can be downloaded via svn://mmweb.animal.net.cn/nb_icesheet/trunk. And documents regarding the code can be found via http://ecosimulation.sourceforge.io. More details about the code is provided in its documentation.

The simulation results were saved in a batch of log files in plain text, GIS raster (GTIFF) or NEXUS tree formats.

The output directory structure generated by the simulation application:

```
├──── Map_Folder //a folder to save the distribution year by year
│    ├──── 100.csv
│    ├──── 200.csv
│    ├──── xxx.csv
│    └──── 120000.csv
├──── Phylogram.html //An HTML file to show the speciation tree in browser
├──── runtime.log    //A log file to save the runtime information.
├──── seeds.csv    //Seed point of the simulation
├──── stats
│    └──── stat.csv    //The number of speciation and extinction events in the simulation
├─── suitable.tif    //suitable area (potential distribution) of the virtual species in the simulation.
└──── tree.new      // the speciation tree in NEXUS format.
```

3. Analyze the log files, extract the useful information, and import data to database.

The scripts were written in Java. All functions are in the following package and classes.

```
package org.ioz.niche.breadth.r2016_r12
class HandleResult
class MeanLat_By_Continent_NB
class MeanLat_By_Continent
class MoreMetrics
class Speciation_Convex_Stat
class Speciation_Extinction_Lat
class overAllFunctions
```

Class	Function	Database Table
MoreMetrics	allMetricsNoNicheBreadth()	Biome
MoreMetrics	allMetricsNoNicheBreadth()	Continent
MoreMetrics	Continent2 ()	Continent2
HandleResult	generateConvexMaps()	Convex
HandleResult	generateConvexMaps()	Convex_with_year
MoreMetrics	allMetricsNoNicheBreadth()	Distribution
HandleResult	generateOriginMaps()	Diversity
HandleResult	generateOriginMaps()	Extinction
HandleResult	generateExtinctionTable()	Extinction_with_year
HandleResult	detectSpeciationExtinctionTemp2Prec_Origin()	Extinction_in_combined_environments
MoreMetrics	allMetricsNoNicheBreadth()	Lat
MoreMetrics	allMetricsNoNicheBreadth()	Lon
MoreMetrics	nicheBreadth()	Niche_breadth

HandleResult	generateOriginMaps()	Origin
Speciation_Convex_Stat	calculate_convex_speciation()	Speciation_Convex
Speciation_Convex_Stat	calculate_convex_speciation()	Speciation_Convex_E
Speciation_Extinction_Lat	speciation_extinction_by_others()	Speciation_Extinction_BIOME
Speciation_Extinction_Lat	speciation_extinction_by_others()	Speciation_Extinction_Continent
Speciation_Extinction_Lat	speciation_extinction_by_others()	Speciation_Extinction_Continent2
Speciation_Extinction_Lat	speciation_extinction_by_lat()	Speciation_Extinction_Lat
Speciation_Extinction_Lat	speciation_extinction_by_others()	Speciation_Extinction_TT
HandleResult	detectSpeciationExtinctionTemp2Prec_XXX()	Speciation_extinction_in_combined_environments_XXX
HandleResult	insertResult_removeUnfinished()	Simulations
MoreMetrics	allMetricsNoNicheBreadth()	Temperate_Tropic
overAllFunctions	insertTree	Tree

4. Extract the required data from the database, analyze the data tables, and create the figures for the manuscripts.

To install the application on Ubuntu

1. Download and install Ubuntu
http://releases.ubuntu.com/14.04/

2. Download Eclipse CPP
http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/helios/SR2/eclipse-cpp-helios-SR2-linux-gtk-x86_64.tar.gz

3. Install JDK in Ubuntu
https://www.digitalocean.com/community/tutorials/how-to-install-java-on-ubuntu-with-apt-get

4. Install g++ v4.8 in Ubuntu
http://askubuntu.com/questions/271388/how-to-install-gcc-4-8

```
sudo apt-get install python-software-properties
sudo add-apt-repository ppa:ubuntu-toolchain-r/test
sudo apt-get update
sudo apt-get install gcc-4.8
sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-4.8 50

sudo apt-get install g++-4.8
sudo update-alternatives --install /usr/bin/g++ g++ /usr/bin/g++-4.8 50
```

5. Install GDAL in Ubuntu
http://www.sarasafavi.com/installing-gdalogr-on-ubuntu.html
sudo add-apt-repository ppa:ubuntugis/ppa && sudo apt-get update
sudo apt-get install libgdal1-dev libgdal-dev libgeos-c1 libspatialite3 libgdal1h libgdal1-dev libproj-dev gdal-bin
make sure that you have /usr/include/gdal/gdal.h

6. Install Subversion in Eclipse
https://eclipse.org/subversive/documentation/gettingStarted/aboutSubversive/install.php

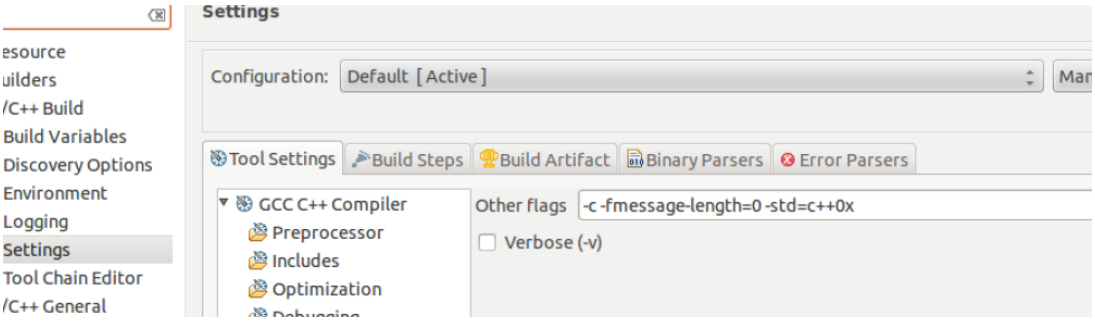
7. Checkout NB simulation code with Subversion in Eclipse via
svn://mmweb.animal.net.cn/nb_icesheet/trunk

8. Add /usr/include/gdal/ to your Include path in Eclipse
Properties -> C/C++ General -> Paths and Symbols -> Include -> GNU C++

9. Install Boost v1.5 in Ubuntu

```
sudo add-apt-repository ppa:boost-latest/ppa
sudo apt-get update
sudo apt-get install libboost1.54-all-dev
```

10. Add -std=c++0x to Eclipse g++ Compiler



11. Compile your application in Eclipse

12. Create a folder ‘NB_Test’ for the application and copy ‘M1000’ to the folder in your virtual machine

13. Copy ‘scenario.seed_1.dispersal_good.nb_large.json’ to ‘Scenario_Configurations’ folder in ‘NB_Test’

14. Copy ‘seed_1.dispersal_good.nb_large.json’ to ‘Species_Configurations’ folder in ‘NB_Test’ in your virtual machine

15. Edit ‘scenario.seed_1.dispersal_good.nb_large.json’ to match the path of M100 folder.

16. Edit src/main.cpp in Eclipse, lines 60 and 61, point the files to ‘M1000/mask.tif’ (dirty code, needs to be modified), and recompile it

17. Create a folder ‘Results’ to save the result.

18. Run the application with the following parameters.
‘root’ ‘json’ ‘result_folder’ ‘max memory usage’ ‘is_overwrite (0:no, 1: yes)’ ‘unused, always 0’

Example:
/home/eesaupe/nb_test scenario.seed_1.dispersal_good.nb_large /home/eesaupe /nb_test/result 8000 1 0