**MANUAL TO CODE OF THE PAPER “EFFECT OF THE PLASTICITY ON ADAPTIVE EVOLUTION IN DIFFERENT CONTEXTS - MECHANISM”**

**INTRODUCTION**

In this paper, we used a package called “gen3sis”. The focus this package is give a background in code to run simulations in macroecology and macroevolution. How it was done in code way, the person who use package can alter it as him/her want. This package was published in PLOS One Journal and authors tested some hypothesis to validate package as good tool to studies in macroecology and macroevolution. They validated it testing premises of the latitudinal gradient biodiversity and results emerged were consistent.

In our work, we used git as control version. So, we had 2 branches: (1) main – to run definitive simulations; and (2) variable along time – to run simulations focused in discovery time necessary to capture well the pattern without lost any relevant process.

Bellow, we presented explanations about code that can help you to understand what it was done and what was the reason from behind the chosen that were done. Repeated code between branches only were explained to first branch (variable along time). Underlined text below refers to mentions to code parts that we modify directly.

**BRANCH VARIABLE ALONG TIME – TEMPORAL SIMULATIONS:**

1. **Packages**

If first part of the code, there was a commentary with general information about goal of our work and that code. After, we carried packages (if you have not any package, you need install with command “install.packages”). “gen3sis”: package with basic functions to deal with simulation configuration and landscape. “here”: package used to call automatically our current directory - it makes more easy carry our data and code in other computers (science reproducibility).

1. **General explanations to all code:**

To run simulation with gen3sis was needed two inputs: landscape and configuration. Landscape was a file with “.rds” format that contains information about landscape in list format in R – in this list had raster information per timestep with spatial cost information and distance local information that it was created per timestep information when you create a landscape file (distance local is located in separate folder). Configuration was an input with general information of the simulation: number of species and individuals, begin and end time of simulation, dispersion, ecology and evolution information, and others. In gen3sis with these two inputs you can use the function “run\_simulation()” specifying landscape, configuration, and other general options to run simulation. When you run simulation, the R run your simulations and generate map and general result information of your simulation (living species, speciation, extinction, etc.). Still, it was generate a list called sgen3sis with general information of your simulation (with phylogenetic, and some specificities). With gen3sis you can change landscape and configuration and there are a lot of functions implemented to save plot, trait, and other things. But, if you want change gen3sis as you want, you need press F2 on top of the “run\_simulation” to see intern code and identify what run simulation do when run your code. So, it is need work intern code and learn a little bit about functions created by authors of the gen3sis paper and package. So, in our work, we work with internal code of the gen3sis within “run\_simulation”. Beyond this, we create our proper landscape with more timestep to run our simulations and test temporal sensibility – to it we used function of the gen3sis to create the “landscape.rds” and “distance place”. We did some modifications in the configuration of the our simulation, but these modifications, different of the landscape, appear in intern code of the “run\_simulation”.

After these general explanations and carry all packages, we define out data path and define the path of the configuration and landscape. We specified general information of the run simulation, as output directory, save state, etc. These information we did not modify, it was as went created. After, directory was created and landscape/configuration was carried. Next code, it was a code to verify if there is or not config input. In the next step, we did modifications in config: to decide specific configuration, previously, we ran test simulation to verify as it behaves along time. We observe that the more species live in system later is simulation. 5000 species more, with our modifications in code that we are going to specify more below, system late much time to pass from one timestep to another and big timestep with 10 values of plasticity surely simulation late much time. Other thing observed it was that the more time pass, more species (until dead) is accumulated in memory ran, and late more (even more if there are much species alive). Based in this observation, we decide run with 300 timesteps (300 timesteps in past to 1 timestep) and we define 5000 species alive as maximum species in our system (if there were more species than it, simulation stop and pass of the one level of plasticity to another – if it was last simulation, it was finished). Start time were 300 timesteps and end 1. We defined 5000 species as maximum species that our system support. And created a variable called “timestep\_total” that was used to define length of our dataframe where our results would go. We too change elements that we wanted save – it was excluded all plots and we only saved traits. This trait was used to calculate trait evolution posteriorly. After, we created some variables called “rep”, “plasti”, “pos”, and “pos2”. Rep was replication value that we use in loop for replicate simulations. Plasti was sequence of plasticity values used in loop (we defined from 0.1 to 1 of the plasticity). Pos was used to mark position in dataframe, each timestep pos receive + 1, that is, the result was save each time simulation and we used this number to define position which this result would be save. Pos2 is the same thing of the Pos, the only difference is Pos2 restart if we did other replicate (Pos did not restart, the value final quantity of the timesteps of the simulation). Pos2 also served to count time simulation in crescent order (different of the timestep that it was in decrescent order). After it, we create a dataframe (table) where results would be save. To define the right size of the dataframe we multiply number of the replications plus plasticity levels plus timesteps – this table present results and some information: value of plasticity, replication, extinction, diversification, trait evolution, timestep, and time simulation. In the next step, we remove all previous trait data saved. It is relevant because if simulations with more timestep were done previously, same if save traits replace ancient values, replace only work in correspondent timestep (it did not work to replace bigger timestep – so our calculation of the trait evolution would be wrong). Doing this, still remove any chance of there are traits that we did not want and any problem with replacement. To do it we list all files in traits folder with paths, and use file.remove to remove files.

Next it, we did added two loops: one for plasticity level and another to number of the replications. In loop of the plasticity we modify the config to each time that loop finish, change plasticity value putting values previously cited and defined by variable plasti. The modify that we had done in ecology section of configuration is: value of trait varies to other values around value original, all this values were subtracted by value of environment temperature, and after we select the minor difference between trait and environment. Following logic of surviving species of the original model, as minor difference more species surviving. So, plasticity increases range of the trait value of the species, and it did species had more chance of surviving. After it, loop of replication begin to one plasticity value (0.1 to 1 – crescent order).

In begin of the replication code, we repeat original code of gen3sis. In this first step, it was created one list which all data of the config, environment, and modifications in code would be storage (called “val”). All input were of the landscape, config was done. Some functions created by authors of the gen3sis create species of the world using configuration that you chose. After create all inputs and statistics of simulation, other loop begin loop of the passage of time (“timesteps”). Time number of loop correspond to timestep define by you. Each time steps speciation, extinction, phylogeny, and other things were updated by things that happen in the world. Great part of this part follow original code. Speciation, dispersal, extinction, and others happen based configuration pre define. Number of species and individuals living were updated. Phylogeny was updated. In end of the loop of the each timestep, time of the simulation was save. All results and outputs were saved in an object called “sgen3sis”. We deactivate some parts of the code that didn’t need to me and serve to optimize it (plot and file of the gen3sis summary as example).

Next step, still in the same timesteps loop, we add some code to get output that we want (speciation, extinction, and trait evolution). First, we begin with trait evolution: all trait values get out of the each timestep of the simulation as “.rds” file in folder called “traits”. In each timestep we list all files of this folder with some commands and, in sorted way, we create one list with all information of the traits to all species in each timestep. We filtered data to select only temperature values of the individuals. After, to each species, we calculate the mean. We removed all NA occurrence. Each species had a mean value of the trait temperature of the individuals per timesteps. How was it calculated trait evolution value? Trait evolution was calculated take mean values of each species in specific timesteps and comparing with same species of the timestep posterior (value of the trait in previous timestep is subtracted by posterior timestep to each species). After, the posterior timestep become previous and compared with the next. The difference in value of the trait per species in each timestep represent the trait evolving along time. In each timesteps all traits that already run is read and we took a mean of the all differences in all species trait along of all timesteps until that moment. We took this difference and divided by number of timesteps already run. This form, we took trait evolution calculation. In code part, we created one loop only to discovery if the simulation was in first timestep. If the simulation was in first simulation trait evolution was not calculated because we did not have any posterior timesteps to calculate the difference between trait values.

**BRANCH MAIN – DEFINITIVE SIMULATIONS:**