1. Connection between Metronamica and EMA workbench

This section guides you through the ***Run_cmd_2050 (1).ipynb.***

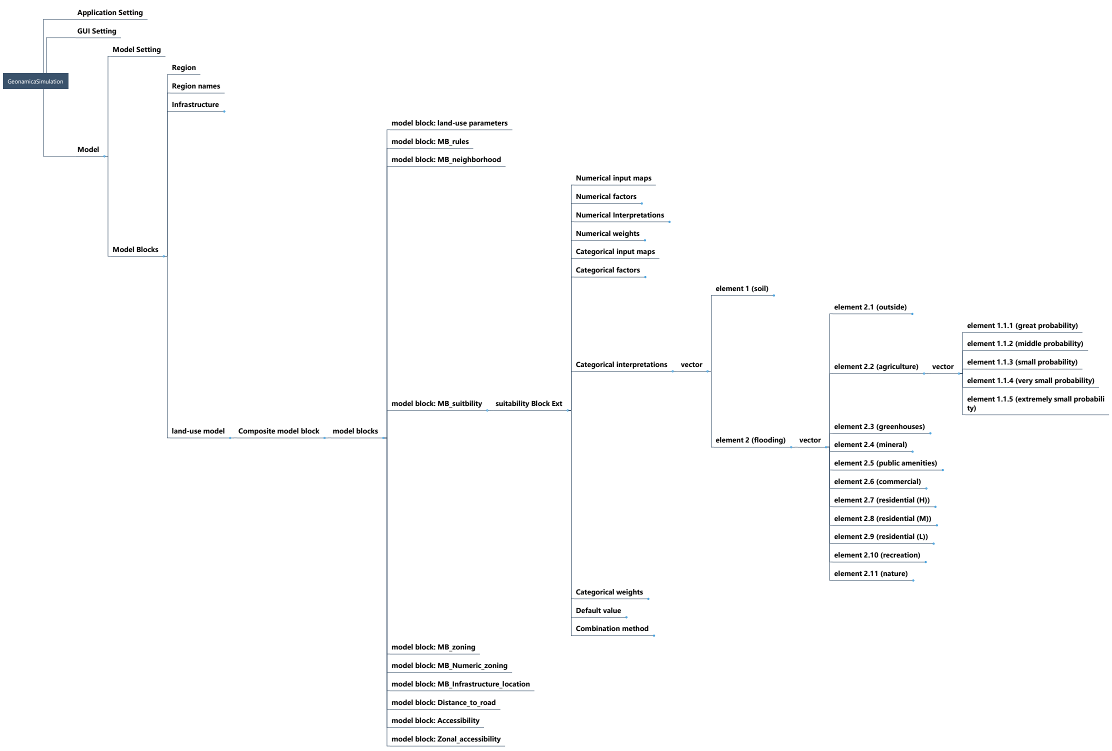
EMA workbench enables the multiple simulations over the Metronamica model. In each run, EMA first needs to access the uncertain parameters and sample over their assigned uncertain space. Then, EMA commands the Metronamica model to execute and store the outputs in the assigned space.

1. How to change the values of the uncertain parameters?

Open the ***.project*** file of the Metronamica model, you'll find the model information are coded in XML language, just like a "tree" structure (see the figure below). In python, you can use the functions from xml.etree.ElementTree to access and edit the value of the model parameters.

You will need to figure out the locations where the parameters you want are and clarify them in the codes. Once they are visited by the codes, you can write some functions to assign these parameters new values.

For more detail, please see the functions *change_suitability* and *edit_demand* in the code file ***run_cmd_2050(1).ipynb***



2. How to execute the Metronamica simulation by python?

You can use the functions (*Metronamica_python_interface, run_model, _generate_options)* directly from ***run_cmd_2050(1).ipynb*** to directly run your Metronamica model.

_generate_options: This function tells the model where to store your logged land-use maps

Run model: This function commands the model to run. In this function, please need to clarify the path where you install Metronamica in your laptop.

Metronamica_python_interface: This is an integrated function of all the functions mentioned above, which means if you want to run your Metronamica model, you just need to run this function. Hence, you need to add all your defined functions here (such as *change_suitability* and *edit_demand*). And you should also add all the new parameter values to the inputs. You have to adapt this function to your new case.

There is one cell in my codes to test whether python could execute the Metronamica simulation.

3. How can the EMA workbench be linked to our codes?

Now our codes can already execute the Metronamica simulation and all we need is to add EMA workbench in to help us sample over the defined parameter space and conduct numerous experiments.

First, we need to tell EMA workbench which model it should work with. The model input to EMA workbench (*geonamica_dtag_ema*) is adapted from the *Metronamica_python_interface,* but instead of giving the suitability parameters fixed values, they are deemed as uncertain and will be assigned with uncertain ranges in the next step.

Next, what you will do is the normal step in EMA workbench – to specify the uncertain space for the uncertain parameters and the policy levers (if any). You could first try a few runs to see if the experiments would work. After that you could move to larger experiments.

Note: when I run the experiments, I did not use the multiprocessing, because it requires some modifications on my laptop settings. I tried this but failed, maybe you can have a try on your laptop and if you make it, it saves your experiment time!

I also run a part of my experiments on another computer in parallel. If you want to conduct your experiments in this way, you will need to have Metronamica installed in that PC as well and adapt the codes a bit (such as the working directory and the file locations, etc).

4. How to process the experiment outputs?

The outputs of the results are stored in array form (in my case a 584 * 757 matrix) and each number of the array indicate the corresponding land-use types (just the same as the order you input to Metronamica). Since our next step is to cluster and analyse these maps, we need to evaluate how similar these maps are. The “distance matrix” I adopt is the Kappa matrix.

Therefore, I conduct the Kappa calculation between each pair of the 2000 results. "Kappa_1.py" file contains the codes for calculating the kappa index between the maps and you can just use it directly. This kappa code requires you to import a mask map, and please make sure your mask map is in the same row and columns as your output maps. In my case, the kappa calculation leads me to a 2000*2000 matrix and each value in this matrix indicates the kappa index (i,e, the index to measure similarity) between the two maps.

2. Cluster the resulting maps

This section explains about ***Cluster.ipynb*** and ***Run Cluster Exploration.ipynb.***

1. Clustering algorithms

We need to cluster the maps after the experiments are done, as it is difficult for us to analyse the 2000 maps one by one. We hope to cluster them based on their similarity and clustering algorithms will help us to cluster the maps. But which clustering algorithms are suitable in our case?

Hence, we will do some tests over some different clustering algorithms and this is what ***Cluster.ipynb*** about. The kappa matrix we obtained in the last step will help us to identify the similarity within each cluster and the dissimilarity between each cluster. The functions to calculate similarity and dissimilarity can be used directly by you, but you can always adapt it if you want to choose the median rather than the mean as the representation of all the interested data.

2. Representative maps

Once the algorithms are determined, you need to select some representative maps by using these algorithms. We go with the maps with the highest and the lowest variance. We collect the maps that are clustered in one group together and compare the Kappa index in the Kappa matrix. The one with the lowest Kappa values means it is the least similar to the rest maps, and hence, it is recognized as the highest variance. Also, the one with the highest Kappa value are identified as the lowest variance map. Based on this, we could gather the map IDs for the representative maps.

3. Analysis on the representative maps

Once we know the IDs of the representative maps, we can extract their data and reconstruct the land-use map. The data is stored in the array form and you need to visualize it. ***Run Cluster Exploration.ipynb*** shows the way I visualize and compare these representative maps. But the maps I reconstruct are just in png or jpeg form, which means they cannot be processed by QGIS or ArcGIS. Probably in your further analysis, you will try how to use these arrays in more advanced analysis.