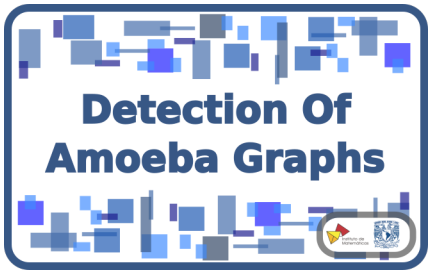
Instruction Manual for the Github Repository
Detection of Amoeba Graphs

@MarcosLaffitte



- Developed By

Lic. Marcos Emmanuel González Laffitte
IMATE - UNAM Campus Juriquilla
marcoslaffitte@gmail.com

and

Dra. Amanda Montejano Cantoral
UMDI - Facultad de Ciencias - UNAM Campus Juriquila
amandamontejano@ciencias.unam.mx

- Publicly available at

`https://github.com/MarcosLaffitte/Amoebas`

- About

Amoebas are a family of simple graphs first defined by Adriana Hansberg, Yair Caro and Amanda Montejano, who initially studied them in the context of Ramsey-Turán Theory **[1, 2]**. The study of these graphs is of interest, in particular, due to its relation with the graph isomorphism problem. All the programs here can be used to detect amoebas and analyze their properties. This repository was developed as part of the work of Marcos E. González Laffitte's Master Thesis in Mathematics under supervision of Dra. Amanda Montejano Cantoral.

- Content

There are 6 relevant folders in this repository:
1_G6_Builder,
2_Detect_Arbitrary_Amoebas,
3_Detect_Tree_Amoebas,
4_Weird_Edge_Replacements,
5_Group_SG_Structure.
6_Examples.

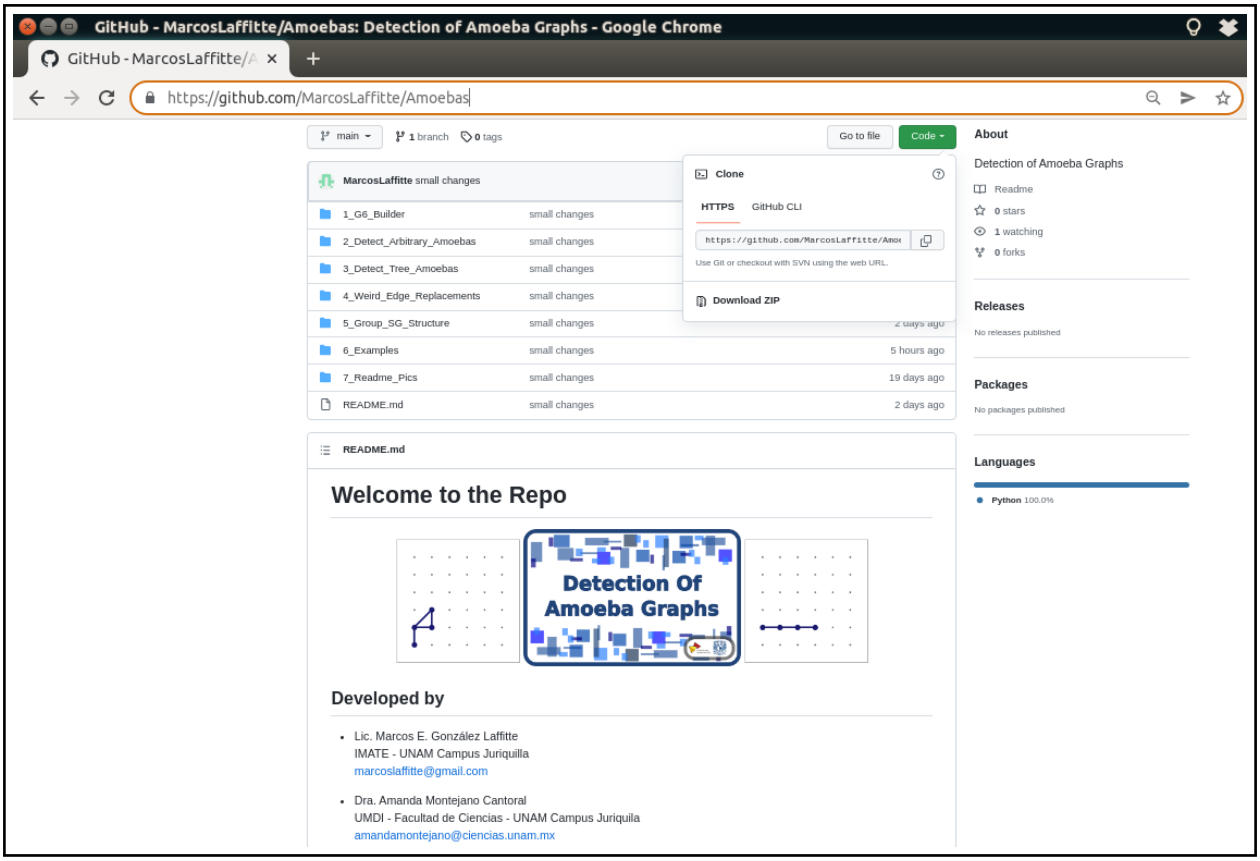
Inside of each of the folders 1 to 5 there are 2 Python scripts, one to develop a specific Analysis, and another for the Visualization of the results obtained from that analysis. Here we will describe how to download the repository and run these programs over sample sets that can be found in the folder 6_Examples. It is important to mention that all the programs here were developed and tested using Ubuntu 20.04 LTS, but haven't been tested in any other OS.

- Python Requirements

In order to run the programs Python 3.7 is needed, as well as Anaconda: 4-bit version - conda 4.10.1. For information on Anaconda see `https://www.anaconda.com/`. After this, we need to install with anaconda the following four packages:
- networkx 2.5
- sympy 1.7.1
- matplotlib 3.3.3
- sage 9.2

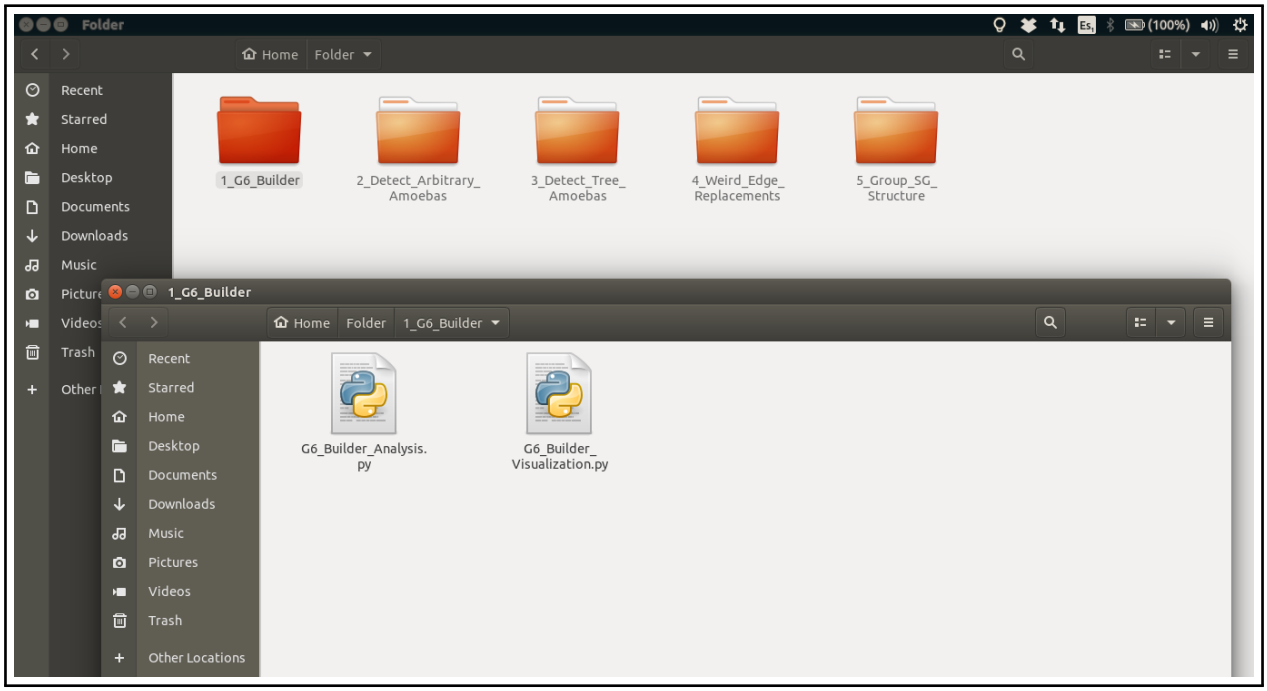
- Downloading the repository

Having accessed to the repository through the link `https://github.com/MarcosLaffitte/Amoebas` we can find all the folders and a brief description of the repository, as shown in the image bellow.



On the top right corner we can find a green button with the text ”Code” on it. We can push this button to display a drop-down menu with various options for downloading the repo. If we don’t have a Github account and only want to download the programs we simply need to select the option ”Download ZIP”. Right after this we will start downloading a single zip file containing all the folders in the repository, which we can later decompress in our own computer/server.

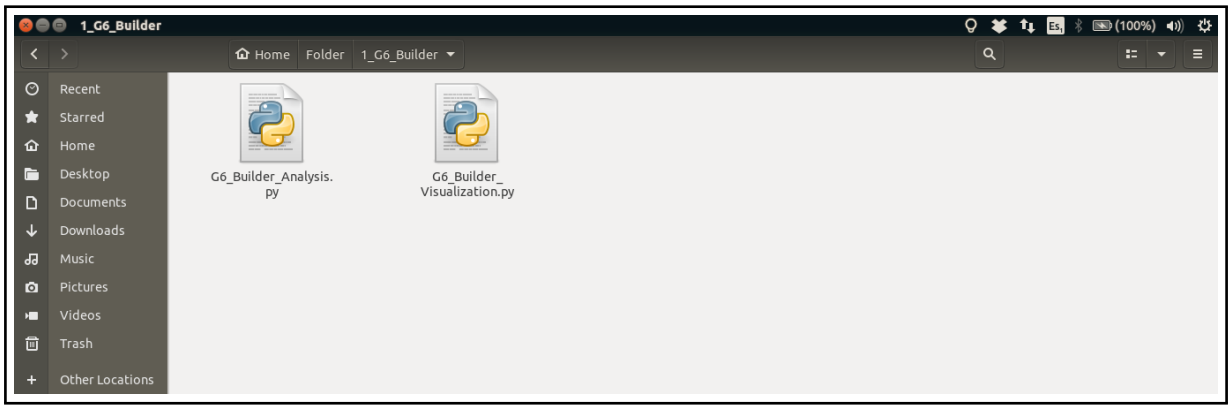
The decompressed zip folder will contain every directory available in the repo at the time of download. For the sake of simplicity in the following image we only show the main folders 1 to 5.



Each folder contains one python script to develop a particular task/analysis, and another python script that produces drawings of the results obtained from said analysis.

- 1_G6_Builder

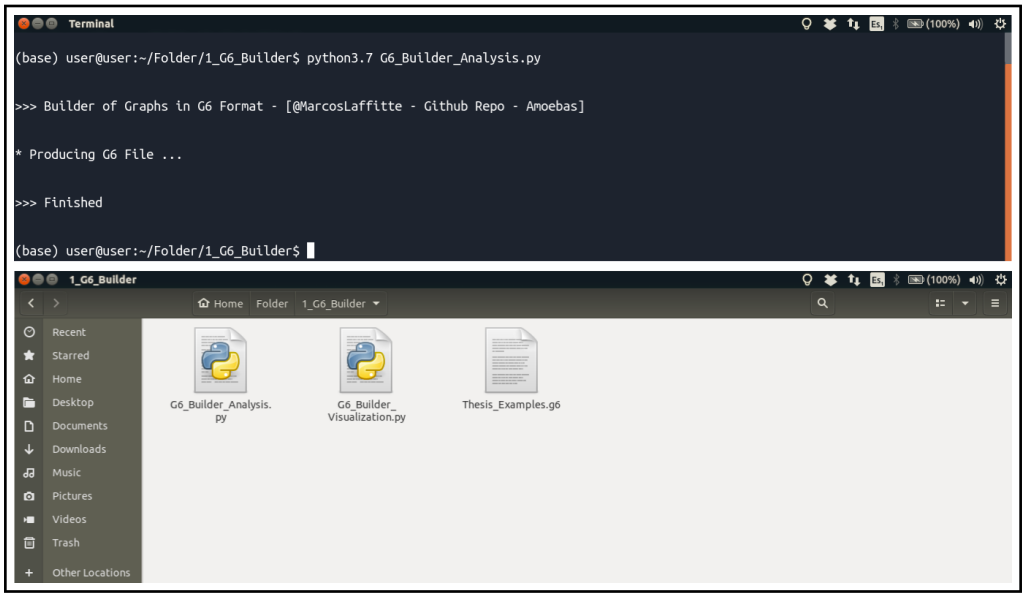
For example, the 1_G6_Builder folder will contain, as show bellow, the script G6_Builder_Analysis.py, which we can use to produce a g6 file containing various simple graphs to be analyzed, and another script G6_Builder_Visualization.py that will allow us to plot these graphs.



By running in the command line the following instruction

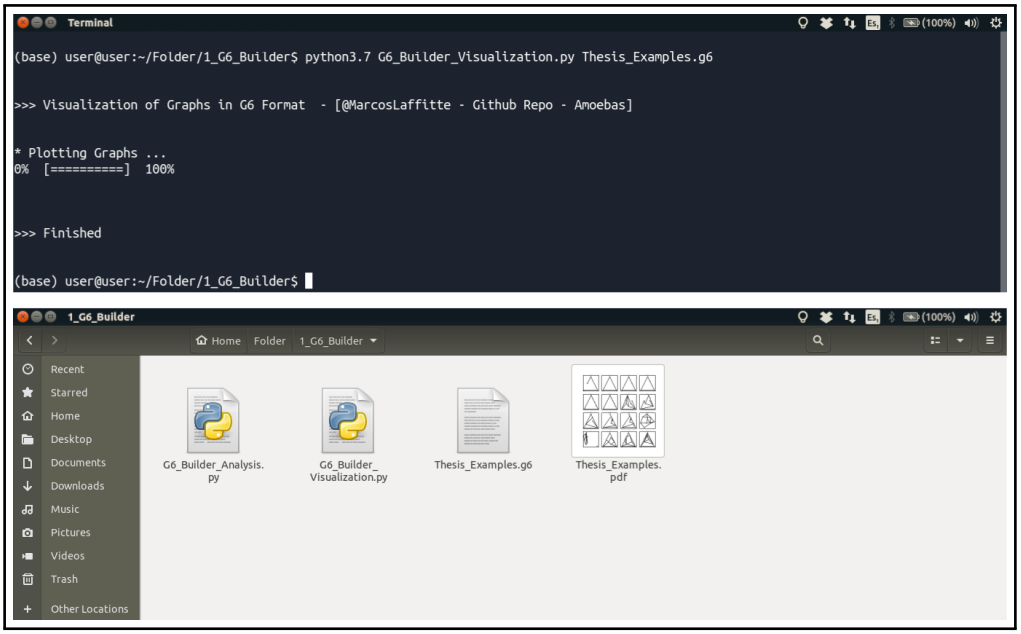
python3.7 G6_Builder_Analysis.py

We should get an output in the command line similar to the one bellow, as well as a new file called Thesis_Examples.g6, containing all of the g6 formats of the graphs included in @MarcosLaffite's thesis.



Then we can immediately run the following instruction to plot all these graphs, creating a new pdf file.

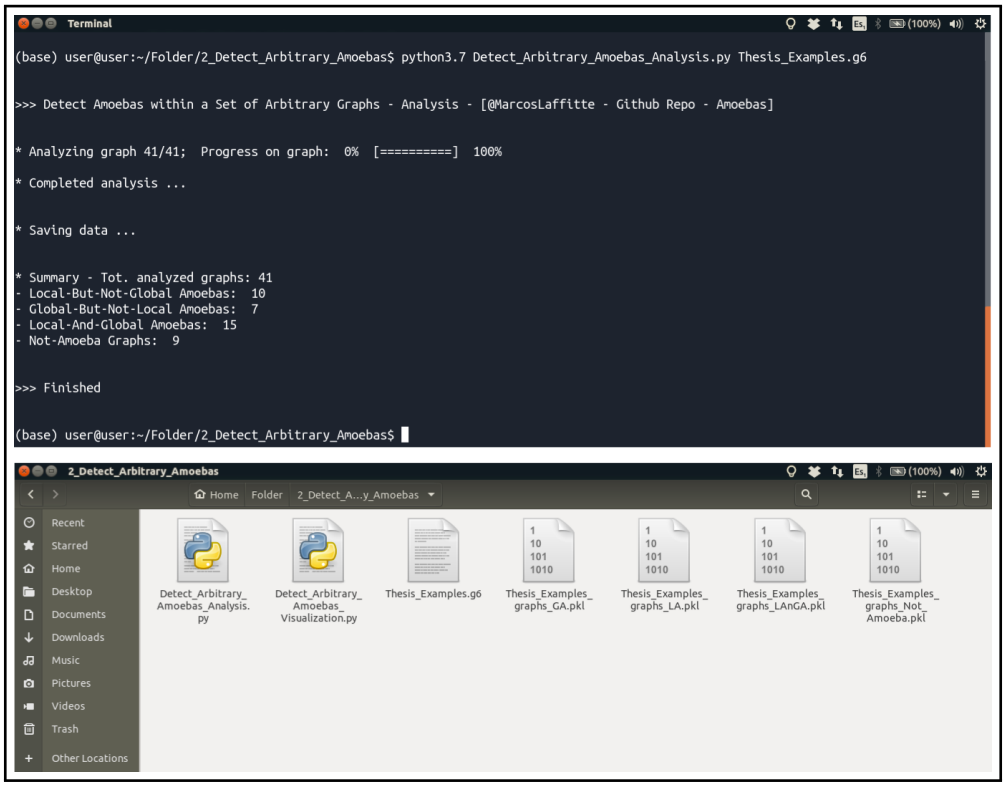
python3.7 G6_Builder_Visualization.py Thesis_Examples.g6



- 2_Detect_Arbitrary_Amoebas

Now we open the 2_Detect_Arbitrary_Amoebas folder and drag-and-drop the Thesis_Examples.g6 file into it, in order to detect the amoebas within it. To do so we run in a new terminal the command

python3.7 Detect_Arbitrary_Amoebas_Analysis.py Thesis_Examples.g6



Remember that this new terminal should be positioned in the 2_Detect_Arbitrary_Amoebas folder as its working directory for the given command to work. This will produce four pkl files containing the results of the analysis. We do NOT need to open these files, neither move them nor rename them.

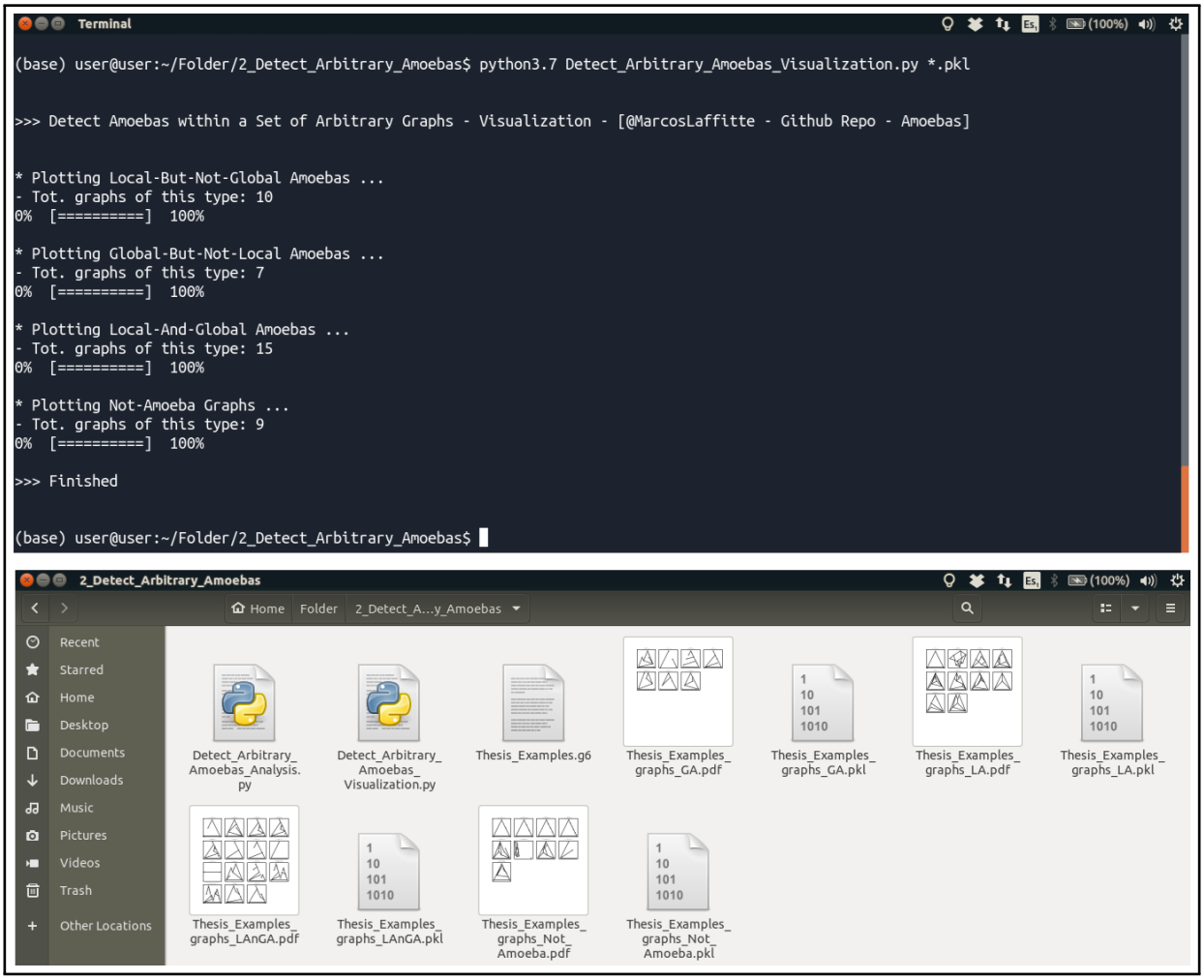
One of these files will contain the global-but-not-local amoebas (GA) detected inside Thesis_Examples.g6. Similarly, the local-but-not-global amoebas will be contained in the file with the (LA) annotation. In the same way we will have the local-and-global amoebas in the LAnGA file, as well as the not-amoeba graphs in the respective file. All these files are pickled python-lists of tuples containing information on these graphs, and may be opened and processed with custom programs if needed.

In order to plot these results we simply need to run the command

python3.7 Detect_Arbitrary_Amoebas_Visualization.py *.pkl

and the visualization program will order and process these files. Nevertheless, for this command to work there should NOT be other pkl files inside this folder at the same time, other than the four produced by the Analysis script.

After running this command we will obtain four pdfs as shown bellow, each displaying the plots of the corresponding amoeba and not-amoeba types.



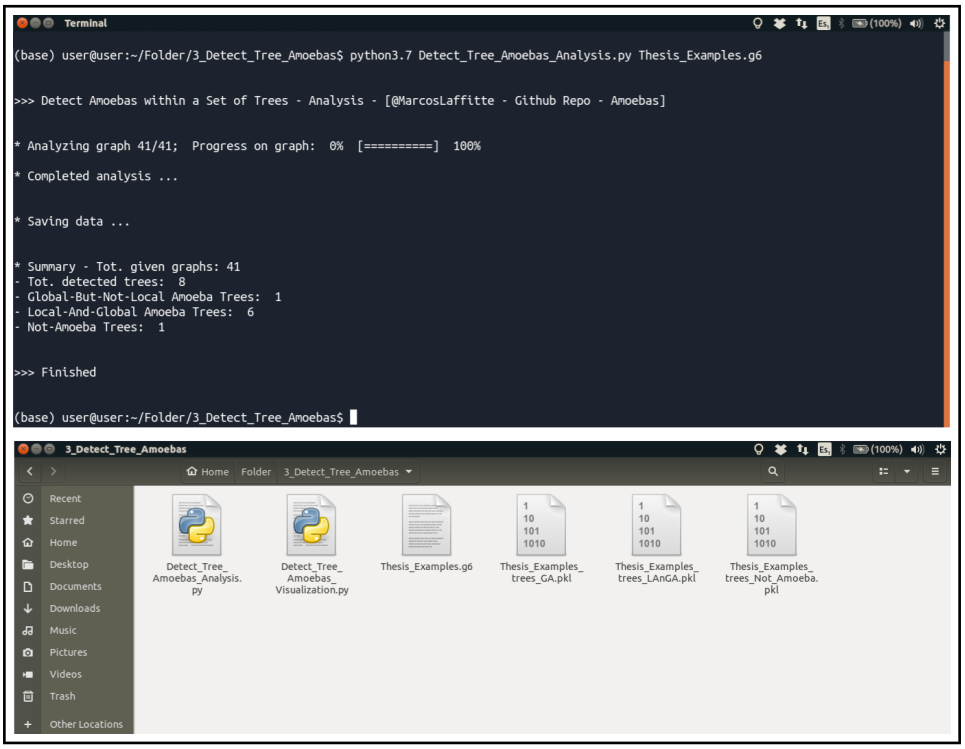
- 3_Detect_Tree_Amoebas

Similarly we can drag and drop the Thesis_Examples.g6 file into the folder 3_Detect_Tree_Amoebas so as to analyze only the trees inside this file. Then we can run the command

python3.7 Detect_Tree_Amoebas_Analysis.py Thesis_Examples.g6

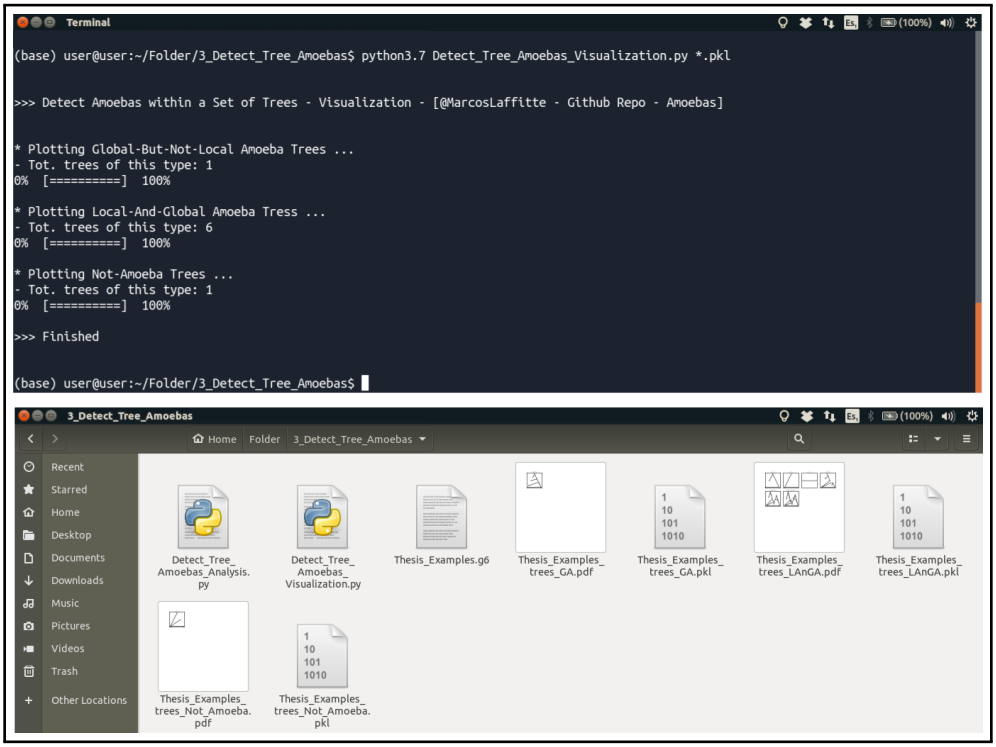
This program will filter out graphs with cycles in the Thesis_Examples.g6 files and only keep the trees. Then it will analyze this trees to determine those that are global-but-not-local amoebas (GA), local-and-global amoebas (LAnGA), or those that are not amoebas. Remember there are no local-but-not-global amoeba trees since every local amoeba with minimum degree 1 (or 0) is also a global amoeba.

All of these results will be again pickled into three pkl files made up from lists of tuples containing information on the analyzed trees and can be processed with a custom script if needed, but otherwise they should NOT be renamed nor removed from the folder.



Finally we can plot the results with the following command. Remember that there should be NO other pkl file inside this folder for the following command to work properly.

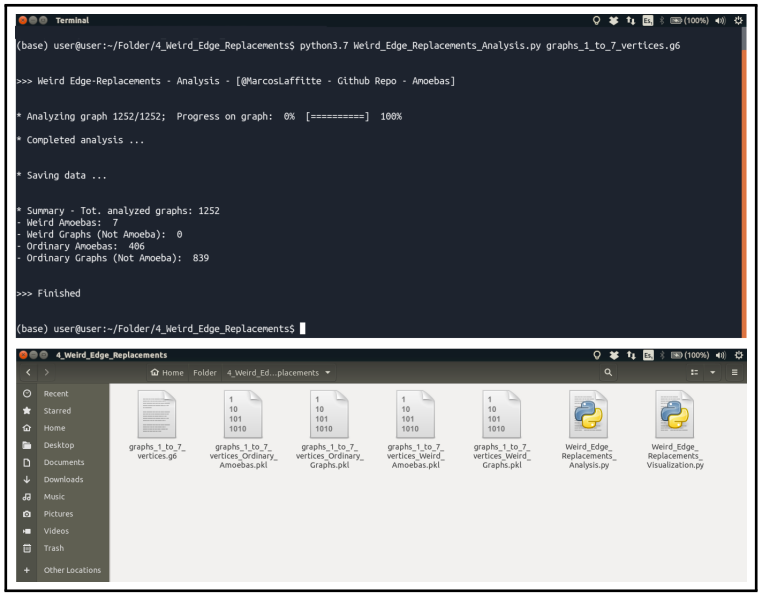
python3.7 Detect_Tree_Amoebas_Visualization.py *.pkl



- 4_Weird_Edge_Replacements

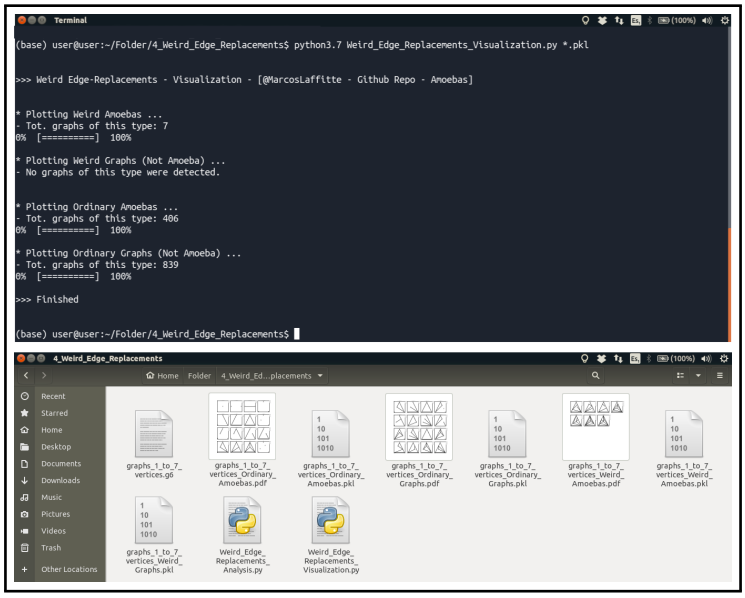
Even though the programs described before do obtain the feasible edge replacements of each graph, they do not report them in the final results. This is the task of the programs in this folder, since in this way we can also classify each non-trivial replacement as either weird or ordinary, which is a task that may itself take a bit longer. The figures bellow show the execution of these programs over the sample set graphs_1_to_7_vertices.g6 which can be found in the 6_Examples folder. To do so copy the file graphs_1_to_7_vertices.g6 into this folder and run the following command in a terminal opened here

python3.7 Weird_Edge_Replacements_Analysis.py graphs_1_to_7_vertices.g6



This can also be done using the Thesis_Examples.g6 file, but remember that in order to plot the results, there should NOT be any pkl file inside this folder, other that those produced by each analysis. To plot these results we can run

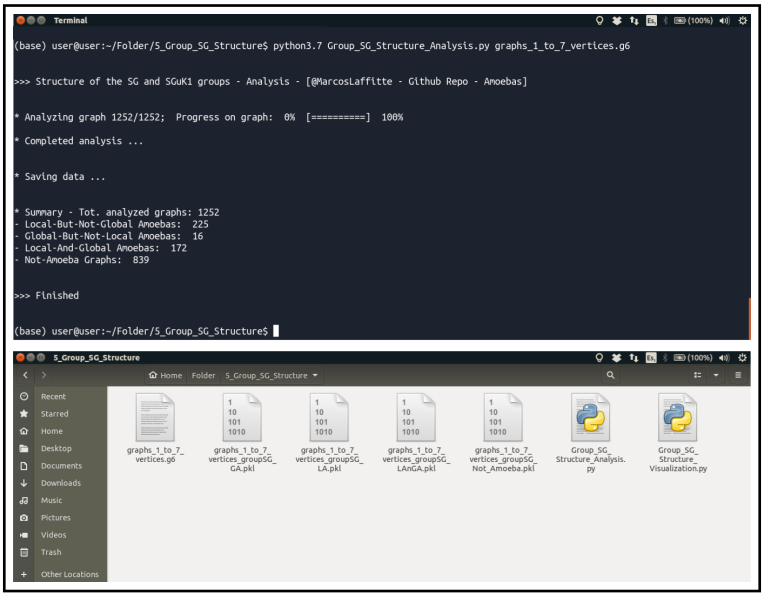
python3.7 Weird_Edge_Replacements_Visualization.py *.pkl



- 5_Group_SG_Structure

Finally the unique structure (up to isomorphism) of the S_G and $S_{G \dot{\cup} K_1}$ groups, as subgroups of the symmetric group S_n, for a given graph G of order n, can be determined with the programs in the folder 5_Group_SG_Structure. Again we drag and drop inside this folder the g6 list of graphs that we want to analyze, for example the file graphs_1_to_7_vertices.g6 from before, and run the following command

python3.7 Group_SG_Structure_Analysis.py graphs_1_to_7_vertices.g6



And similarly we plot the results with the command

python3.7 Group_SG_Structure_Visualization.py *.pkl

But remember again that there should not be any other pkl file inside this folder, different from the ones produced by the analysis, for this last command to work properly, as show below.