1 Tutorial to runnig Monoids on Jupyter-notebook and Sagemath: End-Product Inhibition cases.

First you need to install a jupyter notebook platform.

Step-by-step (Windows / macOS / Linux):
Go to: https://www.anaconda.com/products/distribution

Download the installer for your OS:

-Windows: .exe

-macOS: .pkg

-Linux: .sh

Run the installer and follow the instructions (default settings are fine).

After installation:

Open Anaconda Navigator and click on "Launch" under Jupyter Notebook, OR

Open a terminal / Anaconda Prompt and run: jupyter notebook, in Ubuntu: jupyter-notebook.

Once installed your jupyter-notebook platform. You follow the screenchots to runnig monoids for end-product inhibtion cases.

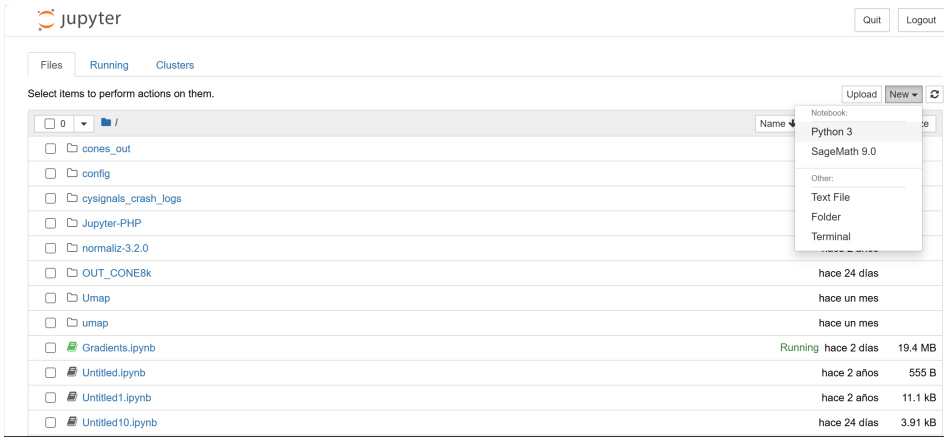

Figure 1: Open a new notebook for Python 3.

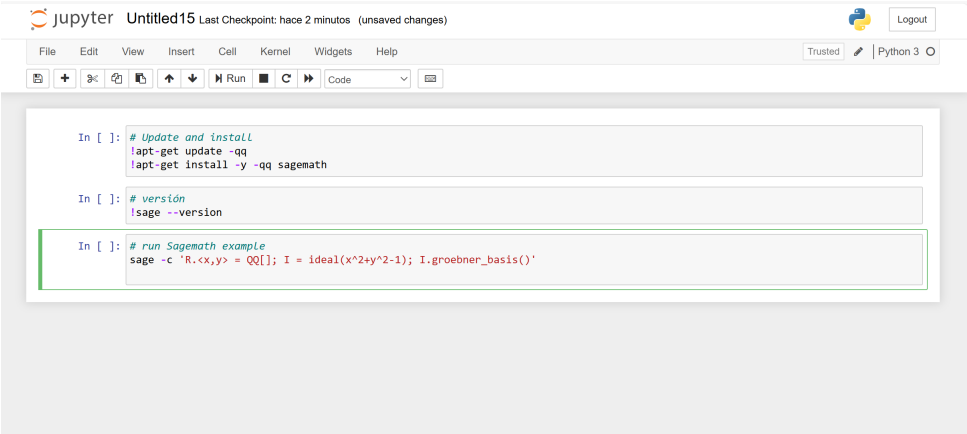

Figure 2: The next step is to install the platform to run algorithms in computational algebraic geometry. In our case to compute generators in monoids, you only copy and paste or simply write the code lines shown in order as is shown in figure.

Note: The script only run monoids for cones of dimension $n = 1$ up to $n = 9$. The corecto name files for each case of length of the pathways $n = 1, 2, ...9$, are given in the following: Cone3n_1.txt, Cone3n_2.txt, Cone3n_3.txt, Cone3n_4.txt, Cone4n_1.txt, Cone4n_2.txt, Cone4n_3.txt, Cone4n_4.txt, Cone4n_5.txt, Cone5n_1.txt Cone5n_2.txt, Cone5n_3.txt, Cone5n_4.txt, Cone5n_5.txt, Cone5n_6.txt, Cone6n_1.txt, Cone6n_2.txt, Cone6n_3.txt, Cone6n_4.txt, Cone6n_5.txt, Cone6n_6.txt, Cone6n_7.txt, Cone7n_1.txt, Cone7n_2.txt, Cone7n_3.txt, Cone7n_4.txt, Cone7n_5.txt, Cone7n_6.txt, Cone7n_7.txt, Cone7n_8.txt, Cone8n_1.txt, Cone8n_2.txt, Cone8n_3.txt, Cone8n_4.txt, Cone8n_5.txt, Cone8n_6.txt, Cone8n_7.txt, Cone8n_8.txt, Cone8n_9.txt, Cone9n_1.txt, Cone9n_2.txt, Con98n_3.txt, Cone9n_4.txt, Cone9n_5.txt, Cone9n_6.txt, Cone9n_7.txt, Cone9n_8.txt, Cone9n_9.txt, Cone9n_10.txt.

Note: In this linux command : cat *_TF_DNA31.sage >> OUT31_TF_DNAsage.txt. You only must to change the number in the name file. If you are in the case for length of the pathway $n = 4$. The output files, are written as:

cat *_TF_DNA41.sage ¿¿ OUT41_TF_DNAsage.txt
cat *_TF_DNA42.sage ¿¿ OUT42_TF_DNAsage.txt
cat *_TF_DNA43.sage ¿¿ OUT43_TF_DNAsage.txt
cat *_TF_DNA44.sage ¿¿ OUT44_TF_DNAsage.txt
cat *_TF_DNA45.sage ¿¿ OUT45_TF_DNAsage.txt
.

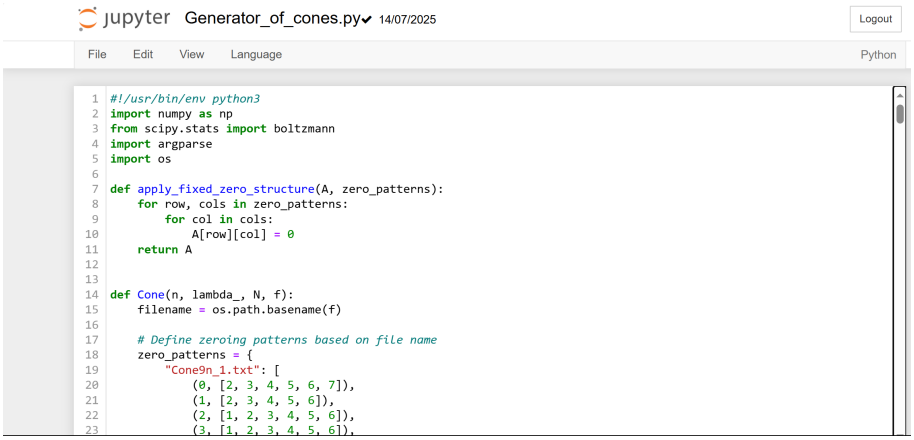

Figure 3: In the following, open the python script file called: **Generator of cones.py**. It which is uploaded in tree of gitgub. Copy and paste in a new notebook in python.

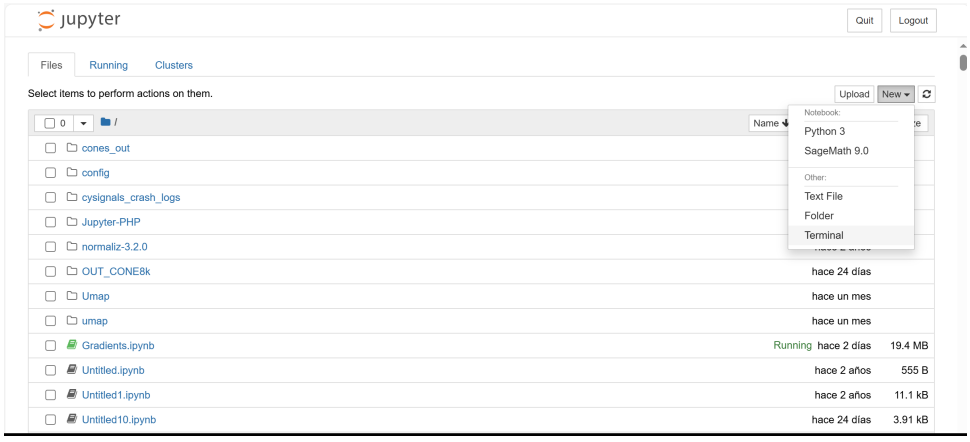

Figure 4: Once installed Sagemath version 9.0- Open a notebook in Sagemath.

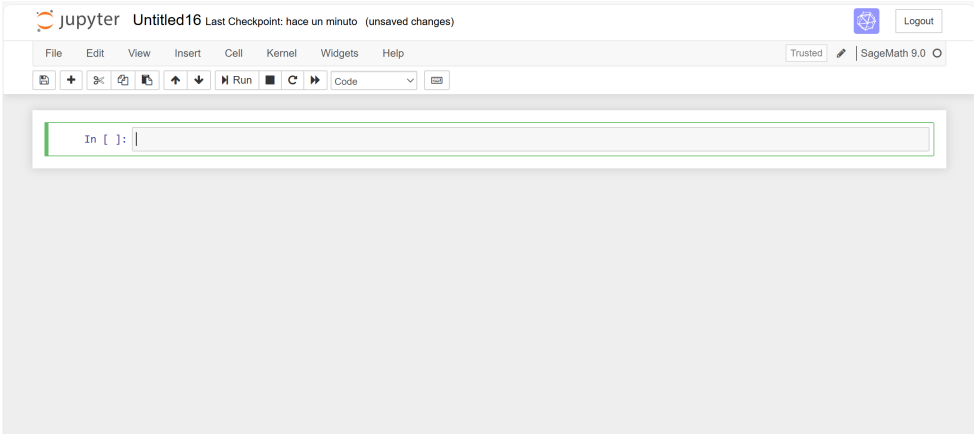

Figure 5: It will open a newbook in Sagemath. In a moment you will run a script in this prompt. You leave it open.

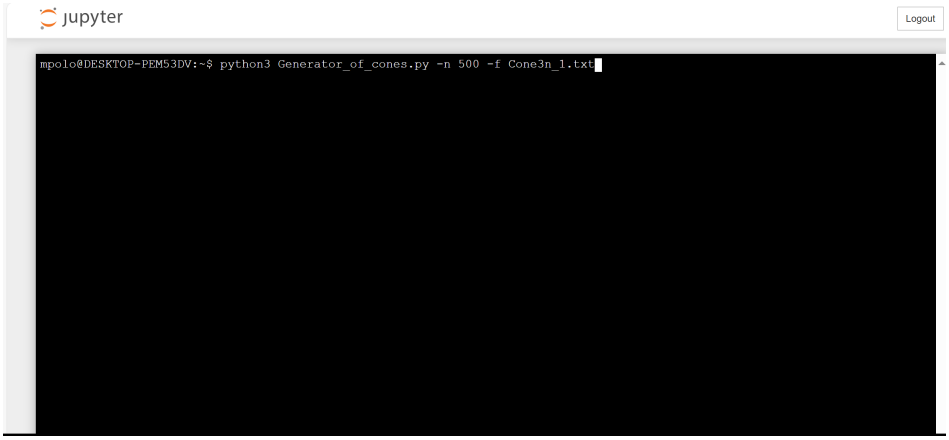

Figure 6: Open a new terminal in Jupyter. You write in the prompt the line shown in the figure. Only follow and respect the name files: Cone3n_1.txt, Cone3n_2.txt,..., or Cone7n_1.txt, Cone7n_2.txt, Cone7n_3,...,Cone7n_8.txt and similar for simulation of more cones and monoids for different length of patways n. The instruction $-n$ 500 lets to simulate 500 random matrices through Boltzmann distribution.

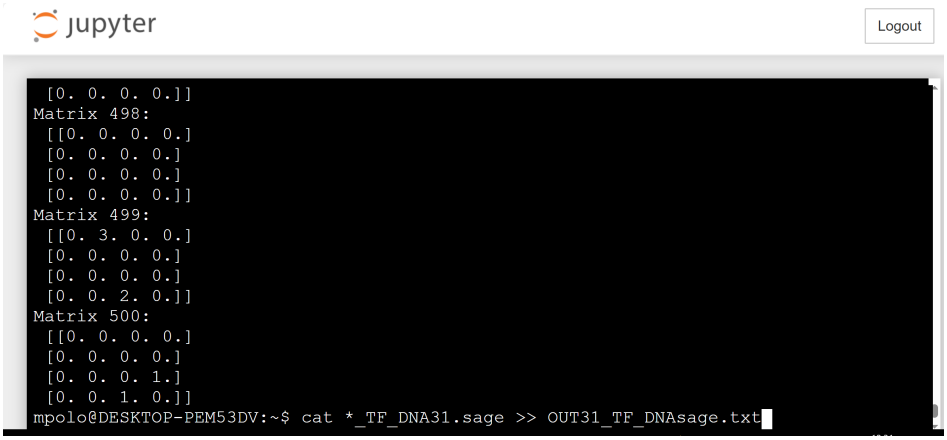

Figure 7: After you have generated 500 random matrices. You can write the instruction shown on the screenshot with the objective to save the whole random matrices in an one file called OUT31_TF_DNAsage.txt

Figure 8: In the following you can verify the content of this file in the Jupyter tree as is shown in the image.

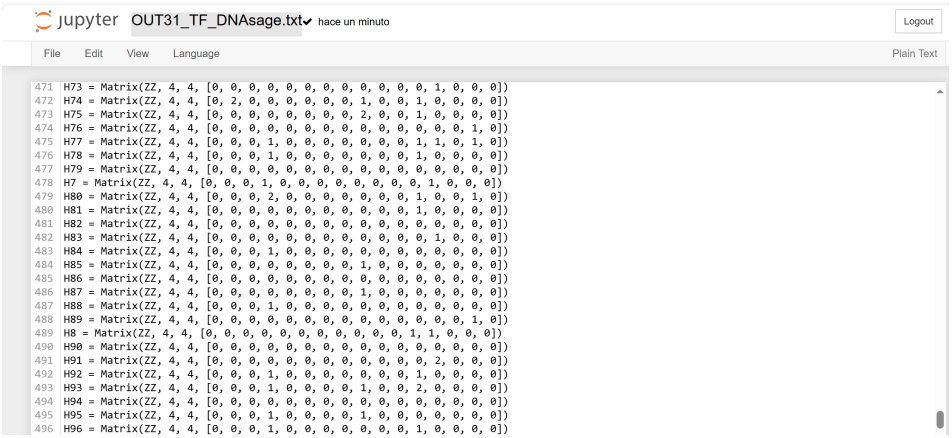

Figure 9: The content of the file it looks as is shown in the image. This files are prepared in format for Sagemath platform.

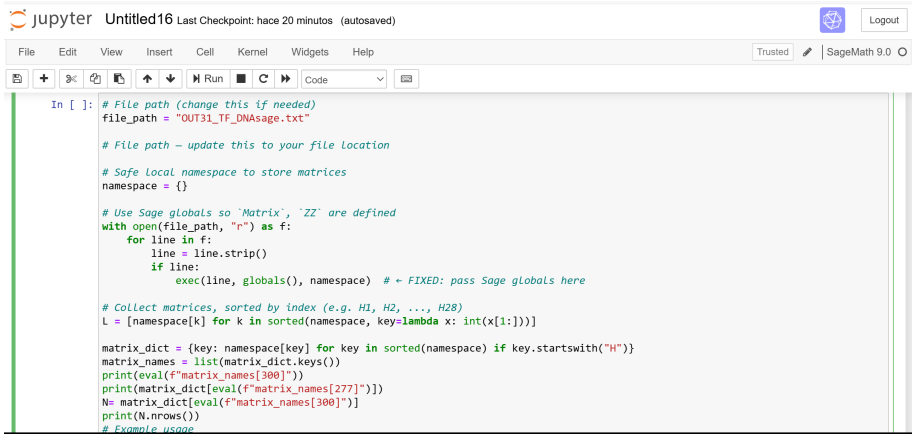

Figure 10: Now you come back to the notebook that you have left open in figure 5. Copy and paste the script called Sagemathscrpt.txt, this file text is uploaded in the github link. You must write in the begining of this script the right name of the file, in this case was "OUT31_TF_DNAsage.txt".

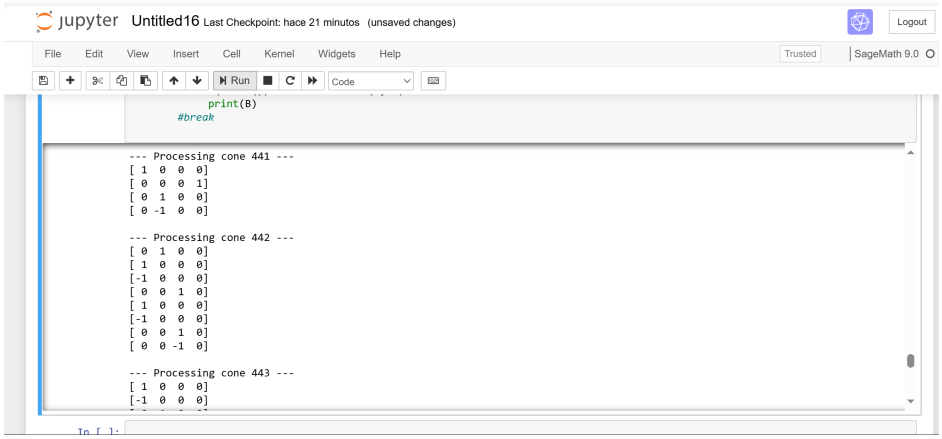

Figure 11: Once you have copied and pasted the script in the Sagemathscrip.txt file. You can run it. In the following you will 500 matrix representation for monoids for the length of the pathway n that you have chosen.