# README

This document is intended to help interested readers of “A von Mises-Fisher Distribution for the Orbital Poles of the Plutinos,” by IC Matheson, R Malhotra, and JT Keane, to reproduce the calculations and figures in the paper with as little fuss and recreation from scratch as possible, given the limitations of the code used by the authors in the first place. This document and its directory are intended to be uploaded to GitHub at <https://github.com/iwygh/mmk23a>. The paper will be submitted to MNRAS, the Monthly Notices of the Royal Astronomical Society. In theory, the journal paper itself should be sufficient to let any interested researcher reproduce the work. Any researcher who downloads this directory from GitHub should be able to reproduce the results in MNRAS nearly exactly, allowing for some small random variation due to the use of different random number seeds or small changes in the underlying software.

# SOFTWARE

This work is done on a 2018 MacBook Pro running MacOS Monterey 12.6. Cluster computations are done on the University of Arizona Puma cluster, which uses SLURM as its batch submission scripting language. Both local and cluster calculations are done partly in Fortran 95 using the **gfortran** compiler and partly in Python 3.8.2. Python packages include **Astroquery, rebound, scikit-learn, urllib, shapely, random**, and **json**.

# INSTRUCTIONS FOR FULL REPRODUCTION OF PAPER

Go to the JPL Solar System Dynamics Small Body Database query at [https://ssd.jpl.nasa.  
gov/tools/sbdb\_query.html](https://ssd.jpl.nasa.gov/tools/sbdb_query.html).

Expand the “Custom Object/Orbit Constraints” menu.

Select “Define New Custom Constraint,” then “Orbit and Model Parameter Fields.”

In the “Select field orbit” menu, choose “a”, then “<”, and finally enter “42” in the field. This is a constraint au. Repeat for au. These are very rough cuts of the approximate semimajor axis limits of the 3:2 mean motion resonance (MMR) that defines the Plutino population. These elements are heliocentric. Later the selection will be more refined through Python scripting.

Expand the menu “Output Selection Controls,” click “select all”, then “add.” Not all available fields will be used later, but it is easier to select all and then remove unneeded fields through Python scripting than to tediously select individual fields.

In the “Output Fields” menu box, click “Full Precision.”

Then click the green “Get Results” button.

Click “Download (CSV-format)”. Save the csv in the working directory for this project as **00\_sbdb\_query\_results\_[YYYYMMDD].csv**. In the GitHub directory, this is **00\_sbdb\_query\_  
results\_20221012.csv**. This is the raw list of KBOs that will be the base of all later computation. There are 887 objects in this list.

Now go to the Minor Planet Center database at [https://www.minorplanetcenter.net/  
data](https://www.minorplanetcenter.net/data). Right-click to save **MPCORB.DAT** in the working directory, as **00\_MPCORB\_[YYYYMMDD]  
.DAT**. In the GitHub directory, this is **00\_MPCORB\_20221012.DAT**. This will later be used to cross-reference against the Small Body database to eliminate some objects from consideration.

Save **00\_MPCORB\_[YYYYMMDD].DAT** in your main working folder. Save a backup copy in your Downloads folder.

Open **01\_prepare\_to\_check\_plutino\_resonances\_noclones.py** in your Python environment or development window. Make sure line 435 has ‘YYYYMMDD’ (e.g. 20221012), and lines 11-17 are for 2022-1-1, 00:00:00. Do the same for lines 790-796 of **00\_check\_plutino\_resonances\_noclones\_template.py**. Then run **01\_prepare\_to\_check\_plutino\_resonances\_noclones.py** locally.

The reduced SBDB database has 690 objects.

Next, upload the entire folder to the Puma cluster (or your local cluster). Note that **00\_MPCORB\_[YYYYMMDD].DAT** will not upload to Puma, as it is too large. This is why you need to back it up in your Downloads folder.

From the working directory on the High Performance Computing online file browser GUI, click “Open in Terminal.”

Submit the batch job using the command **sbatch slurm\_check\_plutino\_resonances\_noclones.  
slurm**. The SLURM file will need to be modified for your computer cluster and user account.

After the batch job has completed, delete the local folder from your computer and download the entire folder from the cluster. If the folder is too large to download using the GUI, it can be downloaded from the Puma cluster at the University of Arizona, using a MacBook command line, as follows:

**ssh [user id]@filexfer.hpc.arizona.edu**

**tar czvf [folder\_name].tar.gz [/path/to/folder]**

The compressed folder is then downloaded using the High Performance Computing online interface and unzipped.

Examine all files **plot\_False\_plutino [MPC packed designation] [MPC unpacked designation] Resonant 3 2\_noclones.pdf** and **plot\_True\_plutino [MPC packed designation] [MPC unpacked designation] Resonant 3 2\_noclones.pdf** on your local machine. Note if any objects seem to be classified incorrectly. Enter the designations of the false negatives (objects classified False that should be classified True) on line 51 of **02\_read\_Plutino\_resonances\_noclones.py** and enter the designations of the false positives (objects classified True that should be classified False) on line 52. Also make sure that lines 29-35 refer to ‘YYYYMMDD’ (e.g. 20221012) and 2022-1-1, 00:00:00.

Run **02\_read\_plutino\_resonances\_noclones.py** on your local machine. This will save the Plutinos found without using clones to the file **plutinos\_for\_mnras\_noclones.csv.**

This will produce 431 no-clone Plutinos.

Open **03\_prepare\_to\_classify\_clones.py** in your Python environment or development window. Make sure lines 265-271 are for 2022-10-12 and 2022-01-01, 00:00:00. Also make sure lines 166-172 of **03a\_classify\_clones\_template.py** are for 2022-10-12 and 2022-01-01, 00:00:00. Then run **03\_prepare\_to\_classify\_clones.py** locally.

Next, delete the entire folder from the cluster. Then upload the entire local version of the folder to the Puma cluster (or your local cluster).

From the working directory on the High Performance Computing online file browser GUI, click “Open in Terminal.”

Submit the batch job using the command **sbatch slurm\_classify\_clones.slurm**. The SLURM file will need to be modified for your computer cluster and user account.

After the batch job has completed, delete the local folder from your computer and download the entire folder from the cluster.

Open **04\_prepare\_to\_check\_Plutino\_resonances\_clones.py** and make sure lines 44-50 are for 2022-10-12 and 2022-01-01, 00:00:00. Then run it locally.

This should take in the list of 690 objects from the reduced SBDB database and find 427 objects that have each of 301 clones classified as Resonant. Next we need to make sure that those Resonant clones are actually in the 3:2 MMR. Delete the entire folder from the cluster. Then upload the entire local version of the folder to the cluster and run **sbatch slurm\_check\_plutino\_resonances\_clones.slurm.**

Download the folder from the cluster.

Examine all files **plot\_False\_plutino [MPC packed designation] [MPC unpacked designation] Resonant 3 2\_clones.pdf** and **plot\_True\_plutino [MPC packed designation] [MPC unpacked designation] Resonant 3 2\_clones.pdf** on your local machine. Note if any objects seem to be classified incorrectly. Enter the designations of the false negatives (objects classified False that should be classified True) on line 93 of **05\_read\_Plutino\_resonances\_clones.py** and enter the designations of the false positives (objects classified True that should be classified False) on line 94.

Now run **05\_read\_Plutino\_resonances\_clones.py** on your local machine. It should return 427 objects for which all 301 clones are Resonant, of which 424 are Plutinos.

Next, run **06\_calculations.py** on your local machine.

Then run **aa\_fortran\_mean\_planes.f95** on your local machine as gfortran aa\_fortran\_mean\_planes.f95 -o test and then ./test.

Next, run **07\_plots\_and\_figures.py** on your local machine.