Hello future scientists, browsing through my horrible workflow! Here’s everything that I did, in a format that’s hopefully readable. It will contain the steps I took in order to run the interactive map that shows all of the cores, and how to run the data synthesis that I performed.

STEPS FOR BOTH:

1. Start with the latest version of get data for map.sql. This is a SQL file that will use the original database to generate a table containing important data originally just intended for the map, including the lat/long, number of samples, and name of the core.
   1. Save the table as its own CSV file, what you name it doesn’t matter, but I’d name it something like data\_for\_map.csv.
2. Open up your Python interpreter of choice, you’re about to run something from the Python Scripts folder! Go ahead and boot up reverse geocoder.py, you’re going to need to change a thing or two before you get started.
   1. Change the filepath on line 10. This line reads in the .csv you saved, so you’ll need to change it to your path with your filename.
   2. Unless it’s been added to the database after I’m gone, there’s another table you’ll need to download. It includes the names of countries and their ISO codes. This is important for utilizing the outputs of the reverse geocoder, as the reverse geocoder only outputs the two-letter codes, and we want the three-letter ones. [This is the table](https://gist.github.com/tadast/8827699).
   3. You’ll need to change the filepath for this one too, on line 24.
   4. Run the script! It’ll take a good time to finish, so feel free to get a snack and/or use the bathroom once you’re sure the program will be running in your absence. Seriously, it takes a LONG TIME, it completely drained my computer’s battery the first time I ran it.
   5. The resulting table will be named data\_for\_map\_with\_countries.csv. It comes with a garbage column I couldn’t get rid of, so you’ll have to delete it manually if you don’t want it there.

STEPS FOR INTERACTIVE MAP:

1. Do all the steps for both above.
2. Use the latest version of interactive map.py. As of me writing this, it’s version 3, but that might have changed.
   1. This will require you to change the filepath at line 48. You’ll need to change your path, and also the name of the file, if you changed it.
   2. And that’s it, you’re done! It should print it to the screen and you can play around with its various features.

STEPS FOR THE DATA SYNTHESIS:

1. Do all the steps for both above.
2. Unless it has been imported into the database itself in my absence, you’ll need to download the Country Xwalk table. It can be found [here](https://docs.google.com/spreadsheets/d/146tHgM4guHj0I00QVkWTAF1n20923GXdyuJA451OzoQ/edit#gid=0).
3. Open up the latest version of entities\_in\_africa.sql. As of me writing this, it’s version 2, but that might have changed.
   1. This SQL query will use the results of the reverse geocoder, as well as the Country Xwalk table, in order to only return the entities that are in Africa. This file is currently set up to return various relevant bits of data, as well as the ones required for the data synthesis tacked on at the end.
   2. Run the query, grab the resulting table, and save it as a csv file. I’d recommend titling it entities\_in\_africa.csv. CSV found [here](https://drive.google.com/drive/u/0/folders/1BmI57_ciBOjtkLq9_o-r_5r1jhGYIbiz).
4. Open up data\_synthesis.py.
   1. You’ll need to change the filepath on line 5 to load the table you just generated from your path with the name you’ve given it.
   2. You’ll also need to change the filepath on line 26, but this one’s a bit more delicate. The first part of the path, the part before you add the entity name as the file name, is the part you’ll need to change. I’d definitely recommend setting up a dedicated folder for this part, because this is where the program will write a bunch of .pngs onto your computer.
   3. Run the program! It will generate a graph from each of the entities in Africa (currently 40)