**INSTRUCTIONS FOR MAC USERS ONLY**

**INSTALLATION OF BREW:**

You first need to install something called brew, which is essentially software that helps you download other things very easily. To install brew, first open terminal, which is where you will be running everything moving forward. To open terminal easily, press the command and space key at the same time, and then type “terminal.” Press return (the enter key) when terminal is highlighted.

Now, let’s download brew from the terminal. Type this into the terminal and then press enter:

**/bin/bash -c "$(curl -fsSL**[**https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh**](https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)**)"**

Some help installing homebrew, especially if using zsh, can be found here: <https://www.digitalocean.com/community/tutorials/how-to-install-and-use-homebrew-on-macos>

It should say “Installation successful!” somewhere in the lines above. Make sure you copy (command+C) the two commands listed under “Next Steps”

Now let’s make sure brew is up to date. Type this into the terminal and then press enter: **brew update**

Most Macs now use zsh, so after changing to zsh use the following command:

**/bin/zsh -c "$(curl -fsSL**[**https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh**](https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)**)"**

**INSTALLATION OF PYTHON:**

Now that we have brew installed and updated, let’s use it to download python3 (the most recent version of python). Type this into the terminal and then press enter: **brew install python3**

The download may take a minute or two.

Once the download is complete, scroll up a bit and make sure in the line of code, something reads like this:

**🡺python@3.xx**

Python has been installed as

/opt/homebrew/bin/python3

**INSTALLATION OF PIP:**

Pip is another software that helps us download other software. Installation of it is similar to downloading brew. Just type this into the terminal and press enter:

**curl**[**https://bootstrap.pypa.io/get-pip.py -o get-pip.py**](https://bootstrap.pypa.io/get-pip.py -o get-pip.py)

Note, you may be able to skip the above step and go straight to this one:

Then type this into the terminal and press enter: **python3 get-pip.py**

**INSTALLATION OF DEPENDENCIES THAT OUR SCRIPT WILL NEED TO RUN:**

If you open the script that we will be running (it should end in .py, because it is a python file), you will notice things like Pillow and numpy. These are dependencies that our script needs to be able to run. Think of them as necessary parts for the script to work correctly.

To install these dependencies, type this into the terminal and press enter: **brew install Pillow numpy**

**LOCATING THE SCRIPT:**

Now we need to run the script. Take note of where the .py script is on your computer. The location of this script can be thought of as a “directory,” … a location on the computer where the file is. For example, a folder called worm may be in this directory: /Users/lcato/worm (this is where it is for me). This means that the folder worm (where my .py file is) is also in a folder called lcato, which is itself in a folder called Users.

Every time we run a script, we have to make sure the terminal we are working in is in the directory that has this script. This can be tricky at first, so I advise that you stick the folder that you are doing all of your work in under your username folder. Click Finder (on mac) and this should be the folder next to the house symbol. This is the default directory for the terminal. Another way to find it is to press Shift+Command+H. To add it to your Favorites, two finger tap on your username at the top of the username folder, select Users and then click+drag your username folder into the Favorites bar on the left. That will make it easier to find in the future.

If you don’t have a folder called “Worm” in your username folder, add one now. Open that folder and create two new folders, “Reports” and “Images”. Copy and paste the worm.py script into the Worm folder.

Let’s try it. Exit out of the terminal. Then re-enter a new terminal like before (see above). Type **pwd** to “print working directory.” This tells you what the directory your are currently in is. Mine says /Users/lcato.

Type **ls** (“el ess”) and then enter to see all files in this directory. I always do this when moving into a new directory. It helps me see what is in the folder I am in. You should see the folder where all of your stuff is, including the worm.py file (this is what I called the script we will use).

To move into that folder, type **cd** and then the name of the folder. The folder where worm.py is on my computer is called worm. So, I will type **cd worm** in the terminal and then press enter. Type **ls** and then enter to make sure all the files are there, including the worm.py file.

Type and enter **pwd** into the terminal. Write this down or save it somewhere. For all future uses, when you open a new terminal, you can just type **cd** followed by whatever pwd gave you to quickly move into the folder from anywhere on the computer. For example, I type cd /Users/lcato/worm to set the directory in that folder.

Add your processed images to be analysed to the Images folder. Don’t forget to remove these images after you run your analysis or they’ll be included in the next analysis.

**MODIFYING THE SCRIPT:**

One of the most critical parts of getting the script working properly is to tell it where to look for the images, where to put the report and what the properties of the images are.

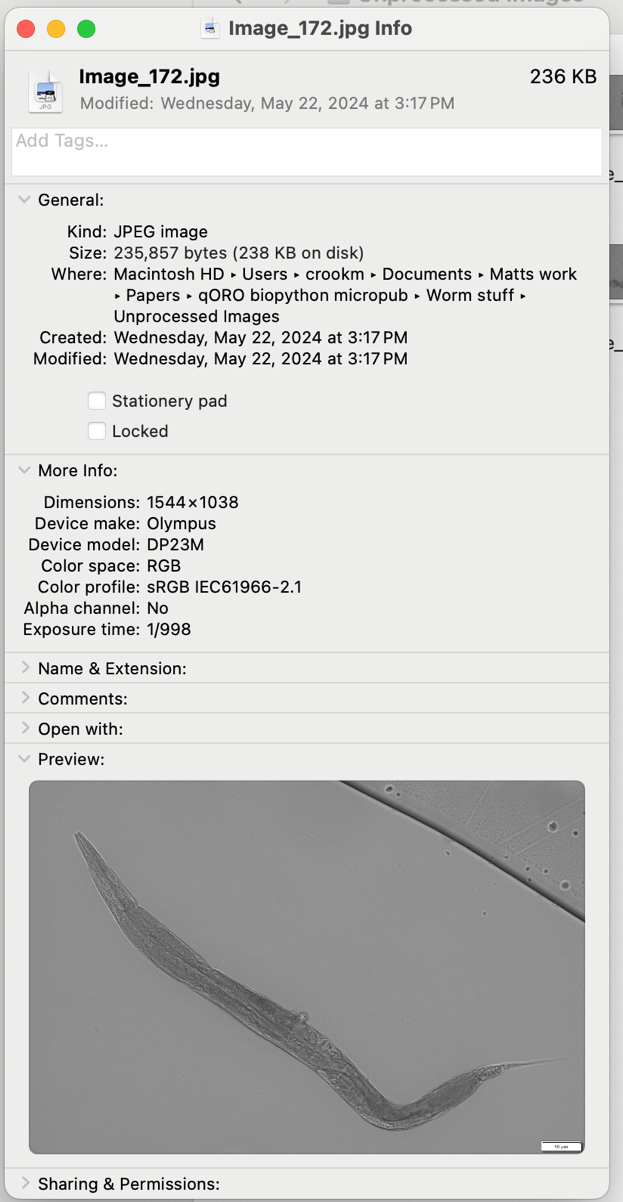
Open the worm.py script in TextEdit and find the part that says:

path\_to\_images = "/Users/lcato/Worm/Images/\*.jpg"

path\_to\_report = "/Users/lcato/Worm/Reports/report.txt"

Change the username to yours (mine is Drmat for some reason). Make sure the image file extension matches your images. If you use .tif or .jpeg files, change that in the script above to match.

Next, you need to make sure the image size in the script matches that of your images. To find your image resolution, two finger tap on the image and click “Get info”. You’ll see something like this:



The numbers you want are next to “Dimensions” under “More info”.

Then find the two places in the script that tell Biopython the size of your images.

First:

pic = Image.new('I', (1544, 1038))

pic.putdata(fg\_data)

and second:

bg\_total = total - fg\_total

bg\_mean = bg\_total / ((1544.0 \* 1038.0) - area)

Change both those sets of numbers to match those for your images. Save the new version of the worm.py script and make sure it’s in your Worm folder.

**RUNNING THE SCRIPT:**

Ok, now that we are in the folder where the .py folder is, all we have to do is use python3 to run it. Type and enter this into the terminal: **python3 worm.py**

Check the text file to ensure that the program runs. This is a text file in the reports folder that should give you new lines of code if it ran. This txt file will add new lines of code every time your run the .py file. If you want a clean txt file, you must delete the txt file before running worm.py.