Farm to table science with JWST

Jordan D'Silva

November 2024

Introduction

We are gonna reduce some data from the James Webb Space Telescope, and then analyse the data. We'll need about $\sim 10 - 15$GB of disk space for this exercise. You shouldn't need to have any programming experience, but we are gonna look at some scripts and run some programs! Don't be afraid to google if you're stuck and would like to try solve the issue yourself!

1 Getting JumProPe and necessary files

I, and our group at UWA, have spent a large part of my PhD designing a novel software package to process JWST data called JumProPe, which stands for JWST UWA Multiwavelength ProTools Processing Endeavour. Lets get some files and the software to start playing around with data.

Suggested time for this section: ~ 15 minutes.

1. Open the Terminal app.
2. Type `git clone https://github.com/JordanDSilva/jpdemo` this will create a new directory with some codes and such for this demo. Next type `cd jpdemo` to navigate to this cloned directory. You can also open Finder and click `Go -> home` to navigate to the `jpdemo` directory in the file explorer.
3. Next, type `Rscript rpckgs.R` to install the necessary packages for the `R` programming language (what JumProPe is built on). You should be able to just follow the on-screen instructions for installing these but let me know if there are any issues. Come and see me if you have problems with this step! This may take a while. But you should see that a new folder `pckgs` has appeared in the finder window (where the packages are being installed).
4. Now, type `bash download.sh` to download some JWST files from the Mikulski Archive for Space Telescopes. Pro tip: you can press `tab` to auto complete the command. So try typing `bash do` and then press `tab` to autocomplete! This will probably take ages because it has to download files all the way from Baltimore, USA. So I'll suggest that you press `CNTRL+C` to cancel the task and instead download the files from dropbox: `https://shorturl.at/BRKML`.

2 Making images for science

With some files and the software now downloaded, we can reduce the data to be used for scientific analysis. In this section, we will run JumProPe and then produce a colour image of an extragalactic field. Suggested time for this section: ~ 30 minutes.

1. Type `source initialise_env.sh`. This will add some 'environment variables' to help us run JumProPe.
2. Navigate to the JumProPe directory using `cd JUMPROPE/PROCESS` and then type `ls` to list all of the processing codes!
3. Let's now run the process using `Rscript zork_process.R`. The software will prompt you to set up some directories as shown in Figure 1:
   Type everything out as it appears in the above figure. Then type `1,2,3,4,5,6,7,8,9,10` to run

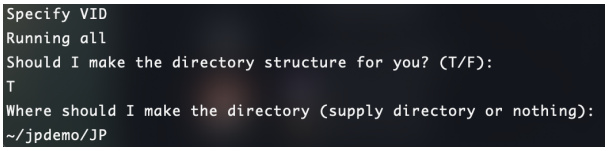

Figure 1:

those 10 steps. The entire process should take around 10 minutes or so. Depending on how we get on time, I can just provide you with the final files.

4. In finder, we can also navigate to the JumProPe directory and check out all of the stuff that we made. Here, we can appreciate the data volumes required for modern day astronomy! Have a look at the colour image in the `Patch_stacks` directory as shown in Figure 2.

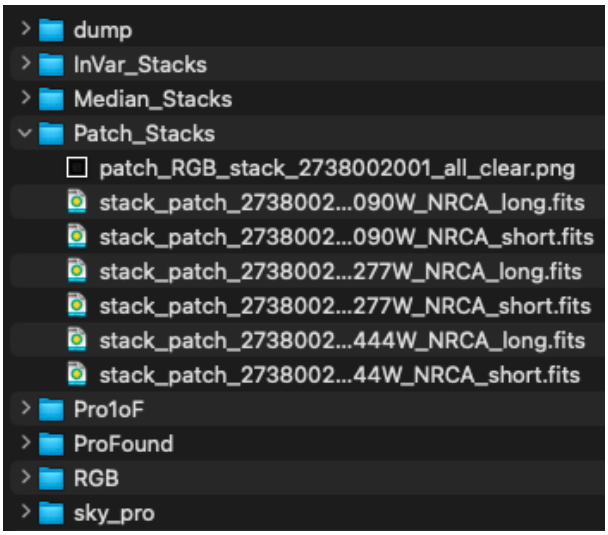

Figure 2:

3 Multiband analysis

With the images, we can now do some science. Navigate to the `PROFOUND` directory in JumProPe. You can 'go back' using `cd ..` and then use `ls` to check where you are.

Suggested time for this section: $\sim$ 10 minutes.

1. Let's make some galaxy catalogues. Type `Rscript zork_profound.R`, follow the on-screen instructions, and type `2,6,10`. Note that here we are only running a few steps out of the pipeline for simplicity. And we can checkout the detection here as per Figure 3.
2. Navigate back to the parent directory with `cd ~/jpdemo` and then type `Rscript plot.R` to make a scientific plot from our fancy JWST data!

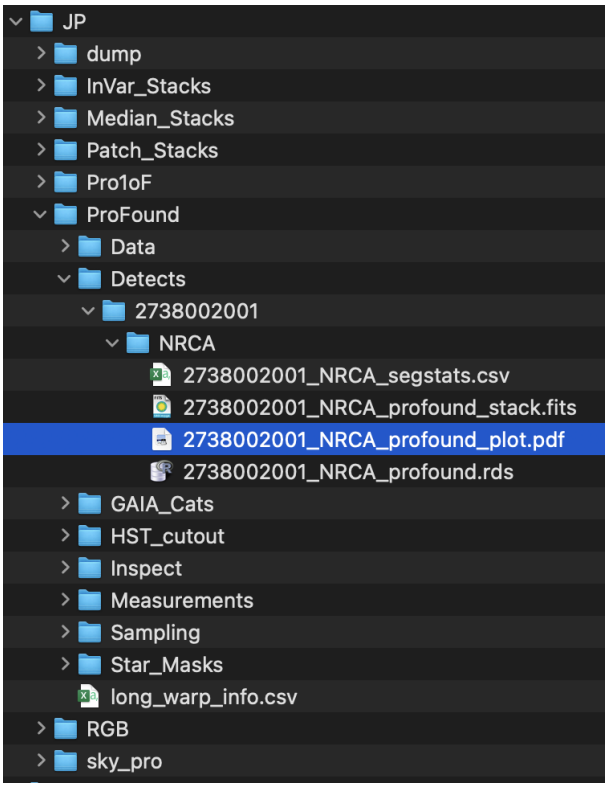

Figure 3:

```
skycut = 2.0, #Edit this,
pixcut = 9.0, #Edit this,
ext = 1.0,

smooth = T,
sigma = 1.0, #Edit this,

tolerance = 10.0, #Edit this,
reltol = 5.0, #Edit this,
```

Figure 4: