**Instructions for RDI**

Creating the reference library:

1. Open up the keck database and find the column with the data of your chosen epoch (for example HR8799’s epoch is 28/08/2012 so the column is 12Aug) <https://docs.google.com/spreadsheets/d/1XLTNtghPAZx4Jhspb8_6mHPhLNo8COMP2fV5Q2EKrCE/edit?invite=CPjP7JYE&ts=5ecf7289#gid=610297215>
2. Make a list of the names of the stars with a “1” in your chosen epoch’s column.
3. To check if there is more than one epoch within this month log onto analysis:

**ssh -XY** [**abc123@analysis.astro.ex.ac.uk**](mailto:abc123@analysis.astro.ex.ac.uk)

**cd ../../data/shinkley/Keck\_Data/<Year of epoch>**

**ls**

If there is only one date within the month of your epoch skip the next step

1. You have to check which stars have the right epoch. This can be done manually using by navigating to each stars folder in DangerZone

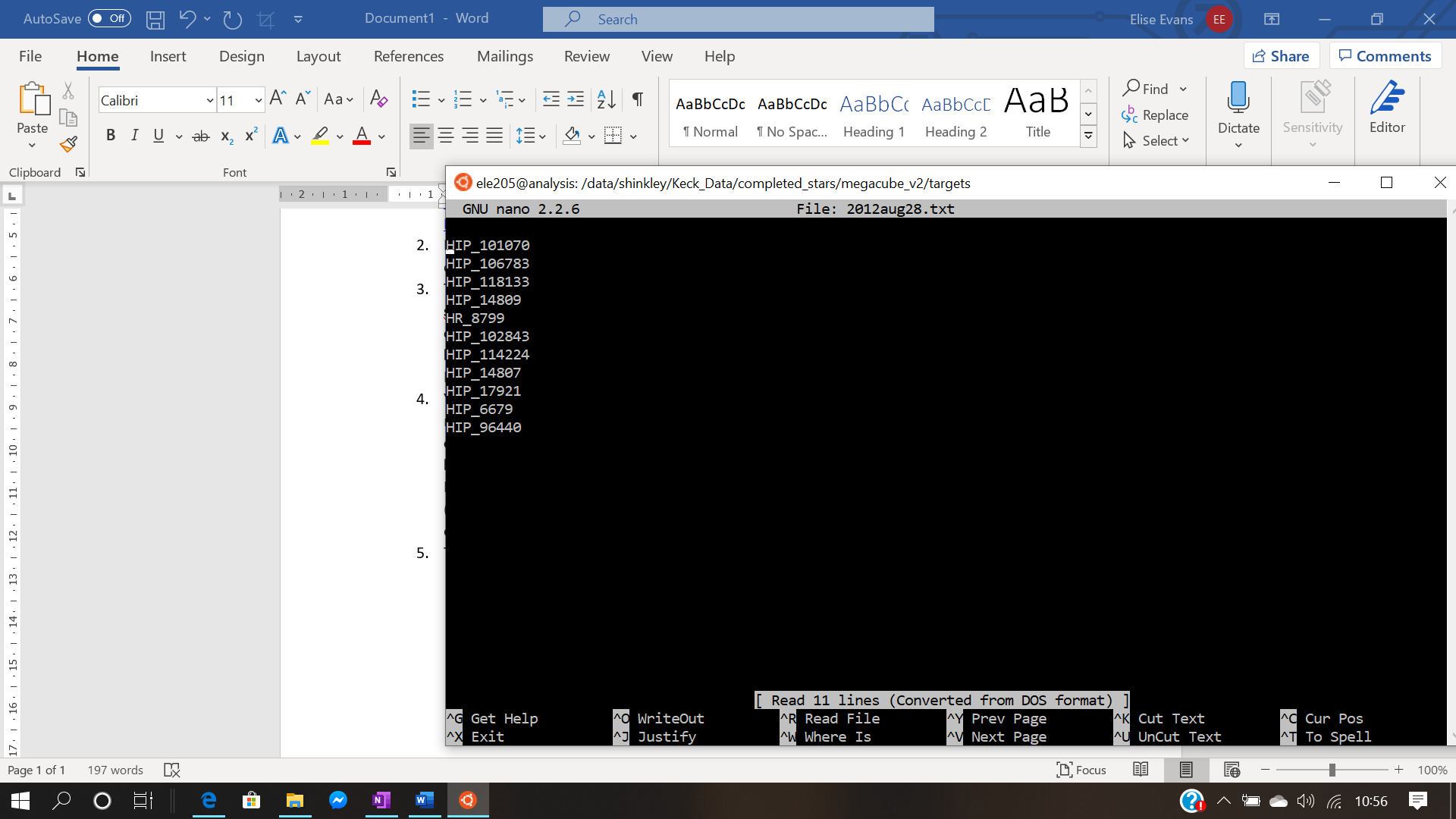
**cd ../completed\_stars/DangerZone/<Name of star>**

**ls**

If the star does not have the right epoch remove it from your text file list.

(You could also wait and just remove the stars once you have run the code as it’ll produce an error if the star doesn’t have the right epoch and they can be removed then)

1. The text file of the list of targets needs to be in the form, including the underscore in the star names:



To create this file navigate to the targets folder in megacube\_v2, which is in completed\_stars. If you’re in the folder completed\_stars:

**cd megacube\_v2/targets**

**nano <epoch>.txt** where epoch is in the form 2012aug28

and then write the list of stars, and save and exit using Ctrl X.

You could also create this file on your laptop and copy it to the targets folder using:

**scp <epoch>.txt abc123@analysis.astro.ex.ac.uk:/data/shinkley/Keck\_Data/**

**completed\_stars/megacube\_v2/targets**

1. Navigate to the megacube\_v2 folder

**cd ..** (if you’re in the targets folder)

1. Run the cubestitch code using:

**python cubestitch3.py -create <epoch>** where epoch is in the form 2012aug28

1. Outputs:

* <epoch>\_megacube.fits (in output folder) --- The reference cube
* <epoch>\_\_target\_frames\_data.csv (in stitch\_data folder) --- A dictionary of each star in the reference cube and their corresponding image numbers
* <epoch>\_\_angles.csv (in stitch\_data folder) --- A dictionary of each star and their parallactic angles

1. Open ds9 and the <epoch>\_megacube.fits file (ds9 is found in analysis in **usr/local/ds9** then **./ds9**) and click through every image, writing down the numbers of any frames that are bad.

You might also find that certain stars don’t have their centre exactly on 512,512. There is a automatic align function but that works if all the stars are in roughly the same place. There is a manual\_align function that can be used if you think every image of one star is massively off. You can open the <epoch>\_\_target\_frames\_data.csv file using nano and find the name of the star corresponding to the frames that are not centred. Then if you run:

**python cubestitch3.py -manual\_align <epoch>**

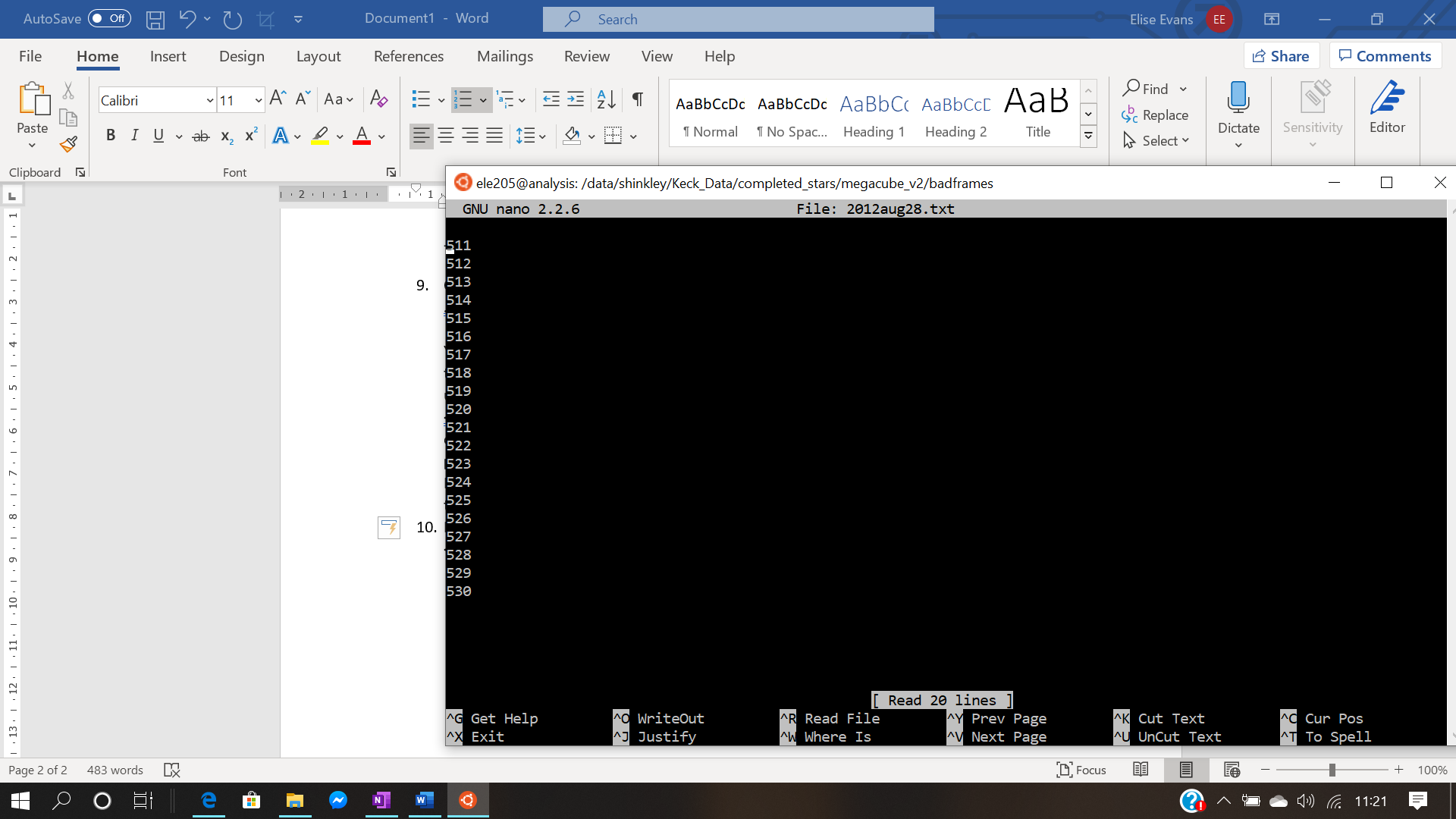
It will loop through asking if you want to align each star and then ask for the coordinates of the centre of the star.

1. To align the images using a 2D gaussian:

**python cubestitch3.py -align <epoch>**

This creates a new fits output <epoch>\_aligned.fits

1. If there are bad frames, a file needs to be created in megacube\_v2/badframes called <epoch>.txt which can be done in the same way the target file was created in step 5.



1. The bad frames can be removed using:

**python cubestitch3.py -remove <epoch>**

This produces a new fits file called <epoch>\_trimmed.fits as well as removing the frames from the two dictionaries in stitch\_data

Running the RDI code:

1. Navigate to your stars folder in DangerZone and create a new folder called RDI.

For example, from megacube\_v2:

**cd ../DangerZone/<Star>**

**mkdir RDI**

**cd RDI**

1. In this folder, you’ll need the RDI script and the psf.fits file

**cp ../../HR8799/RDIv2/RDIv3.py .**

If the psf is in the epoch’s folder:

**cp ../<epoch>/psf.fits**

1. To run ADI:

**python -W ignore RDIv3.py <epoch> <star> <Maximum PCs> ADI**

e.g python -W ignore RDIv3.py 2012aug28 HR\_8799 1 ADI

The epoch needs to be in this form, and the star name needs an underscore (to be the same format as the target list before). The code loops through each number of PCs so it can take a long time if the number is too high (I was testing it just using 1). The -W ignore is just to stop warning messages filling up the screen.

1. The code uses PCA to model the noise so it will ask you for an x,y coordinate input of a bright spot. If you’re doing this on a star you’ve already run the project script for then you might have some coordinates written down. If you don’t you’ll need to open the <epoch>\_\_target\_frames\_data.csv file in nano, find your target star and the corresponding image number (if your star is third in the list then the image numbers will be in the third list) Then open the .fits file in ds9 and write down the coordinates of a bright spot
2. Outputs:

* It prints the optimal number of PCs in the console
* <star>\_ADI\_<epoch>\_PCAS.fits --- A cube containing the processed images for each PC (the first image is PC=1, the second is PC = 2 etc, so the image with the optimal number of PCs can be chosen)
* ADI\_contrast\_curve\_<epoch>.png --- A picture of the contrast curve for the optimal number of PCs
* ADI\_contrast\_curve\_<epoch>.txt --- A text file of all the data from the contrast curve
* ADI\_snr\_<epoch>.txt --- A text file of the signal to noise ratio for each PC, this can be mostly ignored it just demonstrates the optimal number of PCs has the best S/N ratio

1. To run RDI:

**python -W ignore RDIv3.py <epoch> <star> <Maximum PCs>**

e.g python -W ignore RDIv3.py 2012aug28 HR\_8799 1

This produces the same outputs as ADI (but with RDI instead of ADI in the file name)

You can open the PCAS.fits file in ds9 to check it’s worked!

Possible errors:

1. Originally I got an error that the vip module was not installed. I fixed this by installing miniconda on my home directory

**wget https://repo.continuum.io/miniconda/**[**Miniconda3-py37\_4.8.3-Linux-x86\_64.sh**](https://repo.anaconda.com/miniconda/Miniconda3-py37_4.8.3-Linux-x86_64.sh)

**bash** [**Miniconda3-py37\_4.8.3-Linux-x86\_64.sh**](https://repo.anaconda.com/miniconda/Miniconda3-py37_4.8.3-Linux-x86_64.sh)

**pip install vip-hci**

1. I found that some of the stars didn’t have a centred cube in their dangerzone folder so the cubestitch code came up with an error saying the file didn’t exist. You can check you’ve typed the name of the star correctly and then just remove this star from the list of targets. This also applies if the centred cube is not in the normal location as the code won’t be able to find it.