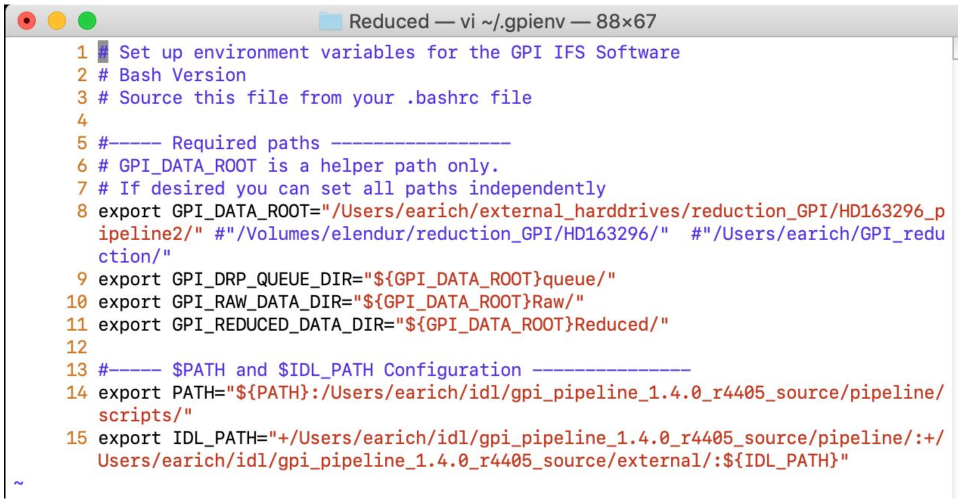
This document will work as a working README including instructions on how to operate the code. Then, a list of changes to the code from the original program written by Anna Laws and used in her 2019 paper. Finally, a list of todo’s that we would like to see changed about the pipeline.

README:

To use the pipeline, you must have python 3, idl, and the IDL GPI reduction pipeline installed.
Note! There are two ‘pipelines’ that we are using. There is the IDL GPI reduction pipeline (http://docs.planetimager.org/pipeline/) that we will refer to as the IDL package. Then there is also the python pipeline that automates (and is basically a wrapper) the IDL package that is originally written by Anna Laws. We will refer to the former as the python pipeline (or the pypeline).
Note! We highly encourage you to reduce one set of data without using the pypeline so you can understand what is the IDL package is doing.

First you want to ensure that the IDL package is pointed at the data that you are interested in looking at and that it will save the data in a location that you want it to be located. To do this, edit the .gpienv file. Example below:

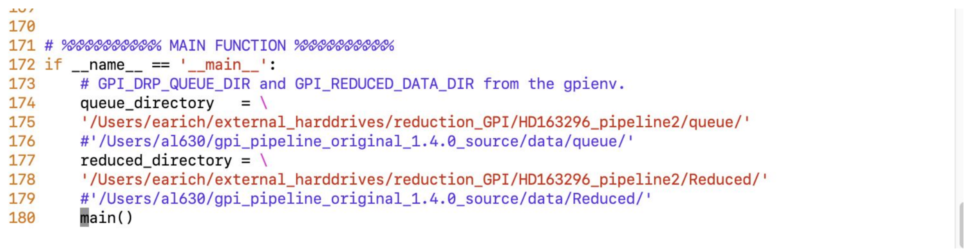


```
# Set up environment variables for the GPI IFS Software
# Bash Version
# Source this file from your .bashrc file

#----- Required paths -----------------
# GPI_DATA_ROOT is a helper path only.
# If desired you can set all paths independently
export GPI_DATA_ROOT="/Users/earich/external_harddrives/reduction_GPI/HD163296_pipeline2/" #"/Volumes/elendur/reduction_GPI/HD163296/"  #"/Users/earich/GPI_reduction/"
export GPI_DRP_QUEUE_DIR="${GPI_DATA_ROOT}queue/"
export GPI_RAW_DATA_DIR="${GPI_DATA_ROOT}Raw/"
export GPI_REDUCED_DATA_DIR="${GPI_DATA_ROOT}Reduced/"

#----- $PATH and $IDL_PATH Configuration ---------------
export PATH="${PATH}:/Users/earich/idl/gpi_pipeline_1.4.0_r4405_source/pipeline/scripts/"
export IDL_PATH="+/Users/earich/idl/gpi_pipeline_1.4.0_r4405_source/pipeline/:+/Users/earich/idl/gpi_pipeline_1.4.0_r4405_source/external/:${IDL_PATH}"
~
```

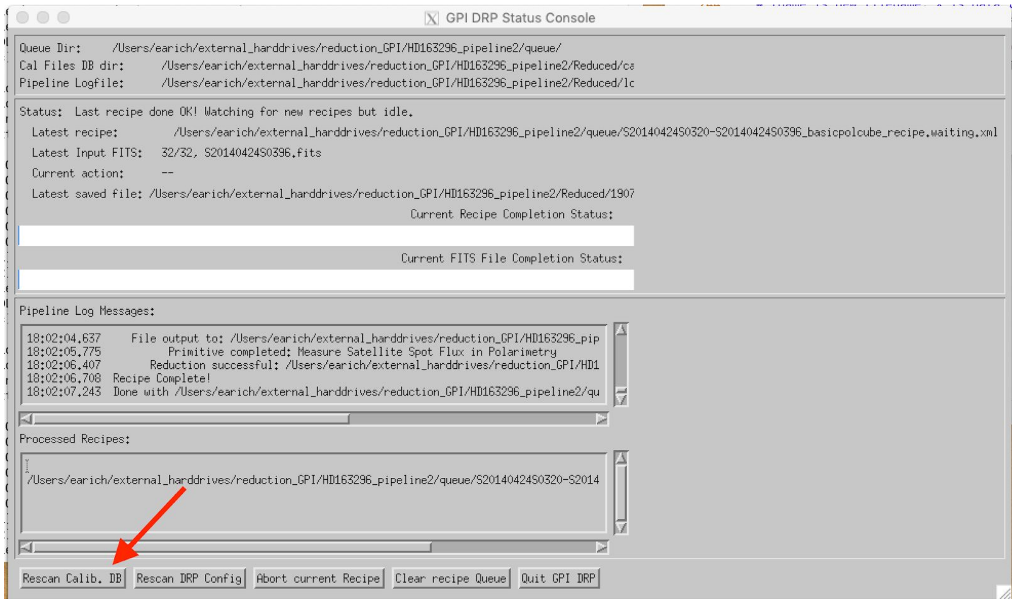
Next, edit the pipline.py file such that it points to the same directories that the IDL package is pointed to in the .gpienv file you just edited above. **Note: This should be changed so this is not necessary in the future.**



```

# %%%%%%%%%%% MAIN FUNCTION %%%%%%%%%%%
if __name__ == '__main__':
    # GPI_DRP_QUEUE_DIR and GPI_REDUCED_DATA_DIR from the gpienv.
    queue_directory   = \
    '/Users/earich/external_harddrives/reduction_GPI/HD163296_pipeline2/queue/'
    #'/Users/al630/gpi_pipeline_original_1.4.0_source/data/queue/'
    reduced_directory = \
    '/Users/earich/external_harddrives/reduction_GPI/HD163296_pipeline2/Reduced/'
    #'/Users/al630/gpi_pipeline_original_1.4.0_source/data/Reduced/'
    main()
```

To operate, first open up the IDL GPI pipeline by executing gpi-pipeline into a terminal window. Ensure that the calibration folder has been scanned. This is done in the GPI DRP Status Console window. Arrow pointing to the button below:



```
Queue Dir:      /Users/earich/external_harddrives/reduction_GPI/HD163296_pipeline2/queue/
Cal Files DB dir:      /Users/earich/external_harddrives/reduction_GPI/HD163296_pipeline2/Reduced/ca
Pipeline Logfile:      /Users/earich/external_harddrives/reduction_GPI/HD163296_pipeline2/Reduced/lc

Status:  Last recipe done OK! Watching for new recipes but idle.
  Latest recipe:         /Users/earich/external_harddrives/reduction_GPI/HD163296_pipeline2/queue/S20140424S0320-S20140424S0396_basicpolcube_recipe.waiting.xml
  Latest Input FITS:   32/32, S20140424S0396.fits
  Current action:      --
  Latest saved file: /Users/earich/external_harddrives/reduction_GPI/HD163296_pipeline2/Reduced/1907
                                                   Current Recipe Completion Status:

                                                  Current FITS File Completion Status:

Pipeline Log Messages:
 18:02:04.637     File output to: /Users/earich/external_harddrives/reduction_GPI/HD163296_pip
 18:02:05.775          Primitive completed: Measure Satellite Spot Flux in Polarimetry
 18:02:06.407        Reduction successful: /Users/earich/external_harddrives/reduction_GPI/HD1
 18:02:06.708  Recipe Complete!
 18:02:07.243  Done with /Users/earich/external_harddrives/reduction_GPI/HD163296_pipeline2/qu

Processed Recipes:
 /Users/earich/external_harddrives/reduction_GPI/HD163296_pipeline2/queue/S20140424S0320-S2014
```

Now you are ready to operate the pipeline. Type python3 pipeline.py into the terminal in the same folder that pipeline.py is located. Note that you will want to use a lot of keywords to change how you reduce the data. An example can be found below:

python3 pipeline.py '/Users/earich/external_harddrives/reduction_GPI/HD163296_pipeline2/Raw/' '0-x_off="0"' '0-y_off="0"' '0-x0="148"' '0-y0="145"' '0-search_window="5"'

If you want to skip over the podc file reduction step (because it has already been completed), include '1-"stokes"' as a keyword in the execution of pipeline.py

Changes to the python pipeline (pypline)

20190709: edited the pipeline so that instead of finding all of the ".fits" files it needs from a directory, it reads the list of files (in the order that you want) from a '.lst' file. Currently the file must be named 'files.lst' and must be located in the Raw files directory, but this will change in the future.

20190709: The pypline execution is now saved to a log file (called: 'logfile_pipeline.log'). It will also save the output directory so the pypeline can be restarted without having to execute the podc reduction part of the IDL package.

20190709: Added the ability to skip over the podc reduction step. Include '1-"stokes"' when executing the pypline and it will read the output directory from the pypeline log file.

20190710: Removes all spaces from object names in the .fits header. This was causing an issue finding files. Done under getfitskeywords only when the keyword ‘OBJECT’ is used.

20190711: Made a copy of the pipeline under GPI_pipeline_v1, which is a good stable version. The GPI_pipeline I am editing is GPI_pipeline.

20190712: Added a centering check such that it outputs a plot of the centers. This will be a quick by-eye look if the centering assumption is good. It produces a *_centers.png file for every podc frame. That podc frame is plotted in intensity where a white circle marks the center of that frame and white x’s mark the centers of the other frames in its set.

20190712: Changed the WAIT function such that it can look for the last file that should be created. If it hasn’t been created yet, it will wait another few minutes and check back. Done with a try, except function.

20190714: Changed the creation to pdoc routine and cut it in half. This is order to allow to mask out the secondary stars in binaries and find the true centers.Currently it runs but does not save the files with the proper ending of their names after the centering routine is run (“*_pdoc.fits”).

20190716: Flexure check. Plots the flexure over the raw image to ensure that the polspot positions are correct. Functions are in the plot.py file under gpi_analysis. The images labeled as “*_flexure.png” are saved in the Reducted directory. If the white and red x’s are not on the spots, then the flexure is wrong and needs to be corrected. Ensure that the spots are pared up correctly, the pair in the middle of the image are extremely week in flux and should be paired together.

20190716: Wait function has been further modified such that it try’s to see if the last file exists rather than trying to open the file. Seeing if the file exists will hopefully avoid the issue of corrupting the file.

20190718: Added flux change measuring function. This should be sensitive to changes in flux AND changes in AO performance. This is accomplished by subtracting two frames in adjacent cycles and at the same waveplate rotation angle, then dividing by the later frame. The result is plotted. Also, if the keyword Bcent is provided, it will also mask over the companion. This feature will later be used for masking when finding the center. The calculated residual is currently written on the image, but should probably be saved into the header in the future.

20190718: Added “Bcent” keyword (eg. '1-Bcent="(181.5,154.0)"') which provides the location of the binary in pixel space in the first frame given.

ToDo’s

- Estimation of Errors. Errors are going to be propagated by bootstrapping the individual stokes frames together. Though large, all of the bootstrap files will be saved for later analysis.
- Integrate the flux calibration step into the pipeline.
- Record more documentation of what exactly the pipeline is doing to the data so these steps can easily be replicated in the future.
- Centering algorithm needs to be implemented. Cannot currently handle binary stars, eg. HD 50138. Need to reduce FU Ori stars to replicate Anna's Results in her 2019 paper. Most likely mask out the binary star and any extra waffle pattern.
- Write a test for the centering algorithm. Preform PSF fitting on the companion on the rotated (North) frames to look for estimates on the accuracy of the centering. Use VHS 1256 PSF centering program as a starting place.