M2FSreduce Step-by-step Tutorial

1) Checkout the github repository:

```
git checkout https://github.com/akremin/M2FSreduce.git
```

2) Download the sample data using a browser or wget:

```
wget https://drive.google.com/file/d/1pryyUpEYfWiTjRLG59Ur-zd99-cngOxJ/view?usp=sharing
```

3) Untar the sample data:

```
tar -xzvf M2FSreduce_example.tar.gz
```

4) Review the Readme file on the github webpage or in the github repository you checked out.

5) Move into the repository directory:

```
cd M2FSreduce
```

6) Edit the pipeline configuration file in the configs subdirectory. Relevant keywords you need to change:

```
[General]:
raw_data_path    = /path/to/M2FSreduce_example/raw_data
path_to_masks    = /path/to/M2FSreduce_example/m2fs_valadded_data
```

These should give the paths to the downloaded data (i.e. change /path/to/M2FSreduce_example to the proper path). The last subdirectories (raw_data and m2fs_valadded_data) should remain the same if you did not rename the downloaded directories.

No other edits should be required to run the code. All remaining defaults and filestructure settings are relative to the code directory and those specified above, and all subdirectories and filenames in the example obey the 'standard' formats defined within the io configuration file.

7) The pipeline configuration is setup to run all the steps, excluding the generation of a merged target list with targeting information. These steps and the m-t-l creation is defined in the pipeline configuration file by these relevant parameters:

```
 [STEPS]
bias       = True  # Performs the bias subtraction
stitch     = True  # Stitches the 4 opamps together.
remove_crs = True  # Removes the cosmic rays from the images
apcut      = True  # Cuts out each 2d spectrum and reduces them to 1d
wavecalib  = True # Two step process finding rough wavelength calibrations followed by a user-input step for final calibration
flatten    = True # Corrects for fiber throughput differences
skysub     = True # Removes skylines from the spectra, also generates a mask of locations
combine    = True # Combines all exposures of an object's spectra, taking account of masked regions
zfit       = True # Fit the spectra to a redshift. Currently the range is defined from 0.1 to 0.5.


[PIPE_OPTIONS]
make_mtl    = False # If true, will look for field file and combine with header information to create a merged-target-list
make_mtlz   = False # If true will combine the mtl with additional header information and redshift results (only useful after redshift fitting).
```

Other relevant settings you may wish to set:

```
[PIPE_OPTIONS]
single_core = False # If true, will run on a single core. Otherwise it will run with two cores.
show_plots  = False # Display summary plots as it runs. Note the pipeline stops while the images are being displayed.
save_plots  = True  # Save the summary plots to the value added data directory.
```

Notes: You can run with all set to True, it will wait at the human-input stage until you proceed. You may also run any sequence of "True" steps, so long as all previous steps have been successfully run in the past. Don't skip intermediate steps, e.g. don't run with bias=True, stitch=False, and remove_crs=True. The newly bias subtracted images won't be stitched, and the remove_crs will operate on the old stitched images that are sitting on disk or crash the pipeline because the stitched files are loaded into the pipeline.

8) To run the pipeline, simply ensure you are in the M2FSreduce directory and run:

```
python3 quickreduce.py
```

which assumes our set of defaults, or equally:

```
python3 quickreduce.py -m A09 -p ./config/pipeline.ini
```

9) This will take O(10's) minutes to run before you are prompted for input on the fine calibrations. Output will be continuously printed to the screen to inform you of the reduction steps being performed as well as performance information. After the calibration step in which you must provide input, the code will run again for O(10's) of minutes until it terminates after redshift fitting and generating the mtlz.

10) Assuming you are using all default settings, the summary plots can be viewed at any time from:

```
/path/to/M2FSreduce_example/m2fs_valadded_data/${mask_name}/summary_plots
```

11) If you would like a merged target list with target and fiber information, you can rerun the pipeline with all steps set to False, but make_mtl and make_mtlz set to True:

```
[PIPE_OPTIONS]
make_mtl    = True # If true, will look for field file and combine with header information to create a merged-target-list
make_mtlz   = True # If true will combine the mtl with additional header information and redshift results (only useful after redshift fitting).
```

12) Again assuming defaults, the merged target list and mtlz files can be found at:

```
/path/to/M2FSreduce_example/m2fs_valadded_data/catalogs/merged_target_lists
```

13) A fits file with the final spectra and masks are in:

```
/path/to/M2FSreduce_example/m2fs_valadded_data/${mask_name}/final_oned
```

And the redshift tables are in:

```
/path/to/M2FSreduce_example/m2fs_valadded_data/${mask_name}/zfits
```

Alternative approach: My typical multi-step use case.

7) Because the wavelength calibration requires some human-intervention, I typically do the reduction in three pieces: pre-wavecalib steps (automated), wavelength calibration (human input), and finally post-calibration steps through redshift fitting (automated).

```
[STEPS]
bias       = True
stitch     = True
remove_crs = True
apcut      = True
wavecalib  = False
flatten    = False
skysub     = False
combine    = False
zfit       = False

[PIPE_OPTIONS]
make_mtl    = False
make_mtlz   = False
```

8) The pipeline configuration is setup to run the initial four automated steps: bias subtraction, opamp stitching, cosmic ray removal, and aperture cutting. The results are saved at each step and the pipeline will terminate after the final step.

Run the first four steps from the M2FSreduce directory:

```
python3 quickreduce.py
```

which assumes our set of defaults, or equally:

```
python3 quickreduce.py -m A09 -p ./config/pipeline.ini
```

9) Following this, the first four steps in the pipeline.ini file can be set to False and the wavecalib step can be set to True.

```
[STEPS]
bias       = False
stitch     = False
remove_crs = False
apcut      = False
wavecalib  = True
flatten    = False
skysub     = False
combine    = False
zfit       = False

[PIPE_OPTIONS]
make_mtl    = False
make_mtlz   = False
```

10) Run the wavelength calibration step from the M2FSreduce directory:

```
python3 quickreduce.py
```

or equally:

```
python3 quickreduce.py -m A09 -p ./config/pipeline.ini
```

11) Following this, the wavecalib step in the pipeline.ini file can be set to False and the the final four steps can be set to True. If you want a summary file with targeting information using the merged target list generators, set the make_mtl and make_mtlz flags to True.

```
[STEPS]
bias       = False
stitch     = False
remove_crs = False
apcut      = False
wavecalib  = False
flatten    = True
skysub     = True
combine    = True
zfit       = True

[PIPE_OPTIONS]
make_mtl    = True
make_mtlz   = True
```

12) Run the flatten, skysub, combine, and zfit steps from the M2FSreduce directory:

```
python3 quickreduce.py
```

or equally:

```
python3 quickreduce.py -m A09 -p ./config/pipeline.ini
```

13) All files can be viewed in the locations described in the original tutorial steps above.