NeoNarval to Narval converter

This script extracts spectra from NeoNarval *.fits* files and outputs them as *.s* files in the format of Narval. The converter is written in python3, which can be downloaded from https://www.python.org/downloads/ .

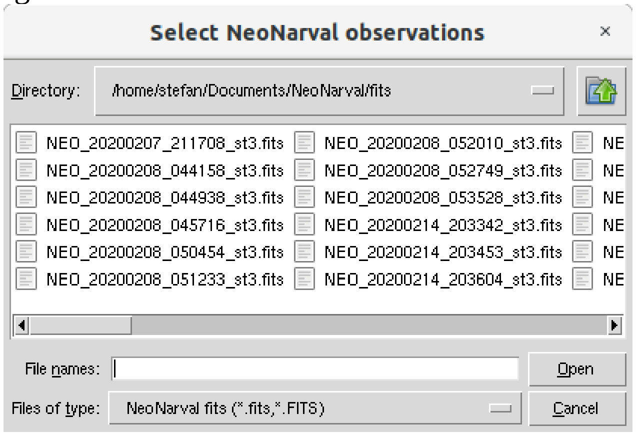
To use the program, please copy the source *.py* file to the directory where your observations are.

1. General use

Open a terminal (command prompt for Windows), navigate to your working directory and type

```
python3 neonarval-to-narval.py
```

You should see a file dialog like this one*:


Select NeoNarval observations

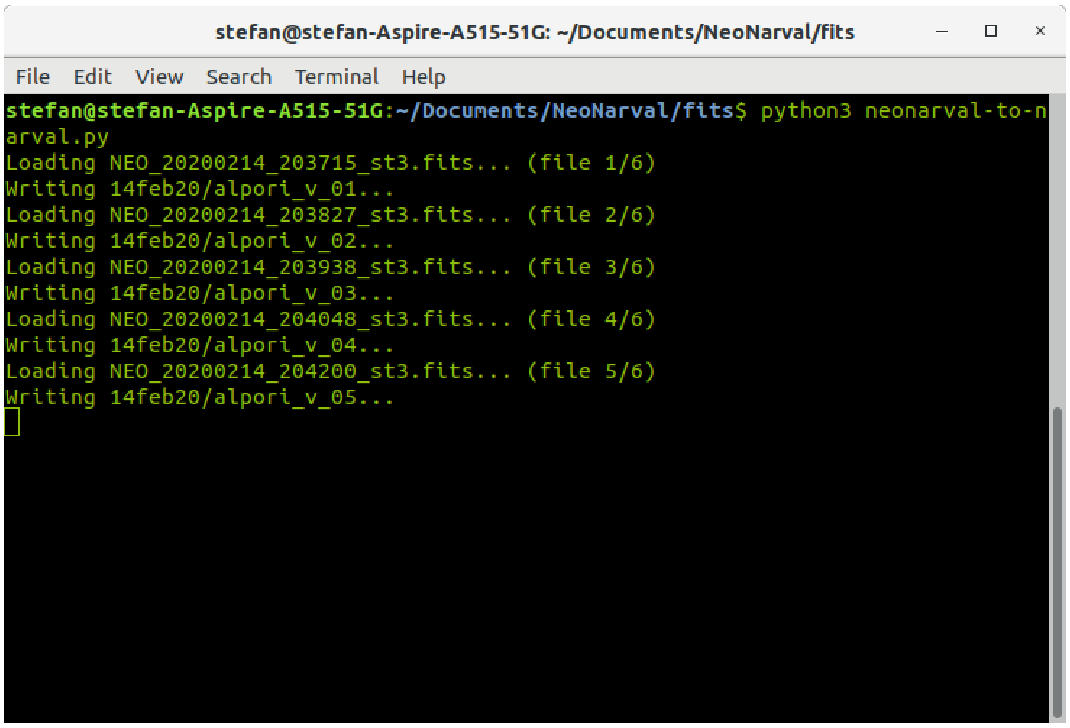
Directory: /home/stefan/Documents/NeoNarval/fits

```
NEO_20200207_211708_st3.fits  NEO_20200208_052010_st3.fits  NE
NEO_20200208_044158_st3.fits  NEO_20200208_052749_st3.fits  NE
NEO_20200208_044938_st3.fits  NEO_20200208_053528_st3.fits  NE
NEO_20200208_045716_st3.fits  NEO_20200214_203342_st3.fits  NE
NEO_20200208_050454_st3.fits  NEO_20200214_203453_st3.fits  NE
NEO_20200208_051233_st3.fits  NEO_20200214_203604_st3.fits  NE
```

File names:

Files of type: NeoNarval fits (*.fits,*.FITS)

You can select one or more *.fits* files for conversion. When you are done, press **Open**. The conversion process will now start and you should see information in the terminal window:


stefan@stefan-Aspire-A515-51G: ~/Documents/NeoNarval/fits
File Edit View Search Terminal Help

```
stefan@stefan-Aspire-A515-51G:~/Documents/NeoNarval/fits$ python3 neonarval-to-n
arval.py
Loading NEO_20200214_203715_st3.fits... (file 1/6)
Writing 14feb20/alpori_v_01...
Loading NEO_20200214_203827_st3.fits... (file 2/6)
Writing 14feb20/alpori_v_02...
Loading NEO_20200214_203938_st3.fits... (file 3/6)
Writing 14feb20/alpori_v_03...
Loading NEO_20200214_204048_st3.fits... (file 4/6)
Writing 14feb20/alpori_v_04...
Loading NEO_20200214_204200_st3.fits... (file 5/6)
Writing 14feb20/alpori_v_05...
```

The program will provide information for each loaded file. Note that the observations will be automatically placed into directories according to the night of observation. The names of the output files will include the object name as written in the *.fits* header, the Stokes parameter (U/Q/V) and the sequence number**. For each input *.fits* file, two output ASCII files will be generated:

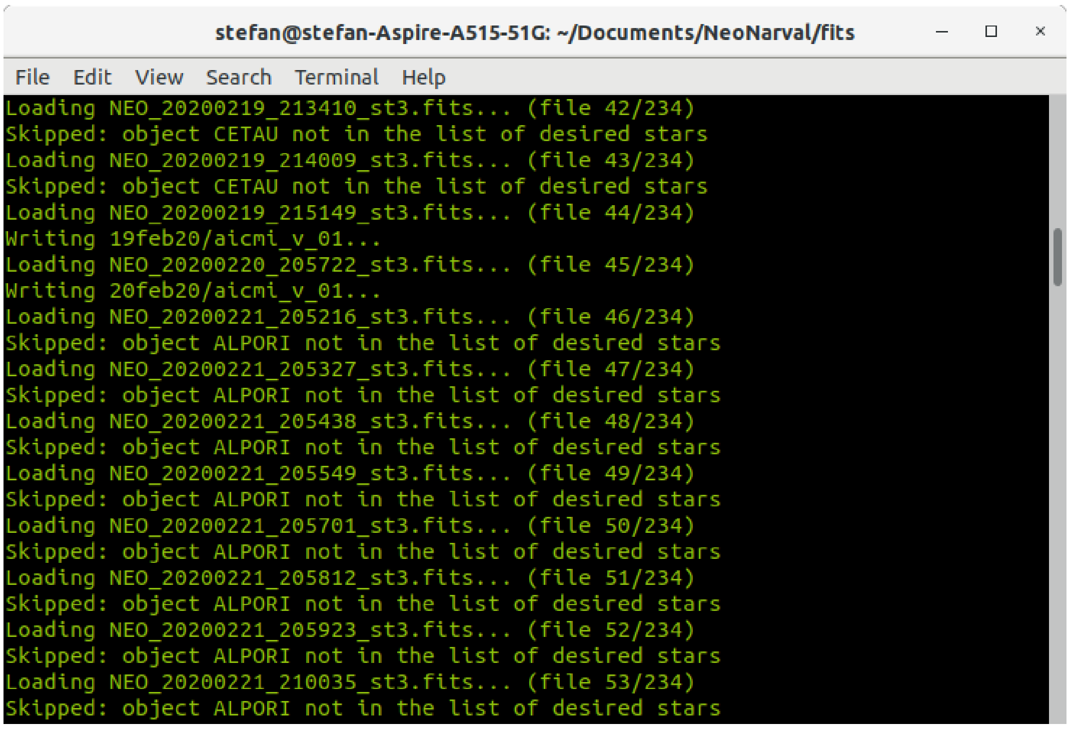
- a *.s* file, containing the extracted spectrum in Narval format;
- a *.header file*, containing all the information from the *.fits* header.

2. Working with selected stars only

In case you would like to convert only observations of certain stars, you can also specify your object names of interest when running the code. Let us say we have downloaded all available data for a certain season, but are interested only in observations of one star, e.g. AI CMi. We don't know which *.fits* files correspond to observations of this star and don't want to spend unnecessary time converting all the available data. In this case, we can start the program as before, but this time typing the name of the star (or stars) we want to process in the terminal like this:

```
python3 neonarval-to-narval.py aicmi
```

This will lead to the following result:


stefan@stefan-Aspire-A515-51G: ~/Documents/NeoNarval/fits
File Edit View Search Terminal Help

```
Loading NEO_20200219_213410_st3.fits... (file 42/234)
Skipped: object CETAU not in the list of desired stars
Loading NEO_20200219_214009_st3.fits... (file 43/234)
Skipped: object CETAU not in the list of desired stars
Loading NEO_20200219_215149_st3.fits... (file 44/234)
Writing 19feb20/aicmi_v_01...
Loading NEO_20200220_205722_st3.fits... (file 45/234)
Writing 20feb20/aicmi_v_01...
Loading NEO_20200221_205216_st3.fits... (file 46/234)
Skipped: object ALPORI not in the list of desired stars
Loading NEO_20200221_205327_st3.fits... (file 47/234)
Skipped: object ALPORI not in the list of desired stars
Loading NEO_20200221_205438_st3.fits... (file 48/234)
Skipped: object ALPORI not in the list of desired stars
Loading NEO_20200221_205549_st3.fits... (file 49/234)
Skipped: object ALPORI not in the list of desired stars
Loading NEO_20200221_205701_st3.fits... (file 50/234)
Skipped: object ALPORI not in the list of desired stars
Loading NEO_20200221_205812_st3.fits... (file 51/234)
Skipped: object ALPORI not in the list of desired stars
Loading NEO_20200221_205923_st3.fits... (file 52/234)
Skipped: object ALPORI not in the list of desired stars
Loading NEO_20200221_210035_st3.fits... (file 53/234)
Skipped: object ALPORI not in the list of desired stars
```

As you can see from the information in the terminal, the code will only process observations of the star we specified and will skip all other objects. You can use this functionality for as many stars as you want simultaneously. For example, you can call the program like this:

```
python3 neonarval-to-narval.py aicmi alpori evlac chicyg bydra
```

Note that the object names must be written without intervals: **alpori** is good, but **alp ori** is not. The code will also ignore the dash (-) and underscore (_) symbols and is not sensitive to case: **alp_ori** is treated the same as **ALPORI**, **AlpOri** and **alp-ori**.

***** In case you get an error message when running the script that looks like this:

```
    Traceback (most recent call last):
    ModuleNotFoundError: No module named 'numpy'
```

this means that the mentioned python module is not installed on your computer. Normally you can solve this by typing in the terminal:

```
    pip3 install numpy
```

This will install the missing library. The example given here is for the **numpy** library, but you might also have this problem for other libraries, like **astropy**. In the unlikely event that the above solution does not work, you can also try (under linux only):

```
    sudo pip3 install numpy
```

****** The code calculates the sequence number by looking into the output directory and **not** from the input data, as the latter is practically impossible. This means that if you run the script more than once on the same observations, you may get a false numbering of sequences in the output directory.
For example, let us say that on 01 January 2021 NeoNarval obtained 6 Stokes V sequences of αOri. Running the script for the first time on these data will produce a directory named *01jan21* that contains files *alpori_v_01.s*, *alpori_v_02.s*, … *alpori_v_06.s*. **This will be alright**. However, if we run the script once more on the same *.fits* files, the code will look into the output directory *01jan21*, see that 6 Stokes V sequences of αOri already exist there and so it will save the converted observations once more as *alpori_v_07.s*, *alpori_v_08.s*, … *alpori_v_12.s*. This could mislead one to think that on 01jan21 there were 12 observational sequences of αOri in Stokes V. In fact, *alpori_v_07.s* will be the exact same file as *alpori_v_01.s*, *alpori_v_08.s* will be the same as *alpori_v_02.s* and so on.
To avoid any confusion, please take care when running the script more than once on the same input data. In case you suspect a doubling of sequences has occurred, refer to the date and time of observation that are stored in the *.header* file.