Draft version April 22, 2025
Typeset using LaTeX **default** style in AASTeX7

Emissions and Oscillations of V927 Her

Carter Garrett,[1] Eric Hintz,[1] Scott Call,[1] and Denise Stephens[1]

[1] *Brigham Young University, Department of Physics and Astronomy*

ABSTRACT

We provide an overview of the continued investigation of the He 1.0835 μm emission feature in the Delta Scuti class variable stars (dS^*) V927 Her. The star exhibits strange apparitions of emission features with seemingly P-Cygni-like profiles. To date, and to the authors' knowledge, this is the first detection of such an energetic shock front in dS^*.

Keywords: Delta Scuti variable stars, magnitudes, oscillatory periods, shock fronts, astroseismology, P-Cygni profiles.

1. INTRODUCTION

1.1. *Background*

Radially pulsating variable stars can exhibit emission features that are similar P-Cygni profiles. As shock fronts propagate in the variable star, they can instigate emission in different regions of the stellar atmosphere.

When a disturbance, such as a pulsation wave, travels faster than the local speed of sound in star, it generates a 'shock front.' This term describes a thin shell where a sudden increase in pressure, temperature, and density occurs.

As the shock front moves through the gas, it compresses and heats the material in its path. This path is referred to as a 'radiative wake.' The atomic material in the wake reaches excited energy levels, and radiatively cools during de-excitation.

The emitted photons in the radiative wake contribute to emission features that coincide with absorption lines, such as the Hα and He I lines observed in RR Lyrae stars and classical Cepheids (Gillet & Fokin (2014), Andrievsky, S. M. et al. (2023)). During different phases of oscillation, different 'apparitions' of emission features can be identified, each corresponding to different astroseismological processes.

These emission profiles are crucial for distinguishing between different types of variable stars, understanding mass loss, and modeling stellar atmosphere dynamics.

1.2. *Objectives*

In 2023, the first emission feature at the He 1.0835 micron line in dS^* was successfully identified, according to the diagnostics provided by Andrievsky, S. M. et al. (2023) as well as Gillet & Fokin (2014). 3 years of spectral data for all kinds of variable stars was analyzed based on the presence of the emission features corresponding to these shock fronts. The star V927 Herculis was the candidate that exhibited this feature most obviously. From the end of 2024 and continuing through the present, more spectral data has been acquired and the feature has been extensively characterized. This paper seeks to establish the *phase of pulsation* that is associated with this transient Helium emission, in order to compare it with the phases and physical phenomenon that have been identified by Gillet & Fokin (2014).

1.3. *Instrumentation*

Spectral data was obtained using the 3.5 m Triple Spec (TSPEC) Spectrograph at the Apache Point Observatory. TSPEC is a cross-dispersed, near-infrared spectrograph that provides simultaneous continuous wavelength coverage from 0.95-2.46 m in five spectral orders. It has a resolving power of 3500. We took 120 second exposures over the course of 3 hours on multiple nights, to cover the entire flux oscillation period. Our photometric data was obtained using the 8 inch telescope at the Orson Pratt Observatory. The 8 inch took 35 second exposures in the V filter, using an FLI ML8300 CCD.

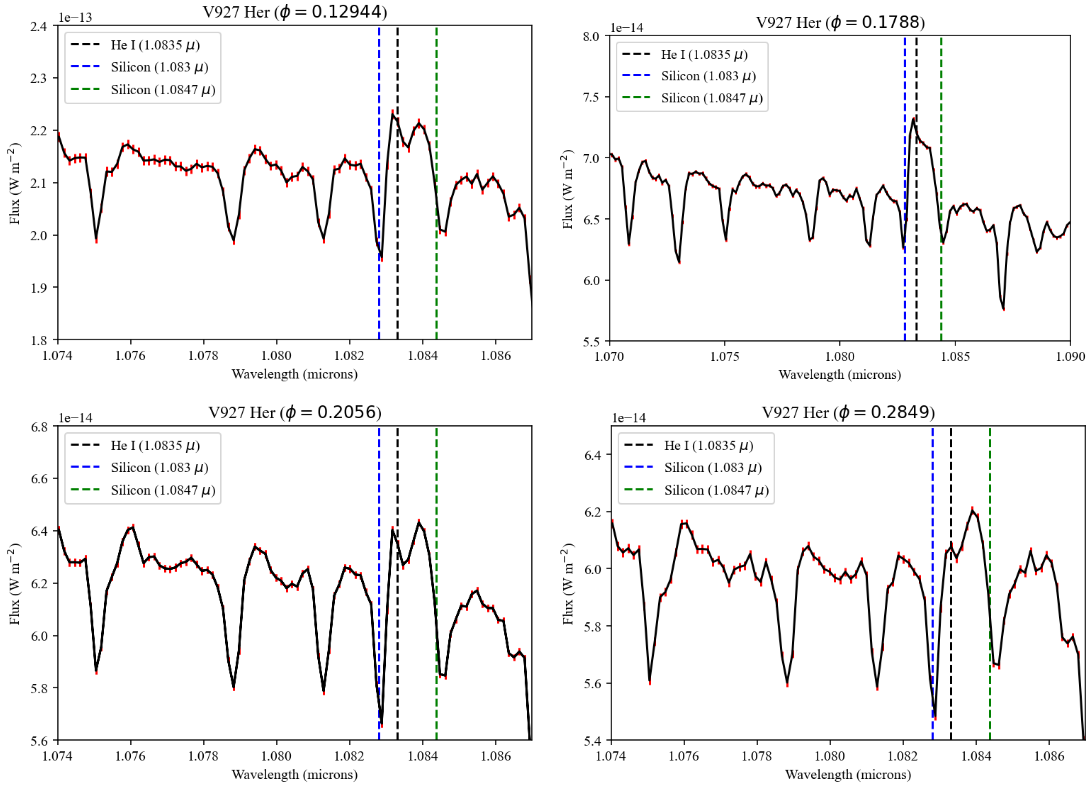

Figure 1. A sequence of spectra showing the evolution of the He 1.0835 μm emission feature. Note the splitting of the peak as well as the P - Cygni reminiscent features. ϕ indicates the phase, where it ranges from 0 to 1.

2. DATA AND ANALYSIS

2.1. *Data Processing*

Spectral data was processed using a GUI in IDL. Standard flat, dark, and bias calibration is applied to the data. Continuum curves are then fitted to the blackbody spectrum of the target star. Telluric effects are then corrected for. The multiple exposures are then merged and cleaned based on the desired separation between datasets.

No archival photometric data was available from the Orson Pratt Observatory (OPO) in Provo, Utah. However, 4 nights of data were collected. The star V927 Her was observable in the second half of the night. Photometric data was obtained using the 8 - inch telescope at the OPO. The data was binned on a 2x2 basis. 35 second exposures in the V - filter had header files and names processed in Python. IRAF (Tody (1996)) was then used to handle the dark, flat, and bias corrections. The data was plate solved using AstroImageJ (AIJ) and then multiple aperture photometry was conducted using the same software.

Plate solving proved to be a grueling and minimally successful process. The amount of time required to solve was incredibly high. Sensitivity to background sky values caused all sorts of problems due to the relative dimness of the targets. Moreover, the telescope had data readout errors. It seemed to alternate between the V and R filter between exposures, even though the files listed V - filter values. Only a few of the nights worth of data from the OPO were employed to find periods– in fact, we only used the OPO data to find known times of *maximum light*.

Frequency analysis and variable aperture photometry was conducted using the LightKurve pipeline, which mainstreams data collected by TESS. Multiple runs on V927 Her were available for frequency analysis.

2.2. *Analysis*

Firstly, spectral data was analyzed and plotted for V927 Her. This can be viewed in Figure 1. The feature clearly evolves in time, which was the case for each other night of data. The task was to determine where in phase each peak and split occurred. At this point in time, AIJ and data from OPO was being extremely uncooperative. It was decided that photometric data from TESS would be used to calculate the period of oscillation.

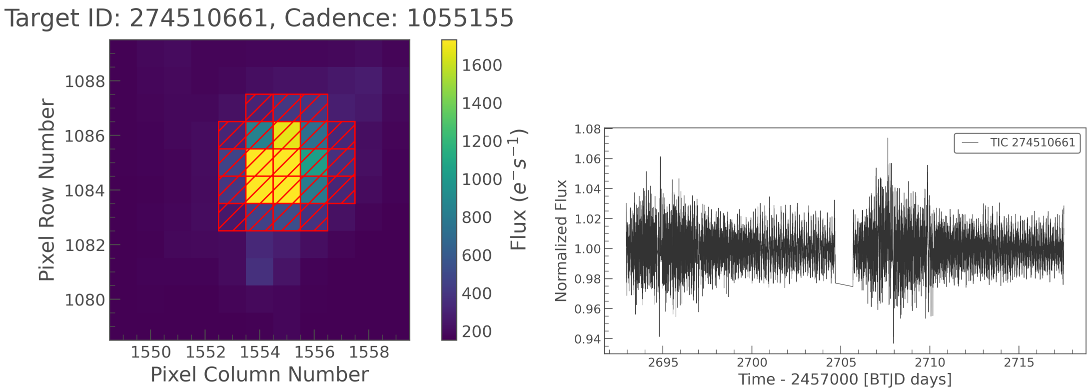

Figure 2. An example of a mask for photometry applied to a target pixel file in LightKurve. This was used to obtain normalized flux values for V927 Her across a large time span. Multiple observing runs were made that contain this amount of data on the star.

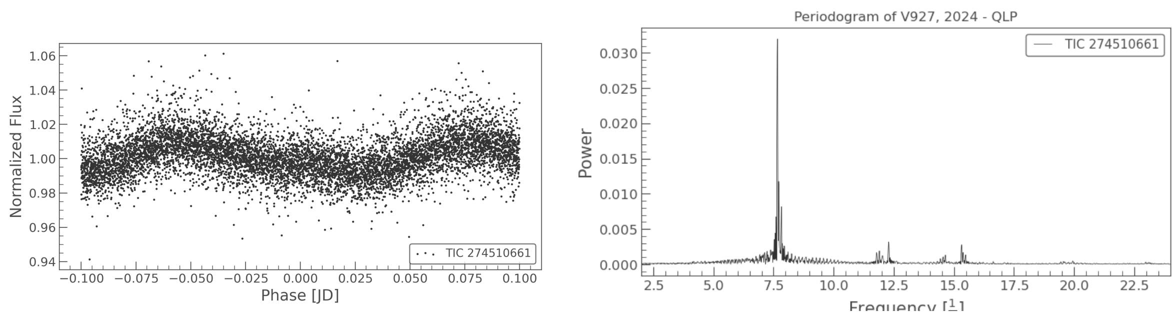

Figure 3. The folded light curve, which is then converted to a periodogram. For each observing run, a periodogram was produced and the dominant frequency used to calculate the average between each observation.

A very handy Python library has been developed for the TESS photometric pipeline. It is called LightKurve (Lightkurve Collaboration et al. (2018)). The process for obtaining periodograms for frequency is as follows:

1. Conduct a search for the target of choice in the TESS databases.
2. Download the target pixel files for custom aperture photometry.
3. Obtain flux values across a broad range of times for various observing runs.
4. Fold the date into manageable and analyzable light curves.
5. Convert each curve into periodograms, and obtain the dominante frequency present in the spectrum.

Figure 2 shows the precision of the custom apertures that LightKurve employs to conduct the photometry. For years of data, brightness fluctuation charts are obtained, also as in Figure 2.

Once these were obtained, the light curve was folded, and analyzed for frequencies. Over 20 periodograms were obtained for V927 Her, as in Figure 3. The most prominent period was acquired for each, and then brought to an average period. The period was calculated to be about 0.13 days, which is exactly consistent with the literature and all other references (Hintz & Garvin (2000)).

Though most of the 8 - inch telescope data was untrustworthy, some of it was usable for this project. Photometry in AIJ was conducted for a night of data on April 4th, and can be viewed in Figure 4. Once times of maximum light were obtained, these values were used in python to find the phase. The calculation is simple– it finds the difference in time between the time of the spectral observation and the maximum light. It then uses *modulo* (%) to find the *remainder* of periods at the time of observation. That corresponds to the phase at the time of observation. The code itself is as follows:

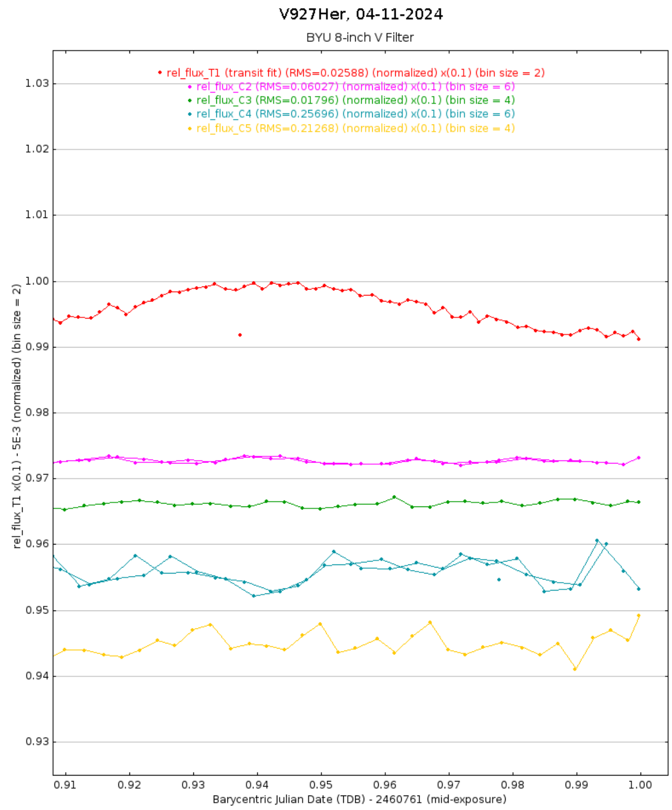

Figure 4. Lightcurve generated in AIJ for V927 Her on April 8th, 2025. This was used to find a recent and surefure time of maximum light. The red T1 line is the flux for V927 Her, at the top. Image stars are also displayed for comparison. Variations in those brightness values are likely due to the resettling of the telescope as it continues to adjust viewing angle.

```
# ...
maxlightOPO = 2460761.946121
period = 0.1304958090565437
phasecalc = ((maxlightOPO - bjd) % period) / period
# ...
```

The periods of each spectral observation can be found in Figure 1, in each title. They are $\phi = 0.12944, 0.1788, 0.2056,$ and 0.2849. Gillet & Fokin (2014) identify the Hα transient emission phenomena at around phases $\phi \approx 0.8, 0.93, 0.00,$ and 0.29. These values do not align with our calculated phases.

V927 Her is an F spectral type, and has an effective temperature of about 7000 K, according to Tian et al. (2023). In this case, we believe it corresponds to the shock front heating the infalling material in either the chromosphere or photosphere. The remission occurs blueshifted in the wing, giving a strange P-Cygni appearance. The feature is exceptionally strong. Gillet & Fokin (2014) propose that the redshift in the later phase of emission occurs during the detachment of the shockwave from the photosphere of the star.

The The P-Cygni contour is a result of the radiative wake of the shock wave as it travels outward and absorbs photons from the photosphere. For an extended envelope, the front speed is estimated using:

$$v_{shock} \approx \frac{\lambda_a - \lambda_e}{\lambda_0}$$

Where λ_a is minimum absorption feature. We obtained a value of around 110 km/s. This is a fairly fast value for a low mass star.

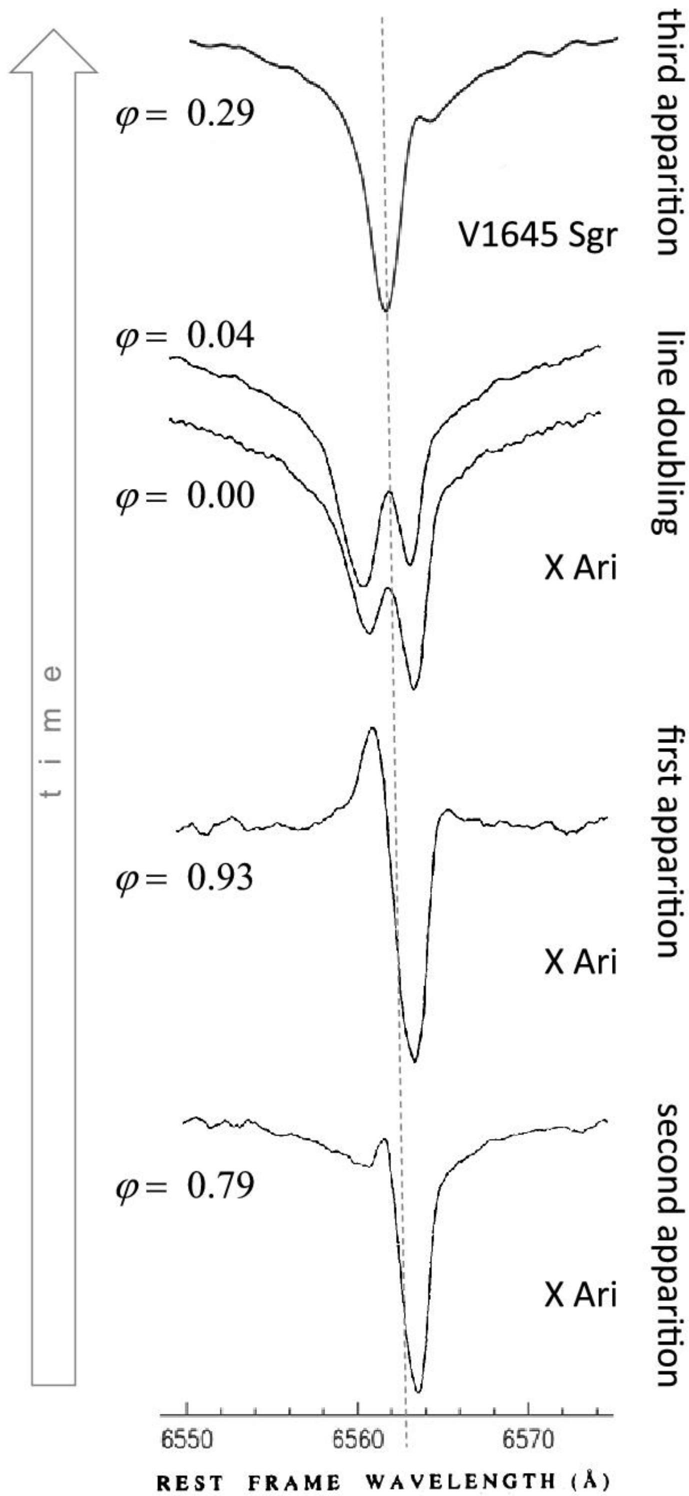

Figure 5. Adapted from Gillet & Fokin (2014). Shows the emission profiles of Hα at their respective phases of oscillation for classical Cepheids.

3. CONCLUSION

3.1. *Deductions*

We continued to observe V927 Her this past quarter. Thankfully, the feature showed continued evolution, which means it is likely associated with the shock front, and not a one - time event

However, it does not match with the Hα phases. Therefore, we conclude (preliminarily and cautiously) that it can be tracked as a connected but time-delayed phenomenon associated with the $H\alpha$ phases. Given that He has a higher ionization energy etc., it may be that He re-emits slower than the other ionized counterparts in the radiative wake. What's more, it may be indicative of Helium content in different regions of the star that ionizes.

The presence of these features is likely (though not affirmatively) a sign of driven mass loss in dS^*. These kinds of shock fronts are clearly energetic enough to ionize He. More work remains to be done to decidedly conclude if this is the case.

However, for all intents and purposes, we successfully accomplished the objective, which was to determine the phase of the He I apparitions and make initial inferences about the nature of the physical processes involved.

3.2. *Future Work*

V927 Her and other dS^* will be subject to continued scrutiny and observation. Stellar radial velocities and $\log(g)$ values will also be calculated for V927 Her, in order to obtain a more accurate picture of the dynamics of the photosphere and chromosphere. We look forward with great anticipation for future discoveries to be had.

4. CODE

This section is a kind of appendix for the code implemented during the procedure. It can be obtained on GitHub at the author's page.

4.1. *Light Curves*

```
# gets lightcurves.

import lightkurve as lk
import numpy as np
%matplotlib inline
import matplotlib.pyplot as plt

target1 = lk.search_targetpixelfile("V927 Her")[1].download()
target1.plot(aperture_mask='pipeline')
lc1 = target1.to_lightcurve(method = "pld", aperture_mask = 'pipeline').normalize().
                                                  remove_outliers().flatten()

period = lc1.to_periodogram("bls").period_at_max_power
lc1.fold(period).truncate(-0.1, .1).scatter()
```

```
# calculates average period based on TESS data and provides periodograms / frequency
                                                  spectra.

search2 = lk.search_lightcurve('V927 Her', mission = 'TESS')

sum = 0
itr = 0

for index, result in enumerate(search2):
    print(f"\n Processing: {index}")
    try:
        lc = result.download()

        if lc is not None:
            pg = lc.normalize().to_periodogram()
            sum += pg.period_at_max_power
            itr += 1
            print(f"Prominent Period: {pg.period_at_max_power}")

            ax = pg.plot()
            ax.set_xlim(2, 26)
            plt.title(f"Periodogram for {result.target}, ID: {result.dp_id}")
            #plt.savefig(f'freqplot{itr}.png')

            plt.show()
            plt.close()

        else:
            print(f"Could not download {result} ")

    except Exception as e:
        print(f"Error processing {result}: {e}")

if itr > 0:
    print(f"Finished. Avg Period = {sum / itr}")
else:
    print(f"Finished, no curves processed though.")
```

4.2. *Phase Calculation*

```
import matplotlib.pyplot as plt
import os
from astropy.io import fits
from astropy.time import Time
```

```
from astropy.coordinates import EarthLocation, SkyCoord
import astropy.units as u
import lightkurve as lk

# working directory
print(os.getcwd())

# readability
print("-------------------APO Calculations-------------------")

# open .fits file
with fits.open('V927Her.fmerged.fits') as hdul:
    header = hdul[0].header

# get dates and times from .fits file
obs_start_time = header['SRT_TIME']
obs_end_time = header['END_TIME']
obs_date = header['SRT_DATE']

print(f"APO Observation Start: {obs_start_time}")
print(f"APO Observation Date: {obs_date}")

observation_time = obs_date + 'T' + obs_start_time # formatting
print(f"Formatted Observation Time: {observation_time}")

# time to Julian
t = Time(observation_time, format='isot', scale='utc')
julian_date = t.jd
print(f"APO Observation Julian Date: {julian_date}")

# setting location (Apache Point Observatory coordinates)
location = EarthLocation.from_geodetic(lon=-105.820417 * u.deg, lat=32.780361 * u.deg,
                                        height=2788 * u.m)
print(f"APO Earth Location Coordinates: {location}")

# coordinates of target
star_coord = SkyCoord(ra=17.9985124872 * u.deg, dec=35.856677532 * u.deg, frame='icrs')
#print(f"Star Coordinates: {star_coord}")

# to barycentric julian date, might be off.
bjd = t.light_travel_time(star_coord, location=location).jd + julian_date
print(f"Barycentric Julian Date of APO Observation: {bjd}")
```

```
#OUR DATA, 1st phase

#import
print("-------------------OPO CALCULATIONS-------------------")

# open file

with fits.open('V927Her1.merged.fits') as hdul:
    header = hdul[0].header

# get dates and times from .fits file

obs_start_time = header['TIME']
obs_date = header['DATE']

print(f"APO Observation Start: {obs_start_time}")
print(f"APO Observation Date: {obs_date}")

observation_time = obs_date + 'T' + obs_start_time # formatting
print(f"Formatted Observation Time: {observation_time}")

# time to Julian
t = Time(observation_time, format='isot', scale='utc')
julian_date = t.jd
print(f"APO Observation Julian Date: {julian_date}")
```

```
# setting location (Apache Point Observatory coordinates)
location = EarthLocation.from_geodetic(lon=-105.820417 * u.deg, lat=32.780361 * u.deg,
                                       height=2788 * u.m)
print(f"APO Earth Location Coordinates: {location}")

# coordinates of target
star_coord = SkyCoord(ra=17.9985124872 * u.deg, dec=35.856677532 * u.deg, frame='icrs')
#print(f"Star Coordinates: {star_coord}")

# to barycentric julian date, might be off.
bjd = t.light_travel_time(star_coord, location=location).jd + julian_date
print(f"Barycentric Julian Date of APO Observation: {bjd}")

maxlightOPO = 2460761.946121
period = 0.1304958090565437
phasecalc = ((maxlightOPO - bjd) % period) / period

print(phasecalc)
```

4.3. *Filename Editor*

```
# my very handy filename editor for IRAF preprocessed data.
# preamble

import numpy as np
%matplotlib inline
from matplotlib import pyplot as plt
from matplotlib.colors import LogNorm

from astropy.visualization import hist
from astropy.stats import histogram

from astropy.io import fits

import glob #for the fits
from pathlib import Path
import os

from astropy import units as u
from ccdproc import Combiner, CCDData
```

```
# creates a new txt file with updated names for each call. also adds the IRAF required '
                                       imagetyp' header section with correct
                                       name.
# double check the detected headers (under # new suffix) match what your files will
                                       contain, or however the observer
                                       formatted them.

# make a list of files

def list_files(folder, extension):
    folder = Path(folder)
    if not folder.exists():
        print(f"Directory {folder} not found")
        return[]

    # gets rid of whitespace for iraf.

    print(f"Working in: {folder}")
    i = 0

    # reading names
    for item in folder.rglob(f"*{extension}"):

        if item.is_file():
```

```
                new_name = item.name.replace(" ", "")

                #renaming
                if new_name != item.name:
                    new_path = item.with_name(new_name)
                    os.rename(item, new_path)
                    #print(f"Renamed {item} to {new_path}")
                    i += 1

    print(f"Removed whitespace from {i} filenames in {folder}")
    return [file.name for file in folder.iterdir() if file.is_file() and file.suffix ==
                                                    extension]

# renames all the files

def rename_fits_files(file_list, folder, output_log = "current_files.txt"):

    # setting paths
    folder = Path(folder)
    log_path = folder / output_log

    # checking / making directories
    if not folder.exists():
        print(f"Directory {folder} does not exist. Creating now.")
        folder.mkdir(parents=True, exist_ok=True)

    print(f"Working in: {folder} \n")

    # going through the names

    for file_name in file_list:

        file_path = folder / file_name

        try:
            with fits.open(file_path, mode = "update") as hdul:
                header = hdul[0].header
                file_type = header.get("IMAGETYP", "").lower()

                # picking a file naming suffix
                if "bias" in file_type or "zero" in file_type:
                    suffix = ".bias"
                    header["IMAGETYP"] = "zero"
                elif "dark" in file_type:
                    suffix = ".dark"
                    header["IMAGETYP"] = "dark"
                elif "flat" in file_type:
                    suffix = ".flat"
                    header["IMAGETYP"] = "flat"
                elif "object" in file_type:
                    suffix = ".target"
                elif "light" in file_type:
                    suffix = ".target"
                else:
                    suffix = ".unknown"

                hdul.flush()

            # making new file name
            new_name = f"{file_path.stem}{suffix}.fits"
            new_path = folder / new_name

            # renaming
            if any(x.lower() in file_name.lower() for x in ["bias", "flat", "dark", "
                                                        target", "unknown"]):
                print("For file " + file_name + ": No changes, file type / extension
                                                            already present. \n")
            else:
```

```
                os.rename(file_path, new_path)
                print(f"Renamed {file_name} -> {new_name}")

        except Exception as e:
            print(f"Error processing {file_name}: {e}")
            print(f"Might want to inspect. \n")

    global current_files
    current_files = list_files(folder, ".fits")

    with open(log_path, "w") as log_file:
        for file in current_files:
            log_file.write(file + "\n")

    print(f"Updated list of file names in: {output_log}")
```

REFERENCES

Andrievsky, S. M., Kovtyukh, V. V., & Korotin, S. A. 2023, A&A, 671, A27, doi: 10.1051/0004-6361/202245384

Carroll, B. W., & Ostlie, D. A. 2017, An Introduction to Modern Astrophysics, 2nd edn. (Cambridge University Press)

Gaia Collaboration. 2020, VizieR Online Data Catalog: Gaia EDR3 (Gaia Collaboration, 2020), VizieR On-line Data Catalog: I/350. Originally published in: 2021A&A...649A...1G, doi: 10.26093/cds/vizier.1350

Gillet, D., & Fokin, A. B. 2014, A&A, 565, A73, doi: 10.1051/0004-6361/201322938

Hintz, E. G., & Garvin, M. B. 2000, Information Bulletin on Variable Stars, 4869, 1

Lightkurve Collaboration, Cardoso, J. V. d. M., Hedges, C., et al. 2018, Lightkurve: Kepler and TESS time series analysis in Python, Astrophysics Source Code Library. http://ascl.net/1812.013

Tian, X.-m., Wang, Z.-h., Zhu, L.-y., & Yang, X.-L. 2023, ApJS, 266, 14, doi: 10.3847/1538-4365/acc4b5

Tody, D. 1996, IRAF in the Nineties, doi.org/10.5281/zenodo.15187343