TTP

0. Clone the GitHub repository:

```
git clone https://github.com/kateivshina/transit_timing_protocols.git
```

Go to the directory of the repository:

```
cd transit_timing_protocols/
```

1. In tess_params.txt file, specify the parameters of the planetary system as well as where to store the output data.

--mission=TESS # name of the mission
--planet=WASP 4b # name of the planet
--cadence=2 # cadence of data (2 or 30 mins)
--radius=0.12 # planet radius in stellar radii
--semi_major_axis=5.299 # orbit's semi-major axis in stellar radii
--b=0.15 # impact parameter
--period=1.338231429 # planet's period
--logg=4.484 # host star's surface gravity
--Teff=5436 # host star's effective temperature
--Z=-0.05 # host star'smetallicity
--parent_dir=/Users/kate/Documents/usrp/TTP
--path_to_data_file=/Users/kate/Documents/usrp/TTP/lc/wasp4_lc.fits
--refolded=False

2. To preprocess the data, run in the terminal:

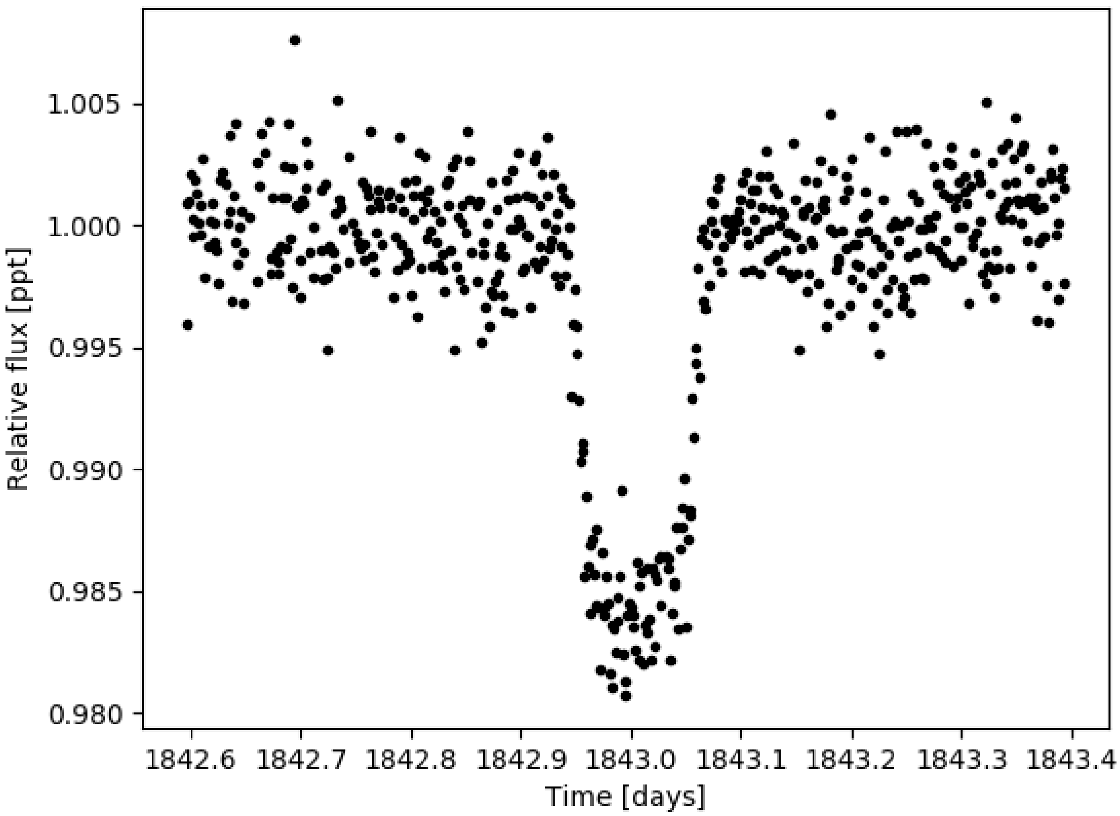
```
python3 process_tess_data.py @tess_params.txt
```

It will create the following files in the output directory:

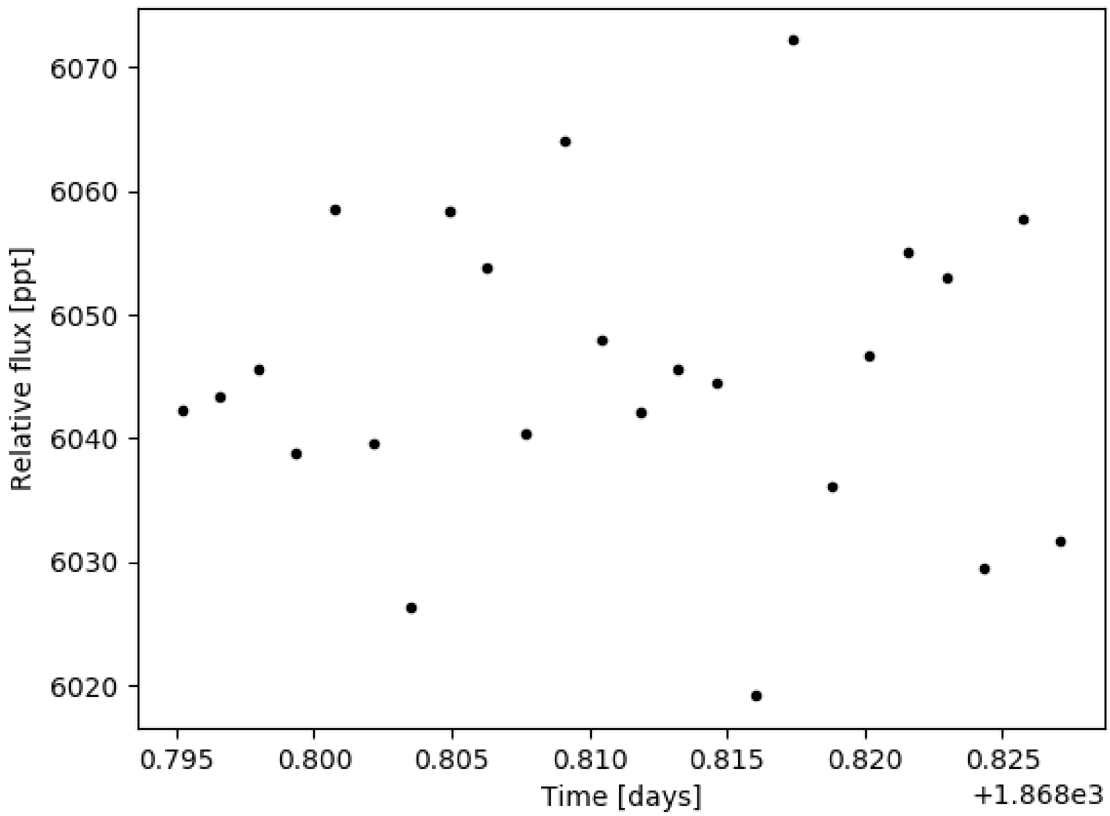
- *~/transit/times.txt* - transit time data
- *~/transit/flux.txt* - transit flux data
- *~/transit/time_folded.txt* - transit mask
- *~/transit_masked/folded_time_masked.txt* - mask for out of transit data
- *~/transit_masked/time_masked.txt* - time for out of transit data
- *~/transit_masked/flux_masked.txt* - flux for out of transit data
- ~/transit/individual_flux_array.npy - separate arrays of fluxes for each transit
- ~/transit/individual_time_array.npy - separate arrays of times for each transit
- ~/transit/individual_time_folded_array.npy - separate arrays of transit masks for each transit
- ~/*transit_masked*/individual_flux_array.npy - separate arrays of fluxes for each out of transit data
- ~/*transit_masked*/individual_time_array.npy - separate arrays of times for each out of transit data
- ~/*transit_masked*/individual_time_folded_array.npy - separate arrays of masks for each out of transit data
- *~/transit/corrected_flux.npy* - intial de-trended flux (separate array for each transit)
- *~/transit/stds.npy* - standard deviation of out-of-transit points outside of each transit
- *~/transit/coeffs.txt* - linear model coefficients used to de-trend the data

This script will also generate figures of transits in the ~/figures/individual_transits_figures folder as well as figures of de-trended transits in the ~/figures/transits_after_detrending folder.

Example:



3. In the *~/figures/individual_transits_figures folder, check the found transits and note indices of events that were accidentally selected as transits, e.g.*



Put these indices into ~/clean.txt file as the --delete_arrs argument.

Example:

--mission=TESS
--planet=WASP 4b
--parent_dir=/Users/kate/Documents/usrp/TTP
--delete_arrs=0 10

Then, to remove these events, run

```
python3 remove_non_transits.py @clean.txt
```

It will create the following files:

- ~/transit/individual_flux_array_clean.npy
- ~/transit/individual_time_array_clean.npy
- ~/transit/individual_time_folded_array_clean.npy
- ~/transit/corrected_flux_clean.npy
- ~/transit/stds_clean.npy

4. Run MCMC on phase-folded data (assuming no timing variations):

```
python3 mcmc_a.py @tess_params.txt
```

This will output the following files:

- ~/figures/MCMCfit.png - a figure of the folded light curve and the transit model
- ~/tranit/theta_max.txt - a file containing the found *rp*, *a*, *b*, *u1*, *u2* parameters

5. Run MCMC on each individual transit:

```
python3 mcmc_b.py @tess_params.txt
```

This will output *~/transit/t0_k_c.txt file* that contains *{t0, k, b}* for each transit where

t0 - mid-transit time

k and *b* - de-trending polynomial coefficients.

6. Run the following script to refold the data with the newly derived mid-transit times:

```
python3 refolding.py @tess_params.txt
```

It outputs the following files:

~/transit/corrected_flux_refolded.npy
~/transit/stds_refolded.npy
~/transit//individual_flux_array_clean_refolded.npy
~/transit/individual_time_array_clean_refolded.npy
~/transit/individual_time_folded_array_clean_refolded.npy

7. Change **--refolded=False** to **--refolded=True** in the *tess_params.txt* file to indicate that you're analyzing refolded data.

Run MCMC on phase-folded data again:

```
python3 mcmc_a.py @tess_params.txt
```

This will output the following files:

- ~/figures/MCMCfit.png - a figure of the folded light curve and the transit model
- ~/figures/theta_max.txt - a file containing the found *rp*, *a*, *b*, *u1*, *u2* parameters

8. Run MCMC on each individual transit:

```
python3 mcmc_b.py @tess_params.txt
```

This will output *~/transit/t0_k_c.txt file* that contains *{t0, k, b}* for each transit where

t0 - mid-transit time

k and *b* - de-trending polynomial coefficients.

The script will also produce corner plots for each transit in the ~/transit/corner_plots folder.

9. To plot O-C (constant period model), run

```
python3 o_c.py @tess_params.txt
```

10. Combine the extracted t0s and their uncertainties with the previously known t0s into file o_c_comb.txt. To plot O-C, run

```
python3 o_c_combo.py @tess_params.txt
```

11. You can use the following script to sample planets from the NASA Exoplanet Archive based on the criteria involving planet's mass, orbit's semi-major axis, and stellar star magnitude.

```
python3 select_planets.py
```