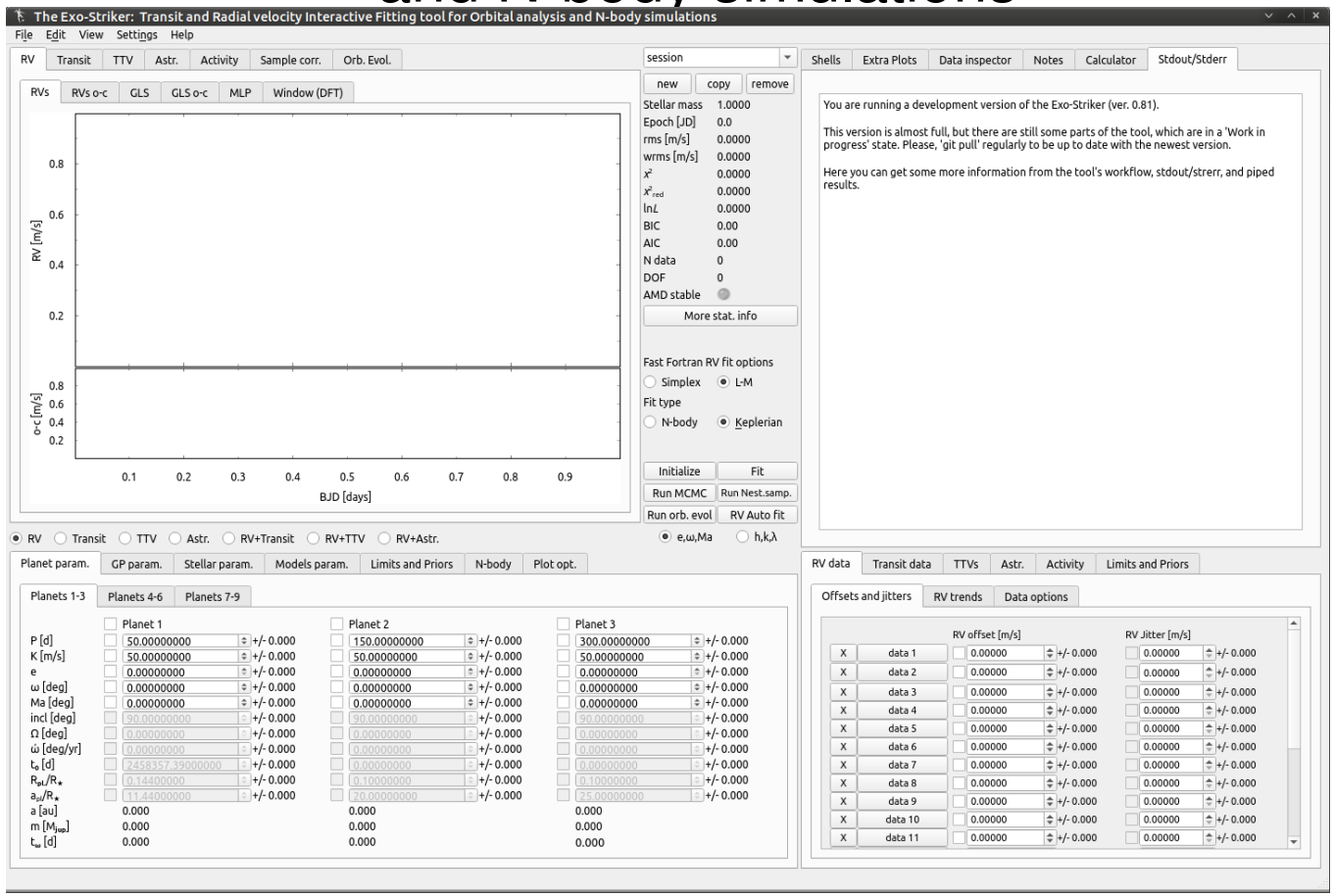
The Exo-Striker

by Trifon Trifonov

What is the Exo-Striker?

Fast and convient GUI toolbox for exoplanet orbital analysis and N-body simulations



Available on the GitHub

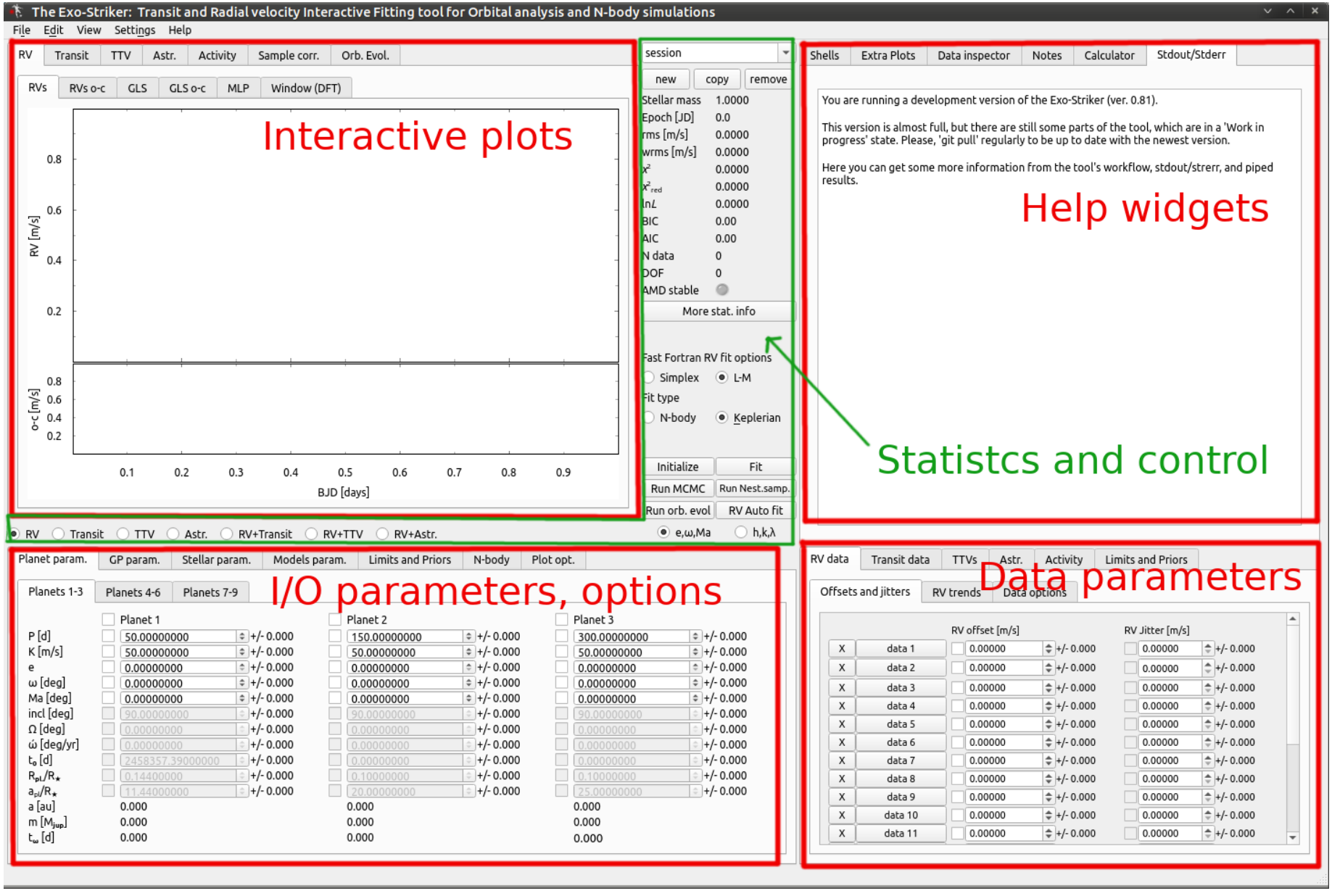
https://github.com/3fon3fonov/exostriker

(inspired by the “Systemic“ project. http://www.stefanom.org/systemic/)

What is it made of?

	Library	Reference
Exo-Striker (Ver. 0.81)		
Core code	Python & Fortran77	
GUI	Python/PyQt5	https://riverbankcomputing.com
Models (Keplerian and N-body)		
RV	Internal	
Transit	batman	Kreidberg (2015)
Astrometry	AstroMod	Schiwy et al. in prep.
TTV	ttv-fast	Deck et al. (2014)
GP	celerite	Foreman-Mackey et al. (2017)
Fitting schemes		
Best fit optimization	Internal / Scipy	
MCMC	emcee	Foreman-Mackey et al. (2013)
Nested sampling	dynesty	Speagle (2019)
Signal search & filtering		
RV	GLS, MLP	Zechmeister & Kürster (2009)
Transit	TLS, wotan	Hippke et al. (2019a,b)
Activity analysis	Internal	
Stability analysis	Swift (modified)	Levison & Duncan (2013)
Interactive plotting	pyqtgraph (modified)	http://www.pyqtgraph.org
Interactive shell	Jupyter, Bash, GPT-3	https://jupyter.org
Huge pub. RV database	Exo-Striker RVBanks	
$>$ 64 000 HIRES RVs	HIRES/HIRES NZP	Butler et al. (2017), Tal-Or et al. (2019)
$>$ 212 000 HARPS RVs	HARPS RVbank	Trifonov et al. (2020)

GUI layout


The Exo-Striker: Transit and Radial velocity Interactive Fitting tool for Orbital analysis and N-body simulations
File Edit View Settings Help
RV Transit TTV Astr. Activity Sample corr. Orb. Evol.
RVs RVs o-c GLS GLS o-c MLP Window (DFT)
Interactive plots
RV [m/s]
o-c [m/s]
BJD [days]
session
new copy remove
Stellar mass 1.0000
Epoch [JD] 0.0
rms [m/s] 0.0000
wrms [m/s] 0.0000
χ² 0.0000
χ²red 0.0000
lnL 0.0000
BIC 0.00
AIC 0.00
N data 0
DOF 0
AMD stable
More stat. info
Fast Fortran RV fit options
Simplex L-M
Fit type
N-body Keplerian
Initialize Fit
Run MCMC Run Nest.samp.
Run orb. evol RV Auto fit
RV Transit TTV Astr. RV+Transit RV+TTV RV+Astr.
e,ω,Ma h,k,λ
Shells Extra Plots Data inspector Notes Calculator Stdout/Stderr
You are running a development version of the Exo-Striker (ver. 0.81).
This version is almost full, but there are still some parts of the tool, which are in a 'Work in progress' state. Please, 'git pull' regularly to be up to date with the newest version.
Here you can get some more information from the tool's workflow, stdout/strerr, and piped results.
Help widgets
Statistcs and control
Planet param. GP param. Stellar param. Models param. Limits and Priors N-body Plot opt.
Planets 1-3 Planets 4-6 Planets 7-9
I/O parameters, options
Planet 1 Planet 2 Planet 3
P [d] 50.00000000 150.00000000 300.00000000
K [m/s] 50.00000000 50.00000000 50.00000000
e 0.00000000 0.00000000 0.00000000
ω [deg] 0.00000000 0.00000000 0.00000000
Ma [deg] 0.00000000 0.00000000 0.00000000
incl [deg] 90.00000000 90.00000000 90.00000000
Ω [deg] 0.00000000 0.00000000 0.00000000
ώ [deg/yr] 0.00000000 0.00000000 0.00000000
t₀ [d] 2458357.39000000 0.00000000 0.00000000
Rpl/R★ 0.14400000 0.10000000 0.10000000
apl/R★ 11.44000000 20.00000000 25.00000000
a [au] 0.000 0.000 0.000
m [Mjup.] 0.000 0.000 0.000
tω [d] 0.000 0.000 0.000
RV data Transit data TTVs Astr. Activity Limits and Priors
Offsets and jitters RV trends Data options
Data parameters
RV offset [m/s] RV Jitter [m/s]
data 1 data 2 data 3 data 4 data 5 data 6 data 7 data 8 data 9 data 10 data 11

import exostriker

IPython: home/tito

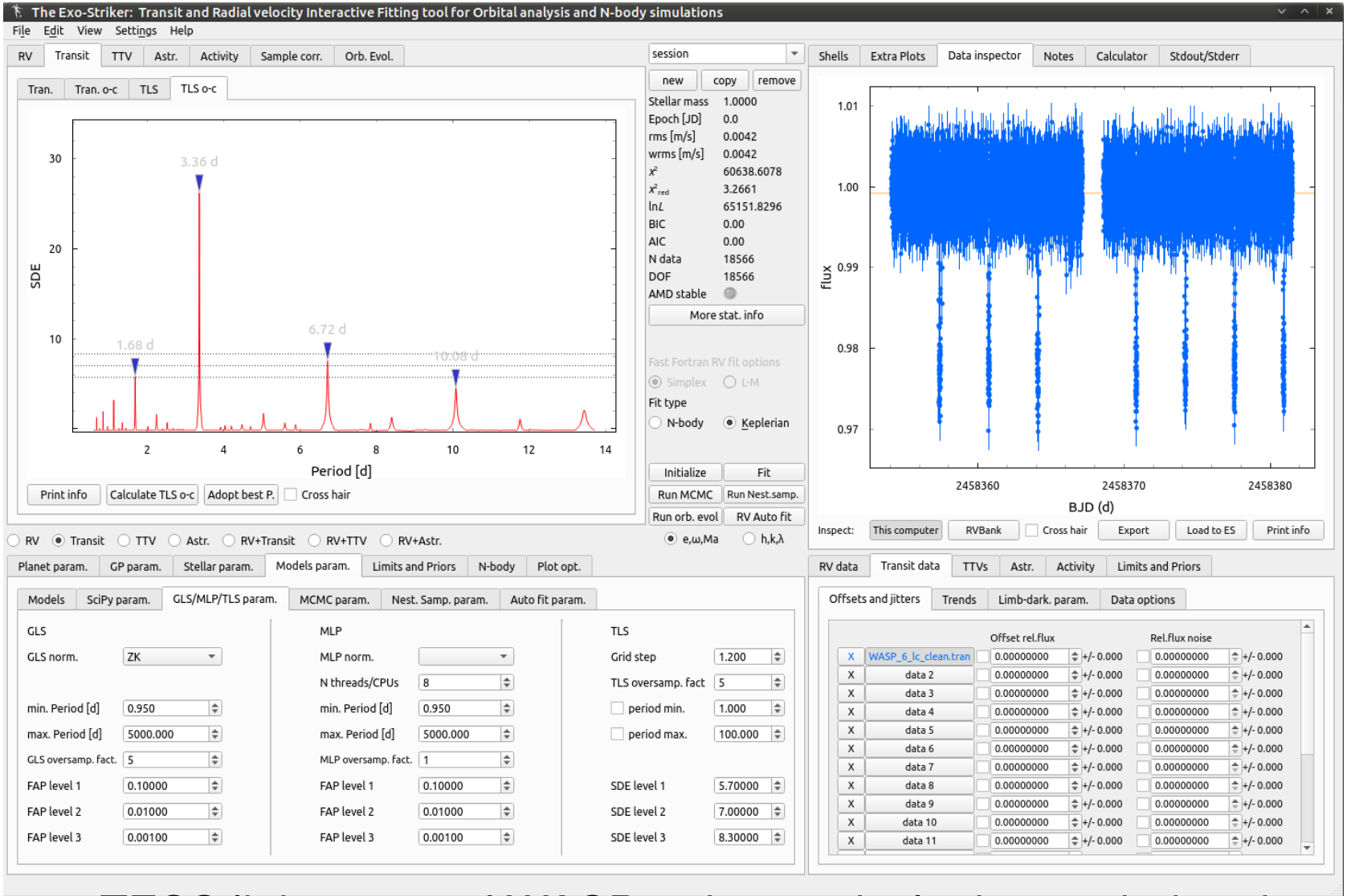
File Edit View Search Terminal Help

```
tito@terminator:~$ ipython3
Python 3.6.9 (default, Apr 18 2020, 01:56:04)
Type 'copyright', 'credits' or 'license' for more information
IPython 7.2.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: import exostriker # to load the Exo-Striker as a library

In [2]: import exostriker.lib.RV_mod as rv # to load a specific module, e.g. the RV fitting module

In [3]: fit = rv.signal_fit(name="hip5364") # creates the fit class object

In [4]: fit.add_dataset("hip5364_lick","./datafiles/hip5364.vels",0.0,10.0) # add RV dataset

In [5]: fit.add_planet(K=50,P=400,e=0,w=0,M0=0,i=90,cap=0) # add planet 1 (approx. init. param.)

In [6]: fit.add_planet(K=50,P=700,e=0,w=0,M0=180,i=90,cap=0) # add planet 2

In [7]: fit.fitting() # let's optimize!

In [8]: fit.P[0],fit.P[1] # e.g. to get the bestfit Periods of planet 1 & 2
Out[8]: (404.22299345507207, 751.8970043845964)

In [9]: rv.run_mcmc(fit, burning_ph=1000, mcmc_ph=5000, threads=20) #for a basic MCMC run
```

A brief demo of the Exo-Striker

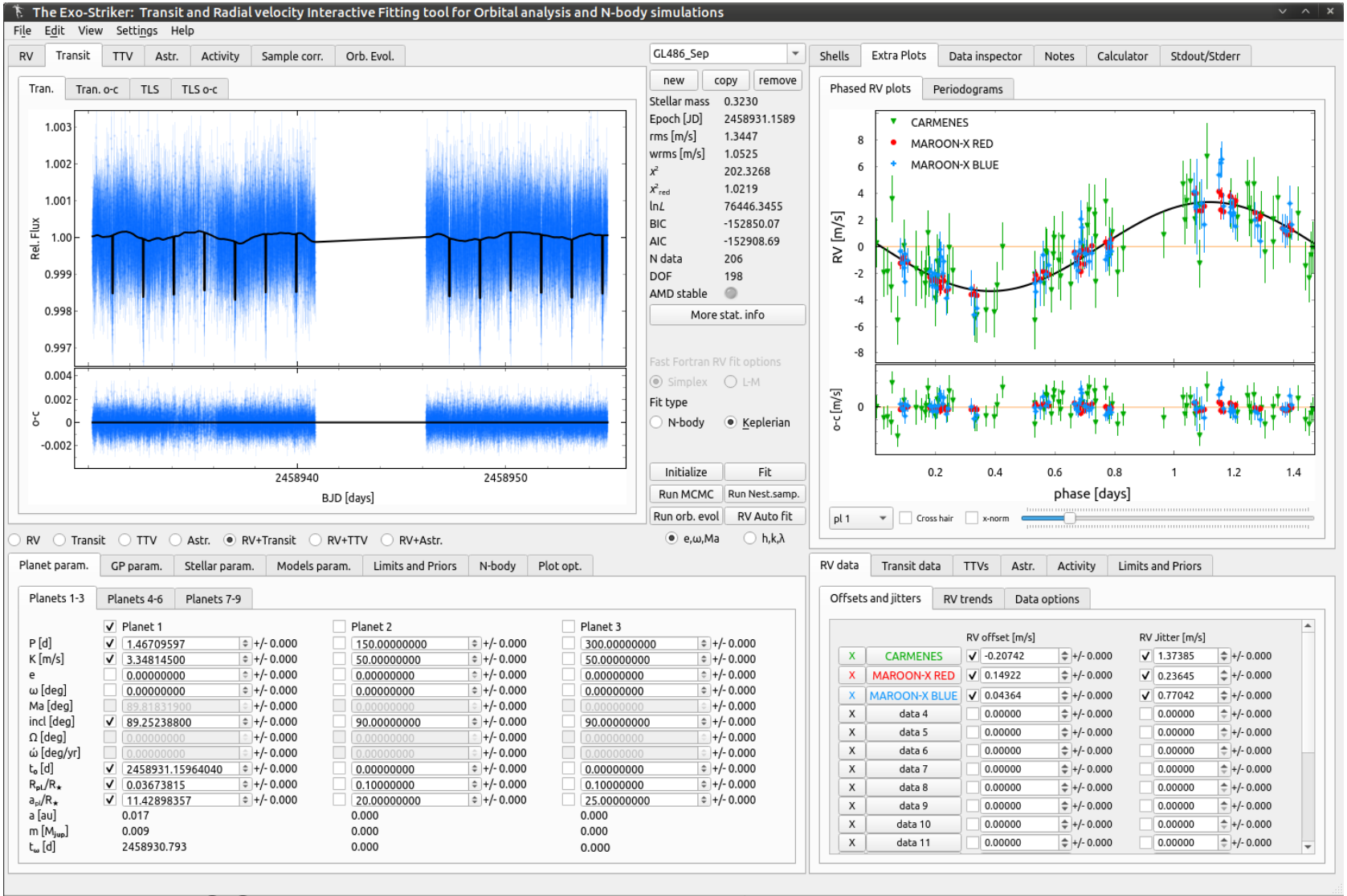
* Analysis of a 2:1 MMR pair discovered on multi-telescope, precision Doppler data, under 5 min.

Transit period search



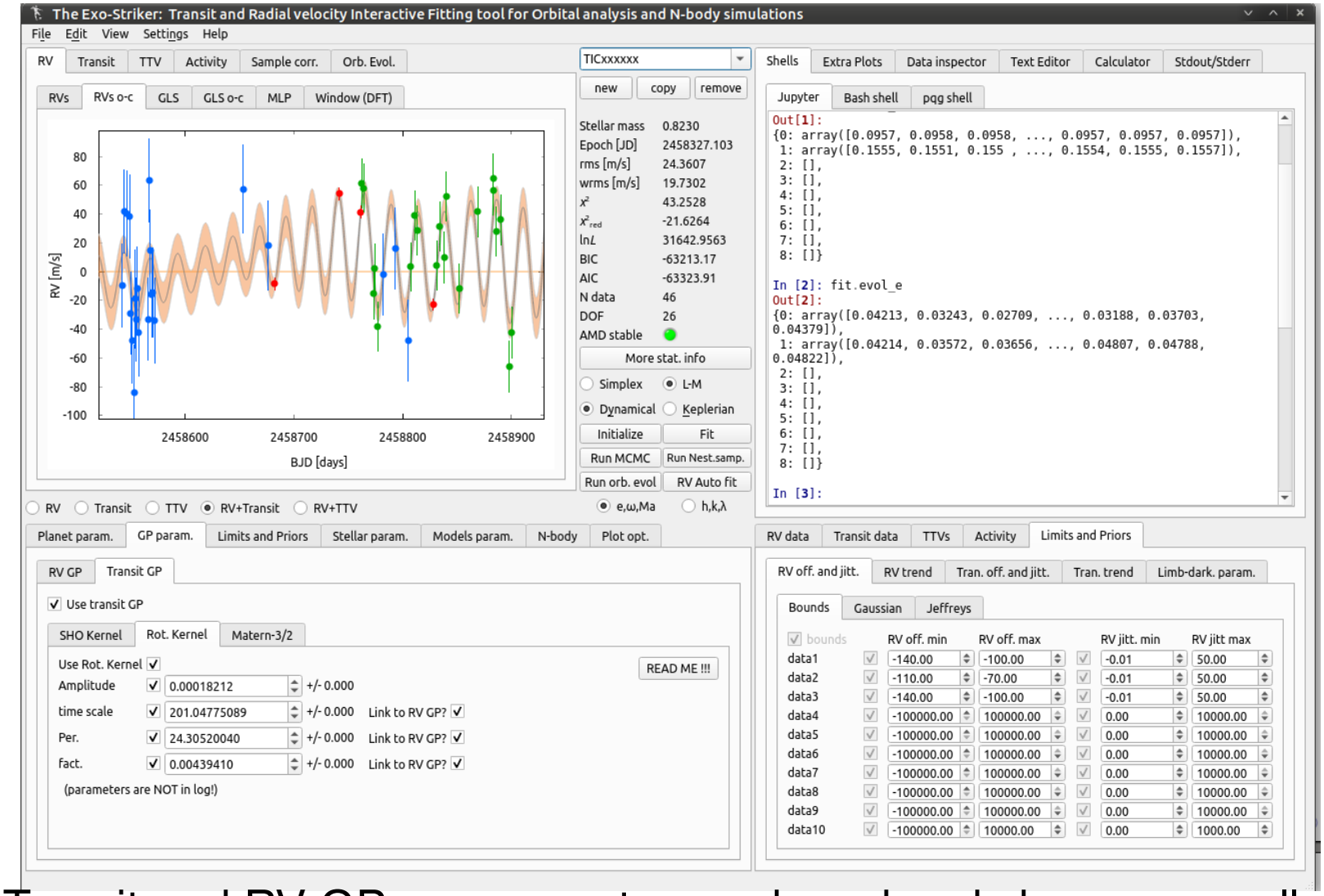
TESS light curve of WASP-6, interactively de-trended and analyzed.

Global data analysis



A TESS light curve + RV data of a nearby transiting planet GJ 486 b (Trifonov et al. 2021a).

GP modeling



Transit and RV GP components are shared and show an excellent agreement (Trifonov et al. 2021b).

What is next?

- Combined modeling with astrometry (ongoing).
- Improved photo-dynamical model.
- *Readthedocs* webpage.

Stay tuned!

Feedback and help in further development will be highly appreciated!

Supported in part by the BNSF program ”VIHREN-2021” project No. KP-06-DV/5