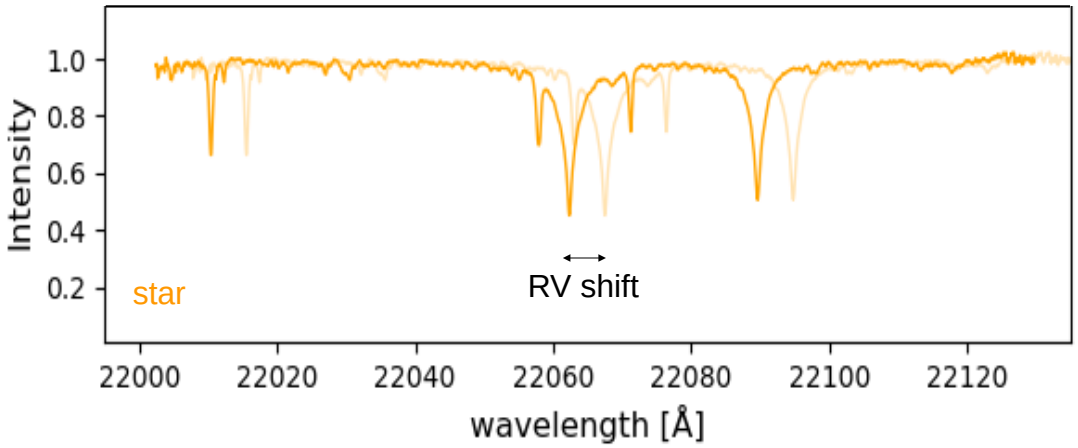
RV estimation with viper
- Theory + Tutorial -

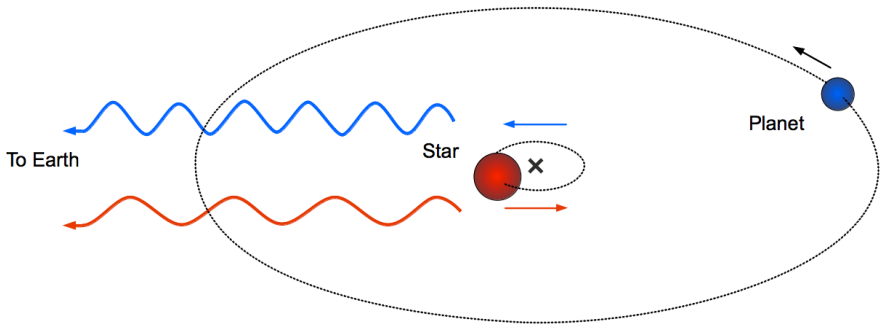
Jana Köhler (TLS Tautenburg)

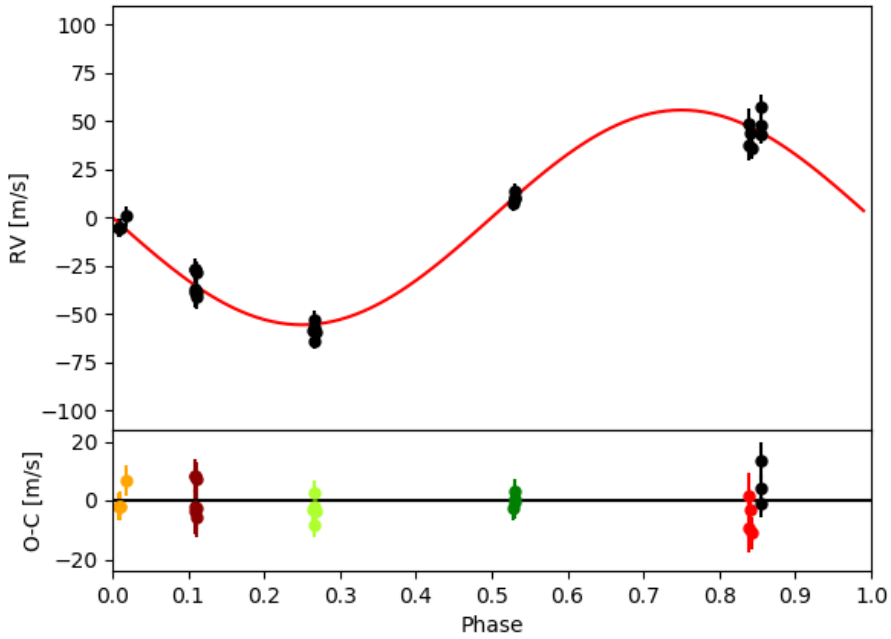
Ondrejov, 2024-08-29

Radial Velocties - RVs



- Stellar light get Doppler shifted due to planet star interaction
- RVs are measured via wavelength shifts of stellar absorbtion lines





The way to high precision RVs

RV estimation at the beginning
- mainly done via cross-correlation
- no correction for instrument instabilities
→ no high precision RV estimation possible

Later: Tellurics (Earth's atmosphere) for calibration
- Telluric lines have fixed wavelength
→ accuracy up to 10 m/s
- used for wavesolution
Problem: Tellurics chance strongly with time and are hard to model

Using Cell for calibration
- same idea as with Tellurics
- but using a stable Cell with known lines

Butler et al., 1996
- model instrument parameters (IP, Blaze, wavesolution) to estimate RV
- uses Cell lines for calculation of wavesolution
→ allows high RV precision

Basic concept of Butler et al.

Observation (with Cell)
- stellar through Cell

Stellar template (without Cell)
- “reference” spectra

FTS – **Cell** Spectra

Input

$$I_m = k\left[T_{cell}(\lambda)\, I_{stellar}(\lambda + \delta\lambda)\right] * IP$$

Output (optimized parameters)

Normalization
→ Blaze function

RV shift

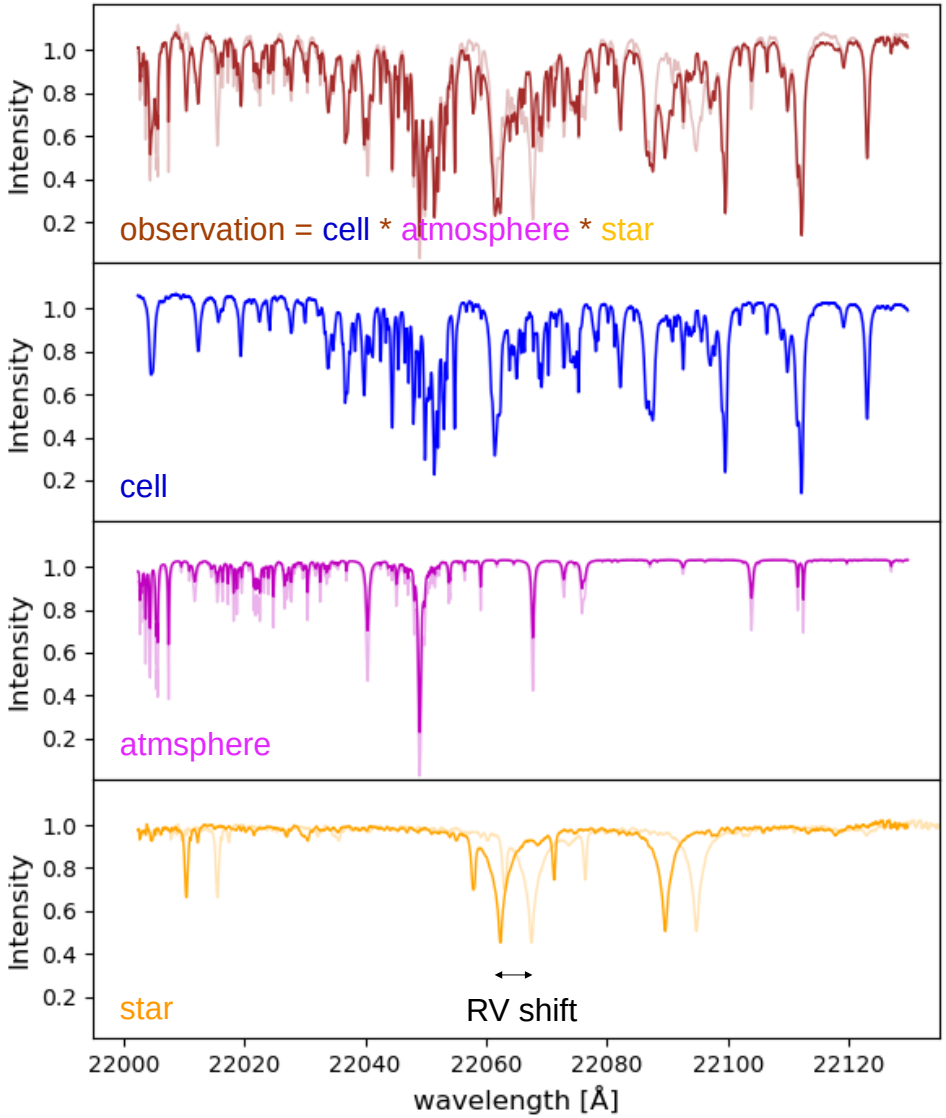
Instrument Profile

wavelength
- estimated via wavelength solution (wavelength vs pixel)
→ Cell Spectra is used as reference

least-square fit is used to estimate all parameters simultaneously

Based on Butler et al. 1996

The way to high precision RVs



$$I_m = k\left[T_{cell}(\lambda)\, T_{atm}(\lambda + \delta\lambda_{atm})\, I_{stellar}(\lambda + \delta\lambda_{star})\right] * IP$$

RV = radial velocity = $\frac{\delta\lambda}{\lambda} c$

RV in example = 70,000 m/s
RV we want ~ 10-20 m/s

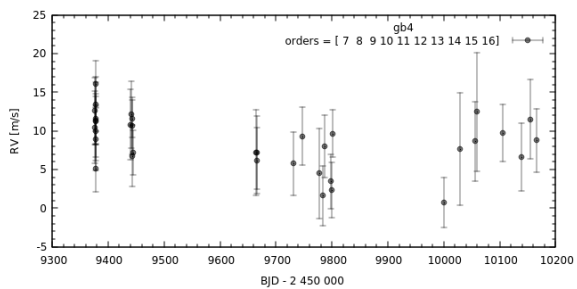
We need a software that can model our observations to estimated the RVs

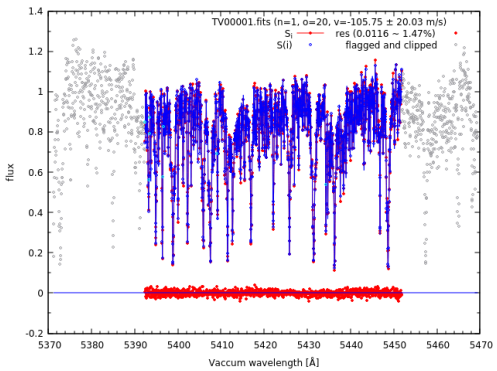
Velocity and IP EstimatoR - VIPER

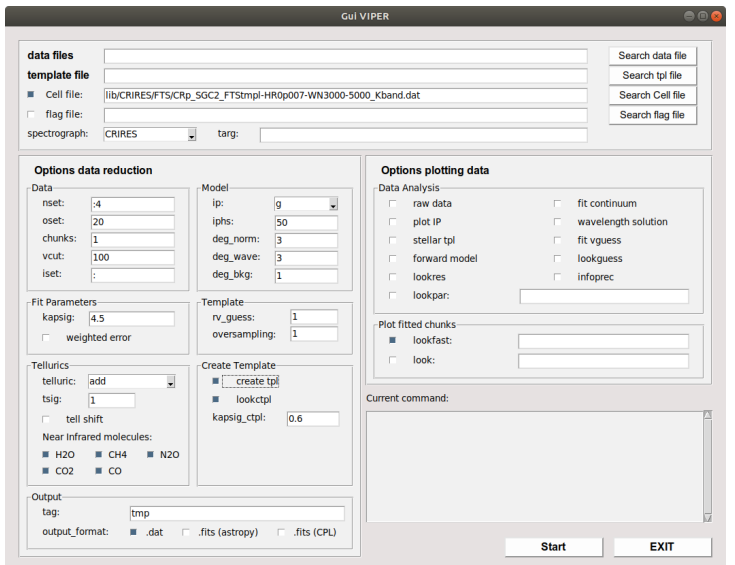
- Started by Mathias Zechmeister and Sireesha Chamarthi
- Allows to obtain high precision **Radial Velocity measurements** for different instruments

- Is available via github
  - https://github.com/mzechmeister/viper
  - If you publish results with viper, please acknowledge it by citing its bibcode from https://ui.adsabs.harvard.edu/abs/2021ascl.soft08006Z.

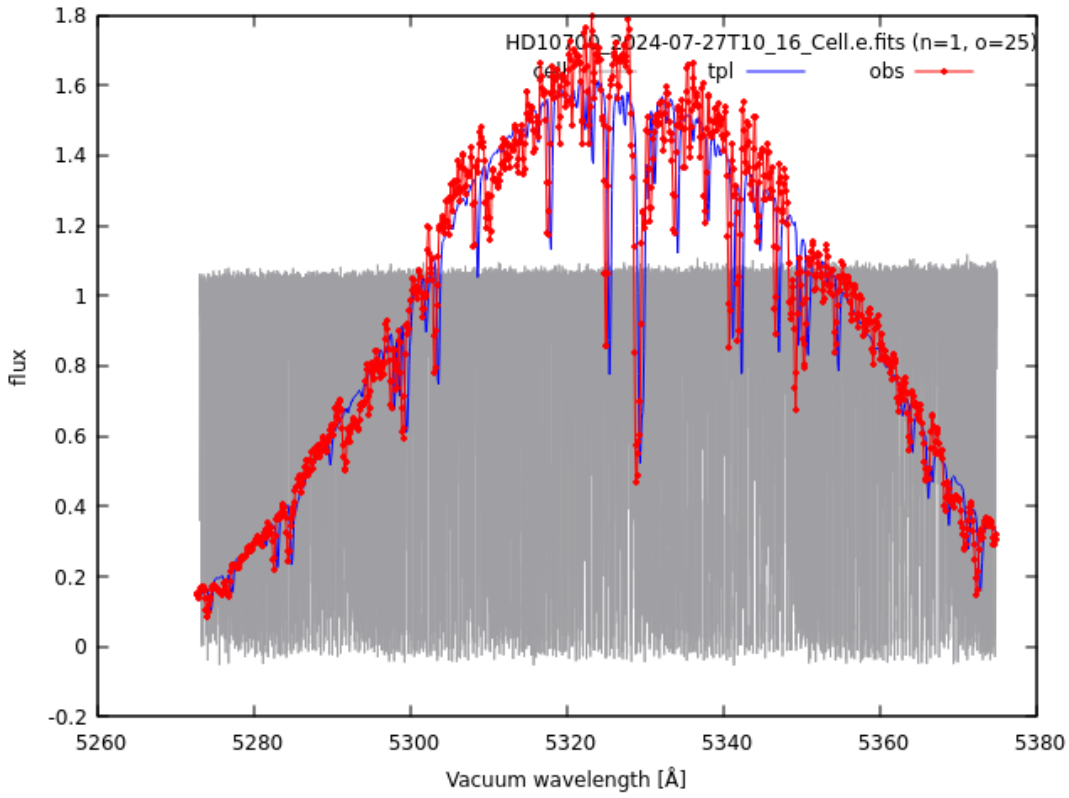
Current instruments:
- CRIRES+
- TLS Spectrograph
- UVES
- KECK
- Ondrejov OES
- CES
- **PUCHEROS**







Modelling the data – raw data



$$I_m = k\left[T_{cell}(\lambda)\, I_{stellar}(\lambda + \delta\lambda)\right] * IP$$

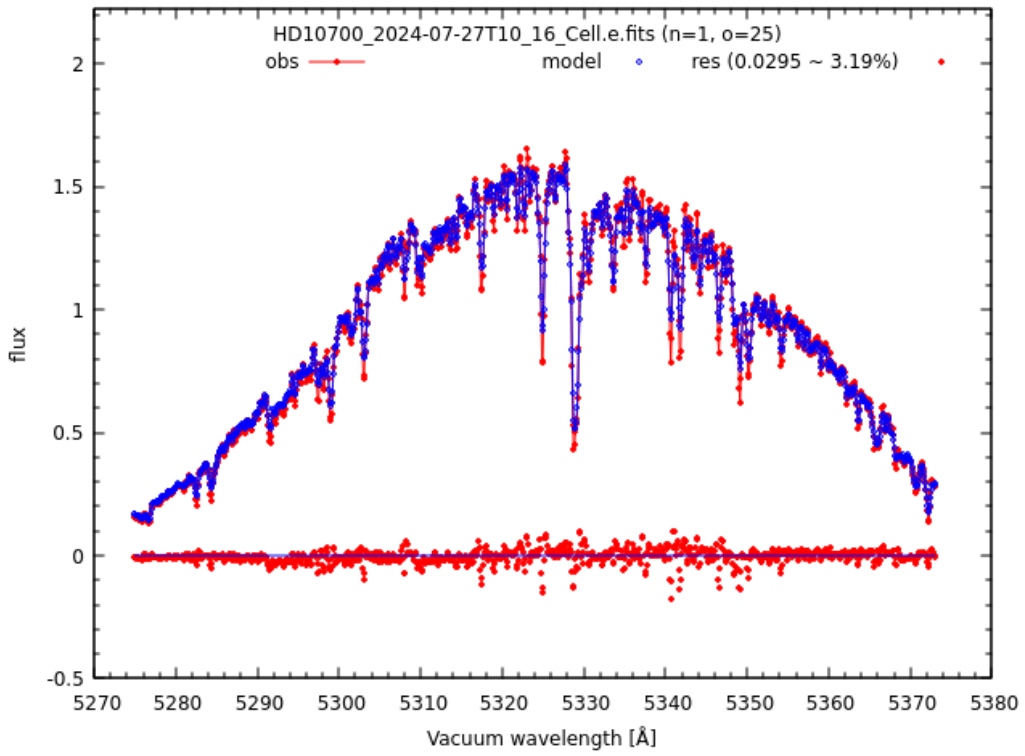
Input to viper:

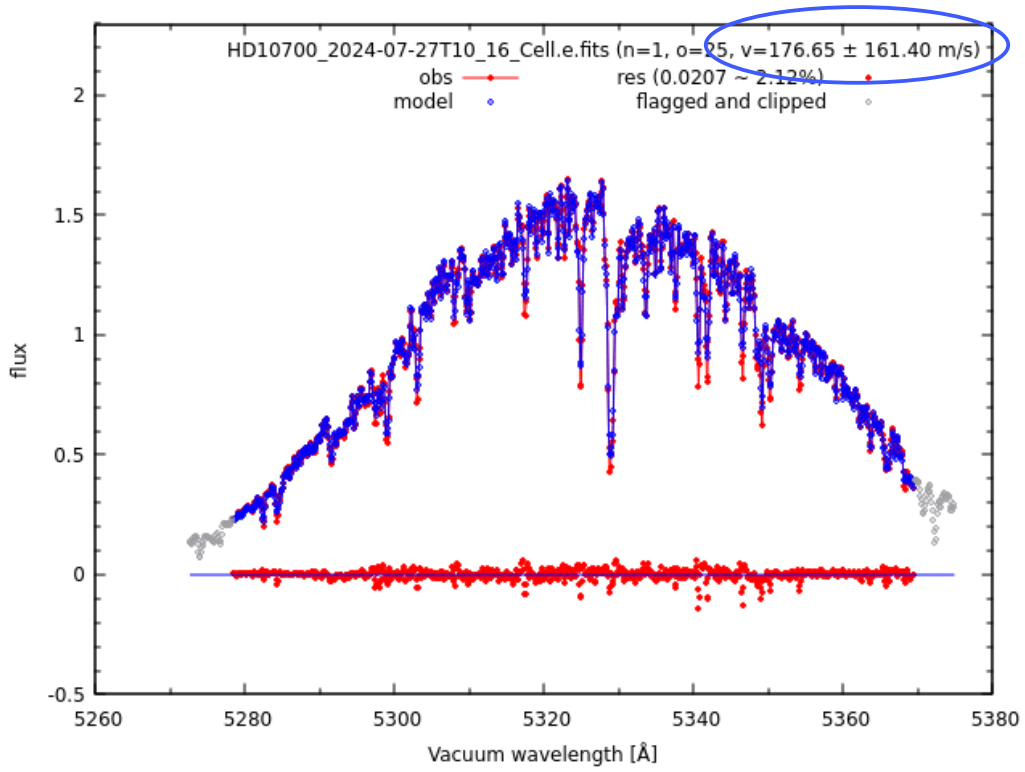
Observation with cell
Tpl – observation without cell
Cell – Fourier Transformed Spectrum (FTS)

Modelling the data

optimize

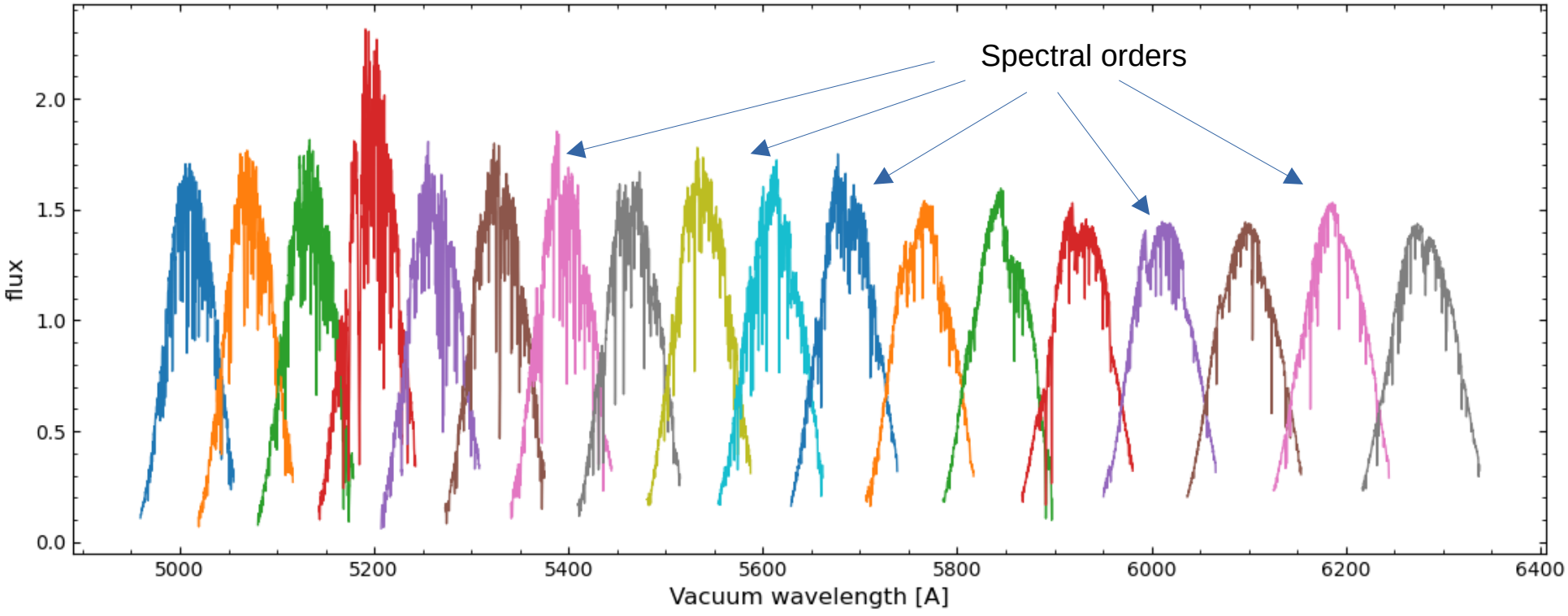
Start guess → $I_m = k[T_{cell}(\lambda) I_{stellar}(\lambda + \delta\lambda)] * IP$ → After optimization





Estimating the RVs

Instrument parameters vary for each spectral order
→ RV estimation will be done for each spectral order separately
→ afterwards combined by weighted mean



The parameters

$$I_m = k[T_{cell}(\lambda)\, I_{stellar}(\lambda + \delta\lambda)] * IP$$

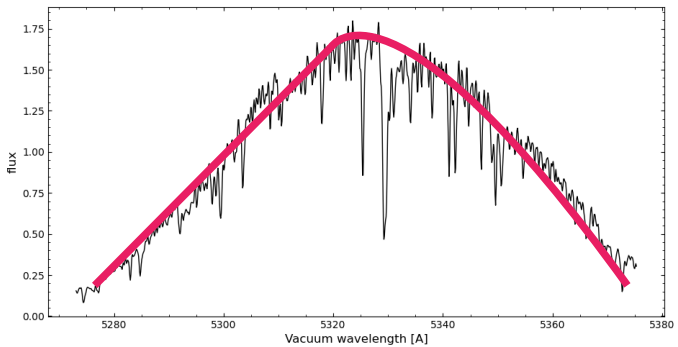
wavelength solution

$$\lambda = a_0 + a_1 \cdot x + a_2 \cdot x^2 + a_3 \cdot x^3$$

x: detector pixel

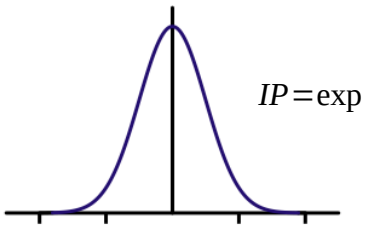
blaze

$$k = b_0 + b_1 \cdot x + b_2 \cdot x^2 + b_3 \cdot x^3$$



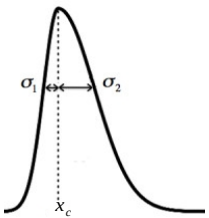
Instrument profil (IP)

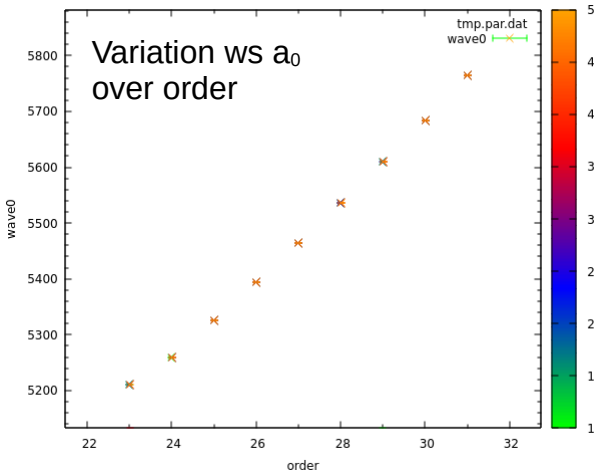
Gaussian

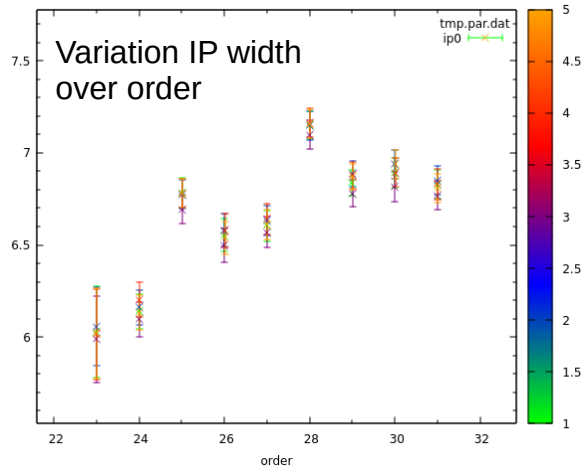


$$IP = \exp\left(-0.5 \cdot \frac{x^2}{\sigma^2}\right)$$

BiGaussian



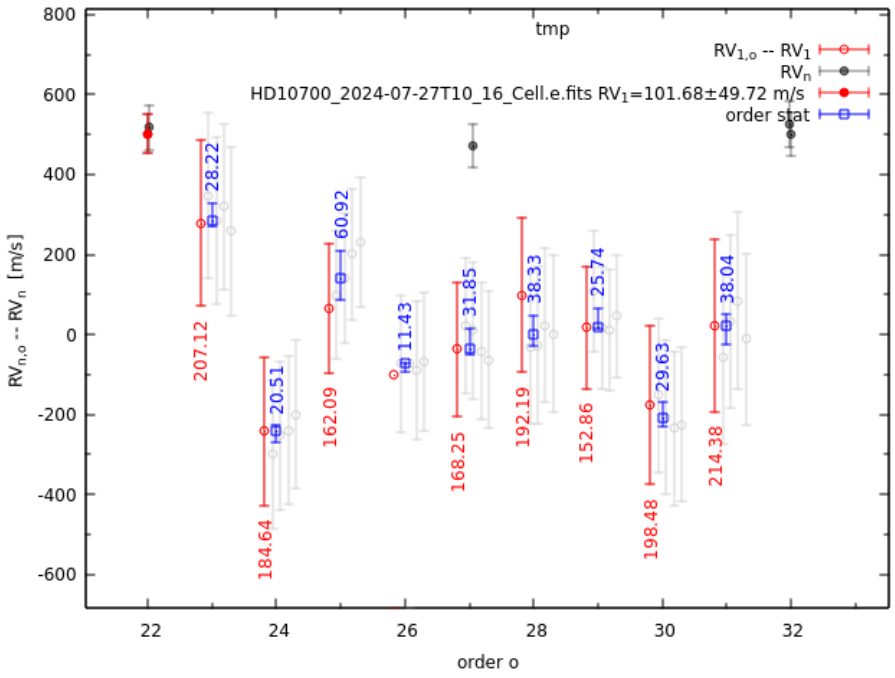


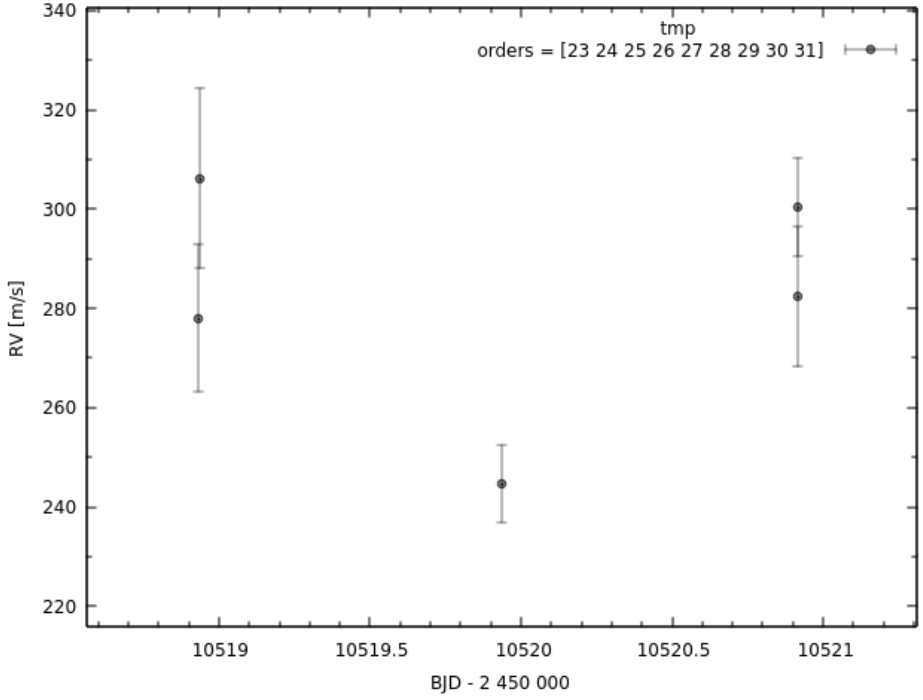


Final RVs – combination of all orders

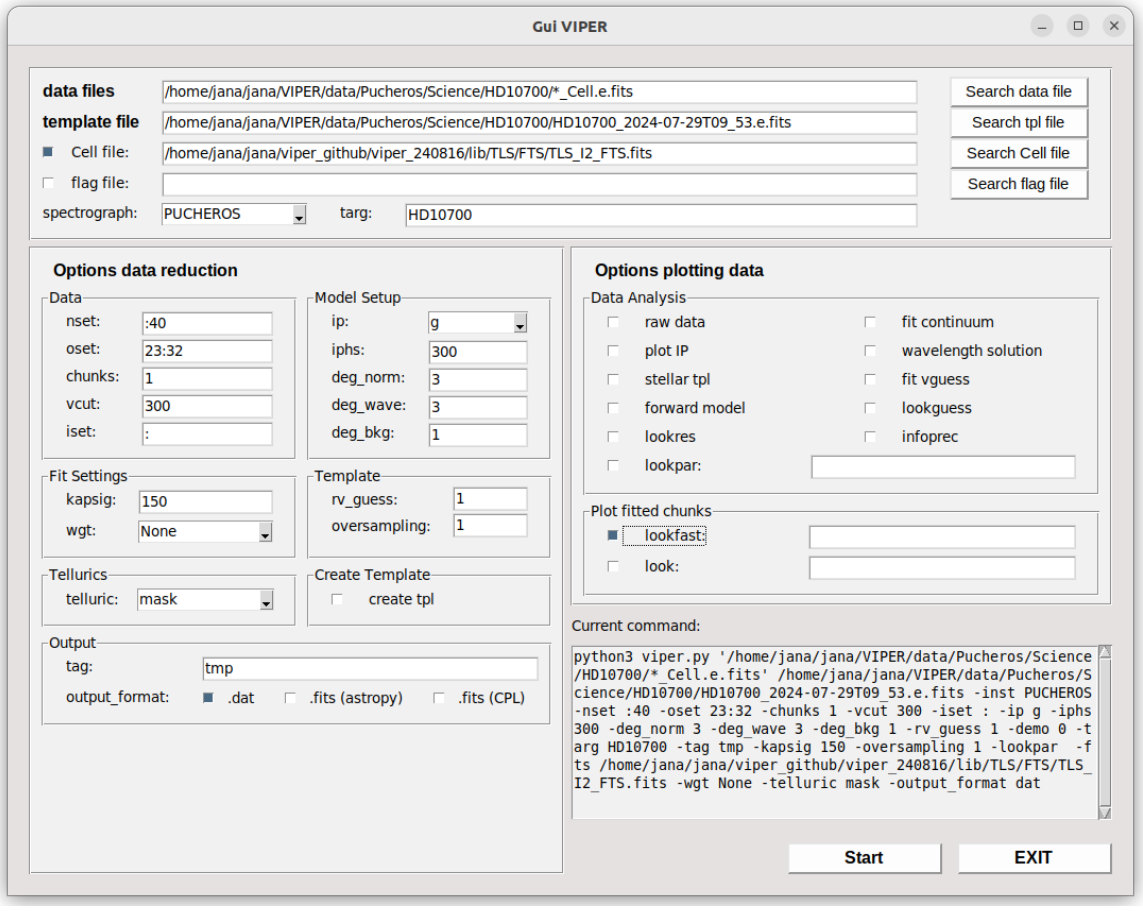
$$RV = \frac{\sum \epsilon_o^{-2} \cdot RV_o}{\sum \epsilon_o^{-2}}$$

RVs per order → combined RVs





Starting viper



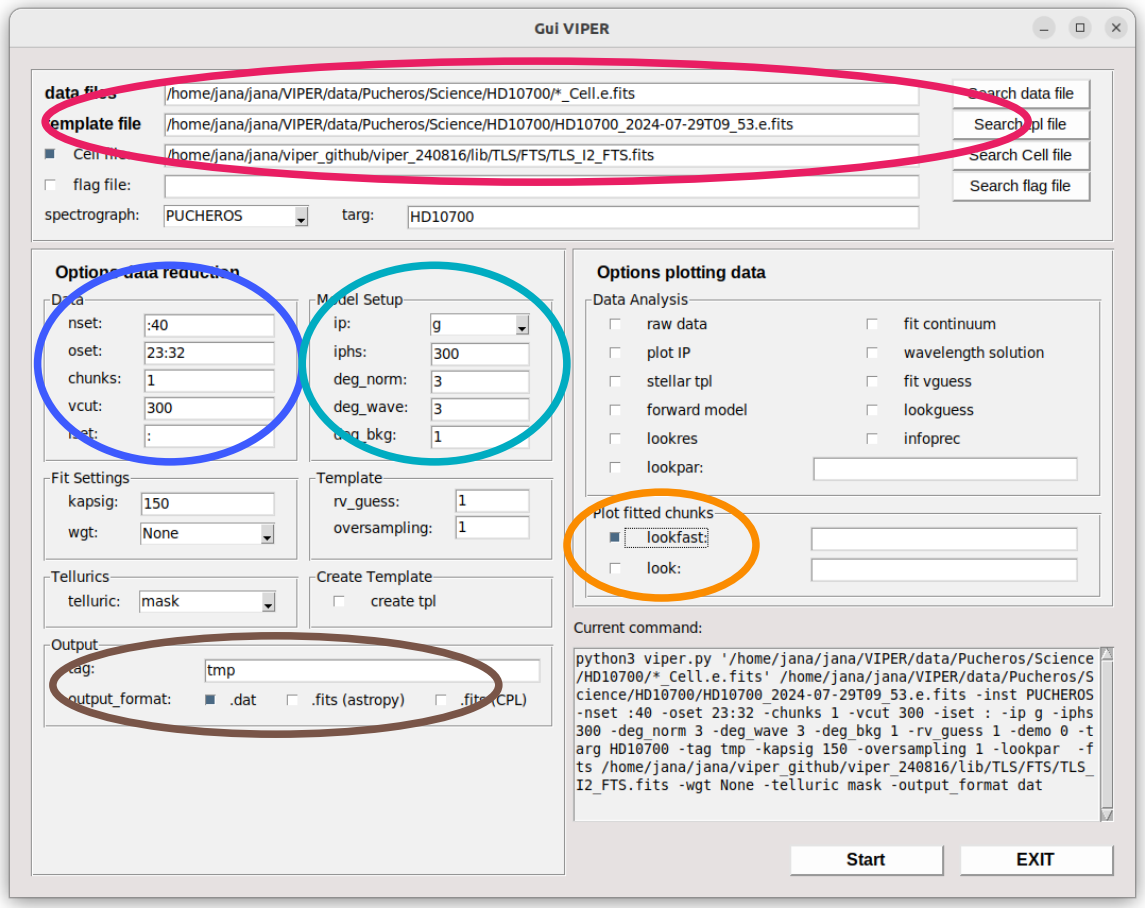

```
> GUI_viper config_viper.ini
PUCHEROS
```

or

```
> python3 GUI_viper.py
config_viper.ini  PUCHEROS
```

```
sudo apt-get install gnuplot
gnuplot-x11 gnuplot-doc
```

Starting viper



Input data
- observation with cell
- stellar template (without cell)
- instrument/ cell file

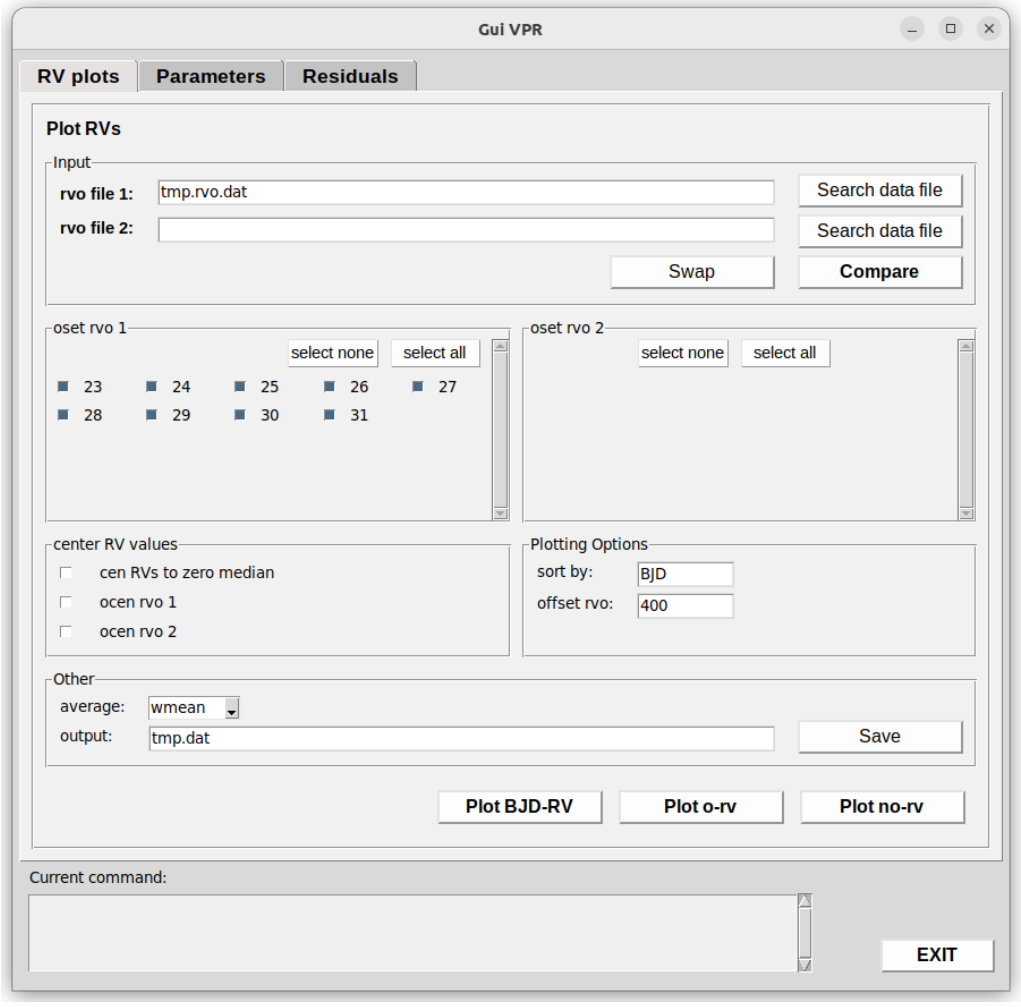
Observation and order selection

Parameter settings
- IP model and width
- degree wavelength solution
- degree blaze

Plotting options
- no plotting
- fast plotting (without pause)
- pause each plot

Output name and format
- default: tmp
- will be overwritten each time!

Post-processing of the RV data

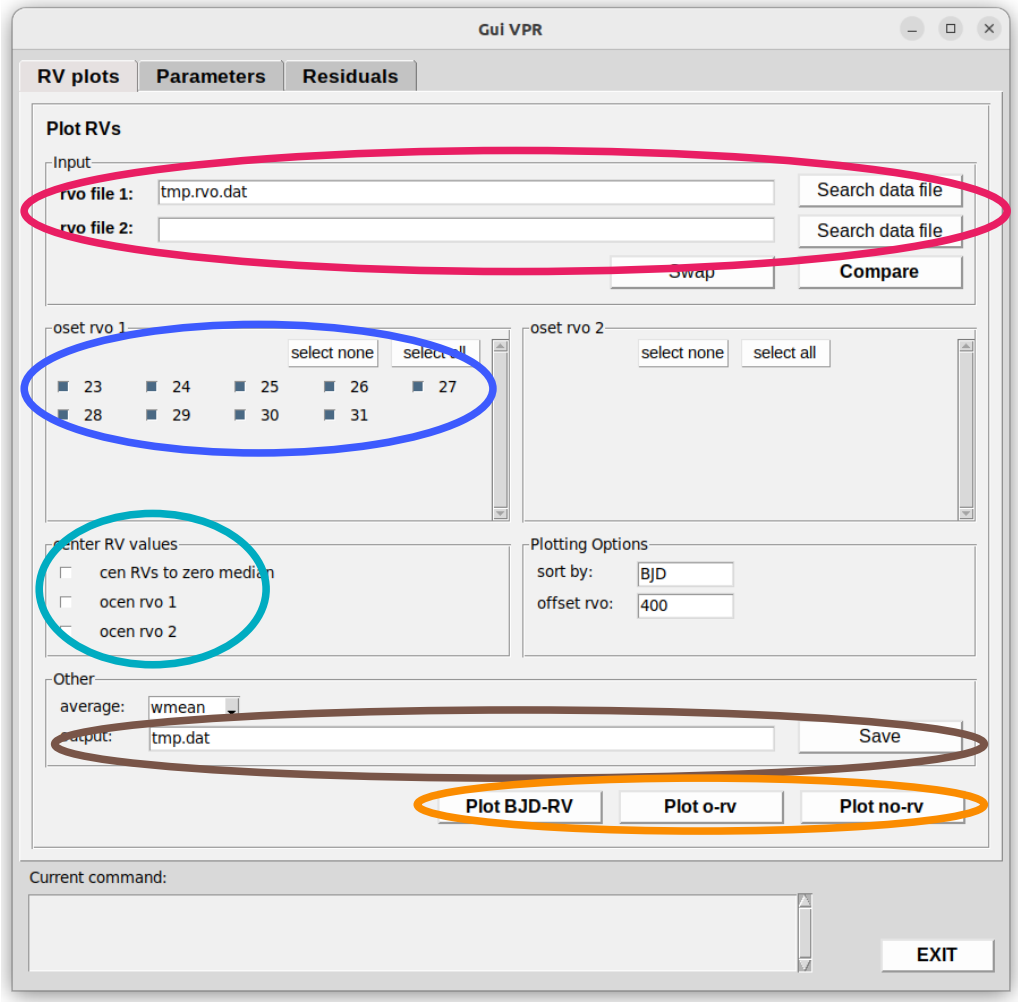



```
> GUI_vpr
```

or

```
> python3 GUI_vpr.py
```

Post-processing of the RV data



Input data
- #.rvo.dat file
- RVs for all orders and observations

Order selection

Parameter settings
- center combined Rvs
- subtract order offset

Plotting options
- combined RVs vs time
- RVs of single orders

Output name and format
- write out final RVs
- will be overwritten each time!