VIPER Test Report

Jana Köhler

2024-01-08

Contents

1 Installation and System Requirements 3

2 Important changes towards the last version 3
- 2.1 Configure files . . . 3
- 2.2 Plotting . . . 3

3 Test dataset 4

4 Quick test 4

5 Recommended reduction steps to obtain the best RV results 4
- 5.1 Creating a telluric free template . . . 5
- 5.2 Improved template creation . . . 6
- 5.3 RV estimation . . . 7

6 Further processing 7

7 Known Issues 9

1 Installation and System Requirements

Viper runs with Python3 and with the need of the following standard packages: numpy, scipy, astropy, argparse, datetime and gnuplot.

For the installation just download *viper*, enter the directory and run

```
1> pip install -e .
```

Afterwards *viper* can be run with

```
1> viper [options]
```

To test if the installation was successful, one can use pytest (inside the *viper* directory) via:

```
1> python3 -m pytest -s
```

The options `-s` and `-m` are needed to avoid problems with used modules and writing the output files.

2 Important changes towards the last version

Beside of a number of minor changes, some major changes have been applied to the code of which the user shoul be aware of.

2.1 Configure files

To allow an easier processing, the option of using configure files for *viper* has been added. As part of that, a default file has been generated, containing default parameters for several instruments. For the examples below, both command options, with and without configure file, will be shown. It is possible to set parameters at the console even if a configure is given. For more information about the configure file please read the manual.

2.2 Plotting

In the last version the plotting of the results was set to true by default. In the newer version, this is not the case anymore. If not directly chosen, no more plots will be shown, when running *viper*. If the user is still interested in watching the results, two options are available.

Adding the option `-lookfast` in the *viper* command line, will switch on the plots and run through them, just like it was the default case in the last version. If the user is interested in having a closer look, the option `-look` will pause at each plot allowing an even more detailed examination.

It is recommended to take a look on the plots to make sure that no problems appear during the calibration process and that the data have the expected quality. Nevertheless, it can be helpful to switch off the plotting, and by this speed up the processing, if the user is aware of the data quality and just want to apply small changes on the parameters to improve the results.

In the following examples, the `-lookfast` option is not added to the commands anymore. More information on both plotting options can be found in the manual.

3 Test dataset

For a fast and simple functionality test, data of the RV standard star GJ588 has been chosen. This target is a M2.5 star with a brightness of K=4.76 mag and located at RA=233.03 deg and DEC=-41.3 deg. While most data come from the weekly performance test observations (from ESO), some data as well were observed during GTO time (Program ID: 108.22CH.001, PI: Nagel). As these data are not used for scientific study but as a reference star, the PI of the program allowed the usage of the data for this test report. The 11 selected observations were performed between March and August 2022, and always contain one observation using the SGC and one without using it. The data were reduced using V1.3.0 of the CR2RES DRS pipeline.

observation dates	DIT	NDIT	NODDING	Program ID
2022-03-25 - 2022-03-27	30	2	AAAABBBB	108.22CH.001
2022-05-31 - 2022-08-09	45	1	AAABBB	60.A-9051

4 Quick test

To test in a quick and easy way how *viper* is working, just enter the *viper* directory via a console and type:

```
1> viper "data/CRIRES/GJ588/withSGC/220325_1.fits" data/CRIRES/GJ588/noSGC/
   220325_1.fits -inst CRIRES -nset :1 -oset 7:17 -deg_norm 2 -deg_wave 2
   -output_format cpl
```

or alternatively (using CRIRES+ default values from configure file):

```
1> viper "data/CRIRES/GJ588/withSGC/220325_1.fits" data/CRIRES/GJ588/noSGC/
   220325_1.fits -inst CRIRES -config_file CRIRES -output_format cpl
```

If no errors appear, *viper* should run through the selected orders of the one selected observation. In the end the final combined RV is plotted.

5 Recommended reduction steps to obtain the best RV results

After the fast functionality test, this section describes the recommended routine to obtain the best results for the RVs.

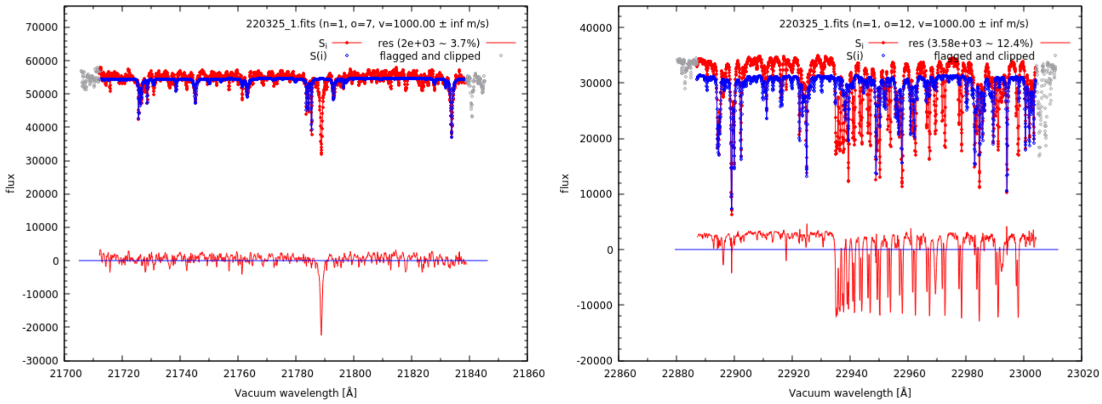

Figure 1: Output from *viper*. Example plots of two orders from one observation. The red points represent the observations, while the blue dots represent the optimized telluric model. In the bottom, the telluric-free stellar template is shown.

5.1 Creating a telluric free template

Due to the high contamination of earth atmosphere lines in the NIR, the best RV precision is obtained by using the forward modelling of the telluric lines on the observed spectra. As a result, the test report will concentrate on this reduction technique. For this, we first generate our own telluric-free stellar template using all observations performed without using the SGC.

By calling the following command:

```
1> viper "data/CRIRES/GJ588/noSGC/22*" -inst CRIRES -nset :12 -oset 7:17
   -deg_norm 0 -deg_wave 2 -deg_bkg 1 -oversampling 1 -createtpl -telluric add
   -tsig 10 -vcut 10 -nocell -rv_guess 1 -kapsig 15 -tag tmp1 -output_format cpl
```

or alternatively (using CRIRES+ default values from configure file):

```
1> viper "data/CRIRES/GJ588/noSGC/22*" -inst CRIRES -output_format cpl
    -config_file CRIRES_tpl1 -nocell -nset :12 -tag tmp1
```

all selected observations will be read in at once. *Viper* then runs through all selected orders and observations and models and removes the telluric lines from the stellar spectra, before combining them to the final telluric-free template.

To have a look at the corrected spectra and the combined created template, add -lookctpl.

Fig. 1 shows two example outputs as can be seen by running *viper*. After the code has run through all the data, the final output of one order is shown in Fig. 2. While on the top of the plot all telluric corrected spectra are over plotted, in the bottom the final combined spectra is shown. The combined spectra of all orders will afterwards be saved in the file tmp1_tpl.fits and can be used for the further analysis.

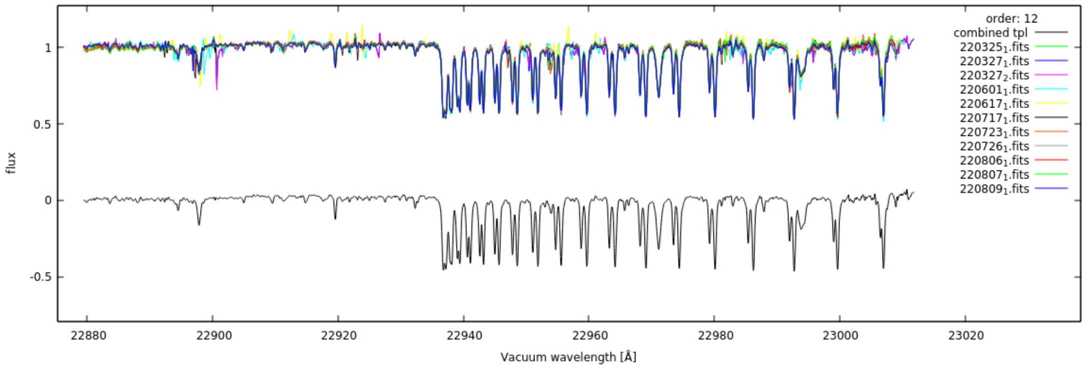

Figure 2: Output from *viper*. Combined template of GJ588 (black line on the bottom) created by using several telluric corrected spectra observed without cell (coloured lines in the top).

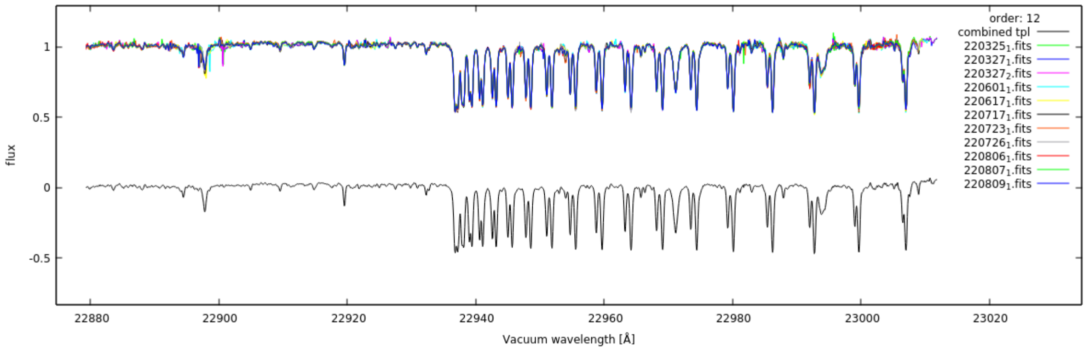

Figure 3: Output from *viper*. Combined template of GJ588 (black line on the bottom) created by using several telluric corrected spectra observed without cell (coloured lines in the top) using the improved template creation.

As no reference template is used, no RV values are modelled and all printed RV values in the console and data files showing the same value of 1000 m/s, which is the RV start guess value. Related to that, the corresponding errors of the RVs are set to inf.

5.2 Improved template creation

To improve the telluric-free template and obtain even better RV results, a second round of template creation can be run. The template generated in the previous sub-section will hereby be used as reference template. Due to that, *viper* can better distinguish between stellar and telluric lines, which leads to a better modelling and correction of the telluric lines. Changing a few other parameters, the command for the template creation would be::

```
1> viper "data/CRIRES/GJ588/noSGC/22*" tmp1_tpl.fits -inst CRIRES -nset :12
   -oset 7:17 -deg_norm 2 -deg_wave 2 -deg_bkg 1 -oversampling 1 -createtpl
   -telluric add -tsig 1 -vcut 10 -nocell -rv_guess 1 -kapsig 15 6 -tag tmp2
   -output_format cpl
```

or alternatively (using CRIRES+ default values from configure file):

```
1> viper "data/CRIRES/GJ588/noSGC/22*" tmp1_tpl.fits -inst CRIRES -nset :12
   -output_format cpl -config_file CRIRES_tpl2 -nocell -tag tmp2
```

The created template for one of the orders is plotted in Fig. 3. Comparing it to Fig. 2 from the previous sub-section make the improvements in the modeling process visible.

As this time a reference spectrum is used, RV values are modelled and differ from the start guess values.

5.3 RV estimation

For the estimation of the RVs, the generated template tmp2_tpl.fits will be read in into *viper* together with the data observed using the SGC as done by the following command:

```
1> viper "data/CRIRES/GJ588/withSGC/22*" tmp2_tpl.fits -inst CRIRES -nset :12
   -oset 7:17 -deg_norm 2 -deg_wave 2 -deg_bkg 1 -telluric add -tsig 1 -tellshift
   -kapsig 15 6 -oversampling 1 -flagfile <path to viper>lib/CRIRES/flag_file.dat
   -output_format cpl
```

or alternatively (using CRIRES+ default values from configure file):

```
1> viper "data/CRIRES/GJ588/withSGC/22*" tmp2_tpl.fits -inst CRIRES -nset :12
   config_file CRIRES -tellshift -flagfile <path to viper>/lib/CRIRES/flag_file.dat
   -output_format cpl
```

In comparison to the template creation, here a value of 1 is selected for the parameter `tsig`. This means all data in the spectra are weighted equally and no distinguish between regions with and without tellurics is done. Note, that in case of a bad template, a value of around 0.7-0.9 can improve the RV precision. Furthermore, a simple flag file is used to remove some noisy parts of individual orders. A possible output when running *viper* can be seen in the left plot of Fig. 4. The final RVs of all observations are plotted in the end when using `-lookfast` and shown in the right plot of Fig. 4. Here, the error bars of the RVs appear pretty large. A few more comments about that and how to reduce them is described in the following selection. All RV values, for the individual orders as well as the weighted mean RV are saved in the tmp_rvo_par.fits file, together with the calculated parameters from the optimization process.

6 Further processing

While running *viper*, the RVs of the individual orders will be saved together with the mean RV value of all orders in the tmp_rvo_par.fits. This file allows the user to study the results in the best way. For an easy first study, the *viper* package comes along with the script vpr.py. This script is already called at the end of *viper*, when the RV estimation is done. Running

```
1> vpr tmp_rvo_par.fits -avg wmean
```

gives the same plot as already seen at the end of the last section. Furthermore, by pressing enter in the console, it will produce the right plot of Fig. 5. Here, the RVs for all observations for the individual orders are shown together with the RV rms. The visible offset between the spectral orders is the reason for the higher error bars, which

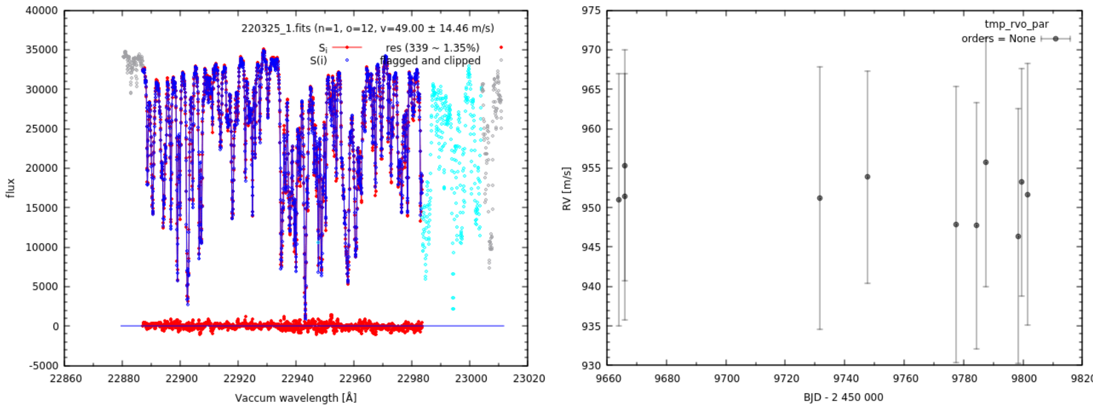

Figure 4: Output from *viper*. Left: Example plot of one order from one observation. The observed data are over plotted with the optimized model (blue dots). In the bottom, the residuals between observation and model are potted. Right: Final RVs for all observations.

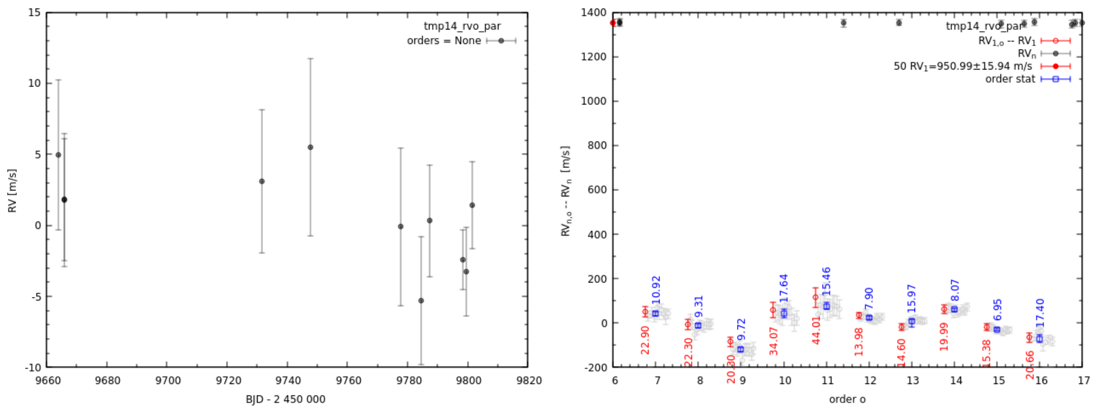

Figure 5: Output from vpr.py. Left: Final RVs of all observations after subtracting the mean values from each individual order. Right: RVs of all orders for all observations (bottom) together with the RV rms for each order.

is calculated using the standard deviation of all orders from one observation. Extending the call to

```
1> vpr tmp_rvo_par.fits -avg wmean -ocen -cen
```

will centre the RVs around zero by subtracting the mean of all RV values by using the option `-cen`. Furthermore, the option `-ocen` will subtract the mean values from each individual order and therefore will reduce the calculated errors of the RVs. The new output is shown in the right plot of Fig. 5, while in the console a RV rms value of around 3.2 m/s and a median e_RV of around 4.5 m/s are printed.

The final RVs can be save with

```
1> vpr tmp_rvo_par.fits -avg wmean -ocen -cen -save GJ588.fits
```

or

```
1> vpr tmp_rvo_par.fits -avg wmean -ocen -cen -save GJ588.dat
```

This way, the BJD, the updated weigthed mean RV and e_RV will be stored whether in a FITS or text file, ready for further processing with other software.

7 Known Issues

As *viper* is still under development and the telluric forward modelling is not trivial, the telluric modelling is not working great for all observations yet. These difficulties are under investigation and (hopefully) will be solved in the future. While the process already has been improved during the last month, especially order 16 of the K2192 setting often shows problems in the modelling process, as it contains one strong saturated telluric line. This leads to RV error values of inf.