
spectro Documentation

Release latest

Aug 25, 2023

Contents

1	Contents	1
1.1	Introduction	1
1.2	Installation	2
1.3	Graphical User Interface	3
1.4	Objects in GUI	5
1.5	Tutorial	7
1.6	Keyboard	9
1.7	Console	12
1.8	Main menu	14
1.9	Fit model	15
1.10	Bayesian inference with MCMC	15
1.11	Plot line profiles widget	15
1.12	Preferences menu	16

1.1 Introduction

1.1.1 What is spectro package?

Spectro package is a package collects the scripts for the analysis of the astrophysical observations. Its main module `svviewer` provided powerful tool for the interactive spectral analysis. For the moment the majority of the tool is for the absorption line analysis. The graphical interface is based on PyQt and pyqtgraph packages, that are high-performance graphical packages. The base of the code is written in Python with use of the external high-level packages provided by astronomical community, however the code also allows one to use Julia language to accelerate extensive calculations of the absorption line profiles models and estimation of their parameters.

1.1.2 What it can do?

The main features of spectro package is

- Powerful *Graphical User Interface* for spectra analysis
- Comprehensive visually guided absorption line modeling
- Parameter estimations of line profiles using Levenberg-Marquard and MCMC methods
- 1d spectrum reduction/coadding/modification routines
- 2d to 1d spectral reduction
- synchronization with external databases, such as SDSS, NIST, ...
- calculation of population ratio for atomic and molecular levels
- ISM neutral phase diagram calculation
- ...

1.1.3 This documentation

The main part of this documentaion describes the `sviewer` - *Graphical User Interface* for the absorption line analysis, which is located in `spectro/sviewer` folder, including some tutorials for the basic introductions in absorption line analysis with `sviewer`.

1.2 Installation

This section describe how to download, install dependencies and config `Spectro` package for the first use

1.2.1 Downloading

Since the `Spectro` package is still developing the most straightforward way is to install `spectro` from github, which provides you access to the latest features and bugfixes:

1. Clone `Spectro` from github:

```
$ git clone https://github.com/balashev/spectro
```

2. You can simply place the `Spectro` folder someplace importable, such as inside the root of another project. `Spectro` does not need to be “built” or compiled in any way.

1.2.2 Dependencies

`Spectro` heavily depends on the list of python packages, that you will need to install:

```
* Python 3.7+
```

and:

```
* adjustText==0.7.3
* astroplan==0.8
* astropy==4.2.1
* astroquery==0.4.3
* ccdproc==2.1.1
* ChainConsumer==0.33.0
* corner==2.2.1
* emcee==3.0.2
* h5py==2.10.0
* julia==0.5.6
* lmfit==1.0.2
* lxml==4.6.3
* matplotlib==3.4.3
* mendeleev==0.9.0
* numba==0.53.1
* numdifftools==0.9.39
* numpy==1.20.1
* pyGPS==1.3.5
* PyQt5==5.15.6
* pyqtgraph==0.12.1
* pytz==2021.1
* scikit_learn==1.0.1
* scipy==1.6.2
```

(continues on next page)

(continued from previous page)

```
* seaborn==0.11.1
* statsmodels==0.12.2
* ultranest==3.3.3
```

The exact versions of the packages listed here are not obligatory (it was generated from the working build), and the code can work with some old/new versions. However, there are sometimes inconsistencies with previous versions of the packages, e.g. for `matplotlib` and `chainconsumer`. For convenience, this package list is automatically stored in `requirements.txt`, therefore you can simply use:

```
$ pip install -r /path/to/requirements.txt
```

1.2.3 Config

(This step can be skipped)

Before first use it can be useful to config some variable inside the code which provide the path to the external databases, e.g. IGMspec. The paths are kept in the file `svviewer\config\config.ini`. In this file some dependencies and variables automatically store during the working progress, but you can also edit it by hands on your own risk.

1.2.4 Julia

Nevertheless that `Spectro` can work in the pure python installation, the performance of line profile fitting routines can be significantly enhanced by using `Julia` language. For this you need to install `Julia` (<https://julialang.org/>) and `PyJulia` package in `Python`. The `Spectro` automatically check that `Julia` is installed and used it. Once `Julia` is installed you can choose between `Python` and `Julia` to perform line profile fitting using settings available in *Preferences menu* menu (F11 or View/Preferences in the main menu).

1.3 Graphical User Interface

`Spectro` package contains **svviewer** - the graphical user interface for the interactive spectral data analysis. It is located in the separate folder `svviewer`.

1.3.1 Run

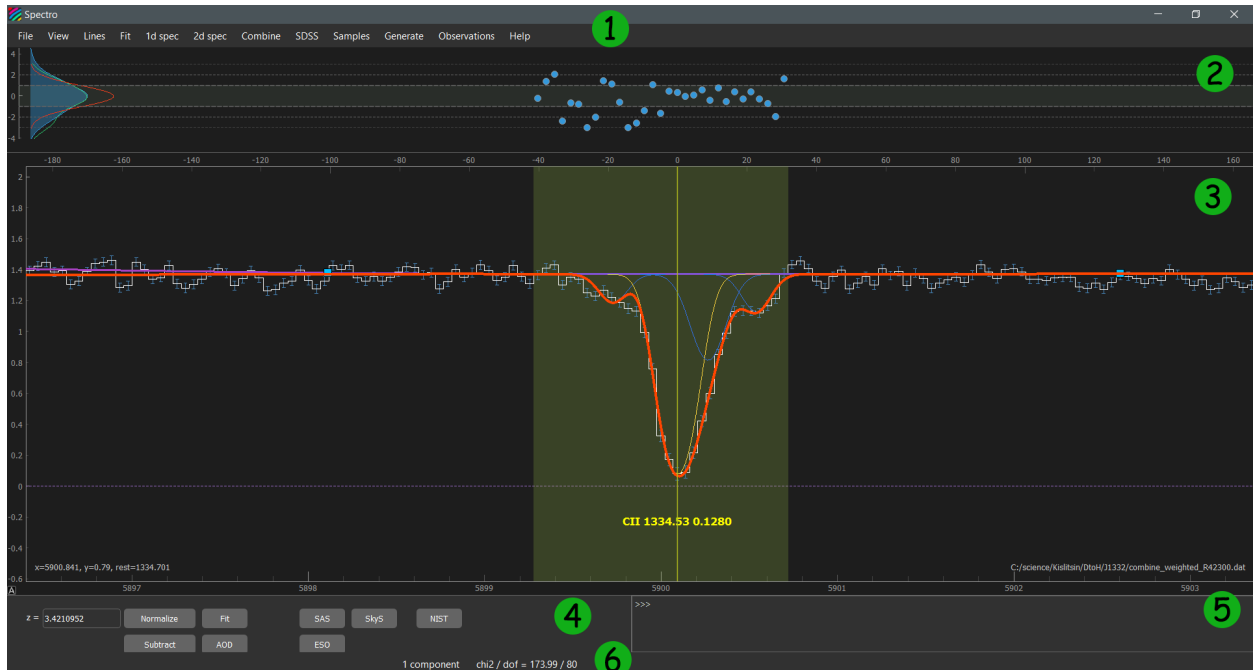
`svviewer` GUI can be run using `path/to/spectro/svviewer/__main__.py`, for example:

```
$ path/to/python.exe path/to/spectro/svviewer/__main__.py
```

Alternatively, one can use convenient IDE to run the `__main__.py`, for example `PyCharms`, `Spider`, `MScode`, etc.

1.3.2 Main window

The main window of **svviewer** is typically looks like



where numbers highlight the following parts of GUI:

1. *Main menu*
2. *Residuals panel*
3. *Spectrum panel*
4. *Control panel*
5. *Console*
6. *Status bar*

1.3.3 Main menu

For the detailed descriptions see [Main menu](#)

1.3.4 Residuals panel

It shows the residuals between spectrum and the fit model. It has shared x-axis with the Spectrum panel. **Residuals panel** can be shown/hide by pressing F4 or `view/Residuals` in the Main Menu. The blue area and green line in the left of this panel show the kde of the residual distribution, that is calculated using pixels from the whole spectrum and from the view window only, respectively. The red line shows the gaussian function with unit dispersion, that should be approaching in case of good fit. Note: that consistency between blue and red lines are not necessary means reliable fit, since you also need to control the structure in residuals.

1.3.5 Spectrum panel

The main interactive window to work with the spectra. It shows the spectrum and the fit model and different graphical objects suitable for the spectral analysis. It is built on the base of `PlotWidget` class from the `pyqtgraph` package. The interaction process is quite essential, but the description of the main features of the interaction module can be found in the `pyqtgraph` manual, for example [mouse interaction](#).

1.3.6 Control panel

This panel contains some useful buttons for the spectral analysis, including:

1. **Redshift line edit field:** input the redshift for the line markers shown in Spectrum panel. The redshift also can be changed either by moving line markers `SHIFT``+ drag` or by using ```LEFT` and `RIGHT` keys on the keyboard (and also with `SHIFT+LEFT` and `SHIFT+RIGHT`).
2. **Normalize:** switch between normalized and raw views (where continuum is explicitly shown), i.e. `NORMALIZED = I/I_0`
3. **Subtract:** switch between subtracted and raw views, i.e. `SUBSTARCTED = I-I_0`
4. **AOD:** switch between aparent optical depth and raw views, i.e. `AOD = np.ln(-I/I_0)`
5. **Fit:** start the fit using LM minimization routine.
6. **SAS:** open spectrum in SDSS science archive server (only for SDSS spectra)
7. **SkyS:** open spectrum in SDSS sky server (only for SDSS spectra)
8. **ESO:** open spectrum in ESO data archive (**not available yet**)
9. **NIST:** show the list of the possible lines available in The NIST database which can be located within the `Region` object within Spectrum panel taking into account specified redshift.

1.3.7 Console

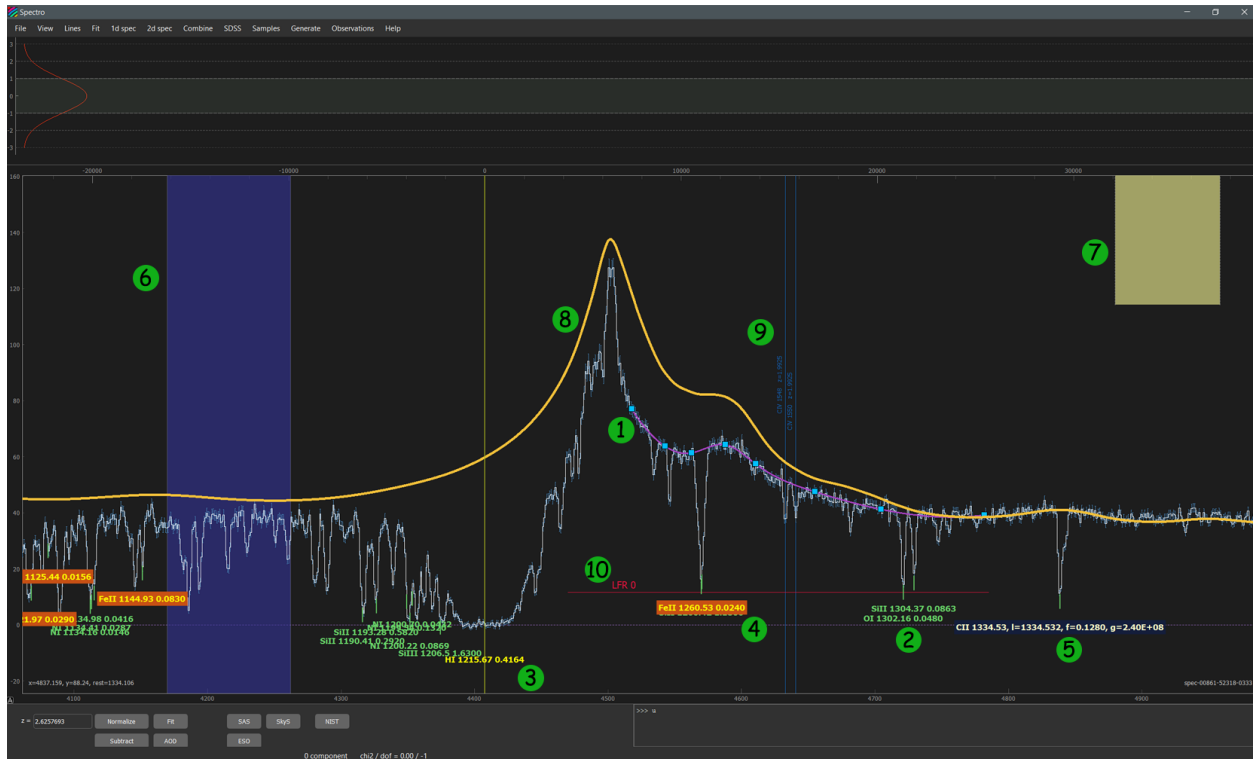
It allows to input commands moslty concerned with GUI management. For the detailed descriptions see [Console](#).

1.3.8 Status bar

Status bar shows some messages and indicate some useful numbers that are can be instructive during the fitting process, e.g. the number of the current fit component, χ^2/dof , etc

1.4 Objects in GUI

There are several interactive objects in the sviewer that can be move by mouse interaction to alleviate and speed up the spectral analysis. This objects are shown in this Figure and their description is given below



1.4.1 B-spline continuum

The continuum that is constructed using B-spline with the point chosen by holding **b** key (see [Tutorial](#) and [Keyboard](#)). It is marked as object 1 on the Figure.

1.4.2 Line labels

Line labels (objects 2, 3, 4, 5 in Figure) indicate possible positions of the absorption/emission lines corresponding to atomic/molecular transitions. The vertical position of the line label is tight to the spectrum, while the horizontal position can be changed either by moving line labels (using drag by mouse and keyboard) or by setting appropriate redshift in the redshift field in the [Control panel](#). One can add/remove line labels using [Console](#) commands: `show <species>` and `hide <species>`. There are useful representations and interaction actions with line labels that ease the spectral analysis:

- **SHIFT + DRAG**: move the line label by changing its redshift.
- **LEFT ARROW** and **RIGHT ARROW**: move the line label by changing its redshift.
- **SHIFT + LEFT ARROW** and **SHIFT + RIGHT ARROW**: move with wider step.
- **SHIFT + LEFT CLICK**: set line label as a reference line (see object 3), the top axis will be scaled with velocity offset from this line.
- **CRTL + LEFT CLICK**: delete line label.
- **h + LEFT CLICK**: highlight all line labels of the species (including level) that correspond to this line, similar to writing in the [Console](#): `high <species>`. Highlighted line label is marked by object 4.
- **ALT + LEFT CLICK**: show extended information about this line (see object 5).
- **K + LEFT CLICK**: add line to the current (highlighted) group of fixed absorption lines.

1.4.3 Regions

Regions (objects 6 and 7 in Figure) are convenient graphical objects to select certain wavelength range of the spectrum. It can be used in several parts of the spectral analysis. There are two representations of the regions in extended (object 6) and minimized (object 7) form. Allowed interactions:

- `r` + LEFT MOUSE BUTTON: create a region.
- DOUBLE LEFT CLICK on region: switch between extended and minimize representation.
- SHIFT + LEFT MOUSE BUTTON: modify (shift and changing size) the region
- CTRL + LEFT CLICK: delete region

1.4.4 Composite spectrum

Composite spectrum (object 8 in Figure) indicates a composite spectrum of QSO. Currently, there are 3 types of the composite spectrum that is shown one after another. Interactions:

- CTRL + `q`: show/hide composite spectrum.
- SHIFT + DRAG: drag composite spectrum, during this its normalization and redshift is changed.
- CTRL + LEFT CLICK: hide composite spectrum.

1.4.5 Fixed Line Labels

Fixed Line Labels (object 9 in Figure) are separate (from objects 2-5) line labels to mark the absorption lines with fixed position, i.e. those positions (i.e. redshifts) do not change with changing of the main redshift in the redshift panel. Usage of this line labels is a convenient way to mark nuisance absorption lines seen in the spectrum (mostly several doublets that). These lines are usually combined in the groups that allow to add/exclude certain line. Interactions:

- `u` + LEFT MOUSE BUTTON: create line group.
- DOUBLE LEFT CLICK on text label: highlight doublet.
- SHIFT + LEFT MOUSE BUTTON on text label: shift (change the redshift).
- ALT + LEFT CLICK on text label: delete line from the group.
- CTRL + LEFT CLICK on text label: delete line group.

1.4.6 Line flux residuals (zero levels)

Line flux residual (object 10 in Figure) is the indicator of zero level, that can be a part of the fit profile construction, to model the partial coverage. Interactions:

- `p` + LEFT CLICK: create line flux residual object.
- SHIFT + LEFT MOUSE BUTTON on text label: shift.
- CTRL + LEFT MOUSE BUTTON on text label: delete.

1.5 Tutorial

Here we provide a brief tutorial to start fitting with the `sviewer` GUI.

1.5.1 Loading the spectrum

The spectrum can be loaded in GUI by several ways:

- **Drag and drop method:** You can load several files in one drop.
- `File/Import spectrum...` in Main menu
- `File/Open...` in Main menu: In that case it load the spectra specified in `.spv` file, which contains saved progress of data analysis.
- `File/Import list...` in Main menu: import a list of spectra from the file with list contains of path to spectra.
- `File/Import folder...` in Main menu: import all spectra from the specified folder.

The basic `svviewer` spectral format is plain ASCII file, which stores spectrum in column-like representation with the following order: `<wavelength> <flux> <uncertainty>`. The spectrum do not necessary need to have `<uncertainty>` column, as well as can read the `<continuum>` provided in the additional forth column. The header of ascii file should be commented out by `#`.

Additionally there is possibility to load the spectrum in FITS format. The program can automatically recognize several FITS format produced by standard reduction routines, such as UVES popler, SDSS, etc.

1.5.2 Constructing continuum

One of the first part of the absorption line analysis is constructing of the unabsorbed continuum. In most partical cases, the continuum is not well defined a priori, and can be considered as some fixed normalization of the spectrum (fit profile) or as a nuisance parameter. There are several way to construct continuum:

- **B-spline: continuum is constructed using B-spline interpolation between data points that can be created using mouse interaction**
 - `b + LEFT CLICK`: add point for B-spline at the cursor position.
 - `b + RIGHT CLICK`: remove the closest point of B-spline at the cursor position.
 - `b + MOUSE REGION`: remove all points of B-spline within selected region.

B-spline is automatically recalculated at each change in the data points array.

- **Iterative Smoothing:** for quick constion of smooth continuum press `q`. For more options see **Continuum widget**
- **Savitsky-Golay interpolation:** available in **Continuum widget**
- **Chebyshev polinomials:** available in **Continuum widget**

Fine tuning options and actions to construct and modify the continuum is available in **Continuum widget** that can be opened iether `ld spec/Continuum...` in Main Menu or pressing `CTRL + c`.

After continuum is constructed one can **normalize/denormalize** the representation of the spectrum by pressing **Normalize** button in *Control panel* or using `n` key.

1.5.3 Selecting fit regions

To perform the fit it is necessary to select the spectrum pixels in which the spectrum will be compared with the fit model. This can be done by:

- `s + MOUSE REGION`: select points within region drawn by mouse using Left button
- `s + SHIFT + MOUSE REGION`: select points to all exposures within selected region

- `d + MOUSE REGION`: deselect points within region drawn by mouse using Left button
- `d + SHIFT + MOUSE REGION`: deselect points to all exposures within selected region

The representation of selected pixels (regions, points, lines) can be changed in **Preferences widget** (`View/Preferences...` in Main menu or by pressing `F11`)

1.5.4 Making fit model

The fit model should be defined in the **Fit model** widget, which can be opened either by `Fit/Fit model...` in Main Menu or pressing `F3`. Detailed description of the **Fit Model** widget is given in [Fit model](#)

After setting/modification of fit model one can update (if it was not update automatically) the fit by pressing

- `f`: this will construct the model profiles in the selected regions only.
- `SHIFT + f`: this will construct the model profiles for all available line in the spectrum.

1.5.5 Fitting

There are two available fit routines:

- Minimizing likelihood using Levenberg-Marquard method. The uncertainties on the fitting parameters estimate from the covariance matrix approach. This fit is performed by the pressing `Fit` button in [Control panel](#). There is a possibility to choose a particular set of the parameters that will be varied during the fit, inside **Fit parameters** widget, which can be opened either by `Fit/Fit parameters...` in Main Menu or pressing `F4`.
- Bayesian approach by Monte Carlo Markov Chain (MCMC) technique with a set of Samplers. The options and control is provide in **MCMC widget**, which can be called using either by `Fit/MCMC Fit...` in Main Menu or pressing `F5`. The detailed description is provided in [Bayesian inference with MCMC](#)

1.5.6 Viewing results

The fit result can be provided inside **Fit results** widget, which can be called either by `Fit/Fit results...` in Main Menu or pressing `F6`. There various option for the output, including plain text, PyQt widget table and latex table.

The fit profiles can be constructed in the publish-ready representation with `matplotlib` by using **Plot Lines** widget, which can be called either by `View/Plot line profiles...` in Main Menu or pressing `F7`. The detailed description of **Plot profiles** widget is provided in [Plot line profiles widget](#)

1.6 Keyboard

The keyboard provide the easy access to the GUI instruments. Most of the useful commands can be assessed form the keyboard.

1.6.1 Keybord bindings

- **F1**: show help/Howto.
- **F2**: show/hide list of exposures widget.
- **F3**: show/hide panel to construct fit model

- **F4**: show/hide choose fitted parameters panel.
- **F5**: show/hide MCMC widget.
- **F6**: show/hide fit results widget.
- **F7**: show/hide plot widget.
- **F8**: show/hide residuals panel.
- **F9**: show/hide 2d spectrum panel.
- **F11**: show/hide Preferences menu.
- **a**: add component:
 - **a + LEFT CLICK** : add component (same as the current) to the model fit at the position
 - **a + CTRL** : remove the current component from the model fit
- **b**: b-splain:
 - **b** : add b-splain point (if key is quickly pressed)
 - **b + LEFT CLICK** : add b-splain point
 - **b + RIGHT CLICK** : remove nearest b-splain point
 - **b + <select region>** : remove all b-splain points in selected region
- **c**: component mode:
 - **c** or **c + RIGHT ARROW** : show/select next component
 - **c + LEFT ARROW** : show/select previous component
 - **c + CTRL** : continuum fit options window
 - **c + SHIFT** : change show component mode (none, one, all)
 - **c + LEFT CLICK** : shift component center at the position (line indicator have to be selected)
 - **c + MOUSE WHEEL** : increase/decrease b parameter of the component
 - **c + <mouse drag>** : increase/decrease column density of the component
- **d**: deselect data points:
 - **d + <select by mouse>** : deselect data points
 - **d + SHIFT + <select by mouse>** : deselect data points for all exposures
- **e**: select exposure:
 - **e** : select next exposure
 - **e + KEY_UP / KEY_DOWN** : select next/previous exposure
 - **e + KEY_RIGHT / KEY_LEFT** : select next/previous exposure
 - **e + <click on exposure>** : choose exposure which is clicked
 - **e + CTRL** : remove exposure
 - **e + <select by mouse>** : select area to calculate flux, FWHM and luminosity within selected region (e.g for emission line)
- **f**: construct fit
 - **f** : show fit (only lines nearby fitting points)

- **f + SHIFT** : show full fit (all available lines in the full wavelength range of exposures)
 - **f + CTRL** : <not set yet>
- **g**: fit gauss
 - **g + CTRL** : show composite Galaxy spectrum (several templates available)
- **h**: highlight labels
 - **h + LEFT CLICK** on line label : highlight all the line labels corresponding to species of selected line label
- **j**: show the dispersion of the model fit from MCMC
- **k**: add line to the fixed line group
 - **k + MOUSE LEFT CLICK** on line Label : add line label to highlighted fixed line group.
- **l**: choose *lya*
 - **l + LEFT CLICK** : set redshift for *lya* line to be at the position of mouse
- **m**: smooth spectrum
 - **m + MOUSE WHEEL FORWARD** : increase smoothness
 - **m + MOUSE WHEEL BACKWARD** : decrease smoothness
- **n**: normalize/unnormalize the spectrum
- **o**: open / change UVES setup
 - **o** : change UVES setup
 - **o + CTRL** : open file
- **p**: partial coverage and polynomial fitting
 - **p + two LEFT CLICKs** : create partial coverage line
 - **p + KEY_UP / KEY_DOWN** : fit selected points with polynomial higher degree
 - **p + KEY_RIGHT / KEY_LEFT** : fit selected points with polynomial lower degree
- **r**: select region:
 - **r + <select by mouse>** : add region (how to work with regions see Tutorial)
 - **r + SHIFT** : force top x axis to show restframe wavelength
- **s**: select data points:
 - **s + <select by mouse>** : select data points
 - **s + SHIFT + <select by mouse>** : select data points for all exposures
 - **s + CTRL** : save to recent file
- **t**: show fit results:
 - **t + CTRL** : show/hide fit result window
- **q**: continuum
 - **q** : make continuum in window using smoothing
 - **q + CTRL** : show composite QSO spectrum (several templates available)
- **u**: find doublet:

- **u + LEFT CLICK** : add line to doublet guess
- **v**: change view of spectra (steps/points/lines + uncertainties)
- **w**: width of region:
 - **w + <select by mouse>** : select area to calculate equivalent width of absorption line. Continuum should be set for width calculation!
 - **w + SHIFT + <select by mouse>** : select area to calculate equivalent width of absorption line, subtracting fit model. (i.e. respective fit model, but no to continuum)
 - **w** : hide w-region
- **x**: select bad pixels:
 - **s + <select by mouse>** : select bad pixels
 - **s + SHIFT + <select by mouse>** : unselect bad pixels
- **y**: likelihood region:
 - **y + LEFT CLICK** on line label : show a region of likelihood in (logN, b) parameter space for selected line label. The grid range is taken from fit model window as .
- **z**: zoom mode:
 - **z + <select by mouse>** : zoom into region
 - **z + CTRL** : return to the previous view

shift:

1. when shift is pressed you can shift absorption pointers using mouse

1.7 Console

1.7.1 Console command

The console provide access to the number of text commands that alleviate the spectrum investigation and analysis. Some action can be accessed only by console. Additionally it is used as an output window to quickly show the results of some procedures.

The list of available commands:

- **<species>**: print the available lines of the species (e.g. H I, Si IV, H2j0 etc) in spectro database.
- **show <species>**: add the line marks of available lines of the species in the spectrum panel. The line marks will attach to the spectrum.
- **high <species>**: highlight the line marks corresponding to the species in the spectrum panel.
- **hide <species>**: hide all the line marks corresponding to the species in the spectrum panel.
- **z <float>**: set redshift of the line marks to be <float>.
- **load <filename>**: load the file with <filename> from the temp folder.
- **fit <options>**: start fit of the model. <options> can be:
 - **comp**: fit only selected component.
 - **mcmc**: load MCMC window to run MCMC fitting procedure.

- **grid**: run profile likelihood estimation on the grid of the paramaters. Be careful, it is too long for number of paramaters more than 3.
- **cont**: fit continuum.
- **logN <species>**: calculate the total column densities along the components of the fit model.
- **me <species> <logNHI>**: calculate the metallicity of the <species> (based on the total column density along components) assuming the total HI column density, <logNHI>.
- **depletion <species1> <species2>**: calculate the depletion of the <species1> relative to <species2>.
- **abundance <species> <logNHI> <me>**: calculate expected column density of the <species> assuming the total HI column density, <logNHI>, and metallicity, <me>.
- **x <num1> <num2>**: adjust the x-axis scale to <num1>..<<num2>.
- **y <num1> <num2>**: adjust the y-axis scale to <num1>..<<num2>.
- **rescale <arg> <float>**: rescale the spectrum/continuum/error by <float> factor. It applies with a window region. The <arg> specified what is rescaled:
 - **y**: rescale flux.
 - **err** or **unc**: rescale only pixel uncertainties.
 - **<no arg>**: rescale both flux and pixel uncertainties.
 - **x**: rescale wavelengths by constant factor.
 - **z**: rescale wavelengths by redshift factor.
 - **spline** or **cont**: rescale B-continuum.
- **shift <arg> <float>**: shift spectrum by <float>, $l = l + \langle \text{float} \rangle$. If <arg>=v than the shift is in velocity space, i.e. $l = l * (1 + \langle \text{float} \rangle / c)$.
- **set <arg1> <arg2> <float>**: set flux or uncertainties or continuum to be equal <float> value. <arg1> should be set **flux** or **unc** or **cont**. <arg2> can be:
 - **<not specified>**: apply for a whole range of the current exposure.
 - **screen** or **window** or **view** or **disp**: apply only within window range of the current exposure.
 - **region** or **regions** or **reg**: apply only for the regions.
- **cont <float>**: set continuum to be <float> for the whole range
- **divide <spec1> <spec2>**: divide <spec1> on <spec2>.
- **subtract <spec1> <spec2>**: subtract <spec2> from <spec1>.
- **apply <arg>**: apply transformation of x-axis. <arg> can be:
 - **vac**: correction to vacuum.
 - **helio <float>**: correction to baryocentric wavelengths, specified by <float> velocity.
 - **restframe**: convert wavlnghths for restframe, using redshift from z panel.
- **ston**: print signal to noise of the spectrum at the region items.
- **stats**: print signal to noise, dispersion of the pixels in the spectrum at the region items.
- **level**: print level of the spectrum in the regions (optimize to measure the line flux residuals).
- **ew**: print equivalent widths of the lines in line regions.
- **Tkin**: print the estimate of the kinetic temperature using $J=1/J=0$ population of the H2.

1.8 Main menu

The main menu contains some, but not all, commands. Some commands are duplicated or exclusively offered by either *Keyboard* or *Console* interactions.

Here are the description of the command in the main menu:

1.8.1 File:

- **New:** Clear the workspace (removing all the objects, fit model details, etc).
- **Open:** Open a file with saved working progress. Typically this files have .spv extension.
- **Save:** Save working progress to a file (last one). Be careful, if you by mistake overwrite the important file, than you can find its backup at `temp/savedfile.spv`.
- **Save as:** Save working progress to a file with choise of the preferences and filename.
- **Import spectrum:** Import spectrum from the file. It can automatically understand several formats, including ascii, fits (from various instruments)
- **Import 2d spectrum:** Currently works only with ESO formats.
- **Import list:** Import the spectra from the file contains the list of paths.
- **Import folder:** Import all the spectra in specifeid folder.
- **Export spectrum:** Export 1d spectrum to file.
- **Export 2d spectrum:**
- **Import data:** Export some data (e.g. continuum, fit model) to the files

1.8.2 View:

1.8.3 Lines

1.8.4 Fit

1.8.5 1d spec

1.8.6 2d spec

1.8.7 Combine

1.8.8 SDSS

1.8.9 Samples

1.8.10 Generate

1.8.11 Observations

1.8.12 Help

1.9 Fit model

to be described

1.9.1 Fit model widget

to be described

1.10 Bayesian inference with MCMC

to be described

1.10.1 MCMC widget

to be described

1.11 Plot line profiles widget

to be described

1.12 Preferences menu

Some of the global preference can be accessed/changed using Preferences menu. To open it please use `F11` or `View/Preferences` in the main menu.

1.12.1 Appearance

Controls appearance of the general graphical elements in the Spectrum panel:

- **Spectrum view:** the representation of the spectrum. Can be `steps`, `lines`, `points`. Each one can also include `uncertainties + errors`.
- **Fitting pixels view:** the representation of the pixels that is chosen for the analysis. Can be `region`, `points` and `colors`.
- **Line labels view:** the representation of the regular line labels (see *Objects in GUI*). Can be `short`, `infinite` (the same as fixed line labels).
- **show inactive exp:** show inactive exposures in the spectrum panel using gray-out representation.
- **f in line labels:** show oscillator strengths in all line labels.

1.12.2 Fit

Provide setting that control the fit profile calculation and appearance:

- **Line profile type:** choose the type of the construction of line profiles between:
 - **regular:** regular python calculation (this include non-uniform wavelength grid for calculation the sub pixel profiles)
 - **fast:** python calculation with uniform wavelength grid, where number in adjacent Line Edit panel set the number of sub-pixels within the spectral pixel. This is typically faster, than regular fit, but can be inaccurate if the number of sub=pixels is not enough.
 - **julia:** line profiles calculated using `Julia`.
 - **fft:** line profiles calculated using Fast Fourier Transform (for fast convolution with instrumental function). **Be careful, it is not fully tested yet!**
- **tau limit:** set the characteristic limit until which the optical depth is calculated.
- **Fit method:** choose the minimization method for least-squares estimation. This setting pass directly to `minimize` routine in `optimize` package within `lmfit` package. Works only for fit calculation using python.
- **Fit components:** choose the representation of individual fit component (besides the total fit profile). This also access by `C + SHIFT`. Can be:
 - **all:** show all components.
 - **one:** show only current component (use `C` and `C + LEFT/RIGHT ARROW` to switch between components).
 - **non:** do not show components.
- **show fit points:** show explicitly the points in which fit profile is calculated.
- **animate fit:** animate fit minimization, i.e. update the fit each step. **Does not properly work yet!**

1.12.3 Colors

To manage the colors of the graphical objects. To be available in the future versions...