SPAN 7.1.X User manual

Daniele Gasparri

October 15, 2025

Contents

What's New in SPAN 7.1

- Expanded and improved the two component fit of the "Stellar and gas kinematics" task, which is now available both for the stellar and gas kinematics, with four presets: old-young components, metal rich-metal poor components, all templates, and two templates mode, with custom selection of age and metallicity following the work of Rubino et al. 2021.
- Added interactive analysis modes to the Preview. You can press the key F, E, S, D, I, L, P to enter the Fitting mode, Equivalent width mode, Signal-to-noise mode, Differential wavelength and velocity mode, Integrated line flux mode, Line-peak finding, snaPshot. Draw a region for the F, E, S, I modes or click on two points in the D mode to see the results in the HUD, in the terminal window and to automatically save them in logging files in the "SPAN_results" folder. Press again the respective key to exit the selected analysis mode and return back to idle (not applicable to L and P modes, they exit automatically). Press C to cancel all the overlays. Double click resets also zoom and panning.
- Minor bug fixes

More from the version 7.X:

- Interactive Preview frame that shows the selected spectrum directly in the main GUI. In the preview you can:
  - Zoom, pan, and reset with trackpad touches and/or mouse left click/scroll buttons.
  - Know in real-time wavelength, flux, and S/N on every point of the spectrum.
  - Look at rest-frame reference spectral features (OII, HBeta, OIII, HAlpha, CaT) for immediate comparison with your spectrum.
  - Manual Redshift Estimation: shift the selected spectrum with right-click and drag to match the spectral features of your spectrum with the reference lines. Reset the shift with double right-click.
- Consistent Zooming (View menu): unified scaling for fonts, widgets, frames, and Preview to fine-tuning the view on screens with different resolution.
- Extended the preliminary check of the loaded spectra with a quick global S/N evaluation and flagging in the listbox (LOW SNR if S/N <= 5), so you can immediately identify problematic spectra.
- Interactive listbox that allows you to:
  - Select multiple spectra with mouse dragging or with crtl or shift + click
  - The single spectrum or the selected spectra can be reordered or deleted, via right click menu (delete works also with "Canc" button).
  - You can undo the latest operation (i.e. if you accidentally deleted the wrong spectra).
  - If two spectra are selected in the listbox, you can also compare them, via right click –> compare.
  - You can save the modified spectra list in "Edit –> save current spectra list…".
  - Double click on a single spectrum to open an info window with some basic data: name, wavelength range, sampling, mean/median flux, standard deviation and global S/N.
- Markov chain Monte Carlo (MCMC) simulations mode available for stellar parameters estimated with Lick/IDS line-strength analysis, in addition to the griddata linear interpolation and the supervised machine-learning GPR.
- Elliptical (or circular) binning mode for DataCube extraction, allowing you to extract radial

spectral bins following the isophotal profiles of your source.

- SPAN now is also available as a standalone self-installing application for Windows systems (10 and 11), without requiring Python.

Attribution

If you use this software for your research, please cite Gasparri et al. (2025). The BibTeX entry for the paper is:

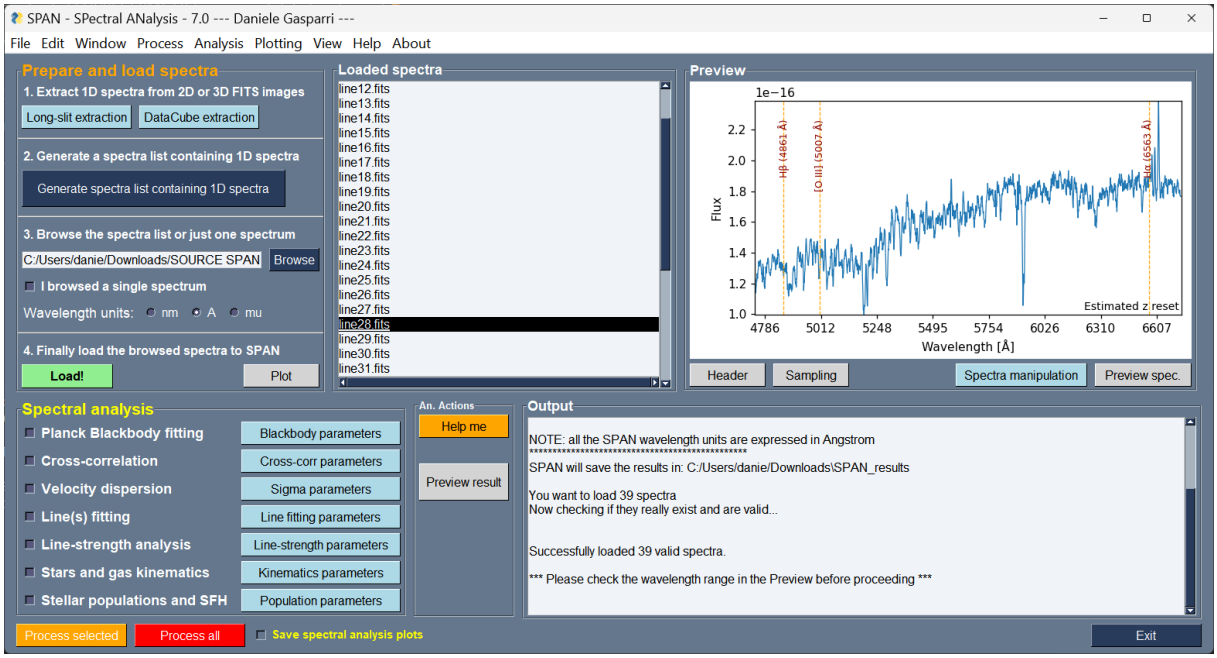
```
@ARTICLE{2025arXiv250801923G,
       author = {{Gasparri}, Daniele and {Morelli}, Lorenzo and
       {Battino}, Umberto and {M{\'e}ndez Abreu}, Jairo and {de
       Lorenzo-C{\'a}ceres}, Adriana},
        title = "{SPAN: A cross-platform Python GUI software for optical
        and near-infrared spectral analysis}",
      journal = {arXiv e-prints},
     keywords = {Instrumentation and Methods for Astrophysics,
     Astrophysics of Galaxies},
         year = 2025,
        month = aug,
          eid = {arXiv:2508.01923},
        pages = {arXiv:2508.01923},
          doi = {10.48550/arXiv.2508.01923},
archivePrefix = {arXiv},
       eprint = {2508.01923},
 primaryClass = {astro-ph.IM},
       adsurl = {https://ui.adsabs.harvard.edu/abs/2025arXiv250801923G},
      adsnote = {Provided by the SAO/NASA Astrophysics Data System}
}
```

Remember also to cite the related software (e.g. pPXF), if you will use it within SPAN.

Purpose

SPAN is a Python 3.X cross-platform graphical interface (GUI) program designed to perform operations and analyses on astronomical fully reduced and wavelength calibrated 1D spectra. SPAN has been developed and optimized to analyze galaxy and stellar spectra in the optical and near infrared (NIR) atmospheric windows. SPAN accepts as input ASCII and FITS spectra files. The 1D spectra files required can be generated also with SPAN, both from long-slit 2D FITS images and 3D data cube (e.g. MUSE data) fully reduced and wavelength calibrated.

SPAN deals with linear sampled spectra, with wavelength in physical units (A, nm and mu). If you don't have linear sampled spectra, SPAN will try to read the spectra, will convert them automatically to linear sampling and will assign a physical wavelength scale, if needed. If these operations fail, your spectra will show a strange wavelength scale in the Preview window on the right or when clicking "Plot". If that is the case, you will need to adjust them with other software before load to SPAN.



Main panel of SPAN as it appears in Linux and Windows systems.

The program has been tested with IRAF-reduced spectra, SDSS spectra, IRTF (also extended version) spectra, SAURON spectra, X-Shooter library spectra, JWST spectra, MUSE, CALIFA, JWST NIRSpec, and WEAVE LIFU data cubes, (E)MILES, GALAXEV and FSPS stellar libraries, and complies with the ESO standard for 1D spectra. SPAN DOES NOT accept ASCII spectra files with Fortran scientific notation, like the PHOENIX synthetic stellar spectra. In this case, you will need to open the file and substitute the scientific notation of flux and wavelength "D" with "E" (you can do this operation even with the embedded text editor of SPAN).

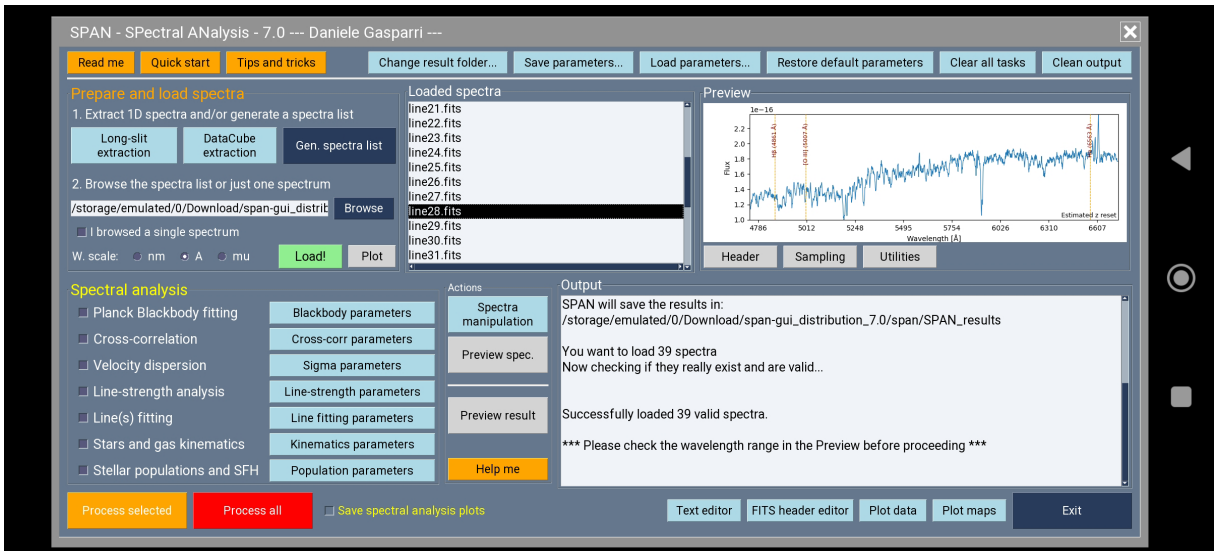
Currently, SPAN considers only the wavelength and the flux, discarding the (potential) column with uncertainties.

Fully cross-platform capabilities

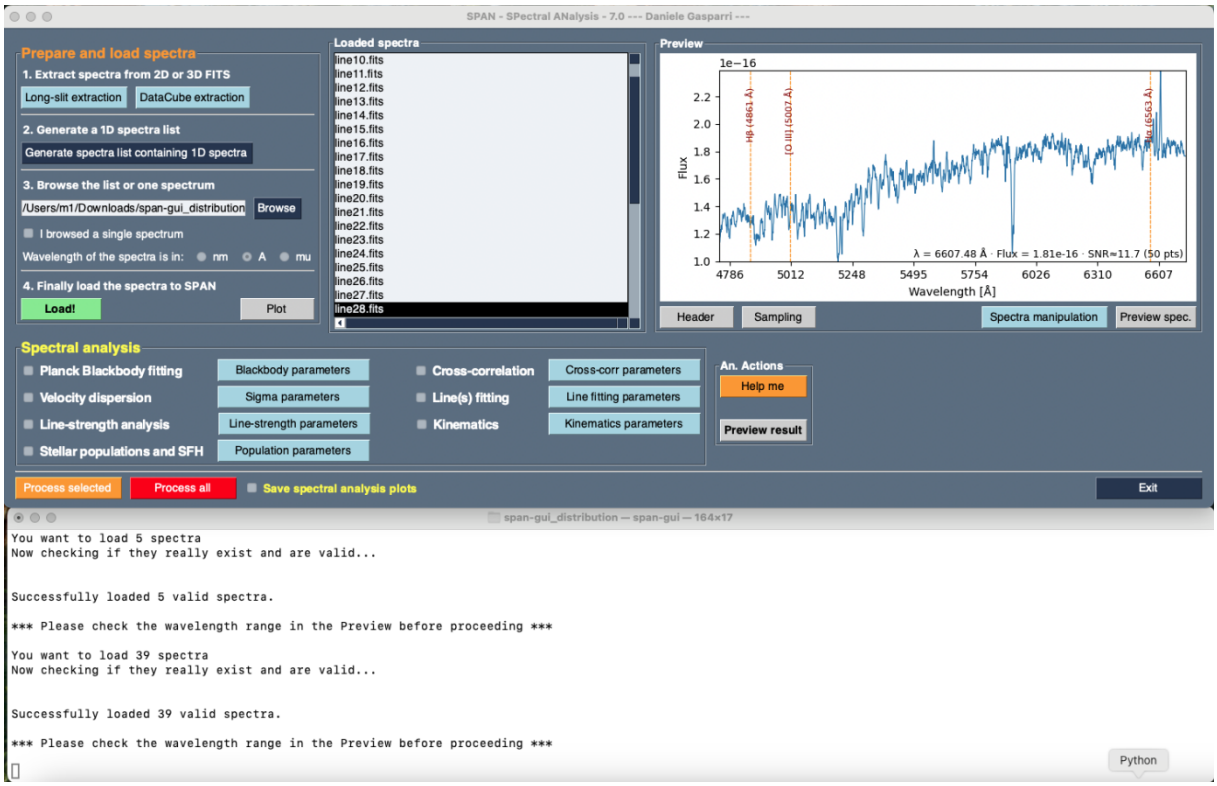
SPAN has been carefully designed and tested to run across different platforms, making it accessible to a wide range of users. While the core functionality remains consistent, there are some minor differences in layout and interaction depending on the operating system:

- **Windows and Linux**: The interface and functionality are nearly identical. Both platforms provide the full set of tools, including the integrated output frame where progress messages and logs are displayed directly within the GUI.
- **macOS**: On macOS, the overall layout of SPAN remains consistent with Windows/Linux, but the integrated output frame has been removed. During testing, this component significantly slowed down performance on macOS systems, so messages are instead displayed in the standard system terminal. Apart from this adjustment, all features of SPAN work exactly as on other platforms.
- **Android** (via Pydroid3 app): SPAN can also be used on Android devices through the Pydroid3 environment (see the README_ANDROID.txt for instructions). The GUI has been adapted for smaller screens: the classic menu bar is replaced by practical buttons placed at the top and bottom of the panel. Some interactive functions of the Preview and Listbox frames, such as right/left mouse click actions, are not available on touch devices. However, the rest of the functionality remains identical to the desktop versions, making Android a fully usable platform for basic analysis tasks.

The images below show the aspect of SPAN on Android and macOS systems.



SPAN running on an Android smartphone (6.67” screen).



SPAN running on an macOS with the standard terminal manually arranged below the GUI.

What do you need to run SPAN

The requirements depend on how you will install SPAN:

- As a Python package, you just need any Python 3.10+ version. Installation via pip will also install the related modules required. Once installed, just type 'span-gui' in your terminal window to start SPAN.
- By downloading and manual compiling the source code. In this case, you need to install via pip the following packages to run:

    1) Numpy
    2) Astropy
    3) Pandas
    4) Matplotlib
    5) Scipy
    6) scikit-image
    7) PyWavelets
    8) joblib
    9) scikit-learn
    10) ppxf
    11) vorbin
    12) certifi
    13) emcee

- As a standalone, self installing Windows application. You need a 64 bit Windows 10 or 11 systems and you do not need Python: SPAN will be installed on your Program files folder as any classical application and will be launched by double click on its icon.
- On Android devices, SPAN requires the Pydroid3 app and the related package repository plugin. You need to install manually the required packages using the embedded pip of Pydroid3, then open the **main**.py file and run it in landscape mode. A more detailed list and version of the packages needed is stored in the "README_ANDROID.txt" file.
- SPAN requires a screen resolution of at least 1600X900, otherwise the panel will be truncated. Optimal resolution for non Android devices: 1920X1080.

Input spectra for SPAN

SPAN can work with just one 1D spectrum, either in FITS or ASCII format, with the first column to be wavelength and the second flux. The wavelength and the flux of the FITS files must be in the primary HDU.

SPAN accepts and processes a list of n 1D spectra, where n must be greater than 1. In order to do this, you need to create and load a text file containing the relative path of the spectra (with respect to the location of the main SPAN program), or the absolute path, and the complete spectra names. The first row of this list file must be commented with # and usually contains something like that: #Spectrum. You can put any type of 1D spectra in this file list, but I strongly suggest to insert spectra with at least the SAME wavelength unit scale. It seems difficult, but don't worry: the button "Generate spectra list containing 1D spectra" will help you to create a spectra list file by selecting a folder containing the spectra you want to process.

You can find example file lists in the example_files directory. They are:

1. xshooter_vis_sample_list_spectra.dat, already preloaded in the main application (you just need to click "Load!"), contains 5 spectra of the central regions of nearby galaxies observed with the VIS arm of ESO XShooter spectrograph at resolution of R = 5000. Sampling is linear and the wavelength units to set ("Wavelength units:") are "nm";
2. ngc5806_bins.dat contains the spatial bins of a spiral galaxy observed with the TNG telescope at resolution FWHM = 3.5 A from 4700 to 6700 A. Sampling is logarithmic and wavelengths are in log(A). SPAN will take care of everything; you just need to set "A" in the "Wavelength units:" option of the "Prepare and load" frame before clicking "Load!".

Quick start

If you installed SPAN as a Python package (pip3 install span-gui), just type in the terminal "span-gui". If you installed SPAN as a standalone application for Windows, just double click on the icon (usually in the Desktop).

At the first run, SPAN will ask you to download the auxiliary SSP spectral templates, which do not come with the Pypi or GIThub distribution for size issues. You can skip the download and SPAN will work, but the spectral analysis tasks devoted to full spectral fitting will use only the SSP sample provided by pPXF (EMILES, FSPS, GALAXEV, and, of course, any of the template that you will provide!). For the Windows app, the templates are already included.

At the first run, SPAN will also ask you to select the location of the SPAN_results folder, that is the folder where ALL the results will be saved.

Once everything is set, press the "Load!" button to load the example files.

The spectra loaded will appear in the upper central frame (the white window). Just select one spectrum with the mouse, then look at the Preview on the right and interact with it. If you want more control on the plot, you can also click on the button "Plot". Close the plot to activate again the main panel. You can analyze the selected spectrum by activating any of the spectral analysis tasks and/or you can modify the spectrum by opening the "Spectra manipulation" panel in the "Preview" frame on the right.

Let's open the "Spectra manipulation" panel and activate one of the many tasks, for example the "Add noise", then we confirm the choice by pressing the "Confirm" button. Now, we are back to the main panel and we press the "Preview spec." button to see the result. The preview will not change because it will always show you the original selected spectrum. If you like it, you can click the "Process selected" button to save this new noisy spectrum (but first you need to close the plot window!). If you press the "Process all" button, you will apply the task selected to all the loaded spectra. The results will be stored in the folder "SPAN_results/processed_spectra", located in the folder you have selected the first time you opened SPAN. The output window is your friend: it will tell you all the things that the program is doing.

Now, let's try something in the "Spectral analysis" frame. We activate the "Line-strength analysis" task and take a look at the parameters to set by clicking the button "Line-strength parameters". We select the "Single index option" and confirm the selection by clicking the button "Confirm". The "Line-strength analysis" window will close automatically and we are back to the main panel. Now, we click the "Preview result" button to see the result of this task. The spectrum does look strange to you? Did you deactivate the "Add noise" task above? If not, the "Line-strength analysis" task is analyzing this noisy spectrum and not the original one. The "Spectral Analysis" frame will consider the spectrum (or the spectra) processed by the tasks activated in the "Spectra manipulation" panel. If you activate 10 tasks, the spectrum processed in this frame will be the sum of all the activated tasks in the "Spectra manipulation" panel. If you don't activate any task, the spectrum processed will be the original loaded.

If you activated so many tasks that the entropy of the program tends to infinite, don't panic. Just click on the menu: "Edit –> Clear all tasks" to start from fresh. If you want to restore also the default parameters, you can do it with: "File –> Restore default parameters". If you want to save your personal parameters, you can do with: "File –> Save parameters" and load them

again whenever you want.

If you want to change the location of the SPAN_results folder, you can do it with: “Edit –> Change result folder…”

You can play with other sample 1D spectra by loading the ready to use spectra list files provided with SPAN, for example the “ngc5806_bins.dat” located in the “example_files” folder. This spectra list contains 39 1D spectra in the optical window of the galaxy NGC 5806. Just browse this spectra list in the “3. Browse the spectra list or one spectrum” section of the “Prepare and load spectra” frame.

General description and usage

SPAN is composed by a main graphical window that shows most of the spectral analysis tasks that the user can perform on 1D spectra. In this window you will find two main panels: the top and the bottom one.

The upper panel

The top panel is divided in three frames. Here is a description of each one.

Prepare and Load Spectra frame

Any operation begins within the upper-left frame, called “Prepare and load spectra”. There are four basic steps to load the spectra to SPAN and start the analysis.

1. **(Optional) Extract spectra from 2D or 3D fits:** This step is mandatory if you do not still have the 1D spectra needed by SPAN. It allows you to extract 1D spectra either from 2D fully reduced fits images or 3D fully reduced fits images, i.e. datacubes.
    - If you have 2D fits of long-slit spectra with the dispersion axis along the X axis of the image, press the “Long-slit extraction” button. There you can fit the trace, correct for distortion and/or slope and extract a 1D spectrum or a series of 1D spectra binned in order to reach a Signal to Noise (S/N) threshold;
    - If you have MUSE, CALIFA, WEAVE LIFU or JWST NIRSpec IFU datacubes, press the “DataCube extraction” button. To achieve the extraction, SPAN uses routines inspired to the famous GIST Pipeline (Bittner et al. 2019). These extraction routines will save the 1D spectra in the “SPAN_results” folder and a spectra file list in the SPAN_results/spectra_lists folder, ready to be loaded.
2. **Generate a spectra list with 1D spectra.** If you already have 1D spectra stored in a folder (and the relative subfolders, if any), you should click on the button “Generate a spectra list containing 1D spectra”. You should then browse the folder where you stored your spectra. SPAN will read all the spectra contained in the selected folder and in any eventual subfolder and will create an ASCII file with their names and paths. The spectra list generated will be automatically loaded in the “Browse the spectra list or one spectrum”. In case you want to load just a single 1D spectrum, you can skip this step.
3. **Browse the spectra list or one spectrum.** If you generated a spectra list in the previous step, this has been automatically loaded here. In this case, you should only select the wavelength units of the spectra contained in the spectra list. It is therefore important that all your spectra share the same wavelength units. It doesn’t matter whether they are linearly or logarithmically rebinned, SPAN will read them correctly as far as you select the correct wavelength units. In case your spectra list is already in your device (i.e. you skipped the step 2.) you should browse it, then select the right wavelength units of the spectra. In case you just want to load a single 1D spectrum, just browse the spectrum and activate the option “I browsed a single spectrum”.
4. **Finally load the spectra to SPAN.** This step is self explicative. Once you browsed the spectra list or the single spectrum and set the right wavelength units in step 3., here you need to press the “Load!” button to effectively load your spectra (or a spectrum) in the listbox on the right. By loading the spectra list or just a single spectrum, SPAN

performs some automatic operations in order to ensure that all the spectra loaded are valid and their path exist in your computer. Moreover, when loading a spectra list, a rough estimation of the global S/N is performed on each spectrum. Spectra with low (<=5) global S/N are automatically flagged as "LOW SNR" in the listbox. This is just a flag, a reminder to you that some spectra might not be suitable for complex spectral analysis. These automatic checks are very quick, requiring usually 20 seconds for 5000 spectra, but if you are absolutely sure that they are all valid and you don't need a first quality check, you can cancel any time this automatic check: SPAN will trust you and will load the whole list anyway.

Loaded Spectra

In this central frame you will find the listbox populated by your loaded spectra. Once you select a spectrum, the Preview on the right will show it. Please, check the wavelength scale and that everything is ok. If you have loaded a spectra file list and not just one spectrum, this listbox becomes interactive. Here are the features available:

- Select multiple spectra with mouse dragging or with ctrl or shift + click
- The single spectrum or the selected spectra can be reordered or deleted, via right click menu (delete works also with "Canc" button).
- You can undo the last operation (i.e. if you accidentally deleted the wrong spectra).
- If two spectra are selected in the listbox, you can also compare them, via right click –> compare.
- You can save the modified spectra list in "Edit –> save current spectra list…".
- Double click on a single spectrum to open an info window with some basic data: name, wavelength range, sampling, mean/median flux, standard deviation and global S/N.

WARNING: The global S/N reported in the quick info box is computed following a robust procedure inspired by Cappellari (2017, pPXF manual). The spectrum is first smoothed with a wide median filter (≈150 Å) to estimate the continuum. The residuals between the observed spectrum and the smoothed continuum are then used to estimate the noise level through the semi-interquartile range, defined as half the difference between the 84th and 16th percentiles of the residual distribution. The signal is defined as the median flux of the spectrum. In spectra with many absorption or emission lines, this value could be underestimated.

Preview

Since version 7, SPAN embeds a dynamical Preview window in the main interface to provide a fast way to visualize, read information and perform essential operations to help you have an idea of the loaded spectra.

What you can do in the Preview

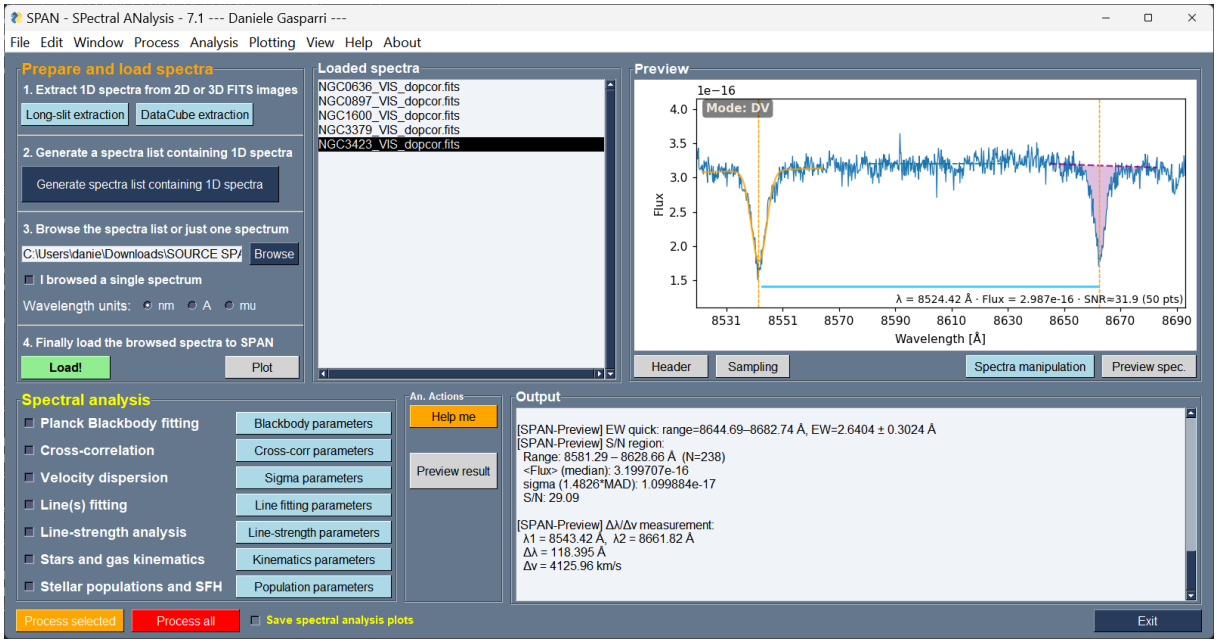
- **Mouse over:** real-time info, such as wavelength, flux and S/N in a +-25 pixel region around the mouse pointer. Values are displayed in the Heads-Up Display (HUD) in the bottom right of the preview.
- **Left-click + drag:** pan
- **Scroll button or trackpad touches:** zoom into the spectrum
- **Press F, then left-click OVER the spectrum and drag**: draw a region containing one emission or absorption line. SPAN will automatically fit the strongest line found in the region,

both in emission and absorption with a gaussian function. In the HDU and in the terminal window you can read some basic values, such as: type of the lime, wavelength range selected, wavelength peak, FWHM, equivalent width, flux, amplitude (intensity), dispersion, and an estimation of the S/N of the line. These values are automatically stored in the "span_preview_fits.log" file in the "SPAN_results" folder in CSV format. Press F again to exit the fit mode

- **Press E, then left-click and drag:** draw a region around an emission or absorption line to automatically measure the equivalent width. The value will be displayed in the HUD, in the terminal and automatically saved in the "span_preview_EW.log" file in the "SPAN_results" folder in CSV format. Press E again to exit the equivalent width mode
- **Press S, then left-click and drag:** draw a region on the continuum to automatically measure the S/N. The value will be displayed in the HUD, in the terminal and automatically saved in the "span_preview_SNR.log" file in the "SPAN_results" folder in CSV format. Press S again to exit the S/N mode
- **Press D, then left-click and drag:** Click on two points while pressing the D key to automatically measure the differences in wavelength and velocity. The values will be displayed in the HUD, in the terminal and automatically saved in the "span_preview_deltav.log" file in the "SPAN_results" folder in CSV format. Press D again to exit the distance mode
- **Press I, then left-click and drag:** draw a region around an absorption or emission line to automatically measure the integrated flux with respect to the mean continuum level. The values will be displayed in the HUD, in the terminal and automatically saved in the "span_preview_intflux.log" file in the "SPAN_results" folder in CSV format. Press I again to exit the integrated flux mode
- **Press L:** automatically find the most prominent peaks in the spectrum. Useful to use in combination with the visual redshift estimation.
- **Press P:** automatically take a snapshot of the current preview and saves it in the "SPAN_results/preview_snapshots" folder
- **Press C:** cancel all the overlays
- **Double left-click:** reset the preview
- **Right-click + drag:** shift the spectrum in wavelength to match the reference lines in the preview and perform a "by eye" estimation of the redshift. This function deserves a better description which I am providing below
- **Double right-click:** reset the shift

General Controls

Action	Mouse / Key	Description
Hover display	Mouse hover	Shows wavelength λ, flux, and S/N.
Zoom in/out	Scroll wheel / trackpad	Zooming. Different zooming modes with crtl or shift + scroll.
Pan (move view)	Left-click + drag	Moves the current viewport.
Reset view	Double left-click	Restores default zoom and clears overlays.
Redshift by eye	Right-click + drag	Shifts the spectrum horizontally to match line references.



Interactive preview modes applied and terminal output

Active Analysis Modes

Key	Mode	Mouse Action	Description	Output / Log
F	**Line Fitting**	Left-drag	Fits a single Gaussian to emission or absorption features.	`span_preview_linefit.log`
E	**Equivalent Width**	Left-drag	Measures EW using a straight-line pseudo-continuum between the selection edges.	`span_preview_ew.log`
S	**Signal-to-Noise**	Left-drag	Computes mean S/N ratio in the selected wavelength interval.	`span_preview_sn.log`
D	**Δλ / Δv Measure-ment**	Two left-clicks	Calculates wavelength and velocity difference between two points.	`span_preview_deltav.log`
I	**Integrated Flux**	Left-drag	Integrates flux relative to a local linear continuum baseline.	`span_preview_intflux.log`
L	**Peak finding**	Nothing	Finds the most prominent emission lines in the spectrum.	

Key	Mode	Mouse Action	Description	Output / Log
P	**Snapshot**	Nothing	Takes a snapshot of the current preview and saves it as PNG format.	`SPAN_results/preview_snapshots`
C	**Cancel**	Nothing	Cancels all the overlays generated.	

All measurement results are automatically appended to CSV-style text files located in "SPAN_results" folder.

Visual redshift estimation in the Preview window

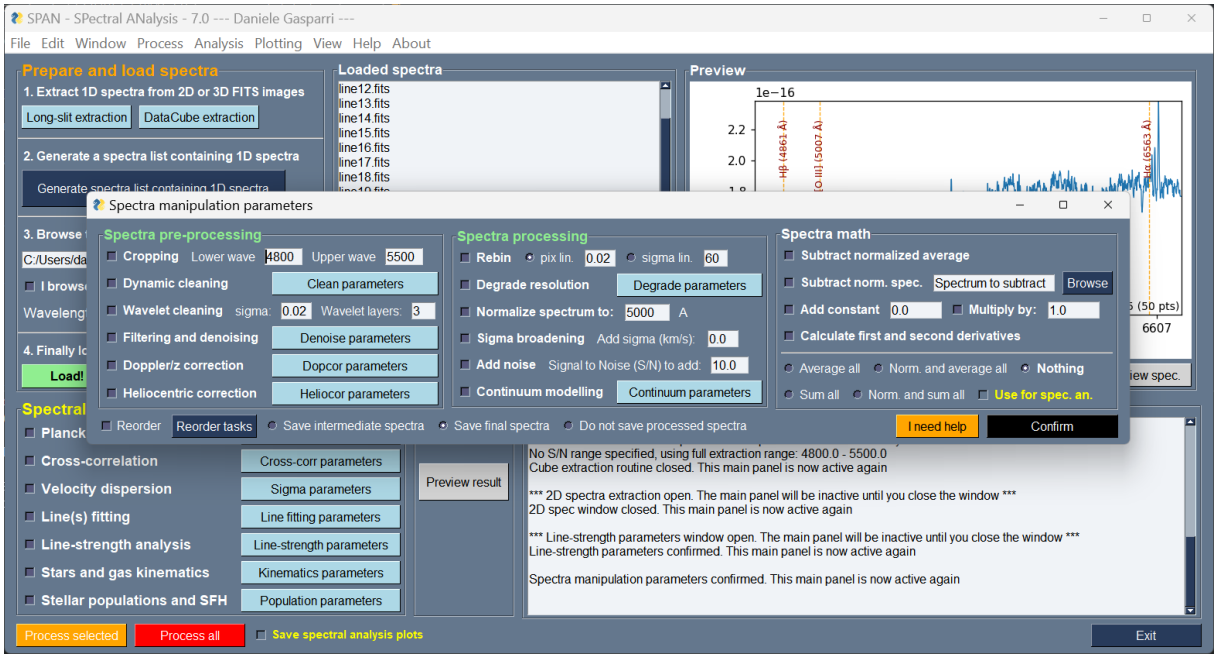
SPAN introduces a unique manual redshift estimation tool directly integrated into the preview window. By right-clicking and dragging the spectrum against fixed rest-frame line markers, you can quickly align prominent spectral features and obtain an approximate redshift value.

This functionality is not intended to replace quantitative fitting, but rather to provide a first-guess estimate in cases where the redshift is completely unknown. Such an estimate is essential for spectral analysis: for example, full spectral fitting with pPXF requires the input redshift to be known within approximately Delta(z) < 0.05 to ensure convergence and reliable results.

In the era of AI and automated black-box pipelines, there is still no substitute for visually inspecting spectra to understand where key features are located and to establish a sensible starting point. A "by eye" redshift estimation may still be the more accurate way to start with a solid base.

How it works
The basic principle is simple: resample the spectrum in a log spaced grid, therefore the redshift will be simply the linear (on the log space) shift of the whole spectrum from the starting point to the end point. The starting point is obvious, but what about the end point? We need some rest-frame reference lines.
Reference lines: Fixed vertical markers are shown at the rest-frame wavelengths of OII, HBeta, OIII, HAlpha and CaT lines. These lines remain at fixed wavelengths in the preview and serve as anchors.
Right-click drag: By pressing and holding the right mouse button, you can drag the entire spectrum horizontally, aligning its observed lines with the fixed rest-frame markers.
Cumulative shifting: Multiple drags accumulate. Therefore, if you are out of dragging space you can pause, pan and zoom and start shifting the spectrum again until you find a match with the anchors.
HUD feedback: While dragging, the HUD at the bottom-right of the preview shows the current estimated redshift, computed as the average shift relative to the reference lines.
Reset: A double right-click resets the spectrum to its original position and clears the estimated redshift. If you change spectrum, the redshift (and the shift) will automatically reset.

The Spectra manipulation panel



The spectra manipulation panel

Since version 6.3, all the tasks devoted to the manipulation of the spectra have been grouped in the "Spectra manipulation" panel. Here you will find some useful tasks that can be performed on spectra, grouped into the "Spectra pre-processing," "Spectra processing," and "Spectra math" frames. Any task executed within these frames modifies the selected spectrum and will have effects on the "Spectral analysis" frame. You can choose multiple tasks (e.g., rebinning, dopcor, adding noise…) without limitations. The "Preview spec." button allows you to observe the effect of the task(s) performed. The spectrum displayed and used in the "Spectral analysis" frame will be the one resulting from the selected tasks.

By default, the tasks are performed in series, following their order in the panel. No intermediate graphical information is available: if you activate three tasks, you will see the combined effect of all when you click the "Preview spec." button in the main panel. If you don't perform any task, don't worry: the original spectrum will be visible and ready to be used for spectral analysis. You can change the order of the tasks performed. Activate the tasks you want to apply, then click the button "Reorder tasks" and change their order as you wish, then confirm the selection. **WARNING:** If you activate or deactivate tasks after reordering, you must reorder again, or tasks will be executed in their default panel order.

The four math tasks in the "Spectra math" frame that involve all the spectra ("Average all," "Norm. and average all," "Sum all," "Norm. and sum all") act on all the original spectra loaded (and don't work if you have loaded just one spectrum), and remain insensitive to other tasks performed. By activating the "Use for spec. an." option, you force the program to use the result of these operations for the spectral analysis, disregarding any other task performed on individual spectra. Be cautious in managing this option. In any case, a message in the terminal window will appear, indicating that you are using the combined original spectra for spectral analysis.

The bottom panel

This panel is composed by two frames. The left one is the “Spectral analysis” frame, which contains the following tasks: 1) Blackbody fitting, 2) Cross-correlation, 3) Velocity dispersion, 4) Line-strength analysis, 5) Line(s) fitting, 6) Stars and gas kinematics, 7) Stellar populations and SFH. Each task is independent from the others and does not modify the spectra.

The “Preview result” button will display the task(s) result on the selected spectrum in a graphic Matplotlib window and in the output frame on the right. If no task is selected, a warning message will pop-up when clicking the button.

The right frame displays the text output of the software. This is how SPAN communicates with you. This panel reproduces the computer terminal and shows the output of the operations performed, including errors and warnings. In the macOS version of SPAN, this embedded terminal is not available for compatibility reasons. You should then rely on the standard terminal window of your computer.

Apply the tasks

Once you are satisfied with your work, you can process the spectra or the single selected spectrum. The “Process selected” button will perform all the tasks activated in the “Spectra manipulation” panel and in the “Spectral analysis” frame, saving the new processed spectrum to a fits file. By default, the program will save the final spectrum if more than one Spectra manipulation task is activated. If you want the intermediate spectra to be saved, activate the “Save intermediate spectra” option. This will save one spectrum for each task performed in a cumulative way, following the order of the tasks performed. For example, if you have selected rebinning, sigma broadening, and add noise, following this order, the program will save a spectrum with rebinning done, a second spectrum with rebinning + sigma broadening applied, and a third with rebinning + sigma broadening + add noise applied. **WARNING:** If you changed the order of the tasks with respect to the default one you will find difficult, with this option enabled, to understand which is the final spectrum of the chain, so use this option with care. If you are planning to use the tasks of the “Spectra manipulation” panel just as preparatory phases for the spectral analysis, maybe you do not want to save the processed spectra every time you will press the “Process selected” or “Process all” buttons. In this case, in the “Spectra manipulation” panel you can select the option “Do not save processed spectra”. **IMPORTANT:** In the “Process selected” mode, the results of the spectral analysis frame will be written only in the output frame.

By clicking “Process all”, you will apply all the tasks to all the spectra in your list. This is the only way to save the results of the “Spectral analysis” frame in an ACII file. You can also store the plots generated during the spectral analysis by activating the option “Save spectral analysis plots” at the very bottom of the SPAN panel. The plots will be saved in high resolution PNG format and stored in the “plots” subdirectory of the “SPAN_results” folder.

Zooming

Since version 7, SPAN includes a cross-platform zooming system designed to improve the readability of the GUI and the embedded Matplotlib preview. The zoom affects the entire application, including GUI elements (buttons, text labels, checkboxes, frames, etc.), subwindows opened during the workflow, Matplotlib figures opened in external windows.

The zoom is controlled through the View → Zoom In / Zoom Out / Reset Zoom menu entries. Zoom In increases the size of all fonts and widgets. Zoom Out decreases them. Reset Zoom restores everything to the original size as defined by the layout.

Because Tkinter behaves differently across platforms, the implementation includes a few platform-specific adjustments. Windows (tested on Windows 10/11): Zooming in and out work good, but the reset zoom sometimes may not properly reset the zoom to the default level. Linux (tested on Ubuntu 22.04, X11 and Wayland): By default, Tk tends to stretch canvases after a resize. SPAN explicitly disables this behavior at reset, so the embedded preview canvas returns to its original dimensions. macOS: The zooming behavior is expected to follow Linux but extensive testing has not been performed.

The input files

In order to work properly, SPAN sometimes needs input text files containing information about your data. To see how they must be formatted, please take a look at those coming with SPAN and already set by default in the graphic interface.

IMPORTANT: The text files MUST always have the first line as header, identified by # (e.g. #spectrum)

1. **Spectra list file:** It is essential. If you don't believe it, try to perform any task without upload the spectra and you will see the effects! It is just an ASCII file containing the path (relative if they are in a subfolder of SPAN, absolute if they are elsewhere) and the complete names (with file extension) of the spectra you want to process. You can use any spectra you want, with different format (fits, ASCII…) and resolutions, but it is mandatory to use spectra with the same wavelength units. If you just want to play with one spectrum, then load the ASCII or fits 1D spectrum and activate the option "I browsed a single spectrum" before clicking the button "Load!".

```
example_list.dat

#filename ---> header: always necessary!
[path/]spectrum1.fits
[path/]spectrum2.fits
[path/]spectrum3.fits
[path/]spectrum4.fits
```

Other ASCII files may be needed in the "Spectra manipulation" panel for some specific tasks. They are:

2. **Doppler correction file** for the "Doppler/z correction" task and the "I have a list file" option selected: It is an ASCII file containing two columns: 1) Name of the spectrum and 2) Radial velocity or z to correct to the spectrum. This file has the same format of the output text file generated by the Cross-correlation task, so you can directly use it.

```
    example_dopcor.dat

#spectrum       RV(km/s) or z ---> header: always necessary!
[path/]spectrum1.fits  1000
[path/]spectrum2.fits  1001
[path/]spectrum3.fits  1002
[path/]spectrum4.fits  1003
```

3. **Heliocentric correction file** for the "Heliocentric correction" task and the "I have a file with location…" option selected: It is an ASCII file containing three columns, separated by a space, following the SAME order of your spectra in the loaded spectra list file: 1) Name of the location, 2) Date of the observation (just year, month, day, not the hour), 3) RA of the object (format: degree.decimal), 4) Dec. of the object (format: degree.decimal).

```
                example_heliocorr.dat

#where  date        RA          Dec
paranal 2016-6-4    4.88375     35.0436389
```

```
paranal 2016-6-30   10.555      1.11121
aao     2011-12-24  -50.034     55.3232
aao     2018-2-13   -11.443     11.2323
SRT     2020-7-31   70.234      55.32432
```

Some external files may be needed for specific options of the "Spectral analysis" tasks. They are:

4. **Cross-correlation and velocity dispersion tasks:** These task require a single template, in fits or ASCII format (i.e.. just a spectrum!).

5. **Line-strength analysis task** and the option "User indices on a list file" selected: It is an ASCII text file containing the index definitions. One index per column. Don't mess it up with the index file, otherwise you will obtain inconsistent results! Luckily, you can always test a single index and see the graphical preview before running the wrong indices on 240913352 spectra and waste one year of your life.

```
example_idx_list_file.dat

#Idx1    Idx2  ---> header: always necessary!
8474   8474 ---> row2: left blue continuum band, in A
8484   8484 ---> row3: right blue continuum band, in A
8563   8563 ---> row4: left red continuum band, in A
8577   8577 ---> row5: right red continuum band, in A
8461   8484 ---> row6: left line limits, in A
8474   8513 ---> row7: right line limits, in A
```

6. **Calculate velocity dispersion coefficients**, located in the "Line-strength parameters" subwindow: It determines 4 spline correction coefficients in order to correct the equivalent width of galaxy spectra broadened by the velocity dispersion. It needs a sample of unbroadened spectra that are a good match of the expected stellar populations of the galaxy spectra you want to correct to the zero velocity dispersion frame. The input file is just an ASCII file containing the list of the spectra used as sample, i.e., a normal spectra list!

```
example_coeff_determ.dat

#filename ---> header: always necessary!
[path/]stellar_spectrum1.fits
[path/]stellar_spectrum2.fits
[path/]stellar_spectrum3.fits
[path/]stellar_spectrum4.fits
```

7. **Correct the line-strength for velocity dispersion task:** To apply the velocity dispersion coefficients and correct the raw equivalent widths to the zero velocity dispersion frame, you need this task and three files:

   - Sigma list file: a file containing the name of the spectra, in the SAME ORDER of the spectra list loaded and processed with SPAN, the velocity dispersion and the relative uncertainties. It has the same format of the output file generated by the Velocity dispersion task, so you might use just that (but DO NOT change the order of your spectra in the listbox after you generated this file!).

example_sigma_vel.dat

```
#Spectrum       Sigma(km/s) err ---> Header: always necessary
spectrum_name1  166.2       3.0
spectrum_name2  241.5       3.1
spectrum_name3  335.1       6.2
spectrum_name4  241.5       3.2
```

- EW file list to correct: the text file containing the raw equivalent widths you want to correct. It has the same format of the output file generated by the Line-strength measurement task. BE CAREFUL to check that the indices are in the EXACT same order of those you used in the "Calculate velocity dispersion coefficients" task for the correction coefficient determination. Also BE CAREFUL to maintain the same order of your spectra in the listbox.

example_uncorrected_ew.dat

```
#Spectrum       idx1    idx2    idx3   idx1err idx2err idx3err
spectrum_name1  0.27    1.38    3.56    0.01     0.01    0.02
spectrum_name2  0.15    1.32    3.43    0.01     0.02    0.02
spectrum_name3  0.08    0.75    2.81    0.01     0.02    0.02
spectrum_name4  0.14    1.25    3.18    0.01     0.01    0.01
```

- Correction coefficients file: it is the output file generated by the "Calculate velocity dispersion coefficients task".

example_correction_coeff.dat

```
#Pa1          Ca1           Ca2          Pa1e         Ca1e         Ca2e
4.3282e-08   1.06712e-08  -2.7344e-09  -5.7463e-09   2.2911e-09   2.8072e-10
-2.9602e-05  -1.2012e-05   -3.5782e-07   3.9353e-06  -1.9246e-06  -2.9293e-0
0.0017       0.0021        8.5793e-05  -0.0001       0.0004       9.9212e-05
-0.0029      -0.0085       -0.0016       0.0053      -0.0003      -0.0002
```

File organization

SPAN generates different types of **results**, which are all stored in the "SPAN_results" folder:

- **Extracted spectra** from the "long-slit extraction" and the "Datacube extraction" sub-programs. The spectra extracted from long-slit data are stored in the "longslit_extracted" folder. The spectra extracted from datacube data are stored in "RUN_NAME" folder, where "RUN_NAME" is the arbitrary name you have to set in the "Datacube extraction" sub-program.
- **Processed spectra** in FITS format, both in the "Process selected" and "Process all" mode. These are processed spectra from the "Spectra Manipulation" panel or auxiliary spectra generated from Spectral analysis tasks (e.g. best fit and residuals from the "Stars and gas kinematics" and "Stellar populations and SFH" tasks). These spectra are stored in the "processed_spectra" folder.
- **ASCII files** in plain text .dat format, containing the results of the Spectral analysis tasks, which are generated only in the "Process all" mode. These products are saved in specific folders with the same name of the spectral analysis task applied.
- **Plots** in high resolution (300 dpi) PNG images. They are generated only for the Spectral analysis tasks and are the plots displayed also in the "Preview result" mode. These plots are saved only in "Process all" mode and if the option "Save spectral analysis plots" is activated. If you just need one specific plot for one spectrum in the list, you can save it directly from the Matplotlib window that opens in the "Preview result" mode. These plots are stored in the "plots" folder.

The **spectra list** files generated by the "Generate spectra list containing 1D spectra" are saved in the "spectra_lists" folder within the "SPAN_results" main folder.

List of operations you can perform

SPAN can perform many operations on the spectra.

WARNING: All the wavelengths of SPAN are given in A, in air, and all the velocities are in km/s.

Here is a description of any task:

1. **Utilities window (Window –> Utilities):**
    - Show the header of the selected spectrum = shows the header of the selected spectrum, only for FITS files;
    - Show the wavelength step of the spectrum = shows the step of the selected spectrum;
    - Estimate the resolution = calculates the resolution of the selected spectrum by trying to fit an emission sky line. In The W1 and W2 you should put a small wavelength interval containing a sky line: it's up to you!
    - Convert the spectrum to = converts the selected spectrum to ASCII or Fits;
    - Compare spectrum with = compares the selected spectrum with another one selected by you. This comparison spectrum should have the same wavelength units;
    - Convert Flux = converts the flux from frequency to lambda and vice-versa. The buttons "see plot", "save one" and "save all" are active to see and save the results for one or all the spectra;
    - S/N = measures the Signal to Noise in the selected spectrum, in the central wavelength selected (W.). The buttons "save one" and "save all" are active to save one or all the S/N computed for the spectra.
2. **Spectra manipulation panel:**
    - Spectra pre-processing frame
        - Cropping = performs a simple cropping of the spectra. If the wavelength window to crop is outside the spectrum, SPAN will ignore the task and will not perform the crop;
        - Dynamic cleaning = performs a sigma clipping on the spectra. The sigma clip factor, the resolving power of the spectrum and the velocity dispersion (instrumental and/or intrinsic) of the selected spectrum is required in order to perform a better cleaning. For the "Process all" mode, the option "R and sigma vel file" is available in order to have R (resolution) and sigma values for all the spectra to be processed. Be VERY careful to use this task with strong emission line spectra;
        - Wavelet cleaning = performs a wavelet denoise of the spectra. The mean standard deviation of the spectra continuum (sigma) and the number of wavelet layers to consider are required. You don't need to measure it, just try different values. Be careful to not delete the signal;
        - Filtering and denoising = smooths the spectra by performing some denoising filters: box window moving average, gaussian kernel moving average, low-pass Butterworth filter and band-pass Butterworth filter;
        - Dopcor/z correction = performs the doppler or z correction of the spectra. Single shot option with user input value of radial velocity (in km/s) or z is available both for one or all the spectra. "I have a file" option only works with the "Process all" mode: you need a text file with the spectra name and the recession

velocities or z values. This file can be generated by the “Cross-correlation” task in “Process all” mode;
  - Heliocentric correction = performs the heliocentric correction on the spectra. The “Single” option requires a location which can be selected from the “loc.list” button (it requires an internet connection the first time!). The other fields are the date in the format YYYY-MM-DD and the RA and Dec. of the observed object (in decimals). In the “I have a file” option, available only for the “Process all” mode, a list file with location, date, RA and Dec. coordinates for each object is required.
- Spectra processing frame
  - Rebin = performs a rebin/resample of the spectra in linear wavelength step (“pix.lin” option, with the step in A) and in sigma linear step (“sigma lin.” option, with the sigma step in km/s);
  - Degrade resolution = degrades the resolution of the spectra from R to R, from R to FWHM and from FWHM to FWHM;
  - Normalize spectrum to = normalizes the spectra to the wavelength provided (in A);
  - Sigma broadening = broads the spectra by convolving with a gaussian function with the standard deviation provided by you, in km/s. Remember that the real broadening of the spectra will be the quadratic sum between the broadening and the instrumental sigma;
  - Add noise = adds a random Poisson noise to the spectra with a S/N defined by you. Remember that the final S/N of the spectra will the sum in quadrature between the added noise and the intrinsic S/N of the spectra;
  - Continuum modelling = models the continuum shape with two options: 1) Simple filtering of the continuum by reducing the spectrum to a very small resolution (R = 50), and 2) polynomial fitting, with the possibility to mask emission/contaminated regions. Both the continuum models can be divided or subtracted to the original spectrum;
- Spectra math
  - Subtract normalized average = subtracts to the spectra the normalized average made from all the spectra loaded;
  - Subtract norm. spec. = subtracts to the spectra a normalized spectrum selected by you, which shares the same wavelength units;
  - Add constant = adds a constant to the spectra;
  - Multiply by a constant = multiplies the spectra by constant value.
  - Calculate first and second derivatives = automatic calculation of the derivatives of the spectra. This task does not modify the original spectra and the derivative spectra cannot be directly used for spectral analysis.
  - Average all = averages all the spectra (only available in “Process selected” mode);
  - Norm. and average all = normalizes to a common wavelength and average all the spectra (only available in “Process selected” mode);
  - Sum all = sums all the spectra (only available in “Process selected”);
  - Norm. and sum all = Normalizes and sum all the spectra (only available in “Process selected”). The option “Use for spec. an.” forces the program to use the result of one of these 4 operations for the following spectral analysis.

3. **Spectral analysis:**

- Blackbody fitting = performs a fit of the spectrum with Planck's blackbody equation and gives the temperature estimation. It works with any type of spectra but it performs better for stellar spectra, with wide (at least 5000 A) wavelength range;
  b) Cross-correlation = performs a cross-correlation of the spectra with any template. You can smooth the template to a velocity dispersion value in order to improve the cross-correlation and should identify a narrow region of the spectrum to be cross-correlated (tip: the Calcium triplet lines are the best features in the NIR);
- Velocity dispersion = performs the measurement of the velocity dispersion of the spectra with a rough (but fats) fit with any template you will select (just one!). Some pre-loaded bands in the visible and NIR are shown but you can input any band. The routine succeeds with strong features (the CaT is the best). It is a little rough but very fast and gives reasonably accurate results;
- Line-strength analysis = performs the equivalent width measurement of the spectra, with a single index, with a list of indices or with the Lick/IDS system. The results are provided in Angstrom. MonteCarlo simulations are run for the uncertainties estimation. The calculation of the Lick/IDS indices can be personalized in many ways: you can correct for the emission, for the velocity dispersion and the recession velocity. You can also perform a linear interpolation with the SSP models of Thomas et al. 2010, Xshooter, MILES and sMILES to retrieve the age, metallicity and alpha-enhancement (not available for the Xshooter models) of the stellar populations via linear interpolation or with machine-learning pre-trained models (Gaussian Process Regression). From the "Line-strength parameters" window, it is possible also to perform the "Calculate velocity dispersion coefficients" task. This task broadens a sample of SSP spectra up to 400 km/s and calculates the deviation of the equivalent width of the indices contained in the index file provided. It works only by pressing the "Compute!" button and creates a text file with a third order polynomial curve that fits the behavior of the broadened index (or indices). The "Correct the line-strength for velocity dispersion" task performs the correction of the equivalent widths based on the coefficients estimated with the "Calculate velocity dispersion coefficients" task. It works only by pressing the "Correct!" button and requires the raw equivalent width measurements stored in the ASCII file generated previously, with the same indices in the same order to that considered in the "Calculate velocity dispersion coefficients". The output files of the "Line-strength analysis", "Calculate velocity dispersion coefficients" and "Velocity dispersion" are ready to be used for this task, if we are considering the same spectra and indices;
- Line(s) fitting = performs the fitting of a line in the inserted wavelength range using a combination of a Gaussian function to model the spectral line and straight line for the continuum. If "CaT lines" is selected, the task will perform an automatic fitting of the Calcium Triplet lines, assuming they have been previously corrected for redshift and/or Doppler velocity;
- Stars and gas kinematics = uses the pPXF algorithm of Cappellari et al. 2023 to fit a wavelength region of the spectra with a combination of templates. You can select the template library you prefer among the pre-loaded EMILES, GALAXEV, FSPS and XSHOOTER, or you can use your custom set of templates. You can decide how many moments to fit, whether fit only the stellar component or also the gas, whether estimate or not the uncertainties with MonteCarlo simulations and much more. It returns the radial velocity, the velocity dispersion and the higher moments up to H6

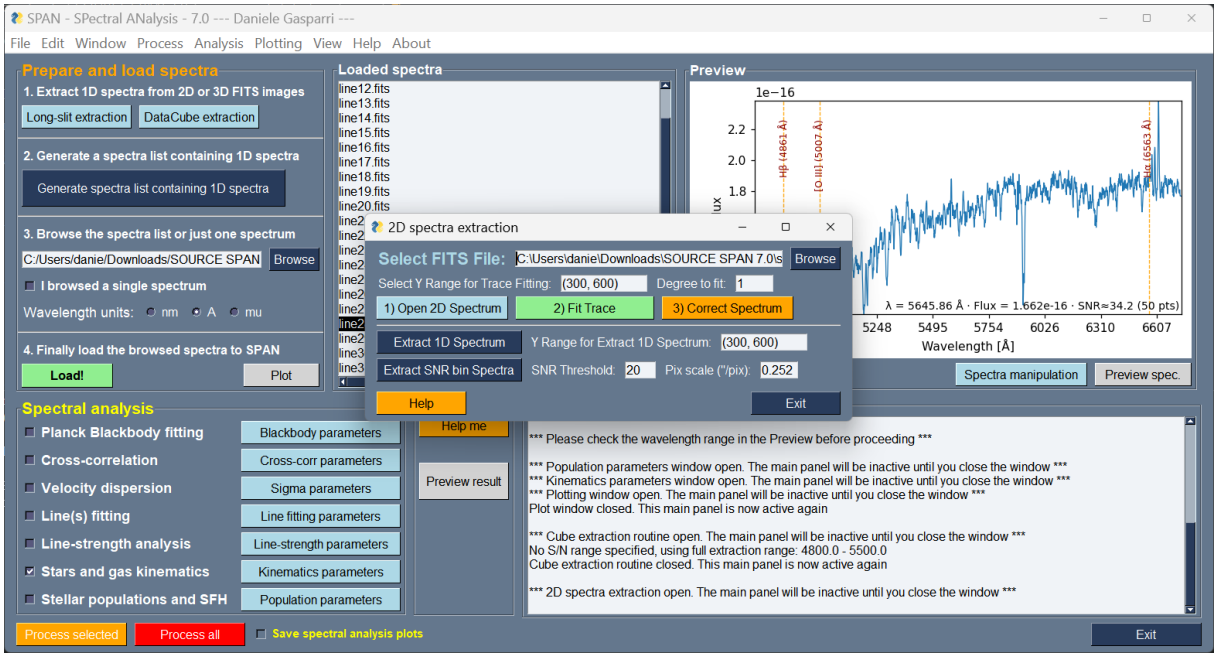
(if needed, and a nice plot courtesy of Cappellari), as well as the spectra product (best fit, residuals, emission corrected spectra, if any, gas spectra, if any, best fit gas spectra, if any);

- Stellar populations and SFH = uses pPXF to fit a wavelength region of the spectra with a combination of templates. You can select the template library you prefer among the EMILES, GALAXEV, FSPS, XSHOOTER and sMILES, add any EMILES custom library, or any .npz file following the pPXF standard. You can decide whether include the gas emission or not, the reddening, the order of multiplicative and additive polynomials of the fit, the age and metallicity range of the templates, and much more. It returns a beautiful plot, the kinematics, the weighted age (in luminosity and mass), metallicity (in luminosity and mass), the M/L, the SFH and saves the best fit template and the emission corrected spectra (if any). Works great in the visible and in the NIR, but this depends on the quality of your spectra.

The sub-programs

The two light-blue buttons in the upper left corner of SPAN (in the “Prepare and load spectra” frame) are sub-programs that might help you to generate the 1D spectra needed. Here is how they works:

Long-slit extraction

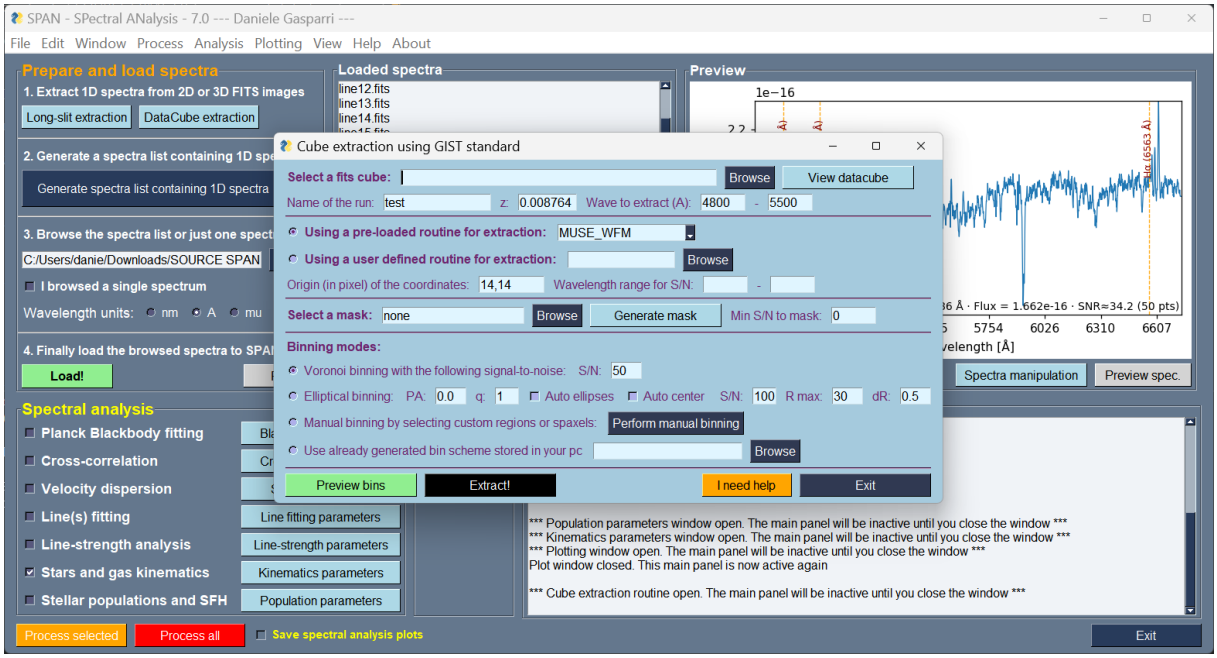


Long-slit extraction panel.

Allows the extraction of a single 1D spectrum or a series of 1D spectra from a reduced and wavelength calibrated 2D fits image containing the long-slit spectrum of a source, with dispersion axis along the X-axis and the spatial axis along the Y-axis. Before proceed to the extraction, you need to load a valid 2D fits image, then you need to:

- Open the spectrum and see if everything is ok;
- Fit the photometric trace in order to find the maximum along the dispersion axis. You need to set the degree of polynomial curve that will be used to fit the trace and correct the distortion and slope of the spectrum;
- Correct the spectrum for distortion and slope using the model trace obtained in the previous step. Then, you can:
- Extract and save only one 1D spectrum within the selected Y range (useful for point sources);
- Extract and save a series of n 1D spectra covering all the spatial axis and obtained by binning contiguous rows in order to reach the desired S/N. A spectra list file ready to be loaded to SPAN is also generated, as well as a text file containing the position of the bins relative to the central region of the galaxy and the S/N. The S/N threshold that you must insert is just a very rough estimation of the real S/N. A good starting value to produce 1D spectra with bins with realistic S/N > 30 is 20. Adjust the SNR Threshold to your preference by looking at the real S/N of the bins. The pixel scale parameter is optional. If you set to zero it will not be considered. This option is useful if you have the spectrum of an extended source (e.g. a galaxy) and want to sample different regions.

DataCube extraction



Datacube extraction panel.

This sub-program allows you to extract a series of n 1D spectra from a 3D FITS image (i.e., a datacube) following the GIST pipeline standard (Bittner et al., 2019, 2021). The required parameters are similar to those used in the GIST pipeline (and for any datacube extraction in general). The data product generated (i.e. spectra and bin table) are fully compatible with both with SPAN and GIST.

Overview

The essential files and parameters required for this module to function correctly are highlighted in bold. The first step is to load a fully reduced, valid FITS datacube. Given the lack of a standardized convention for FITS keywords and data storage extensions, it is strongly recommended to use the “View datacube” button to verify that the datacube is correctly read.

This module has been tested with MUSE, CALIFA, JWST NIRSpec IFU, and the new WEAVE LIFU data products. Other formats may only be partially supported. If at least the flux values are properly read, the “View datacube” button should display the image, though the wavelength slider in the Matplotlib window may show only a generic “Wavelength index” instead of actual wavelength values.

Basic parameters for extraction

Once the datacube is correctly loaded, configure the following basic parameters:

- A run name (arbitrary).
- The source redshift (use zero if no de-redshifting is required).
- The wavelength range for extraction. **WARNING**: if de-redshift is needed (i.e. inserted redshift value different than 0), the wavelength range to extract is referred to the equivalent rest-frame range. Example: we have observations in the near-infrared from 9200 to 12500 A from a galaxy located at z = 1.17. If redshift value is set to zero (no de-redshift)

we can insert this wavelength range (or a subrange) for extraction, but if we want to perform the de-redshift, we must insert the equivalent rest-frame wavelength range for extraction. In this case, it will be roughly 4200-5800 A.

- Selecting the extraction routine. In the middle panel, choose the routine for reading and extracting the data. Pre-loaded routines are available for MUSE, CALIFA, WEAVE LIFU, and JWST NIRSpec IFU datacubes. You can write your own extraction routines as .py files following the structure of the available ones and load using the option "using a user defined routine for extraction".
- Configure the zero point for spatial coordinates. This is typically the center of the datacube spatial axes but can also be set to the spaxel coordinates of the galaxy's center. The "View datacube" option is useful for retrieving this information.
- Wavelength range to consider for S/N estimation. If left empty, the S/N for the Voronoi rebinning will be estimated using the provided extraction wavelength range. Otherwise, enter the wavelength range where estimate the S/N. This field has effect only for Voronoi binning mode.

Applying a spatial mask

Locate the required "Select a FITS mask" field to browse and load a valid mask FITS file. This mask must have the same spatial dimensions as the datacube. If mask is not needed, simply leave this field empty. If no mask is available but you need it, generate one by clicking "Generate mask". A Matplotlib window will open, displaying the datacube with a slider to navigate across wavelength indices (note: these are indices, not actual wavelengths!). To mask specific spaxels (e.g., to exclude contaminated regions and/or bad spaxels), Ctrl+left-click to mask and Ctrl+right-click to unmask. You can also mask/unmask larger areas by clicking and dragging with the Ctrl+left or right mouse button. On touchscreen devices (i.e. Android), masking is performed by tapping and dragging. Unmasking is not yet available. Ensure you remain within the plot boundaries while dragging, or the selection may not be applied. Once satisfied, close the Matplotlib window. The mask will be automatically saved and loaded in the relative field. If you open the "Generate mask" window but no masking is needed, simply close the window. An empty mask file will be created but it will not have effect on your data. Optionally, you can also mask spaxels with low S/N values by specifying a threshold. **Important**: bad and/or invalid spaxels are automatically masked out by SPAN.

Binning

Binning spaxels together to extract spectra with higher signal is a very common practice when dealing with datacubes. SPAN offers three options for binning:

1. **Voronoi binning:** The standard for datacube data. This option bins contiguous spaxels using the Voronoi method (Cappellari et al., 2003) to achieve the desired S/N.
2. **Elliptical radial binning:** Creates concentric annuli around the selected center with a minimum thickness of dR (in arcsec) and a minimum S/N threshold. The center, orientation and eccentricity of the ellipses can be set manually. In this case, the center is taken as the zero point for spatial coordinates given above ("Origin (in pixel) of the coordinates"), while the position angle (PA, assuming standard astronomical orientation of the cube, i.e. north up, east left) and the eccentricity (q = b/a) are inserted here. If automatic placing is selected ("Auto center"), SPAN will find the photometric center for you from the 2D collapsed image automatically generated and will center the ellipses around this point.

If automatic geometry is selected ("Auto ellipses") SPAN will fit the ellipses following the photometric profiles of the 2D collapsed image and will find the best values of PA and q for you. If you have reliable data at hand, please use the manual method, at least for the estimation of the photometric center (the option "View datacube" may be very useful). The coordinates of the generated bins written in the _table.fits (XBIN, YBIN) will be the coordinates of the elliptical bins on the major axis corresponding to the elliptical radius R_flux. If the center of your coordinate system corresponds to the photometric center of the ellipses, these coordinates just give you the distance between the center and the corresponding bin point (signal weighted mean) on the major axis. If the origin of your coordinate system is different than the center of the ellipses, then you must account for the offset to treat XBIN and YBIN as distances. **TIP:** Enable the "Plot radial profile" option in the plot maps sub-program if you want to plot directly the radial profiles of your elliptical binned and analyzed spectra.

3. **Manual binning:** Defines manual bins with mouse multiple selection. The manual bin selection lets you to have the full control on the spaxels to bin. By clicking to the "Perform manual binning" button, an iterative Matplotlib will appear showing your DataCube. You can select any region to be binned by Ctrl+left-click or dragging on the image. Deselecting can be performed by Ctrl+right-click or dragging. On touchscreen devices (i.e. Android), manual binning regions are selected by tapping and dragging. Deselecting is not yet available. Any region fully separated by the others by at least one spaxel, is considered by SPAN as a single bin. You can draw as many regions as you want. Once done, close the matplotlib window and the binning info will be saved.
4. **Use pre-existing bin info:** This is useful, for example, if you have a datacube split between blue and red arm (e.g. WEAVE LIFU) and you want to sample exactly the same regions, either with the Voronoi or manual binning. You just need to select the folder where the bin info and masking files are stored from the previous run that you want to take as reference for the binning info. SPAN will find the required files in the folder, copy them to the folder where you are saving the new extracted spectra and rename them according to the folder name. Therefore, the extraction routine will use these bin and mask info instead of generating new ones.

You can always preview the results by clicking on the "Preview bins" button. If Voronoi or Elliptical binning are activated, you will see the bins color coded by their S/N. If Manual binning is activated, you will see the regions you selected that will be binned. In this case you will see still the single spaxels inside these regions, color coded by their S/N.

Starting the Extraction

Once all parameters are set, click "Extract!" and grab a coffee! The extraction process may take several minutes for large datacubes (i.e. MUSE).

Upon completion, both GIST-standard and SPAN-standard spectra will be saved.

IMPORTANT: Once the extraction has been performed, to prevent accidental overwriting of the files any operation of the panel is locked until you will change the "Name of the run", including the preview. If you want to change any parameters and see the result, you MUST first change the "Name of the run". If you try to perform another extraction without changing the "Name of the run", a terminal message will warn that the results in this folder are already available and no extraction is performed.

Other useful sub-programs

In the menu bar you can find more sub-programs that might help you in the difficult work of analyzing and processing astronomical spectra. They work independently from the main program, so you can also not load any spectra if you don't need to perform tasks on them. Here is how they works:

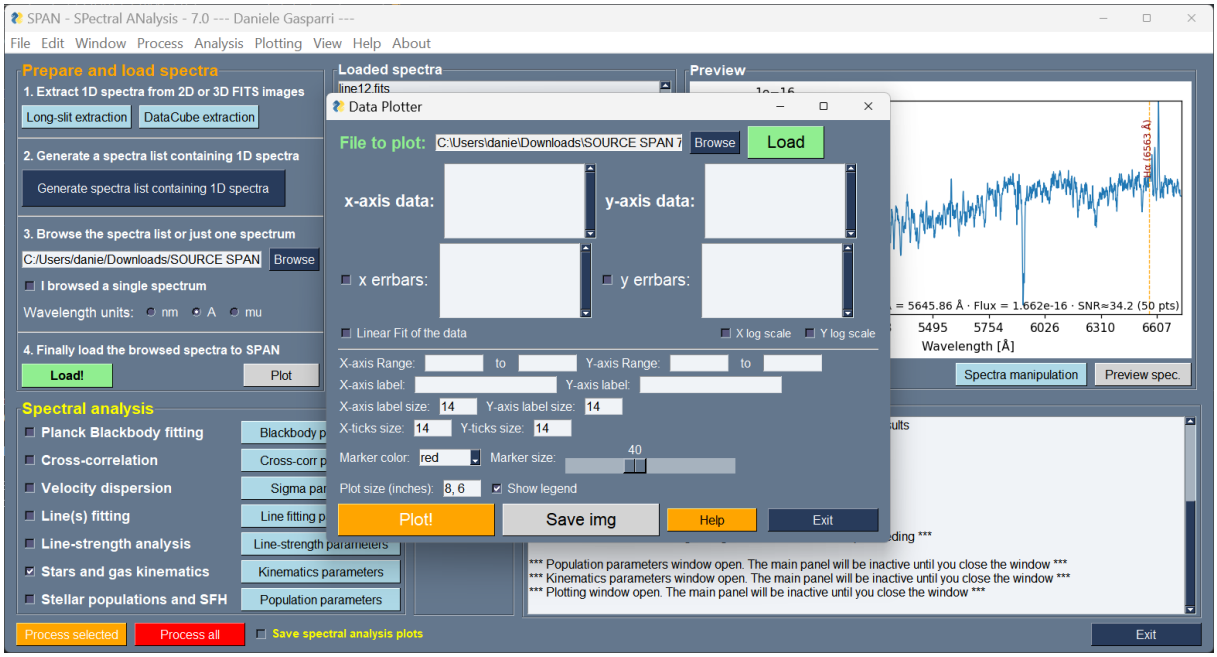
Text editor

A simple ASCII file editor where you can create, read or modify ASCII files, included those generated by the SPAN tasks. Some basics operations are available, such find, replace and merge rows.

FITS header editor

An header editor to add, remove and save the keywords of fits header files. You can select between: "Single fits header editor" to work with the keywords of one fits file, "List of fits header editor" to modify the keywords of a list of fits files, "Extract keyword from list" to extract and save in an ASCII file one or more keywords from the headers of a list of fits files.

Plot data

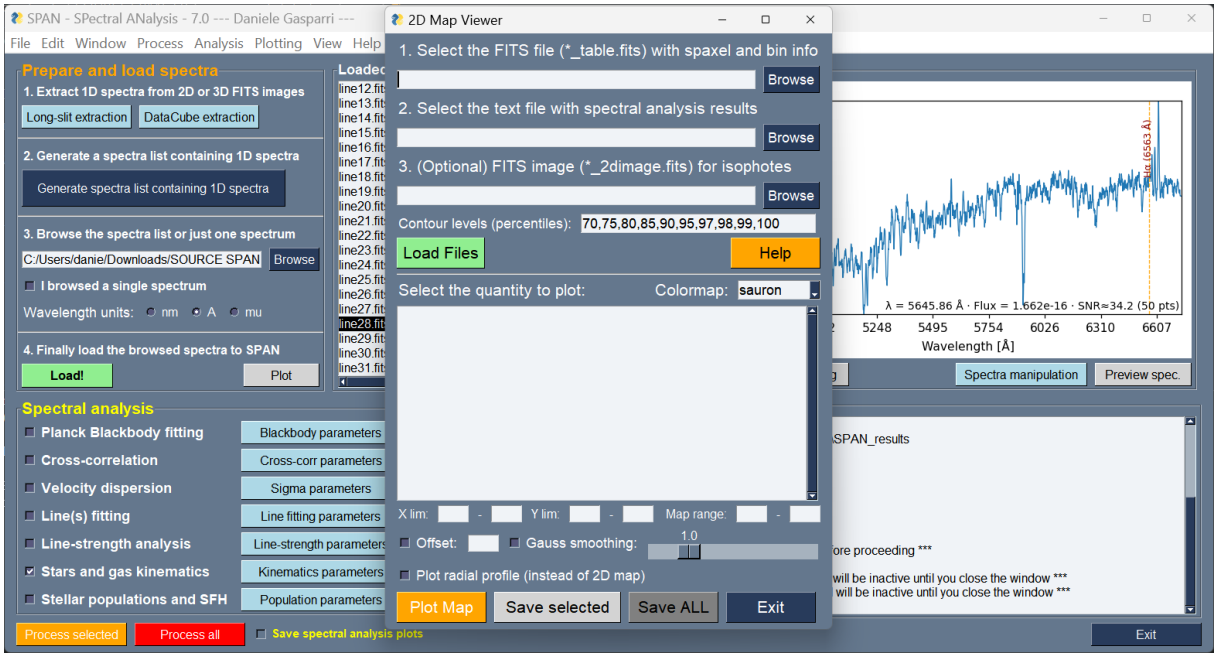


Plotting data panel.

A sub-program to plot the data generated by the "Spectral analysis" frame and, in general, all the data stored in ASCII space-separated data. Once you browse for the text file and click the "Load" button, the program will automatically recognize the column names. Select a name for the x and y axis and plot the data to see them in an IDL style plot. You can personalize the plot by adding the error bars, set the log scale, add a linear fit (simple fit without considering the uncertainties), set the labels, the range, the font size, size and colors of the markers and decide

if visualize the legend or not. You may also save the plot in high resolution PNG image format, in the directory where you run SPAN. If any error occur, the program will warn you.

Plot maps



Plot maps panel.

A sub-program to plot 2D maps from extracted quantities from datacubes. If you extracted datacube spectra and performed some spectral analysis with SPAN, you can load the RUN-NAME_table.fits file stored in the extracted spectra folder within the "SPAN_results" folder, and any of the ASCII files generated by the spectral analysis tasks in the "Process all" mode. SPAN will show you the quantities available from this file and generate beautiful and customizable 2D maps.

Usage

You need two mandatory files in order to plot the maps:

1. The RUN_NAME_table.fits generated during the extraction process of the "DataCube extraction" routine, which contains the information of the Voronoi (or manual) spectral bins generated. This file is stored in the "SPAN_results/RUN_NAME" folder, once the datacube extraction is completed.
2. The RUN_NAME_*.dat file(s) generated by any spectral analysis task on the same sample of 1D extracted spectra. This ASCII file is available once a spectral analysis task is performed with the "Process all" mode.

Optional file

If you want to overplot to the maps also the isophotal contour of your datacube image, you should browse and insert the FITS file containing the image, which has been automatically generated by the "Datacube extraction" sub-program and stored in the "SPAN_results/RUN_NAME" folder. This file has the suffix: "RUN_NAME_2dimage.fits" If this image is provided, you can play with the contour levels to adjust the isophotal contours to your needs. The contour levels are expressed in percentiles, from 0 to 100, and comma separated.

If an invalid value is inserted, the default pre-loaded contours will be used and a message will warn you in the output/terminal window.

Plotting and saving the maps
Once the required files are browsed, just click “Load files” and in the listbox below will appear the quantities available to plot, taken from the ASCII .dat file loaded. You can change the color code of the maps in the “Colormap” menu and then you can plot any of these quantities and display an interactive map on a Matplotlib window by clicking “Plot Map”. In this window, you can click on the map to show the identifier of the selected spectral bin and the relative information (e.g. velocity). You can also change the plot and colormap limits.

Smoothing
You can also perform a smoothing of the Voronoi bins (if Voronoi has been performed) and their values to create more pleasant plots. This is achieved by activating the “Gaussian smoothing” option. Adjust the smooth value to your needs and see the results by clicking “Plot map”.

Radial profiles
You can alternatively plot the radial profiles of the selected quantity by activating the “Plot radial profile (instead of 2D map)” option. If selected, the isophotes, colormap scheme, limits, and smoothing (if activated) will be ignored and the radial profiles will be plotted instead of the 2D maps. This radial profiles option is particularly suited for spectra extracted with the Elliptical binning mode.

Saving the plots
Once you are satisfied, you can save the current map in high resolution PNG image by clicking the “Save selected” button, or you can automatically save the maps for all the quantities available by clicking the “Save all” button.

Utilities

A standalone frame that allows you to find out information about the selected spectrum, such as the header, the sampling (in A), the S/N, or simply convert the spectrum to ASCII or binary fits.

Tricks in the menu bar

The menu bar was introduced in version 4.5 of SPAN, offering several helpful options to enhance your experience with spectral analysis. Here is a detailed overview of some options that you won't find in the main panel (unless you are using the Android version):

1. File –> Save Parameters…: Allows to save all parameters and values from the main panel and the various parameter windows of the tasks in a .json file. This feature is very useful as it enables you to preserve any modifications made to parameters, facilitating the restoration of your session each time you reopen SPAN;
2. File –> Load Parameters…: Allows to load the parameters saved in the .json file. **WARNING**: the loaded parameters will also load the spectrum or the spectra list you were working with, but you need to manually click the "Load!" button to effectively load to SPAN your saved and restored spectra;
3. File –> Restore Default Parameters: Resets all the parameters to their default values. Useful if numerous parameter modifications during a lengthy session have resulted in issues, allowing you to start from fresh;
4. Edit –> Clear All Tasks: Immediately deactivates all tasks activated during the session, enabling a clean restart;
5. Edit –> Clean Output: Deletes the content of the output window. Particularly useful during extended sessions where the generated output may become quite large.
6. Edit –> Show result folder: shows the location of the "SPAN_results" folder in case you forgot;
7. Edit –> Change result folder…: create a new "SPAN_results" folder wherever you want.

Basic spectral analysis with SPAN

SPAN allows you to perform some basic analysis tasks thanks to built-in routines, both for stellar and galaxy spectra. These tasks are useful when speed is essential, e.g. when dealing with thousands of spectra or to a first-order view of some physical quantities. They work with rest-frame corrected spectra. If you do not have rest-frame spectra, you can correct them with the "Doppler/z correction" task of the Spectra manipulation panel and pass directly to the analysis tasks. Here is a brief summary of how they work.

Planck blackbody fitting

This task fits the spectral continuum with a Planck blackbody function and returns the best estimation of the effective temperature. The fit is accurate for stellar spectra spanning a wide wavelength range (about 3000 A) and works at best if the peak of the Plank function is included in the spectral region to fit. The result is the best estimation of the effective temperature of the considered stellar spectrum and the best fit Planck function model.

Cross-Correlation

This task cross-correlates any custom template spectrum with the input spectra and measures the wavelength shift, both in terms of radial velocity and redshift. It uses the "cross_corr" function of the "spectra_analysis" module. The result is the radial velocity or z of the spectrum, as well as uncertainty estimates via Monte Carlo simulations. You can select whether use the velocity mode (e.g. for stars and local galaxies/stellar clusters) of the redshift mode to perform the cross-correlation. Works good with strong absorption lines (e.g. the Calcium Triplet in the NIR) and for Doppler velocities up to +- 2000 km/s or redshift up to 0.1. For galaxies, this task is useful when used with the manual redshift estimation of the Preview window. **Example**: we have unknown redshift galaxy spectrum. We find the approximate value by manually shifting the selected spectrum in the preview, then we open the "Doppler/z correction" task of the Spectra manipulation panel and we set this approximate redshift value. We therefore use the cross-correlation task to find the fine-tuning differential value to be added to the estimated redshift (or you can use the "Stars and gas kinematics" task, if you prefer).

Velocity dispersion

This task provides a fast and robust estimate of the stellar velocity dispersion of a spectrum by fitting any stellar or galaxy template to the selected spectrum, within the selected spectral band. This task is designed for exploratory analysis and quick diagnostics. For final, publication-quality results, use the dedicated "Stars and gas kinematics" task, which rely on the pPXF algorithm.

The observed spectrum and the template are first log-rebinned to a constant velocity scale. The template is broadened by Gaussian kernels corresponding to trial velocity dispersions and matched in spectral resolution to the instrumental resolution of the data. A multiplicative Legendre polynomial is fitted simultaneously to correct continuum shape differences. The best-fitting velocity dispersion is identified by χ^2 minimization, with error estimation via bootstrap resampling (100 simulations).
If the template resolution is broader than (or comparable to) the instrumental resolution of the spectrum, the returned velocity dispersion value represents an **upper limit**. The task tolerates

moderately velocity shifts up to +- 50 km/s and it is about 4 times faster than the pPXF-based full kinematic analysis.

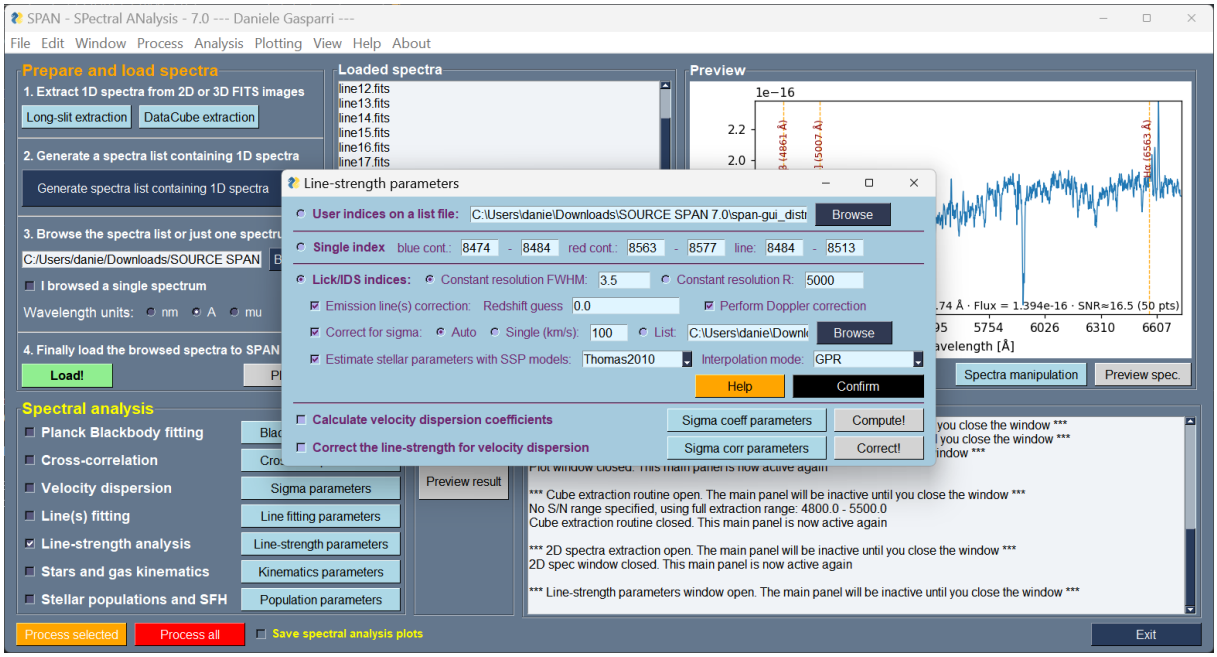
Line(s) fitting

This task allows to fit an absorption or an emission line with a combination of a straight line for the continuum and a Gaussian function for the line. You can automatically fit the three Calcium Triplet (CaT) lines in the NIR or any custom absorption/emission line. It returns the central wavelength of the line(s) and the width (sigma) in terms of Angstrom and km/s.

Advanced spectral analysis with SPAN

SPAN can perform all the major analyses of modern spectral analysis, by leveraging built-in routines and the pPXF full spectral fitting algorithm. Here is a detailed description of the more advanced analysis tasks.

Line-strength analysis



Line-strength analysis parameters

SPAN provides a complete and customizable line-strength analysis framework for both stellar and galaxy spectra using built-in routines. These routines have already been used in published works (e.g., Gasparri et al. 2021, 2024) and thoroughly tested. They measure the equivalent width (EW) of absorption and emission lines and their uncertainties with Monte Carlo simulations.

What you can do in this task

- Measuring the EW of a single index, a fully custom set of indices and the preset Lick/IDS using the definition stored in SPAN
- Calculating the velocity dispersion correction coefficients to correct the raw EW for the broadening effect due to velocity dispersion of the galaxies.
- Applying the velocity dispersion coefficients found to the raw EW measured for the custom indices
- Automatic Lick/IDS index line-strength analysis, which includes the emission line correction, rest-frame correction, correction of raw EWs for the broadening and estimation of stellar parameters (age, metallicity, alpha-enhancment) using the line-strength definition of preset SSP models.

Fully custom analysis using any line-strength index definition

This mode includes the **"User indices on a list file"** and **"Single index"** options. Your spectra **must already be redshift/Doppler corrected** before using them. The wavelength range and the resolution values should be set to the rest-frame reference. For **"User indices on a list file"**, you must create and load into SPAN a valid ASCII file containing the names and definitions of the indices you want to measure. An example file is included, and additional index templates can be found in the "example_files" folder. More information can be found in the user manual. For people in a hurry, here some tips:

- The ASCII file can be in .txt or .dat format, with single spaced values.
- The first row of the file must contain the arbitrary names of the indices.
- For any index, starting from the second row, you must provide in the following six rows: blue band limits (min, max), red band limits (min, max), line limits (min, max). Don't worry if the file is not properly formatted: SPAN will notice it and will complain, but it will not crash.

From raw EWs to science values

If you measure the EW of your indices on stellar or model spectra, you are set. But if you perform the measurements on galaxy spectra, this is just the first step. You have the raw EW, not necessarily the final, scientific values. The velocity dispersion of the galaxies, indeed, broadens the spectral lines and has effect on the EWs measured. If you are measuring the indices on a single galaxy with constant velocity dispersion, you maybe are ok, but this is a very specific situation that happens few times in the life of an astronomer. In most situations, to assign meaningful values to your measured raw EWs, you should account for the effect of the velocity dispersion broadening and therefore correct the raw values.

In the custom line-strength analysis you should take care personally of these two steps, but SPAN will help you. Following the approach of Trager et al., 1998 on the Lick/IDS indices, which has been replied (with some modifications) also in the near-infrared (NIR, e.g. Cesetti et al., 2013, Morelli et al., 2020, Gasparri et al., 2021), you can find two dedicated tasks for this:

1. **Calculate velocity dispersion coefficients:** By loading a set of stellar or model templates to SPAN and the same list of indices you measured on your spectra, SPAN will calculate the corrections and the uncertainties to apply for the raw EW or any index as a function of the velocity dispersion broadening. This is achieved by broadening each template spectrum to different velocity dispersion values, from 0 to 400 km/s in 50 km/s steps, and then fitting a spline function to retrieve the relation between the EW and velocity dispersion, for each index.
2. **Correct line-strength for velocity dispersion:** Once you have the coefficients, this task will help you to correct the raw EWs generated in the first step for the velocity dispersion broadening. You should also know the velocity dispersion of each spectrum. You can measure it with the "Velocity dispersion" or "Stars and gas kinematics" tasks. The "Velocity dispersion" tasks generates ASCII files ready for the "Correct line-strength for velocity dispersion" task. For the "Stars and gas kinematics" output, you just need to delete the RV row. See more details about how the required text files must be formatted in the user manual.

Automatic Lick/IDS index analysis

In this case, SPAN can do the dirty job for you. Activating all the options available, first a fit with pPXF is performed in order to measure the velocity dispersion and the redshift/Doppler velocity of any single spectrum, considering also the gas emission lines and subtracting them from the spectra. For high redshift galaxies, you do NOT need to de-redshift them. If the corresponding rest-frame wavelength range falls within the Lick/IDS index range, SPAN will notice it, if you inserted the right redshift value and activated the "Perform Doppler correction" option. In this case, the resolution is referred to the wavelength range of your redshifted spectra. If you are working with high redshift spectra (z > 0.01) SPAN will automatically adjust the resolution (if expressed in terms of FWHM) to the equivalent rest-frame wavelength range of the Lick/IDS indices. The definitions of the Lick/IDS indices are stored in the "system_files" folder of SPAN and automatically loaded. Also the resolution adjustment to bring your spectra to the Lick/IDS standard is performed by SPAN.

Since some of the Lick/IDS indices are used to constrain the properties of the stellar populations in unresolved spectra, you can activate the "Estimate stellar parameters with SSP models" and select which pre-loaded model use and the algorithm to perform the interpolation between data and models. The interpolation with the model-based Lick/IDS indices is performed considering the Hbeta–MgFe' indices of Thomas et al., 2003 and the Fe–Mgb index-index grids.

The interpolation between the measured and model indices is performed with three different approaches:

1. **Linear interpolation**
   A multi-dimensional linear interpolation using the "griddata" function from the "SciPy" module retrieves age, metallicity, and alpha/Fe (where available) from the observed Lick/IDS indices projected onto the model grids. Off-grid values are not calculated and return NaNs.
2. **Supervised Machine Learning**
   The **Gaussian Process Regression (GPR)** has proven to be a powerful tool for deriving stellar parameters from stellar spectra (e.g. Bu et al., 2015,2020). To my knowledge, no attempt has yet been made to apply GPR to unresolved galaxy spectra in the context of Lick/IDS line-strength studies. In SPAN, I trained the GPR algorithm with the line-strength predictions of the SSP models. The trained models, stored in the "system_files" folder, are used to constrain the age, metallicity, alpha/Fe and the relative uncertainties of the stellar populations. The GPR method results to be 10 times faster and yields results in excellent agreement with the linear interpolation method for spectra with S/N > 20. For lower S/N spectra, the GPR still converges whereas the simple linear interpolation fails to find a solution or gives very large uncertainties. The GPR option is available for the preset SSP models with alpha/Fe parametrization, such as Thomas 2010, sMILES and MILES. It is not available for the Xshooter models for which the simple linear interpolation will give fast and reliable results. **WARNING:** off-grid values are calculated by model extrapolation. Please, check the grid limits of your SSP models and decide whether trust the extrapolated off-grid values. This method is not available with Xshooter models which do not have alpha/Fe parametrization and for which the linear interpolation with the "griddata" function represents the best option for a simple 2 parameter space (age and metallicity). **IMPORTANT:** the GPR models have been trained using the version 1.7.0 of the scikit-learn module. If you are using a different version, results may be unreliable and Python will tell you loud and clear. You need to train new models. Don't worry, SPAN

takes care of almost everything. You just need to go to the "system_files" folder of your SPAN installation and delete the *.pkl files of Thomas2010, sMILES and MILES (3 files for each model, therefore 9 files in total). When using the GPR for the first time, SPAN will train again the models using your current version of scikit-learn and store them in the same folder. You do not need to worry about that if you installed SPAN as a standalone app for Windows, since the right scikit-learn module has been already embedded.

3. **Markov chain Monte Carlo simulations**
   The MCMC option fits age, metallicity, and alpha/Fe by forward-modeling the Lick indices used as population diagnostics from SSP grids and sampling the posterior in (log(age),met,alpha). It first anchors (log(age),met) with Hbeta and MgFe' of Thomas et al, 2003 (which is nearly alpha-insensitive), alpha/Fe from Mgb and Fe via linear interpolation with the "griddata" function, and then performs an MCMC with a Gaussian prior around the anchor(s) and a Gaussian likelihood with diagonal covariance (measurement errors plus small calibration floors). By default, it uses 48 Walkers and 3000 steps. The method is convex-hull safe: if the observed indices fall outside the SSP model domain, SPAN returns NaNs instead of extrapolating. Reported values are posterior medians with 16–84% credible intervals. **WARNING:** this method is computationally expensive and should be used with care for a small sample of spectra. It is available only for SSP models with Alpha/Fe parametrization (i.e. NOT for Xshooter models).

Outputs

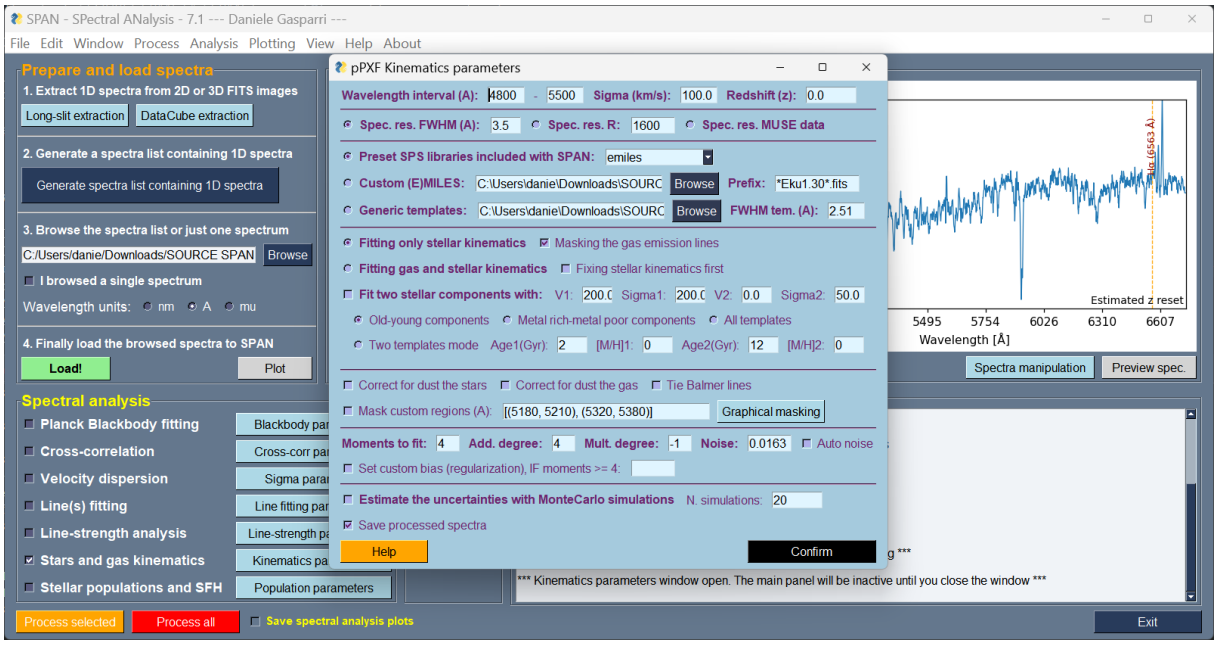
Using the custom line-strength analysis in the "Process all" mode, SPAN will save three ASCII files in the "line-strength_analysis" folder contained in the "SPAN_results" folder:

- A file with the raw EWs and uncertainties expressed in Angstrom.
- A file with the raw EWs and uncertainties expressed in Magnitudes.
- A file with the S/N per pixel calculated in the pseudo-continuum bands for each index.

Using the automatic Lick/IDS analysis in the "Process all" mode, SPAN will save the same ASCII files of the custom analysis, which contains the corrected EWs for the velocity dispersion broadening, if this option ("correct for sigma") has been activated. In addition:

- An ASCII file containing the EWs of the Lick/IDS indices used for stellar population studies (Hbeta, Mg2 (in mag), Mgb, Fe, MgFe') and their uncertainties.
- An ASCII file containing the EWs of these indices, and the stellar parameters estimated (age, metallicity and alpha/Fe) and their uncertainties, if "Estimate stellar parameters with SSP models" option has been activated.

Stars and gas kinematics



Stars and gas kinematics parameters

SPAN utilizes the capabilities of the pPXF algorithm (Cappellari 2023 and references therein) to perform a comprehensive kinematic analysis of both stellar and gaseous components in galaxy spectra, covering the optical and near-infrared (NIR) regions. This task can measure the line-of-sight velocity moments both of stars and gas, as well as the fluxes of the gas emission lines automatically found in the spectral region considered.

This interface does not display all the parameters and operations available in pPXF, which is a highly versatile tool for various astrophysical applications. However, it provides a user-friendly setup that will be suitable for most science cases.

The default parameters in this window offer a solid starting point for typical galaxy spectra. To perform an initial fit, simply enter the redshift of the spectrum and ensure that the spectral range falls within the selected wavelength range. Fine-tuning the fit requires adjusting specific parameters.

What you can do in this task

- Fitting one stellar component alone
- Fitting one stellar component and gas emission lines at the same time
- Fitting one stellar component, then fix the kinematics to fit the gas kinematics
- Fitting two stellar components, alone or with the gas (one kinematic component for the gas), automatically or manually selecting template considering: old-young, metal rich-metal poor, two exact templates or all the templates. Useful for e.g. counter-rotating galaxies.
- Using any set of SSP templates for one stellar component mode. Two components mode works with the embedded templates and any (E)MILES custom template set.
- Setting the moments to fit, the noise (also with the “auto noise” option), the regularization (bias) if different than the default pPXF value, stellar dust, gas dust, additive and/or multiplicative polynomial degrees

- Masking custom regions you do not want to be considered in the fit
- Perform MonteCarlo simulations for uncertainties estimation, when the formal uncertainties given by pPXF cannot be trusted.

Parameter Overview

The parameters are grouped into seven sections, separated by horizontal dividers. Below is a brief description of each:

First Section: Basic Parameters

- Wavelength Range: Defines the spectral range to be fitted. If your spectra are not rest-frame corrected, the wavelength range to insert is the one displayed by your spectra. A robust estimation of both stellar and gaseous components is achieved by selecting the visible band region 4800-5500 A (or its equivalent range for high-redshift galaxies), which includes key absorption and emission lines. This is a commonly used standard range. In the NIR, the most prominent spectral features for stellar kinematics are the Ca II triplet (CaT) lines in the 8400-8800 A rest-frame range.
- Sigma (km/s): An initial estimate of the expected velocity dispersion of your galaxy spectrum.
- Redshift (z): An approximate redshift value of the spectrum. **Important:** Avoid loading spectra with significantly different redshifts. If necessary, de-redshift your data before performing kinematic analysis. This can be done using the "Doppler/z correction" task in the Spectra manipulation panel.

Second Section: Spectral Resolution

This section requires the spectral resolution of the spectra. This is a fundamental parameter in order to perform a correct estimation of kinematic parameters. It is essential to **avoid mixing spectra with different resolutions within the same dataset.** The resolution can be expressed in terms of:

- FWHM (Full Width at Half Maximum)
- Resolving Power (R = Lambda/DeltaLambda)
- MUSE LSF. SPAN contains the parametrization of MUSE Line Spread Function based on the Eq.8 of Bacon et al. (2017). If you have MUSE data, please select this option. If fitting a narrow spectral region (<= 1000 A), the choice between constant FWHM, R or MUSE resolution is not critical. However, for broader wavelength ranges, selecting the appropriate resolution type is crucial. **Important:** For high-redshift galaxies (z > 0.01) and spectra which have not been de-redshifted, SPAN will automatically correct the spectral resolution given in FWHM and MUSE LSF to the corresponding rest-frame wavelength range.

Third Section: Template Selection

Here you can select the SSP Model Library to use for the fit. The available pre-loaded libraries are:

1. E-MILES (subsample)

2. Galaxev (subsample)
3. FSPS (subsample)
4. X-shooter Spectral Library (XSL) (complete sample with Salpeter IMF, better suited for higher resolution spectra, R = 10,000)

If your spectra have a higher resolution than the templates, you should degrade them to match the template resolution using the "Degrade Resolution" tool in the Spectra manipulation panel. You can use any kind of EMILES templates or generic ones. The generic templates should have the wavelength scale in linear units and Angstrom. **WARNING:** Two stellar component fit is not available with a generic template set option (see next section).

Fourth Section: Which component to fit

1. "Fitting only stellar kinematics": Considering only the stellar component. To prevent gas emission lines in the optical, the option "Mask emission lines is activated by default". If you are fitting the optical window and are absolutely sure to not have emission (usually in HBeta and HAlpha) you can disable this option.
2. "Gas and Stars Kinematics": Considers both stellar and gaseous emission lines. The gas components are divided into three families: Balmer lines, forbidden lines, other lines. SPAN will automatically fit all the components found in the spectral range given, using a modified version of the line emission template provided by pPXF and embedded in the "spectral_analysis.py" module in the "span_functions" folder. By default, stars and gas kinematics are calculated together with one fit. However, often is necessary to fix the stellar kinematics to perform a more accurate gas fit. The option "Fixing stellar kinematics first" will perform a first fit only for the stellar components and will use the kinematics moments derived to fix them in the subsequent fit for the gas component.
3. Two stellar component fit option: Here you can also decide to fit two stellar components by activating the "Fit two stellar components with:" checkbox. This option is important only when you have kinematic decoupled stellar components (e.g. counter-rotating discs) with high S/N (> 50). It can be activated both with stellar and gas fit option, but gas will always be fitted with one kinematic component. Dust correction is not available (both for star and gas) in this mode. You need to insert an accurate velocity and velocity dispersion guess for both the stellar components. Beside this, you can select the templates to assign to each component, based on three options: 1) Old (> 5 Gyr) and young (< 5 Gyr) components, 2) Metal rich ([M/H] > 0) and metal poor ([M/H] < 0) components, 3) pick two templates with fixed age and metallicity. This option has been used in some works with counter-rotating discs as such Rubino et al., 2021. **WARNING:** Two component fit is available only by using the pre-set template set of SPAN and with any (E)MILES template. Not available for fully custom templates, since it needs to read the information about age and metallicity of the SSPs. **IMPORTANT1:** For the two component fit you must always insert a GOOD guess of the velocity and velocity dispersion of the two components you think to see in your spectrum. I stress out that a good guess is necessary (according to Cappellari et al., 2023 and the pPXF documentation), so first take an accurate look at your spectrum. Some tries may be necessary to find the correct parameters and it is better to perform these tests on spectral regions not affected by gas (e.g. the CaT lines) with the option "Fit only stellar kinematics". **IMPORTANT2:** If "save processed spectra" is activated, in the two component fit mode SPAN will save the bestfit template from pp.bestfit provided directly by pPXF. However, this bestfit template contains the two components summed up. If you want to separate and analyze each stellar compo-

nent, SPAN will also save the two besftit templates for each stellar component (suffix comp1_ and comp2_) so that comp1_+comp2_ = pp.bestfit. This uses the pp.matrix keyword of pPXF to reconstruct the bestfit models of each stellar component. Be aware that if you intend to use these separated bestfit templates for further analysis besides the kinematics (e.g. line-strength, SFH) you SHOULD perform the two component fit using ONLY multiplicative degree and deactivating the additive (set to -1). If using additive degree, SPAN will reconstruct and save the two separated bestfit templates without considering the additive polynomials you gave, so you should expect comp1_+comp2_ != pp.bestfit. This should preserve the line fluxes anyway, but it's better to simply avoid the additive degree polynomial, where possible. When gas emission is present, is expected that comp1_+comp2_ != pp.bestfit, since here pp.bestfit contains also the emission lines. **IMPORTANT3:** pPXF requires spectra to be normalized to unity to avoid instabilities, but the gas flux calculated on normalized spectra is not physically useful. Therefore, during the "Stars and gas kinematics" task with the option "Stellar and gas kinematics", SPAN saves the normalization factor of each spectrum (np.median(galaxy)) and multiplies by this factor the gas flux calculated by pPXF to give a physical flux for the gas. These gas flux values are NOT shown in the pPXF output in the terminal (I cannot modify the pPXF output), but are saved and stored in the ASCII file generated in the "Process all" mode.

Fifth Section: Dust and masking

Here you can activate the dust/extinction corrections or decide to mask custom regions of your spectra.

The dust correction uses the Cappellari (2023) 2-parameter attenuation model for the stellar component and the Calzetti (2000) 1-parameter attenuation model for the gas. If you are interested only to kinematics (i.e. no gas flux) you can safely neglect the dust parametrization in most situations. **WARNING:** Dust correction is not available for the two component fit mode.

The masking will act directly on the "goodpixels" keyword of pPXF and can be used also when the emission line masking is activated. There are two masking options available:

- Manual masking, by inserting the wavelength interval(s) to be masked, separated by commas and parenthesis in the text box. If invalid values are inserted, SPAN will warn you.
- Graphical masking mode, activated by pressing the "Graphical masking" button. In this mode, an interactive Matplotlib window will open displaying the spectrum selected. You can then mask custom portion directly on the spectrum by ctrl+left click and drag. You can deselect the masked region by ctrl+right click and drag. On touchscreen devices (i.e. Android systems), masking and unmasking modes are activated by a tap on the screen and the relative selection is done by tapping and dragging on the spectrum. When graphical masking in done, you can close the Matplotlib window and the text box on the left will update with the new ranges selected.

Sixth Section: Kinematic Parameters

- Gauss-Hermite Moments: Determines the complexity of the Line-of-Sight Velocity Distribution (LOSVD) model. Minimum moments to be fitted: 2 (radial velocity + velocity dispersion). Maximum moments: 6. Typical values: 2 for regular galaxies and/or for the two stellar component fit, 4 for interacting or asymmetric galaxies. **IMPORTANT:**

moments are always referred to the stellar component. Gas moments are set to 2 by default and cannot be changed.

- Polynomial Degree: Specifies the degree of additive and multiplicative polynomials used by pPXF to adjust the continuum level of spectral templates. Additive polynomials give reliable kinematics results. A degree of 4 is a good starting point. **WARNING:** Additive polynomials are fine for pure kinematic analysis but not for measuring the gas line fluxes. If you are interested also to a correct measurement of gas line fluxes, you should deactivate the additive degree polynomials (set the degree to -1) and use ONLY the multiplicative degree polynomials.
- Noise Level: Represents the expected mean noise level in the spectrum (assumed constant across the wavelength range), used to compute the Chi^2 of the fit and derive formal uncertainties. As per the pPXF documentation, formal uncertainties are meaningful only if Chi2 = 1. If unsure about the noise level, enable "Auto Noise". This feature will:
  1. Perform an initial fit using the user-provided noise level without regularization (bias keyword set to zero).
  2. Compute a refined noise estimate.
  3. Re-run the fit using the updated noise value to obtain more accurate LOSVD parameters.
- Custom bias keyword: To be used only when you fit at least 4 moments. This lets you to change the default bias value that pPXF uses (bias = 0.7*np.sqrt(500./goodpixels.size), from Cappellari & Emsellem 2004) and try to adjust the fit to a better estimate of H3 and H4 moments.

Seventh Section: Uncertainty Estimation

To estimate uncertainties in stellar kinematics, you can enable Monte Carlo simulations. This option is recommended if you do not fully trust the formal uncertainties computed by pPXF. Particularly useful for galaxies with very low velocity dispersion compared to the instrumental resolution (velscale). This operation is very time consuming, so try to not insert too many simulations.

Outputs

In **"Process selected"** mode, if the option "Save processed spectra" is activated, the task produces:

- bestfit model spectrum
- bestfit model spectrum of the stellar component 1 and 2 (if "Fit two stellar components" option is activated)
- bestfit gas model spectrum (if kinematics of stars and gas is activated)
- emission corrected spectrum (if kinematics of stars and gas is activated)
- continuum subtracted gas spectrum (if kinematics of stars and gas is activated).
  These spectra are saved in the "processed_spectra" subfolder within the "SPAN_results" folder. If you are not interested in these spectra products, you can just disable the option "Save processed spectra" and save some space on the disk.

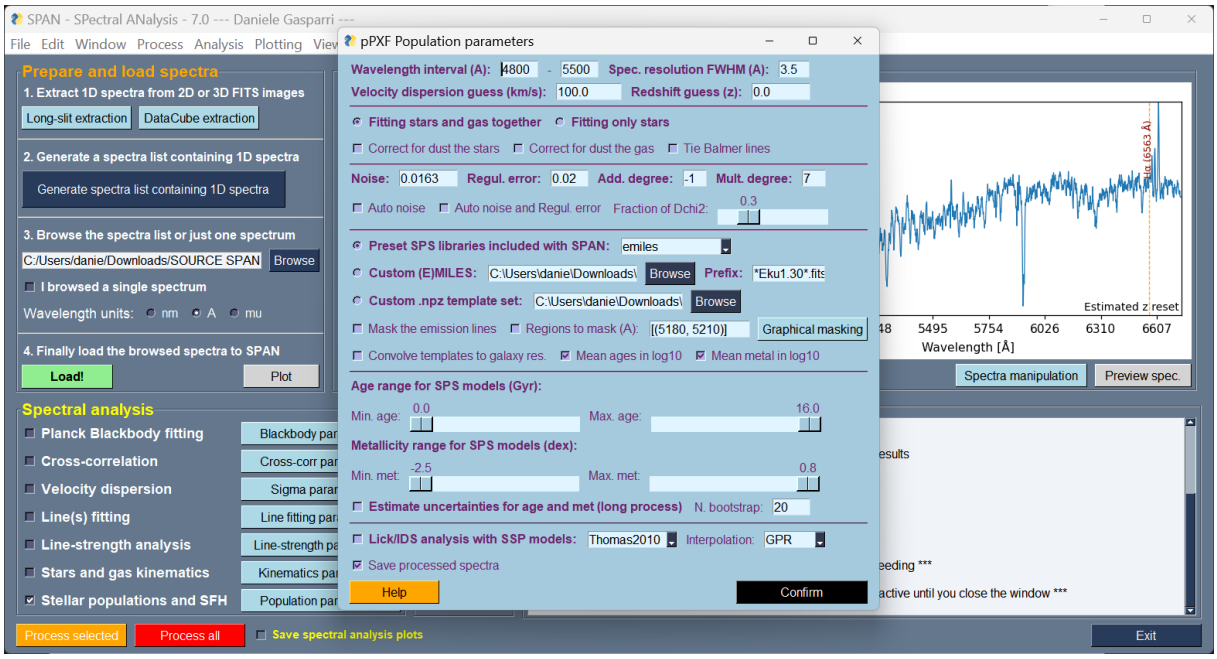
In **"Process all"** mode, for each spectrum, if "Save processed spectra" is activated, the task produces all the spectra in the "Process selected" mode, for each spectrum processed.

Moreover, an ASCII file (.dat) is stored in the "stars_and_gas_kinematics" subfolder containing

the kinematics moments and the formal errors of the stellar component(s). If the option “Fitting gas and stellar kinematics together” is activated, an ASCII file containing the kinematics, flux and formal errors of the emission lines found is generated.

If “Estimate the uncertainties with MonteCarlo simulations” is activated, another ASCII file is generated containing the stellar kinematics and the MonteCarlo uncertainties. MonteCarlo errors are not calculated for gas.

Stellar populations and SFH



Stellar populations and SFH parameters

This task is one of the most critical within the spectral analysis framework and extragalactic astronomy in general. It leverages the well-known pPXF algorithm (Cappellari 2023 and references therein) to fit a galaxy spectrum using a set of Simple Stellar Population (SSP) models, extracting key stellar population parameters such as age, metallicity, Star Formation History (SFH) and mass growth.

The provided settings allow for an optimal use of pPXF, though not all of its parameters and functionalities are included in this interface.

Requirements for Reliable Results

The full spectral fitting method requires spectra with an acceptable Signal-to-Noise Ratio (S/N):

- S/N > 15 for basic results.
- S/N > 50 for high-accuracy results.

The default settings offer a good starting point for most local galaxy spectra and common spectrographs. To perform an initial fit, simply enter the redshift of the spectrum. For fine-tuning, adjust the parameters based on your specific dataset.

What you can do in this task

- Fitting one stellar component alone

- Fitting one stellar component and gas emission lines at the same time
- Using the embedded templates, any set of SSP templates following the MILES standard name convention, or any template set saved in .npz file following the pPXF standard
- Setting the moments to fit, the noise (also with the “auto noise” option), the regularization (also using “auto noise and Regul. error” option), stellar dust, gas dust, additive and/or multiplicative polynomial degrees
- Masking custom regions you do not want to be considered in the fit
- Custom selecting the age and metallicity range of your templates to be considered for the fit
- Performing Bootstrap simulations for uncertainties estimation for the SFH and stellar mean parameters
- Performing also the Lick/IDS line-strength analysis by leveraging the results of pPXF

Parameter Overview

The parameters are grouped into six sections, separated by horizontal dividers. Below is a brief description of each:

First Section: Basic Parameters

- Wavelength Range: Defines the spectral region to be fitted. A good estimate of stellar population parameters in galaxies is obtained by focusing on the 4800-5500 A (rest-frame) which includes age, metallicity, and SFH-sensitive lines such as Hbeta, OIII, Mg, and Fe. Il you have near-infrared (NIR) spectra, the 8400-8800 A (rest-frame) range is good, which includes the Ca II triplet.
- Spectral Resolution FWHM (A): Approximate resolution of your spectrum **in the selected wavelength range**. This is relevant only if including gas emission lines in the fit or if the option “Convolve templates to galaxy resolution” is activated (not mandatory). A precise value is not required, but a good estimate is recommended. If working with high redshift spectra which have not been de-redshifted, the resolution value in terms of FWHM is referred to your actual wavelength range. SPAN will scale this value considering the redshift inserted by a factor: FWHM_rest = FWHM/(1+z).
- Velocity Dispersion Guess (km/s): An approximate estimate of the actual stellar velocity dispersion.
- Redshift Guess (z): An initial estimate of the spectrum’s redshift. If this value is not within +/- 0.005 the real one, the fit will likely fail.

Second Section: Gas Emission Lines and Dust Attenuation

- If the spectrum contains emission lines, you should select “Fitting with gas” for more accurate results. Alternatively, you can mask out the emission lines with the “Mask emission lines” option and perform a fit without considering the gas (“Fitting only stars”)
- If the spectrum includes multiple Balmer emission lines, you may enable “Tie Balmer” to constrain their ratios. This automatically applies the Calzetti et al. (2000) dust attenuation curve. Two additional options allow for dust correction. They should be used only for spectra with very good flux calibration:
- “Correct for dust the stars”: Uses the Cappellari (2023) 2-parameter attenuation model.
- “Correct for dust the gas”: Uses the Calzetti (2000) 1-parameter attenuation model.

Using or not dust correction depends on case-to-case basis. Usually, multiplicative polynomials gives reliable results and the addition of dust correction produces very similar results. For a deeper understanding of dust correction ad its effect on the stellar populations, please see Lee et al., 2024 (https://arxiv.org/pdf/2406.13924)

Third Section: Noise Estimation, Regularization and polynomials

Noise and Regularization: These are critical for obtaining accurate results. Refer to the pPXF documentation for an in-depth guide. Recommended workflow for finding the best parameters:

- Perform an unregularized fit (Regul. error = 0).
- Adjust the noise level to achieve chi^2 = 1.
- Set the regularization so that the current delta Chi^2 falls between 20-50% of the desired delta Chi^2.

This workflow can be automated by SPAN. Use "Auto Noise" option, which performs an initial unregularized fit to determine the optimal noise level. For small spectral ranges (<1000 A) and limited templates, "Auto Noise and Regul. error" can be used to optimize both the noise and the regularization. A good starting point for S/N = 50 is setting "Fraction of Dchi2 to reach" to 0.20.

Polynomial Adjustments:
Additive Degree: Leave disabled (-1) for reliable results, since they may change the absorption line profiles. Multiplicative Degree: Adjust based on spectral range (about 1 degree per 100 A). **Example:** For 4800-5500 A, set Mult. degree = 7.

Fourth Section: Template Library Selection

Choose the SSP model library for fitting. The pre-loaded templates are:

- E-MILES, Galaxev, FSPS (pPXF defaults, they are a subsample of the respective libraries).
- X-shooter Spectral Library. A full sample in the optical range with Salpeter IMF and Parsec/Colibri isochrones. They have a resolving power of R = 10,000 and are better suited for medium to high resolution galaxy spectra.
- sMILES Library (Knowles et al., 2023). Full sample in the optical window with 4 alpha/Fe values and Salpeter IMF. They are useful to extract also the alpha/Fe values besides age and metallicity.

You can also use any kind of template set with the standard (E)MILES file naming convention by selecting the "Custom (E)MILES" option. Browse the folder where your (E)MILES templates are stored, adjust the common prefix of the templates and just fit. **WARNING1:** pPXF requires a regular grid of age and metallicity, so be sure that your template set fulfil this request. **WARNING2:** If your spectra have much higher resolution than the templates, apply the "Degrade Resolution" task in the Spectra manipulation panel before fitting may improve the results.

Template Handling:
The sMILES templates are stored in spectralTemplates/sMILES_afeh. This folder can hosts any kind of sMILES templates, not just those provided with SPAN. If replacing templates, ensure:

- A regular grid of ages and metallicities.
- No mixing of different IMF templates.

You can use any set of (E)MILES templates that follows the standard MILES file naming convention (see http://research.iac.es/proyecto/miles/pages/ssp-models/name-convention.php) and also create and browse a custom .npz file following the standard of pPXF and containing any set of templates.

Important Notes on Stellar Population Models
E-MILES, sMILES, and X-shooter work best for quiescent and moderately star forming galaxies but lack very young stellar populations (<30-60 Myr). For highly star-forming galaxies, consider FSPS (Conroy et al., 2010) or other templates with (also) very young ages. If you use a set of templates that does not properly sample the needed very young regime, you will see a very young, very metal poor population appearing in your fit. That is a red flag!

Fifth Section: Custom Masking and Stellar Constraints

- You can mask out the emission Lines. In this case, you should select the "Fitting without gas" mode (see Section 2).
- Custom masking is available and can be added also to the automatic masking of the emission lines. There are two custom masking options available:
    1. Manual masking, by inserting the wavelength interval(s) to be masked, separated by commas and parenthesis in the text box. If invalid values are inserted, SPAN will warn you.
    2. Graphical masking mode, activated by pressing the "Graphical masking" button. In this mode, an interactive Matplotlib window will open displaying the spectrum selected. You can then mask custom portion directly on the spectrum by ctrl+left click and drag. You can deselect the masked region by ctrl+right click and drag. On touchscreen devices (i.e. Android systems), masking and unmasking modes are activated by a tap on the screen and the relative selection is done by tapping and dragging on the spectrum. When graphical masking in done, you can close the Matplotlib window and the text box on the left will update with the new ranges selected.

Age and Metallicity Ranges:
Limit the maximum template age and metallicity based on the galaxy's cosmic age. For high-redshift galaxies, reducing the upper age limit is strongly recommended. **Example:** we are analyzing a galaxy located at z = 1.20, corresponding to an epoch when the Universe was about 5 Gyr old. You can move the maximum age slider to 5 Gyr to consider only SSP younger than this value for the fit.

Sixth Section: Uncertainty Estimation and Lick Indices

Bootstrap Uncertainty Estimation (Kacharov et al., 2018) can be enabled to compute error on age and metallicity. Suggested simulations: 20-50 (balance between accuracy and speed).

Lick/IDS Index Analysis:
Uses the pPXF emission-corrected spectrum to estimate stellar population parameters through the measurement of Lick/IDS spectral indices. You can select the model grids and interpolation mode in the same way as in the "Line-Strength Analysis" task. If your spectra have a significantly different resolution compared to the SSP templates used in pPXF, it is recommended to activate the option "Convolve templates to galaxy resolution". This ensures a more accurate estimate of the velocity dispersion during the fit, which is required to properly correct the Lick/IDS indices for spectral broadening.

Final Notes

By default, templates are V-band normalized (5070-5950 A), meaning age, metallicity, and SFH are V luminosity-weighted. Using pPXF .flux attribute, SPAN derives mass-weighted results from the same fit. Best regularization differs for mass-weighted vs. luminosity-weighted results. If mass results are the priority, fine-tune the "Regul. error" accordingly.

Outputs

Every fit generates three plots both in the **"Process selected** and **Process all"** mode:

- Fitted Spectrum
- Age and Metallicity Distribution (luminosity-weighted and mass-weighted).
- Non-parametric SFH (luminosity-weighted and mass-weighted). Carefully analyze these plots to determine whether the fit is physically meaningful. No computer can yet decide if a fit, even if statistically excellent, makes sense in the real universe.
- An ASCII file (.dat) containing the luminosity and mass fraction and cumulative as a function of the age (mass growth). If the "Estimate the uncertainties for age and met (long process)" is activated, also the relative uncertainties are added to the file. This file is stored in the "stellar_population_and_sfh/SFH/" subfolder.
- Two ASCII files containing the luminosity and mass weights respectively. These files are stored in the "stellar_population_and_sfh/weights/" subfolder
- FITS files containing: 1) residual of the fit, 2) bestfit template, 3) emission corrected spectrum (if gas is considered), stored in the "processed_spectra" subfolder.

In **"Process all"** mode, the task stores all the files of the "Process selected" mode for each spectrum. In addition:

- An ASCII file with kinematics, mean luminosity and mass age, metallicity and alpha/Fe (if available) and the relative uncertainties if "Estimate the uncertainties for age and met (long process)" if activated, stored in the "stellar_population_and_sfh" subfolder.

Tips and tricks

Here are some useful tips to help you master SPAN.

1. **Setting the Correct Wavelength Units**
   Due to the variety of 1D spectral formats, SPAN requires you to manually set the wavelength units before loading the spectra. It does not matter whether the spectra use linear or logarithmic wavelength values: you must specify if the units are in Angstrom (A), nanometers (nm), or micrometers (mu). If the wavelength units are incorrect, SPAN will display the spectrum with the wrong scale. **Example:** If your optical spectra are in A but you set nm, the Preview frame will show a range from 48000 to 55000 A, which is incorrect. To fix this, select the correct wavelength units, click the "Load!" button again and see if things are right now.

2. **Saving and Loading Parameters**
   You can save your session settings, active tasks, and parameters as a .json file via "File –> Save Parameters…". If you prefer to always load custom default parameters instead of SPAN defaults, save your parameters via "File –> Save Parameters…" and overwrite the "default_settings.json" file in the "system_files" subfolder. **WARNING**: If you modified the ordering and/or the number of your spectra in the listbox with respect to the original spectra list you loaded, SPAN will NOT save this new arrangements with the "File –> Save Parameters…" command, but only the name and the path of the original spectra list you loaded. If you want to store also the new modified spectra list within in the "Save parameters…" command, you first need to save the modified spectra list manually ("Edit –> Save current spectra list…"), then load it to SPAN, and therefore save your session with "File –> Save Parameters…". If you are interested to save only the parameters and the selected tasks, you can ignore this step.

3. **Managing Spectra Lists and File Organization**
   Spectra loaded into SPAN are organized via a spectra list, generated using the "Generate Spectra List containing 1D spectra" button. Spectra and spectra lists can be stored anywhere on your computer. The spectra lists may also contain non-spectral files (e.g., documentation files): SPAN will automatically ignore invalid files when loading the spectra list. If your spectra folder contains subdirectories, SPAN will scan all subfolders for valid spectra. Use this feature carefully, as it may include unintended files.

4. **Managing SPAN Results and Directories**
   SPAN stores results in the "SPAN_results" directory. The location of this directory must be decided by you the first time SPAN is run. You can move or delete its content at any time, but it is recommended to do so only when SPAN is not running. If SPAN detects missing directories upon startup, it will automatically recreate them.

5. **Improving Lick/IDS Line-Strength Analysis Accuracy**
   The "Emission Line(s) Correction" option in the Lick/IDS line-strength analysis uses pPXF to remove gas emissions from the spectra. This process fits the gas component using E-MILES templates in the 4000-6500 A wavelength range. If the spectra contain edge artifacts (e.g., from X-shooter UVB and VIS arms), the fit may be imperfect. Solution: Activate the "Cropping" task in the Spectra manipulation panel and crop the spectra to remove problematic regions. If you want to analyze only the 4800-5500 A range for stellar populations, crop the spectra accordingly.

6. **Lick/IDS analysis within the "Stellar populations and SFH" task** You can perform the Lick/IDS line-strength analysis also within the "Stellar populations and SFH" task by leveraging the results of the fit from pPXF. You just need to activate the option "Lick/IDS analysis with SSP models". Here, you can fit the spectra with any template, on the contrary of the Lick/IDS analysis in the "Line-strength analysis" task, and fully adjust the parameters of the fit. In this case, it is strongly recommended to fit also the stellar and gas component and to activate the "Convolve templates to galaxy resolution" option, which allows pPXF to perform a reliable estimation of the velocity dispersion. This is needed to correct the equivalent widths of the Lick/IDS indices for the velocity dispersion broadening and give more accurate results.

7. **Using "Stellar Populations and SFH" for Kinematics**
   The "Stellar Populations and SFH" task can be used for kinematics analysis too (with some limitations), as an alternative to "Stars and Gas Kinematics". This will let you to just perform one fit both for kinematics and star formation history. To adapt the settings also for kinematics:

   - Enable "Convolve templates to galaxy resolution".
   - Enable "Mask emission lines" if focusing on stellar components.
   - Set Regul. error = 0 to disable regularization.

   WARNING: The returned kinematics (only stellar component) are good in most cases but remember that the regularization factor here is targeted for the star formation history and not kinematics, so with asymmetric galaxies the higher moments (H3, H4) could not be very accurate.

8. **Handling High-Redshift Spectra (z > 0.01)**
   You have two options for handling redshifted spectra:

   - Option 1 (Recommended for multiple tasks): De-redshift the spectra using the "Doppler/z Correction" in the Spectra manipulation panel. Then, set redshift = 0 in the analysis tasks.
   - Option 2: Enter the real redshift value directly in each spectral analysis task. **Important:** When de-redshifting spectra, the corrected spectral resolution, if expressed in terms of delta lambda, is: FWHM_gal/(1 + z). If spectra are not de-redshifted, SPAN will automatically adjust the resolution in kinematics and stellar population analyses.

9. **Key Considerations for pPXF and Lick/IDS Analysis**
   When performing kinematics or line-strength index analysis with pPXF, the resolution of the templates must be higher (i.e., lower FWHM) than the resolution of the spectra. If this is not the case, degrade the resolution using "Degrade Resolution" in the "Spectra Manipulation" panel. The Lick/IDS index analysis uses a pre-loaded subset of E-MILES templates (Padova isochrones, FWHM = 2.51 A) to estimate the velocity dispersion and subtract the gas emission lines from the stellar flux. If your spectra have higher resolution (FWHM_spec < 2.51 A), degrade the spectra to at least 2.51 A before fitting.

10. **Stellar populations with Lick/IDS indices and machine learning GPR**
    The determination of the stellar parameters with the Lick/IDS indices and the GPR method uses predictive machine-learning models and the scikit-learn module. The default models stored in the system_files folder have been trained with the scikit-learn 1.7.0 version. If you are using different version for this library, the predictions may be

inaccurate and a warning is issued in the output window. In this case you should delete the trained models stored in the "system_files" folder (the files have the .pkl extension) and run again SPAN. If the Lick/IDS task with GPR stellar parameters determination does not find the trained models, it will create again the first time is executed, using the scikit-learn version you have installed. This will require a little time, but then you can use these new trained models until you change the version of the scikit-learn module. If you are using SPAN as a standalone application, you can ignore this point since the embedded scikit-learn module is already set at version 1.4.2 and you will not receive any warning.

11. **Using sMILES Templates for Stellar Populations and SFH**
    The sMILES SSP templates included in SPAN are a subset of the full library, using a Salpeter unimodal IMF. They are the only semi-empirical SSPs that sample a wide range of alpha/Fe values and are stored in the "SpectralTemplates/sMILES_afeh" folder. You can replace the content of the "sMILES_afeh" folder with a different sMILES SSP dataset (e.g., using a different IMF) and work with any sMILES template.

12. **Processing Spectra with Different Redshifts**
    If your dataset contains spectra with different redshifts (Delta(z) > 0.002) and you wish to use the "Process All" mode, you must first de-redshift them, since you can insert only one z value in the analysis tasks that require a redshift guess (i.e. Lick/IDS indices, Stars and gas kinematics, Stellar populations and SFH). Use "Doppler/z Correction" –> "I have a file" In the Spectra manipulation panel to apply individual redshifts. This ensures that all spectra are in the rest frame before spectral analysis.

13. **Handling High-Emission Line Galaxies**
    If analyzing spectra with strong emission lines (i.e. with high star formation), using E-MILES or X-Shooter SSP templates may produce an artificially young and metal-poor population. If this occurs, use the FSPS library, which includes SSPs as young as 1 Myr (compared to 30-60 Myr in E-MILES and X-Shooter), or create your set of templates.

14. **Speeding Up Spectral Analysis**
    Cropping spectra to include only the wavelength regions of interest can significantly speed up spectral analysis.

FAQ

1. **I cannot load any spectra because SPAN always complains. What's wrong?**
   Well, this is a quite general question. Here I give you some advices:

   a) If you want to load just one spectrum, load it directly without generating a spectra list, but activate the "I browsed a single spectrum" option before pressing the "Load!" button

   b) If you have a spectra list file which has not been generated by SPAN, check its format. Remember that the first line should contain a comment (#comment), then you need to list the absolute path and the full name of your spectra, unless they are stored in a subfolder of SPAN: in this case also the relative path is good

   c) Check the architecture of your spectra. Due to the fact that astronomers rarely agrees with each others, there are plenty of spectra formats out there. SPAN tries to consider all, but it may fail. Try with different spectra to see if the problem persists.

2. **Why do I see a weird wavelength scale in my loaded spectra?** SPAN will always show the wavelength scale of the plots in Angstrom (A). If the Preview will show you a strange wavelength range of your spectra, you likely selected a wrong wavelength unit scale. In order to be as general as possible and account to the many creative ways that astronomers have to define the wavelength units and keywords in the spectra, SPAN doesn't even try to guess it. You should set manually in the "Wavelength units:" option in the "Prepare and load spectra" frame. Of course, all the spectra in your list should share a common wavelength units (but they can be both logarithmically and linearly binned). If the problem persists, it is possible that your spectra have "fancier" wavelength units and cannot be properly read by the program. The available wavelength units handled by SPAN are Angstrom (A), nanometers (nm) and micrometers (mu). If your spectra use different wavelength units, SPAN cannot read them.

3. **How do I load just one spectrum?** If you want to work with just one spectrum, you DO NOT need a spectra list. Simply browse your spectrum and activate the checkbox "I browsed a single spectrum". Set the correct wavelength units and click "Load!".

4. **How do I handle a lot of spectra stored in different folders and within a lot of non spectra files?** There is a simple answer for that. The "Generate spectra list containing 1D spectra" will scan the selected folder and all the relative sub-folders looking for fits, .txt and .dat files. You just need to put all your spectra in this root folder, which can contain as many subfolders as you want. Then, once you load this spectra list, SPAN will recognize automatically which file is a spectrum and which not and will delete from the listbox all non valid spectra. Anyway, I advice you to create working folder or subfolders as clean as possible from non spectra files.

5. **How do I reset the parameters and the tasks if I played for so long that the entropy of SPAN now tends to infinite?** Don't panic! Click "Edit → Clear All Tasks" to deactivate all the tasks, or a more aggressive "File → Restore Default Parameters" to restore the default parameters.

6. **Can I save and reload my custom parameters?** Yes! Use "File → Save Parameters…" and "File → Load Parameters…".

7. **Where do my processed spectra and analysis results are saved?** Outputs are stored in the:

`SPAN_results/`

folder, whose location is determined by you the first time you open SPAN. You can, of course, change the location of the SPAN_results folder whenever you want, by “Edit → Change result folder…”. If you don’t remember the location of the folder, try “Edit → Show result folder”.

Known issues

SPAN's features work on all supported operating systems. However, some GUI quirks can appear depending on how Tkinter, the display server (Wayland/X11), and OS scaling interact. The following are minor visual issues currently known that I was not able to solve.

- **Reset zoom on Windows:** After zooming in/out, if you open and close one or more subwindows and then choose View → Reset Zoom, the main panel may not fully return to its initial geometry. **Workaround:** Set your preferred zoom level at the start of a session and keep it unchanged.
- **Screen scaling on Linux:** On some Linux distributions (especially Ubuntu on Wayland), when desktop scaling > 100% the main GUI elements can lose perfect alignment. This is cosmetic only; functionality is unaffected. The behavior stems from how Tk handles HiDPI/fractional scaling on certain Linux setups. **Workaround:** work always with the screen scaling set to 100% (this is a general advice, since some Linux distributions do not work well with fractional scaling). If the GUI panel appears small, use the zooming option of SPAN: View → zoom in. If you find a reliable fix for your environment (e.g., specific Tk/OS settings), please let me know: I'm happy to test and incorporate improvements.
- **Output window on macOS:** The embedded terminal window of SPAN on macOS systems has been disabled since it slowed-down all the tasks by a factor 4-5.

License

1. This software is licensed **for non-commercial, academic and personal use only**.
2. The source code may be **used and modified** for research and educational purposes, but any modifications must remain for private use unless explicitly authorized in writing by the original author.
3. Redistribution of the software in its **original, unmodified form** is permitted for non-commercial purposes, provided that this license notice is always included.
4. Redistribution or public release of **modified versions** of the source code is **prohibited** without prior written permission from the author.
5. Any user of this software must **properly attribute the original author** in any academic work, research, or derivative project.
6. **Commercial use** of this software is strictly prohibited without prior written permission from the author.

DISCLAIMER:
THIS SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES, OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT, OR OTHERWISE, ARISING FROM, OUT OF, OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Changelog

Version 7.1 (20251020):

- Improved and bug corrected the smoothing option in the 'Plot maps' sub-program.
- Improved and expanded the two component fit in the 'Stars and gas kinematics' task to be more general and with four presets available: old-young components, metal rich-metal poor components, all templates, and two template with fixed age and metallicity mode, following the work of Rubino et all., 2021.
- Modification of the 'kinematic parameters' layout to be more clear and to allow the new options for the two component fitting.
- Improved the 'correct_distortion_slope' of the 'system_span.py' module used by the 'longslit extraction' sub-program to perform a better correction of the slope of 2D FITS containing the long-slit spectrum to extract. Now the flux is rigorously conserved while correcting for the slope.
- Added new interactive functionalities to the Preview: F for the line Fitting mode, E for the equivalent width mode, S for the S/N mode, D for the Differential wavelength and velocity mode, I for the Integrated line flux mode, L for peak finding, P for saving snapshot. C cancels the overlays.
- Gas flux in "Stars and gas kinematics" now is calculated and saved in physical units. pPXF requires spectra to be normalized to unity to avoid instabilities, but the gas flux given has not physical meaning, so now SPAN saves the normalization factor (np.median(galaxy)) and multiplies by this factor the gas flux calculated by pPXF on normalized spectra. These gas flux values are NOT shown in the pPXF output in the terminal (I cannot modify the pPXF output), but are stored in the ASCII files generated in the 'Process all' mode.
- Minor bug fixes

Version 7.0 (20250928):

- Major changes to the layout: The Utilities frame is now a standalone window in the 'Window –> Utilities' menu. This frame has been replaced with a real-time and interactive preview of the selected spectrum in the listbox which allows panning, zooming (left mouse click) and redshift estimation by shifting the spectrum to match the reference lines marked (right-click and drag). Some minor cosmetic adjustments to the layout (e.g. tittle for the listbox and better alignment of the items) have been performed. The main GUI is now resizable to account the zooming options (see below).
- Introduced the 'View' menu where you will find zoom options to better adapt the GUI to different screen resolutions. You can zoom-in, zoom-out or reset the zoom to the default level. The zooming acts on all the GUI windows. This feature is still experimental and some glitch are expected, especially for Windows systems, but they do not have impact on the stability of SPAN.
- Enforced the preliminary check of the loaded spectra with an estimation of a global S/N and flag in the listbox if S/N <=5.
- Interactive listbox: now you can move and delete the spectra by right-click on the selected spectrum. Deletion works also with the 'Canc' keyboard button (use with care, you cannot undo the operation).
- Implemented the Markov chain Monte Carlo (MCMC) simulations for the determination of the stellar parameters from the Lick/IDS analysis. This option is available for SSP models with alpha/Fe parametrization (e.g. Thomas, sMILES, MILES) and represents a more

accurate alternative to the simple linear interpolation with griddata.
- Substituted the 'sigma_measurement' function in the 'spectral_analysis' module with the 'measure_sigma_simple' function, which is used by the 'Velocity dispersion' task. Now it's faster and more accurate.
- Introduced the custom 'bias' keyword setting of pPXF for the 'Stars and gas kinematics' task. Now you can select the bias (regularization) besides the default value of pPXF. To be used only when you fit at least 4 kinematic moments.
- Now in the 'Stellar populations and SFH' task you can decide whether save or not the spectra products generated (best fit template, residuals, emission corrected spectra, if available).
- Added the elliptical binning mode in the "DataCube extraction" sub-program, to radially rebin the datacube using isphotal ellipses profiles (or any kind of custom circle/ellipses).
- Compiled this beautiful manual!

Version 6.6 (20250909):

- Now the 'Plot maps' sub-program stores the parameter values until SPAN is closed.
- Fixed a bug that prevented to save spectra products (i.e. fit residuals, bestfit template) for the 'Stellar populations and SFH' task when no gas lines where included of not found.
- Added new functionalities to the 'Plot maps' sub-program: isophotes overplotting and smoothing. This required also to add the 'save_image' function to the 'cube_extract' module that extract and saves the 2D image by averaging all the wavelengths selected from the datacube. This image is required in the 'Plot maps' sub-program, if isophotal contouring is needed (optional).
- Fixed a bug that prevented the Lick/IDS measurements in the 'Line-strength analysis' in high-redshift spectra, since SPAN was checking the wavelength compatibility of the spectra with the Lick wavelength range not considering the rest-frame corrected wavelengths.
- Replaced the np.chararray function in the code since it has been deprecated with recent Numpy versions (2.X.X).
- Added a luminosity slider to the 'View datacube' option of the 'Datacube extraction' subprogram to facilitate reading datacube slices.
- Added the 'spectra_lists' folder to the 'SPAN_results' path where SPAN saves all the spectra file lists generated with different routines (i.e. 'Generate spectra list containing 1D spectra', 'Long-slit extraction' with the S/N spectral bins option, and 'Datacube extraction' spectral bins.
- Added the option 'Fixing stellar kinematics first' to the 'Stars and gas kinematics' task and the option 'Fitting gas and stellar kinematics' to perform a first fit only to the stellar component and then fix the LOSVD values of the stars to perform a second fit including the gas, as suggested by Cappellari 2017. Added the new function 'build_goodpixels_with_mask' in the 'spec_analysis.py' module in order to use only the 'goodpix' keyword to allow both the emission lines and user mask. The keyword 'mask' is not used anymore for kinematics and sellar populations and SFH since it is not compatible with goodpix, so until now it was not possible to use both the masking of emission lines provided by pPXF and the user mask.
- Added the option to set multiplicative polynomial to 'Stars and gas kinematics' task, which are needed if a reliable estimation of gas flux is needed besides kinematics.
- Added support to JWST NIRSpec datacubes for the cube extraction sub-program
- Corrected minor bugs caused by updating the modules (especially numpy and scipy) required by SPAN

- Fixed an error for macOS systems that prevented to download the required files and also the SSPs from pPXF repository (macOS requires an SSL certificate to download files from internet).
- Updated the versions of the modules compatibles in order to include also the newest versions of NumPy (2.X.X), and solved the compatibility problems with macOS. Now SPAN requires Python >= 3.10 to run but should be more stable and easy to install for Android and macOS systems.
- Changed the default wavelength units used by SPAN. Now is Angstrom everywhere
- Substituted the cross-correlation functions with more reliable ones written from scratch. Now the cross-correlation task is much more solid.
- The 'Stars and gas kinematics' task now caches the loaded templates at the first iteration in the 'Process all' mode, reducing the execution time by about a factor 2, instead loading the templates for any spectrum to process in the list. Working to do so also to the 'Stellar populations and SFH' task but it is much more difficult. Anyway, for this task the large execution times depend on the fit, so caching the templates will reduce the execution times by only 10 to 20%.
- The 'Stars and gas kinematics' now can save also product spectra from the fit (bestfit, bestfit gas, gas with continuum removed, continuum with gas removed).
- Added the calculation of t50 and t80 for the 'Stellar populations and SFH task'
- Added the calculation of uncertainties (1sigma) also for the luminosity and mass fraction and cumulative mass of the 'Stellar populations and SFH' if bootstrap simulations for error estimation is activated.
- Changed the fit parameters when using pPXF to subtract the emission lines for the 'Line-strength analysis' using the semi-automatic 'Lick/IDS' indices. The fit with pPXF now is performed by calculating first the real noise value of the spectrum (auto_noise = True).
- The 'Stars and gas kinematics' now saves also the flux and formal errors of the emission lines, if gas calculation is activated.
- Refactored the 'resolution' function of the utilities.py module to correct a bug that sometimes showed the resolution as negative value. Now also an error estimation of the FWHM is calculated and showed.

Version 6.5 (20250531):

- Added the Plot maps sub-program for reading and displaying the 2D maps from datacube data
- Introduced new options in the Spectra Manipulation panel for saving the intermediate spectra, to save only the final spectra, and to NOT save processed spectra
- Changed the name of the folders to store the results. Now they have the same name of the tasks to improve clarity.
- Added the WEAVE_LIFU.py routine for datacube extraction.
- Added the option to use pre-existing mask and bin info files for datacube extraction.
- Added the convolution with MUSE LSF for the 'Stars and gas' kinematics task
- Added selecting the resolution in R or MUSE LSF also for the custom template option in the 'Stars and gas kinematics' task.
- Added the custom masking for the 'Stars and gas kinematics' task.
- Added the use of a generic template set in 'Stars and gas kinematics'. For now, only fits files with linear sampling in A can be used.
- Added the 'Degrade MUSE data' option in the 'Degrade resolution' of the 'Spectra manipulation' panel. This allows to degrade/convert to a fixed FWHM resolution the variable

resolution of MUSE.
- Refactored the read_datacube function in the system_span.py module.

Version 6.4 (20250220):
Complete refactoring of the code, split in different modules. Use of the DataClass to store the GUI parameters and use.

Version 6.3 (20250205):

- The spectral manipulation tasks are now grouped in a secondary panel called Spectra manipulation.
- Removed the GIST pipeline for copyright issues.
- Added the two component stellar fitting for the stars and kinematics task.
- Major modifications to the 'DataCube extraction' sub-program. Added the manual binning mode by drawing custom regions on the datacube.

Version 6.2 (20250115):

- Added the option to insert a custom template set in pPXF .npz format for the 'Stellar populations and SFH' task.
- Major changes to the 'cube_extract' module to improve efficiency.

Version 6.1 (20250107):

- Major change to the spectra manipulation frame: re-arranged the spectra math tasks and introduced the possobility for the user to re-order the spectra manipulation tasks.
- Added the dust parametrization for the 'Stars and gas kinematics' tasks.

Version 6.0 (20241118):

Major change to the layout and the structure of the main span code. Now the code recognizes the operating system used and adapts the layout. No need to compile different files for different operating system anymore.

Version 5.8 (20241114):

- Modified the 'Stars and gas kinematics' and the 'Stellar populations and SFH' tasks to deal with cosmological redshift galaxy spectra, fixed some minor bugs, cleaned the code and rearranged the plots generated for better clarity.
- Modified the Stellar populations and SFH task to deal either with log10 ages and linear ages.
- Changed from ASCII files to FITS 1d format the residual, emission corrected and best template spectra generated by the stellar populations and SFH task.
- Fixed a bug in the Line-strength analysis and the Lick/IDS option that caused wrong measurements when the emission correction was disabled and doppler correction enabled.
- Fixed the plots generated by the Line-strength analysis task in the 'Process all' mode.
- Added the selection of custom EMILES SSP templates also for kinematics
- Now it is possible to insert any sMILES template in the spectralTemplates/sMILES_afeh folder to work with alpha/Fe in the 'Stellar populations and SFH' task.

Version 5.7 (20241029):

- Major changes to 'Cube extraction' to be more flexible and more user friendly. Introduced the datacube dynamic visualisation in a matplotlib window and the 'Generate mask' button that allows the user to graphically generate a spectral mask.

- Major changes to the “Stellar populations and SFH” task to include: sMILES models with alpha/Fe, dust absorption (replacing the obsolete reddening keyword of pPXF), uncertainties estimation with bootstrap simulations, auto noise estimation for the fitting so that Chi2 = 1 for non regularized fits and masking user defined spectral regions. Improved plots, now plotting also the alpha/Fe-[M/H] weights. Saving the luminosity and mass weights.
- Stellar and gas kinematics: fixed the Xshooter library that could not be used with spetra with R = constant resolution.
- Changed the Dopcor task, now called ‘Doppler/z correction’: now the user can correct the spectra both with radial velocity values and redshift (z).
- Adjusted the layout of the ‘Cross-corr parameters’ and ‘Sigma parameters’ windows. Added and improved the cross-correlation task: now it can consider also the cosmological redshift instead only velocity values.
- Added the help files for the ‘Cube extraction’
- The user defined parameters are saved now also for the sub-programs.
- Corrected minor bugs in the system_span read_spec function and added the read_datacube function.
- Corrected minor bugs in the spec_analysis.py module

Version 5.6 (20240824):
Modified the “Stars and kinematics” task to include more parameters.

Version 5.5 (20240724):
Integrated the GPR Machine learning for Lick/IDS analysis. Corrected minor bugs.

Version 5.4 (20240527):

- Integrated a modified, updated and running version of the GIST pipeline as independent sub-program fo facilitate users in the migration process from GIST to SPAN (NOTE OF THE EDITOR: well, this was not a good idea since I’ve never got the authorization to publish this modified version of GIST, so I had to remove from further SPAN distributions. Goodbye GIST).
- Added the Xshooter SSP library (XSL) of Verro et al. 2022 to ‘Stars and gas kinematics’ and ‘Stellar populations and SFH’ tasks.
- Added the custom (E)MILES template selection for ‘Stellar populations and SFH’ task, the possibility to mask the emission lines and an algorithm to find automatically the best ppxf parameters for stellar populations: noise and Regul. err. Changed the format of the fits file saved in the ‘Convert spectrum to’ task. Corrected minor bugs.

Version 5.3 (20240515):

- Integrated a modified version of the GIST pipeline in order to bin and extract data cube spectra.
- Rearranged the Open spectra frame. Changeg the names of some spectral analysis tasks.

Version 5.2 (20240505):
Embedded Freesimplegui, improved ppxf stellar populations with more parameters and better plots.

Version 5.1 (20240502):
Changed Pysimplegui with Freesimplegui since pysimplegui is not free anymore. Added pixel noise to ppxf stellar populations layout and changed the file format of saved plots, from eps to

high resolution png.

Version 5.0 (20240221):

- Rearranged the spectra pre-processing framework by incorporating crop and wavelet cleaning tasks.
- Added parameters buttons to sigma clip cleaning (now dynamic cleaning), dopcor, and heliocor.
- Fixed a bug in the "Compare with" task.
- Introduced the continuum modeling task, allowing for a more refined way to model and subtract (or divide) the continuum of the spectra.

Version 4.7 (20240201):

- Since PySimpleGUI is not free anymore, I switched to FreeSimpleGUI. Moreover FreeSimpleGUI license permits redistribution, so I embedded it in the distribution. Thank you FreeSimpleGUI guys for saving SPAN!
- Adapted the code to the new release of ppxf. Ppxf is no longer included in the code package and must be installed via pip.
- Provided the option to choose between three stellar libraries for the ppxf tasks (kinematics and stellar populations).
- Added the capability to save all parameters and load them.
- Implemented some minor changes to the layout.

Version 4.6 (20240114):

- Included the emission correction and interpolation of the Lick/IDS indices in the EW measurement task. This provides information about the age, metallicity, and alpha-enhancement of the stellar populations of galaxies using the models of Thomas 2010.
- Implemented Monte Carlo simulations to estimate uncertainties on stellar parameters via Lick/IDS index interpolation.

Version 4.5 (20231221):
Added the Lick/IDS index measurements and sigma correction in the EW task.

Version 4.4 (20230215):
Introduced the 2D spectra extraction standalone panel (still experimental), enabling the correction and extraction of long-slit 1D spectra from a 2D fits wavelength-calibrated images.

Version 4.3 (20230207):

- Added the plot window to display the results generated by SPAN.
- Implemented the "match rows" feature in the text editor to match and merge rows of two ASCII files, space-separated, with one column in common.

Version 4.2 (20231203):
Introduced the fits header editor button with three possible operations on the fits files: Modify the keyword of a single file. Modify the keywords of a list of files. Extract a keyword from a list of fits files.

Version 4.1 (20231202):
Added the blackbody fitting task.

Version 4.0 (20231123):

- Adjusted the layout of the spectral analysis.
- Added the terminal output.
- Fixed various bugs.
- Rectified the progress bar.
- Introduced the text editor and the menu bar.

Version 3.5 (20221120):
Added the ppxf algorithm for kinematics and stellar populations.

Version 3.4 (20221114):

- Included the Line(s) fitting task.
- Corrected minor bugs.

Version 3.3 (20221109):

- Modified the Cross-correlation calculation.
- Fixed a bug that caused the program to crash if two spectral analysis tasks were selected.
- Implemented checks and existing conditions on the spectral analysis files and values.

Version 3.2 (20221106):
Added tooltips to the tasks/buttons for better clarification of their functions.

Version 3.1 (20221105):

- Rearranged the GUI.
- Fixed a bug in the sigma coefficient calculation task.

Version 3.0 (20221104):
Implemented key values instead of increasing numbers in the GUI. This allows adding and modifying entries without changing all the value numbers.

Version 2.95 (20220128):
Added the calculation of the equivalent width (EW) and uncertainties also in magnitude.

Version 2.9 (20220119):
Incorporated SNR determination in the EW task. Now, when measuring the EW, the program also provides the SNR calculated in the pseudo-continuum bands.

Version 2.7 (20211124):
Adjusted the Sigma coeff and EW correct tasks to be truly standalone, with a proper cycle and progress bar.

Version 2.6 (20211123):
Adjusted the position of the preview result button to fit the low resolution of my screen after the upgrade of PySimpleGUI, which enlarged my window automatically and without reason.

Version 2.5 (20211123):
Added a progress meter for the “Process All” event.

Version 2.4 (20211122):
Now you can finally process just one spectrum!

Version 2.3 (20211122):
Added the degrade function in lambda (FWHM) to degrade the spectrum to a certain FWHM in Angstrom.

Version 2.2 (20210927):
Adjusted the utility menu and added the possibility to save the SNR of the spectra to a file.

Version 2.1 (20210916):
Rearranged the utility menu and introduced the conversion of flux from Jansky to MKS and MKS f_lambda.

Version 2.0 (20210630):

- Added the 'compare spectra' utility.
- Improved the sigma clipping, now called the clean task.
- Rearranged the tasks in the pre-processing and processing frames.
- Enhanced the plot for velocity dispersion measurements.
- Implemented a check on the existence of the loaded spectra.

Version 1.9 (20210626):
Some cosmetic changes to the graphical layout.

Version 1.8 (20210626):

- Added the save plot option in the spectral analysis frame.
- Improved plots of the sigma coeff determination task.

Version 1.7 (20210615):
Changed the file names of the result files to include a timestamp, preventing overwriting of older files. Changed the name of the probram.

Version 1.6 (20200711):
Added the task to measure the resolution of the selected spectra in the Utility tab. Some major changes to layout.

Version 1.0 (20200615):
Improved the read_spec function for the 1dFit spec, excluding PyAstronomy.pyasl since it is not working properly. Now the program is called SPAN and not SPEIM anymore. Welcome SPAN!

Please, report any bug or comment to daniele.gasparri@gmail.com Have fun!

Daniele Gasparri
2025-09-20
Greetings from the Atacama desert!