Astrocook: your starred chef for spectral analysis

Guido Cupani[a,b], Valentina D'Odorico[a,b], Stefano Cristiani[a,b], Stefano A. Russo[a], Giorgio Calderone[a], and Giuliano Taffoni[a,b]

[a]INAF–Osservatorio Astronomico di Trieste, Via G.B. Tiepolo, 11, I-34143 Trieste, Italy
[b]IFPU–Institute for Fundamental Physics of the Universe, via Beirut 2, I-34151 Trieste, Italy

ABSTRACT

Astrocook is a software environment to analyze quasar spectra in a variety of ways. It aims to break the static pipeline paradigm by enforcing a new flexible approach to data treatment, in which complex automatic workflows are dynamically created from a wide set of atomic operations (hence the tagline: "a thousand recipes to cook a spectrum"). We will focus both on the novel algorithms that have been implemented and on the scientific validation and reproducibility of the results. To highlight the benefits of the Astrocook approach for both interactive and automatic analysis, two specific use cases are discussed (one of which was used in practice to process observational data from the QUBRICS survey).

Keywords: Astrocook, Data Analysis, Pipelines, Algorithms, Automation, Reproducibility, Python package

1. INTRODUCTION

The spectra of distant quasars, as observed in the visual and near-infrared bands, contain a wealth of information of both astrophysical and cosmological relevance, literally "shedding light" not only on the emitting sources themselves but also on the structures situated along the line of sight. A large number of outstanding science cases (including, but not limited to: understanding the nature of dark matter, observing a possible variation of the fundamental constants, testing the predictions of General Relativity and ΛCDM cosmology, constraining the evolution of galaxies and their interplay with the intergalactic medium, etc.) critically depend on a proper extraction of this information from the data. As spectroscopy evolves into a precision science, with new ultra-stable instrumentation now operating at existing facilities (e.g. ESPRESSO at the ESO VLT[1]) and being conceived for future facilities (e.g. the High-Resolution spectrograph for the ESO ELT[2]), the development of software tools to analyze the quasar spectra faces significant challenges:

- It must provide procedures to model spectral features with high accuracy, to meet the demands of the most delicate science cases;
- It must be flexible enough to accommodate changing requirements and, if needed, incorporate specific procedures from external sources;
- It must implement as much as possible automatic procedures to ease the analysis of large sets of data;
- It must ensure that the analysis process is properly logged throughout, in a way that makes it fully reproducible by other research groups.

Astrocook is a software environment to analyze quasar spectra, specifically conceived to address these challenges. It is currently focused on modeling the absorption features observed in the spectra after disentangling them from the emission component, but its scope is not limited to this single task by design. A first incarnation of the software as a Python 2 package has been presented in Ref. 3 with the tagline "a thousand recipes to cook a spectrum", to emphasize the philosophy underlying the project: provide the science users ("cooks") with the right tools (a "kitchen table" and a book of "recipes") to get information from the data ("prepare" and "taste" their "dishes"). Ref. 3 documented the version 0.3.0 of the code, which was already provided with a Graphical

Send correspondence to guido.cupani@inaf.it.

User Interface (GUI) and several analysis tools, but was still lacking an integrated way to design and run complex workflows and to ensure they could be effectively reproduced. We hereby present the version 1.0.0 of Astrocook, which overcomes these limitations and supersedes any previous implementation of the package.

The new Astrocook is the product of an extended development and testing phase, which, while retaining the core philosophy of the project, resulted in a major overhaul of its functionality and look-and-feel:

- the code has been ported to Python 3,[4] with a complete refactoring of the existing classes and methods;
- the GUI has been redesigned, abandoning the one-window model to increase its versatility;
- an integrated logging and scripting system has been implemented, allowing for a complete reproducibility of the analysis procedures;
- the pool of available recipes has been significantly expanded, to include new practical solutions in handling the data through the GUI and more robust algorithms to extract their information content.

The refinement of the software led to its actual application to scientific purposes. One of these applications, the treatment of visual and near-infrared quasar spectra acquired within the QUasars as BRIght beacons for Cosmology in the Southern Hemisphere (QUBRICS) survey[5,6] is first presented in this article.

The article is organized as follows: Sect. 2 describes the architecture of the code (data structures, recipes, algorithm), also providing an overview of the GUI (Sect. 2.3) and of the scripting utility (Sect. 2.4); Sect. 3, on the other hand, describes two practical applications of the package, namely the interactive analysis of a quasar spectrum (3.1) and the automatic workflow designed to treat the QUBRICS data (3.2). Conclusions and future prospects are outlined in Sect. 4. Throughout the text, we use *italic* for technical terms, `monotype` for code and file names, and sans-serif for interface elements. We denote atomic transitions by their ionization state (I for neutral species, II for singly ionized species, etc.) and, when they first occur in the text, by their rest-frame wavelength in nm preceded by λ.

2. ARCHITECTURE

2.1 Core structures

Analysis jobs are called *sessions* in Astrocook. A session is an abstraction that includes, at any given moment, both the data structures and the procedures that have been executed on them. This information is continuously updated during the analysis, and can be saved and retrieved at a later time.

The main data structures in a session are the following (in parentheses, we give their names as they appear in the code, as attributes of the `Session` class):

Spectrum (`spec`): a table that pairs a flux-like quantity to a wavelength-like quantity. We use the expressions "flux-like" and "wavelength-like" to be as general as possible: from the point of view of structure, it doesn't matter if the spectrum is expressed in wavelengths or frequencies, or if the flux density is calibrated in physical units or not. From now on we will use "flux" and "wavelength" for simplicity, without loss of generality.

List of lines (`lines`): a table with selected entries from the spectrum table, corresponding to the locations where line features were detected. Currently, only absorption features are included in the line list.*

*The `lines` table is liable to change nature throughout the analysis (including, at different stages, both real absorption features and features resulting from modeling and removal of absorption complexes); it should not be regarded as a list of confirmed absorbers, which is given by the table `systs` instad.

List of systems (`systs`): a table with associations between lines, based on their identification to specific ionic transitions. By our convention, a system has *one and only one* redshift value (all its lines/transitions have the same redshift by definition)[†]. Several systems are typically needed to model complex spectral features, either in emission or absorption. Currently, modeling of features is limited to absorption complexes (an example of absorption complex modeled with seven distinct systems is shown in the right plot of 5, panel C).

These structures can be inspected and modified using the Astrocook GUI (see Sect. 2.3). Additional structures (like the list of nodes used to model the emission continuum, `nodes`, and the list of models for the absorption systems, `mods`) are handled internally and can be modified only by the analysis procedures (see Sect. 2.2).

The spectrum is the only required structure in a session. The simplest session starts as a bare `spec` table that Astrocook constructs from a FITS[‡] file provided by the user. The list of supported FITS formats is continuously updated; in general, Astrocook is able to natively recognize FITS files containing either three image extensions (for wavelength, flux density, and flux density error, respectively) or one table extensions with wavelength, flux density, and optionally flux density error columns.[§]

When a session is saved, the data structures are saved both in FITS and ASCII format. The FITS format is more complete as it keeps track of all the relevant metadata in the file headers. Upon saving, the user chooses a session name which is inherited by the individual files, with an additional suffix specifying the structure contained in the file (`_spec`, `_lines`, `_systs`, `_nodes`) and extensions `.fits` and `.dat` for FITS and ASCII, respectively. All the files are bundled in a gzip-compressed tarball with special extension `.acs` (for "Astrocook session").

2.2 Processing tools

Data are analyzed by procedures or *recipes*. Recipes are designed to be *unitary* and *complete*, i.e. to address a single, well-defined question (e.g. "where are the absorption lines in this spectrum?") and provide a definite answer (e.g. "the lines are at these wavelengths and have these flux densities at their peaks"). They are allowed to modify the data structures they uses, and to create new data structures when they need to (e.g. a `lines` table to contain the results of line detection). In some cases, recipes can even start new sessions (e.g. when a region is extracted from a larger spectrum) or exchange information between sessions (e.g. when a `lines` table is copied from a different spectrum of the same target), but in general the scope of recipes is limited within a given session.

Internally, most recipes are arranged in *cookbooks* for easier reference, based on their range of applications. Five cookbooks are currently available in Astrocook (in parentheses, we give their class names as they appear in the code):

General cookbook (`CookbookGeneral`). A collection of common recipes that apply to the spectrum as a whole: convolution (`convolve`), rebinning (`rebin`), conversion to and from the rest-frame (`shift_to_rf`, `shift_from_rf`), extraction of a spectral region (`region_extract`), estimation of the signal-to-noise ratio (`snr_est`), etc.).

Cookbook for continuum (`CookbookContinuum`): recipes to disentangle the absorption and emission components in the spectrum and estimate the unabsorbed emission level (see Sect. 2.2.1, Continuum estimation.).

Cookbook for absorbers (`CookbookAbsorbers`): recipes to identify absorption systems and model them as complexes with composite Voigt profiles (see Sect. 2.2.1, Absorption system modeling.).

Template cookbook (`CookbookTemplate`): recipes to add a template to a spectrum (as a column of the `spec` table) and adjust it. Currently, a power-law and a black-body template are available.

[†]According to a different convention, absorption complexes as a whole can be referred to as "systems", while line associations defined by a single redshift value are referred to as "components of the system".

[‡]https://fits.gsfc.nasa.gov

[§]Accepted names for FITS table columns are `x`, `wave`, `WAVE`, and `col1` for wavelength, `y`, `flux`, `FLUX`, and `col2` for flux density, and `dy`, `err`, `ERR`, `fluxerr`, `FLUXERR`, and `col3` for flux density error.

Graphical cookbook (`CookbookGraph`): recipes to control the GUI and the graphs (needed to include graphical actions, e.g. menu selections and mouse clicks, in repeatable scripts).

Additional recipes to manipulate data structures within and across sessions are provided as attributes of the classes `Session` and `GUISessionPanel` (see Sect. 2.3).

The organization of the cookbooks is not to be taken as fixed; on the contrary, it is designed to facilitate the continuous addition of new recipes and cookbooks as well, while enforcing a modular architecture to assure backward compatibility and reduce the effort of maintaining the code.

Recipes are designed to be used as the building blocks to create *workflows*, i.e. complex procedures to extract information from the data in a controlled and reproducible way. Only one workflow (namely, the estimation of the emission continuum and the modeling of the Lyman-α forest) is currently provided as a stand-alone (macro)-recipe. As a general rule, maximum freedom is left to the users in designing *their own* workflows by combining recipes in the desired order, with logical consistence as the only limitation. Workflows can be run interactively through the GUI, by calling the recipes one at a time (see Sect. 2.3), or automatically via scripting (see Sect. 2.4). Astrocook is provided with a native logging system that keeps track of all the recipes executed in a given analysis session (in compliance with the idea that a session is defined by both data and procedures, see Sect. 2.1). The log is saved in JSON format whenever a session is saved (with the session name and extension `.json`).

2.2.1 Algorithms

By design, the goal of Astrocook is not to issue full-fledged algorithms for data analysis as "black boxes", but to provide the astrophysical community with suitable *tools* to design and implement new algorithms (in agreement with the common perception that users should rather be "cooks" than "diners"). Nevertheless, Astrocook does in fact contain new solutions to common problems, mostly derived from a refactoring of the algorithms originally conceived for the Data Analysis Software (DAS)[7,8] of the spectrograph ESPRESSO at the ESO VLT.[1,9] Compared to the ESPRESSO DAS, Astrocook recipes are designed to be more granular and reusable outside of their original scope.

Here we give a brief review of the most important algorithms for quasar analysis implemented in the DAS, and how they are replicated by Astrocook recipes. The algorithms make extensive use of the NumPy[10] and SciPy[11] libraries.

Coaddition. Coaddition is the combination of two or more exposures of the same object into a single spectrum. It can be either limited to a *combination* of the data points (i.e. pixels) from the original exposures, each one considered with its own wavelength span, or extended to include a *rebinning* of the combined spectrum into a wavelength grid of choice. Rebinning reduces the size of data and the computational time required by further processing, but it also introduces correlation among adjacent bins (whenever it is not possible to ensure a perfect alignment between the wavelength grids of the input exposures and the wavelength grid of the output spectrum). The best trade-off (which is enforced in the ESPRESSO DAS) is achieved by rebinning the spectrum only once, after collecting and weighting the contributions from all exposures, instead of rebinning the exposures separately and then combining them.[12]

Astrocook provides separate recipes for combining different sessions (`GUIPanelSession.combine`) and rebinning the resulting spectrum (`CookbookGeneral.rebin`), to be preferably executed in this order. The user can control the range and the step used for rebinning; the step can be espressed both on wavelength or log-wavelength units. An additional equalization recipe (`GUIPanelSession.equalize`) is available, to adjust the flux density in the input spectra prior to combination. This is done by selecting a fiducial wavelength region and computing the rescaling factors from the ratios of median flux densities in that region.

Continuum estimation. The estimation of the intrinsic emission spectrum of a quasar (i.e. the spectrum of the source alone, before it has been processed by the intervening absorbers, either associated to the active galactic nucleus or distributed along the line of sight) is a notoriously difficult task, which is assessed in many different ways. Absorption from neutral hydrogen in the intergalactic medium is particularly tricky

to model, as it gives rise to a "forest" of lines bluewards from the Ly α emission, which gets thicker and thicker as the redshift increases (due to the higher fraction of H I) until it degenerates into large saturated complexes (Gunn-Peterson troughs) at the end of the reionization epoch. At such high redshift, all solutions to the problem of continuum estimation are essentially model-based (see e.g. Ref. 13); at low redshift, on the contrary, is generally easier to pick out and mask individual absorption lines, and interpolate the emission continuum across them, as it is the region redwards from the Ly α emission (which is only affected by metal-line absorption). This latter solution is the one adopted by the ESPRESSO DAS, which includes a more sophisticated approach to treat the Ly α forest, based on an iterative fit of the absorption complexes.[12]

Astrocook implements recipes to find lines (`CookbookContinuum.lines_find`) and to interpolate continuum from nodes distributed across the non-masked regions (`CookbookContinuum.nodes_cont`). Lines are detected as prominent local minima in the flux density spectrum, previously smoothed with a gaussian kernel to reduce noise. Different values of the gaussian variance are used in decreasing order to catch lines of different width; the user can control the range of values (expressed in velocity units) and the threshold that defines the prominence of the mimima (as a multiple of the local flux density error).[¶] The regions between the local maxima adjacent to each line are then masked. Nodes are placed at fixed log-wavelength steps across non-masked regions, taking the median flux density of the neighbouring region, and σ-clipped to reject outliers; the user can control both the size of the steps (expressed in velocity units) and the multiple of σ to be used for clipping. Finally, nodes are interpolated with a univariate spline of chosen degree.

Absorption system modeling. Absorption systems are first detected by cross-matching the `lines` table with a list of known transitions and finding coincidences among the possible redshift values resulting from the cross-matching (see e.g. Ref. 14). Detected systems are then modeled with Voigt profiles, depending on the system redshift z, its column density N, and its Doppler broadening b (which may be thermal or turbulent), and convolved with the instrumental PSF. Composite Voigt profiles of several partly-overlapping systems are generally required to model complex absorption patterns; in this case, the best-fitting model can be determined by least-square minimization of the reduced χ^2 between the observed flux density profile (normalized to the continuum) and the theoretical profile obtained from a reasonable guess of the parameters (see e.g. FITLYMAN[15] and VPFIT[16]). The ESPRESSO DAS follows this approach, with the added benefit of a graphical user interface to set up the guess parameters, their range, and possible constraints on their values.[12]

Astrocook further improves on the ESPRESSO DAS functionality by providing a wide range of recipes to handle different use cases. The simplest recipe is `CookbookAbsorbers.syst_new`, which add a new entry to the `systs` table and fits it to the spectrum starting from input values for z, N, and b. A more sophisticated recipe is `CookbookAbsorbers.syst_new_from_lines`, which cross-matches the `lines` table with a series of transitions and automatically adds new systems wherever it spots a redshift coincidence among two or more lines. The user can select the series of transitions by their tag, as they appear in the `atom_par.dat` table included in the package (e.g. `CIV_1548` for C IV λ154.8; `CIV` alone selects all transitions starting with `CIV_`, namely the doublet C IV $\lambda\lambda$154.8, 155.0). Comma-separated transitions are fitted as a single system (i.e. with a single set of Voigt parameters), while semicolon-separated transitions are fitted independently on the same run. Further recipes are available to add new systems to already modeled complexes (`systs_improve`) and to refine the fit by finding new transitions associated to the already detected systems (`systs_complete`).

χ^2 minimization is implemented using the Lmfit library.[17] Lmfit models are handled internally during the analysis sessions; only the best-fit Voigt parameters are saved when the session is saved, and the models are recomputed when a saved session is loaded. Astrocook automatically merge models when they happen to overlap, this occasionally requires a re-fit when a new model is added merging two systems that were

[¶]It is also possible to find the minima in a column of `spec` different from `y`, i.e. the original flux density. A reasonable choice, when a model of the absorption systems is available, is the column `deabs`, i.e. the flux density reconstructed after removing the absorption systems. This allows to improve both the system model and the continuum example by detecting as new "lines" the residuals of the fit; see Sect. 3.1.

previously separated, or when the fitting procedure extends the range of a model to touch neighbouring models. In this case, the recipes let the user control the number of optional re-fit cycles.

The set up of guess parameters and their constraints is controlled through the GUI and is discussed in Sect. 2.3.

2.3 Graphical User Interface

The GUI has been one of Astrocook's assets since its inception.[3] When the code was ported from Python 2 to Python 3, the GUI was completely reshaped to be more flexible in displaying the data (allowing for a finer tuning of the elements of the graphs) and to give a more direct approach to the analysis procedures (e.g. using mouse clicks to select regions on the graph and call functions). The GUI is based on wxPython[‖], which provides a native look-and-feel of the graphical elements across many platforms (Linux, macOS, Windows), and uses Matplotlib[18] for plotting. It is launched with the command

```
$ python ac_gui.py [path]
```

where `[path]` is an optional path to a FITS file or a saved `.acs` archive (see Sect. 2.1).

The GUI includes three main windows (Fig. 1; in parentheses, we give the window name and its class name as it appears in the code):

Session window (Sessions; class name: `GUIPanelSession`): always visible, it shows a (possibly empty) list of working sessions. When `ac_gui.py` is launched without a valid path, this is the only window to open, allowing the user to manually load sessions through the menu. Each loaded session is listed with a recap of its main characteristics (spectral range, number of lines, number of systems). Whenever a session is loaded, it is appended to this list with an increasing tag.

Main graph (Spectrum): visible whenever a session is loaded, it shows a graph of the spectrum (flux and error v. wavelength), possibly with some features highlighted (lines, continuum nodes, emission and absorption models, etc.). This window is automatically shown when `ac_gui.py` is launched with a valid path. The main graph is equipped with a drop-down menu to extract regions, compute spectral statistics, and retouch the plotted data (e.g. by zapping out flux spikes, or adjusting continuum nodes).

Detail graph (Spectrum detail): visible whenever a system is selected, it shows a detail of the spectrum in the rest frame of the system itself (normalized flux and error v. relative velocity shift), revealing the system in the context of the absorption complex it belongs to. If the system is a doublet or a multiplet, the graph is split into as many panels as the number of transitions that were fitted together.

Ancillary windows are (Fig. 2):

Table windows (Spectrum table, Line table, System table): three grid-like windows, one for each of the three main data structures (see Sect. 2.1), to display the content of the tables. They are equipped with a drop-down menu that appears upon right-clicking on a given table row, allowing the user to selectively remove rows. The System table window, in particular, is equipped with additional features to control the guess parameters of the Voigt profile fit (see Sect. 2.2.1):

- By clicking on Fit or Improve from the drop-down menu, the user can (re-)fit a system or improve the modeling of the absorber by adding further systems.
- By clicking on a cell from columns z, logN, or b, the user can modify the guess value of the Voigt parameters z, $\log N$, and b, to be used in further re-fitting procedures.
- By right-clicking after selecting one or more cells from columns z, logN, and b, the user can freeze and un-freeze their values (when a value is frozen, is not changed during the fitting procedure).

[‖] https://wxpython.org.

- By right-clicking after selecting two or more cells from columns z, logN, and b, the user can link and un-link their values (when two or more values are linked, they are treated as a single parameter during χ^2 minimization).

System controls window: a dialog window that appears whenever a system is selected. Using this dialog, the user can display an interactive cursor on both graphs, showing the position of a mock system at a given redshift. The Series box defines the transitions to be displayed, the Redshift box defines the position of the system, and the Half window defines the extension of the displayed region in velocity space ($\pm\Delta v$ with respect to the redshift of the systems, with Δv the value in the box). Sliding the cursor across the spectrum, the user can detect and fit new systems, improving the modeling of absorption complexes.

Metadata window: a text panel, it lists the editable keywords from the FITS header of the selected session spectrum, allowing the user to edit them.

Graph elements window: also a text panel, it lists the elements shown on the main graph, allowing the user to toggle the visibility of the elements, add new elements and adjust their appearance.

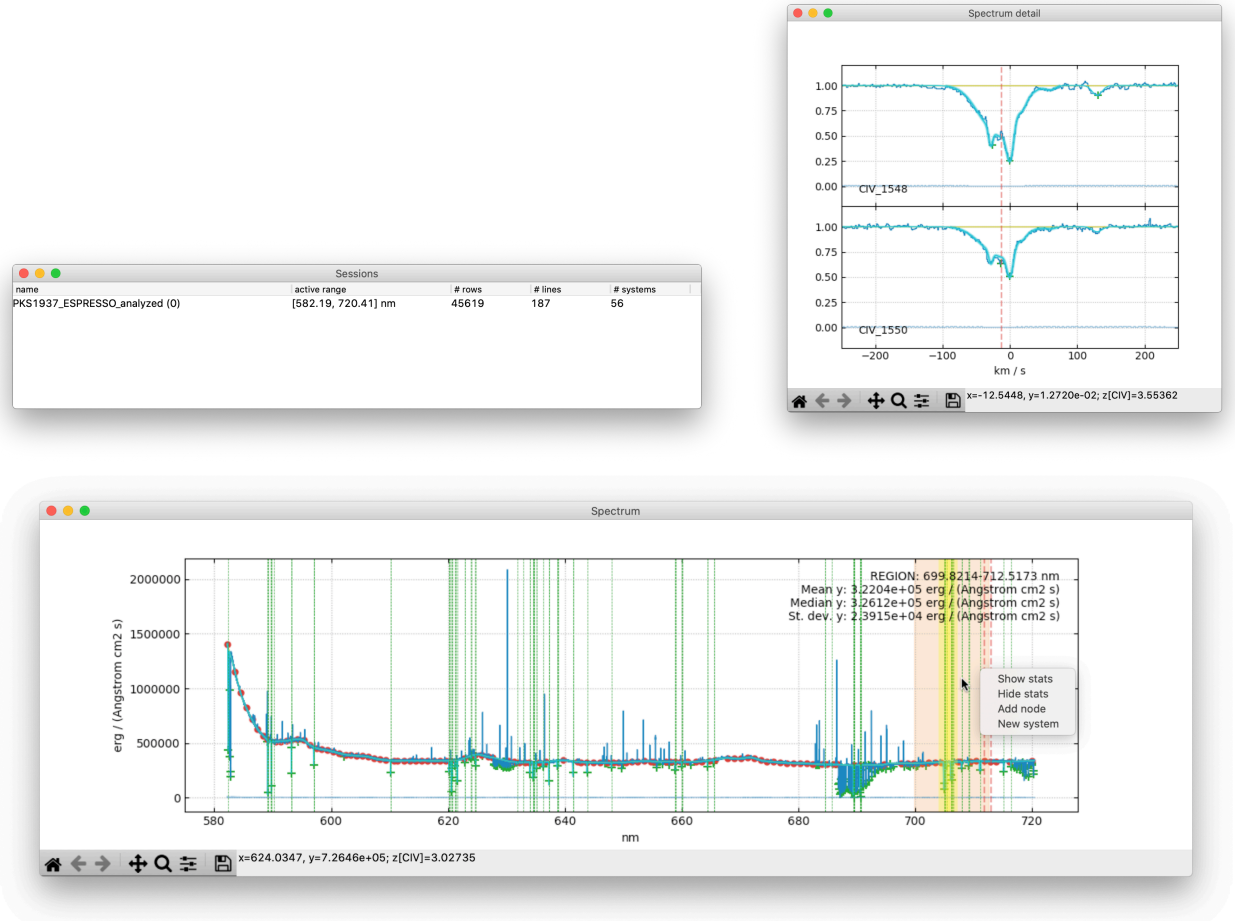

Figure 1. Main Astrocook windows. Counterclockwise from top left: session window, main graph window, detail graph window.

The windows are continuously updated to reflect changes in the displayed information or actions performed by the user. For instance:

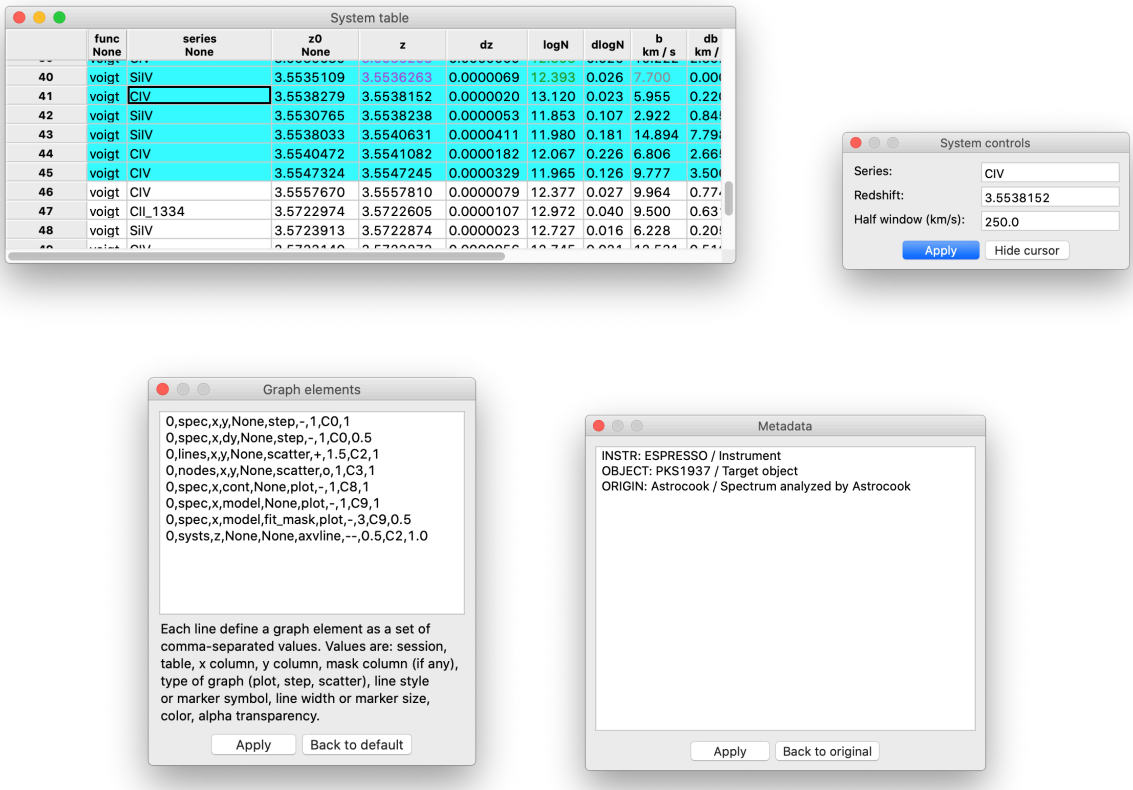

Figure 2. Additional Astrocook windows. Counterclockwise from top left: system list window, graph elements window, metadata window, system control window.

- When a session is selected, the main graph is refreshed to show the current spectrum; the detail graph and the table windows are also refreshed if they are visible.
- When a system is selected in the system list window, the detail graph is refreshed to focus on that particular system, and the main graph highlights the system with a yellow bar.
- When a new system is added to the model of an absorption complex, the graphs are refreshed and the new system is added to the system list window.

Only the sliding cursor is not synchronized between the main graph and the detail graph, to allow for more freedom in switching between the two.

The GUI is the primary way to access and run the recipes for the analysis provided by Astrocook. From the GUI point of view, recipes are organized into three groups:

- Recipes to manipulate sessions, combining and sharing data structures among them: these are grouped in the Edit menu.
- Recipes to manage the graph windows and the additional windows to interact with the data: these are grouped in the View menu;
- General recipes to "cook your spectrum", i.e. extract physical information from the data: these are spread between the Recipes menu (atomic operations and simple workflows) and the Set menus menu (complex workflows for specific tasks).

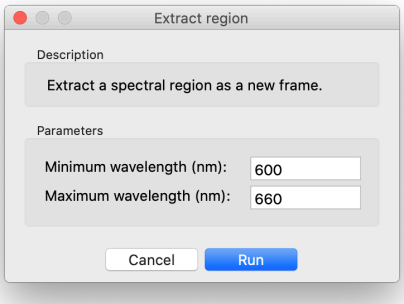

Figure 3. Example of Astrocook recipe dialog window.

In general, not all recipes are available at all stages of the analysis. Recipes that require some piece of information that has not yet been extracted from the data (e.g. a list of systems, for the recipe that fits a model to systems) appear greyed-out in the menus, and cannot be launched. Astrocook automatically recognizes when the relevant information is extracted and makes available more and more recipes as the analysis progresses.

Most recipes require input from the user, in the form of parameters that control the recipe execution. Recipe parameters are handled graphically by dedicated dialog windows (see Fig. 3 for an example) which also provide a brief description of the tasks performed by the recipe. The dialog disappears once the execution is completed. Information from the recipe execution (including warnings and error messages) is printed on the standard output.

2.4 Scripting

As already remarked in Sect. 2.2, recipe execution is logged in JSON format. An example of log file is given in Fig. 4: the JSON format allows for an easy breakdown of the workflow, which in this case consists in starting a session (by loading a FITS file), calling three recipes from the standard cookbooks with specific parameters, and refreshing the GUI afterwards. More specifically, when constructing this example (which will be further expanded in Sect. 3.1) recipe `region_extract` was called after selecting a spectral range directly on the plot (as it appears from the values of `xmin` and `xmax`) while recipes `lines_find` and `nodes_cont` where called from the Recipes menu. This makes no difference in a logging perspective, as all actions are equally recorded. The whole session can be replicated exactly as it was firstly run by launching

```
$ python ac_gui.py [path and name of the JSON file]
```

provided that the input file `[path]/J0234-1806.fits` exists on the system.

JSON files like the one shown in Fig. 4 can be naturally used as blueprints for different, possibly more complicated workflows. This ensures not only that the analysis sessions are fully reproducible, but that they can be easily scaled up to large datasets by manual or automated scripting in JSON format. The envisioned approach is the following:

1. First, a few selected spectra are analysed interactively, to find out the best workflow by freely trying out several recipes in various arrangements;
2. Then, the JSON file produced at step 1 is inspected to eliminate redundancies and generalize the workflow to all spectra (using also shell scripting to automatic generate new JSON files with proper parameters from the blueprints);
3. Finally, a fully automatic analysis on all spectra is launched using the JSON file generated at step 2.

In the following sections, we will provide concrete examples of data analysis developed according to this approach, gradually moving from full interactivity (steps 1 and 2 above; see Sect. 3.1) to full automation (Sect. 3.2).

```
{"set_menu":
  [
    {
      "cookbook": "_panel_sess",
      "recipe": "_on_open",
      "params": {
         "path": ".../J0234-1806.fits"
      }
    },
    {
      "cookbook": "cb",
      "recipe": "region_extract",
      "params": {
        "xmin": 646.3324786128208,
        "xmax": 921.9498131374257
      }
    },
    {
      "cookbook": "cb",
      "recipe": "lines_find",
      "params": {
        "std_start": "100.0",
        "std_end": "0.0",
        "col": "y",
        "kind": "min",
        "kappa_peaks": "5.0",
        "resol": "None",
        "append": "True"
      }
    },
    {
      "cookbook": "cb",
      "recipe": "nodes_cont",
      "params": {
        "delta_x": "500",
        "kappa_nodes": "5.0",
        "smooth": "0"
      }
    },
    {
      "cookbook": "",
      "recipe": "_refresh",
      "params": { }
    }
  ]
}
```


Figure 4. Example of JSON file used for scripting Astrocook.

3. USE CASES

3.1 Interactive analysis of a VLT X-shooter spectrum

We showcase some of the capabilities of Astrocook by analizing one target from the XQ-100 catalogue,[19] acquired by X-shooter at the ESO VLT. The visual spectrum of the target, quasar J0234-1806 (emission redshift $z_{\rm em} \simeq 4.305$), is shown in Fig. 5, panel A (blue: flux density, orange: flux density error). The spectrum displays a wealth of both emission and absorption features, most notably the Ly α λ121.6 and C IV λ154.9 emission lines (redshifted to $\lambda \simeq 645$ nm and $\lambda \simeq 822$ nm respectively), the forest of Ly α absorbers bluewards of the Ly α emission, and several metal absorption complexes scattered throughout. Telluric absorptions not corrected by the reduction procedure are also observed at $\lambda \simeq 687$ nm–690 nm, $\lambda \simeq 759$ nm–768 nm, and from $\lambda \simeq 900$ nm redwards.

A first possible task is to model the C IV absorption doublets situated along the line of sight to the quasar. At this stage, only the region between the Ly α and C IV emissions is considered for simplicity. The analysis workflows goes as follows (the instructions are meant to be replicated by the reader interactively):

1. The region between the Ly α and C IV emissions is extracted by left-right clicking on the spectrum graph and selecting Extract region from the drop-down menu. A new session is created from the extracted spectral region.

2. The regions most affected by telluric absorption are zapped from the spectrum, to avoid interference with the detection of the spectral lines. This is done by left-right clicking on the graph and selecting Zap feature from the drop-down menu.

3. Absorption lines are detected by selecting Find lines... from the Recipes menu. In running the recipes, the parameter Resolution is assigned a value of 8900 (corresponding to the nominal resolution of X-shooter in

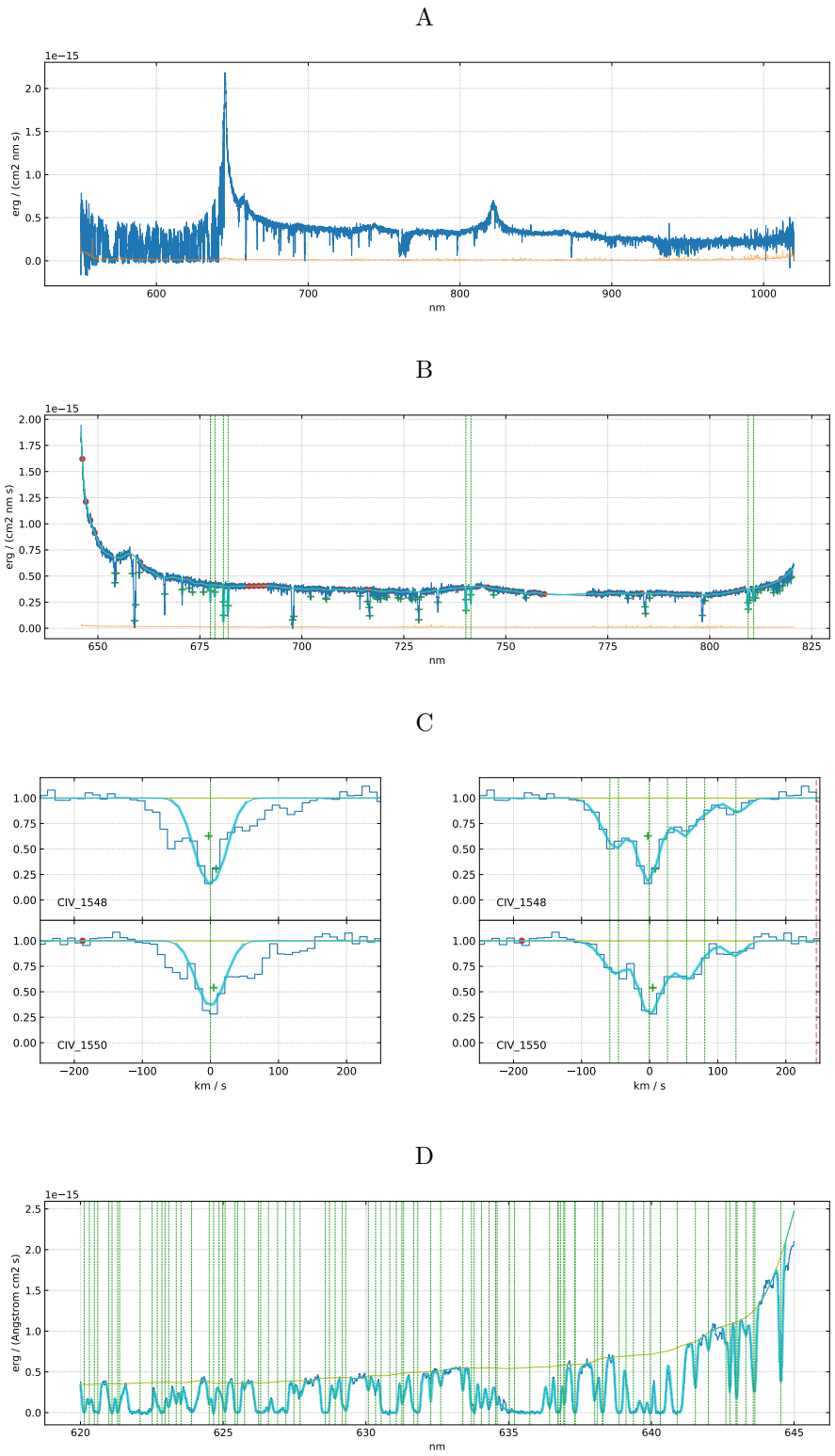

Figure 5. Analysis of quasar J0234-1806: A) spectrum as taken from the XQ-100 catalogue; B) detection of C IV absorption systems; C) detail of one absorption complex, before and after improving the model; D) detail of the fitted Ly α forest. A more complete explanation is given in Sect. 3.1.

the visual band for the adopted observational setup). A table `lines` is created and the detected lines are displayed as green crosses on the graph.

4. The emission continuum is estimated by selecting Continuum from nodes... from the Recipes menu. On the graph, the nodes are displayed as red dots and the continuum as an orange line.

5. The continuum is refined by manually deleting some nodes and inserting new ones. This is done by right clicking on the graph and selecting Remove node or Add node.

6. C IV systems are detected by selecting New systems from lines... from the Recipes menu and setting `CIV` as the chosen series of transition. After scanning the spectral region, the recipe should retrieve four systems that the user should be able to inspect by selecting System table from the View menu. System redshifts are displayed as dotted green bars, while system models are displayed as thick cyan lines. At this point, the graph should look similar to panel B of Fig. 5.

7. Individual systems can be inspected by clicking on the corresponding entry in the System table. An absorption complex at $z \simeq 3.397$ is shown in Fig. 5, panel C, left plot. A single system is clearly not enough to model this complex; the model is then improved by sliding a cursor (Show cursor in the System controls window) over the Spectrum detail window, and adding systems at the desired positions (New system from the drop-down menu that appears on right-click). New systems are automatically added to the model and inserted into the System table. A possible result, with six new systems added, is shown in Fig. 5, panel C, right plot.

A second task is to model the Ly α forest bluewards of the emission line at $\lambda \simeq 645\,\mathrm{nm}$. We restrict to a portion of the forest in the range $\lambda = 620\,\mathrm{nm}$–$645\,\mathrm{nm}$ and we assume that the line detection and continuum estimation have already been performed. In this case, we propose a slightly more advanced workflow, which goes as follows:

1. Detected lines are modeled as Ly α systems with New systems from lines... from the Recipes menu.

2. The first model is improved by automatically adding new systems in correspondence of strong fitting residuals. This is done by selecting Improve systems... from the Recipes menu. We enforce a refit after systems are added (Number of refit cycles equal to 1) to ensure that possible overlaps between adjacent models resulting from the addition of systems are taken into account. The whole addition/refitting procedure is iterated three times (Number of improve cycles equal to 3).

3. The information from the modeled systems can be used to improve the estimation of the continuum. The procedure is the same, but instead of rejecting the lines and interpolating the nodes in the unabsorbed regions, we correct the flux for the absorption (using the model) and interpolate the nodes all across the corrected spectrum. This is done again by selecting Find lines... and Continuum from nodes... from the Recipes menu (or faster by selecting Guess continuum... from the Set menus menu), with a particular choice of the parameters:

   - Column to convolve is set to deabs, to detect residual absorption in the de-absorbed flux column[**];
   - Append lines to existing line list is set to False, to replace the existing line mask with a new one masking only the residual absorption;
   - Finally, Size of slices is set to 200 km/s, to obtain a finer sampling of the continuum nodes.

   The result should look similar to Fig. 5, panel D.

4. Steps 2 and 3 can be iterated at will, to further refine the simultaneous modeling of the emission continuum and the absorption forest.

[**] deabs is computed as cont + y − model, where cont is the estimated continuum, y the absorbed flux density, and model the model of the absorbers.

The possible paths for further analysis are potentially endless (including e.g. the detection of metal-line contamination in the Ly α forest, based on the systems already detected redwards from the Ly α emission). The workflows above are meant to provide just an example of the capabilities of Astrocook, and to serve as templates to design different, possibly more sophisticated workflows.

3.2 Automatic analysis of a set of visual/spectra

Astrocook was also used to analyze the spectral observations of some peculiar quasar candidates from the QUBRICS survey.[5,6] These targets were observed in the visual band by different spectrographs (EFOSC2 at ESO-NTT, LDSS3 at the Magellan Clay Telescope, WFCCD at the du Pont Telescope, and MagE at the Magellan Baade Telescope) and in the near-infrared (NIR) band with FIRE at the Magellan Baade Telescope. Resolving powers ranged from ∼300 to ∼5000, depending on the instrument. A detailed discussion of the data set and its scientific significance will be the subject of an forthcoming publication; in this article we only describe the analysis procedure developed to treat these data, as an example of practical Astrocook usage.

The procedure was carried out on the exposures of 14 quasar candidates and their associated reference stars. Visual spectra of the targets had already been calibrated in relative flux densities by the reduction procedure, while NIR spectra were still expressed in terms of photon counts. The analysis workflow included seven steps:

1. NIR exposures of the quasars were equalized to have matching photon counts in the wavelength range $\lambda \simeq 1550\,\mathrm{nm}$–$1650\,\mathrm{nm}$ (recipe `GUIPanelSession.equalize`) and combined into a single spectrum (recipe `GUIPanelSession.combine`). Combined spectra were then rebinned into a fixed log-wavelength grid of $\Delta v = c\Delta \log \lambda = 50\,\mathrm{km\,s^{-1}}$ (recipe `CookbookGeneral.rebin`). The start and end wavelengths used for rebinning varied across the 14 targets to better match the wavelength range of the original exposures. Reference stars were rebinned to the same wavelength grid of the associated quasar spectra, to maintain the compatibility.

2. NIR spectra of the reference stars were cleaned from their most prominent absorption lines, identified by visual inspection (recipe `Session.spec._zap`). The resulting spectra are ideally free from absorption features intrinsic to the stars.

3. A black-body template with appropriate effective temperature was computed for each reference star (recipe `CookbookTemplate.bb`); cleaned NIR spectra of the stars were then normalized to these templates (recipe `GUIPanelSession.struct_modify`). The obtained spectra are ideally free from emission features intrinsic to the stars, and contain only the signatures of the telluric absorption and the instrument response. Quasar spectra were divided by these calibrators (recipe `GUIPanelSession.struct_modify`), resulting in telluric-free, flux-calibrated spectra.

4. The exposures of quasars that were observed more than once in the visual band were equalized to have matching relative flux densities in the superposition region (recipe `GUIPanelSession.equalize`), combined into a single flux-calibrated spectrum (recipe `GUIPanelSession.combine`), and rebinned into a fixed log-wavelength grid of $\Delta v = c\Delta \log \lambda = 300\,\mathrm{km\,s^{-1}}$ (recipe `CookbookGeneral.rebin`), to accomodate for the different resolving power of the instrument and avoid oversampling. Noisy regions at the ends of the spectra were cut out before coaddition (recipe `GUIPanelSession.extract_region`). When a single exposure was available, only rebinning was performed. The start and end wavelengths used for rebinning varied across the 14 targets to better match the wavelength range of the original exposures.

5. Visual and NIR flux-calibrated spectra were again equalized, combined, and rebinned to create full-length merged spectra (recipes `GUIPanelSession.equalize/combine/rebin`). The last rebinning further reduced the wavelength sampling to a log-wavelength step of $\Delta v = c\Delta \log \lambda = 500\,\mathrm{km\,s^{-1}}$. In some cases, the procedure pointed out a discrepancy between the visual and NIR wavelength scale. This discrepancy was corrected by rigidly shifting the NIR spectrum along wavelengths to match the visual spectrum (recipe `GUIPanelSession.struct_modify`), using shared feature as a reference.

6. Merged spectra were cleaned from the most prominent spurious features left by the reduction procedure, identified by visual inspection (recipe `Session.spec._zap`). The resulting spectra are deemed suitable for science.

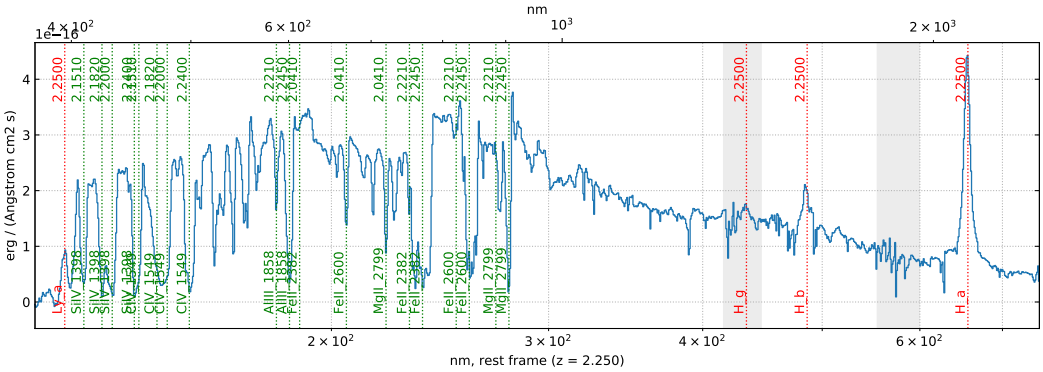

Figure 6. Example of quasar spectrum from the QUBRICS survey processed with the workflow described in Sect. 3.2.

7. Emission redshifts were determined by shifting an emission line mask along the spectra and visually finding the better alignment with the emission features (recipe `CookbookGraph.menu_view_cursor_z_series`). The mask contained transitions from the Lyman series (Ly α, Ly β λ102.6, Ly γ λ97.3) and Balmer series (H α λ656.3, H β λ486.1, H γ λ434.0) and metal transitions (N V $\lambda\lambda$123.9, 124.3; Si II λ130.4; Si IV $\lambda\lambda$140.0, 140.3; C IV; Al III $\lambda\lambda$185.5, 186.3; Mg II $\lambda\lambda$279.6, 280.4); the list of transitions was adjusted to match the actual presence of feature and the wavelength coverage of the spectrum. A similar operation was carried out to identify the most relevant absorption systems along the line of sight; in this case the list of possible metal transitions was expanded to include Fe II λ238.3 and Fe II λ260.0, and occasionally other less frequent transitions. Absorption and emission lines were highlighted in the final graph of the spectra (recipe `CookbookGraph.menu_view_cursor_z_series_stick`/`menu_view_graph_elem_edit`).

An example of processed spectrum is given in Fig. 6. It is worth noting that while the workflow was interactively refined by trials and errors on a few targets, it was subsequently scripted for easier re-execution and extension to the whole data set. The JSON scripts to run the analysis will be released together with the upcoming publication, to ensure full reproducibility of the results.

4. CONCLUSIONS

We have presented the latest implementation of Astrocook, a Python 3 package aiming to provide a comprehensive environment for the analysis of quasar spectra. We have described the main features of the software, namely its modular architecture based on small, reusable procedures ("recipes") suitable to be arranged into complex workflows, its interactive GUI to display the data and experiment with different analysis approaches, and its integrated scripting system which ensure an easy and reproducible control of the processing workflows. We have also presented some practical applications of the package, including the treatment of observational data from the QUBRICS survey. Other scientific applications of Astrocook will be discussed in forthcoming publications. This article provides the main reference to the version 1.0.0 of the software; future versions 1.n.n are expected to provide only minor upgrades of its functionalities and not to alter its core structure and overall look-and-feel.

There are, however, several extension of the Astrocook package being considered for a possible major overhaul in the near future:

- An integrated system to handle the inclusion of new recipes: Astrocook is designed to accommodate an ever-growing pool of recipes and to accept submission from the user community (e.g. through forks and pull requests on the GitHub repository). The process has not yet been standardized, and will possibly require a dedicated interface within the package itself.

- An extension to data reduction: while devoted to data analysis, the Astrocook approach can be naturally applied also to the calibration of raw data and the removal of instrumental signature. This can be done in two ways: by refactoring common reduction procedures as Astrocook recipes, or by enabling the Astrocook

to access existing reduction tools (e.g. the ESO HDRL[20] or the HiFLEx package[21]). Both alternatives will be possibly explored in the development of the data reduction software for the future CUBES spectrograph at the ESO VLT.[22]

- An extension to Bayesian techniques and machine learning: these approaches have been successfully applied to the modeling of both emission and absorption features in quasar spectra (see e.g. Refs. 23–25). It is very reasonable for the χ^2 minimization approach adopted by Astrocook to be complemented by similar techniques, to effectively cover a wider range of use cases.

The Astrocook team would greatly benefit from feedback and contributions from the user community in assessing these and other options for further development.

REFERENCES

[1] Pepe, F., Cristiani, S., Rebolo, R., Santos, N. C., Dekker, H., Cabral, A., Di Marcantonio, P., Figueira, P., Lo Curto, G., Lovis, C., Mayor, M., Mégevand, D., Molaro, P., Riva, M., Zapatero Osorio, M. R., Amate, M., Manescau, A., Pasquini, L., Zerbi, F. M., Adibekyan, V., Abreu, M., Affolter, M., Alibert, Y., Aliverti, M., Allart, R., Allende Prieto, C., Álvarez, D., Alves, D., Avila, G., Baldini, V., Bandy, T., Barros, S. C. C., Benz, W., Bianco, A., Borsa, F., Bourrier, V., Bouchy, F., Broeg, C., Calderone, G., Cirami, R., Coelho, J., Conconi, P., Coretti, I., Cumani, C., Cupani, G., D'Odorico, V., Damasso, M., Deiries, S., Delabre, B., Demangeon, O. D. S., Dumusque, X., Ehrenreich, D., Faria, J. P., Fragoso, A., Genolet, L., Genoni, M., Génova Santos, R., González Hernández, J. I., Hughes, I., Iwert, O., Kerber, F., Knudstrup, J., Landoni, M., Lavie, B., Lillo-Box, J., Lizon, J. L., Maire, C., Martins, C. J. . A. P., Mehner, A., Micela, G., Modigliani, A., Monteiro, M. A., Monteiro, M. J. P. F. G., Moschetti, M., Murphy, M. T., Nunes, N., Oggioni, L., Oliveira, A., Oshagh, M., Pallé, E., Pariani, G., Poretti, E., Rasilla, J. L., Rebordão, J., Redaelli, E. M., Santana Tschudi, S., Santin, P., Santos, P., Ségransan, D., Schmidt, T. M., Segovia, A., Sosnowska, D., Sozzetti, A., Sousa, S. G., Spanò, P., Suárez Mascareño, A., Tabernero, H., Tenegi, F., Udry, S., and Zanutta, A., "ESPRESSO@VLT – On-sky performance and first results," *arXiv e-prints* , arXiv:2010.00316 (Oct. 2020).

[2] Marconi, A., Di Marcantonio, P., D'Odorico, V., Cristiani, S., Maiolino, R., Oliva, E., Origlia, L., Riva, M., Valenziano, L., Zerbi, F. M., Abreu, M., Adibekyan, V., Allende Prieto, C., Amado, P. J., Benz, W., Boisse, I., Bonfils, X., Bouchy, F., Buchhave, L., Buscher, D., Cabral, A., Canto Martins, B. L., Chiavassa, A., Coelho, J., Christensen, L. B., Delgado-Mena, E., de Medeiros, J. R., Di Varano, I., Figueira, P., Fisher, M., Fynbo, J. P. U., Glasse, A. C. H., Haehnelt, M., Haniff, C., Hansen, C. J., Hatzes, A., Huke, P., Korn, A. J., Leão, I. C., Liske, J., Lovis, C., Maslowski, P., Matute, I., McCracken, R. A., Martins, C. J. A. P., Monteiro, M. J. P. F. G., Morris, S., Morris, T., Nicklas, H., Niedzielski, A., Nunes, N. J., Palle, E., Parr-Burman, P. M., Parro, V., Parry, I., Pepe, F., Piskunov, N., Queloz, D., Quirrenbach, A., Rebolo Lopez, R., Reiners, A., Reid, D. T., Santos, N., Seifert, W., Sousa, S., Stempels, H. C., Strassmeier, K., Sun, X., Udry, S., Vanzi, L., Vestergaard, M., Weber, M., and Zackrisson, E., "EELT-HIRES the high-resolution spectrograph for the E-ELT," in [*Ground-based and Airborne Instrumentation for Astronomy VI*], Evans, C. J., Simard, L., and Takami, H., eds., *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series* **9908**, 990823 (Aug. 2016).

[3] Cupani, G., Calderone, G., Cristiani, S., Di Marcantonio, P., D'Odorico, V., and Taffoni, G., "Astrocook: a thousand recipes to cook a spectrum," in [*Software and Cyberinfrastructure for Astronomy V*], Guzman, J. C. and Ibsen, J., eds., *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series* **10707**, 1070723 (July 2018).

[4] Van Rossum, G. and Drake, F. L., [*Python 3 Reference Manual*], CreateSpace, Scotts Valley, CA (2009).

[5] Calderone, G., Boutsia, K., Cristiani, S., Grazian, A., Amorin, R., D'Odorico, V., Cupani, G., Fontanot, F., and Salvato, M., "Finding the Brightest Cosmic Beacons in the Southern Hemisphere," *Astrophysical Journal* **887**, 268 (Dec. 2019).

[6] Boutsia, K., Grazian, A., Calderone, G., Cristiani, S., Cupani, G., Guarneri, F., Fontanot, F., Amorin, R., D'Odorico, V., Giallongo, E., Salvato, M., Omizzolo, A., Romano, M., and Menci, N., "The Spectroscopic Follow-up of the QUBRICS Bright Quasar Survey," *Astrophysical Journal Supplement Series* **250**, 26 (Oct. 2020).

[7] Cupani, G., Cristiani, S., D'Odorico, V., Pomante, E., Calderone, G., Di Marcantonio, P., and Marconi, A., "E-ELT HIRES the high resolution spectrograph for the E-ELT: integrated data flow system," in [*Observatory Operations: Strategies, Processes, and Systems VI*], Peck, A. B., Seaman, R. L., and Benn, C. R., eds., *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series* **9910**, 99102F (July 2016).

[8] Cupani, G., D'Odorico, V., Cristiani, S., González Hernández, J. I., Lovis, C., Sousa, S., Di Marcantonio, P., and Mégevand , D., "Field Tests for the ESPRESSO Data Analysis Software," in [*Astronomical Data Analysis Software and Systems XXVI*], Molinaro, M., Shortridge, K., and Pasian, F., eds., *Astronomical Society of the Pacific Conference Series* **521**, 362 (Oct. 2019).

[9] Pepe, F., Cristiani, S., Rebolo, R., Santos, N. C., Dekker, H., Mégevand, D., Zerbi, F. M., Cabral, A., Molaro, P., Di Marcantonio, P., Abreu, M., Affolter, M., Aliverti, M., Allende Prieto, C., Amate, M., Avila, G., Baldini, V., Bristow, P., Broeg, C., Cirami, R., Coelho, J., Conconi, P., Coretti, I., Cupani, G., D'Odorico, V., De Caprio, V., Delabre, B., Dorn, R., Figueira, P., Fragoso, A., Galeotta, S., Genolet, L., Gomes, R., González Hernández, J. I., Hughes, I., Iwert, O., Kerber, F., Landoni, M., Lizon, J. L., Lovis, C., Maire, C., Mannetta, M., Martins, C., Monteiro, M. A., Oliveira, A., Poretti, E., Rasilla, J. L., Riva, M., Santana Tschudi, S., Santos, P., Sosnowska, D., Sousa, S., Spanò, P., Tenegi, F., Toso, G., Vanzella, E., Viel, M., and Zapatero Osorio, M. R., "ESPRESSO — An Echelle SPectrograph for Rocky Exoplanets Search and Stable Spectroscopic Observations," *The Messenger* **153**, 6–16 (Sept. 2013).

[10] Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del R'ıo, J. F., Wiebe, M., Peterson, P., G'erard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E., "Array programming with NumPy," *Nature* **585**, 357–362 (Sept. 2020).

[11] Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods* **17**, 261–272 (2020).

[12] Cupani, G., D'Odorico, V., Cristiani, S., González-Hernández, J. I., Lovis, C., Sousa, S., Calderone, G., Cirami, R., Di Marcantonio, P., and Mégevand, D., "Integrated data analysis in the age of precision spectroscopy: the ESPRESSO case," in [*Software and Cyberinfrastructure for Astronomy IV*], Chiozzi, G. and Guzman, J. C., eds., *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series* **9913**, 99131T (July 2016).

[13] Bosman, S. E. I., Ďurovčíková, D., Davies, F. B., and Eilers, A. C., "A comparison of quasar emission reconstruction techniques for $z \geq 5.0$ Lyman-α and Lyman-β transmission," *arXiv e-prints* , arXiv:2006.10744 (June 2020).

[14] Aaronson, M., McKee, C. F., and Weisheit, J. C., "The identification of absorption redshift systems in quasar spectra.," *Astrophysical Journal* **198**, 13–30 (May 1975).

[15] Fontana, A. and Ballester, P., "FITLYMAN: a Midas tool for the analysis of absorption spectra.," *The Messenger* **80**, 37–41 (June 1995).

[16] Carswell, R. F. and Webb, J. K., "VPFIT: Voigt profile fitting program," (Aug. 2014).

[17] Newville, M., Stensitzki, T., Allen, D. B., and Ingargiola, A., "LMFIT: Non-Linear Least-Square Minimization and Curve-Fitting for Python," (Sept. 2014).

[18] Hunter, J. D., "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering* **9**(3), 90–95 (2007).

[19] López, S., D'Odorico, V., Ellison, S. L., Becker, G. D., Christensen, L., Cupani, G., Denney, K. D., Pâris, I., Worseck, G., Berg, T. A. M., Cristiani, S., Dessauges-Zavadsky, M., Haehnelt, M., Hamann, F., Hennawi, J., Iršič, V., Kim, T. S., López, P., Lund Saust, R., Ménard, B., Perrotta, S., Prochaska, J. X., Sánchez-Ramírez, R., Vestergaard, M., Viel, M., and Wisotzki, L., "XQ-100: A legacy survey of one hundred $3.5 \lesssim z \lesssim 4.5$ quasars observed with VLT/X-shooter," *Astronomy and Astrophysics* **594**, A91 (Oct. 2016).

[20] Gabasch, A., Modigliani, A., Salmistraro, M., Escartin, J. A., Neeser, M. J., Moehler, S., and Freudling, W., "The High Level Data Reduction Library," in [*Astronomical Data Analysis Software and Systems XXVII*], Ballester, P., Ibsen, J., Solar, M., and Shortridge, K., eds., *Astronomical Society of the Pacific Conference Series* **522**, 631 (Apr. 2020).

[21] Errmann, R., Cook, N., Anglada-Escudé, G., Sithajan, S., Mkrtichian, D., Semenko, E., Martin, W., Tanvir, T. S., Feng, F., Collett, J. L., and Jones, H. R. A., "HiFLEx—A Highly Flexible Package to Reduce Cross-dispersed Echelle Spectra," *Publications of the Astronomical Society of the Pacific* **132**, 064504 (June 2020).

[22] Evans, C. J., Barbuy, B., Castilho, B., Smiljanic, R., Melendez, J., Japelj, J., Cristiani, S., Snodgrass, C., Bonifacio, P., Puech, M., and Quirrenbach, A., "Revisiting the science case for near-UV spectroscopy with the VLT," in [*Ground-based and Airborne Instrumentation for Astronomy VII*], Evans, C. J., Simard, L., and Takami, H., eds., *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series* **10702**, 107022E (July 2018).

[23] Liang, C. and Kravtsov, A., "BayesVP: Full Bayesian Voigt profile fitting," (Nov. 2017).

[24] Ďurovčíková, D., Katz, H., Bosman, S. E. I., Davies, F. B., Devriendt, J., and Slyz, A., "Reionization history constraints from neural network based predictions of high-redshift quasar continua," *Monthly Notices of the Royal Astronomical Society* **493**, 4256–4275 (Apr. 2020).

[25] Lee, C.-C., Webb, J. K., Carswell, R. F., and Milakovic, D., "Artificial intelligence and quasar absorption system modelling; application to fundamental constants at high redshift," *arXiv e-prints* , arXiv:2008.02583 (Aug. 2020).