A Python-based tool for spectral line identification in RGS spectra from XMM-Newton

Parag Bhattacharya[1], Bansy M. Lyngdoh[2], Jessica I. Nongrum[3], Ranjeev Misra[4], Monmoyuri Baruah[5]

[1]Department of Physics, Assam Don Bosco University
Tapesia Gardens, Kamarkuchi, Sonapur – 782402, Assam. INDIA.
parag.bhattacharya@dbuniversity.ac.in

[2]Department of Physics, Assam Don Bosco University,
Tapesia Gardens, Kamarkuchi, Sonapur – 782402, Assam. INDIA.
bansylyngdoh@gmail.com

[3]Department of Physics, Assam Don Bosco University,
Tapesia Gardens, Kamarkuchi, Sonapur – 782402, Assam. INDIA.
jessicaibawansuknongrum@gmail.com

[4]Inter-University Centre for Astronomy and Astrophysics
Post Bag 4, Ganeshkhind, Pune – 411007, Maharashtra. INDIA.
rmisra@iucaa.in

[5]Department of Physics, Assam Don Bosco University,
Tapesia Gardens, Kamarkuchi, Sonapur – 782402, Assam. INDIA.
monmoyuri.baruah@dbuniversity.ac.in

***Abstract*: The identification of absorption and emission lines due to atomic species in the spectra of a X-ray binaries can reveal a wealth of information regarding the composition and physics of the stellar atmosphere. With the availability of high-resolution X-ray spectra from the RGS equipment on-board the XMM-Newton satellite, the study of such lines offers valuable diagnostics into the behaviour and evolution of the source object. Currently, data related to various atomic transitions, which lead to line formation, are made publicly available in the form of credible databases, one of them being the Atomic Spectra Database (ASD) at the National Institute of Standards and Technology (NIST). This work seeks to provide a single Python-based tool that accesses the relevant atomic data at NIST ASD for a given set of atomic species in a specific wavelength range and then overlays these lines on top of an X-ray spectrum obtained by the RGS spectrum of the XMM-Newton. With this tool, the astronomer can perform the important preliminary task of rudimentary line identification, before proceeding to advanced analysis of the X-ray data.**

***Keywords*: X-ray spectral analysis, line identification, atomic line database, Python for astronomy**

I. INTRODUCTION

XMM-Newton is an X-ray space observatory which was launched by the European Space Agency on December 10, 1999. It is designed to be a high throughput X-ray spectroscopy mission with spectral resolution of up to 0.025 Å at 1 keV from the *Reflection Grating Spectrometer* (RGS) detectors and an angular resolution of up to 1.1 arcsec from the *European Photon Imaging Cameras* (EPIC) [1, 2]. With a bandpass of 5-38 Å, corresponding to the energy range 0.33-2.5 keV, the spectra detected by RGS spans a substantial number of X-ray emission lines, which include the K-shell transitions and He-like triplets of light elements, such as C, N, O, Ne, Mg and Si, as well as the L-shell transitions of heavier elements like Fe and Ni. These factors enable the RGS spectra to be useful as diagnostic tools that can be used to investigate the physical conditions as well as the composition of the source of the spectra.

The RGS spectra are available in the public domain in the *Flexible Image Transport System* (FITS) file format [3], which is an open standard describing the digital file format for the storage, transmission and processing of data – formatted as multi-dimensional columns of tables. In spite of having the word ‘image’ in its acronym, FITS files are most often used to store non-image data as well. This standard was designed, keeping in mind specifically astronomical data, namely images, spectra, lightcurves, photon lists, event lists, and source lists. In order to download the raw science data collected by XMM-Newton for any given source, there are two options:

a) By accessing the *XMM-Newton Science Archive* (XSA) at the European Space Agency (ESA), using a web service (via a search form or a URL command access or methods from the `astroquery.esa.xmm_newton` Python module) or by *table access protocol* (TAP) queries to the XSA database [4].

b) By accessing the *High Energy Astrophysics Science Archive Research Center* (HEASARC) at the National Aeronautics and Space Administration (NASA) using a web service [5].

The XMM-Newton Science Operations Centre (SOC) provides a software package called the *Science Analysis System* (SAS), which is a collection of tasks, scripts and libraries designed for the specific tasks of the reduction and analysis of the raw science data collected by the XMM-Newton observatory [6]. Using SAS, one can extract an RGS spectrum from the science data for a given X-ray source.

An X-ray spectrum from RGS is found to contain a multitude of absorption and emission lines corresponding to atomic transitions in elements heavier than H and He. One of the initial and crucial tasks of the observer is to make a preliminary identification of the elemental lines at the wavelengths where they appear. This line identification also needs to incorporate the shift in the apparent position of the wavelength due to the Doppler effect. Currently, such line identification is performed by spectral-fitting programs such as Xspec or Spex while fitting the spectrum to prior theoretical models in which one can include the desired elemental abundances. However, when the spectrum used is of very high-resolution (such as that obtained from XMM-Newton), it often leads to poor fits, especially when one includes non-LTE model atmospheres. Therefore, the motivation behind this work are as follows:

a) It seeks to upend the approach for fitting such high-resolution spectra by allowing the user to first quickly identify the prominent lines present in the spectrum as well the Doppler shift in the lines (if any)

b) The user may then proceed to estimate the elemental abundances (from the presence of specific lines) and the radial velocity of the emitting material (from the Doppler shifts)

c) This would allow the user to develop models for the individual lines and thereby engineer composite theoretical models which are more phenomenological as opposed to traditional approaches [7].

As present, while there are available codes that enable the parsing of atomic line data from NIST ASD (such as the `nist-asd` Python package on PyPI) or object-oriented interface for Xspec (i.e. PyXspec), there is a lack of tools or codes that allow line identification prior to the development of models which may be then fitted to spectra. The current work is an attempt in this particular direction.

In order get started, one first needs to obtain data of the atomic lines from credible databases, such as the *Atomic Spectra Database* (ASD) at the *National Institute of Standards and Technology* (NIST) [8] and *the Opacity Project online atomic database* (TOPbase) [9]. To obtain this data, one may either use the web service provided to send queries to these databases, or one may use the relevant methods in various Python modules.

Having identified the prominent lines in the RGS spectrum, one can then proceed to estimate the radial velocity of the source, the abundance of the elements in the emitting material and the interstellar absorption along the line-of-sight.

II. Extraction of RGS spectra

A. *Obtaining source-specific science data*

XMM-Newton data is publicly available at the online multi-mission science data archive known as HEASARC (`https://heasarc.gsfc.nasa.gov/`). The easiest way to access relevant data for any specific source is by using its browse service at `https://heasarc.gsfc.nasa.gov/db-perl/W3Browse/w3browse.pl` where one can enter the name of the source (or its right ascension and declination, if these are known) and the mission name (such as XMM-Newton or Chandra). One can supply additional information about the source as well, such as the observation dates. The more specific the query, the lesser would be the time taken to access and download the data.

HEASARC provides the relevant data products in the form of a `.tar` file, which may then be extracted to obtain the *Observation Data Files* (ODF), in case of XMM-Newton.

B. *Reprocessing of science data*

The downloaded ODF files are compressed in `.gz` format and need to be extracted before reprocessing them using the software package SAS. Two SAS commands are used during the reprocessing – `cifbuild` and `odfingest`. While the former produces an index file of all the calibration data relevant to the specific source under consideration, the latter creates a summary file of the ODF using the house-keeping data files and the calibration database [6].

Subsequently, the data from both RGS instruments for the first and second spectral orders is extracted using the SAS task `rgsproc`. In five processing stages, this task extracts the events, the angles, the filter events, the spectra, the response matrices and the lightcurves.

III. Accessing spectral data from FITS file

One of the outputs of the `rgsproc` task are two fluxed spectra – one for each spectral order. Each of these files contain the spectra fluxed from both the RGS instruments. One must bear in mind that these fluxed spectra are inherently just a qualitative summary of the data, and should not be used for quantitative analysis. However, in the current work, because the objective is to merely aid the user in the identification of atomic lines – which could be a preliminary step before serious spectral modeling, these fluxed spectra are more than sufficient as input files.

A. *The `astropy` Python module*

The fluxed spectrum file is saved in the FITS format. So the appropriate way to read its contents and extract the data is by using `astropy` [10], which is an open-source and community-developed Python package providing various functionalities that are important to astronomical applications.

The `astropy.io.fits` package provides the user with access to FITS files. One first reads the list of *header data units* (HDU), which is the highest level component of a FITS file – consisting of a header and a table (or data array). This can be done using the `astropy.io.fits.open()` function which takes the FITS filename as input and returns an object of the `HDUList` class – a list-like collection of HDUs.

B. *Reading spectral data from FITS file*

One can access the individual HDUs in a FITS file using the array indexing syntax – the first index pointing to the primary HDU by default. For instance, if the object returned by the `astropy.io.fits.open()` function is named `hdul`, then `hdul[0]` would be the primary HDU, `hdul[1]` the first extension HDU and so on.

In case of the fluxed spectra from RGS, the FITS files contain a single extension HDU corresponding to the spectrum – accessed using `hdul[1]`, which is a table containing 3600 rows and 3 columns. Then the `data` attribute of `hdul[1]` (or `hdul[1].data`) would return a numpy numerical array containing the channels/wavelengths (in Å), the flux and the errors (both in $s^{-1}cm^{-2}Å^{-1}$). The fluxes and errors corresponding to all bad channels contain `NaN` values, which can be filtered using the `isnan()` function and set to zero.

IV. Collecting atomic lines

In order to be able to identify atomic lines in a spectrum, one needs to have a line list that contains information about the lines present within a given wavelength range for all the ion stages of the atoms under consideration. With tremendous improvement in computing power, the various online databases of atomic data provide more accurate and updated information that may be used to built better line lists.

So a code for qualitative atomic line identification in a spectrum becomes more reliable if it is able to access the current atomic data available online. This work achieves this objective by retrieving data for atomic transitions directly from ASD at NIST during runtime. So it is important to have an internet connection available while running the code for the first time or when one modifies the list of ions considered. As long as one keeps running the code subsequently on the same terminal session, using an unaltered ion list, the code would proceed to use the atomic data which is cached locally and one may work offline in that case.

A. *The `astroquery` Python module*

Atomic data from NIST can be accessed using the `astroquery` module [11], which uses the Python `requests` package for making HTTP requests and utilizes the data parsing functionality of the `astropy` package. The NIST ASD can be sent query requests and data retrieved therefrom using the `astroquery.nist` package.

B. *Reading atomic line data from NIST*

The `astroquery.nist.Nist.query()` function provides an easy way to send HTML requests to NIST ASD. This function takes, as inputs, the lower and upper wavelength values and the name of the ion to be considered. One can also make use of the `astropy.units.AA` attribute to convert the wavelength value to Å units.

This function returns an object of the `astropy.table.Table` class. Then individual table rows can be accessed by the indexing syntax. The elements of a row can be subsequently accessed using the corresponding column header as a dictionary key. For instance, if the object named `ion_table` stores the output of the `query()` function, then one can access the Ritz wavelength on the 17th row as `ion_table[16][‘Ritz’]`.

V. Demonstration

What follows is a demonstration of the developed tool using the flux spectrum of the galactic luminous supersoft X-ray source RX J0925.7-4758, which was discovered in the Galactic Plane Survey of the ROSAT All-Sky Survey [12]. The specific observations were made during 16-17 December 2000 with the RGS exposures being carried out for a duration of 61.2 ks [13] under the observation ID 0111150101. The ODF files were downloaded from HEASARC and RGS data were processed using the SAS task `rgsproc` under default settings.

Because the objective of this work is to enable the identification of atomic lines for specfic ion stages present in the spectrum, the fluxed spectrum file named `P0111150101OBX000fluxed1000.FIT` (produced by `rgsproc`) is sufficient for the code to work. This specific file fluxes together the first-order spectra from both the RGS instruments. A copy of this file is provided with the distribution of the code for the user to reproduce the demonstrations that follow, as well as to try out variations of the run.

Three specific Python scripts are provided with the distribution, whose outputs are presented in this work. A study of these scripts provides the user with an experience of how to use the methods developed in this work. All the scripts take, as the second command-line argument, the name of the file containing the fluxed spectrum. Briefly, these scripts perform the following tasks:

i. `line-overlaid-flux.py`: computes all the atomic lines within the wavelength range chosen and produces a plot of the fluxed spectrum with these lines overlaid.

ii. `flux-inspect.py`: allows the user to examine specific regions of interest, such as the vicinity of a known atomic line, within the chosen wavelength range.

iii. `line-shift.py`: displays the Doppler shifts in the Lyman α, β, and γ lines for the H-like and He-like C, N and O ions, as well as some common Fe ion lines.

A. *Reading fluxed spectrum from FITS file*

The use of the `astropy.io.fits` module is wrapped up in a method named `get_flux`. This method takes a single argument, which is the name of the FITS file containing the fluxed spectrum – in this case, the file named `P0111150101OBX000fluxed1000.FIT`. It returns the wavelengths (in Å), the normalized flux and the error (both in s^{-1} cm^{-2} $Å^{-1}$) in the form of three lists, which can be subsequently manipulated.

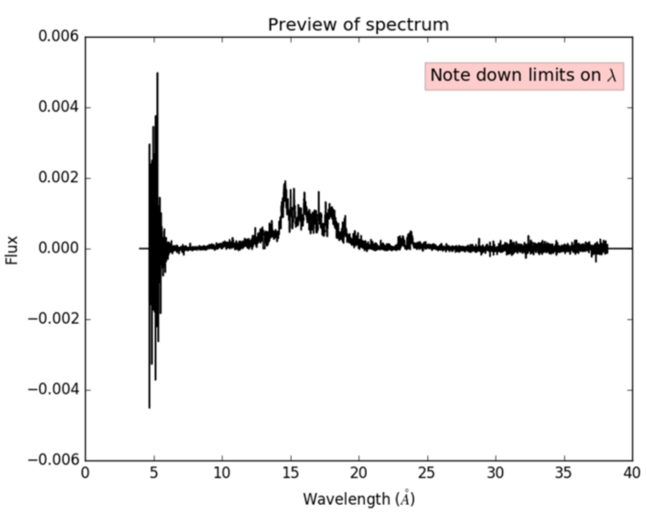

Figure 2: Preview of the fluxed spectrum produced to enable the user to decide a wavelength range.

B. Preview of spectrum

When one is working with the spectrum of a new source, one may not be aware of the wavelength region which contains most of photon flux. Also, RGS spectra tend to have a very low SNR in the small wavelengths up to ~7 Å. So, in order to help the user to ascertain a more pragmatic range, the function named `preview_flux` quickly plots the entire spectrum.This method takes, as arguments, two lists – the wavelengths in Å and the flux. The preview of the first-order fluxed RGS spectrum for RX J0925.7-4758 is shown in figure 1.

As it can be seen from the preview in figure 1, there seems to be plenty of information in the vicinity of 6 Å. But this is not usable because it is just noise arising from the low effective area of the RGS instrument in the low wavelength region which severly limits the resolving power of the RGS instrument, as described in the XMM-Newton User Handbook [1]. Also, the region above 26 Å has a very low SNR. Looking at the wavelength region between 10 Å and 26 Å one can clearly visualise a typical spectrum as expected from a luminous X-ray binary source with a good SNR. These wavelength limits are subsequently taken as inputs from the user by the method `get_wave_limits`.

C. Line list within chosen wavelength range

One needs the line list from the NIST database for the given list of ions within a chosen wavelength range. To achieve this, a method called `get_lines` is provided. This takes, as input arguments, the lower and upper wavelength values (in Å). It uses the list of ions provided in the file `ion.list`, which is accessed by the script using the `ASTRODAT` environment variable. Then it queries the NIST database using the `astroquery.nist.Nist.query`

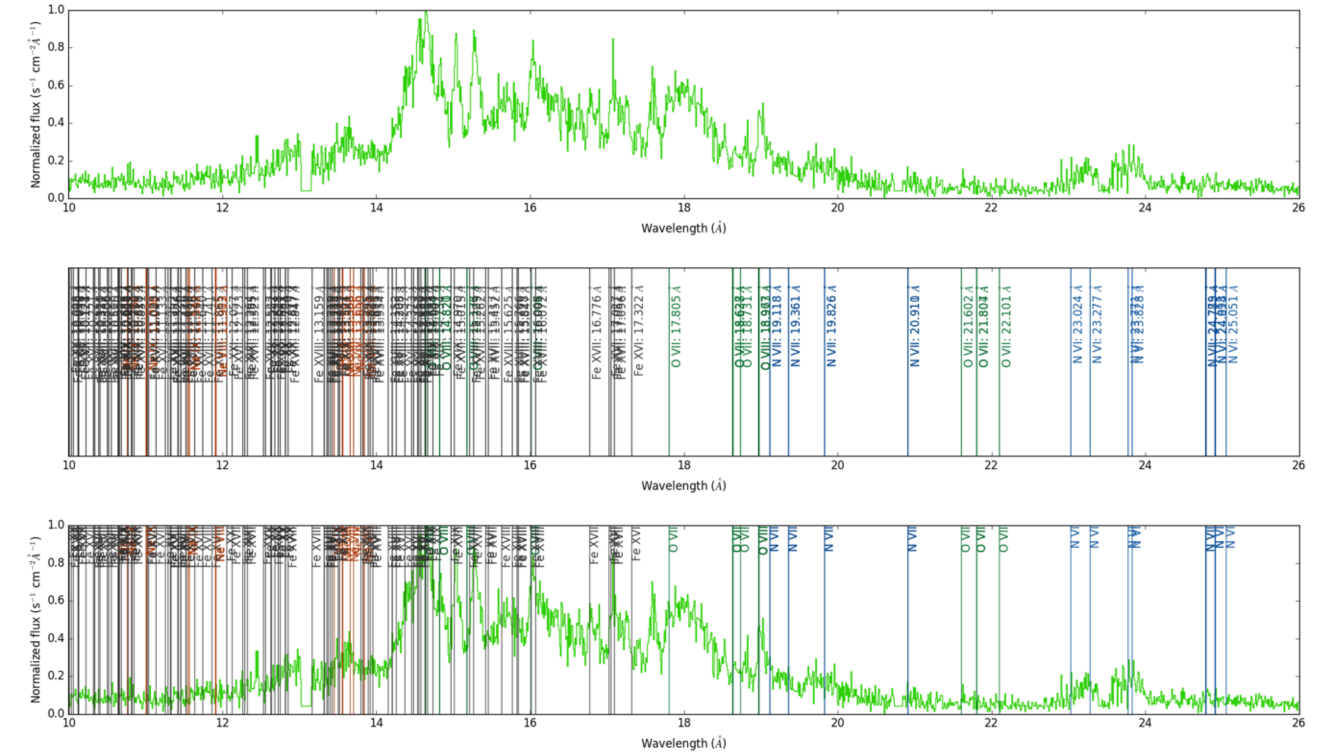

Figure 1: Output of the plot_spectrum method. The first subplot displays the entire spectrum within the chosen wavelength range. The second subplot displays a map of all the lines for the ions considered, which lie within the chosen wavelength. The third plot is the overlay of the second plot on the first.

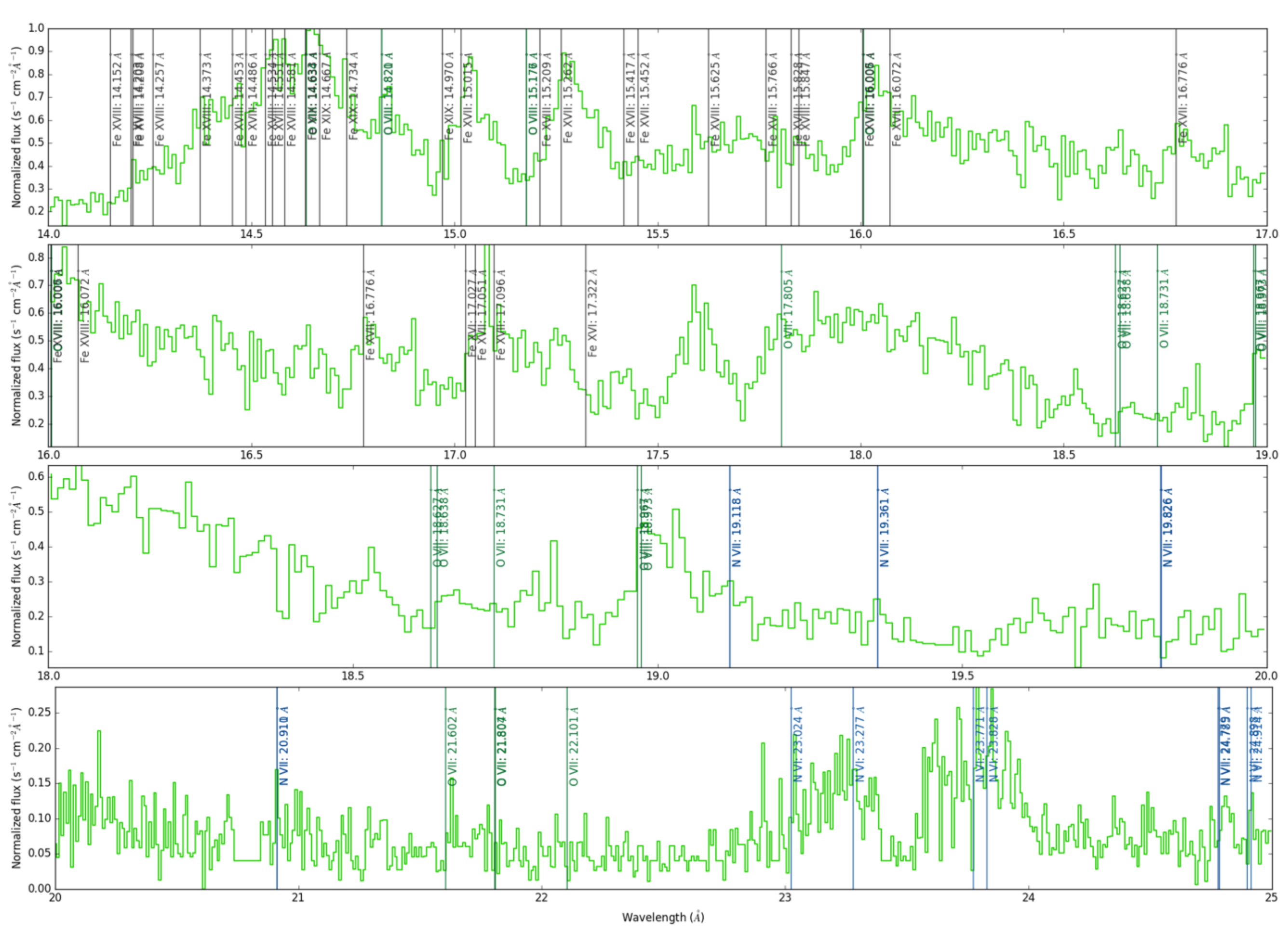

Figure 3: Partitioning of the fluxed spectrum within specific regions of interest - (from top to bottom) at 14.0 to 17.0 Å, at 16.0 to 19.0 Å, at 18.0 to 20.0 Å, and at 20 to 25 Å. The relative flux is normalized and the lines are labeled with wavelength values.

method.

The `get_lines` method returns a list of dictionaries, with each dictionary containing four entries: the wavelength (in Å), the associated ion, a boolean flag which indicates whether that line needs to be included and the colour with which the line is to be plotted. The colour of a line for a specific ion is obtained from another method called `get_colour`. This method uses a pre-defined dictionary of colour maps, from which a colour for a particular ion stage is sampled with the help of a `numpy` array. One may add more colour maps corresponding to other ions which may not be included in the code at present.

This method has two more default arguments that allow the user to provide a Doppler shift and to filter out ground level transitions. These arguments are as follows:

i. An argument `v_radial` to provide a radial velocity $v_{\rm rad}$ (in km/s), which is then used to compute the Doppler-shifted position λ of the line at λ_0 as

$$\lambda = \left(1 + \frac{v_{\rm rad}}{c}\right)\lambda_0 \qquad (1)$$

The default value of this argument is 0.

ii. A flag named `ground_transitions_only` which ensures that only those lines are considered which are a consequence of transitions into/from the ground level of the individual ion. The default value of this argument is `True`, which greatly reduces the density of lines being taken into account.

D. Line-overlaid spectrum

Having read the FITS file containing the fluxed spectrum and queried the line list from NIST, one can then proceed to produce a plot of the spectrum, along with the lines overlaid on it. This can be done using the `plot_spectrum` method. This method takes, as input arguments, the lists containing the wavelengths, the fluxes, the errors in the flux, the line list and the lower and upper limits of wavelength.

This method produces a set of three subplots: a plot of the spectrum only, a map of the lines within the wavelength region and a plot of the spectrum with the pertinent lines overlaid. The output of this code for the fluxed spectrum of RX J0925.7-4758 is shown in figure 2. As expected, one can observe a profusion of spectral lines due to K-shell transitions of N and O, and due to L-shell transitions of the heavier element Fe. Such an overlay of spectral lines provides a grid of markers of prominent lines. But the actual presence of any line in the spectrum may be latched on to after a closer inspection of sub-regions, as described in the following sub-section.

Prior to running the `plot_spectrum` method, one may also choose to display the wavelength values (in Å) along with the ion label, using the boolean flag `show_wavelength` input argument, whose default value is `False`.

One may also choose to normalize the fluxed spectrum to the range [0, 1] in s^{-1} cm^{-2} $Å^{-1}$ using the method `normalize`, which takes a list of floats as input and returns another list of floats normalized appropriately. This is helpful in comparing the relative strengths of different lines.

E. Inspecting regions of interest

While the previous method described produces the entire spectrum with overlaid lines, it can be more helpful if one could precisely zoom into any specific region of interest, say between 16.5 Å and 18.4 Å. This can be done with the `examine_ROI` method, which takes the lists of wavelengths, fluxes and errors, the line list, the chosen wavelength range and the lower and upper wavelengths (in Å) of the region of interest (e.g. 16.5 and 18.4).

Such an inspection of specific regions of interest was carried out for RX J0925.7-4758 and the resulting plots are shown in figure 3. One can observe here, for instance, two lines (one due to Fe XVII and the other due to O VIII) nearly overlapping with an emission line in the spectrum around 16.03 Å. Further analysis of the flux of other lines due to both atomic species would resolve this ambiguity.

F. Doppler shifts of lines

The emitting matter in stellar sources often have a radial velocity along the line-of-sight because of which their atomic lines often exhibit a Doppler shift, which translates the lines either blue-ward (i.e. towards smaller wavelengths) if the radial velocity is negative (moving towards the observer) or red-ward (i.e. towards larger wavelengths) if the radial velocity is positive (moving away from the observer).

This shift in the wavelength of a line can be obtained as

$$\Delta\lambda = \lambda - \lambda_0 \qquad (2)$$

where λ_0 is the central wavelength of the line and λ is the apparent position of the same line. The relation between these two is given by equation (1). By measuring the $\Delta\lambda$ value from the spectrum for some prominent and well-known atomic lines, one can estimate the radial velocity of the stellar matter which has produced the given spectrum.

One can make an initial estimate of the Doppler shift by plotting the vicinity of a specific line by transforming the horizontal axis from λ (in Å) to $v_{\rm rad}$ (in km/s). Then $v_{\rm rad}$ is obtained by inverting equation (1) as

$$v_{\rm rad} = \left(\frac{c}{\lambda_0}\right)(\lambda - \lambda_0) \qquad (3)$$

A method named `waveshift_to_velocity` is defined which returns $v_{\rm rad}$ in km/s, given λ_0 and λ in Å as input arguments.

The method named `plot_doppler_lines` can be used to plot the relative flux of different lines for a specific ion with respect to $v_{\rm rad}$. At present, this method can handle the Lyman α, β, and γ lines for the H-like and He-like C, N and O ions, as well as some common lines of Fe XVII and Fe XVIII.

The list of these lines is given in table 1 below.

*Table 1: List of lines along with transitions, considered for inspection with respect to the radial velocity. All wavelengths values are obtained from the NIST, except those marked with a * which were obtained from the Chianti database [14]*

Ion	Transition	Wavelength (in Å)
C V	Ly α	40.2678
	Ly β	34.9728
	Ly γ	33.4262
C VI	Ly α	33.7396
	Ly β	28.4663
	Ly γ	26.9901
N VI	Ly α	28.7870
	Ly β	24.8980
	Ly γ	23.7710
N VII	Ly α	24.7846
	Ly β	20.9106
	Ly γ	19.8261
O VII	Ly α	21.6020
	Ly β	18.6270
	Ly γ	17.7680*
O VIII	Ly α	18.9725
	Ly β	16.0067
	Ly γ	15.1765
Fe XVII	$2s^22p^6 \rightarrow 2s^22p^53d$	15.2620
	$2s^22p^6 \rightarrow 2s^22p^5(2P°_{3/2})3s$	17.0510
	$2s^22p^6 \rightarrow 2s^22p^5(2P°_{1/2})3s$	16.7760
Fe XVIII	$2s^22p^5 \rightarrow 2s^22p^4(1D)3d$	14.2080
	$2s^22p^5 \rightarrow 2s^22p^4(3P)3d$	14.3730*
	$2s^22p^5 \rightarrow 2s^22p^4(3P)3s$	16.0050*

Another method named `map_spec_to_vel` maps the λ values to the $v_{\rm rad}$ values in the vicinity of a particular line using the previously described methods. It takes as input arguments the central wavelength of the line λ_0, the lists containing the wavelengths, fluxes and errors, and an argument called `v_lim` which defines the bounds of $v_{\rm rad}$ values, the default value of which is ±5000 km/s.

Finally, the method named `plot_doppler_lines` wraps all of these and plots the line profiles for all the lines given in table 1. In case of RX J0925.7-4758, no lines of C V and C VI and also for N VI were detected within the chosen wavelength range. Shown in figures 4, 5 and 6 are the line profiles obtained for the N VII, O and Fe ions.

The `plot_doppler_lines` method takes, as input arguments, the symbol of the element (`'C'`, `'N'`, `'O'` or `'Fe'`), the list of wavelengths, fluxes and errors and the symmetric bound of the radial velocity about 0 (which corresponds to the line centre).

It also takes another argument named `mode` which serves to draw a dotted vertical line marker either through the lowest flux or the highest flux or both (within the range of radial velocities considered). Either of these options may be chosen by setting `mode='abs'` (default value) or `mode='ems'` respectively. One can also set `mode='Pcyg'` to display both the markers simultaneously.

However, one must be cautious while drawing any conclusions about the true position of the line centre from these markers because its position is merely ascertained from the flux values and are meant only to aid the user to get started with an approximate indication about the nature of the shift. For a more accurate evaluation, one needs to rely on a detailed quantitative analysis of the spectrum by using theoretical models.

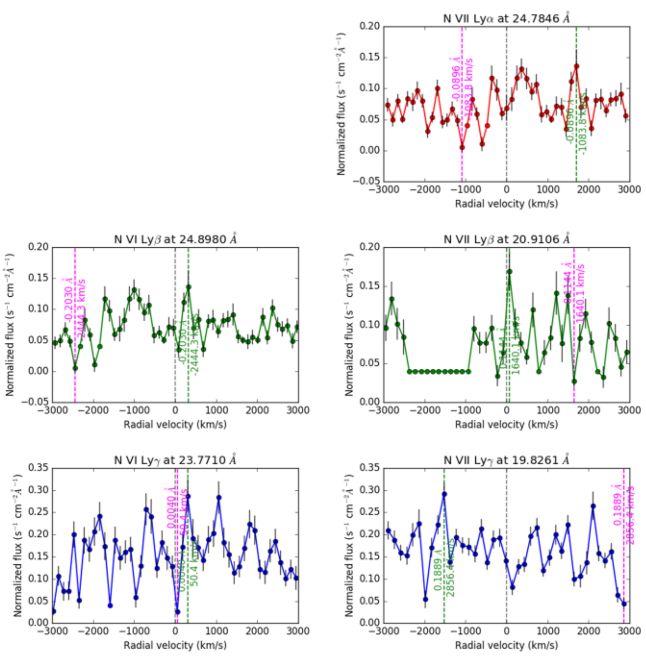

Figure 4: Velocity profiles of the Lyman α, β, and γ lines for N VI(left) and N VII (right). There is no Ly α line for N VI within the chosen wavelength range.

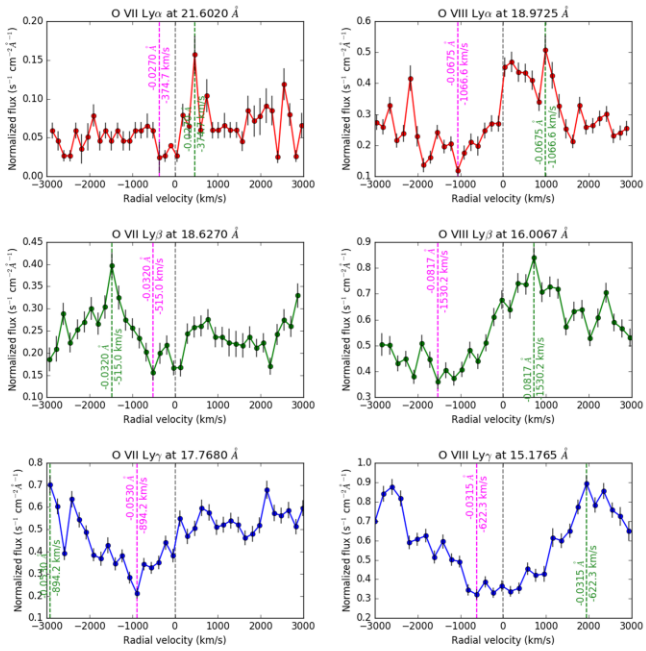

Figure 5: Velocity profiles of the Lyman α, β, and γ lines for O VII (left) and O VIII (right).

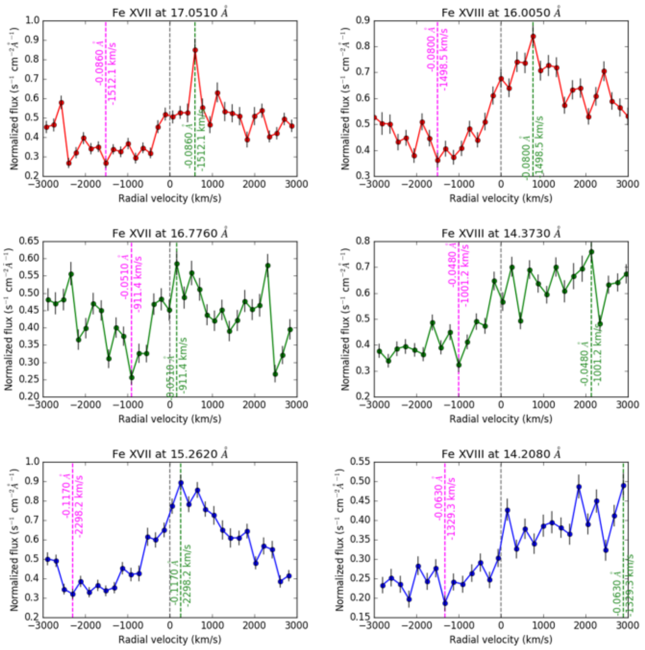

Figure 6: Velocity profiles of three Fe XVII (left) and Fe XVIII (right) lines.

VI. Distribution

The current version of the entire Python tool is publicly available for download and free use on Github at `https://github.com/pararover/xmmrgs_lines`. The authors welcome critical evaluation and suggestions, which would enable further improvement to the code. Future versions of the code would evolve on the basis of such feedback from an open community.

VII. Future scope

At present, the code is designed to work only with the RGS fluxed spectra from XMM-Newton. While there is no plan to extend its application to binned spectra, it is however planned that future versions of the code would be extended to work on the X-ray spectra from the EPIC MOS and PN instruments of XMM-Newton.

Currently, the range of velocity profiles produced by the code is highly restrictive – it provides the same only for the first three Lyman lines for the H-like and He-like ion stages of C, N, O and Fe only. In future versions, the authors intend to introduce further generalisation by allowing the same for any ground transition of any ion, provided any line for the same exists within the chosen wavelength range. The filtering of lines, however, is planned to remain on ground transitions only. The reason for this being that transitions between excited states are much less likely. Also, it is planned that lines could be additionally filtered out on the basis of transition probability – this would help reduce the cluttering of Fe lines in the lower wavelengths (see the 10 Å – 16 Å range in figure 2).

It is intended that the online atomic database accessed by the code would be extended to include other databases, such as Chianti database and TOPbase. In addition, further improvement of the code can be achieved by including some form of curation of the atomic line list across multiple different databases. The next version of the code would involve vectorized arrays, which is expected to optimize the computations. Improvements in the performance of the code due to such updates would be comparatively followed up in the future.

(1) Acknowledgment

The authors express their appreciation to Dr. Peter Nemeth from Astroserver (`www.astroserver.org`), for his valuable inputs and suggestions towards the development of the code used in this work.

(2) References

[1] M. Ehle, M. Breitfellner, M. Dahlem, M. Guainazzi, P. Rodriguez, M. Santos-Lleo, N. Schartel, and L. Tomas, "XMM-Newton Users' Handbook," Issue 2, 2003.

[2] F. Jansen, D. Lumb, B. Altieri, J. Clavel, M. Ehle, C. Erd, C. Gabriel et al., "XMM-Newton observatory-I. The spacecraft and operations," Astronomy & Astrophysics, vol. 365 no. 1, pp. L1-L6, 2001.

[3] FITS Working Group, "Definition of the flexible image transport system (FITS)," FITS Standard Version 4.0, 2016.

[4] C. Arviset, M. Guainazzi, J. Hernandez, J. Dowson, P. Osuna, and A. Venet. "The XMM-Newton Science Archive," arXiv preprint astro-ph/0206412, 2002.

[5] P. Barrett, "HEASARC the High Energy Astrophysics Science Archive Research Center," AnIPS, vol. 10, p. 276, 1993.

[6] I. de la Calle, "Users Guide to the XMM-Newton Science Analysis System", ESA: XMM-Newton SOC, Issue 15.0, 2019.

[7] J. U. Ness, "The complications of learning from Super Soft Source X-ray spectra," Advances in Space Research, vol. 66, no. 5, pp. 1202-1208, 2020.

[8] Y. Ralchenko, A. E. Kramida, and J. Reader, "NIST atomic spectra database," National Institute of Standards and Technology, Gaithersburg, MD, 2008.

W. Cunto, C. Mendoza, F. Ochsenbein, and C. J. Zeippen, "TOPbase at the CDS," Astronomy & Astrophysics, vol. 275, pp. L5, 1993.

[9] T. P. Robitaille, E. J. Tollerud, P. Greenfield, M. Droettboom, E. Bray, T. Aldcroft, M. Davis et al., "Astropy: A community Python package for astronomy," Astronomy & Astrophysics, vol. 558, p. A33, 2013.

[10] A. Ginsburg, M. Brigitta, C. Sipőcz, E. Brasseur, P. S. Cowperthwaite, M. W. Craig, C. Deil, A. M. Groener et al., "astroquery: An Astronomical Web-querying Package in Python," The Astronomical Journal, vol. 157, no. 3, p. 98, 2019.

[11] C. Motch, G. Hasinger, and W. Pietsch, "Discovery of a luminous galactic supersoft X-ray source in the ROSAT all-sky survey," Astronomy & Astrophysics, vol. 284, pp. 827-838, 1994.

[12] C. Motch, H. Bearda, and C. Neiner, "XMM-Newton observations of MR Vel/RX J0925. 7-4758," Astronomy & Astrophysics, vol. 393, pp. 913-920, 2002.

[13] K. P. Dere, E. Landi, H. E. Mason, B. C. Monsignori Fossi, and P. R. Young, "CHIANTI-an atomic database for emission lines," Astronomy and Astrophysics Supplement Series, vol. 125, no. 1, pp. 149-173, 1997.

(3) Author Profile

Parag Bhattacharya
Assistant Professor, Department of Physics, Assam Don Bosco University.
Areas of interest: X-ray binaries, astrophysical data analysis, stellar atmospheric modelling, development of Python scripts in astrophysics