The Swift Reduction Package Users' Manual

by Stefano Covino, 25 Mar 2014, v. ***3.11.0***

Background
The Swift Reduction Package (hereafter SRP) is a packet of tools executable on the command line to perform basic reduction and analysis tasks of optical/NIR astronomical data. There are several tools to organize observations, to manage FITS files, etc.
SRP is not meant to be "yet another" analysis package, our goal is indeed to provide simple tools to solve common problems in our daily working activities.
SRP was originally developed in the context of the *Swift* follow-up activities of the Milan GRB team at the INAF/Brera Astronomical Observatory. The package is designed to be an aid to any researcher to drive further observation of a followed-up GRB counterpart and "swift" can therefore be read simply as "rapid", "agile", etc.
The package is continuously upgraded and improved. Within the limits of our basic project choice (i.e. to provide an as simple as possible tools rather than a very powerful but complex reduction/analysis environment) any help is absolutely welcome.

Some technical comment
This package, written in Python (2.7 is the suggested version), has been widely tested only on PC-Linux and on Mac OS X workstations. You are anyway absolutely free to use, modify, redistribute this package as you like.
Of course, in any case, we decline any responsibility for the use of this package. Given that the sources are available, and the algorithm public, the results are entirely under your own responsibility.

Mailing-List
Due to the nature of the **SRP** project it quite likely to have frequent update and improvements of the various routines as well as a continuous bug fixing (and new bugs are definitely introduced after any feature additions...). Therefore, if you want to be warned each time a new version is delivered, please send an e-mail to the following address: stefano.covino@brera.inaf.it with subject simply **SRP**.

Installation
If you are just updating **SRP** the simplest and suggested solution is to download the package from the PyPI archive with:

sudo easy_install --script-dir=/usr/local/bin -U SRPAstro

provided of course you are connected to the web, and that you want your executable files in "/usr/local/bin".

If you, instead, are installing **SRP** for the first time or maybe you are upgrading to a new **Python** release, it is likely you need to install many different libraries **SRP** relies on.

In principle the command:

sudo easy_install --script-dir=/usr/local/bin -U SRPAstro

should again do the job. You might also consider to install the package in a virtual python environment if you do not want to interfere with the system python installation.

However, some of the required libraries can (will) require more concerned actions to allow their installation. Indeed, in essentially all cases, browsing the web you can quickly find the solution to any problem.

An alternative and strongly advised procedure is to install one of the available open-source self-contained scientific python installations as the Anaconda distribution. Most of the required libraries would then available with no further efforts and SRP is installed smoothly (the Ureka project also deserves consideration).

It is always possible to make a smart use of the various package managers available on many platforms (macports, yum, apt-get, etc.). A possible sequence of operations on Mac OSX is the following:

i) *sudo port -v selfupdate*
ii) *sudo port install python27*
iii) *sudo port select --set python python27*
iv) *sudo port install py27-distribute*
v) *sudo port install py27-numpy*
vi) *sudo port install py27-matplotlib*
vii) *sudo port install py27-scipy*
viii) *sudo port install py27-pyfits*
ix) *sudo port install py27-pil*
x) *sudo port install py27-astropy*
xi) *sudo easy_install --script-dir=/usr/local/bin SRPAstro*

while, on other Linux platforms, using yum or apt-get rather than port, an analogous sequence should work.
For instance, on a linux-PC running Fedora:

i) *sudo yum update*
ii) *sudo yum install python*
iii) *sudo yum install python-devel*
iv) *sudo yum install python-setuptools*
v) *sudo yum install numpy*
vi) *sudo yum install python-matplotlib*
vii) *sudo yum install scipy*
viii) *sudo yum install python-imaging*
ix) *sudo easy_install -s /usr/local/bin SRPAstro*

Finally, do not forget to install these two independent packages widely used by several **SRP** tools (you might use again port, yum, etc.).

1. The ESO-Eclipse package. **Eclipse** is a general purpose reduction system that can very easily be interfaced with regular UNIX commands to run pipelines, etc. The package should run on essentially any UNIX release. The suggested version is 5.0.
2. SExtractor, the well known general use photometric package. **SRP** has been tested with **SExtractor** version 2.0 or later. The suggested version is 2.5 essentially because later versions require quite a large set of libraries to be installed.

In addition to the "core" **SRP** package there are additions to cover specific tasks or applications:

a. SRP.FITS, providing general tools to handle FITS files usually, but not necessarily, obtained by optical/NIR telescopes.
b. **SRP.REM**, providing a few commands to work with REM telescope data.
c. **SRP.TNG**, providing a few commands to work with TNG telescope instruments.
d. SRP.NOT, providing a few commands to work with NOT telescope instruments.
e. **SRP.SRPPipelines**, general tools to develop pipelines for massive data reduction
f. **SRP.SRPPipelines.REM**, REM telescope pipeline command.
g. **SRP.SRPPipelines.TNG**, TNG telescope pipeline commands.

Only the SRP.FITS, SRP.REM and SRP.TNG sub.packages are distributed for general use. Installation simply requires to run: *sudo easy_install -s /usr/local/bin -U subpackage_name*

Step by step "how to"

This are just examples of what you can do with **SRP**. Please, pay attention that the main emphasis in developing these tools is put in the rapidity and friendly use rather than in getting the very best solution for any possible case. However, experience says that in most cases the results are fully acceptable.

Spectroscopy data analysis

1. Air / Vacuum wavelength conversion
   - This is a simple tool to convert air wavelength to/from vacuum wavelength. **SRPAirVacuum -A 6562.801**.
2. DLA , IGM, and line profiles

- We can also derive the absorption factor due to Dumped Lyman$_\alpha$ systems with **SRPDLA -l 0.4 -n 1e21 -z 1.0** and due to IGM with **SRPIGM -l 1.1 -x 0.9 -t 7.8**. For a general transition and a more accurate modeling you can derive the absorption factor with **SRPLineProfile -n 1e20 -b 1 -l 1300 -t TiII_3073**. This function compute a Voigt profile taking into account both Döppler and intrinsic broadening.

3. Solar abundances
   - Element abundances in the Sun can be retrieved i.e. with **SRPSolarAbundance -e Ne**. Data are from Asplund et al. (2009, ARA&A, 47, 481).

Magnitude, flux, frequency, absorption and reddening

1. Magnitude to/from flux conversion
   - It is quite common to have to convert magnitudes to physical units. To do that it is enough to run **SRPMagFlux -b R -m 18.0 0.03**. You will get the effective frequency, flux and 1-б error for magnitude, 1-б error and band reported. The output unit is Jansky or erg s-1cm-2A-1. It is of course possible to derive magnitudes from fluxes with the same command: **SRPMagFlux -b R -j 1e-5 2e-7**.
2. Reddening determination and absorption factors
   - Assuming to know the color excess E_{B-V} it is possible to estimate the amount of reddening at a given wavelength with **SRPDustAbs -g MW -w 0.7**. The wavelength unit is micron and the output is expressed in magnitudes. Extinction curves for the MW, the LMC and the SMC (Pei, 1992, ApJ, 395, 130) and a generic starburst galaxy (Calzetti et al., 2000, ApJ, 533, 682). It is also possible to know the absorption factor due to neutral gas in the X-rays (Morrison & McCammon, 1983, ApJ, 270, 119) with **SRPNhAbs -e 1 -n 1e21**.
3. Flux density and energy-frequency conversion
   - **SRPPLFluxDensity** allows you to compute the flux density at any given energy assuming to have a power-law spectrum and the integrated flux, i.e. **SRPPLFluxDensity -f 2.4e13 -s 1.7**. The integrated flux is supposed to be expresses in erg/s*cm2, the energy in keV, and the flux density is expressed in Jy. With **SRPEnergyFreqFlux -e 1 -j 1e-6** you can convert energy to frequency or wavelength and vice-versa and flux densities from Jy to erg/s cm^2 Å and viceversa. Energy is expressed in eV, frequency in Hz and wavelength in micron.
4. Atmospheric absorption
   - Atmospheric extinction coefficients for a few observing sites can be derived with **SRPAtmExtinction -w 0.55 -s LaSilla**.

Tables and catalogues

1. Catalogue query

- Given a position in the sky and a radius with the command **SRPQuery -a 01 23 45 -d 23 32 00 -c BS -r 120**. The radius unit is arcmin.

2. Table extraction
   - This command helps you to extract only selected columns from a table for further analyses. It is also possible to skip header rows as in **SRPTabExtract -t table.txt -c '2 3 4 5' -o _output.txt -j 2**.
3. Object extraction
   - You can select a single, or more, objects in a table basing on their coordinates with **SRPGetTabEntry -i table.dat -c 2 3 -C 240.45 312.23 -t 1**. It is possible to work with angular or Cartesian coordinates. You can also filter selected objects from a table with the same command.
4. Table match
   - This command allows you to find the common objects between two tables (i.e. the **SRPPhotometry** output). You must provide the file names of the table, shifts in both axes, and the maximum tolerance (first match and fine tuning) for object matching: **SRPMatch -r refile.dat -m matchfile.dat -t 5 2.5 -o _outfile.dat**. Same table scale and orientation is assumed. If, on the contrary, you want to find common objects with the same angular coordinates in two tables, you can try **SRPMatchCoord -r refile.dat -m matchfile.dat -t 1 -o _outfile.dat**.

Observation management

1. Target visibility
   - Providing geographic position of the observer, target coordinate and a time reference you can know the azimuth and altitude of the target and the Moon and Sun separation. It is thought as a simple aid for observation planning: **SRPVisibility -o 10:11:12 -20:21:22**. In case you do not provide any input apart from target coordinate the ESO-La Silla site and the present date are assumed.
2. Date conversion
   - It could be useful to quickly convert from/to dates in "regular" format (yyyy/mm/dd hh:mm:ss.dd) to Julian Date of Modified Julian Date. This can be done with **SRPCalendar -v -d 2009/4/8 10:51:00**.
3. Finding-chart creation
   - In order to quickly generate nice-looking (hopefully) finding charts you can use **SRPFindingChart -f fitsfilename**.

Statistics

1. Gaussian distribution
   - Providing mean and standard deviation you can generate an arbitrary number of values following a given Gaussian distribution: **SRPGaussDistrib -m 3 -s 0.1 -n 10**.
2. Gaussian probability
   - You can compute Gaussian 1-tail or 2-tail probability distribution given a value in units of sigma: **SRPGaussProb -s 3 -2**.

3. Chi square surface
   - You can easily derive the increment for the chi square function given some degrees of freedom at a given probability with **SRPChiSqIncrement -p 90 -d 5**. Else, you can compute the probability to have a chisquare higher than the obtained value by chance with **SRPChiSqIncrement -c 10 -d 7**.
4. Sigma-clipped average
   - It is often useful to can evaluate a sigma-clipped average for input data from a file. You can do that with **SRPAverSigmaClipping -i inputtab.txt -d 4 -k 5**.
5. Histogram creation
   - This command takes a column from a file as input and computes an histogram. The output is a table with bin center, bin size, and number of objects in the bin. Histogram parameters as minimum (i.e. 1), maximum (i.e. 10) and bin size (i.e. 0.2) can be chosen as in **SRPHistogram -c 2 -t table.txt -o _output.txt -b 1 10 0.2**.
6. Multi-parametric fits
   - This command is an attempt, far from being perfect in particular considering the error management, to provide a tool to carry out multi-parametric fits and Montecarlo error search. Fits are driven minimizing a χ^2 function, and the minimization algorithm is the popular Nelder-Mead Simplex algorithm using only function calls. It is not the most powerful, but it is easy and probably fully adequate for an “as friendly as possible” tool. The function to be fit must be provided in a file, i.e. *inp.py*, which is nothing more than a regular piece of python code imported by the command. This means that you may in principle write as complex functions as you need. An example is reported below:

```
import math

# pars[0] Normalization
# pars[1] Center
# pars[2] FWHM
# pars[3] Continumm level

# vars[0] First independent variable
# vars[1] Second, if any, independent variable and so on

def myfun (pars, vars):
# function Gaussian + constant
        y = pars[0]*math.exp(-((pars[1]-vars[0])**2)/(2*pars[2]**2)) + pars[3]
return y
```

   - The only requirement is that the function to fit is named *myfun*. Some more examples can be:

```
import math

# pars[0] slope
```

```
# pars[1] intercept

# vars[0] First independent variable

def myfun (pars, vars):
# a simple straight line
        y = pars[0]*vars[0] + pars[1]
return y
```

- Or, in case you prefer a smoothed power-law:

```
import math

# pars[0] first power-law index
# pars[1] second power-law index
# pars[2] break “time”
# pars[3] normalization
# pars[4] constant

# vars[0] First independent variable

def myfun (pars, vars):
          y= pars[3]/(math.pow((vars[0]/pars[2]),pars[0]) +
                math.pow((vars[0]/pars[2]),pars[1]))+pars[4]
return y
```

- The table with data must follow a simple but strict format: all the independent variables, dependent variable and error. The error is mandatory. An example is reported below:

```
7230.2 1.13E-17 1.13e-18
7232.8 1.40E-17 1.40e-18
7235.4 1.58E-17 1.58e-18
7238.0 1.68E-17 1.68e-18
7240.5 2.10E-17 2.10e-18
```

- where we have a wavelength and corresponding fluxes and errors. Of course all the suggestions to carry out these operations as keeping numbers of the order of unity, etc. are still fully valid. The format of a table can be modified with **SRPTabExtract**.
- A typical call to **SRPFit** can be: **SRPFit -v -d inputData.dat -g '1 1 1 1' -m inp.py**, where the string reported after the -g parameter is the string of the best guesses for the fitted parameters.
- After having obtained a stable fit you may try to estimate the confidence ranges for the fitted parameters. This task is usually rather tricky and not in general easy to automatize. **SRPFit** implements a Montecarlo search within some provided limits. A typical command could be **SRPFit -v -d inputData.dat -g '1 1 1 1' -m inp.py -o _errData.dat -e 90.0 -i '0.9 1.1 0.8 1.2 0.99 1.01 0.7 1.1'**. The best strategy could be to begin the search close to the best fit parameters and then enlarge the search range ap-

pending each iteration result to form a large sample of trial to constrain the error confidence regions.

7. F-Test computation
   - You can check the improvement of a fit quality adding parameters with the F-test: **SRPFTest -n 14.45 12 -o 16.34 14**.

Afterglow data and cosmology

1. Afterglow flux and frequencies
   - **SRPAftTypSynchrFluxConst(Wind)** allows you to compute the cooling (typical)(self absorption) synchrotron frequency (Hz) in case of constant ISM (wind). You compute the flux (Jy) at the typical synchrotron frequency and the flux for a typical afterglow spectrum at any frequency. From Zhang & Meszaros, IJMPA A19, 2385 (2004) and Hurley, Sari and Djorgovski, in "Compact Stellar X-ray Sources" (Cambridge University Press).
2. Cosmological parameter computation
   - This command allows one to compute the Hubble, angular diameter, co-moving and luminosity distances from a minimum to a maximum redshift in any expanding universe. The distance modulus is also derived. Universe parameters, i.e. matter density, cosmological constant, etc. can be provided. **SRPCosmology -v -z 1.5** you get the various distances for the default "concordance" cosmological model.

Miscellanea

1. **SRP** running version
   - You can know the present **SRP** version with **SRPVersion**.
2. Polarization data
   - Polarization data can be converted to/from Stokes parameters with **SRP-StokesPol -s 0.1 0.01 -0.1 0.01 0.0 0.0.**

List of commands

3. **SRPAftSynchrSpectrumConst**
   - Its purpose is to compute the afterglow synchrotron spectrum in case of constant density ISM.
   - SRPAftSynchrSpectrumConst [-b arg1] [-d arg2] [-e arg3] [-f arg4] [-g arg5] [-h] [-n arg6] [-p arg7] [-t arg8] [-v] [-z arg9]
     -b Epsilon B (0,1).
     -e EPSE Epsilon E (0,1).
     -d DIST Luminosity distance (cm).
     -g Isotropic energy (erg).

-n Particle density (cm^-3).
-f Frequency (Hz).
-p Electron distribution index.
-t Time from burst (days).
-z Source redshift.

Afterglow model for constant density environment.

4. **SRPAftSynchrSpectrumWind**
   - Its purpose is to compute the afterglow synchrotron spectrum in case of wind.
   - SRPAftSynchrSpectrumWind [-a arg1] [-b arg2] [-d arg3] [-e arg4] [-f arg5] [-h] [-g arg6] [-p arg7] [-t arg8] [-v] [-z arg9]
     -a Astar.
     -b Epsilon B (0,1).
     -e Epsilon E (0,1).
     -d Luminosity distance (cm).
     -g Isotropic energy (erg).
     -f Frequency (Hz).
     -p Electron distribution index.
     -t Time from burst (days).
     -z Source redshift.

     Afterglow model for wind shaped environemnt.
5. **SRPAirVacuum**
   - Its purpose is to convert air wavelength to vacuum wavelength and vicev-ersa.
   - SRPAirVacuum -A arg1 / -V arg1 [-h] [-v]
     -A Air wavelength (Angstrom)
     -V Vacuum wavelength (Angstrom)
6. **SRPAtmExtinction**
   - Its purpose is to derive atmospheric extinction coefficients.

   SRPAtmExtinction [-h] -l wave [-s site] [-v] [--version]
     -l Wavelength for extinction coefficient (micron)
     -s Site for the extinction curve
7. **SRPAverSigmaClipping**
   - Its purpose it to compute a sigma-clipped average for input data.
   - SRPAverSigmaClipping -i arg1 -d arg2 [-e arg3] [-h] [-k arg4] [-v]
     -i File with input data
     -d Column positions for data.
     -e Column positions for data errors.
     -k Sigma-clipping value
8. **SRPCalendar**
   - Its purpose is to convert dates from/tom various formats.
   - SRPCalendar [-h] [-v] -d arg1 / -j arg2 / -m arg3 / -n arg4
     -j Julian Date
     -m Modified Julian Date (MJD)
     -d Regular Date (UT) (yyyy/mm/dd hh:mm:ss)
     -n Present date

9. **SRPChiSqIncrement**
   - Its purpose is to compute increment for chi squares.
   - SRPChiSqIncrement [-a arg1] [-c arg2] -d arg3 [-p arg4] [-v]
     -a is the accuracy of the chisquare increment computation
     -c is the resulting chisquare for a fit
     -d is the number of degrees of freedom
     -p is the probability.

     The routine allows one to compute the increment for the chi square having a probability 100-prob% to occur randomly. Typical usage is for deriving uncertainties for multiparametric fits. Typical usage is for deriving uncertainties for multiparametric fits. Alternatively, one can compute the probability to have randomly a higher chisquare than the one obtained in a fit.
10. **SRPCosmology**
   - Its purpose is to derive cosmological data.
   - SRPCosmology [-h] [-hubbleconstant arg1] [--omegalambda arg2] [--omegamatter arg3] -z arg4 [-v]
     --hubbleconstant Hubble Constant
     --omegamatter Omega Matter
     --omegalambda Omega Lambda
     -z Redshift
11. **SRPDLA**
   - Its purpose is to derive the absorption factor due to DLA systems.
   - SRPDLA -l arg1 [-h] [-v] -n arg2 -z arg3
     -l Observed wavelength (micron)
     -n Nh (>= 0, cm^-2)
     -z DLA system redshift (>= 0)
     DLA modeling performed according to Totani et al. (2006, PASP 58, 485)
12. **SRPDustAbs**
   - Its purpose is to compute the amount of reddening at a given wavelength.
   - SRPDustAbs [-c arg1] -g arg2 [-h] [-r arg3] [-v] -w arg4
     -c E(B-V) color excess (mag)
     -g Kind of extinction curve
     -r Extinction ratio
     -w Wavelength (micron)
     Extinction curves for the MW, LMC and SMC galaxies following Pei (1992, ApJ, 395, 130) and starburst galaxies following Calzetti et al. (2000, ApJ, 533, 682)
13. **SRPEnergyFreqFlux**
   - Its purpose is to convert energy to frequency or wavelength and vice-versa.
   - SRPEnergyFreqFlux [-a arg1 / -j arg1] -e arg2 / -f arg2 / -w arg2 [-h] [-v]
     -a Flux density in Erg / cm s A
     -e Energy (eV)
     -f Frequency (Hz)

-j Flux density in Jy
-w Wavelength (micron)

14. **SRPFit**
    - Its purpose is to carry out multi-parametric fits and Montecarlo error search.
    - SRPFit -d arg1 [-e arg2] -f arg3 -g 'arg4' [-h] [-i 'arg5'] -m arg6 [-n arg7] [-o arg8] [-v]
        -d Table containing data
        -e ERR Error search and confidence level (i.e 90.0)
        -g Guess values for parameters to fit [i.e. '2.2 15.2']
        -i Min,max values for error search [i.e. '0 4 10 20']
        -m File with function to fit (i.e. myfunc.py)
        -n Number of trial for Montecarlo search (default 1000)
        -o Output error file
        -f Output function file
      Perform a multi-parametric fit allowing error search by means of a Montecarlo run.
15. **SRPFTest**
    - Its purpose is to compute F-statistics.

    SRPFTest [-h] -n chi2 dof -o chi2 dof [-v] [--version]
        -n Chi2 and degrees of freedom
        -o Chi2 and degrees of freedom
16. **SRPGaussDistrib**
    - Its purpose is to generate number following the Gaussian distribution.
    - SRPGaussDistrib -e arg1 [-h] -n arg2 -s arg3 / -a arg 3 arg4
        -e Distribution expectation value
        -n Number of repetitions
        -s Distribution standard deviation
        -a Distribution asymmetric standard deviation (left, right)
17. **SRPGaussProb**
    - Its purpose is to compute probability for Gaussian distributions.
    - SRPGaussProb -s -1/-2 [-v]
        -s Value in sigma units)
        -1 1-tail distribution
        -2 2-tail distribution
18. **SRPGetTabEntry**
    - Its purpose is to find selected objects in a table.

    SRPGetTabEntry [-a] -c arg1 arg2 -C arg3 arg4 [-h] -i arg5 [-j arg6] [-o arg7] -t arg8 [-v]
        -a Angular distance if set, else Cartesian distance
        -c Column coordinate positions for input table (col1 col2)
        -C Object coordinates (coord1 coord2)
        -i Input table
        -j Number of lines tio be skipped
        -o Output table without identified entries

-t Maximum tolerance for object association (same units as for the coordinates)

19. **SRPHistogram**
    - Its purpose is to compute an histogram of input data.
    - SRPHistogram -c arg1 [-h] [-j arg2] -o arg3 -t arg4 [-v]
      -b Bin data [i.e. min max bin_size]
      -c Column for histogram
      -j Number of header lines to skip
      -o Output file
      -t Table containing data to extract
20. **SRPIGM**
    - Its purpose is to derive the absorption factor due to IGM systems.
    - Usage: SRPIGM -b arg1 [-l arg2] [-h] -t arg3 [-v] -x arg4
      -b Lower IGM redshift (>= 0) [default 6.0]
      -l Observed wavelength (micron)
      -t Upper IGM redshift (>= 0)
      -x Neutral hydrogen fraction (0 <= xHI <= 1)
21. **SRPLineProfile**
    - Its purpose is to compute line profile for a specific transition.
    - SRPLineProfile -b [arg1] -l arg2 [-h] [-v] -n arg3 -t arg4 -z arg5
      -b Dumping parameter (>= 0), km s^-1)
      -l Observed wavelength (Angstrom)
      -n Column density (>= 0, cm^-2)
      -t Transition
      -z Redshift (>= 0)

    Line profile computed by Voigt function computation.
22. **SRPMagFlux**
    - Its purpose is to convert magnitudes to/from fluxes.
    - SRPMagFlux -b band -f arg1 arg2 / -m arg1 arg2 / -j arg1 arg2 [-h] [-v]
      -b magnitude/flux band
      -f flux and error (Erg/s/cm2/A)
      -j flux and error (Jy)
      -l more information about adopted zero-points.
      -m magnitude and error
23. **SRPMatch**
    - Its purpose is to find common objects between two tables.
    - SRPMatch [-c arg1 arg2 arg3 arg4] [-h] [-j arg5] -m arg6 [-n arg7] -o arg8 -r arg9 [-s arg10 arg11] -t arg12 [-v]
      -c are the x,y column numbers for the reference table and the matching table
      -j is the number of entries at the beginning of both tables to be skipped.
      -m the matching table
      -n the character to identify comment lines to skip
      -o is the output file
      -r reference table

-s the shift for coordinate match
-t is the maximum tolerance for a positive match.

24. **SRPMatchCoord**
    - Its purpose is to find common objects, with the same angular coordinates, in two tables.
    - SRPMatchCoord [-c arg1 arg2 arg3 arg4] [-h] [-j arg5] -m arg6 [-n arg7] -o arg8 -r arg9 -t arg10 [-v]
      -c are the x,y column numbers for the reference table and the matching table
      -j is the number of entries at the beginning of both tables to be skipped.
      -m the matching table
      -n the character to identify comment lines to skip
      -o is the output file
      -r reference table
      -s the shift for coordinate match
      -t is the maximum tolerance for a positive match.
25. **SRPNhAbs**
    - Its purpose is to compute the amount of absorption at a given energy.
    - SRPNhAbs -e arg1 [-h] -n arg2 [-v]
      -e Energy (KeV, 0.03-10).
      -n Nh (>= 0, cm^-2)
      Based on photoelectric absorption cross sections in Morrison & McCammon (1983, ApJ, 270, 119)
26. **SRPPLFluxDensity**
    - Its purpose is to derive flux density known a power-law spectrum and integrated flux.
    - SRPPLFluxDensity [-d arg1] [-e arg2 arg3] -f arg4 [-h] [-s arg5] [-v]
      -d Output flux density (keV)
      -e Energy limits min max (keV)
      -f Integrated flux (erg/cm2 s)
      -s Spectral slope
27. **SRPQuery**
    - Its purpose is to extract a region from a catalogue.

    SRPQuery -c arg1 arg2 / -f arg3 -C arg4 [-h] [-m arg5] [-o arg6] -r arg7 [-S] [-v]
      -c is to input J200 RA and DEC coordinates
      -C is the acronym of the catalogue to browse
      -f FITS file to be used for coordinate center
      -m Maximum number of entries in output
      -o optional output file
      -r is the search radius in arcmin. The search is carried out in a cone
      -S to have an ouput compatible to be shown with the ESO-skycat package
      For several catalogues you need a working internet connection. Other catalogues are local.

28. **SRPSelect**
    - Its purpose is to allow to create list of frames satisfying some criterion.
    - SRPSelect [-i arg1] -k arg2 -o arg3 [-v]
      -i passes to the scripts the file with the list
          of FITS file and keyword values as created, for instance, by
          SRPClassify.
      -k is the keyword to be searched for.
      -o is the output file with results of the selection.
29. **SRPSessionName**
    - Its purpose is to define a new session name.
    - SRPSessionName [-h] -n arg [-v]
      -n allows one to provide a base prefix for many of the files
      created by other SRP commands.
30. **SRPSolarAbundance**
    - Its purpose is to derive the Solar abundance of various chemical elements.
    - SRPSolarAbundance [-e arg1] [-h] [-v]
      -e is the element to look for, e.g. Fe
      Data are from Asplund et al. (2009, ARA&A, 47, 481)
31. **SRPStokesPol**
    - Its purpose is to convert from polarization to Stokes parameter and viceversa.
    - SRPStokesPol [-h] [-m] [-n n] [-p P eP Theta eTheta Chi eChi]
    - [-s Q eQ U eU V eV] [-v] [--version]
    - -m Compute errors by means of a MonteCarlo run
    - -n Number of trials for error computation
    - -p Normalized polarization parameters
    - -s Normalized Stokes parameters
32. **SRPTabExtract**
    - Its purpose is to extract selected columns from a table.
    - SRPTabExtract -c 'arg1' [-h] [-j arg2] -o arg3 -t arg4 [-v]
        -c Columns for columns [i.e. '2 3 1 2']
        -j Number of header lines to jump
        -o Output file
        -t TABLE Table containing data to extract
33. **SRPVersion**
    - its purpose is to show the running **SRP** version.
    - SRPVersion [-h] [-v]
34. **SRPVisibility**
    - Its purpose is to compute the visibility of a sky object.
    - SRPVisibility -a arg1 arg2 / -o arg3 arg4 [-d arg5 arg6 arg7] [-h] [-l arg8 arg9 / -s arg10] [-t arg11] [-v]
      -a Altazimuthal coordinates (dd.dddd dd.dddd)
          -d Altitude (m), pressure (mBar) and temperature (C) of the
          observing site

-l Coordinate location of observing site (dd:mm:ss or dd.dddd)
-s Observing site
-o Object coordinates (hh:mm:ss dd:mm:ss or hh.ddd dd.ddd)
-t Computation time ('yyy/mm/dd hh:mm:ss')

Bugs, comments, etc.

Of course, as already stated, any contribution from anyone is welcome. In case you find bugs, have improvements to suggest, would like to contribute to the code, etc. Please send an e-mail to Stefano Covino, stefano.covino@brera.inaf.it. We can not promise to take into account all your comments, but we will anyway try to improve the package to meet your needs.

Evolution

- **From 1.0 to 1.1**:
  - Command to convert magnitudes to fluxes and to determine reddening were added. The possibility to create SExtractor files for different instruments was also implemented.
- **From 1.1 to 1.2**:
  - The command to perform a local catalogue query was implemented. More training steps proposed.
- **From 1.2 to 1.3**:
  - Aperture photometry is now reported by **SRPPhotometry**, there is also the possibility to extract subimages from a frame saving the astrometric information.
- **From 1.3 to 1.4**:
  - Bug correction for aperture photometry. A table matching tool was added.
- **From 1.4 to 1.5**:
  - The table match now works by a FFT of the input data. The possibility to computer target sky position is implemented. Some minor correction to help data are provided. It is now possible to provide new zero-point to **SRPPhotometry**. The command **SRPMatchCoord** finds common entries for object with the same angular coordinates in two tables.
- **From 1.5 to 1.6**
  - Better management of coordinate matching for **SRPMatchCoord**. Bias and flat can now be constants in **SRPScienceFrameImaging**. Better coordinate management and output in SRPVisibility. **SRPGaussDistrib** added.
- **From 1.6 to 1.7**
  - Correction to **SRPMatch** to work with updated numarray library. New parameter set for TNG Dolores imaging frame classification. **SRPDao2Sky** added. Improved in final match algorithm in **SRPMatch**. Now it is found the closest companion and not the first within the given tolerance. For **SRPMatch** and **SRPMatchCoord** the sequence of reference and matched tables are followed in the output table too.
- **From 1.7 to 1.8.0**
  - Improvement for **SRPMatch** allowing the possibility to force the amount of the displacement between the two tables. Minor corrections to **SRPDao2Sky**. ASIAGO AFOSC and VLT ISAAC imaging keywords. Move from optik to optparse library. **SRPTabExtract** and **SRPHistogram** added.
- **From 1.8.0 to 1.9.0**
  - **SRPFlatSpectroscopy** and **SRPFit** added. Minor corrections to **SRPMagFlux** and **SRPTabExtract**. New keyword for the TNGDOLORESIMA set. **SRPVisibility** with Sun altitude.
- **From 1.9.0 to 1.9.5**

- Removal of automatic error estimate from **SRPFit** because it is too much time consuming and not fully reliable. Error search is something intrinsically difficult to automatise in general. Larger number of function calls and evaluations are allowed. Better pair association algorithm for **SRPMatchCoord**. Calzetti's extinction law added to **SRPDustAbs**. REM/ROSS photometric parameter set. NTT EMMI, TNG NICS and NOT AFOSC imaging parameter sets added. Additions to VLT FORS spectroscopy keywords. UVOT filters added to **SRPMagFlux**. A better porting to the cygwin UNIX flavour has been obtained. A bug occurring when not existent directories are reported in the PATH has been corrected. New keywords for VLT ISAAC. Different filename output extension for **SRPPhotometry** output filenames if “skycat” format is selected. A correction to the algorithm of **SRPTabExtract** has been applied. Various minor bugs have been corrected.

- **From 1.9.5 to 2.0.0**
  - Constant density ISM and wind afterglow parameters as in Hurley, Sari, Djorgovski (in "Compact X-ray Stellar Sources", 2003). X-ray absorption as in Morrison & McCammon (1983). **SRPNhAbs** command. **SRPAftTypSynchrConst**, **SRPAftTypSynchrFreqWind**, **SRPAftCoolSynchrConst**, **SRPAftCoolSynchrFreqWind**. Data analysis parameter set for the 2.2m Calar Alto telescope with CAFOS and upgrade for the NOT with AFOSC. Coordinates of the NOT and Calar Alto observatories. **SRPMatchCoord** now reads "hh:mm:ss" and "dd:mm:ss" coordinate format too.
- **From 2.0.0 to 2.1.0**
  - Minor bug corrections. Danish with DFOSC, VLT with NACO and TNG with NICS imaging parameters added. User's Manual revised. **SRPCosmology** command added.
- **From 2.1.0 to 2.2.0**
  - Minor bug corrections. UVOT photometry calibration upgraded. Better data reading in case of high background for **SRPDao2Sky**. NTT SofI imaging parameters added. **SRPCosmology** was rewritten. Some improvements to dust absorption by **SRPDustAbs** computation were developed. Zero-points for magnitude to flux conversion by **SRPMagFlux** were upgraded. Conversion from coordinates to pixel is now possible with **SRPWCS2Pixel**. Conversion of the output of the GAIA-Photom package to other formats can be carried out with **SRPGAIA2Sky**.
- **From 2.2.0 to 2.3.0**
  - LBT site coordinates. Better management of jump option in **SRPMatch** and **SRPMatchCoord**. GEMINI-N coordinates added. Corrections to site coordinates applied. **SRPEnergyFreq** and **SRPPLFluxDensity** added.
- **From 2.3.0 to 3.0.0**
  - New filters. VLTFORSIPOL added. SExtractor photometric parameter file for REMIR added. **SRPImageMapping**, **SRPRotoTrasla**, **SRPMyPhotometry** and **SRPREMPhotometry** added.
- **From 3.0.0 to 3.1.0**
  - Improved flexibility of **SRPREMphotometry** and **SRPMyPhotometry** commands. Increased execution velocity for **SRPImageMapping**. **SRPRTAlingImaging** and **SRPAdvAverage** commands added.
- **From 3.1.0 to 3.2.0**
  - Improved rapidity for **SRPAdvAverage**. Minor improvements to **SRPImageMapping** and **SRPRotoTransla**. Better management of objects not in the field of view for **SRPREMPhotometry**. Installation procedure now much better explained. Better parameter management for **SRPEnergyFreq**. **SRPCalendar** added and a few bugs fixed. Better centering algorithm and magnitude computation. **SRPMagFlux** is improved. **SRPDLA** added. **SRPFit** management improved.
- **From 3.2.0 to 3.3.0**
  - **SRPIGM** added. Minor improvements to **SRPVisibility**, **SRPEnergyFreq**, **SRPGaussDistrib**, **SRPQuery**, **SRPMyPhotometry** and **SRPREMPhotometry**. More options for **SRPImageMapping**. Better algorithm for **SRPRTAlignImaging**.

- **From 3.3.0 to 3.4.0**
  - Unicode strings in **SRPWCS2Pixel**. Simplified algorithm for **SRPMatch**. Bug correction in **SRPMagFlux**. Upgrade for **python 2.6** and later versions. Various minor upgrade and bug corrections. **SRPGaussProb** added. *X-shooter* parameters added. Minor correction to **SRPVisibility**.
- **From 3.4.0 to 3.5.0**
  - Better sky and zero-point computation with **SRPMyPhotometry**. Bug correction in **SRPAlignImaging**. Possibility to force integer shifts for pure translation and to provide maximum tolerance in **SRPImageMapping**. **SRPAirVacuum**, **SRPGaussProb**, **SRPSourceFinder** and **SRPFindingChart** added. Improvements to **SRPGaussDistrib**.
- **From 3.5.0 to 3.5.1**:
  - Minor bug correction in **SRPAlignImaging**. **2MASS** catalogue and more functionalities added to **SRPQuery**. TNG Dolores spectroscopy parameters added. Exposure maps for **SRPAdvAverage**.
- **From 3.5.1 to 3.6.0**:
  - Sorted output and various improvements to **SRPSourceFinder**. A few minor bugs corrected. Deepness of search selectable in **SRPImageMapping**. **SRPAstrometry** was added. **TNG-LRS** SExtractor parameter files added. **SRPPhotParSet** improved. **SRPQuery** improved and catalogues of Stetson optical standard stars, Astro-wise standard stars and USNO-A2 added. **SRPZeroPoint** added. **nose** python library added to the installation list. **SRPAverSigmaClipping** added. **asciitable** and **ATpy** now required for installation. **SRPTNGPipelineManager** added. **SRPFitsStats** added. **SRPBias**, **SRPFlatImaging**, **SRPScienceFramesImaging** and **SRPCut** improved.
- **From 3.6.0 to 3.7.0**:
  - Better management of cut area for frame binning in **SRPTNGPipelineManager**. New keywords for **SRPTNGManager** and data saved as integer. **SRPGetTabEntry** added. Better magnitude difference averaging algorithm for **SRPZeroPoint**. **SRPREMPipelineManager** added.
- **From 3.7.0 to 3.8.0:**
  - Position on the detector of the selected object in **SRPREMPipelineManager**. Maximum number of log files in SRP pipeline managers. Collection of NIR catalogues (Arnica, Conica, ESO, Isaac, LCO, MSSSO, SAAO, UKIRT) added to **SRPQuery**. Present time in **SRPCalendar**. Bug correction in **SRPFindingChart**. A few bugs corrected in **SRPMyPhotometry**. **SRPPix2WCS** added. Minor bug corrections for **SRPScienceFramesImaging** and **SRPAstrometry**. Improvement to **SRPCosmology**. **SRPAftSynchrSpectrumConst** and **SRPAftSynchrSpectrumWind** rationalized.
- **From 3.8.0 to 3.9.0:**
  - New filters in **SRPMagFlux**. Better output for GRBs in **SRPREMPipelineManager**. Better Voigt profile in **SRPLineProfile**. Better coding for **SRPDustAbs**. Better check for user identity in **SRP** pipelines. AGN optical standard stars catalogue added for **SRPQuery**. Improved AGN photometry for **SRPREMPipelineManager**. Better effective wavelengths for several filters in **SRPMagFlux**. Better frame downloading for **SRPREMPipelineManager**. Better management of file download in **SRPREMPipelineManager**. **SRPWCSPixel** simplified. **SRPAverage** improved. **SRPFitsHeader** added.
- **From 3.9.0 to 3.10.0:**
  - Bug correction and new parameter in **SRPMatch**, **SRPMatchCoord** and **SRPKeyword**. New filters in **SRPMagFlux**. **SRPChiSqIncrement** added. New filter data added to **SRPMagFlux**. **SRPSolarAbundance** added. **SRPImageFilter** added. Better parameter management in **SRPBias** and **SRPFlatImaging**. Bug correction for frame weighted sigma-clipped average. Bug corrected in **SRPRotoTransla**. Bug in absolute path file opening corrected. Improvement of **SRPTNGPipelineManager**. **SRPVersion** added. **SRPFitsExtension** added. Possibility to use a Fits file as a reference in **SRPQuery**. Logic and computation corrections to **SRPIGM**.

- **From 3.10.0 to 3.10.14:**
  - Improvements for **SRPFitsHeader**. New bands added to **SRPMagFlux**. **SRPFitsSpectrum2ASCII** added. Bug correction in **SRPAdvAverage**. **SRPEnergyFreq** converted to **SRPEnergyFreqFlux**. *Ellipticity* parameter added to **SRPPhotometry** output. New and better organized zero-points for **SRPMagFlux**. More flexible coordinate format for **SRPAstrometry**. Bugs in **SRPAlignImaging** corrected. **SRPMyPhotometry** and **SRPREMPhotometry** moved to **SRP.REM** sub-package. GAIN in sextractor files for REM-REMIR and REM-ROSS. Better management of data file path. More information (sidereal time, hour and parallactic angles) added to **SRPVisibility**. Possibility to read extensions in **SRPFitsHeaders**. **SRPStokesPol** added. Parameters for TNG Dolores spectroscopy and polarimetry added. Minor bugs and more options in **SRPDAO2Sky** and **SRPGAIA2Sky**. **SRPFitsExtensions** is now able to manage FITS cubes. Minor bug in and new options in **SRPAstrometry**. Standard deviation in **SRPAverSigmaClipping** output. New optical data for the AGNOPT catalogue. **SRPAtmExtinction** added. Improvements to **SRPVisibility**. Better algorithm for frame sigma-clipping average. Improved keywords for **SRPClassify**. New option in **SRPGetTabEntry**. Better output for **SRPQuery**. **SRPSpectralExtraction** added. Default parameters for **SRPCosmology** updated. **SRPStokesPol** improved. SDSS catalogue added to **SRPQuery**. Bug correction in **SRPFlatSpectroscopy**. **SRPFTest** added. APASS catalogue added to **SRPQuery**. **SRPGetTabEntries** reports not selected objects too. Minor corrections and improvements for **SRPAstrometry** in shift computation. Better parameter management for **SRPPhotometry**. More control in **SRPQuery** output. Bug correction and more parameters in **SRPAstrometry**. More flexibility for dust extinction **SRPDustAbs**. **SRPFitsComposer** has been added. ez_setup.py added to the distribution.
- **From 3.10.14 to 3.11.00:**
  - Creation of the **SRP.FITS** sub-package for a better separation of duties.

Credits, thanks, etc.

A lot of people gave some contribution to the **SRP** and among them I want to quote Nino Cucchiara, Paolo D'Avanzo, Luca Di Fabrizio, Dino Fugazza, Auvet Harutyunyan, Nauzet Hernandez, Domenico Impiombato, Gianluca Israel, Daniele Malesani, Emilio Molinari and Ruben Salvaterra. I also thank ESO and TNG since part of this code was developed during visitorships in Garching and La Palma.