Writing Software Which Will Continue to Work

Jessica Mink

Smithsonian Astrophysical Observatory, Cambridge, MA, USA
jmink@cfa.harvard.edu

Abstract. In the case of files of images, spectra, and object catalogs, there are lots of formatting, mapping, and translation problems which can be solved with reliable software that relatively few people can write. The RVSAO package of spectral tools which include redshifts in IRAF SPP and WCSTools package of image and world coordinate system tools in C have been in use for such tasks since 1989 and 1994 respectively. It has come time to remove the dependency of RVSAO on a local installation of the much larger IRAF system. The programming and user interface questions to be dealt with in that translation to RVTools are discussed.

1. The Problem

Alongside any effort being made to create well-documented archives of astronomical data, there needs to be a parallel effort to created well-documented software to enable use of that data into the future. In discussions with people outside of astronomy, the lifetime required for software to be considered long-lived seems to be around five years. The ongoing archive of astronomical spectra from the CfA FAST Spectrograph (Mink et al. 2021) , which turns thirty in January 2024, has been reduced and analyzed using IRAF (Tody 1986) and its useful spectrum reduction and analysis software libraries (Valdes 1986) using RVSAO (Kurtz et al. 1992) (Kurtz & Mink 1998) (Kurtz & Mink 1999) (Mink 2013) for the entire time.

As a precedent for a standalone software package, WCSTools (Mink 1996) (Mink 2011) (Mink 2019) has been used on data taken in the nineteenth century, preserved on photographic plates, and scanned in the twenty-first century (Mink et al. 2006) (Grindlay et al. 2009) (Laycock et al. 2010). No current software is as old as that photographic data!

With some experience keeping two widely-used data access and analysis software packages in use for several decades each, I have found several ways to make packages portable, user-installable, and easily-repairable. These are not the only way to do this, but with software involving special knowledge of particular astronomical data types, more detailed expertise is required than most astronomers and astrophysicists are likely to have. As IRAF is falling out of general use while the RVSAO package is still being used, there is a need for a replacement package that can run on its own like WCSTools does.

2. The Choices

2.1. What is good about IRAF

IRAF has been around for 40 years, and the author has been maintaining a local installation and writing code in IRAF's SSP preprocessed Fortran language at the Center for Astrophysics for 39 of those years. RVSAO, a package of tasks which compute radial velocities of object spectra from solar system objects to distant galaxies using cross-correlation against template spectra or fitting lists of emission lines, started with a simpler package in 1989 and has been developed and maintained within the IRAF environment ever since. It has run with relatively few changes on a variety of computers under a variety of mostly Unix-based operating systems. Its libraries can read, extract, and calibrate spectra from many different instruments, and with some skill, it is possible to run IRAF tasks from a Unix/Linux command line using IRAF's CL/ECL scripting language, making them usable in any kind of shell script. Many programs have more parameters that can be modified than will fit comfortably on a command line. Every task has its own parameter files which can be saved and modified by the user for re-use.

2.2. What is bad about IRAF

Now that most scientists work on their own computers, they all-too-often need to set up and manage their own IRAF installations just to use this single package. Even with decades of personal experience maintaining an IRAF installation for an entire research institution, that can be time-consuming and not necessarily easy. Each operating system update may need a new version, and new version may change APIs of subroutines used. The system is huge, and libraries are deep, relatively undocumented, and hard to debug, though over time with lots of users, they have gotten better. Bugs can affect both SPP and the Fortran code it creates, so they are often quite hard to find. And last, but not least, there are various limitations on subroutines caused by the preprocessing which can be troublesome.

2.3. Some other choices

Python has good astronomical library support, but not yet for spectra. Attempts have been made, but support has come and gone. I would have to learn it, which I have not done. Python and its libraries change occasionallly, there are different versions of the same functions in different libraries with different characteristics, and nothing is as permanent as C.

IDL has been around for longer than IRAF and has some good astronomical libraries available. while some of the libraries are open source, they are dependent on the commercial system which has a continuing cost and may not be available in the distant future.

Newer languages such as Julia look good and are starting to gain some of the basic libraries. but they are not yet ready to last for decades.

2.4. What is bad about Unix/C

Fewer people can program in C in the twenty-first century. A C compiler needs to be available to create an usable copy of the software, and it is somewhat difficult to make sure that the code works on every version of Unix/Linux. Not every useful external library is available in C.

2.5. What is good about Unix/C

Both Unix and C have been around for a long time, and C and its variants are still among the most popular programming languages. WCSTools has been running for almost 30 years on Unix systems–longer if you count its Fortran version (McCord et al. 1977) (Mink & Klemola 1982) running on IBM and Data General systems.

Open source libraries are available for most things and are compact enough to be included. Many time-tested subroutines for dealing with astronomical problems such as formatted data, and metadate, parameters, arguments, astrometry, and catalogs are already available in WCSTools and elsewhere. Tasks can be written so as to be easy to script with any UNIX shell.

3. From IRAF to C

Having decided on a specific path, the next choice was what should be included from IRAF, what should be added from IRAF, what should be included from WCSTools, and what needs to be written more than translated. Simple linearized spectra are easy and can be read from FITS files by the WCSTools library. More complicated dispersion functions will require more code translation, though some of the necessary C code has been written for WCSTools to read IRAF-generated 2-dimensional] world coordinate systems. Most of the spectrum processing code will be rewritten from the RVSAO SPP code, which is well-understood by the author. One of first translated features was parameter files. Examples of parameter files for the linespec emission line spectrum generator in IRAF and RVTools follow.

3.1. IRAF rvsao.linespec Parameter File

Data type information is included in the parameter file.

```
linefile,s,a,"",,,"Filename with list of line centers"
linedir,s,h,"",,,"Directory for line list"
linewidth,r,h,2.,0.,25.,"HWHM of instrument resolution"
maxwidth,b,h,no,,,"Use line max or instrument width (yes or no)"
zspec,r,h,0.,,,"Shift line centers to this Z not velspec"
. . .
spec_int,b,h,no,,,"True to interact with the created spectrum"
device,s,h,"stdgraph",,,"Display device"
plotter,s,h,"stdplot",,,"Hardcopy output device"
verbose,s,h,"yes",,,"Print summary to log file (yes or not)"
logfiles,s,h,"STDOUT,linespec.log",,,"List of log files"
debug,b,h,no,,,"Displays intermediate results"
cursor,*gcur,h,"",,,"Graphics cursor input"
mode,s,h,"ql",,,
```

3.2. RVTools linespec Parameter File

Version information is included in the parameter file.

```
RVTools.linespec 0.1 (2023-08-18) run 2023-Nov-01T13:35:25.000
# Parameters for LINESPEC task of RVTools Package
specname = 'Galaxy Emission Lines' / Object name for created spectrum
specfile = 'emtemp.fits'  / Filename for created spectrum
```

```
linefile = emtemp.dat / Filename with list of lines and widths
linedir = '/home/mink/WCS/rvtools/lib'  / Directory for line list
linewidth = 2.0    / HWHM of instrument resolution in Angstroms
maxwidth = 0 / If =1 use line max else convolve instrument width
zspec = 0.0  / Shift line centers to this Z not velspec
. . .
spec_int = 0    / 1 to interact with the created spectrum, else 0
verbose = 1     / Print summary to log file (yes or not)
debug = 1       / Displays intermediate results
end
```

Keeping the long lists of parameters that each task needs seemed best replicated through the use of parameter files with a simple enough structure that the contents can be passed intact between subroutines and set up with default values if not appropriate file exists in the working directory or is included on the command line. I have used a FITS header-like structure for parameters for an image processing pipeline in the past and much of the needed code to extract values from such a text string is already included in the WCSTools library. Here are the IRAF and RVTools parameter files for a task which creates an emission line spectrum from a file with a list of emission lines.

References

Grindlay, J., Tang, S., Simcoe, R., Laycock, S., Los, E., Mink, D., Doane, A., & Champine, G. 2009, in Preserving Astronomy's Photographic Legacy: Current State and the Future of North American Astronomical Plates, edited by W. Osborn, & L. Robbins, vol. 410 of Astronomical Society of the Pacific Conference Series, 101

Kurtz, M. J., & Mink, D. J. 1998, PASP, 110, 934. astro-ph/9803252

Kurtz, M. J., Mink, D. J., Wyatt, W. F., Fabricant, D. G., Torres, G., Kriss, G. A., & Tonry, J. L. 1992, in Astronomical Data Analysis Software and Systems I, edited by D. M. Worrall, C. Biemesderfer, & J. Barnes, vol. 25 of Astronomical Society of the Pacific Conference Series, 432

Kurtz, M. J., & Mink, J. 1999, RVSAO 2.0: Digital Redshifts and Radial Velocities. 9912.003

Laycock, S., Tang, S., Grindlay, J., Los, E., Simcoe, R., & Mink, D. 2010, AJ, 140, 1062

McCord, T. B., Huguenin, R. L., Mink, D., & Pieters, C. 1977, Icarus, 31, 25

Mink, D., Doane, A., Simcoe, R., Los, E., & Grindlay, J. 2006, in Virtual Observatory: Plate Content Digitization, Archive Mining and Image Sequence Processing, edited by M. Tsvetkov, V. Golev, F. Murtagh, & R. Molina, 54

Mink, D. J. 1996, in Astronomical Data Analysis Software and Systems V, edited by G. H. Jacoby, & J. Barnes, vol. 101 of Astronomical Society of the Pacific Conference Series, 96

— 2011, WCSTools: Image Astrometry Toolkit. Astrophysics Source Code Library, 1109.015

Mink, D. J., & Klemola, A. 1982, AJ, 87, 1881

Mink, J. 2013, in Astronomical Data Analysis Software and Systems XXII, edited by D. N. Friedel, vol. 475 of Astronomical Society of the Pacific Conference Series, 291

— 2019, in Astronomical Society of the Pacific Conference Series, edited by P. J. Teuben, M. W. Pound, B. A. Thomas, & E. M. Warner, vol. 523 of Astronomical Society of the Pacific Conference Series, 281

Mink, J., Brown, W. R., Chilingarian, I. V., Fabricant, D., Kurtz, M. J., Moran, S., Rhee, J., Tokarz, S., & Wyatt, W. F. 2021, AJ, 161, 3. 2012.03470

Tody, D. 1986, in Instrumentation in astronomy VI, edited by D. L. Crawford, vol. 627 of Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, 733

Valdes, F. 1986, in Instrumentation in astronomy VI, edited by D. L. Crawford, vol. 627 of Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, 749