Writing Software Which Will Continue to Work

Jessica Mink

Smithsonian Astrophysical Observatory

RVSAO Programs

(from IRAF help rvsao)

rootrvsao.rvsao:

emsao - Find redshifts by measuring emission line shifts

pemsao - Fit an emission line in a spectrum, results to parameters

xcsao - Find redshifts by cross-correlating spectra with templates

pxcsao - Cross-correlate spectra with template, results to parameters

eqwidth - Compute equivalent widths for specified lines in spectra

contpars - Set parameters for continuum removal

sumspec - Combine multiple spectra into one spectrum

contsum - Set continuum removal parameters for template making

linespec - Make a spectrum from a list of emission / absorption lines

listspec - Print an ASCII list of a spectrum with optional per pixel info

bcvcorr - Compute radial velocity correction to solar system barycenter

emplot - Plot a spectrum with emission velocity and labelled lines

pix2wl - Compute the wavelength for a specific pixel in a spectrum

qplot - Check results of previously-run xcsao and emsao

qplotc - Check results of previously-run xcsao and emsao (-continuum and bad lines)

relearn - Set parameter values in parameter file of new version of a task using the values in the current parameter file

setvel - Shift a single spectrum to a several radial velocities

skyplot - Plot a night sky spectrum with labelled emission lines

velset - Set velocity by modifying starting wavelength

What was good about IRAF

• It was widely available and supported by a staff of programmers at NOAO and later STScI.

• There are vast subroutine libraries which worked on most computers (though never Windows), and packages and parameter files made it easy to configure.

• The CL, then NCL and ECL, scripting language was tunded to astronomical uses.

• RVSAO has already been running for almost 35 years on systems around the world.

What is bad about IRAF

• Now that most scientists work on their own computers, they might need to set up an manage their own IRAF installations just to use this single package, which is not necessarily easy.

• Each operating system update may need a new version, and new version may change API's of subroutines used.

•The system is huge, and libraries are deep, relatively undocumented, and hard to debug.

Rejected options for implementation

Python
• It has good library support, but maybe not for spectra.

• I'd have to learn it, which I haven't done.

• It and its libraries change occasionallly, and nothing is as permanent as C.

IDL
• It's commercial and depends on non-open-source code.

Newer languages such as Julia
• Not ready to last for decades

After having some experience keeping two widely-used data access and analysis software ages 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 specinowledge of particular astronomical data types, more detailed expertise is required than most astronomers and astrophysicists are likely to have.

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. RVSAO in IRAF SPP and WCSTools in C have been doing more and more of that since 1989 and 1994 respectively.

It has come time to translate the RVSAO spectral redshift package out of IRAF, so the programming and user interface questions needed for that translation to RVTools are discussed.

RVTools Programs

(from the package command rvtools)

RVTools 0.1 Spectrum and Radial Velocity/Redshift Tools

(Jessica Mink, SAO Telescope Data Center, October 31, 2022)

bcvcorr: Compute radial velocity correction to solar system barycenter

contpars: Set parameters for continuum removal

contsum: Set continuum removal parameters for template making

emplot: Plot a spectrum with emission velocity and labelled lines

Emsao: Find redshifts by measuring emission line shifts

eqwidth: Compute equivalent widths for specified lines in spectra

linespec: Make a spectrum from a list of emission / absorption lines

Listspec: Print an ASCII list of a spectrum with optional per pixel info

pemsao: Fit an emission line in a spectrum results to parameters

pix2wl: Compute the wavelength for a specific pixel in a spectrum

Pxcsao: Cross-correlate spectra with templates, results as parameters

qplot: Check results of previously-run xcsao and emsao

qplotc: Check results of xcsao and emsao (-continuum and bad lines)

setvel: Shift a single spectrum to a several radial velocities

skyplot: Plot a night sky spectrum with labelled emission lines

sumspec: Combine multiple spectra into spectrumat one redshift

velset: Set velocity by modifying starting wavelength

wl2pix: Compute the pixel for a specific wavelength in a spectrum

wlrange: Find the overlapping wavelength range in a list of spectra

xcplot: Plot a spectrum with xc velocity and labelled lines

xcsao: Find redshifts by cross-correlating spectra with templates

zvel: Run emsao and xcsao on list of spectra

* <program name> help lists possible arguments

* <program name> version lists version of program

What I am doing

Since I already have experience keeping specialized code alive for decades, I am putting the lessons I have learned in writing and maintaing WCSTools to work.

• Rewriting IRAF SPP code into C for all of RVSAO and spectrum data formats. (I'm about ½ done)

• Implementing parameter files, with the defaults in the source code so that no external parameter files are needed to start, but not all parameters need to be included on the command line as in WCSTools.

• Using FITS as the default standard so that the WCSTools library can be used for data access.

• Using CSH for scripted tasks because it doesn't change. WCSTools hasseveral small tasks useful for such scripting

---

WCSTools Programs

### (from the package command *wcstools*)

WCSTools 3.9.6 Programs

addpix: Add a constant value(s) to specified pixel(s)

bincat: Bin a catalog into a FITS image in flux or number

char2sp: Replace this character with spaces in output (default=_)

conpix: Operate on all of the pixels of an image

cphead: Copy keyword values between FITS or IRAF images

crlf: Change CR's to newlines in text file (for imwcs, imstar logs)

delhead: Delete specified keywords from FITS or IRAF image file headers

delwcs: Delete the WCS keywords from an image

filename: Drop directory from pathname, returning just the file name

filedir: Drop filename from path name, returning directory path

fileroot: Drop file name extension, returning path name without it

edhead: Edit the header of a FITS or IRAF file

getcol: Extract specified fields from an space-separated ASCII table file

getdate: Convert between two date formats

getfits: Extract portion of a FITS file into a new FITS file, preserving WCS

gethead: Return values for keyword(s) specified after filename

getpix: Return value(s) of specified pixel(s)

gettab: Extract values from tab table data base files

httpget: Send contents returned from URL to standard output

i2f: Read two-dimensional IRAF image file and write FITS image file

imcat: List catalog sources in the area of the sky covered by an image.

imextract: Extract 1D file from 2D file or 2D file from 3D file

imfill: Replace bad pixels in image files with 2-D Gaussian, mean, or median

imhead: Print FITS or IRAF header

immatch: Match catalog and image stars using the WCS in the image file.

imrot: Rotate and/or reflect FITS or IRAF image files

imresize: Block sum or average a file by integral numbers of columns and rows

imsize: Print center and size of image using WCS keywords in header

imsmooth: Filter FITS and IRAF image files with 2-D Gaussian, mean, or median

imstack Stack 1-dimensional images into a 2-dimensional image

imstar: Find and list stars in an IRAF or FITS image

imwcs: Match FITS or IRAF image stars to catalog stars and fit a WCS

isfits: Return 1 if argument is a FITS file, else 0

isnum: Return 1 if argument is an integer, 2 if it is floating point, else 0

isrange: Return 1 if argument is a range of the format n1[-n2[xs]],...

keyhead: Change keyword names in headers of FITS or IRAF images

newfits: Create blank FITS files (dataless by default with BITPIX=0)

remap Rebin an image from its current WCS to a new one

scat: Search a source catalog given a region on the sky

sethead: Set header keyword values in FITS or IRAF images

setpix: Set specified pixel(s) to specified value(s)

simpos: Return RA and Dec for object name(s) from SIMBAD, NED, VizieR

sky2xy: Print image pixel coordinates for given sky coordinates

skycoor: Convert between J2000, B1950, galactic, and ecliptic coordinates

sp2char: Replaces space in string with specified character (default=_)

subpix: Subtract a constant value(s) from specified pixel(s)

sumpix Total pixel values in row, column, or specified area

wcshead: Print basic world coordinate system information for images

xy2sky: Print sky coordinates for given image pixel coordinates

* <program name> help lists possible arguments

* <program name> version lists version of program

What is good about Unix/C

• Both Unix and C have been around for a long time. WCSTools is running for almost 30 years on systems all over the world.

• Open source libraries are available for most things and are compact enough to be included.

• I have lots of time-tested subroutines for dealing with astronomical problems such as formatted data, parameters, arguments, astrometry, and catalogs.

• Easy to script with any UNIX shell

What is bad about Unix/C

• Fewer people can program in C these days

• Package needs to be compiled, and it's hard to make sure that it works on every version of Unix/Linux

• Many useful libraries are now not in C
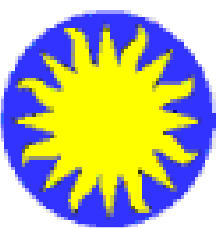
Jessica Mink
jmink@cfa.harvard.edu
http://tdc-www.harvard.edu/mink/
@MinkJessica

Telescope Data Center
SMITHSONIAN ASTROPHYSICAL OBSERVATORY

WCSTools   Jessica Mink