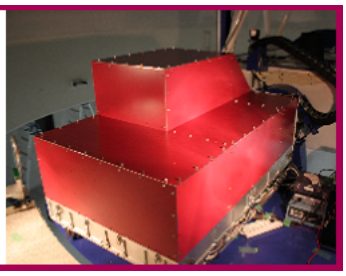

WARP

WINERED Automatic Reduction Pipeline

Version 1.1

WINERED team

winered-contact@cc.kyoto-su.ac.jp

(H. Kato, S. Hamano, N. Matsunaga et al.)

May 22, 2023

Contents

1 Introduction

WARP (WINERED Automatic Reduction Pipeline) is the pipeline software, written in Python 3, for reducing WINERED data. It is publicly available at the WARP home on github[1].

2 Installation

2.1 Necessary environment

- Python 3 (ver 3.6 or later)
- Python libraries — numpy, matplotlib, PIL, astropy
- PyRAF

2.2 How to install

Download the WARP as a zip file or using the 'git clone' command. Let us suppose we download the zip file, `WARP-main.zip`, to the `/usr/local/src` directory, although this directory can be anywhere else.

```
$ cd /usr/local/src
$ ls
WARP-main.zip
$ unzip WARP-main.zip
```

In the `WARP-main` directory, you can find WARP python scripts, `README.md` (which is not very informative yet), etc, together with the test scripts `testWarpSci.sh` and `testWarpSciFull.sh`. If necessary python libraries and tools are already available in your environment, the next and final step for the installation is to run a test script. Before doing so, check if the script knows the command name (or alias) you use for running Python; edit the `PYTHON` command (`python`, `python3`, or something else) according to your environment.

```
$ ls WARP-main/TEST/WIDE
4_Ari_list.txt   WINERED_calibration_LCO22b_WIDE100_20220914_v2
```

[1] https://github.com/SatoshiHamano/WARP

```
WINA00036701.fits WSVA00005637.fits
WINA00036702.fits paramSample.txt
WINA00036703.fits paramSample.txt~
$ vi testWarpSci.sh
$ sh testWarpSci.sh
```

If this test goes well, there should appear a new directory, TEST/4_Ari_WIDE_test, containing the report PDF (4_Ari_2022-09-15.pdf) and various data including 1D spectra of individual exposures and combined spectra.

3 How to use

3.1 Warp_sci.py

Warp_sci.py is the main script to call for making the pipeline reduction of a data set for a scientific object. You can fine basic explanations by calling it with the -h option. In the following examples, call an appropriate Python verion (python, python3, or anything else according to your environment).

```
$ python Warp_sci.py -h
#####
usage: Warp_sci.py [-h] [-r RAWDATAPATH] [-v VIEWERPATH]
                   [-c CALIBPATH] [-d DESTPATH] [-q] [-s]
                 [-p PARAMETERFILE] [-o] [-f] [-a] [--noreport]
                   listfile

positional arguments:
  listfile             fits file list to be reduced

optional arguments:
  -h, --help           show this help message and exit
  -r RAWDATAPATH, --rawdatapath RAWDATAPATH
                       directory path of input raw data
  -v VIEWERPATH, --viewerpath VIEWERPATH
                       directory path of slit vierer data
  -c CALIBPATH, --calibpath CALIBPATH
                       directory path of calibration data
  -d DESTPATH, --destpath DESTPATH
                       destination directory path
```

```
  -q, --query       query mode for setting pipeline parameters
  -s, --save         save all data
  -p PARAMETERFILE, --parameterfile PARAMETERFILE
                     pipeline parameter file
  -o, --oldformat    old (-ver3.6) input list format
  -f, --fastMode    Run WARP with the fast mode. (CR detection,
                     wavelength shift are skipped.)
  -a, --autoCalib       Choose the appropriate calib data
                        automatically.
  --noreport            Not generate a pdf report.
#####
```

3.2 Step 1 — Prepare of raw data and calibration data

Let's check the TEST/WIDE directory which was used for the test reduction made in the installation mentioned above.

```
$ ls TEST/WIDE/
WINA00036701.fits # Raw data
WINA00036702.fits # Raw data
WINA00036703.fits # Raw data
WINERED_calibration_LCO22b_WIDE100_20220914_v2/ # Calibration data
WSVA00005637.fits # Slitviewer data
```

The three WINA fits files are raw data, while the WSVA fits file is an image taken by the slit viewer and it will be used for the report PDF. The directory for the calibration data contains fits files for flat and comparison lamp together with the database directory to be used for the PyRAF tasks. The calibration data shall be made publicly available after each observing run (to be confirmed).

3.3 Step 2 — Prepare the list of target frames

Each set of frames to be analyzed together requires a text file listing pairs of object and sky frames. The following example lists two pairs with the same frame as the sky because they are taken with the OSO mode.

```
$ cat /TEST/WIDE/4_Ari_list.txt
WINA00036701 WINA00036702
WINA00036703 WINA00036702
```

Each line should include the frame number (not with ".fits") of the object data (column 1) and that of the sky data (column 2) separated by spaces or tabs as the delimiter.

The script `make_rawdata_list.py` can be used for creating such target lists automatically checking raw data frames included in a given directory:

```
$ python WARP/make_rawdata_list.py ./rawdata/
```

This will create a text file for each set of consecutive exposures with the same target name (under the same instrument mode), and the file name looks, e.g., like `HD_139129_1_WIDE100_list.txt` in which the "1" preceding the mode and slit width ("`WIDE100`") indicates the sequential number for this object found in this directory. Even if the same object is observed with the same instrument mode, frames taken after a long time gap (7200 sec by default, but can be changed with the -t option) will be grouped separately.

3.4 Step 3 — Run the pipeline

```
$ python Warp_sci.py ./TEST/WIDE/4_Ari_list.txt -r ./TEST/WIDE/
  -c ./TEST/WIDE/WINERED_calibration_LCO22b_WIDE100_20220914_v2/
   -v ./TEST/WIDE/ -d ./TEST/4_Ari_WIDE_test
```

The Warp_sci.py can be run with slitviewer images but the report PDF would lack the presentation of slitviewer images. You can use the `-a` option instead of identifying the calibration data with the `-c` option, as long as the appropriate calibration data exists in the directory identified by the `-c` option. The setting (run and mode) of a given set of frames is taken from the fits header, and the script determines a directory of calibration data to use. The pipeline would fail if there are no directory or more-than-one directories for a given setting.

3.5 Step 4 — Examine the outputs

The pipeline would produce a directory with output files for each set of frames.

```
./TEST/4_Ari_WIDE_test
|- 4_Ari_list.txt       # List of frames
|- 4_Ari_2022_09_15.pdf # WINERED pipeline reduction log
|- 4_Ari_NO# # Spectrum data of each frame
|- onedspec  # 1D spectrum data
|- twodspec  # 2D spectrum data
|- 4_Ari_sum # 1D spectrum data of summed all frame
|- AIR_flux  # Flux spectrum data in Air wavelength
     |- fsr1.05  # Spectrum data of 1.05 times free spectral range
     |- fsr1.30  # Spectrum data of 1.30 times free spectral range
|- AIR_norm  # Normalized Flux spectrum data in Air wavelength
|- VAC_flux  # Flux spectrum data in Vacuum wavelength
|- VAC_norm  # Normalized Flux spectrum data in Vacuum wavelength
|- SNR_dat   # Signal-to-Noise ratio data
|- calibrationdata  # calibration data
|- rawdata_image    # raw image data
|- reduction_log    # reduction logs
|- slit_viewer      # slit viewer data
|- spectra_image    # spectrum plot
```

3.6 Parameters for the reduction

Parameter	Description	Default	Option(s)
Apscatter	To remove the scattering background	yes	[yes, no]
Manual Aperture	Adopt aperture ranges read from the input file?	no	[yes, no]
Background Subtraction	Subtract background spectra from object spectra?	no	[yes, no]
Cosmic Ray Correction	Detect and interpolate the cosmic rays?	yes	[yes, no]
Cosmic ray threshold sigma	The threshold of cosmic ray detection.	10.0	float value
Cosmic ray maximum sigma	Maximum threshold of cosmic ray detection.	20.0	float value
Cosmic ray Var/Ave ratio	The threshold of variance/average ratio of cosmic ray distribution to prevent the false detection.	2.0	float value
Cosmic ray ratio between slit positions	The threshold of the cosmic ray ratio between the slit positions to prevent the false detection.	1.5	float value
Cosmic ray fix sigma	Fix the threshold of cosmic ray detection?	no	[yes, no]
Set cut range	Wavelength range to extract (in the factor of the free spectral range)	1.05, 1.3	List of float values
Sky Emission	To extract the sky spectrum	no	[yes, no]
Measure Shift	Measure the spectra offsets among multiple frames?	yes	[yes, no]
Correct Shift	Correct the spectra offsets among multiple frames?	yes	[yes, no]
Manual Shift	Use the spectra offsets values written in list file?	no	[yes, no]
CUTRANSFORM flux	The flux option in IRAF(PyRAF) transform task.	no	[yes, no]

3.7 Changing the analysis parameters from default

There are two ways of making the pipeline reduction with changing the analysis parameters: (1) give the analysis parameters in a text file with the `-p` option, and (2) calling the `Warp_sci.py` script with the `-q` option for the intercive mode.

The aperture and background ranges and the spectrum shift values can follow the frame numbers of object and sky in the target list. The plots of aperture cut like Figure 1 are useful for determining these ranges, and such plots can be created with the pipeline reduction with default parameters.

```
example :
TEST/WIDE/4_Ari_list.txt
 WINA00036701 WINA00036702 ap=-7:3 bg=-22:-12,8:18 ws=0.
 WINA00036703 WINA00036702 ap=-7:3 bg=-22:-12,8:18 ws=0.
# Column 3-(in no particular order)
# ap : set aperture range in pix(from ApStart to ApEnd)
# bg : set background region in pix(from BgStart to BgEnd).
       Multiple regions can be listed, separated by commas.
# ws : set wavelength shift in pix
```

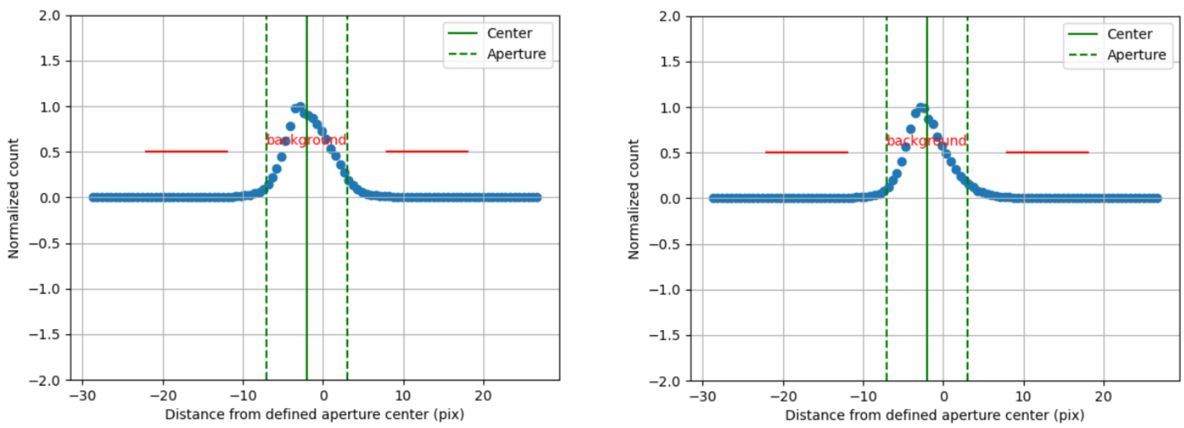

Figure 1: Aperture cuts for `4_Ari`, NO1 on the left and NO2 on the right. Such plots are available in the directories named `4_Ari_NO?/images/spatial_profile/`.

(1) The analysis parameters can be given in a text file. The `Warp_sci.py` script reads the file given with the `-p` option to get specified parameters. A sample of such a parameter file is given in the `Warp-main` directory.

```
$ cat sampleParameter.txt
Apscatter: yes
Manual Aperture: yes
Background Subtraction: yes
Cosmic Ray Correction: yes
Cosmic ray threshold sigma: 10.0
Cosmic ray maximum sigma: 20.0
Cosmic ray Var/Ave ratio: 2.0
Cosmic ray ratio between slit positions: 1.5
Cosmic ray fix sigma: no
Set cut range: 1.05, 1.3
Sky Emission: no
Measure Shift: yes
Correct Shift: yes
Manual Shift: no
CUTRANSFORM flux: no
$ python Warp_sci.py ./TEST/WIDE/4_Ari_list.txt -r ./TEST/WIDE/
  -c ./TEST/WIDE/WINERED_calibration_LCO22b_WIDE100_20220914_v2/
   -v ./TEST/WIDE/ -d ./TEST/4_Ari_WIDE_test_p
   -p ./TEST/WIDE/sampleParameter.txt
```

(2) With the -q option, the Warp_sci.py runs in the interactive mode and ask you the analysis parameters.

```
$ python Warp_sci.py ./TEST/WIDE/4_Ari_list.txt -r ./TEST/WIDE/
  -c ./TEST/WIDE/WINERED_calibration_LCO22b_WIDE100_20220914_v2/
   -v ./TEST/WIDE/ -d ./TEST/4_Ari_WIDE_test_q -q
#####
=======================================================
=== Please answer to the following questions.      ===
===                     OR                         ===
=== Just press enter key (adopt default settings). ===
=======================================================
Adopt aperture ranges read from the input file? (def:no) :yes
Subtract background spectra from object spectra? (def:False) :yes
Detect and interpolate the cosmic rays? (def:yes) :yes
Measure the spectra offsets among multiple frames? (def: yes) :yes
Correct the spectra offsets among multiple frames? (def: yes) :yes
Use the spectra offsets values written in list file? (def: no) :yes
#####
```

4 Reference

1. Ikeda et al. 2022, PASP, 134, 015004, "Highly Sensitive, Non-cryogenic NIR High-resolution Spectrograph, WINERED"

2. Hamano et al. 2023, in preparation

5 Contact

If you have a question or feedback about the WINERED Pipeline (or WINERED in general), please send an e-mail to winered-contact@cc.kyoto-su.ac.jp.

Here is some useful links related to the WINERED.

- WINERED Home Page
- WINERED Exposure Time Calculator (ETC)
- WINERED Automatic Reduction Pipeline (WARP) on github