A quick guide to Heisline data reduction platform
For Heisline version 2.7
31st December, 2017

What is Heisline?

Heisline is linear interactive (under update) data reduction platform, initially constructed to analyze fits files from the direct imager of the Skinakas observatory 1.3m Richey Chrétien Telescope. Its major advantage is to able to work with extended and diffused objects, removing any influence of sky and continuum. It initially bias-subtracts, flat-fields, aligns, combines, astrometrically solves, starlight-subtracts, continuum subtracts, sky-subtracts and bins images. Followingly, it calculates the photometry of the diffused extended objects, that survived the previous procedure, based on an inserted flux calibration (future updates will enable the code to flux-calibrate). Finally, based on the input $[SII]$ and $H\alpha + [NII]$ images, calibrated shock velocity maps are produced, using Allen's MAPPINGS III library.

Request

Please read the manual appropriate for your version, found in the folder where the code was installed. It is very important to have the latest info on the software you are about to use

Minimum Hardware Specifications

In order to properly function, Heisline requires:

1. 1 GHz CPU with at least 1 availeble core
2. 20MB hard disk space + 7 times the space required for the under-reduction data
3. 1 GB of RAM

Recommended Hardware Specifications

In order to fully and quickly function, Heisline requires:

1. 2.3 GHz CPU with at least 2 availeble core
2. 100MB hard disk space + 10 times the space required for the under-reduction data
3. 3 GB of RAM

Minimum Software Dependencies

In order to properly function, Heisline requires:

1. Linux
2. Python version >=2.7.0
3. pip
4. solve-field
5. IRAF
6. PyRAF

Recommended Software Dependencies

In order to smoothly function, Heisline requires:

1. Fedora-Based Linux
2. Python 2.7.13
3. pip 9.0.1
4. solve-field revision 0.67, with full dependencies
5. IRAF revision 2.16
6. PyRAF 2.1.11

First Installation

Follow these steps when installing Heisline for the FIRST TIME:

PART 1

1. Download the following file: ftp://iraf.noao.edu/iraf/v216/support/linux/x_daophot.e
2. Locate your "noao/bin.linux64" folder
3. Replace the original x_daophot.e file in the previously named folder, with the one you were instructed to download

PART 2

1. Download the preparehere.sh script you were provided
2. Place the provided preparehere.sh file in the folder where IRAF was initialized (usually this is where the login.cl file exists) and move to that folder
3. Remove any residual files (that usually means strip the folder of all files and folders, except: [login.cl, uparm, pyraf, preparehere.sh])
4. Run preparehere.sh (./preparehere.sh) as NON-SUDO user (your admin password will be asked at some point, to install some needed dependencies)
5. If everything ran smoothly there will be no red-colored messages on the terminal and you will have been provided with the latest version of the platform
6. Check the heisline_version file for the version downloaded
7. Check that you have the following files: [heisline_main.py, heisline_calls.py, heisline_version, preparehere.sh, heisline_readers.py, heisline_skyandbin.py, heisline_fluxcal.py, heisline_map.py, heisline_manual.pdf]

Updating the Code

UPDATING the code is HEAVILY RECOMMENDED. If possible it should happen every 1 or 2 days, to ensure no bugs occur and that the results are scientifically valid. If a bug occurs in a version and gets solved at the next one, the version with the solution will clearly state the bug and what kinds of results of the bugged version cannot be trusted (if any), so that you can revise your strategies. Additionally, no previous versions are deleted from your computer. Older versions of the code are stored within separate folders named after the version number, so that you can replicate all results, using the appropriate version of the code and keep track of your version history. The procedure to update is as follows:

1. Enter the folder where you have initially placed your preparehere.sh file
2. Run preparehere.sh (./preparehere.sh) as NON-SUDO user (your admin password will be asked at some point, to install possible needed dependencies)
3. If everything ran smoothly there will be no red-colored messages on the terminal and you will have been provided with the latest version of the platform
4. Check the heisline_version file for the version downloaded
5. Check that you have the following files: [heisline_main.py, heisline_calls.py, heisline_version, preparehere.sh, heisline_readers.py, heisline_skyandbin.py, heisline_fluxcal.py, heisline_map.py, heisline_manual.pdf]

Keep in mind that you have to keep updating yur software prerequisites (sections about Software Dependencies) according to the version you are using. This manual names the prerequisites for the version named at the beginning of the document.

Before running the code

Before running the code consult the following checklist to make sure you will have the best results possible:

1. Update the code
2. Gather all your .fits images in one folder (NOT the one the code is in)
3. Make sure that all images of the same type (bias, flat, object) have the same basename (e.g. BIAS_*, FLAT_sdssr_*, FLAT_Ha_*, G65.8-0.5_Ha_*).
4. Make sure that the image names do not contain weird characters (safest choice is to use underscore, dot, minus)

5. Write down the names of the filters as exist in the basename (not necessary but quite a good habit)
6. Create a list of all bias frames names (e.g. by running "ls Bias_*.fit > biaslist")
7. Create a list of all flat frames in a specific filter, named flatlist_[filtername], for each filter (e.g. by running "ls Flat_Ha_*.fit > flatlist_Ha")
8. Create a list of all object frames in a specific filter, named images_[filtername], for each filter (e.g. by running "ls G65.8-0.5_Ha_*.fit > images_Ha")
9. It might be helpful to note the exposure times of the images as well as few characteristics of them (fwhm of the PSF in pixels for each filter (consider using funtools), center RA and Dec (consultimage header))
10. Cross your fingers

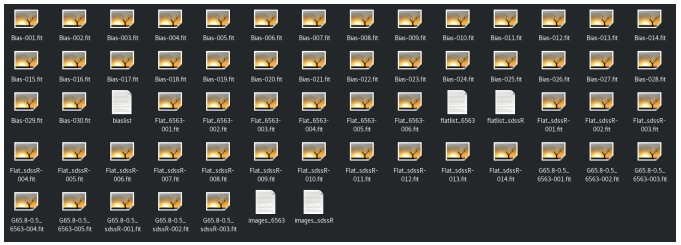

Figure 1: The condition inside the code-containing directory before running it.

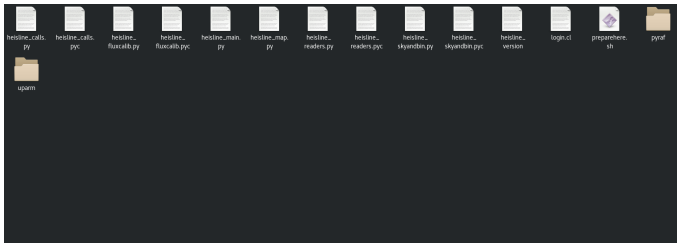

Figure 2: The condition inside the images-containing directory before running the code.

Running the code

To run the code simply move to the folder that contains it (and the login.cl file) and run: "python heisline_main.py". It might also help you to have launched another instance of terminal in the folder where the images exist, so that you have an independent view of the procedures performed. In the part to follow we will call the terminal running the code "Terminal 1" and the terminal in the directory of the images "Terminal 2".

Code procedures

The following part of the manual will analyze in practice the operation of Heisline.

Preliminary messages

Immediately after running the code, a window pops notifying you of the version you are using, when you downloaded it and whether any bugs have been detected. Press enter to dismiss that screen.

After dismissing it you should see a bunch of IRAF packages loaded, but no message saying that a login.cl does not exist. If it does you may need to initialize IRAF again (mkiraf) in the folder where the code was installed

A message is now asking you to enable or not the interactive mode. It is recommended to enable it, since it allows much more control over the pipeline functions. The following text assumes you enabled interactivity.

Directory Selection

A message is asking you for the image directory. Run "pwd" in Terminal 2 and copy the result over to Terminal 1. Press enter.

Bias List

A message is asking you to enter the bias list. Run "ls" in Terminal 2, locate the bias-frames list and copy its name to Terminal 1. Press enter.

Bias Frame Combination

A message is asking whether you want to subtract the combined bias frame from the images. Answer it.

The master bias frame was created by average-combining all the other bias frames. It was named BIAS.fit. If needed, you can now replace it and press enter.

Filter Selection

Now enter the number of the filters, images of which you are about to reduce (including the narrowband ones).

Enter now the names of the filters, pressing enter after every one. Name them the same way you named the appropriate flat- and image- lists.

The computer is now asking you to make sure you have the needed file-lists saved in your image directory. If you followed the preparation procedure from above you will have those files. Press enter.

A message saying something about the "tmpremoveline" is not to be taken seriously.

Bias Subtraction

The code just bias-subtracted all frames (both flat- and science-frames). It is asking you to replace all bias-subtracted images (ending in _b.fit) and resume pressing enter.

Flat Frame Combination

The computer is asking whether you want to flat-field the images. Answer it.

It then median-combines all provided flat frames, then normalizes them and asks you whether you want to replace the combined/normalized flat frame for any filter (FLAT_[filtername].fit). After that press enter.

Flat-Fielding

After that, the code will flat-field the science images, dividing them by the corresponding norm flat frame. Check the result of the flatfielding (images ending in _b_f.fit), replace what needs replacement and then press enter.

Aligning

The code now asks whether to align the images (prior to median-combining them) or not. Answer and press enter.

The first image of the first filter was selected to be the pivot image for the alignment. All other images will be shifted and trimmed to match it. The lists "merged_b_f" and "merged_b_f_sh" contain all the image names before and after alignment. Change the first image name in both lists to select a new pivot or simply press enter to move on.

The code now asks for the max shift needed. Move to Terminal 2 and open all the images in ds9. Visually inspect the images to find the maximum diagonal shift needed. Close ds9, move to Terminal 1, enter the noted shift and press enter.

Now enter your rough estimate of the fwhm of the PSF measured in pixels (see section "Before running the code") and press enter.

The code now asks you to select "5-10 stars that are isolated and well-defined". Go to the automatically oppened ds9 window, wait for the image to open, then visually inspecting the image, detect isolated, non-saturated stars ACROSS THE WHOLE IMAGE, and mark them by placing the cursor on them and pressing the 'a' keystroke. After having selected 5-10 stars SPREAD OUT ACROSS THE WHOLE IMAGE, quit the procedure using the 'q' keystroke and closing the ds9 window.

You now have the option to remove any lines (stars) accidentaly selected, by editing the "aligcoordslog" file in the images folder. Do the necessary modifications and press enter.

Alignment, shifting and trimming will occur, then a ds9 window will open, displaying all the final images. Check if the alignment is appealing and if yes answer y, if it was faulty, answer n. If answered n, the alignment will be redone, altering the initial parameters and the ds9 window will reopen for you to answer again. If after 10 tries the code cannot align the images, you will be requested to do so yourself. If it has come to that, align the images any way you can, then press enter.

Basename Selection

The program asks for a basename. It is a good habit to enter as basename the actual basename of the images up to this point. Enter your selected basename, then press enter.

Median Combining Images

The code asks whether to median-combine the images. Answer using y or n and enter. You will be expected to have provided the file 'basename_filter.fit' in the images folder if you want to skip median-combining for whatever reason.
The images are median combined.

Astrometry

The code asks whether to add astrometricalsolution in the form of WCS to the images. Answer y if you want to do so. This step is essential even if WCS has previously been added to the images, since the astrometry performed here is taking into account possible distortions of the image.
The program now requests the rough estimate of the R.A. and Dec. of the object (see section "Before running the code"). Enter it in the format requested and press enter.
The code will now add WCS to the images, detecting stars (using SExtractor) and triangulating (using the USNO-B catalog) within a radius of half degree around your rough estimate.

Starlight Removal

It is now time to remove the stars from the images. The procedure for each filter is as follows:

1. You are asked if you want to apply star removal for the current filter. Answer y/n + enter.
2. You are asked whether you think the field in that filter is very heavily crowded. Answer y/n + enter.
3. Enter the exposure time of the image in that filter (present in header) in seconds.
4. A ds9 window opens. Wait for the image to be displayed. Using the 'a' keystroke while the cursor is over a star will display (in Terminal 1) the emission characteristics of the star. Using the 'm' keystroke while the cursor is over a background area will display (in Terminal 1) the noise characteristics of the background. Use these keystrokes to acquire few star and few background data, then press 'q' and THEN close ds9 (if you close ds9 earlier the code will hang). Please note that in this case the emission of the extended, diffuse object (e.g. Supernova Remnant) is considered background and its noise characteristics should also be taken into account (in simple words use parts of the diffuse object, both close to its center and in the outskirts of it, when computing the standard deviation of the background).
5. Enter the estimate of the fwhm of the PSF of the stars, based on your examined results (we propose MOFFAT to be the best approximation of the PSF of the stars).
6. Enter your estimate of the standard deviation of the background in counts.
7. The stars in the image are detected (using daofind) and photometricized (using phot).
8. Read the message and press enter.

9. A ds9 window opens and soon after the image loads. Switch to the ds9 window and press n. Switch to the graph that just oppened and examine the mesh plot of the brightest detected star. If you like what you see (no-saturation, well-definement, no nearby sources) press a, if you don't like it press d. Switch back to the ds9 window and press n. Switch back to the graph window, examine the next star sccepting it ('a') or declining it ('d'). Repeat the process until one of the following occurs: aTerminal 1 displays the message:"Total of 90 PSF stars selected", b You think you've had enough.
10. If you accidentally accepted a star, mark it's ID, you will later have the opportunity to remove it.
11. Switch to the ds9 window and press 'q'.
12. Switch to Terminal 1 and press 'q'.
13. A message reads: "If you selected a star you think you shouldnt have selected please now find the file named selection.pst inside the filters folder and delete the line(s) beginning with the stars ID(s)". Go to Terminal 2. Open the folder named after the current filter. Edit the selection.pst file, removing the row beginning with the bad star's ID. Return to Terminal 1.
14. Press enter.
15. The variable-PSF model is calculated (using psf).
16. Allstar removes the stars from the image 3 times (this takes some time).
17. The star-subtracted image is displayed in ds9.
18. If you like the way th stars were removed close the ds9 and enter y. If you didnt like the result, close ds9 and enter n.
19. If entered n, allstar will keep running and displaying images until you like what you see and eventually answer y. Note that the time it takes to complete the full procedure is exponentially proportional to the initial number of stars detected and exponentially proportional to the number of allstar passes required.
20. Once you like the result, the whole procedure is repeated for the remaining filters.

Scaling Factor Calculation

Now you are asked whether you want the code to calculate the scaling factor. This will be done automatically, using the photometry files from the Allstar step. Answer y/n + enter. The result will be exported to a file for convenience.

The available filters are presented. Enter how many of the filtersare narrowband.

Enter the name of the first narrow filter.

Enter the name of its corresponding continuum filter.

Enter the next narrowband filter name.

And so on...

The scaling factor for all pairs is calculated and stored in the scaling/scalefactors file in format:

[[narrow1,cont1,scalingfactor,uncert],
[narrow2,cont2,scalingfactor,uncert], ...]

Scaled Subtraction

You are now presented with the option to scale all star-removed continuum images appropriately and subtract the scaled result from the star-removed narrowband images. Choose y/n + enter.

Flux Calibration

Now press n + enter (since the automated flux calibration method is under development)
Enter the absorption coefficients and ZeroPoints for the different narrow filters as asked.
You will be now prompted to select which filters correspond to the "Ha+[NII]", [SII] and sdss-r'.

Photometry and Velocity Maps

The Ha image will open in ds9. Mark (circular in preference) regions encompassing background regions and save them.
Close ds9 and enter the name of the saved region file in Terminal 1. Repeat with the [SII] and sdss-r' images.
The scientific analysis and the photometry of the Ha emission of the object will be performed at the same sigma level. Enter that level now and press enter.
The photometry of the [SII] emission of the object will be performed at different sigma level than Ha. Enter that level now and press enter.
Enter now the desired binning factor (the results will have greater S/N ratio as the binning factorincreases (with measure)) and press enter.
Enter the absrption coefficients and zero points at the Ha and [SII] bands, as requested by the code.
Enter the scaling factors calculated before.
The photometry of the extended object is saved under results/SNRphotometry.cat and the velocity maps are being displayed on screen. You can manipulate and save them using the window controls.

Wishes for the most awesome images
John A. Kypriotakis
University of Crete, Department of Physics
heisenbergk.com