Mémoire

Auteur : Van Den Abbeele, Matthieu
Promoteur(s) : Jehin, Emmanuel
Faculté : Faculté des Sciences
Diplôme : Master en sciences spatiales, à finalité approfondie
Année académique : 2016-2017
URI/URL : http://hdl.handle.net/2268.2/3219

The TRAPPIST Cometary Data Reduction Cookbook

This cookbook provides basic instructions for reducing and analysing TRAPPIST cometary data in order to produce, from raw images, the radial luminosity profiles, the production rates of various gas species (CN, C_2, C_3, NH and OH) aswell as the $Af\rho$ parameter for the dust activity. We describe all the softwares and scripts used for this purpose and illustrate the various steps of data processing.

1 Hardware and software prerequisites

The data reduction method presented in this cookbook was performed on a PC (Asus G75VX) powered by GNU/Linux Fedora 24 (distributions close to Fedora 24 should work aswell) and equipped with NOAO's IRAF (Image Reduction and Analysis Facility) software system [1]. We will also need the GNU Emacs text editor to modify text files provided by NASA/JPL's HORIZONS system for ephemerides. Emacs is free and easily accessible through the GNOME Software tool included within Fedora. An internet connection will be necessary on several occasions. In addition, we recommand installing a FITS/FTS image viewer, such as SAOImage DS9 [2], installed as auxiliary in order to check the results of your work through the reduction process.

2 Description of the scripts and programs

In addition to the softwares mentionned above, the data reduction involves several files an scripts specifically developed by ULiège's OrCA [3] team for this procedure :

File/Script	**Description**
afrhocalcext.cl	Performs the flux calibration and the removal of the sky contribution ; computes the radial brightness profiles and the $Af\rho$ parameter (aswell as the errors on both).
azimmedian	Computes the median values of the pixels in successive circles around the nucleus. This is done in order to compute the the radial profiles. Called by afrhocalcext.

1. Available here : `http://iraf.noao.edu/`
2. Available here : `http://ds9.si.edu/site/Download.html`
3. Origins in Cosmology and Astrophysics

calib[MMYY].dat	Provides all the zero points, extinction and calibration coefficients required for the flux calibration. Part of the data come from Farnham et al. (2010). [MMYY] stand for the month and the year when the extinction coefficients and zero points were measured.
calibint.cl	Hub for the flux calibration scripts. Used within afrhocalcext.
calibint.sh	Performs the flux calibration on the basis of observations of star HD52266 (when available). Called by calibint.cl.
calibint1.sh	Performs the flux calibration with the help of the data provided by calib[MMYY]. Called by calibint.cl.
ephemXXXX.brol	First ephemeris file of the comet. It will used by hasercalctest.cl (and hasercalc.cl). The "XXXX" suffix is a placeholder for the comet's name.
ephXXXX.dat	Second ephemeris file of the comet. Will be entered in and used by afrhocalcext.cl. The "XXXX" suffix stands again for the comet's name.
haser	Computes the Haser profiles. Used within by pghaser.
hasercalctest.cl	Hub of the Haser-related scripts and programs. Returns the production rates and their error. Despite the "test" written in the name of the file, this script is the definitive one and is **not** to be modified (with exceptions that will be discusssed later). **Be carefull that the names of hasercalc and hasercalctest are counterintuitive.**
hasercalc.cl	Editable version of hasercalctest.cl. For example, you can try running the program with different scalelenghts for the parent and daughter species.
Haserimput.test	Input file fo hasercalc.
Haserimput.testXX-BC	Input file used hasercalctest. The content of this file is discussed in section 8.
interpephem	Interpolates the data provided by the ephemerides (in ephXXXX.dat) at the time of exposure start. Used within afrhocalcext.cl
interprddot	Interpolates the data provided by the ephemerides (in ephemXXXX.brol). Used within hasercalctest.

loginuser.cl	Contains commands to execute when IRAF is launched. It gives command names to some of our scripts.
pghaser	Performs the removal of the dust contribution, removes the pedestal and computes Haser profiles and the production rates. Called by hasercalctest.cl and hasercalc.cl
plothaserfit	Creates a plot with both the Haser model curve and the observed radial brightness profile on which it was fitted.
progtrap2	Creates master bias, dark and flat and performs the data reduction.
renamefits.cl	Copies the images from their original directory (in `trappistraw`) to an output directory (`trappist/data`) while giving them new TRAP.* format names.
sky2xy	Finds the location of the center of the coma using the astrometric solution. Called by afrhocalcext.cl
solarirrad.dat	Provides the values for the solar flux at 1AU required to compute the $Af\rho$ parameter.
subsets	Translation file for the various filters. Used within progtrap2.
trapccd	TRAPPIST's instrument translation file used by IRAF's ccdred package. Allows IRAF to read and translate the headers properly.
truc.sh	Small script linking calibint.cl to calibint.sh or calibint1.sh
wgetschleicher	Gets the fluorescence efficiencies (g-factors) on David Schleicher's website (`http://asteroid.lowell.edu/comet/gfactor.html`). Used within by hasercalc and hasercalctest.

TABLE 1 – List of the scripts needed for the data reduction and analysis. More details are available through the comments within the scripts themselves.

3 Preliminary steps

Before getting to the reduction, some preleminary steps are required. First of all, the following files and scripts must be moved to the *bin* directory (`/home/username/bin`) :

– azimmedian
– calibint.sh
– calibint1.sh
– haser
– interpephem
– interprddot
– sky2xy
– wgetschleicher

We recommend working in a single main directory (aside from the bin). As an example, we will be working in a "Trappist" directory located in the "Documents" directory (full path :`/home/username/Documents/Trappist`). The following files an scripts will be moved to this directory :
– progtrap2.cl
– renamefits.cl
– subsets
– trapccd

Next, inside the same Trappist directory, we create several subdirectories :
– trappistraw (whose name must *not* be changed)
– Test

The "trappistraw" directory will contain all the raw images taken with TRAPPIST. The "Test" directory will be the one in which the radial profiles, $Af\rho$ and production rates will be computed.

Move the following files and scripts to the "Test" directory :
– afrhocalcext.cl
– calib[MMYY].dat
– calibint.cl
– ephemXXXX.brol
– ephXXXX.dat
– hasercalc.cl
– hasercalctest.cl
– Haserimput.test
– Haserimput.testXX-BC
– pghaser.cl
– plothaserfit
– solarirrad.dat

Finally, simply move the "loginuser.cl" script in the home directory.

We are now going to launch IRAF and load the required packages. Open a terminal and enter `xgterm&`. In the newly opened window, enter `cl` to launch IRAF. To load the packages, enter successively :

```
noao
imred
ccdred
tables
onedspec
```

We must point the main directory in wich we will be working. Considering the example given above, still in IRAF, enter :

```
cd /home/username/Documents/Trappist
set TRAPDAT = /home/username/Documents/Trappist
```

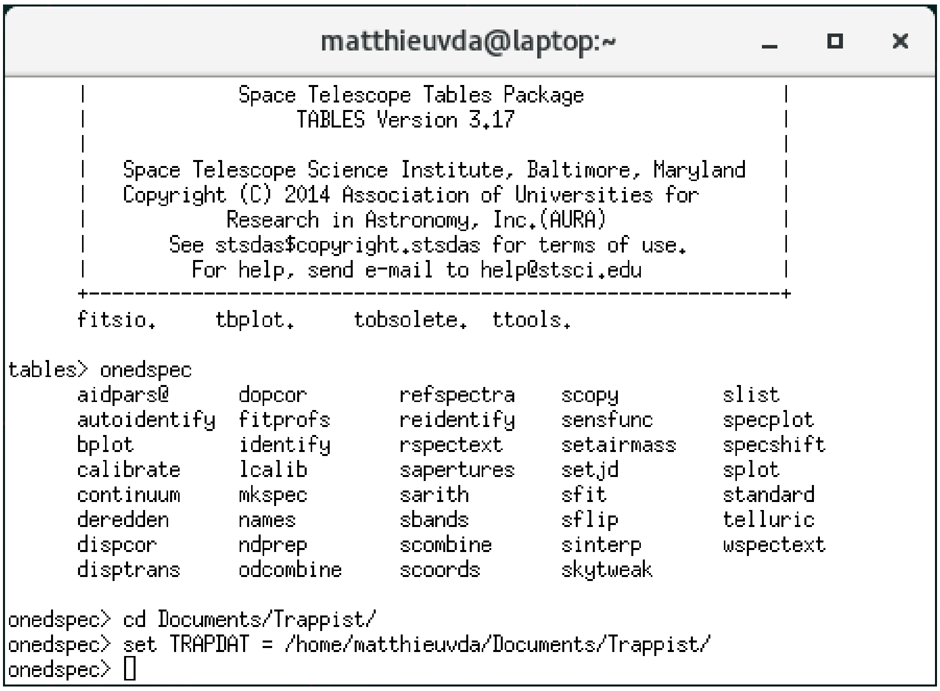


Figure 1 – Screen capture of the IRAF terminal after pointing the Trappist directory.

4 Renaming the images

The procedure for reducing and analysing the images will not work with their original filenames and extension. Both must therefore be changed beforehand.

First of all, all the folders containing the raw TRAPPIST images[4] must be moved in the "trappistraw" directory. As an example, we will work with images taken on April 15, 2016 and gathered in a folder named "20160415". This folder also contains two subfolders : Calibration and AutoFlat. They contain the bias, dark and flat frames taken on the same night as the cometary images.

The raw images produced by TRAPPIST-South have a FTS extension that needs to be changed into a FITS extention. To do this, in a newly opened terminal (not the one used for IRAF), open the directory containing your raw images and use the `rename fts fits *` command to change the extension, as illustrated in figure 2. Make sure **all** the images undergo the same treatment (including flat, bias and dark frames in the Calibration and Autoflat subfolders).

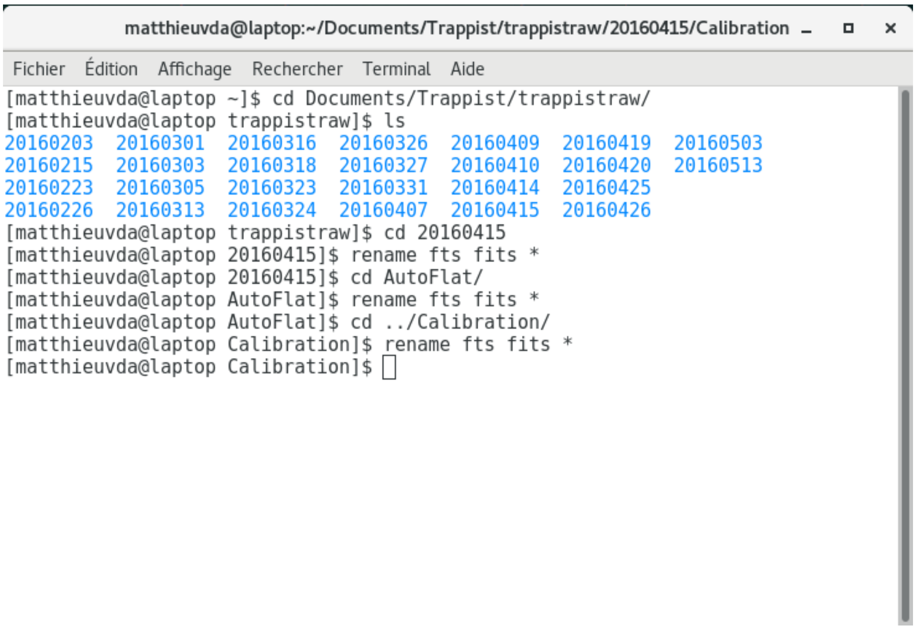


FIGURE 2 – Screen capture of the method for changing image extensions.

We now need to change the name of the images. We will use the renamefits.cl script. This script will copy the images from their subdirectory in trappistraw (in our case, 20160415) to a new "data" subdirectory located at `/home/username/Documents/Trappist/trappist/data` and give them new TRAP.* format names. If the `/trappist/data` subdirectory does not exist, renamfits will create it.

In the IRAF terminal, enter :

```
task renamefits = /home/username/Documents/Trappist/renamefits.cl
renamefits
```

4. These folders are normally named according to the date when the images were taken.

The script will ask you to enter the subdirectory in which are located the images you wish to rename. It will also ask if you wish to erase preexisting files in the `trappist/data` directory. In our example, for a more efficient process, we enter directly :

```
renamefits 20160415 yes
```

All the images within 20160415 (including those in AutoFlat and Calibration) will be found in the output directory with their new names.

NB : Make sure you have calibration frames for **each** filter used for your observations with TRAPPIST. If some are missing, retrive corresponding flat frames from other nights and copy them in the Autoflat subfolder before changing the extensions and using renamefits.

5 Master bias/dark/flats and data reduction

All our images are now ready to be reduced. First of all, in the main directory (Trappist), create two new subdirectories : "tmpdata" and "tmpout". These directories serve respectively as input and output directories for the progtrap2 program. Then, move all the previously renamed images from trappist/data to tmpdata.

In IRAF, enter :

```
task progtrap2 = /home/username/Documents/Trappist/progtrap2.cl
```

Progtrap2 will ask for the values of two boolean variables. The first one determines whether you want to delete the master bias, dark and flat frames once the process is complete (yes or no). The second determines if you want to skip the creation of the master flat, bias and dark frames and execute the reduction process with the help of preexisting master calibration frames (yes or no). In our case, we want to keep the master bias, dark and flat, and execute the entire process. To do that, we enter :

```
progtrap2 no no
```

All the reduced images of the comet along with the master bias, flat and dark can now be found in the tmpout directory. Pay attention to the fact that the content of tmpout will be erased before each execution of progtrap2. We thus strongly recommend moving all the reduced images from tmpout to another custom folder before using progtrap2 again. In our case, we will use a "Final Data" folder.

NB : progtrap2 will create a "ccdlist.log" file. This files contains the name of the reduced pictures aswell as several pieces of information coming from their header, such as the filter. Use this file to look for pictures taken in the NaI filter. Indeed, this cookbook does not apply for the processing of NaI images.

The images of the comet are now reduced and renamed. However, before going further in the data analysis, an additional step is required : we need the ephemerides of the comet.

6 Ephemeris

The images of the comets and their headers do not carry all the information we need to compute the radial profiles of luminosity, the $Af\rho$ parameter, the Haser profiles and the production rates. For our scripts and programs to work, we need to know the evolution of several orbital parameters that can be provided by the emphemerides of the comet. To obtain them, we will use NASA/JPL's HORIZON Web-Interface. Open a web browser and use the following link :
`https://ssd.jpl.nasa.gov/horizons.cgi`

There is a series of options to configure. First, enter the comet you are working on as the Target Body, and La Silla–Trappist [I40] as the Observer Location.
Then, enter the time span of your observations and set the step to about 10 minutes.
In Table Settings, we only select options 1 (Astrometric RA & DEC), 19 (Heliocentric range & range-rate) and 20 (Observer range & range-rate).
Below, in the Optional observer-table settings, change the date/time format into Julian Days and the angle format into decimal degrees. Make sure there are no elevation cutoff and that the "skip daylight" option is not selected. Once all this is done, click on "Use Selected Settings".
Finally, the Display/Output setting must be set to "download/save".
Click on "Generate Ephemeris" to download the text file containing all the required ephemerides for your comet. This file will be called "horizons_result.txt".

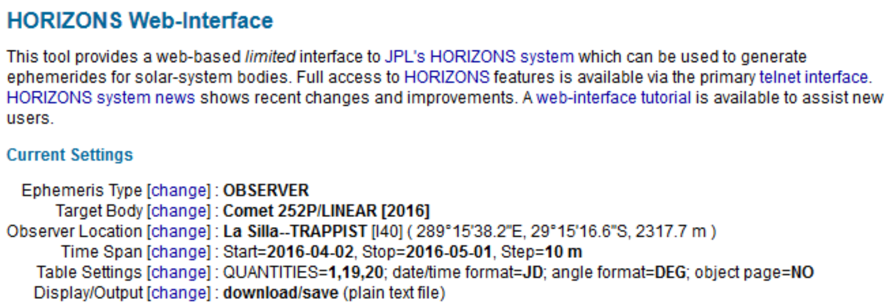
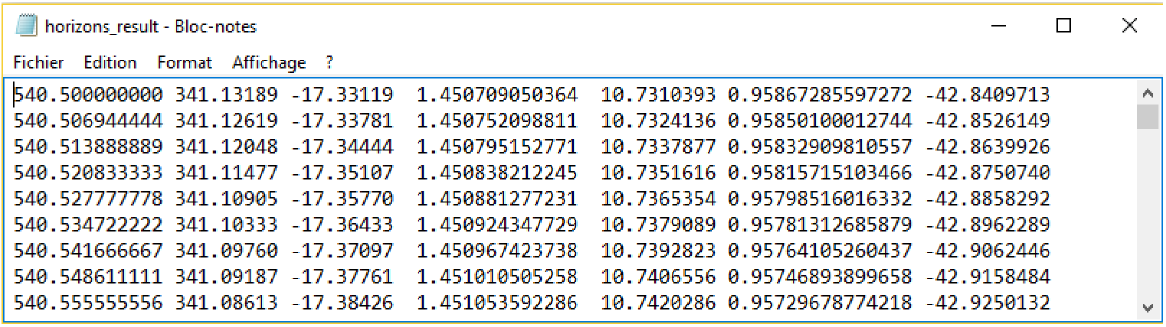


FIGURE 3 – Example of settings configuration in the case of comet 252P/LINEAR.

This file must undergo some modifications before being used in our scripts and programs. First of all, we need to erase all the unnecessary text before and after the actual data. All you need to keep is the data table between the two lines beginning with "$$SOE" (which must also be erased).

The next modification will be done with the help the GNU Emacs text editor. Open horizons_result.txt with Emacs. This software will help us deleting entire sets columns (or "rectangles[5]"). To do so, select the character in the upper left corner of the rectangle and use the keyboard shortcut ctrl+space (toggling in rectangle mark mode). Then, select the character at the lower right corner of the rectangle, and use the shortcut Esc+x to allow the use of commands. Use "kill rectangle" to erase the selected columns[6].

First, we only want to keep the first three digits of the the julian days column **before** the decimal point. Second, we need to erase the Solar and Lunar presence column. It is a small column located between the date and right ascension columns and containing symbols such as '*', 'c', 'm', and so on. In the end, your text file should look like this :



540.500000000	341.13189	-17.33119	1.450709050364	10.7310393	0.95867285597272	-42.8409713
540.506944444	341.12619	-17.33781	1.450752098811	10.7324136	0.95850100012744	-42.8526149
540.513888889	341.12048	-17.34444	1.450795152771	10.7337877	0.95832909810557	-42.8639926
540.520833333	341.11477	-17.35107	1.450838212245	10.7351616	0.95815715103466	-42.8750740
540.527777778	341.10905	-17.35770	1.450881277231	10.7365354	0.95798516016332	-42.8858292
540.534722222	341.10333	-17.36433	1.450924347729	10.7379089	0.95781312685879	-42.8962289
540.541666667	341.09760	-17.37097	1.450967423738	10.7392823	0.95764105260437	-42.9062446
540.548611111	341.09187	-17.37761	1.451010505258	10.7406556	0.95746893899658	-42.9158484
540.555555556	341.08613	-17.38426	1.451053592286	10.7420286	0.95729678774218	-42.9250132

FIGURE 4 – Typical apprearance an ephemeris file after modification using Emacs.

Save this modified file as "ephemXXXX.brol" (pay attention to the new extension), with "XXXX" being a placeholder for the name of the comet.

We also need to create a *second* file, but this time, the rate columns (rdot and deldot) will also be deleted. Use the same procedure as before, and save this new file as "ephXXXX.dat". Move both ephemXXXX.brol and ephXXXX.dat to the "Test" directory we created earlier inside the Trappist directory.

5. See the documentation : `https://www.gnu.org/software/emacs/manual/html_node/emacs/Rectangles.html`
6. If you have trouble manipulating rectangles, we suggest watching this small tutorial by Mike Zamansky (timecode 3 :10) : `https://youtu.be/pcA5NeEudgU`

```
ephXXXX - Bloc-notes
Fichier  Edition  Format  Affichage  ?
516.395833333 257.59879   9.06968  1.237933334666 0.28562209172983
516.402777778 257.59710   9.06987  1.237986750707 0.28566518646324
516.409722222 257.59541   9.07005  1.238040169972 0.28570823239351
516.416666667 257.59370   9.07024  1.238093592461 0.28575123166058
516.423611111 257.59199   9.07043  1.238147018174 0.28579418649969
516.430555556 257.59026   9.07062  1.238200447108 0.28583709923714
516.437500000 257.58852   9.07081  1.238253879263 0.28587997228580
516.444444444 257.58677   9.07100  1.238307314639 0.28592280814051
```

FIGURE 5 – Example of an ephXXXX.dat file.

7 Radial profiles and $Af\rho$

We now have everything we need to start computing the radial profiles. First of all, move the cometary images from which you want to compute the radial profiles, $Af\rho$ parameter and production rate to the "Test" directory. Before declaring any new task in IRAF, we need to make sure that the right calibration file (calib[MMYY].dat) will be read by the "calibint1.sh" script (which, as a reminder, is located in `/home/username/bin`). The element we need to modify is in the fourth line of the script, as shown in figure 6. Open calibint1.sh in a text editor and change the calibration file name into the correct one.

```
4  awk '{if($7=='$1'){ZP=$9; print 10.^(10-0.4*ZP)}}' calib[MMYY].dat
```

FIGURE 6 – Fourth line of the calibint1.sh script. The element to be modified is highlighted.

Changes must also be made in the "afrhocalcext.cl" script in lines 50 and 51. In line 50, we need to enter the ephXXXX.dat file we created earlier as the value of the ephem variable. In line 51, we have to enter again our calibration file. Figure 7 shows how the modified lines should look like after modification.

```
46    #delete("centerlist")
47    files ("TRA*.fits", >"listim")
48
49    pi=3.14159
50    ephem="ephXXXX.dat"
51    calibration="calib[MMYY].dat"
52
53    ilist="listim"
54    fout2 = "profdata1.dat"
```

FIGURE 7 – Highlight of the lines to be modified in afrhocalcext.cl

Now, enter the following commands in IRAF :

```
cd /home/username/Documents/Trappist/Test/
task afrhocalcext = /home/username/Documents/Trappist/Test/afrhocalcext.cl
task calibint = /home/username/Documents/Trappist/Test/calibint.cl
afrhocalcext
```

When running, afrhocalcext will create several new data files. The radial profiles of the images are given in text files wearing the exact same filename as the pictures themselves (including the ".fits"), but with by a "rad_" prefix and a ".txt" extension. Same for the upper and lower errors with "radeplus_" and "rademoins_" prefixes respectively. Note however that "radplus_" and "radmoins_" files are actually the **sum** of the radial profiles and the upper and lower errors respectively. The true errors are given by the difference between "radplus_" or "radmoins" data and "rad_" data.
As an example, let's consider an image named "TRAP.2016-04-15T04 :18 :02.fits". Its radial profile will be "rad_TRAP.2015-08-24T04 :18 :02.fits.txt", the upper error, "radeplus_TRAP.2016-04-15T04 :18 :02.fits.txt" and the lower error "rademoins_TRAP.2016-04-15T04 :18 :02.fits".

The "rad_", "radplus_" and "radmoins_" files consist of 18 columns :

1. Name of the image.
2. Distance rx from the comet's nucleus in pixels.
3. Number of pixels forming the circle of radius rx around the nucleus.
4. Median flux at a distance rx (in $ADU\ s^{-1}$).
5. Distance r from the nucleus (in arcsec).
6. Total number of pixels in a disk of radius rx.
7. Total flux in a disk of radius rx (in $ADU\ s^{-1}$).
8. Median flux at a distance r from the comet center (in $ADU\ s^{-1} arcsec^{-2}$)
9. Median magnitude per $arcsec^2$ at a distance r from the comet center.
10. Flux at a distance r from the nucleus per unit wavelenght (in $erg\ cm^{-2}s^{-1}\text{Å}^{-1}arcsec^{-2}$).
11. Flux at a distance r from the nucleus in the entire filter's band (in $erg\ cm^{-2}s^{-1}arcsec^{-2}$).
12. Total magnitude of a disk of radius r.
13. Integrated flux in a disk of radius r (in $erg\ cm^{-2}s^{-1}\text{Å}^{-1}$).
14. Integrated flux in a disk of radius r in the filter's band (in $erg\ cm^{-2}s^{-1}$).
15. Name of the filter.
16. Time in Julian Days.
17. Heliocentric distance (in AU).
18. Geocentric distance (in AU).

The $Af\rho$ parameter, along with the error on it, is also computed for images in the continuum filters (UC, BC, GC, RC, B, V, R and I) and delivered in "afrhoXX-tot.txt" text files named according to the filter. For example, for images in the BC filter, the results will be found in the file "afrhoBCtot.txt". These files consist of 14 columns :

1. Name of the image.
2. X coordinate of the comet's nucleus on the image (in pixels).
3. Y coordinate of the comet's nucleus on the image (in pixels).
4. Time in Julian Days.
5. Geocentric distance (in AU).
6. Heliocentric distance (in AU).
7. Distance rx from the nucleus in pixels.
8. Distance ρ from the nucleus in cm.
9. $Af\rho$ (in cm).
10. Upper error on the $Af\rho$ (in cm).
11. Lower error on the $Af\rho$ (in cm).
12. Integrated flux in a disk of radius ρ in $ADU\ s^{-1}$.
13. Integrated flux in a disk of radius ρ in $erg\ cm^{-2}s^{-1}\text{Å}^{-1}$.
14. Flux in ADU per pixel and per second at a distance ρ from the nucleus.

8 Production rates

We must now use the radial profiles generated previsouly to extract the production rates of the gases. Let's consider several radial profiles in a generic XX filter (XX is a placeholder for CN, OH, NH, C2 or C3 depending on the filter in which your images are taken).

Create a file named "Haserimput.testXX-BC" (again, replace XX with the name of the appropriate filter). In this file, you will have to enter four pieces of information in the following order :

1. The name of the file containing the radial profile in the XX filter, but **without** the "rad_ " prefix.
2. The name of the file containing the radial profile in the BC filter. This time, keep the "rad_ " prefix. Make sure the BC radial profile is the closest in time to the XX radial profile. In the absence of such a BC radial profile, use a one taken in the GC filter. In that case, the input file must be renamed "Haserimput.testXX-**GC**".

3. The fc coefficient of the XX filter. This coefficeint is required for removal of the dust contribution to the radial profile. A list of recommended values are given in table 2.
4. A background (or pedestal) compensation factor. This factor will almost always be zero. Details are available in the comments within pghaser.cl.

Filter	fc
OH	5
NH	20
CN	25 ± 5
C_3	190 ± 15
C_2	170 ± 10

Table 2 – Recommanded values for the fc coefficient for each filter.

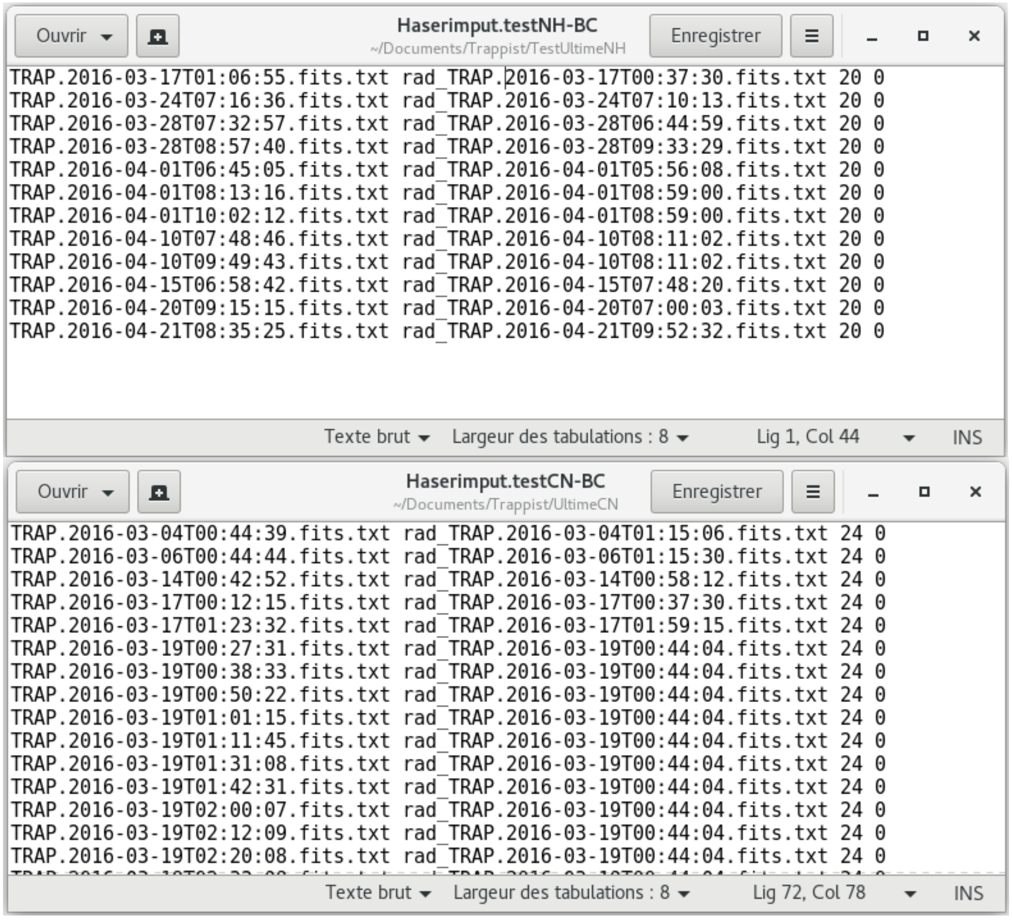


Figure 8 – Examples of input files. The top file is used for radial profiles in the NH filter derived from images taken on different nights. The other one is used for another, much more numerous batch of radial profiles in the CN filter derived from images taken over several months.

You can enter several lines corresponding to several radial profiles in the same input file (as illustrated by figure 8). However, we recommend creating separate files for each filter.

We now need to modify some lines in hasercalctest.cl, as shown in figure 9. In line 31, enter the "ephemXXXX.brol" file that we created earlier. Next, in line 32, enter the input file we just created (Haserimput.testXX-BC). Finally, inline 35, enter a name for the *output* file in which the production rates will be delivered (this file will be created by the program). We recommend using the same suffix as the input file. In our case, we called it "outputhasertestXX-BC".

```
31  ephem = "ephemXXXX.brol"
32  listinput="Haserimput.testXX-BC"
33  JD1=0
34  JD2=999.99
35  logoutput="outputhasertestXX-BC"
```

FIGURE 9 – Highlight of the lines that ought to be modified in hasercalctest.cl

Go back to IRAF and enter the following commands :

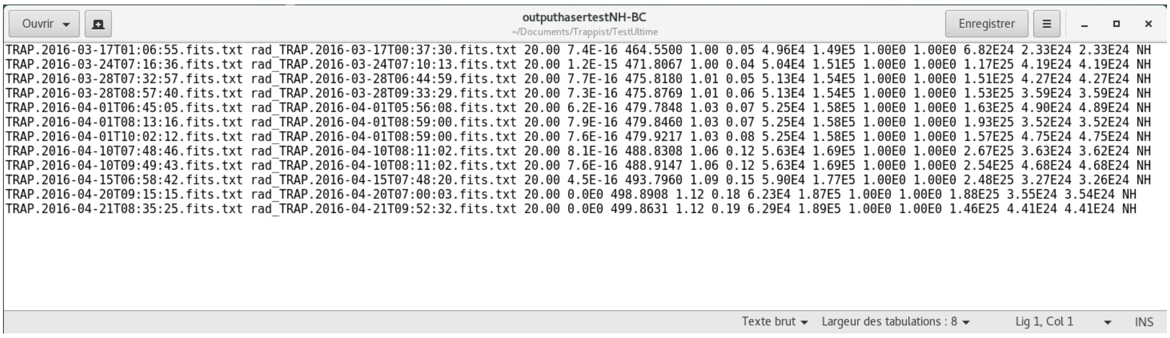
```
task pghaser = /home/username/Documents/Trappist/Test/pghaser.cl
task hasercalctest = /home/username/Documents/Trappist/Test/hasercalctest.cl
hasercalctest
```

Hasercalctest will ask for the value of a boolean variable. It determines whether you wish to erase all previous content of the output file (yes) or just append the new results to it (no).

The results of the procedure are delivered in "outputhasertestXX-BC.txt". This file consists of 15 columns (in that order) :

1. Name of the gaz radial profile file (the same as the one written in the input file)
2. Name of the BC radial profile file (the same as the one written in the input file).
3. fc coefficient (the same as the one written in the input file).
4. Corrected background compensation (pedestal). Usually zero, or very close to zero.
5. Time of the observation in Julian days.
6. Heliocentric distance of the comet (in AU).
7. Geocentric distance of the comet (in AU).

8. Scalelength of the parent species (in seconds).
9. Scalelenght of the daughter species (in seconds).
10. Ejection velocity of the parent species (in $km\ s^{-1}$).
11. Ejection velocity of the daughter species (in $km\ s^{-1}$).
12. **Procution rate** of the daughter species (in molecule per second).
13. Upper errors on the production rate (in molecule per second).
14. Lower errors on the production rate (in molecule per second).
15. Name of the filter.




```
TRAP.2016-03-17T01:06:55.fits.txt rad_TRAP.2016-03-17T00:37:30.fits.txt 20.00 7.4E-16 464.5500 1.00 0.05 4.96E4 1.49E5 1.00E0 1.00E0 6.82E24 2.33E24 2.33E24 NH
TRAP.2016-03-24T07:16:36.fits.txt rad_TRAP.2016-03-24T07:10:13.fits.txt 20.00 1.2E-15 471.8067 1.00 0.04 5.04E4 1.51E5 1.00E0 1.00E0 1.17E25 4.19E24 4.19E24 NH
TRAP.2016-03-28T07:32:57.fits.txt rad_TRAP.2016-03-28T06:44:59.fits.txt 20.00 7.7E-16 475.8180 1.01 0.05 5.13E4 1.54E5 1.00E0 1.00E0 1.51E25 4.27E24 4.27E24 NH
TRAP.2016-03-28T08:57:40.fits.txt rad_TRAP.2016-03-28T09:33:29.fits.txt 20.00 7.3E-16 475.8769 1.01 0.06 5.13E4 1.54E5 1.00E0 1.00E0 1.53E25 3.59E24 3.59E24 NH
TRAP.2016-04-01T06:45:05.fits.txt rad_TRAP.2016-04-01T05:56:08.fits.txt 20.00 6.2E-16 479.7848 1.03 0.07 5.25E4 1.58E5 1.00E0 1.00E0 1.63E25 4.90E24 4.89E24 NH
TRAP.2016-04-01T08:13:16.fits.txt rad_TRAP.2016-04-01T08:59:00.fits.txt 20.00 7.9E-16 479.8460 1.03 0.07 5.25E4 1.58E5 1.00E0 1.00E0 1.93E25 3.52E24 3.52E24 NH
TRAP.2016-04-01T10:02:12.fits.txt rad_TRAP.2016-04-01T08:59:00.fits.txt 20.00 7.6E-16 479.9217 1.03 0.08 5.25E4 1.58E5 1.00E0 1.00E0 1.57E25 4.75E24 4.75E24 NH
TRAP.2016-04-10T07:48:46.fits.txt rad_TRAP.2016-04-10T08:11:02.fits.txt 20.00 8.1E-16 488.8308 1.06 0.12 5.63E4 1.69E5 1.00E0 1.00E0 2.67E25 3.63E24 3.62E24 NH
TRAP.2016-04-10T09:49:43.fits.txt rad_TRAP.2016-04-10T08:11:02.fits.txt 20.00 7.6E-16 488.9147 1.06 0.12 5.63E4 1.69E5 1.00E0 1.00E0 2.54E25 4.68E24 4.68E24 NH
TRAP.2016-04-15T06:58:42.fits.txt rad_TRAP.2016-04-15T07:48:20.fits.txt 20.00 4.5E-16 493.7960 1.09 0.15 5.90E4 1.77E5 1.00E0 1.00E0 2.48E25 3.27E24 3.26E24 NH
TRAP.2016-04-20T09:15:15.fits.txt rad_TRAP.2016-04-20T07:00:03.fits.txt 20.00 0.0E0 498.8908 1.12 0.18 6.23E4 1.87E5 1.00E0 1.00E0 1.88E25 3.55E24 3.54E24 NH
TRAP.2016-04-21T08:35:25.fits.txt rad_TRAP.2016-04-21T09:52:32.fits.txt 20.00 0.0E0 499.8631 1.12 0.19 6.29E4 1.89E5 1.00E0 1.00E0 1.46E25 4.41E24 4.41E24 NH
```

FIGURE 10 – Final results of the procedure.

In the case you would like to use hasercalc.cl instead of hasercalctest.cl, the procedure the same, except for two differences. First, the name of the input and output files do not need any "XX-BC" suffix. Second, in the imput file, you now need to write the "rad_" prefix before the name of the radial profile. Hasercal does not provide the errors on the production rates. This script is mainly is used as a test script to test different scalelenghts for the parent and daughter species.

Finally, you can use the plothaserfit.cl to create of plot of the radial profiles and the Haser model fitted on it. Like other scripts, you need to declare them as task in IRAF :

```
task $plothaserfit = /home/username/Documents/Trappist/Test/plothaserfit.cl
plothaserfit
```

The plot generated will only display the **last** profiles processed by hasercalctest (or hasercalc), however.

You are now able to compute radial profiles, gas production rates and $Af\rho$ values from raw TRAPPIST cometary images. For a better mastery of the process, we recommend going through the various scripts and programs used.