Mid-IR data reduction manual of CANARICAM: RedCan pipeline

O. González-Martín (omairagm@iac.es)
P. Esquej (pilar.esquej@cab.inta-csic.es)

October 21, 2013

1 Purpose of the document

This document summarises the procedure to analyse imaging and low-resolution spectroscopic (bands N and Q) observations[1] performed with the infrared camera CanariCam on the Gran Telescopio de Canarias (GTC). The pipeline was initially developed using R-ReCS data given that CanariCam was not available at the time of writing the code, and thanks to the similarities of both instruments. This pipeline has been now successfully tested using a set of both T-ReCS and CanariCam data.

All observations are performed in chop/nod mode. The basic feature of all ground-based mid-infrared observations is having a short exposure time, called *frame time*, to avoid saturation due to the strong thermal emission from the telescope and the atmosphere. The array is read frequently and summed up in a temporary buffer for each chop position. These buffers are read out as *savesets*. One or more of these *savesets* are written per nod position. In the raw data file there is one extension per nod position. The image for each nod has a size of [XSIZE, YSIZE, i, N], where N is the number of *savesets* per nod. The third dimension i selects the chop position, and it is set to 1 for the on-source beam and 2 for the off-source beam respectively (taken from T-ReCS webpage[2]).

The header keywords relevant to this document are the following:

- FRMTIME
- FRMCOADD
- CHPCOADD

This document is structured as follows: Sect. 2 describes how to obtain the software, the system requirements, and the environmental settings to run the pipeline. Sect. 3 presents the different options and the input parameters. Sect. 4 describes the steps to follow for the data reduction. These include the identification of observations, flatfield correction, stacking of observations, image flux calibration, slit loss correction, wavelength calibration, spectral extraction, absolute flux calibration and the combination of multiple exposures (when available) to get the final calibrated spectrum.

Pipeline authors: O. González-Martín, J.M. Rodríguez Espinosa, T. Díaz Santos, C. Packham, and P. Esquej.

[1] Please note that the analysis of high-resolution spectroscopy is not included here.

[2] http://www.gemini.edu/?q=node/10061

2 Downloading and installing

2.1 System requirements

- IRAF [3]:
  The IRAF configuration file **login.cl** must be located in the $HOME directory. In addition to the NOAO package included in the main distribution of IRAF, other external packages are needed. These are: the Gemini IRAF package, *fitsutil*, *gmisc* and *nmisc*[4]. These external packages are distributed separately but can be easily installed.

- *Python, sys*
  They are installed by default in any unix system.

- *Numpy, scipy, pyfits, pyraf, dfist, fitsort*

- IDL
  The *astrolib*[5] library is needed. Other IDL routines specifically developed for this work are included in the pipeline distribution.

- An internet connection is required to run the pipeline, as the NED database is called to obtain the redshift of the sources.

We strongly recommend to install the UREKA astronomy package[6] as it contains the main IRAF and PYTHON distributions plus external packages and modules used here[7]. Please note that the pipeline does not work using software installed through SCISOFT given that there is an incompatibility of versions.

The pipeline has been successfully tested both under MAC OSX Lion, MAC OSX Snow Leopard and Ubuntu 11.04, with the following configuration:
– IDL 7.0, IDL 6.3, IDL 8.0
– IRAF 2.15
– pyraf 1.10
– numpy 1.5.1
– dfits 1.2
– fitsort 1.1

[3]IRAF is distributed by the National Optical Astronomy Observatories, which is operated by the Association of Universities for Research in Astronomy, Inc., under cooperative agreement with the National Science Foundation.

[4]http://www.gemini.edu/node/10795 and http://iraf.noao.edu/iraf/ftp/iraf/extern

[5]http://idlastro.gsfc.nasa.gov/

[6]http://ssb.stsci.edu/ureka/

[7]Note for MAC users: to get and install *dfits* and *fitsoft* please select your preferred directory for software installation and execute the commands in /PIPELINE/CSHELL/MAC_extrasoft.csh

2.2 Downloading

While the pipeline is being developed the best way to test it is using DROPBOX, since its last version will be periodically updated within a directory called CANARICAM. This will allow the user to check the last version of the pipeline.

After installing DROPBOX (see http://www.dropbox.com/), you should send an e-mail to O. Gonzalez Martin (omairagm@iac.es) requesting to share the pipeline with you. You will then receive an invitation to the folder CANARICAM that, after your acceptance, will appear in your DROPBOX. You can run the most recent version of the pipeline by setting the *redcan* environment in your initialisation script to the Dropbox location in your computer (see Sect.2.3). Note that when the pipeline is finished it can be locally installed by unpacking a tar file.

The pipeline is distributed in a directory called CANARICAM. This is composed by two main subdirectories called PIPELINE – where all the scripts are included – and MANUAL – where the last version of the manual can be found. Within the PIPELINE directory one can find IDLPRO, PYPRO and CSHELL, containing routines written in IDL, Python and CSHELL languages respectively. FILTERS contains the CanariCam filters, and COHEN comprises the list of standard stars for the spectroscopic flux calibration. This document and the CanariCam manual can be also found in the CANARICAM folder.

Note: If you find any problem with the pipeline please contact O. Gonzalez Martin for suggestions and/or improvements. Please do not try to fix it in the DROPBOX directory, as your modifications will be overwritten the next time that is synchronised by the developer.

If you prefer not to install DROPBOX and/or favour a local copy of the pipeline, a temporary version of the software can be download from http://dl.dropbox.com/u/3484086/RedCan.tar.gz
To decompress: tar -zxvf Redcan.tar.gz

2.3 Initialisation and setup

To initialise the software, the environmental variables have to be properly set up. We recommend to modify the user configuration file in your home directory.

For bash users, please open your $HOME/.bashrc (or $HOME/.login, or $HOME/.profile) and include the following lines:

```
# RedCan
REDCANDIR=/location/CANARICAM/PIPELINE
export REDCANDIR
```

alias redcan="csh $REDCANDIR/CSHELL/RedCan.csh"

For cshell users, please edit your $HOME/.cshrc and include:

```
# RedCan
setenv REDCANDIR /location/CANARICAM/PIPELINE
alias redcan "csh $REDCANDIR/CSHELL/RedCan.csh"
```

3 Usage

To display the help of the pipeline please type: *redcan -h*

3.1 Options

The software has to be executed in the directory where observational data live. There are three options to run it:

1. *redcan -d*: Half-Interactive mode.
   It will prompt for the name of the input file, default values are assumed for the rest of parameters.

2. *redcan -i*: Interactive mode.
   All the input parameters must be included when prompted.

3. *redcan parameters*: Command line mode.
   Input parameters have to be included in the command line:
   e.g. redcan infile.lst 0 1 1 0
   (please see below for the parameter space description.)

3.2 Input parameters

The following describes the different choices for the input parameters in the order of appearance. Parameters are key inputs for the analysis, we refer the user to read the pipeline description (Sect. 4) for further information.

1. *Infile*: Text file with the list of names for the FITS files of the observations to process (e.g. infile.lst), including science and calibration files. There is not a default name for this input and it must be created by the user before starting *redcan*, containing one line per file name.

2. *Entry point*: Starting point of the process - useful to skip initial steps when already done. Entry points are:
   (0) identification of files
   (1) flatfield correction & stacking
   (2) image flux calibration

(3) slit losses
(4) wavelength calibration
(5) trace determination
(6) spectral extraction
(7) combining spectra
The default value is 0.

3. *Method*: Selection of the method to determine the width to extract the spectrum:
   (0) fixed aperture along wavelength
   (1) width determined using the trace of the standard.
   The default value is 1.

   Important note: if *Method* is set to 1, the pipeline goes directly to the *Additional stacking parameters* (i.e. items 4 to 6 are skipped).

4. *Target extension*: Selection of the extension of the source.
   Note: If the object is extended, the slit loss correction is not performed.
   (0) point-like source
   (1) extended
   (2) interactive selection.
   The default value is 0.

5. *Target aperture*: Array of one or more values (in pixels) for the radius of the aperture for spectral extraction. If more than one aperture is required, different values have to be comma separated (e.g. 2.5,5,10.5,20). See note in the description of the *Offset* parameter.
   The default value is 1.

6. *Offset*: Array of one or more values for the number of pixels with respect to the target centre to perform the extraction.
   Note: the offset option prevails with respect to the target aperture in such a way that in case that multiple values are given for both parameters, apertures of one size (the first one in the input array) will be extracted for the different offset values.
   The default value is 0 (target).

7. *Additional stacking parameters*: A Y/N keyword is set to use non-default parameters for the stacking procedure. This keyword is needed when the entry point is 0 or 1. If "N" or left empty, default values are considered. If and only if "Y", the following possibilities are prompted:
   – skip flat field correction. Default value is no
   – bad chop sigma value (i.e. remove bad chops above the input

number of sigmas). Default value is 5
– bad-chop checking. Default value is no (i.e. automatic rejection); yes implies manual rejection.
The default value is "N".

When running the pipeline, a file is created in the working directory – date and time of the processing are part of the filename – summarising the set of input parameters used in the processing.

4 Pipeline description

All steps in the processing are called within a procedure aliased as **redcan**, which is executed in the directory containing the data. A flowchart summarising the processing is shown in Fig. 1. Intermediate resulting files will be saved in a subdirectory called PRODUCTS, while the final calibrated spectra plus other files will be saved in a directory called OUTPUTS. To check the entire process a log file called Status_*LISTNAME*.txt will be created in the PRODUCTS directory.

The final spectrum (in fits format) appears in the root directory as Spec_*FILENAME*.fits. The first dimension of the output file is the flux calibrated spectrum, while the second is the spectrum in counts per second.

Each subsection described below explains one step in the data reduction with their possible options (see also Sect. 3), along with the name of the main script used and corresponding outputs.

- Script: CSHELL/RedCan.csh

4.1 Identification of individual observations

First, an input list is provided (*LISTNAME*.lst[8] hereinafter) including all files for the data reduction – e.g. files from one observation night, these can contain one or more science objects. In case of processing of CanariCam data, original filenames are too long to be used within the pipeline and they are renamed to shorter strings. Files are also checked to look for possible problems, and faulty files are removed from the processing. Then, the pipeline identifies each file following two possibilities:

1. Target, standard (PSF or flux) or acquisition.
2. Spectrum or image.

[8] In this manual, an input file named LISTNAME.lst will be used to describe the example. The user can select a filename of their choice, and the resulting files will have the user filename instead of *LISTNAME* but the extensions stated in this document, as the initial extension of the input file will be truncated. Each file in *LISTNAME.lst* will be named here as *FILENAME*.FITS

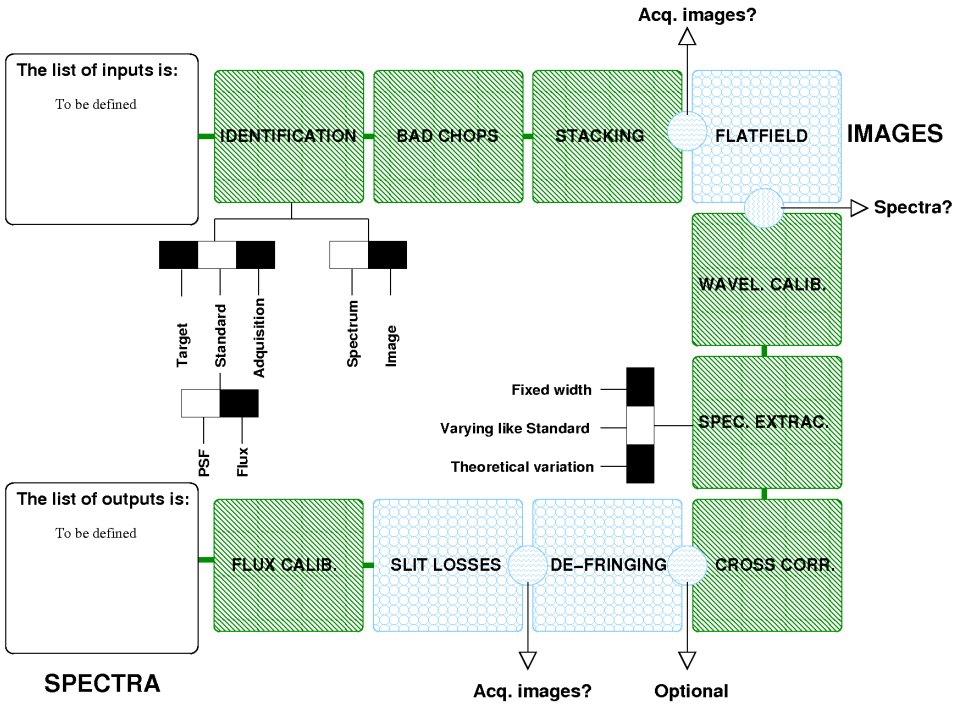

Figure 1: Flowchart of the data reduction process. Green boxes (filled with diagonal lines) are mandatory steps of the pipeline while blue (filled with circles) are optional steps. Black and white small boxes show additional information about the process.

Files will be used in the pipeline according to this classification derived from the FITS header. The outputs of this step will be saved in a text file that can be checked by the user (PRODUCTS/id*LISTNAME*.lst). The acquisition images are indexed to the next observation (standard or target), while the closest standard (in time) is linked to the target. Observations of standards have to be performed using the same slit and grism than the science object.

In addition, a set of key parameters will be extracted or calculated from each FITS header to be used along the code. Files containing this information are saved in PRODUCTS as ID*NUMLISTNAME*.lst (e.g.ID1infile.lst), where *NUM* is an integer from 1 to 4. The columns of the ASCII list for each *NUM* value are detailed in the following:

1. *FILENAME*, OBJECT, OBSTYPE, OBSCLASS, FILTER1, FILTER2, GRATING.

2. *FILENAME*, TIMEOBS, DATEOBS, starting time.

3. *FILENAME*, FRMTIME, FRMCOADD, CHPCOADD, NSAVSETS, NNODS, NNODSETS, exposure, OBJTIME.

4. *FILENAME*, NNODS, NNODSETS, NSAVSETS.

The identification is performed using the OBSCLASS keyword, that must be defined in the header of the observations. Possible options for OBSCLASS are: ACQ, SCIENCE and CALIB.

Note that all parameters are header keywords except "starting time" and "exposure". The "starting time" is determined using TIMEOBS and DATEOBS to order files and to assign acquisition images to the next observation; or a standard to each target. The "exposure" (in secs) is computed as 0.001 $\times$ FRMTIME $\times$ FRMCOADD $\times$ CHPCOADD.

The redshifts of the objects are obtained by accessing to NED automatically (distance independent of redshift if available). If a redshift is not found in the database, its value is automatically set to zero.

- Scripts: CSHELL/ChangeNames.csh – only for CanariCam data
  PYPROC/Checkfits.py
  PYPROC/Obsidentify.py
  IDLPRO/Order.pro
  PYPROC/sigml2.py
- Output: PRODUCTS/id*LISTNAME*.lst
  PRODUCTS/ID*NUMLISTNAME*.lst
  PRODUCTS/redshift_*LISTNAME*.lst

4.2 Flatfield correction

If possible, a normalised (to the median) off-source acquisition image will be created. Images will be divided by this normalised flatfield. If acquisition images are not available for an observation, or they cannot be used due to the observational strategy, the final flatfield-corrected files will just be symbolic links to the corresponding input images.

- Scripts: IDLPROC/Flatfield.pro
  PYPROC/BackAndPrepare.py
  IDLPROC/applyflat.pro
  IDLPRO/FlatCorrect.pro
- Output: PRODUCTS/FL_*FILENAME*.fits
  PRODUCTS/atb_*FILENAME*_ref.ps

4.3 Stacking of individual observations

Images are examined, and output files with any "bad" saveset marked with header keywords in the extension header are created. In case that there

are no "bad" savesets, the output files will be identical to the input files. Savesets are considered as "bad" – therefore excluded from the combined output – based on statistics with respect to a reference frame. Any frame that deviate from the mean by more than a number of sigma – the default option is 5σ – standard deviations is marked as "bad". In that case, the keywords NBADSET and BADSET are written to the extension headers. To preserve the number of savesets in an AB nod pair, the corresponding saveset in the other nod of the pair is also marked as "bad".

These bad savesets can be excluded interactively along the process. The number of sigma for rejected savesets will be added as keyword of the pipeline (not implemented yet).

Then, savesets for each chop position of a given nod position are averaged. Each output file will contain three dimensions: (1) the average on-source, (2) the average off-source, and (3) on-source – off-source images. Resulting images are then rescaled to counts per second using the integration time computed in the previous stage.

- Scripts: PYPROC/Stacking_iraf.py
  IDLPROC/PerSec.pro
- Output: OUTPUTS/stck_*FILENAME*.fits

4.4 Image flux calibration

Flux calibration on images is performed using spectral templates of standards from the flux catalog of Mid-Infrared Standard Stars by Cohen[9] and the appropriate filter pass-band used for the observations (CanariCSam filters are currently implemented).

Aperture photometry of the observed standard is calculated using a radius of 30 pixel. An annular region centred on the source position is used to characterise the background, with inner/outer radii of 50/70 pixel respectively. A flux-calibration correction factor is then calculated dividing the calculated value of the theoretical standard convolved with the filter pass-band by our calculation from the observation. The resulting factor will be included both into the standard and the target FITS headers (keyword FLUXCAL) to obtain flux calibrated images.

- Script: IDLPROC/MainImageFlux.pro
- Output: PRODUCTS/Fluxes_*s**.dat
  OUTPUTS/FC_stck_*FILENAME*.fits
  PRODUCTS/Fluxes_stck.txt

If no spectra are identified in the dataset the procedure will stop here, it will continue otherwise.

[9] http://www.eso.org/sci/facilities/paranal/instruments/visir/tools/zerop_cohen_Jy.txt

4.5 Slit loss calculation

For point-like sources, slit losses are determined using the acquisition spectral images of the source. A slit-loss wavelength-dependent correction is calculated as the ratio between fluxes with and without the source being seen throughout the slit. If acquisition images are not available or the source is defined as extended, the slit loss correction will be set to 1.0. The header of the file will be updated including a new keyword, SLITCORR, with the correction factor. If more than one calibration image - with and/or without slit - exists, the closest to the target is considered here.

- Script: IDLPROC/MainSlitLosses.pro
- Output: OUTPUTS/FC_s*_*FILENAME*.fits - header keyword added

4.6 Wavelength calibration

The wavelength calibration of the images is performed using the combined chop-nod image of the sky. Computing the pixel-to-wavelength conversion factor along the slit involves four steps:

1. Identify features in one dimensional image vectors – collapsing the Y axis.
2. Reidentify features along the spatial direction – collapsing every 10 lines.
3. Find the dispersion of the lines along the spatial direction.
4. Determine a conversion from pixels to wavelength. This is included as a keyword in the header of the object and standard.

- Script: IDLPROC/WLcalib.pro
  PYPRO/WLtransform.py
  IDLPRO/ReAgroupGen.pro
- Output: OUTPUTS/WL_s*_*FILENAME*.fits
  PRODUCTS/WL_s*_*FILENAME*.ps

4.7 Source and standard star spectral extraction

The trace of the standard is determined by fitting data to a Moffat profile, which this is a better description than that of a Gaussian (see Radomski et al. 2008, ApJ, 681, 141, for discussion), every 10 pixel along the dispersion axis to look for the centre and the width of each spectrum. Centres of the Moffats are used to fit a polynomial expression of order 3 to take into

account that the spectrum might be curved. Then, the region to be extracted is traced along the dispersion axis. Output products include the Moffat profiles used for the extraction

The extraction aperture of the source can be selected as :

– Method = 0: fixed aperture – useful for extended sources

– Method = 1: wavelength dependent aperture – i.e. using as source aperture the width of the Moffat profile used to derive the trace of the standard (or a multiple of that value).

The latter option is recommended, as the extraction aperture should increase with the wavelength. It should follow the trace of the standard due to the resolution power of the instrument. An offset with respect to the target of the observation for a extraction region is also allowed.

- Script: IDLPROC/XtractMain.pro
- Output: PRODUCTS/Moff_stck_*FILENAME*.ps

4.8 Absolute flux calibration of the spectrum

Standard star calibration measurements are used to determine the system sensitivity as a function of wavelength. This is done by comparing the theoretical standard spectrum to the observed data and taking the SLITLOSS correction into account.

To obtain a flux-calibrated science spectrum, the source spectrum is multiplied by the slit correction and the sensitivity previously derived. Finally, this step aims to converts the final spectrum to the units of interest (mJy in this case) and save it as ASCII and postcript formats.

In the output files, the method used for spectral extraction is written in the filename, as *m0* or *m1* for fixed or wavelength-dependent apertures respectively. Files with the *logs* string in their name contain one page per science exposure with three panels, these are:

– Upper panel: observed standard associated with the science exposure.

– Middle panel: sensitivity function - i.e. theoretical/observed standard spectrum.

– Lower panel: calibrated science exposure - science spectra from previous steps times the sensitivity function.

- Script: IDLPROC/XtractMain.pro – cont.
- Output: OUTPUTS/Spec_*method*_s*_*FILENAMESCIENCE*.dat
  OUTPUTS/Cal_*method*_stck_*FILENAMESTANDARDS*.dat
  PRODUCTS/Specs_*method*_s*_logs.ps

4.9 Combination of multiple exposures

Spectra from each science image have been individually extracted and they are then combined in this step. A file called List_Spec_positions.lst is created in the PRODUCTS directory. It contains a list name for each source that include the corresponding files to combine. Files are combined using the median. The spectrum is shifted to the rest-frame of the source using the redshift obtained in Sect. 4.1.

The output is an ASCII file per science object with the following columns: wavelength (in Å), flux (in Jy), flux error (in Jy) and spectrum in counts (i.e. before the flux calibration). A postcript file with the flux calibrated spectra is saved in the PRODUCTS directory with the positions of the most prominent IR emission lines marked in the spectrum.

- Script: IDLPROC/MeanSpectra.pro
- Output: OUTPUTS/Spec_method_s*_*SOURCENAME*.dat
  PRODUCTS/Spec_method_s*_*SOURCENAME*.ps