
IRDAP Documentation

Release 1.3.4

R.G. van Holstein

May 02, 2022

Contents

1	Contents
----------	-----------------

3

IRDAP (IRDIS Data reduction for Accurate Polarimetry) is a highly-automated end-to-end pipeline to reduce SPHERE-IRDIS polarimetric data using polarimetric differential imaging (PDI). Its core feature is the model-based correction method of the instrumental polarization effects as described in van Holstein et al. (2020). IRDAP handles data taken both in field- and pupil-tracking mode and using the broadband filters Y, J, H and K_s. Data taken with the narrowband filters can be reduced as well, although with a somewhat worse accuracy. For pupil-tracking observations IRDAP can additionally apply angular differential imaging.

Reducing data with IRDAP is very straightforward and does not require the user to do any coding or have knowledge of Python (IRDAP is written for Python 3.6 and 3.7). IRDAP is simply run from a terminal with only a few commands and uses a configuration file with a limited number of input parameters. Within several minutes, IRDAP performs a complete data reduction from raw data to final data products.

Note: If you use IRDAP for your publication, please *cite our papers*.

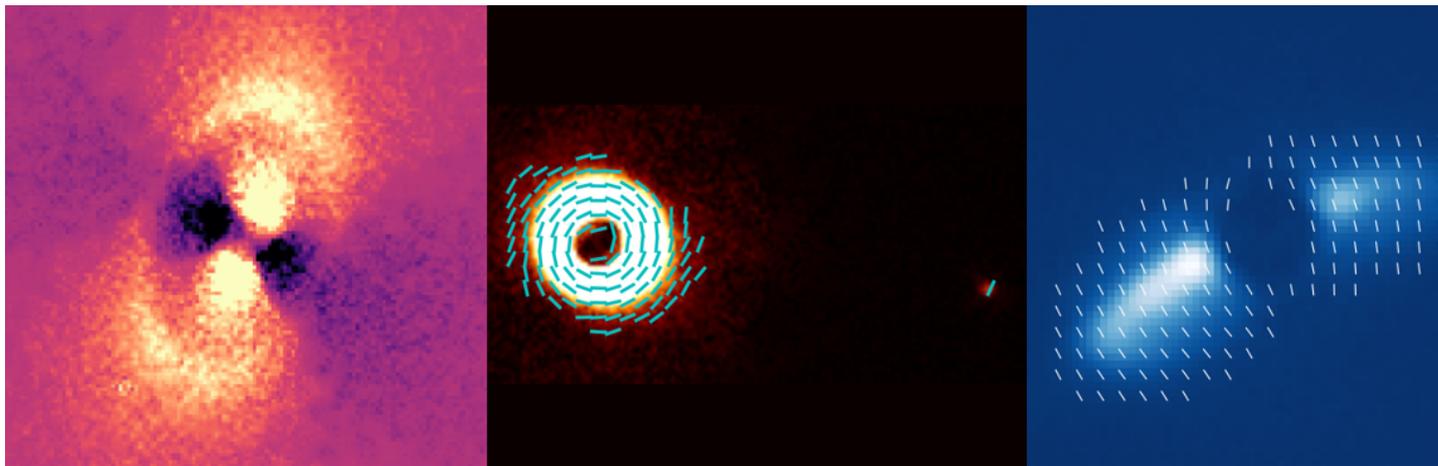


Fig. 1: Three examples of data sets reduced with IRDAP: the circumstellar disk of PDS 70 including its inner disk that polarizes the central star (Keppler et al. 2018), the circumstellar disk and polarized companion of CS Cha (Ginski et al. 2018) and the circumstellar disk of T Cha (Pohl et al. 2017; van Holstein et al. 2020).

IRDAP yields a multitude of improvements for polarimetric observations of circumstellar disks: it enables us to accurately study the morphology of disks, measure non-azimuthal polarization and determine scattering phase functions and particle properties. Because IRDAP discerns instrumental polarization from stellar polarization, it is a vital tool for accurate radiative transfer modeling of disks and enables the detection of unresolved (inner) disks and the measurement of the polarization of substellar companions. Finally, IRDAP enables accurate data reduction for targets that cannot be reduced with the conventional data-reduction methods (because there is no bright star in the field of view), such as solar system objects.

1.1 Installation

To run IRDAP the user needs to have Python 3.6 or 3.7 installed.

1.1.1 Installing IRDAP

IRDAP is available on [PyPI](#) and can be installed by simply typing in a terminal:

```
pip install irdap
```

Because IRDAP is being actively developed, it is recommended to regularly check for updates through:

```
pip install irdap --upgrade
```

and install the latest version. To be kept up-to-date via e-mail, sign up to the [mailing list](#).

The most recent release of IRDAP is also available for download from the [GitHub repository](#).

1.1.2 Testing the installation

In the terminal, try to access the basic help and the version of IRDAP:

```
irdap --help  
irdap --version
```

If that works, IRDAP is most likely installed correctly.

As a next step, it is recommended to run the [example reduction](#).

1.2 Example reduction

As an introduction to the user, but also to check if IRDAP is installed correctly, a quick end-to-end example reduction can be performed. This demo uses a small portion of the H-band polarimetric data of the circumstellar disk of T Cha from ESO Program 096.C-0248(C). This data set was the first to be reduced with IRDAP's correction method and is published in Pohl et al. (2017). It is also used in van Holstein et al. (2020) to exemplify the correction method.

To run the demo, change the current directory of the terminal to a directory of your choice and type:

```
irdap --demo
```

IRDAP then performs a complete data reduction and writes the resulting files to the terminal's current directory.

Note: An internet connection is required to run the demo, because 56.6 MB of raw data needs to be downloaded from the [GitHub repository](#).

After running the demo, you can navigate to the `reduced_pdi/no_star_pol_subtr` subdirectory and open the files `T_Cha_2016-02-20_Q_phi.fits` and `T_Cha_2016-02-20_U_phi.fits` that show the final Q_ϕ - and U_ϕ -images. Opening these files with **DS9**, setting color to cool, scale to linear and the limits to -500 to 1500 and -300 to 150, respectively, you will see the following:

These images correspond to Figure 14 (left column) of van Holstein et al. (2020), but with lower signal-to-noise because for the demo the data of only 1 instead of 30 HWP cycles was used.

To start reducing your own data with IRDAP, continue to the *usage instructions*.

1.3 Usage instructions

The usage instructions are divided into five sections:

- *Running IRDAP*
- *Pre-processing in a nutshell*
- *PDI in a nutshell*
- *ADI in a nutshell*
- *Combining multiple data sets*

1.3.1 Running IRDAP

Reducing data with IRDAP is very straightforward and only requires two commands to be entered in the terminal. First, create a directory (e.g. `/home/T_Cha_2016-02-20`) containing a subdirectory called `raw` (i.e. `/home/T_Cha_2016-02-20/raw`) in which you place the raw FITS-files.

Attention: The raw FITS-files should have *DPR TYPE* equal to OBJECT, OBJECT,CENTER, OBJECT,FLUX or SKY. If the user wishes to make a new master flat and bad pixel map (see *Pre-processing in a nutshell*) *DPR TYPE* can also be equal to FLAT,LAMP, DARK,BACKGROUND or DARK. FITS-files with a *DPR TYPE* other than the ones mentioned above will automatically be ignored.

Hint: To only download the raw data that is required for IRDAF, use the [SPHERE data archive](#) and set the search field *DPR TECH* to *any POLARIMETRY*.

In the terminal navigate to the main directory (e.g. `cd /home/T_Cha_2016-02-20`) and type:

```
irdap --makeconfig
```

This creates a default configuration file `config.conf` in the main directory. You can then adjust the parameters in the configuration file with a text editor (see [Configuration file](#)). Finally, in the terminal type:

```
irdap --run
```

to perform the data reduction.

Note: In general a first reduction can be performed without making changes to the configuration file. After this first reduction, some of the input parameters can be adjusted and IRDAF can be run again (but this is often not necessary!).

1.3.2 Pre-processing in a nutshell

The data reduction is divided into a pre-processing, polarimetric differential imaging (PDI) and angular differential imaging (ADI) part.

Warning: The data-reduction steps are only described concisely here. The log file that IRDAF creates describes the steps in more detail and also briefly describes the output files created. It is therefore recommended to carefully read the log file the first time you run IRDAF.

The pre-processing starts with reading the configuration file and creating the log file that contains all the lines printed on screen. In addition, IRDAF creates overviews of the relevant header keywords for each file in the `raw` subdirectory, sorts the raw data and creates subdirectories to write the output files to.

Hints

- The header overviews can also be created without running IRDAF by typing `irdap --headers` in the terminal;
 - When reducing data with IRDAF, keep an eye out for the WARNING messages printed on screen and in the log file;
 - When running IRDAF a copy of the configuration file is made. When subsequently re-running IRDAF, this copy is moved together with the log file to the `logs` subdirectory as a record of the performed reductions.
 - In case some steps of the reduction are skipped when re-running IRDAF (i.e. setting `perform_preprocessing`, `perform_pdi` and/or `perform_adi` equal to `False` in the [configuration file](#)), the sections pertaining to these skipped reduction steps are copied from the previous to the new log file and configuration file. This way the latest log file and configuration file always contain all relevant information and input parameters of the reduction steps taken.
-

After these initial steps, IRDAF will pre-process the OBJECT-files. To this end, it loads the static bad pixel map and the static master flat of the right filter, and creates a master sky frame from the provided SKY-files. The OBJECT-files

are then sky (or background) subtracted, flat fielded, bad-pixel filtered and centered with a method chosen by the user. The centering would generally be performed using the CENTER-files, which are then processed accordingly.

Note: Rather than using the static bad pixel map and master flat, the user can also create a master flat and bad pixel map by including a sequence of FLAT,LAMP- and DARK,BACKGROUND-files (or DARK-files) in the `raw` subdirectory (e.g. a sequence with exposure times 1, 2, 3, 4 and 5 s). The FLAT,LAMP-files preferably have the P0-90 polarizer set inserted, because it causes strong vignetting at the edges of the field of view which will otherwise not be corrected.

A cube of master flux frames is created for the left and right detector halves by processing the FLUX-files in a similar fashion as the OBJECT-files. If the data contains SKY-files with the same exposure time and neutral density filter as the FLUX-files, IRDAP processes these to subtract the sky background from the master flux frames.

Hint: IRDAP automatically determines the reference fluxes from the master flux frames and writes them to a CSV-file. These reference fluxes can be used to convert the final images produced by IRDAP (e.g. the I_Q - or Q_ϕ -images, or the images produced through ADI) from units of counts (ADU) into units of contrast/arcsec². If the user can determine the stellar flux in Jansky, the final images can be expressed in Jansky/arcsec². See the log file created by IRDAP for more details.

The pre-processed OBJECT-data is written to the subdirectory `preprocessed` and the processed SKY-, CENTER- and FLUX-data (and the user-created bad pixel map and master flat) to the subdirectory `calibration`.

Important: Always check the FITS-files and figures that IRDAP writes to the subdirectories `calibration` and `preprocessed`. In case the pre-processed OBJECT- or FLUX-data is not correctly centered, IRDAP should be run again after removing images of bad quality and/or adapting the *Advanced pre-processing options* of the configuration file.

1.3.3 PDI in a nutshell

For the polarimetric differential imaging (PDI) part, IRDAP computes the double sum and double difference from the pre-processed OBJECT-files. It then applies the model-based correction method as described in [van Holstein et al. \(2020\)](#) to remove the instrumental polarization and cross-talk. The correction method for pupil-tracking observations has some differences compared to that of field-tracking observations (see [van Holstein et al. 2017](#)). Subsequently, the background in the images is subtracted and the polarization of the star determined. The FITS-files of the final images are then written to two subdirectories:

- `reduced_pdi\no_star_pol_subtr`, containing the final images with the polarization of the star still present;
- `reduced_pdi\star_pol_subtr`, containing the final images with the polarization of the star subtracted.

Finally, if FLUX-files are included in the raw data, IRDAP creates contrast curves for the detection of polarized extended sources and polarized point sources.

Important: By default, the polarization of the star is determined with an annulus over the AO residuals. However, for the most accurate results the annulus should only contain signal from the star, and no signal from for example a circumstellar disk, companion or background star. Therefore the user often needs to adjust the input parameter `annulus_star` in the configuration file.

Important: The possibility to measure the polarization of the central star and the ability to create final images with and without this stellar polarization is a big advantage of IRDAP (see [van Holstein et al. 2020](#)). For images of a star or circumstellar disk for example, the stellar polarization can indicate the presence of an unresolved (inner) disk if it can be proven (or reasonably expected) that the polarization does not originate from interstellar dust. In that case one would use the images with the star polarization still present when making a comparison with radiative transfer models (see e.g. [Keppler et al. 2018](#)). Measuring the polarization of the star is also vital when measuring the polarization of substellar companions.

Warning: For targets without a bright star (e.g. solar system objects), one would always use the images in the subdirectory `reduced_pdi\no_star_pol_subtr`, i.e. those without the polarization subtracted.

Note: The units of the I_Q -, I_U - and I_{tot} -images are the number of counts (ADU) when summing the left and right frame halves of a single exposure (a single DIT). The images are averaged over the NDIT and the number of FITS-files used. Similarly, the units of the Q -, U -, Q_ϕ - and U_ϕ -images are the number of counts when subtracting the right from the left frame half of a single exposure.

Note: The final PDI-images are corrected for the true north offset.

1.3.4 ADI in a nutshell

With the angular differential imaging (ADI) part, IRDAP enables the detection of companions and disk signal in total intensity by performing both a classical ADI reduction and a reduction combining ADI with principal component analysis (ADI+PCA). IRDAP performs separate reductions for the frames on the left and right halves of the detector. The resulting images, plus an image of their sum, are written to the subdirectories `reduced_adi\classical` and `reduced_adi\pca`. For the ADI+PCA reduction, the user can set the number of principal components to be subtracted and the annuli over which the reduction needs to be optimized (see the *Basic ADI options* of the configuration file).

Warning: IRDAP does not calibrate or correct the flux self-subtraction induced by ADI. This should be kept in mind when converting the final ADI-images from units of counts (ADU) into units of contrast/arcsec² or Jansky/arcsec². Self-subtraction can be especially problematic when the total parallactic rotation of a data set is limited.

Hint: The names of the FITS-files produced by the ADI+PCA reduction indicate the number of principal components subtracted and the annuli used to optimize the reduction over. For example `com_2-4_rad_10-30-100-512` in a file name indicates that two frames are created with 2 and 4 principal components subtracted (*com*), and that three annuli were used with inner and outer radii equal to 10 and 30, 30 and 100, and 100 and 512 pixels (*rad*). When repeating the ADI reduction with different values of the input parameters *principal_components* and *pca_radii* in the configuration file, any previously created FITS-files are retained.

Note: The units of the summed ADI-images are the number of counts (ADU) when summing the left and right frame halves of a single exposure (a single DIT) and averaging over the NDIT and the number of FITS-files used. Therefore

the units of the summed ADI-images are the same as that of the I_Q -, I_U - and I_{tot} -images produced by the *PDI part* of IRDAP.

Note: The final ADI-images are corrected for the true north offset.

1.3.5 Combining multiple data sets

If a target was observed using multiple observation blocks (OBs), it is recommended to first reduce each OB separately. After that, the final images of the PDI and ADI reductions can be mean-combined by using the terminal to navigate to a directory of your choice and typing:

```
irdap --meancombine path1 path2 ... pathx
```

The space-separated paths are absolute paths to the main directories of the reductions, e.g.:

```
irdap --meancombine /home/T_Cha_2016-02-20 /home/T_Cha_2016-02-21
```

The mean-combined images will be written to the current working directory of the terminal.

To understand the input parameters, continue with the *configuration file*.

1.4 Configuration file

The configuration file with the input parameters for IRDAP is shown below:

```
# For a detailed explanation of each input parameter and
# the options of the configuration file, see:
# https://irdap.readthedocs.io/en/latest/configfile.html

[Basic pre-processing options]

perform_preprocessing      = True
sigma_filtering           = True
object_collapse_ndit     = False
object_centering_method   = automatic
frames_to_remove          = []

[Basic PDI options]

perform_pdi                = True
annulus_star               = automatic
annulus_background        = large annulus
normalized_polarization_images = False

[Basic ADI options]

perform_adi                = True
principal_components      = companion+disk
pca_radii                 = automatic

[Advanced pre-processing options]
# Generally do not need to be changed
```

(continues on next page)

(continued from previous page)

```

center_subtract_object      = True
center_param_centering      = (12, None, 30000)
object_center_coordinates   = automatic
object_param_centering      = (60, None, 30000)
flux_centering_method       = gaussian
flux_center_coordinates     = (478, 522, 1504, 512)
flux_param_centering        = (60, None, 30000)
flux_annulus_background     = large annulus
flux_annulus_star           = automatic

[Advanced PDI options]
# Generally do not need to be changed

double_difference_type      = conventional
single_posang_north_up     = True
combination_method_polarization = least squares
combination_method_intensity = mean

```

The input parameters are divided into five groups:

- *Basic pre-processing options*
- *Basic PDI options*
- *Basic ADI options*
- *Advanced pre-processing options*
- *Advanced PDI options*

Below we discuss for each parameter of the configuration file what it does and what the valid input options are.

Important: While the user will quite frequently have to adapt the basic options, the advanced options generally do not need to be changed.

1.4.1 Basic pre-processing options

perform_preprocessing:

True, False (default = True)

If True, perform *pre-processing* on the raw data, i.e. prepare the data for the subsequent *polarimetric differential imaging (PDI)* and/or *angular differential imaging (ADI)* steps. *perform_preprocessing* must be True the first time IRDAP is run on a particular data set. When re-running IRDAP on the same data set, *perform_preprocessing* can be set to False to skip the pre-processing and save time when tweaking the input parameters of the PDI and/or ADI steps.

sigma_filtering:

True, False (default = True)

If True, use sigma-filtering to remove bad pixels that are still in the OBJECT-, CENTER- and FLUX-frames after applying the master bad pixel map. After the sigma-filtering the frames should have no bad pixels. If False, only use the master bad pixel map to speed up the reduction. The latter is not recommended for a final reduction however, because a significant number of bad pixels will remain in the frames.

object_collapse_ndit:

True, False (default = False)

If `True`, compute the mean over the (NDIT) frames of the OBJECT-files before sky (or background) subtraction, flat-fielding, bad pixel removal and centering to speed up the pre-processing. If `False`, perform the above steps for each frame and then compute the mean over the frames. The latter will result in the most accurate reduction and is recommended for a final reduction.

object_centering_method:

`automatic, center frames, gaussian, cross-correlation, manual (default = automatic)`

Method to center the OBJECT-frames. If `center frames`, center the OBJECT-frames with the center coordinates determined from the CENTER-files. If `gaussian`, center the OBJECT-frames by fitting a 2D Gaussian to each frame. If `cross-correlation`, perform the centering by fitting a 2D Gaussian to the first frame and then using cross-correlation to align (register) the other frames onto the centered first frame. Generally `gaussian` yields more accurate results than `cross-correlation`. For `gaussian` and `cross-correlation` the determined center coordinates are plotted for each image. If `manual`, use fixed coordinates as defined with `object_center_coordinates` to center the OBJECT-frames. If `automatic`, center the OBJECT-frames using the CENTER-files if they exist, and fit a 2D Gaussian to each OBJECT-frame if there are no CENTER-files.

Attention: In case the centering is not accurate, the advanced parameters `center_subtract_object`, `center_param_centering`, `object_center_coordinates` and `object_param_centering` can be adapted. In addition, if the shifts among the frames are small when centering using `gaussian` or `cross-correlation`, the quality of the final images can sometimes be improved by repeating the pre-processing with `object_centering_method` set to `manual` and `object_center_coordinates` equal to the mean center coordinates found earlier with the Gaussian fitting or cross-correlation (see the log file).

frames_to_remove:

`list (default = [])`

List of integers and length-2-tuples of integers indicating which files and frames not to use for the data reduction. This parameter allows the user to remove frames of bad quality and to exclude complete files from the reduction without (re)moving them from the raw subdirectory. A complete file can be removed by specifying its integer index, while a frame of a specific file can be removed by specifying a tuple (`file_index, frame_index`). For example `[2, 4, (5, 3)]` will exclude files 2 and 4 from the reduction and will remove frame 3 of file 5. If no files or frames should be removed, use an empty list `[]`. The files are sorted in chronological order from oldest to newest. The file indices of each file are shown in the header overview that is created when you run IRDAF.

Attention: The indices of the files and frames are 1-based, i.e. the first file or frame has index 1.

Hint: To check for frames of bad quality or frames that are not centered correctly, the user can examine the sub-images of the (pre-)processed OBJECT- and FLUX-frames that IRDAF writes to the `preprocessed` and `calibration/flux` subdirectories. Because the file and frame indices are displayed next to the sub-images, the bad frames can straightforwardly be removed by inserting the corresponding indices in the list of `frames_to_remove`.

1.4.2 Basic PDI options

perform_pdi:

`True, False (default = True)`

If True, perform *polarimetric differential imaging (PDI)* using the pre-processed data. To perform this step, *perform_preprocessing* must be True or the raw data must have been pre-processed before. If False, do not perform PDI. The latter can be useful when tweaking the input parameters of the ADI step only.

annulus_star:

automatic, ao residuals, star aperture, *list, tuple* (default = automatic)

Annulus used to measure the polarization of the central star (or any other source in the field of view) from the Q -, I_Q -, U -, and I_U -images. This measured polarization signal is subtracted from the polarization images written to the subdirectory `reduced_star_pol_subtr`. For the most accurate results the annulus should only contain signal from the star, and no signal from for example a circumstellar disk, companion or background star. The measured polarization signal is affected by the subtraction of the background in the images (see *annulus_background*).

If `ao residuals`, the annulus will be centered on the central star and be located over the AO residuals. The inner and outer radius of the annulus will automatically be adapted to the filter used. If `star aperture`, the star polarization will be determined from a small aperture with a radius of 11 pixels located at the position of the central star. If `automatic`, an annulus over the AO residuals (as for `ao residuals`) will be used in case of coronagraphic data, and a small aperture at the position of the central star (as for `star aperture`) in case of non-coronagraphic data.

Generally the user would do a first data reduction using `automatic`. If after this first reduction it appears that more control over the exact shape of the annulus is required, the user can define the annulus with a tuple of 6 parameters:

- *coord_center_x*: x-coordinate of center (pixels)
- *coord_center_y*: y-coordinate of center (pixels)
- *inner_radius*: Inner radius (pixels)
- *outer_radius*: Outer radius (pixels)
- *start_angle*: Start angle of annulus sector (deg)
- *end_angle*: End angle of annulus sector (deg)

For example `(512.5, 512.5, 60, 95, 0, 360)` will create an annulus that is centered on the central star and is located over the AO residuals in H-band.

Attention: IRDAP uses the same 1-based indexed coordinates as DS9. Therefore coordinates read from DS9 can be used for the configuration file without applying any conversion.

IRDAP centers the frames at the coordinates $(x, y) = (512.5, 512.5)$. Therefore a centered annulus will have *coord_center_x* and *coord_center_y* equal to 512.5.

Attention: *start_angle* and *end_angle* are defined with respect to the orientation of the final images with 0 deg oriented to the right and positive angles rotating counterclockwise. Therefore an angle of 0 deg points to the West, except when *single_posang_north_up* is set to `False` for observations taken in field-tracking mode with a single derotator position angle.

The definition of the angles is the same as used in DS9. When using DS9 to draw a line starting at the center coordinates, the angle indicated can be used for the configuration file without applying any conversion.

If even more control over the shape of the annulus is required, the user can define a list of length-6-tuples. This is for instance useful to exclude signal of a circumstellar disk from the annulus. `[(512.5, 512.5, 60,`

95, 105, 255), (512.5, 512.5, 60, 95, 285, 75)] will for example create an annulus with a gap at the top and bottom each spanning 30 deg.

annulus_background:

large annulus, *list, tuple* (default = large annulus)

Annulus used to measure and remove the background in the Q -, I_Q -, U -, and I_U -images. This background subtraction also affects the computation of the polarization of the central star (see *annulus_star*). For the most accurate results the annulus should not contain any signal from the star or any other source in the field of view.

If `large annulus`, the annulus will be centered on the central star and be located far away from the star with an inner radius of 360 pixels and an outer radius of 420 pixels. In case more control over the exact shape of the annulus is required, the user can define the annulus with a (list of) length-6-tuple(s) in the same way as for *annulus_star*.

normalized_polarization_images:

True, False (default = False)

If `True`, create final images of degree of linear polarization, normalized Stokes q and u (see van Holstein et al. 2020) and degree and angle of linear polarization computed from q and u :

- $DoLP = \sqrt{Q^2 + U^2} / [0.5 (I_Q + I_U)]$
- $q = Q / I_Q$
- $u = U / I_U$
- $AoLP_norm = 0.5 \arctan(u / q)$
- $DoLP_norm = \sqrt{q^2 + u^2}$

These additional final images are only valid if all flux in the images originates from the source of interest. This is generally the case for observations of for example solar system objects or galaxies. The images are generally not valid for observations of circumstellar disks or companions because in that case a large part of the flux in the total intensity images originates from the central star.

$DoLP_norm$ and $AoLP_norm$ are potentially more accurate than $DoLP$ (above) and $AoLP = 0.5 \arctan(U / Q)$ (as always created), especially when there are significant variations in seeing and sky transparency among the measurements. If `False`, do not create these images.

1.4.3 Basic ADI options

perform_adi:

True, False (default = True)

If `True`, perform *angular differential imaging (ADI)* using the pre-processed data. ADI is only performed on data taken in pupil-tracking mode. To perform this step, *perform_preprocessing* must be `True` or the raw data must have been pre-processed before. If `False`, do not perform ADI. The latter can be useful when tweaking the input parameters of the PDI step only.

principal_components:

companion+disk, companion, disk, *integer, list* (default = companion+disk)

The number of principal components to subtract for the reduction combining angular differential imaging (ADI) and principal component analysis (PCA). If `companion`, create two frames with 10 and 16 principal components subtracted to search for companions. If `disk`, create two frames with 2 and 4 principal components subtracted to search for disk signal. If `companion+disk`, create four frames with 2, 4, 10 and 16 principal components subtracted to search for companions and disk signal.

If more control is required, a strictly positive integer or a list of unique and strictly positive integers can be provided. For example providing [10, 16] will yield two frames with 10 and 16 principal components

subtracted. Because we cannot remove more components than the number of frames in the cube of pre-processed frames, the number of components to be subtracted is reduced if there are not enough frames.

pca_radii:

automatic, list (default = *automatic*)

Annuli used to optimize the ADI+PCA reduction over. If *automatic*, the reduction is optimized over three annuli with inner and outer radii equal to 10 and 30, 30 and 100, and 100 and 512 pixels. If more control is required, the user can provide a list of at least length 2 containing positive and increasing integers (including 0) indicating the inner and outer radii of the annuli in pixels. For example, if the list has two elements, these elements define the inner and outer radius of a single annulus. If the list has more elements, the second element defines the outer radius of the first annulus and the inner radius of the second annulus. Likewise the third element defines the outer radius of the second annulus and the inner radius of the third annulus, etc. Therefore setting *pca_radii* to `[10, 30, 100, 512]` is equivalent to setting it to *automatic*.

1.4.4 Advanced pre-processing options

center_subtract_object:

True, False (default = *True*)

If *True*, subtract the OBJECT-file(s) taken closest in time from the CENTER-file(s) before determining the center coordinates from the satellite spots. This results in a more accurate determination of the center coordinates because the background and any other sources in the field of view are suppressed. When the difference between the image orientations of the CENTER- and OBJECT-frames is large (i.e. difference in derotator position angle for field-tracking and difference in parallactic angle for pupil-tracking), the OBJECT-frame will be rotated around the initial guess of the centers as defined by *object_center_coordinates* before subtracting it from the CENTER-frame. If *False*, do not subtract the OBJECT-file(s). Parameter is ignored if no CENTER-files are used for the centering (see *object_centering_method*).

center_param_centering:

tuple (default = `(12, None, 30000)`)

Tuple of 3 parameters pertaining to the fitting of the satellite spots of the the CENTER-frames with 2D Gaussians to determine the center coordinates. In case the center coordinates are not accurately determined, the 3 parameters can be adapted in addition to *object_center_coordinates*:

- *crop_radius*: Half the length of the sides of the square cropped sub-images that should each contain a satellite spot and are used to fit the 2D Gaussians to (pixels). Must be integer. *object_center_coordinates* gives the initial guess of the center coordinates to place the sub-images approximately at the locations of the satellite spots. If *None*, the complete frame is used for the fitting and *object_center_coordinates* is ignored.
- *sigfactor*: All sub-image pixels with values smaller than *sigfactor**standard deviation are replaced by random Gaussian noise to mask them for fitting the 2D Gaussian. This can be useful when there is a lot of noise in the background. If *None*, no pixels are replaced by Gaussian noise.
- *saturation_level*: All pixels within the smallest circle encompassing the pixels with a value equal to or higher than *saturation_level* are ignored when fitting the 2D Gaussian. Ignoring the saturated pixels improves the accuracy of the fit. We use a circle because strongly saturated pixels in the peak of the PSF often have values lower than *saturation_level*. If *None*, no pixels are ignored.

center_param_centering is ignored if no CENTER-files are used for the centering (see *object_centering_method*).

object_center_coordinates:

automatic, tuple (default = *automatic*)

Tuple of 4 floats with (initial guess of) center coordinates of OBJECT-frames:

- *x_left*: x-coordinate of center of left frame half (pixels)
- *y_left*: y-coordinate of center of left frame half (pixels)
- *x_right*: x-coordinate of center of right frame half (pixels)
- *y_right*: y-coordinate of center of right frame half (pixels)

If *automatic*, *object_center_coordinates* is set to (478, 535, 1504, 524) when the coronagraph *N_ALC_Ks* is used and to (478, 522, 1504, 512) otherwise.

Attention: IRDAP uses the same 1-based indexed coordinates as DS9. Therefore coordinates read from DS9 can be used for the configuration file without applying any conversion.

The center coordinates are defined in the complete frame, i.e. with both detector halves. Therefore *x_right* has a value larger than 1024 pixels.

The center coordinates of *object_center_coordinates* are used at various steps of the pre-processing:

- During the processing of the CENTER-files, they give the coordinates to rotate the OBJECT-frame around before subtracting it from the CENTER-frame (see *center_subtract_object*).
- When determining the center coordinates from the CENTER-frames, they give the initial guess of the center coordinates to place the sub-images approximately on the location of the satellite spots (see *center_param_centering*).
- They give the initial guess of the center coordinates when centering the OBJECT-frames by fitting 2D Gaussians or using cross-correlation (see *object_centering_method* and *object_param_centering*).
- In case *object_centering_method* is *manual*, they are the actual center coordinates used to center the OBJECT-frames.

object_param_centering:

tuple (default = (60, None, 30000))

Tuple of 3 parameters pertaining to the centering of the OBJECT-frames by fitting a 2D Gaussian to each frame or using cross-correlation. In case the centering is not performed accurately, the 3 parameters can be adapted in addition to *object_center_coordinates* (especially useful for non-coronagraphic data of multiple stars):

- *crop_radius*: Half the length of the sides of the square cropped sub-images used to fit the 2D Gaussians to and used for cross-correlating the images (pixels). Must be integer. The sub-image is centered on the coordinates as provided by *object_center_coordinates*. If *None*, the complete frame is used and *object_center_coordinates* is ignored.
- *sigfactor*: see *center_param_centering*.
- *saturation_level*: see *center_param_centering*.

object_param_centering is only used when *object_centering_method* is *gaussian* or *cross-correlation*.

flux_centering_method:

gaussian, *manual* (default = *gaussian*)

Method to center the FLUX-frames. If *gaussian*, fit a 2D Gaussian to each frame. The determined center coordinates are plotted for each image. If *manual*, use fixed coordinates as defined with *flux_center_coordinates*. In case the centering is not accurate, the parameters *flux_center_coordinates* and *flux_param_centering* can be adapted.

flux_center_coordinates:

tuple (default = (478, 522, 1504, 512))

(Initial guess of) center coordinates of FLUX-frames, defined in the same way as *object_center_coordinates*. The center coordinates are the initial guess when centering the FLUX-frames by fitting 2D Gaussians (see *flux_centering_method* and *flux_param_centering*). In addition, in case *flux_centering_method* is `manual`, the center coordinates are the actual center coordinates used to center the FLUX-frames.

flux_param_centering:

`tuple` (default = (60, None, 30000))

Tuple of 3 parameters pertaining to the centering of the FLUX-frames by fitting a 2D Gaussian to each frame. In case the centering is not performed accurately, the 3 parameters can be adapted in addition to *flux_center_coordinates* (especially useful in case of multiple stars):

- *crop_radius*: Half the length of the sides of the square cropped sub-images used to fit the 2D Gaussians to (pixels). Must be integer. The sub-image is centered on the coordinates as provided by *flux_center_coordinates*. If `None`, the complete frame is used and *flux_center_coordinates* is ignored.
- *sigfactor*: see *center_param_centering*.
- *saturation_level*: see *center_param_centering*.

flux_param_centering is only used when *flux_centering_method* is `gaussian`.

flux_annulus_background:

`large annulus, list, tuple` (default = large annulus)

Annulus used to remove the background in the FLUX-frames. For the most accurate results the annulus should not contain any signal from the star or any other source in the field of view. If `large annulus`, the annulus will be centered on the central star and be located far away from the star with an inner radius of 320 pixels and an outer radius of 380 pixels. In case more control over the exact shape of the annulus is required, the user can define the annulus with a (list of) length-6-tuple(s) in the same way as for *annulus_star*.

flux_annulus_star:

`automatic, list, tuple` (default = automatic)

Annulus used to measure the total flux of the star in the master flux frames. The total fluxes are converted into reference fluxes that can be used to express the final images (e.g. the I_Q - or Q_ϕ -images) in units of contrast/arcsec² or Jansky/arcsec². The annulus should contain all signal from the central star, but no signal from for example a (stellar) companion or background star.

If `automatic`, the annulus will be an aperture of radius 120 pixels located at the position of the central star. In case more control over the exact shape of the annulus is required, for example to exclude an object close to the central star, the user can define the annulus with a (list of) length-6-tuple(s) in the same way as for *annulus_star*.

1.4.5 Advanced PDI options

double_difference_type:

`conventional, normalized` (default = conventional)

Type of double difference to be computed. In almost all cases one would use `conventional`. When there are large variations in atmospheric seeing and/or sky transparency among the measurements (often resulting in a large spread of the measured star polarization as a function of HWP cycle number), using `normalized` can suppress spurious polarization signals and improve the quality of the final images (see van Holstein et al. 2020).

single_posang_north_up:

`True, False` (default = True)

For observations taken in field-tracking mode with a single derotator position angle (header keyword `INS4.DROT2.POSANG` either 0 or with a fixed offset), the final images are rotated with North up if `True`, and kept in the orientation of the raw frames if `False`. In the latter case, the images are more accurate as they suffer less from interpolation errors caused by image rotation. This is useful when for example extracting the polarized

surface brightness distribution of a circumstellar disk. Parameter is ignored for field-tracking observations with more than one derotator position angle or observations taken in pupil-tracking mode. In these cases the final images produced are always oriented with North up.

combination_method_polarization:

least squares, median, *float* (default = least squares)

Method to be used to produce the incident Q - and U -images, i.e. the images that are corrected for the instrumental polarization effects. If `least squares`, the images are obtained by solving for every pixel the system of equations describing the measurements using linear least squares (see Eq. 35 of [van Holstein et al. 2020](#)). If `median`, the images are obtained by solving the system of equations for each pair of double-difference Q - and U -images (each HWP cycle) separately, and then computing the median over all resulting images. If a `float` in the range $0 < \text{value} < 0.5$ is supplied, the system of equations is solved for each pair of double-difference images (similar to `median`), after which the trimmed mean is computed over the resulting images. In that case the fraction to be cut from both tails of the distribution is equal to the supplied float. `least squares` yields the most accurate images with the highest signal-to-noise ratio and is therefore the recommended option. Using `median` may suppress image artifacts (e.g. the spider diffraction patterns), but the images have a lower signal-to-noise ratio and are less accurate. Using the trimmed mean (by supplying a float) may produce images with a signal-to-noise ratio close to that of the images produced with `least squares` while also suppressing image artefacts. When using `median` or a float, some Q - and U -images may be discarded in case the number of Q - and U -images are unequal (see the log file for details).

combination_method_intensity:

mean, median, *float* (default = mean)

Method to be used to produce the incident I_Q - and I_U -images. If `mean` or `median`, the images are obtained by computing the mean or median, respectively, over the I_Q - and I_U -images of all HWP cycles. If a `float` in the range $0 < \text{value} < 0.5$ is supplied, the images are obtained by computing the trimmed mean over the images. In that case the fraction to be cut from both tails of the distribution is equal to the supplied float. `mean` yields the most accurate images with the highest signal-to-noise ratio and is therefore the recommended option. Using `median` may suppress image artifacts (e.g. the spider diffraction patterns), but the images have a lower signal-to-noise ratio and are less accurate. Using the trimmed mean (by supplying a float) may produce images with a signal-to-noise ratio close to that of the images produced with `mean` while also suppressing image artefacts.

1.5 Citing IRDAP

If you use IRDAP for your publication, please cite [van Holstein et al. \(2020\)](#). For data in pupil-tracking mode please additionally cite [van Holstein et al. \(2017\)](#). Please use all upper case letters when mentioning IRDAP, and include the version number.

BibTeX citation [van Holstein et al. \(2020\)](#):

```
@ARTICLE{2020A&A...633A..64V,
  author = {{van Holstein}, R.~G. and {Girard}, J.~H. and {de Boer}, J. and
    {Snik}, F. and {Milli}, J. and {Stam}, D.~M. and {Ginski}, C. and
    {Mouillet}, D. and {Wahhaj}, Z. and {Schmid}, H.~M. and
    {Keller}, C.~U. and {Langlois}, M. and {Dohlen}, K. and {Vigan}, A. and
    {Pohl}, A. and {Carbillet}, M. and {Fantinel}, D. and {Maurel}, D. and
    {Orign{\`e}}, A. and {Petit}, C. and {Ramos}, J. and {Rigal}, F. and
    {Sevin}, A. and {Boccaletti}, A. and {Le Coroller}, H. and
    {Dominik}, C. and {Henning}, T. and {Lagadec}, E. and {M{\`e}nard}, F. and
    {Turatto}, M. and {Udry}, S. and {Chauvin}, G. and {Feldt}, M. and
```

(continues on next page)

(continued from previous page)

```

    {Beuzit}, J. -L.},
    title = "{Polarimetric imaging mode of VLT/SPHERE/IRDIS. II. Characterization
↪and correction of instrumental polarization effects}",
    journal = {\aap},
    keywords = {polarization, techniques: polarimetric, techniques: high angular
↪resolution, techniques: image processing, methods: observational, protoplanetary
↪disks, Astrophysics - Instrumentation and Methods for Astrophysics, Astrophysics -
↪Earth and Planetary Astrophysics},
    year = "2020",
    month = "Jan",
    volume = {633},
    eid = {A64},
    pages = {A64},
    doi = {10.1051/0004-6361/201834996},
    archivePrefix = {arXiv},
    eprint = {1909.13108},
    primaryClass = {astro-ph.IM},
    adsurl = {https://ui.adsabs.harvard.edu/abs/2020A&A...633A..64V},
    adsnote = {Provided by the SAO/NASA Astrophysics Data System}
}

```

BibTeX citation van Holstein et al. (2017):

```

@INPROCEEDINGS{2017SPIE10400E..15V,
    author = {{van Holstein}, Rob G. and {Snik}, Frans and {Girard}, Julien H. and
    {de Boer}, Jozua and {Ginski}, C. and {Keller}, Christoph U. and
    {Stam}, Daphne M. and {Beuzit}, Jean-Luc and {Mouillet}, David and
    {Kasper}, Markus and {Langlois}, Maud and {Zurlo}, Alice and
    {de Kok}, Remco J. and {Vigan}, Arthur},
    title = "{Combining angular differential imaging and accurate polarimetry
↪with SPHERE/IRDIS to characterize young giant exoplanets}",
    keywords = {Astrophysics - Earth and Planetary Astrophysics, Astrophysics -
↪Instrumentation and Methods for Astrophysics},
    booktitle = {Society of Photo-Optical Instrumentation Engineers (SPIE)
↪Conference Series},
    year = "2017",
    series = {Society of Photo-Optical Instrumentation Engineers (SPIE)
↪Conference Series},
    volume = {10400},
    month = "Sep",
    eid = {1040015},
    pages = {1040015},
    doi = {10.1117/12.2272554},
    archivePrefix = {arXiv},
    eprint = {1709.07519},
    primaryClass = {astro-ph.EP},
    adsurl = {https://ui.adsabs.harvard.edu/abs/2017SPIE10400E..15V},
    adsnote = {Provided by the SAO/NASA Astrophysics Data System}
}

```

1.6 Contributing & issues

Active help with the development of new functionalities and fixing bugs is very welcome. In case you would like to contribute, feel free to fork the [GitHub repository](#) and to create a pull request. Please have a look at the *to-do list* for the functionalities that still need to be added.

If you encounter a bug or problem, please [submit a new issue on the GitHub repository](#). Provide as much detail as possible (error message, operating system, Python version, etc.).

If you have feedback, questions or comments you can also send an e-mail to [Rob van Holstein](#).

1.7 Changelog & to-do

1.7.1 Changelog

v1.3.4 May 2022, R.G. van Holstein

- Corrected an error due to a scikit-image function being deprecated

v1.3.3 June 2021, R.G. van Holstein

- Updated code so that it correctly handles the change of the header names INS4 DROT2 BEGIN and INS4 DROT3 BEGIN to INS4 DROT2 START and INS4 DROT3 START in the FITS files

v1.3.2 January 2021, R.G. van Holstein

- Corrected a bug that prevented reductions to be performed when dark and flat files are included in the raw subdirectory

v1.3.1 September 2020, R.G. van Holstein

- Added a command-line option (`irdap -print`) to toggle printing of log statements in the terminal

v1.3.0 September 2020, R.G. van Holstein

- Implemented two new configuration parameters (`combination_method_polarization` and `combination_method_intensity`) for the computation of the final Q -, U -, I_Q - and I_U -images. The Q - and U -images can now be computed using least squares (as was previously the only option), the median and the trimmed mean. The I_Q - and I_U -images can be computed from the mean (as was previously the only option), the median and the trimmed mean
- Old configuration files that do not have these parameters can still be used: IRDAP automatically uses the default least squares and mean options if these new configuration parameters are missing from the configuration file
- Some minor additional improvements

v1.2.4 February 2020, R.G. van Holstein

- Put temporal uncertainty as uncertainty of the star polarization in the log file, added the accuracy of the Mueller matrix model and clarified which uncertainty to quote in papers
- Made IRDAP check for updates when run
- Implemented correct link to ADS of van Holstein et al. (2020) in program and documentation

v1.2.3 December 2019, R.G. van Holstein

- Clarified meaning of standard deviation of fitted center coordinates in log file
- Very minor correction to computation of HWP angle from headers
- Corrected minor bug when computing polarimetric efficiency when the number of Q and U cycles is unequal
- Added warning when spread of star polarization per HWP cycle is large and suggest to try normalized double difference

v1.2.2 November 2019, R.G. van Holstein

- Added polarimetric efficiency to crosstalk/transmission plot, a print statement with the range of the polarimetric efficiency of the observations, and a warning in case of low polarimetric efficiency
- Corrected a bug in the checks of `flux_annulus_star`
- Corrected a bug that when mean combining ADI+PCA-images only the first frame of the cube was written to the FITS-files

v1.2.1 November 2019, R.G. van Holstein

- Corrected bug introduced in v1.2.0 with computation of DoLP and AoLP of star when there is an unequal number of Q and U measurements

v1.2.0 November 2019, R.G. van Holstein

- Added computation of statistical uncertainty (photon and background noise) on measured star polarization in PDI part
- Added contrast curves for the detection of polarized extended sources and polarized point source with PDI

v1.1.0 October 2019, R.G. van Holstein & J. Milli

- Added angular differential imaging (ADI) including principal component analysis (PCA) of total intensity images for pupil-tracking observations
- Added mean combination of ADI final images to command-line option `irdap --meancombine`

v1.0.0 September 2019, R.G. van Holstein & J. Milli:

- No backward compatibility with previous versions
- Added analysis of master flux frames to enable the user to express the final images in `contrast/arcsec^2` or `Jansky/arcsec^2`
- Restructured code and configuration file to allow for the separate execution of pre-processing, polarimetric differential imaging (previously referred to as post-processing) and angular differential imaging
- Added plot of center coordinates of center frames vs time
- Added possibility to open documentation through terminal command `irdap --website`
- Confirmed that polarimetric data with dithering applied is correctly reduced

v0.3.0 July 2019, R.G. van Holstein:

- Added command-line option `irdap --meancombine` to mean-combine final images of multiple observation blocks

v0.2.1 June 2019, R.G. van Holstein:

- Added the possibility to reduce data taken with the narrowband filters using the existing model of the broadband filters

v0.2.0 June 2019, R.G. van Holstein:

- Changed the definition of the tuple of the input parameters `annulus_star`, `annulus_background` and `flux_annulus_background`: `width` is replaced by `outer_radius` and the definition of `start_angle` and `end_angle` is now the same as in DS9
- Replaced static flat in Ks by one with P0-90 inserted

v0.1.2 June 2019, R.G. van Holstein:

- Release of IRDAP package on PyPI and GitHub
- Documentation put online

1.7.2 To-do high priority

- Add possibility to reduce data with center frames having a different DIT than the object frames (i.e. automatically scale object frames to DIT and ND filter of center frames before subtracting it; test with data of HD142527)
- Automatically optimize the annulus for star polarization subtraction by iteratively updating the annulus by excluding disk signal and companions etc.
- Improve computation of stellar DoLP and AoLP and corresponding uncertainties (standard error of mean) by sampling from Gaussian distributions with Monte Carlo
- Correct error in computation of photon noise for Q and U images that uses signal in Q and U rather than I_Q and I_U.
- Make the data reduction work for data taken before approximately May 2015 that do not contain the header keyword indicating the Stokes parameter (using the HWP angle instead)
- Make IRDAP throw out less files when the order of the Stokes parameters of the files is not Qplus, Qminus, Uplus, Uminus
- Add possibility to use CENTER-frames with a different exposure time than the OBJECT-frames
- Add possibility to express contrast curves of extended sources in Jansky/arcsec²
- Make PDI part more memory efficient by re-using names of cubes
- Remove bad pixels in frames by setting them to NaN, and when frames are rotating (in PT and FT with derotator offsets) use nanmean to filter bad pixels out
- Make an overview of the warnings at the end of the log file
- Add classical ADI for polarimetric data reduction of pupil-tracking observations
- Add accurate (calibrated) model correction for narrowband filters
- Add option to apply accurate plate scale and distortion correction for data (especially important for pupil-tracking and bright sources; do we need calibrations?)
- For pupil-tracking data mask bad pixel clusters when computing least squares solution to get rid off sectors of bad data; for field-tracking data set bad pixel clusters to NaN.

1.7.3 To-do low priority

- Add computation of mean stellar polarization and uncertainty when mean-combining data sets
- Add images and plots aimed at analysis of polarimetric data (R² scaling, AoLP arrows in polarized intensity or DoLP images), maybe using terminal commands.
- Add option to subtract DARK,BACKGROUND-frames if sky files are lacking (especially important for Ks; test effect first before completely implementing)
- Add option to scale the master sky frame to subtract from the object frames (especially for Ks; see also Gallicher et al. 2011)
- Add option for ‘stupid ADI’ for field-tracking data with derotator offset and option to subtract 180 deg rotated image if no derotator offset
- Create contrast curves of final ADI-images
- Make figures with sub-images horizontal, or make multiple lines of left and right images in a single figure
- Add options for various methods to shift and rotate images (interpolate, ndimage-fourier, sci-image functions; similar to VIP)

- Improve centering of non-coronagraphic data (center found depends a lot on first PSF and affects Qphi and Uphi images). Perhaps fit coordinates on each PSF, but do the actual shifts with the mean of these fitted values. This has proven to give a more accurate final result.
- Add centering method that minimizes the SSR of the left minus right frames in an aperture centered on the star
- Test finding of satellite spots of center files when waffle pattern is '+'
- Add weighted least-squares as option for model correction (depending on image quality or polarimetric efficiency)
- Exclude saturated pixels in aperture to determine star polarization (same way as used in function `fit_2d_gaussian`)
- Determine star polarization as a function of aperture radius
- Add optional RDI for total intensity images

1.8 Mailing list

The mailing list is used to announce new releases and functionalities of IRDAF. To join the mailing list, send an empty e-mail with subject *IRDAF mailing list* to [Rob van Holstein](#).

1.9 Acknowledgements

Special thanks go out to the following people for their contributions to the development and publishing of the IRDAF Python package:

- Julien Milli (ESO Santiago, now IPAG)
- [Christian Ginski](#) (University of Amsterdam)
- Frédéric Vogt (ESO Santiago, now MeteoSwiss)
- Jozua de Boer (Leiden Observatory)

Copyright notice:

The IRDAF Python module is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, version 3 of the License.

The IRDAF Python module is distributed in the hope that it will be useful, but without any warranty; without even the implied warranty of merchantability or fitness for a particular purpose. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with the IRDAF Python module. If not, see <http://www.gnu.org/licenses/>.