Data processing of VISIR (Upgrade) data: FAQs

Paola Popesso - 2022-04-25 - Comments (0) - data processing FAQ for each instrument

Data processing of VISIR (Upgrade) data: Frequently asked questions

- **How can I run the pipeline when there are no bright objects in the individual image?**

Answer: The default use of the visir_img_combine recipe of the VISIR pipeline is with the "ref" option set to false. In this mode, the VISIR pipeline does a shift-and-add of the images purely based on the offset information included in the fits file headers, i.e. it does not try to detect the image positions, or expects an input file with explicit anchor points. So for a faint source this is typically how the pipeline is run. Only when the "ref" flag is set to "true" the visir_img_combine recipe will try to cross-correlate the images based on their contents (optionally with user-defined anchor points), which seems to be the situation you are describing in your e-mail. So simply running the VISIR pipeline with the "ref" option set to false should produce the desired results.

You are indeed correct in noticing that in both cases (with the "refine" flag set to false and true), the "visir_img_combine" recipe is trying to detect the sources in your image. However, this step is taken after the basic combination of the nodded images is created in order to create the image with the combination of the beams within the combined image (i.e. the combination of the three (for parallel chop/nod) or four (for a perpendicular chop/nod direction) star images within the combined frame).

The "visir_img_combine_onebeam_0000.fits" file (which contains this combination of the beams), is only created if all beams are either detected automatically (when the "refine" flag is set to "false"), or specified by the user (when the "refine" flag is set to "true" and anchor points are given), which explains why, in your case, it was only created when the "refine" flag was set to "true". So for faint sources it is indeed usual that the "visir_img_combine_onebeam" frame is not created, and one has to derive the source flux (or an upper limit to it) from the "visir_img_combine" stacked image.

So in your particular case, my advice would be to run the "visir_img_combine" recipe with the "refine" flag set to false (i.e. the default setting). The VISIR pipeline will create the combination of the nodded images using the ESO SEQ CUMOFFSETX and ESO SEQ CUMOFFSETY offset information in the fits file headers. If the source is so faint that it is still not seen in the combined image (the visir_img_combine_xxxx.fits file), the combination of the beams will fail and no file with the combination of the beams (visir_img_combine_onebeam_xxxx.fits) will be created. The upper limit to the flux of your source can be derived from the visir_img_combine_xxxx.fits frame.

- **I am trying to install the latest version of the VISIR Pipeline under Mac OSX, but the installation fails with the following error:**
- **make[2]: *** [org_eso_cpl_jni_CPLControl.h] Error 2**
- **make[1]: *** [all-recursive] Error 1**
- **make: *** [all] Error 2**
- **The command make failed with code 512 at ./install_pipeline line 800, <STDIN> line 2. This should not have happened.Please send an error report with the file ./install.log to usd-help@eso.org using subject: Installation of pipeline visir-kit-3.5.1 failed.**
- **What do I do?**

Answer: The problem you are encountering is because the Java Development Kit is installed on this platform in ways that are different from the one on Linux. Thus the installation script fails to find all the necessary executables and ancillary files. However, there is an easy workaround which consists in setting an environment variable to point directly to the root of your Java installation. Here is how to proceed:

- Download and unpack the pipeline kit of your choice (in your case, the VISIR one)
- Set up the JAVA_HOME environment variable. According to our investigations, for your MacOS X 10.7 machine you should do this via the command:

export JAVA_HOME=/System/Library/Frameworks/JavaVM.framework/

Please make sure this is indeed the correct path. The script will expect to find two subdirectories in JAVA_HOME that are named 'Headers' and 'Commands', where the files 'jni.h' and 'javah' should be located, respectively.

- Run the installation script 'install_pipeline'. The compilation should now go through after you have answered the initial questions about where you want the files installed.
- Please make sure to set up the additional environment variables CPLDIR, DYLD_LIBRARY_PATH and PATH according to the instructions displayed at the end of the installation process. You should be able to copy/paste the suggested commands to achieve this. Issue also a 'hash -r' command after you have set the PATH variable (this step may not be necessary).

Just run 'gasgano' from the command line.

- **How do I interpret the SEQ TIME keyword values?**

Answer: For VISIR, the SEQ TIME keyword is set to the requested exposure time. The actually achieved exposure time will, in most cases, deviate from this value, as only discrete values are allowed for some observational parameters, such as DIT, and others, such as T_chop (time spent in one chopping position), and T_nod (time spent in one nodding position) are pre-defined. The latter is 90 s. Therefore, following page 16 of the VISIR user manual, the number of nodding cycles is derived from the total integration time (user defined SEQ.TIME), but has to be an integer multiple of 2.

In your example, for NDIT to be 3, with NDITSKIP=1, f_chop must have been 0.025, giving T_chop=40 s and N_cycle_chop=2. Then, the total on-source integration time is indeed 240 s (see equation 3 of the VISIR User Manual).