MOLECFIT Experimental version

Paola Popesso - 2026-11-09 - Comments (0) - Data processing and analysis software resources

MOLECFIT Experimental version

The MOLECFIT Experimental versions implement improvements and changes relative to the current Public Release versions.

However as an experimental version, it may also include bugs, so **user beware, use at your own risk**...

This experimental version also includes experimental versions of the **FORS** and **XSHOOTER** pipelines & workflows that integrate telluric correction into the workflows and a version of the **GIRAFFE** pipeline & workflow that fixes a number of issues that show up for the latest versions of python modules such as matplotlib as experienced by up-to-date MacPorts/Homebrew systems.

This experimental version is generally used to test improved user friendliness and support for additional instruments in the Molecfit workflow, and telluric correction for other instruments. New features will generally be incorporated into the official versions at the next ESO pipeline annual public release update (usually in May of each year).

The latest release of this experimental version is release-82@2025-12-30T16:48:23+0000, including the following packages, that may differ from the currently available Public Release versions (i.e the versions here may be pre-release versions not yet fully verified and include modifications specific to this release channel, **use at your own risk**!):

fors 5.8.6.82
giraf 2.19.4.82
xshoo 3.8.9.82
molecfit 4.4.4.82

Subscribe to this article to receive notifications of updates.

Please follow the instructions below to install it.

First, make sure you have the software pre-requisites required for "pipeline source installations" and "KMOS and Molecfit pipeline source installations", see here.

In addition to those software pre-requisites, for linux the additional package "libtool" is also required.

Once all pre-requisites are installed, install as follows:

```
bash
INSTALL_DIR=${HOME}/pipelines/exp_reflex/$(date +%Y-%m-%d)
 mkdir -pv ${INSTALL_DIR}
 cd ${INSTALL_DIR}
 curl -O https://ftp.eso.org/pub/usg/molecfit/install_experimental_esoreflex
 env -i PATH=$(getconf PATH) bash ./install_experimental_esoreflex
```

On macOS on **Apple Silicon** hardware, if you have used HomeBrew to satisfy the Software Prerequisites, then replace the last line with:

```
env -i LDFLAGS="-headerpad_max_install_names" PATH=/opt/homebrew/bin:$(getconf PATH)
bash ./install_experimental_esoreflex
```

On macOS on **Intel** hardware, if you have used HomeBrew to satisfy the Software Prerequisites, then replace the last line with:

```
env -i LDFLAGS="-headerpad_max_install_names" PATH=/usr/local/bin:$(getconf PATH) bash
./install_experimental_esoreflex
```

On macOS, if you have used MacPorts to satisfy the Software Prerequisites, then replace the last line with:

```
env -i LDFLAGS="-headerpad_max_install_names" PATH=/opt/local/bin:$(getconf PATH) bash
./install_experimental_esoreflex
```

Alterantively see EPL that supports the Experimental Version as the default variant.

To run the experimental workflows

Use the following commands:

```
${INSTALL_DIR}/install/bin/exp_esoreflex molecfit
${INSTALL_DIR}/install/bin/exp_esoreflex fors_spec_molecfit
${INSTALL_DIR}/install/bin/exp_esoreflex xshooter_molecfit
```

You should try running the workflow on the tutorial data for one or more instruments. Simply start up the workflow and click the "play" button to run the tutorial for XSHOOTER. To try other instruments, just change the instrument name in the main canvas.

Please create a ticket if you have any trouble with the installation or running the tutorial. **Please understand though, that as this is a "use at your own risk" experimental version, your ticket may not be handled as quickly as it would be for the Public Release version.**

Experimental support for additional instruments

Experimental support can be added to this experimental version as follows:

CRIRES

To add experimental support for CRIRES, do the following:

```
curl -O https://ftp.eso.org/pub/usg/molecfit/user_instruments/CRIRES.tgz
tar -C ${HOME} -zxvf $(pwd)/CRIRES.tgz
```

Then set the INSTRUMENT variable in the molecfit main canvas to CRIRES to use.

The CRIRES experimental support implements support for the following CRIRES PRO.CATG products produced by the CRIRES pipeline:

- OBS_NODDING_EXTRACT_COMB
- OBS_NODDING_EXTRACTC_IDP
- OBS_NODDING_EXTRACTA
- OBS_NODDING_EXTRACTA_IDP
- OBS_NODDING_EXTRACTB
- OBS_NODDING_EXTRACTB_IDP
- OBS_NODDING_THROUGHPUT
- OBS_NODDING_THROUGHPUT_IDP
- OBS_STARING_EXTRACT
- OBS_STARING_EXTRACT_IDP
- OBS_STARING_IDP

Prior to 2025-04-07, the CRIRES experimental support created a MolecFit Compatiblie [MFC] file from the input product files produced by the pipeline with a single FITS-Binary table in one HDU extension. As of 2025-04-07, the CRIRES experimental support creates an MFC file from the input product files produced by the pipeline with one FITS-Binary table **per CRIRES phiysical chip**, i.e. three HDU extensions. The main impact of this concerns wavelength correction made by molecfit_model. molecfit_model computes one wavelength correction per "chip" which it assumes come in different extensions. So previously a single wavelength correction was computed for the entire data range. Now a wavelength correction will be computed for each CRIRES physical chip. This seems to produce better results, **but has not been evaluated thoroughly**.

The CRIRES experimental support now supports four MFC file structures, corresponding to four different Order & Chip Grouping Methods [OCGM]:

1. A single HDU : OCGM='singleHDU'
2. One HDU per order (3 chips per order) : OCGM='one_HDU_per_Order'
3. One HDU per chip and order : OCGM='one_HDU_per_Chip_and_Order'
4. One HDU per chip : OCGM='one_HDU_per_Chip'

So the current default OCGM is OCGM='one_HDU_per_Chip'. The CRIRES experimental support can be forced

to use different OCGM by setting the FITS keyword OCGM to the values above in the primary header of the input FITS file you wish to process with molecfit.

ERIS

To add experimental support for CRIRES, do the following:

```
curl -O https://ftp.eso.org/pub/usg/molecfit/user_instruments/ERIS.tgz
tar -C ${HOME} -zxvf $(pwd)/ERIS.tgz
```

Then set the INSTRUMENT variable in the molecfit main canvas to ERIS to use.

The ERIS experimental support implements support for the following ERIS PRO.CATG products produced by the ERIS pipeline:

- OBJECT_CUBE_COADD

HARPS and HARPN

To add experimental support for HARPS, do the following:

```
curl -O https://ftp.eso.org/pub/usg/molecfit/user_instruments/HARPS.tgz
tar -C ${HOME} -zxvf $(pwd)/HARPS.tgz
```

Then set the INSTRUMENT variable in the molecfit main canvas to HARPS to use.

To add experimental support for HARPN, do the following:

```
curl -O https://ftp.eso.org/pub/usg/molecfit/user_instruments/HARPN.tgz
tar -C ${HOME} -zxvf $(pwd)/HARPN.tgz
```

Then set the INSTRUMENT variable in the molecfit main canvas to HARPN to use.

The HARPS & HARPN experimental support implements support for the following HARPS & HARPN PRODCATG (note, *not* PRO.CATG) products produced by the HARPS & HARPN pipeline -- note this is the old version of the HARPS pipeline, before it was supported by the ESPRESSO pipeline:

- SCIENCE.SPECTRUM

SINFONI

To add experimental support for SINFONI, do the following:

```
curl -O https://ftp.eso.org/pub/usg/molecfit/user_instruments/SINFONI.tgz
tar -C ${HOME} -zxvf $(pwd)/SINFONI.tgz
```

Then set the INSTRUMENT variable in the molecfit main canvas to SINFONI to use.

The SINFONI experimental support implements support for the following SINFONI PRO.CATG products produced by the SINFONI pipeline:

- COADD_OBJ
- COADD_STD

- Tags
- ESO software
- molecfit
- pipeline
- software installation