What can I do with my NA added value data products?

Tristan Ashton - 2023-07-07 - Archive & Data Retrieval

The added-value delivery structure is designed to be fully compatible with use of ALMA Pipeline tasks and facilitate comparison with your delivered ALMA calibration + imaging pipeline weblog during analysis of the data and any reprocessing you choose to do. A number of workflows are supported in the new delivery format:

- You can proceed with your scientific analysis starting with the uid*targets.ms files and supply them to CASA tasks such as tclean, uvcontsub, or gaincal as a list (vis=['MS1.ms', 'MS2.ms', etc]). Examining the casa commands for each stage of the delivered ALMA calibration + imaging weblog will give examples of this (e.g. you can get the tclean command for any image that was made by clicking within the relevant hif_makeimages() stage).

- You can use scriptForReprocessing.py to restore the continuum subtracted data, re-image the data in the ALMA pipeline using new imaging parameters, or view the weblog (see below for usage). Here you can also easily modify cont.dat and rerun the continuum subtraction and/or imaging with a different continuum selection.

- You can generate the old style calibrated_final.ms either using scriptForReprocessing.py, or by hand via concat(). If you use scriptForReprocessing.py, there is also an option to generate an analogous calibrated_final_line.ms.

### scriptForReprocessing.py OVERVIEW ###

tscriptForReprocessing.py is intended to be a convenient wrapper for many of the ALMA pipeline functions that users may wish to use on their NA delivered value-added products. See the ALMA Pipeline Users Guide and Reference Manual for a full description of the ALMA pipeline: https://almascience.nrao.edu/processing/science-pipeline

The script can be launched via CASA with any version of CASA that includes the ALMA pipeline. See the above link for a mapping
of ALMA Cycle, CASA version, and Pipeline version. Thus it should be launched as:

```
$ casa --pipeline -c scriptForReprocessing.py [options]
```

optional arguments:
- `-h, --help` show this help message and exit
- `--contsub` Fit and subtract continuum using the channel ranges from the local cont.dat file. Generates new *uvcont.tbl tables in working_reprocess/
- `--contsub_fast` Continuum subtract data via uvcontsub and the local *uvcont.tbl files, but only using the CASA commands rather than pipeline calls (faster).
- `--image [IMAGE]` Run the imaging pipeline and place images in the specified directory (default='images'). NOTE: unless cont.dat or the imaging options in this script are modified, the images produced will be identical to those on the ALMA Science Archive
- `--cleanup` Remove working_reprocess/ directory and log files after any other options are executed. WARNING: removes weblogs inside of working_reprocess/
- `--weblog [WEBLOG]` Launches a browser to view weblog after other tasks are run. By default ('latest'), displays the latest weblog generated locally. Other options are to use the specific pipeline folder name (e.g. 'pipeline-20221010T192458')
- `--calibrated_final` Concatenate uid*targets.ms to produce calibrated_final.ms in measurement_sets/
- `--calibrated_final_line` Concatenate uid*targets_line.ms (if they exist) to produce calibrated_final_line.ms in measurement_sets/

Example usage:
Continuum subtract to get uid*targets_line.ms and cleanup:
   casa --pipeline -c scriptForReprocessing.py --contsub_fast --cleanup

Continuum subtract with a modified cont.dat, reimage, and view the resultant weblog:
   Modify cont.dat in caltables/
   casa --pipeline -c scriptForReprocessing.py --contsub --image
modified_images_folder --weblog

Make new mfs and agg cont images with different robust value and view the resultant weblog:
   Modify robust parameter in scriptForReprocessing.py
   casa --pipeline -c scriptForReprocessing.py --image new_robust_images_folder --weblog

Continuum subtract and generate all pipeline images with no mitigation
   Modify mitigate parameter in scriptForReprocessing.py -> False
   casa --pipeline -c scriptForReprocessing.py --contsub --image no_mitigation_folder

Make the old calibrated_final.ms and clean up:
   casa --pipeline -c scriptForReprocessing.py --calibrated_final --cleanup

If you open the script in a text editor, you will notice a block of user editable options at the top of the script (shown below). These options modify the imaging pipeline in various ways. Some of the most useful options may be to change mitigation parameters to image all of your science targets and spectral windows, to reimage a portion of your data with a different robust value, or to reimage with a uvtaper applied.

```
make_mfs_images = True                               # generate mfs (per spw) images
make_cont_images = True                             # generate aggregate continuum images
make_cube_images = True                            # generate cube images
make_repBW_images = True                       # generate images corresponding to the requested representative bandwidth
mitigate = True                                                    # run hif_checkproductsize() and mitigate created products if necessary.
# Set to false if you want all spws and all targets imaged at full resolution.
# WARNING: turning off mitigation may result in very large disk usage. Consider
```
# adjusting other mitigation parameter first, or manually selecting the target/spw combinations you want.

# for all values below, see Pipeline Users Guide and Reference Manual for detailed descriptions:
# https://almascience.nrao.edu/processing/science-pipeline
#
# any parameters not selected here can be edited manually in relevant section of the script below

maxproductsize = 350. # for mitigation; in GB
maxcubesize = 40. # for mitigation; in GB
maxcubelimit = 60. # for mitigation; in GB

field = None # String specifying fields to be imaged; default is all (pending mitigation)

phasecenter = None # Direction measure or field id of the image center. The default phase center is set to the mean of the field directions of all fields that are to be imaged together.

spw = None # Spw(s) to image; default is all spws

uvrange = None # Select a set of uv ranges to image; default is all

hm_imsize = None # Image x and y size in pixels or PB level; default is automatically determined

hm_cell = None # Image cell size; default is automatically determined

nbins = None # Channel binning factor for each spw; default is none

robust = None # Robust value to image with; default is
automatically determined

#          Example: 0.5
uvtaper = None                      # Uvtaper to apply to data; default is none
#          Example: ['1arcsec']

Tags
delivery
qa2