Doc 5.12, ver. 2.0 | May 2018

ALMA QA2 Data Products for Cycle 5

www.almascience.org

ALMA is a partnership of ESO (representing its member states), NSF (USA), and NINS (Japan), together with
NRC (Canada), NSC and ASIAA (Taiwan), and KASI (Republic of Korea), in cooperation with the Republic of Chile.
The Joint ALMA Observatory is operated by ESO, AUI/NRAO and NAOJ.

User Support:

For further information or to comment on this document, please contact your regional Helpdesk through the ALMA User Portal at **www.almascience.org**. Helpdesk tickets will be directed to the appropriate ALMA Regional Center at ESO, NAOJ or NRAO.

Revision History:

Version	Date	Editors
0.1-0.4	January 2014	Dirk Petry
0.5, 1.0	February 2014	Dirk Petry
2.0, 2.1	September 2014	Dirk Petry
2.2	February 2015	Dirk Petry
3.0	November 2015	Dirk Petry
5.0	March 2018	Dirk Petry, Arielle Moullet, Mark Lacy
5.1	May 2018	Dirk Petry, Mark Lacy

Contributors

Harold Francke (JAO), Misato Fukagawa (NAOJ), Mark Lacy (NRAO), John Hibbard (NRAO), Arielle Moulet (NRAO), Hiroshi Nagai (NAOJ), Theodoros Nakos (JAO), Kouichiro Nakanishi (NAOJ), Dirk Petry (ESO), Tony Remijan (NRAO), Thomas Stanke (ESO), Catarina Ubach (NRAO), and Martin Zwaan (ESO)

In publications, please refer to this document as:

Petry, D. et al., 2018, ALMA QA2 Data Products for Cycle 5, Version 2.0, ALMA

Table of contents

1 Introduction

The goal of ALMA Quality Assurance (QA) is to ensure that a reliable final data product is delivered to the PI that has reached the desired control parameters outlined in the science goals, is calibrated to the desired accuracy and contains no significant calibration or imaging artifacts. For Cycle 5, as for all previous Cycles, QA will be done on a "best effort" basis, covering all the major issues affecting data quality.

The first level of QA (QA0) is determined shortly after observation based on metadata and quicklook-analysis. The second level of Quality Assurance (QA2) deals with QA at the level of data reduction using the Science Pipeline or performed semi-interactively by the ALMA Data Reduction team. This document describes the final science data products that are delivered to PIs (and the archive) at the end of the QA2 process.

The ALMA QA2 data reduction team performs for each ALMA science data set a detailed analysis to confirm that the observations have achieved the science goals requested by the PI. In particular, frequency setup, spatial setup, angular resolution, and continuum and/or line detection sensitivity are verified.

If the requirements are met within the cycle-dependent tolerances (see chapter 11 of the ALMA Technical Handbook), the data are declared "QA2 pass", packaged in a standardized way, and delivered to the PI.

If on the other hand the requirements are not met but additional observations of the SB are no longer possible within the cycle, the data are declared "QA2 semipass" (if they contain any potentially useful information) and also delivered soon after it has become clear that re-observation is not possible ("timed-out"). Data may also be delivered as QA2-semipass if it is determined that the data quality cannot be improved by further observations, or if the only data taken were QA0_semipass, in which case only the raw data are delivered.

For both cases, the delivery is made in two parts:

(a) the science products and supporting material

(b) (optionally) the raw data.

Part (b), the raw data, consist of visibility datasets in ALMA's native format, the ASDM, which are only needed if the PI would like to perform custom data calibration and imaging. Part (a) of the delivery on the other hand, the "QA2 Products", consists of the QA2 data reduction scripts, log files, calibration, and flagging tables, and the imaging products on which the quality assurance decision was based. This document describes the contents of the Products section of the delivery in detail and is aimed at ALMA users.

Up to September 2014, the QA2 work was entirely performed by analysts using the QA2 script-generator tool. We refer to data processed in this way as "**analyst-calibrated**" or "**manually calibrated"**. In September 2014, the first official version of the ALMA Science Pipeline became available and has since then been upgraded several times. It is now used to calibrate most of the data taken in standard observing modes. We refer to data processed by the Science Pipeline as “**pipeline-calibrated**” data. The user can tell whether the data are pipeline-calibrated by the presence of a “*PPR*.xml” or a “*pprequest.xml” file in the “script” subdirectory of the delivery package. See section 4 for more details.

At the beginning of Cycle 4, the Science Pipeline was enabled to also perform imaging. This increased the processing capacity of the QA2 team significantly. We refer to data processed by the imaging section of the Science Pipeline as **“pipeline-imaged”.** If it was processed by an analyst using CASA directly without the help of the pipeline, we call it **“manually imaged”**. In Cycle 5, ca. 60% of all science data is pipeline-imaged. The approval of the imaging products remains in the hands of the analysts and Data Reduction Managers. Note that manual imaging is sometimes performed on pipeline-calibrated datasets, occasionally in addition to pipeline-imaging. See https://almascience.org/processing/science-pipeline and chapter 13 of the Cycle 5 ALMA Technical Handbook for more details on the pipeline.

2 Group ObsUnitSet (GroupOUS), Member ObsUnitSet (MemberOUS), and Scheduling Block (SB)

By design, the smallest scheduling entity which is processed and delivered individually is the Member ObsUnitSet (MemberOUS or MOUS). QA2 therefore is carried out on individual MemberOUSs. As in previous Cycles, the MemberOUS contains also in Cycle 5 essentially always only one Scheduling Block (SB), even in the case of Total Power Array observations.

Hence also in Cycle 5, QA2 continues to be carried out together on all the Execution Blocks (EBs) of a single SB corresponding to one MemberOUS. The products for each MemberOUS are delivered separately, and the delivery of the products for each MemberOUS is labeled with the MemberOUS UID, the unique ALMA archive key.

The MemberOUSs are organized into Group ObsUnitSets (in short: GroupOUSs). In the trivial case there is only one GroupOUS for each Science Goal. In Cycle 5, the following types of MemberOUSs can in principle be in one GroupOUS:

a) The 12 m Array MemberOUS(s) (up to two)

b) The 7 m Array MemberOUS

c) The Total Power observation MemberOUS

d) The Total Power calibration MemberOUS (may be omitted)

The delivery packaging results in one or a number of tar files which are named as follows:

```
<Project ID>_<MemberOUS_UID>_m_of_n.tar
```

for each member where n is the total number of tar files into which the delivery is split and m is the number of the individual tar file (typically both m and n are 1). It is planned that the archive will soon offer the possibility of downloading individual product files.

3 Overview

An ALMA data delivery in Cycle 5 consists of the seven items shown in the directory tree below. Items 1 to 6 are delivered in one package (part (a)) which should be downloaded first. Item 7 (part (b)) is available for download separately and is only required if the users would like to work with the visibility data. Once unpacked, all data fall into a standardized directory structure:

```
|-- project_id/
|  |-- sg_ouss_id/
|  |  |-- group_ouss_id/
|  |  |  |-- member_ouss_id/
|  |  |  |  |-- README          (1)
|  |  |  |  |-- product/        (2)
|  |  |  |  |-- calibration/    (3)
|  |  |  |  |-- qa/             (4)
|  |  |  |  |-- script/         (5)
|  |  |  |  |-- log/            (6)
|  |  |  |  |-- raw/            (7)     (only present when part (b) was unpacked)
```

The directory contents is summarized in the following. When an item is marked as “best effort” this means that it is generated by the analysis team when sufficent resources are available and it is compatible with the science goal.

1) README

A text file explaining where to find information about the QA2 results. The main source of information on the data processing is now the “weblog”, item (4), in the subdirectory “qa”.

2) Imaging Products (in FITS format) in subdirectory “product”

a) for pipeline-imaged data:

— A set of continuum and line images for all science spectral windows covering an as large as possible part of the spectral range. This set is as complete as possible given the computing resources of the processing pipeline, i.e. for datasets with extremely high spatial and/or spectral resolution, some parts of the set may have been omitted in order to keep the size of the product package within reasonable limits. See the pipeline documentation at **https://almascience.org/processing/science-pipeline** .

b) for pipeline-calibrated data generally (pipeline- or manually imaged):

— Continuum images of the bandpass and phase calibrators and (if present) the check source(s) for each of the science spectral windows

c) Manually imaged line cubes:

— For the representative spectral window: One spectral cube of "representative channels" for at least one target in the MemberOUS, made at the requested spectral resolution considering the bandwidth specified for sensitivity. If the achieved spatial resolution was higher than requested, tapering may have been applied if needed to reach the science goal. Coarse continuum subtraction is applied for sources with bright continuum.

— (best effort) For each source and each spectral window: cubes of all species specifically listed in the proposal, made at an angular resolution as close as possible to the requested one and including line-free channels on either side. Continuum subtraction is applied for sources with bright continuum. A note will be added if additional species are found in the data, but imaging may not be carried out.

— (best effort) Continuum image of all non-edge, non-line channels

d) Manually imaged continuum and or aggregate bandwidth images:

— One image of all continuum spectral windows, all non-edge channels

— (best effort) One image of all continuum spectral windows, all non-line channels

<u>3) CASA Tables (in CASA Table format) in subdirectory "calibration"</u>

a) for pipeline-calibrated data

— Calibration tables (Antenna Position, Tsys, WVR, Bandpass, Gain, Amplitude)

— Flagversions tables

— Calibrator fluxes from the ALMA calibration database (flux.csv)

— Antenna positions (antennapos.csv)

— List of the applycal commands for the delivered calibration table (*calapply.txt)

— Flagging commands added manually by the pipeline operator (*flagtemplate.txt and/or *flagtargetstemplate)

— auxcaltables.tgz, auxcalapply.txt

— hifa_calimage.auxproducts.tgz

— cont.dat

See **<u>https://almascience.org/processing/science-pipeline</u>** for more information on the role of these files.

b) for analyst-calibrated data

— Calibration tables (Tsys, WVR, Bandpass, Gain, Amplitude)

— Flagversions tables

— Calibration plots and wvrgcal diagnostic output

4) QA documentation in subdirectory “qa”

a) for pipeline-calibrated data

— The Pipeline Weblog – a system of webpages containing all the diagnostic plots and other information generated by the pipeline. This is the main source of information on all properties of the data and the steps taken to calibrate and flag them.

b) for analyst-calibrated data

— QA Reports for all the imaged data (png images and txt files).

5) Data Reduction Scripts (ASCII files) in subdirectory “script”

a) for pipeline-calibrated data

— the Python scripts needed to restore the calibrated MS(s) or rerun the entire pipeline

— the pipeline processing request file (PPR*.xml or *pprequest.xml)

— the imaging preparation script (if necessary) and the imaging script (may be a dummy)

b) for analyst-calibrated data

— CASA reduction scripts including: calibration scripts, imaging preparation script (if necessary), imaging script, and script to set up and run the calibration.

6) calibration and imaging log files in subdirectory “log”

— CASA log files from the QA2 processing

7) Datasets (in ASDM format) in subdirectory “raw” (if downloaded)

— (for optional download) the ExecBlocks that were used in the imaging/QA2 assessment

— (for optional download if applicable) Additional datasets that were NOT used in the data reduction but may still contain valuable scientific information

4 Data Delivery Products

4.1 README

The complete name of this file in normal deliveries is <MOUS UID>.README.txt . Up to the end of Cycle 4, this was a text file summarizing the QA2 results and explaining the structure of the tarball and file naming conventions. Starting with Cycle 5, this is normally just a reference to where one can find additional information.

4.2 Data Reduction Scripts (directory “script”)

In the subdirectory “script”, the user finds the CASA data reduction scripts (Python) which were used to calibrate and image the data. Depending on whether the Science Pipeline was used or the data were manually calibrated, the contents of this directory varies:

1) uid....ms.scriptForCalibration.py: one calibration script for each Execution Block of analyst-calibrated interferometric data.

2) uid....ms.scriptForSDCalibration.py: one calibration script for each Execution Block of analyst-calibrated single-dish data. See section 5.6.

3) scriptForImagingPrep.py (for manually imaged datasets, if needed): This script contains all necessary steps to prepare imaging which need to be performed on all calibrated EBs together (and can therefore not be performed by the calibration for the individual EBs). These steps may include alignment of frequency grids before concatenation, the concatenation into one MeasurementSet (MS), adjustments to the flux calibration etc. The result is a MS or a set of MSs which are ready for the final imaging. If present, this script is run automatically as the last step by the scriptForPI.py (see below).

4) scriptForPolCalibration.py (for analyst-calibrated full-polarisation data): special calibration routine for full-polarisation data which is executed after the per-EB-calibration scripts. If present, it will be run automatically by scriptForPI.py (see below).

5) scriptForImaging.py: (for interferometric data) this script regenerates the imaging products (see next section) as they are stored in the “products” subdirectory of the delivery. Where convenient for the analyst, the scriptForImaging.py may also contain elements which could in principle be put into a scriptForImagingPrep.py. In that case, a scriptForImagingPrep is not created.

6) scriptForSDImaging.py: (for analyst-imaged single-dish data) this script regenerates the imaging products (see next section) as they are stored in the “products” subdirectory of the delivery. See also section 5.6.

7) **scriptForPI.py**: in all cases, whether pipeline- or manually calibrated, run this script in the “script” directory in order to regenerate the calibrated MeasurementSet (MS) of the delivered data. The README file contains instructions (or points to them) on how to do this. Running the scriptForPI.py requires that you have downloaded the ASDM datasets and unpacked them such that they reside in the “raw” directory (created during unpacking). See section 5 for more details.

   The only type of dataset which was up to the middle of Cycle 5 *not* delivered with a scriptForPI.py was the *analyst-calibrated Total Power* data. See section 5.6.

8) casa_piperestorescript.py: This script is used by scriptForPI.py to restore the calibrated MS(s) from the raw ASDMs by applying the necessary calibration tables. This is only provided in the case of pipeline calibration. If it is not provided or removed, the scriptForPI will run the casa_pipescript instead.

9) casa_pipescript.py: Performs the calibration from scratch using Pipeline tasks. See the Pipeline documentation (see section 6) for more details.

10) the pipeline processing request (PPR): an XML file which was used to drive the Pipeline calibration. See **https://almascience.org/processing/science-pipeline** .

Furthermore, there may be the following script file present in case of solar observations: sun_reduction_util.py (contains helper routines specific to solar data analysis),

4.3 Imaging Products (directory “product”)

The “product” subdirectory of the delivery contains the imaging products that were either generated by the imaging section of the Science Pipeline or by an analyst with the script “scriptForImaging.py” from the “script” subdirectory based on the calibrated dataset which is obtained when the “scriptForPI.py” is run. For pipeline-calibrated interferometric data it also contains calibrator images generated by the Pipeline.

A coherent naming scheme was introduced for all imaging products:
<MOUS>.<target>_<target type>.spw<SPWs>.<image type>.<stokes>.<comment>.<x>.<ext>

where <*MOUS*> is the MOUS UID, <*target*> is the name of the imaged field, <*target type*> is “sci” for science target, “ph” for phase calibrator, “bp” for bandpass calibrator, “amp” for flux calibrator, “chk” for check source, “pol_leak” for the polarisation calibrator, <*SPWs*> are the imaged spectral window Ids, <*image type*> is “mfs” for an aggregate bandwidth image, “cube” for a spectrally resolved image, “cont” for an image with the line channels excluded, <*stokes*> is the name of the image Stokes parameter (e.g. “I”), <comment> can be an explanatory text, <*x*> is “pbcor” for the primary beam corrected image, “pb” for the primary beam itself, “mask” for the mask image, and <*ext*> is the file name extension describing the file type, e.g. “fits”.

Note that the <*comment*> field typically contains the word “manual” if the image was obtained by manual imaging.

The goal of ALMA QA2 in Cycle 5 remains that the calibration is reliable and "science ready", but the imaging products may not be. A sufficient fraction of the possible imaging products is created to verify that the data meet the science goals set by the PI (resolution and sensitivity), within some tolerance, but the imaging may be incomplete. In general, investigators should expect to need to re-image their data using the provided scripts (for manual imaging) or the pipeline documentation (for pipeline-imaged data) as a guideline. In particular, care should be taken to optimize continuum subtraction (if relevant) and fine-tune the image deconvolution parameters (e.g. clean masks and thresholds).

Care has been taken that the images conform with the FITS standard 3.0.

Where necessary (i.e. where a clean mask was actually applied), the imaging masks are provided as CASA images. These represent the final mask used in the last cleaning iteration. The masks used in previous iterations may have been different (tighter).

The science images included in deliveries of interferometric data are corrected for the primary beam (PB), i.e. the dependence of the instruments sensitivity on direction within the FOV. For each image, two files are being delivered:

a) the PB-corrected image (file name ending in ".pbcor.fits")
b) the image of the PB used for the correction ("*.pb.fits" or “*.flux.fits”, possibly gzipped)

The image noise was measured in the uncorrected image. The corrected image (a) was then obtained by dividing the uncorrected image by the PB image (b). The uncorrected image can be recovered using the CASA task “impbcor” in mode "m":

```
impbcor(imagename='image.pbcor.fits',pbimage='image.flux.fits',
         mode='m', outfile='image.recovered')
```

For **Total Power (TP)** data, images without primary beam correction are delivered. Also images of calibrators are not included (as the beam calibrator is actually part of a separate observation).

4.4 CASA Tables (directory “calibration”)

In order to further document the process of data calibration, all calibration tables generated by the calibration process are stored in the “calibration” subdirectory of the delivery.

In the case of analyst-calibrated data, information on how each table was created can be found in the scriptForCalibration.py and the scriptForImagingPrep.py scripts. CASA provides functionality to view these tables. Please refer to the CASA documentation (see section 6) on how to do that. Many of the plots of interest in this context are, however, also already available in the QA documentation (see next section).

In the case of pipeline-calibrated data, the files in this directory are needed to restore the calibrated MS with “casa_piperestorescript.py” or, in absence of the latter, with “casa_pipescript.py” which in turn is called by “scriptForPI.py”. For details on how they were generated, please refer to **https://almascience.org/processing/science-pipeline** . All diagnostic plots are also in this case already available in the Pipeline Weblog which can be found in directory “qa” (see next section).

4.5 QA Documentation (directory “qa”)

In the subdirectory “qa” of the delivery, the user finds diagnostic information obtained during calibration and imaging.

In the case of *pipeline-calibrated data,* all diagnostic information is assembled in a system of html pages which is called the **Weblog**. To access the Weblog, run

```
tar xzf *weblog.tar.gz
```

in the “qa” directory and then use your favourite web browser (firefox recommended) to open the resulting file

```
pipeline*/html/index.html
```

For more information on the contents of the Weblog, please refer to the Pipeline documentation at **https://almascience.org/processing/science-pipeline** .

In the case of analyst-calibrated data, there are, for each EB and labeled with the EB UID, several png images and a textfile. The png files can be viewed with standard system tools such as “eog” (under Linux). They show a set of diagnostic plots describing the following aspects of the data:

1) Observing Schedule (observation intent vs. time)
2) Mosaic Pointing Configuration
3) Antenna Array Configuration
4) Effectiveness of the WVR correction for each antenna
5) Temporal gain calibration solutions for each antenna
6) Temporal phase calibration solutions for each antenna
7) Average bandpass solution for each spectral window
8) System Temperature vs. frequency for each antenna and each spectral window
9) Phase calibrator amplitude and phase vs. frequency for each spectral window and polarisation
10) Flux calibration model and data (visibility amplitude vs. UV distance)
11) Target visibility amplitude vs. UV distance for each spectral window and polarisation
12) Phase calibrator amplitude and phase vs. frequency for each spectral window and polarisation
13) Target field UV coverage
14) Test image of the target
15) Target imaging synthesized beam (PSF)

The contents of the uid*textfile.txt file is mostly self-explanatory. Of particular interest is the "Check of a target image and sensitivity" which mentions approximately the achieved spatial resolution and sensitivity. Note that the finally achieved resolution and sensitivity could be different (since it might have used additional procedures in imaging).

Based on the calibration tables in the "calibration" directory, the user can produce additional diagnostic plots in CASA using the tasks *plotms, plotcal, and plotbandpass*. See the CASA documentation for more details.

If additional diagnostic plots on the visibility data itself are required, the calibrated MeasurementSet (MS) first has to be generated from the raw data (see the next section). Once this is done, the CASA task *plotms* should be used to generate the plots.

5 Generating the calibrated visibilities

In ALMA Cycle 0 and early Cycle 1, the visibility data were delivered with all other products in ready-to-use MeasurementSet (MS) format. Since the middle of Cycle 1, the products are separated from the raw visibility data (as described in the introduction), and downloads of the latter are now optional. To minimize download time, the raw visibilities are delivered in the native ALMA format, the ASDM (ALMA Science Data Model).

If the user would like to only modify or extend the imaging of the dataset but accept the observatory calibration of the data, he/she needs to download the raw data and apply the calibration via the scripts provided with the delivery as described in the following. Otherwise, the user can only download the raw data and use the calibration scripts or pipeline documentation as a guidance and develop a custom calibration.

The download of the raw data depends on the details of where the data were staged for delivery. The user should refer to the information in the delivery email as to how exactly the raw data should be obtained. Unless there is a technical problem, this will work via the ALMA Archive Request Handler. See http://almascience.org/alma-data .

5.1 Running the scriptForPI.py in the case of analyst-calibrated data

For analyst-calibrated data, the scriptForPI will find the scripts named scriptForCalibration.py and scriptForImagingPrep.py (if present), set up every necessary item for their execution, and then run them to obtain the MS ready for imaging.

Once the data products tar file is downloaded and unpacked, the user finds a certain directory structure on disk (see the description in section 3). The following directory tree shows an example of analyst-calibrated (rather than pipeline-calibrated) data. However, also for pipeline-calibrated data, the procedure for running “scriptForPI.py” is the same and has essentially the same final product.

```
|-- 2017.1.12345.S/
|  |-- sg_ouss_id/
|  |  |-- group_ouss_id/
|  |  |  |-- member_ouss_id/
|  |  |  |  |-- member.uid___A001_X1284_X1a50.README.txt
|  |  |  |  |-- script/
|  |  |  |  |  |-- member.uid___A001_X1284_X1a50.scriptForPI.py
|  |  |  |  |  |-- member.uid___A001_X1284_X1a50.scriptForImaging.py
|  |  |  |  |  |-- member.uid___A001_X1284_X1a50.scriptForImagingPrep.py
|  |  |  |  |  |-- uid___A002_Xc96f17_X792b.ms.scriptForCalibration.py
|  |  |  |  |  |-- uid___A002_Xc97aeb_X1f4.ms.scriptForCalibration.py
|  |  |  |  |-- calibration/
|  |  |  |  |  |-- uid___A002_Xc96f17_X792b.calibration.tgz
|  |  |  |  |  |-- uid___A002_Xc96f17_X792b.calibration.plots.tgz
|  |  |  |  |  |-- uid___A002_Xc97aeb_X1f4.calibration.tgz
|  |  |  |  |  |-- uid___A002_Xc97aeb_X1f4.calibration.plots.tgz
|  |  |  |  |-- log/
|  |  |  |  |  |-- member.uid___A001_X1284_X1a50.analysis.log.tgz
|  |  |  |  |  |-- member.uid___A001_X1284_X1a50.calibrated.log.tgz
|  |  |  |  |  |-- member.uid___A001_X1284_X1a50.uid___A002_Xc96f17_X792b.log.tgz
|  |  |  |  |  |-- member.uid___A001_X1284_X1a50.uid___A002_Xc97aeb_X1f4.log.tgz
|  |  |  |  |-- product/
|  |  |  |  |  |-- J1315+031_sci.spw0.cube.I.manual_H13CN.image.pbcor.fits
|  |  |  |  |  |-- J1315+031_sci.spw0.cube.I.manual_H13CN.mask.tgz
|  |  |  |  |  |-- J1315+031_sci.spw0.cube.I.manual_H13CN.pb.fits.gz
|  |  |  |  |  |-- J1315+031_sci.spw0_1_2_3.mfs.I.manual.image.pbcor.fits
|  |  |  |  |  |-- J1315+031_sci.spw0_1_2_3.mfs.I.manual.mask.tgz
| | | | | |-- J1315+031_sci.spw0_1_2_3.mfs.I.manual.pb.fits.gz
| | | | | |-- J1315+031_sci.spw1.cube.I.manual_HCO.image.pbcor.fits
| | | | | |-- J1315+031_sci.spw1.cube.I.manual_HCO.mask.tgz
| | | | | |-- J1315+031_sci.spw1.cube.I.manual_HCO.pb.fits.gz
| | | | | |-- J1315+031_sci.spw2.cube.I.manual_H13CO.image.pbcor.fits
| | | | | |-- J1315+031_sci.spw2.cube.I.manual_H13CO.mask.tgz
| | | | | |-- J1315+031_sci.spw2.cube.I.manual_H13CO.pb.fits.gz
| | | | | |-- J1315+031_sci.spw3.cube.I.manual_cHCCCH.image.pbcor.fits
| | | | | |-- J1315+031_sci.spw3.cube.I.manual_cHCCCH.mask.tgz
| | | | | |-- J1315+031_sci.spw3.cube.I.manual_cHCCCH.pb.fits.gz
| | | | |-- qa/
| | | | | |-- uid___A002_Xc96f17_X792b__qa2_part1.png
| | | | | |-- uid___A002_Xc96f17_X792b__qa2_part2.png
| | | | | |-- uid___A002_Xc96f17_X792b__qa2_part3.png
| | | | | |-- uid___A002_Xc96f17_X792b__qa2_part4.png
| | | | | |-- uid___A002_Xc96f17_X792b__textfile.txt
| | | | | |-- uid___A002_Xc97aeb_X1f4__qa2_part1.png
| | | | | |-- uid___A002_Xc97aeb_X1f4__qa2_part2.png
| | | | | |-- uid___A002_Xc97aeb_X1f4__qa2_part3.png
| | | | | |-- uid___A002_Xc97aeb_X1f4__qa2_part4.png
| | | | | |-- uid___A002_Xc97aeb_X1f4__textfile.txt
```

Note that for simplicity, the names of the files in the “product” and the “qa” directory were shortened. In a real delivery, they would all be preceded by the MOUS UID.

This dataset has two Execution Blocks (EBs) with UIDs “uid___A002_Xc96f17_X792b” and “uid___A002_Xc97aeb_X1f4”. In the directory “script” it has therefore two scripts for calibration:

```
| | | | | |-- uid___A002_Xc96f17_X792b.ms.scriptForCalibration.py
| | | | | |-- uid___A002_Xc97aeb_X1f4.ms.scriptForCalibration.py
```

and, since apparently additional commands were necessary to combine the results from the individual calibrations, a script to finalize the dataset for imaging is present:

```
| | | | | |-- member.uid___A001_X1284_X1a50.scriptForImagingPrep.py
```

Furthermore, like in every delivery, the directory “script” contains the scripts “*scriptForImaging.py” and “*scriptForPI.py”.

```
| | | | | |-- member.uid___A001_X1284_X1a50.scriptForPI.py
| | | | | |-- member.uid___A001_X1284_X1a50.scriptForImaging.py
```

The user is encouraged to inspect all these scripts as they also may contain helpful comments on special properties of the data.

Unpacking the raw data after download should be done at the top level (in the directory containing the directory 2017.1.12345.S in this example). It should result in an additional directory “raw” at the level of the directory “script”:

```
| | | | |-- raw/
| | | | | |-- uid___A002_Xc96f17_X792b.asdm.sdm
| | | | | |-- uid___A002_Xc97aeb_X1f4.asdm.sdm
```

In order to just reproduce the observatory calibration, the user can then run the “scriptForPI.py” by typing at the shell prompt of the operating system:

```
cd script
casa -c "execfile('member.uid_A001_X1284X1a50.scriptForPI.py')"
```

(Note that the same version of CASA should be used as was used for the original processing.)

This will first perform various tests on the directory structure and the presence of the necessary files and then run the corresponding scriptForCalibration on each of the ASDMs. If there is more than one ASDM and further steps were necessary in the QA2 process to prepare the dataset for imaging, the scriptForImagingPrep will be present and will be run as well.

The results will be placed in a new directory “calibrated” at the same level as the directories “raw” and “script”:

```
| | | | |-- calibrated
| | | | | |-- calibrated.ms
| | | | | |-- uid___A002_Xc96f17_X792b.calibration
| | | | | |-- uid___A002_Xc96f17_X792b.ms.split.cal
| | | | | |-- uid___A002_Xc97aeb_X1f4.calibration
| | | | | |-- uid___A002_Xc97aeb_X1f4.ms.split.cal
```

The final calibrated visibility data combined for all EBs can in this case be found in the MS “calibrated.ms”. If the analyst did not deem it necessary to perform a concatenation, the “calibrated.ms” can also be absent and the imaging will operate directly on the MSs for each EB. The intermediate output of CASA to achieve the calibration of each EB is stored in the calibration working directories:

```
| | | | | |-- uid___A002_Xc96f17_X792b.calibration
| | | | | |-- uid___A002_Xc97aeb_X1f4.calibration
```

while the calibrated visibilities for each EB individually are stored in the MSs with names ending in “.ms.split.cal” (if there is only one EB, this MS will take the role of calibrated.ms).

5.2 Running the scriptForPI.py in the case of pipeline-calibrated data

The following shows the contents of the directory tree for the data set discussed in the previous section, however, for the case that the data were *pipeline-calibrated*. This can easily be seen from the fact that there is a file “***pprequest.xml**” present in the “script” directory. Like in the case of analyst-calibrated data, you find the “scriptForPI.py” in the “script” directory.

```
|-- 2017.1.12345.S/
|   |-- sg_ouss_id/
|   |   |-- group_ouss_id/
|   |   |   |-- member_ouss_id/
|   |   |   |   |-- member.uid___A001_X1284_X1a50.README.txt
|   |   |   |   |-- script/
|   |   |   |   |   |-- member.uid___A001_X1284_X1a50.scriptForPI.py
|   |   |   |   |   |-- member.uid___A001_X1284_X1a50.scriptForImaging.py
|   |   |   |   |   |-- member.uid___A001_X1284_X1a50.scriptForImagingPrep.py
|   |   |   |   |   |-- member.uid___A001_X1284_X1a50.calimage.pipeline_manifest.xml
|   |   |   |   |   |-- member.uid___A001_X1284_X1a50.calimage.product_rename.txt
|   |   |   |   |   |-- member.uid___A001_X1284_X1a50.casa_piperestorescript.py
|   |   |   |   |   |-- member.uid___A001_X1284_X1a50.casa_pipescript.py
|   |   |   |   |   |-- member.uid___A001_X1284_X1a50.hifa_calimage.pprequest.xml
|   |   |   |   |-- calibration/
|   |   |   |   |   |--  member.uid___A001_X1284_X1a50.antennapos.csv
|   |   |   |   |   |--  member.uid___A001_X1284_X1a50.flux.csv
|   |   |   |   |   |--  member.uid___A001_X1284_X1a50.hifa_calimage.auxproducts.tgz
|   |   |   |   |   |--  member.uid___A001_X1284_X1a50.session_1.caltables.tgz
|   |   |   |   |   |--  uid___A002_Xc96f17_X792b.flagtargetstemplate.txt
|   |   |   |   |   |--  uid___A002_Xc96f17_X792b.flagtemplate.txt
|   |   |   |   |   |--  uid___A002_Xc96f17_X792b.ms.calapply.txt
|   |   |   |   |   |--  uid___A002_Xc96f17_X792b.flagversions.tgz
|   |   |   |   |   |--  uid___A002_Xc97aeb_X1f4.flagtargetstemplate.txt
|   |   |   |   |   |--  uid___A002_Xc97aeb_X1f4.flagtemplate.txt
|   |   |   |   |   |--  uid___A002_Xc97aeb_X1f4.ms.calapply.txt
|   |   |   |   |   |--  uid___A002_Xc97aeb_X1f4.flagversions.tgz
|   |   |   |   |-- log/
|   |   |   |   |   |-- member.uid___A001_X1284_X1a50.casa-20171117-010710.log
|   |   |   |   |   |-- member.uid___A001_X1284_X1a50.casa-20171117-034517.log
|   |   |   |   |   |-- member.uid___A001_X1284_X1a50.casa-20171117-035420.log
|   |   |   |   |   |-- member.uid___A001_X1284_X1a50.hifa_calimage.casa_commands.log
|   |   |   |   |-- product/
|   |   |   |   |   |-- J1315+031_sci.spw0.cube.I.manual_H13CN.pbcor.fits
|   |   |   |   |   |-- J1315+031_sci.spw0.cube.I.manual_H13CN.mask.tgz
|   |   |   |   |   |-- J1315+031_sci.spw0.cube.I.manual_H13CN.pb.fits.gz
|   |   |   |   |   |-- J1315+031_sci.spw0_1_2_3.mfs.I.manual.pbcor.fits
|   |   |   |   |   |-- J1315+031_sci.spw0_1_2_3.mfs.I.manual.mask.tgz
|   |   |   |   |   |-- J1315+031_sci.spw0_1_2_3.mfs.I.manual.pb.fits.gz
|   |   |   |   |   |-- J1315+031_sci.spw1.cube.I.manual_HCO.pbcor.fits
|   |   |   |   |   |-- J1315+031_sci.spw1.cube.I.manual_HCO.mask.tgz
|   |   |   |   |   |-- J1315+031_sci.spw1.cube.I.manual_HCO.pb.fits.gz
|   |   |   |   |   |-- J1315+031_sci.spw2.cube.I.manual_H13CO.pbcor.fits
|   |   |   |   |   |-- J1315+031_sci.spw2.cube.I.manual_H13CO.mask.tgz
| | | | | |-- J1315+031_sci.spw2.cube.I.manual_H13CO.pb.fits.gz
| | | | | |-- J1315+031_sci.spw3.cube.I.manual_cHCCCH.pbcor.fits
| | | | | |-- J1315+031_sci.spw3.cube.I.manual_cHCCCH.mask.tgz
| | | | | |-- J1315+031_sci.spw3.cube.I.manual_cHCCCH.pb.fits.gz
| | | | | |-- J1617-5848_bp.spw19.mfs.I.mask.fits
| | | | | |-- J1617-5848_bp.spw19.mfs.I.pb.fits.gz
| | | | | |-- J1617-5848_bp.spw19.mfs.I.pbcor.fits
| | | | | |-- J1617-5848_bp.spw21.mfs.I.mask.fits
| | | | | |-- J1617-5848_bp.spw21.mfs.I.pb.fits.gz
| | | | | |-- J1617-5848_bp.spw21.mfs.I.pbcor.fits
| | | | | |-- J1617-5848_bp.spw23.mfs.I.mask.fits
| | | | | |-- J1617-5848_bp.spw23.mfs.I.pb.fits.gz
| | | | | |-- J1617-5848_bp.spw23.mfs.I.pbcor.fits
| | | | | |-- J1617-5848_bp.spw25.mfs.I.mask.fits
| | | | | |-- J1617-5848_bp.spw25.mfs.I.pb.fits.gz
| | | | | |-- J1617-5848_bp.spw25.mfs.I.pbcor.fits
| | | | | |-- J1617-7717_ph.spw19.mfs.I.mask.fits
| | | | | |-- J1617-7717_ph.spw19.mfs.I.pb.fits.gz
| | | | | |-- J1617-7717_ph.spw19.mfs.I.pbcor.fits
| | | | | |-- J1617-7717_ph.spw21.mfs.I.mask.fits
| | | | | |-- J1617-7717_ph.spw21.mfs.I.pb.fits.gz
| | | | | |-- J1617-7717_ph.spw21.mfs.I.pbcor.fits
| | | | | |-- J1617-7717_ph.spw23.mfs.I.mask.fits
| | | | | |-- J1617-7717_ph.spw23.mfs.I.pb.fits.gz
| | | | | |-- J1617-7717_ph.spw23.mfs.I.pbcor.fits
| | | | | |-- J1617-7717_ph.spw25.mfs.I.mask.fits
| | | | | |-- J1617-7717_ph.spw25.mfs.I.pb.fits.gz
| | | | | |-- J1617-7717_ph.spw125.mfs.I.pbcor.fits
| | | | |-- qa/
| | | | | |-- hifa_calimage.weblog.tgz
```

(Like above, the file names in the directories "product" and "qa" were shortened by not showing the MOUS UID. Note the additional calibrator images created by the Science Pipeline.)

Also here, as for an analyst-calibrated dataset, you will first need to download and unpack the raw data. This will create an additional subdirectory "raw" containing the ASDM(s). In order to then run "scriptForPI.py", you will need to verify that you have installed the CASA version indicated in the QA2 Report or Weblog. Then start up CASA in the working directory "script" using

```
casa --pipeline
```

At the CASA prompt then type

```
execfile('member.uid_A001_X1284X1a50.scriptForPI.py')
```

The final calibrated MSs will be placed in the directory "calibrated".

The casa_piperestorescript.py can also be run stand-alone without the scriptForPI. See the pipeline documentation for more details.

5.3 Running casa_pipescript.py

In the case of pipeline-calibrated data, there is an alternative way to generate the calibrated visibilities. It uses the script “casa_pipescript.py” which is provided in the “script” directory. In principle the user can modify this script to adjust the calibration but it is recommend to try the delivered version first. The details of the usage of this script are described in the Pipeline documentation (see Section 6 below).

Also the script “casa_pipescript.py” can be run via the scriptForPI. Simply move the “casa_piperestorescript.py” out of the “script” directory. The scriptForPI will then instead use “casa_pipescript.py”. For some pipeline-calibrated data (e.g. Total Power data), the script “casa_piperestorescript.py” may not have been provided. In that case, the scriptForPI will run the “casa_pipescript.py” by default.

5.4 Saving disk space during and after the execution of the scriptForPI.py

The ALMA data sets are increasingly large. If the user would like to save disk space and only work with the calibrated data once it was regenerated with “scriptForPI.py”, he/she can either delete all calibration directories (cd calibrated; rm -rf *.calibration) or, in case of analyst-calibrated data, only delete some or all of the intermediate MSs contained in the calibration directories using, e.g.,

```
cd calibrated; rm -rf *.calibration/*.ms; rm -rf *.calibration/*.ms.split
```

Since the middle of Cycle 1, the scriptForPI.py also offers the “SPACESAVING” option to limit the disk space usage during and after the run of the scriptForPI.py. In order to make use of this, the Python global variable SPACESAVING needs to be set before starting the script, e.g. using

```
cd script
casa -c "SPACESAVING=N; execfile('scriptForPI.py')"
```

where N is an integer from 0 to 3 with the following meaning:

SPACESAVING	= 0	same as not set (all intermediate MSs are kept)
	= 1	do not keep intermediate MSs named *.ms.split
	= 2	do not keep intermediate MSs named *.ms and *.ms.split
	>= 3	do not keep intermediate MSs named *.ms, *.ms.split, and *.ms.split.cal (if possible)
	= -1	do not check disk space

With SPACESAVING=0, the required additional diskspace is up to 14 times as large as the delivered data (products and rawdata) while with SPACESAVING=3 (maximum savings), it is up to 6 times as large. The script will estimate the required disk space and will not execute if there is not sufficient free space available. You can disable the disk space checking by setting the value to -1.

5.5 Splitting out the calibrated data at the end of running the scriptForPI.py

For manually calibrated data, it is standard practice to conclude the calibration by copying the calibrated visibilities into a new MS at the end of calibration. This is done using the CASA tasks *mstransform* or *split* and is called “splitting out” the calibrated data. The final calibrated MSs are named *uid*.ms.split.cal.* They contain only the science spectral windows.

For pipeline-calibrated and -imaged data, this step is by default omitted in order to facilitate the continuation of the processing with the imaging section of the Science Pipeline. So the final MS is called *uid*.ms* and contains all original spectral windows and, what is more, both the calibrated and the original uncalibrated data. The calibrated data can be found in the MS table column “CORRECTED_DATA” which is automatically picked up by the CASA imaging tasks.

If you want to force the scriptForPI.py to perform the “splitting out”, i.e. to create *uid*.ms.split.cal* for each EB, you can set the variable DOSPLIT to True before starting the script:

```
casa -c "SPACESAVING=N; DOSPLIT=True; execfile('scriptForPI.py')"
```

5.6 Analyst-calibrated TP data

If your TP data was reduced manually, you will find scripts “*.scriptForSDCalibration.py” in the script folder. If the scriptForPI.py is also present, it will work, and you can proceed by running it as for other data. For details on manual TP data calibration, please see the official CASA guide available at https://casaguides.nrao.edu/index.php?title=M100_Band3_SingleDish_5.1 .

In older deliveries, the scriptForPI was not present. In that case, you needed to downoad the raw data (ASDMs) from the ALMA archive, remove the '.asdm.sdm' extensions, and then run the scripts ('*.scriptForSDcalibration.py' from the scripts folder) in the same folder with the ASDM.

For the imaging, we recommend you use the provided script, as it contains certain values (e.g. beams) that must be set to a precise value. You could run the script step-by-step, as for the calibration script. Before doing that, you may need to update the paths to the calibrated data, in the msNames variable near the top of the script.

6 Relevant documentation and help

ALMA Quality Assurance is described in chapter 11 of the ALMA Technical Handbook

https://almascience.org/documents-and-tools/cycle5/alma-technical-handbook

For more information on the usage of CASA, please refer to the CASA home page

http://casa.nrao.edu/

Detailed examples are given in the ALMA guides

http://casaguides.nrao.edu/index.php?title=ALMAguides

and the documentation of the ALMA Science Pipeline is available from

https://almascience.org/processing/science-pipeline

If you have any problems with your ALMA data, please contact the ALMA helpdesk at

https://help.almascience.org/

The Atacama Large Millimeter/submillimeter Array (ALMA), an international astronomy facility, is a partnership of the European Organization for Astronomical Research in the Southern Hemisphere (ESO), the U.S. National Science Foundation (NSF) and the National Institutes of Natural Sciences (NINS) of Japan in cooperation with the Republic of Chile. ALMA is funded by ESO on behalf of its Member States, by NSF in cooperation with the National Research Council of Canada (NRC) and the National Science Council of Taiwan (NSC) and by NINS in cooperation with the Academia Sinica (AS) in Taiwan and the Korea Astronomy and Space Science Institute (KASI).

ALMA construction and operations are led by ESO on behalf of its Member States; by the National Radio Astronomy Observatory (NRAO), managed by Associated Universities, Inc. (AUI), on behalf of North America; and by the National Astronomical Observatory of Japan (NAOJ) on behalf of East Asia. The Joint ALMA Observatory (JAO) provides the unified leadership and management of the construction, commissioning and operation of ALMA.