ALMA Memo 623
Mitigation Statistics for ALMA Cycle 7

Amanda A. Kepley,[1] Andrew Lipnicky,[1] Urvashi Rao Venkata,[2] and Remy Indebetouw[1]

[1] *National Radio Astronomy Observatory, 520 Edgemont Road, Charlottesville, VA 22903*
[2] *National Radio Astronomy Observatory, 1003 Lopezville Road, Socorro, NM 87801*

ABSTRACT

The ALMA Pipeline mitigates, or reduces in size, image products to be able to successfully deliver them to PIs. In this memo, we use archive information and pipeline weblogs from ALMA Cycle 7 to determine how many MOUSes were mitigated and what the properties of the migitated MOUSes were. We find that mitigation affects long baseline (> 1km) MOUSes most significantly with mitigation affecting approximately 50% of MOUSes in configuration C43-5/6 and 100% of MOUSes taken in C43-9. While some avenues to reduce the number of mitigated MOUSes are being explored now, processing time estimates for `tclean` and theoretical estimates of the maximum compute load for parallel cube imaging suggest that significant changes to both software (CASA and the Pipeline) as well as the volume of processing hardware are needed to efficiently process large cubes. Fully imaging these cubes would require a factor of 10 increase in storage for imaging products and for science users to have to potentially view and analyze ∼1-2 TB cubes. Ultimately, the reduced quality and less complete nature of the long baseline imaging products affects the ability of PIs and archival researchers to do science with these observations. These issues will only become more severe with the upcoming ALMA Wideband Sensitivity Upgrade (WSU).

1. INTRODUCTION

ALMA observations can produce data that are extremely challenging for the ALMA imaging pipeline (ALMA Pipeline Team 2021, Hunter et al. in prep) and CASA (CASA Team 2022), its underlying computational software, to process because of the large number of pixels per field of view (FOV) and large number of channels. The ALMA pipeline currently mitigates these projects by reducing the size and number of products produced in order to successfully process these projects. To do this, it averages two channels together, reduces the imaged area, reduces the number of pixels per restoring beam, and/or reduces the number of sources and spectral windows (spws) imaged until the size of the image products are below the mitigation limits set by the pipeline. The full heuristic is given in ALMA Pipeline User's Guide (ALMA Pipeline Team 2021) and is reproduced here in Appendix A. Overall, these mitigations reduce the quality of the final image products from optimal, preventing PIs from proceeding directly to science from the delivered, pipeline-processed images. Even more seriously, mitigation means that not all sources and spws observed by ALMA have images and cubes present in the archive, making archival research more difficult. The amount of mitigation possible is limited by numerical factors – the spectral and spatial resolution must be at least Nyquist-sampled – as well as policy decisions since ALMA has chosen to image the entire spectral range. The most extreme cases thus cannot be reduced to an appropriate size and image products are not produced automatically, but must instead be manually imaged, which is time-consuming and human resource intensive.

The goal of this memo is to quantify the amount of mitigation done in operations, the properties of the mitigated data, what would be required to lift these limits, and what the effects of these lifted limits may be. To do this, we use data from ALMA Cycle 7. Although this Cycle was interrupted by the COVID-19 pandemic, it is the most recently completed Cycle to include long baselines and the configuration schedule resumed at the same place it stopped for the COVID-19 shutdown. We describe our sample and derived parameters in Section 2 and present our results in Section 3. Our conclusions are given in Section 4.

2. DATA

For our sample, we included all 2019 projects that had some QA2 pass data taken for them by the end of Cycle 7, excluding solar, total power, and solar system projects. Solar projects were excluded because they have special calibration and imaging requirements that are not typical of ALMA data as a whole. Total power projects follow a different calibration and imaging pathway than interferometric data, so were also excluded. Finally, solar system projects were excluded because the FOV reported by the archive is not the imaging FOV, but instead the area on the sky over which the observations were taken.

We downloaded high level information about these projects from the archive including field of view, resolution, array, spectral window information (frequency, bandwidth, spectral resolution, and polarization), and number of targets. This information is determined from the proposal and observing information rather than derived by the pipeline and thus represents what the observations would have achieved had they not been mitigated. Since the spectral window information in the archive had been transformed to match International Virtual Observatory Alliance (IVOA) standards for reporting quantities (i.e., wavelengths and resolving power), we converted the spectral window parameters to frequencies and reverse engineered the resulting instrumental properties like number of channels and bandwidth based on information in the ALMA Technical Handbook (Cortes et al 2022).

For information on the pipeline processing, we downloaded the pipeline weblogs for all MOUSes from 2019 projects and scraped them for information on the calibration and imaging runtime, image and cube properties including image size and number of channels, and mitigation information[1]. We note that this data set will not include MOUSes that were manually calibrated and/or imaged. It will also not include data that failed mitigation, i.e., either the cube and/or product size could not be reduced to less than the maximum allowed size. These data are either sent to manual imaging (standard procedure) or may be re-run in the pipeline with the default maximum size limits increased by the data reducer (occasionally done by some ARCs).

The information collected from the mitigation stage includes the values of the three control parameters for mitigation:

- maxcubesize (default: 40GB): cubes greater than this size will trigger the cube mitigation (averaging two channels, reducing the imaged area, and/or reducing the number of pixels per beam). This quantity can be thought of as the threshold at which cube mitigation starts.

- maxcubelimit (default: 60GB): maximum cube size that the pipeline will image. If the final mitigated cube size is larger than this, the imaging pipeline will stop with "mitigation failed". This parameter also controls the total number of large cubes produced.

- maxproductsize (default: 350GB (Cycle 7 pipeline), 500GB (Cycle 8 pipeline): total product size at which the number of science targets imaged is reduced. The cubes will be mitigated if necessary. If the maxproductsize limit is exceeded, the maximum number of targets that will be imaged is 30 (out of a maximum of 150 targets allowed by ALMA).

The pipeline defaults for these parameters are listed above. However, the default mitigation limits may be lifted to avoid manual processing depending on the ARC that data is processed at. We also collected information on the following values calculated by the pipeline during the mitigation stage:

- predicted cube size,
- mitigated cube size,
- initial product size, and
- mitigated product size.

Finally, we obtained information on what mitigations were used for the mitigated MOUSes:

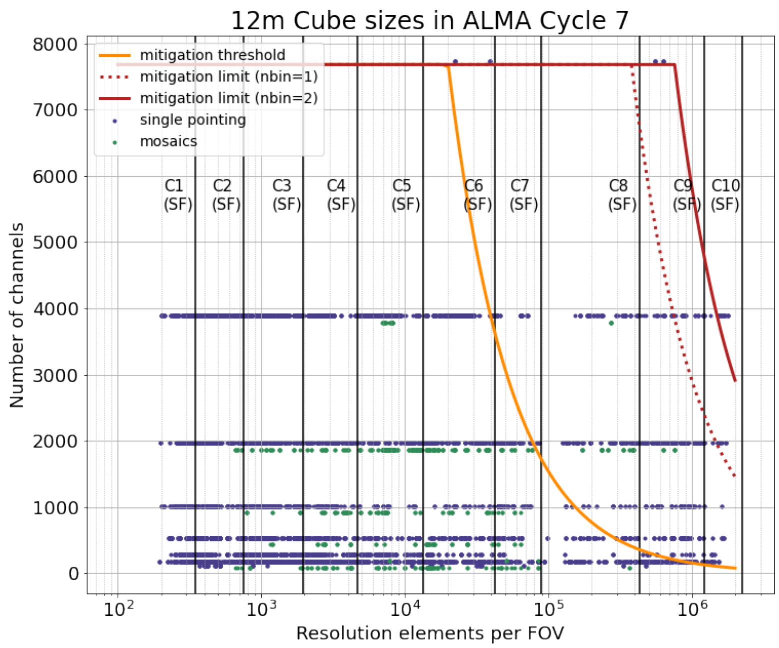
- nbins: bin channels by 2
- imsize: reduce the field of view (for single fields only)
- cell: reduce the number of pixels per beam from 5 to 3.
- field: reduce the number of fields imaged (representative target is always retained)
- spw: reduce the number of spectral windows imaged (representative spw is always retained)

3. RESULTS

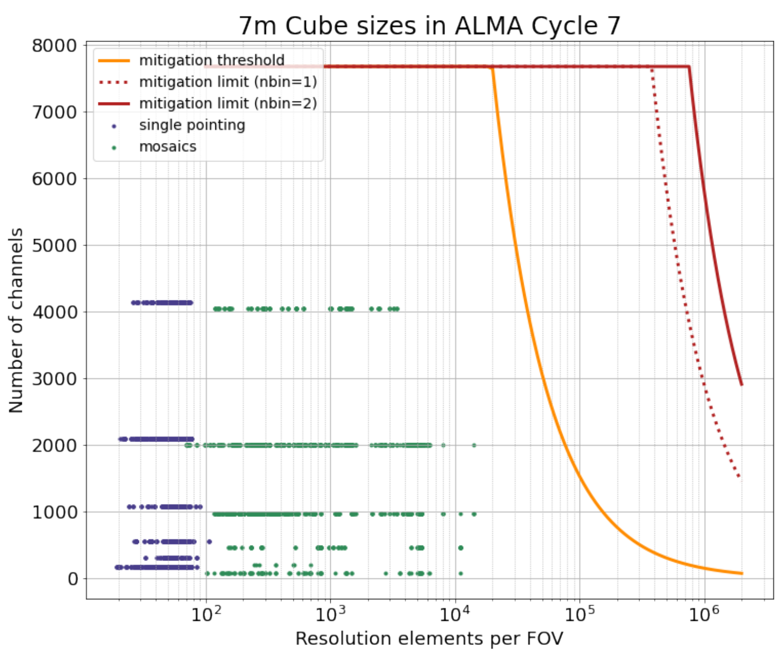
Figure 1 shows the resolution elements per field of view[2] and number of channels derived from the archive values for our sample for the 12m and 7m array, respectively. These are the image parameters that would have been achieved had there been no mitigation. The threshold at which the cubes will start to be mitigated and the maximum cube size possible are indicated. This plot demonstrates that the cube size mitigation will only affect 12m data and that data sets with higher spatial resolutions (i.e., more resolution elements per field of

[1] https://github.com/indebetouw/weblogstats

[2] These values can be converted to approximate linear imsizes by multiplying by 25 pixels^2/beam, taking the square root, and multiplying by a factor of 1.54 to image down to the 0.2 PB level.



(a)



(b)

Figure 1. The number of channels as a function of resolution elements per field of view for ALMA Cycle 7 data for the 12m array (a) and the 7m array (b). Blue points indicate single pointings and green points mosaics. The threshold above which cubes will start being mitigated is given as an orange line. The maximum cube size is indicated with red lines with the dotted line showing the maximum if the channels in the cube cannot be binned and the solid line showing the maximum if the channels can be binned. For (a), the configuration corresponding to the number of resolution elements per FOV for a single field is indicated.

Table 1. Mitigation Statistics for Cycle 7 MOUS Sample

Category	All	Mitigated	% Mitigated
Total	2565	325	13
12m	1672	325	19
7m	893	0	0

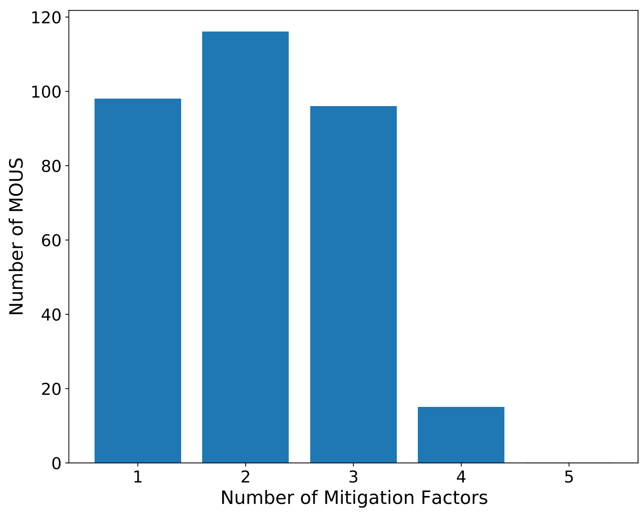


Figure 2. Number of mitigation factors triggered per MOUS in our sample. The maximum number of mitigation factors that can be triggered is five, but no MOUS triggered all five.

Table 2. Types of Mitigation Done For Cycle 7 MOUS Sample

Type	Number	Fraction
Cube size	77	24
Cube size and number of spws	150	46
Cube size and number of fields	12	4
Cube size, number of spws, and number of fields	13	4
Number of spws and fields	73	22

view) will be more significantly impacted. However, the cube size mitigations only set a lower limit on the mitigation done by the pipeline: projects with cubes below the allowed cube size limits but with large numbers of targets or spectral windows may still be mitigated based on overall product size.

Table 1 shows the number of MOUSes mitigated by the pipeline in Cycle 7 as determined from their weblog information. We find that none of the 7m MOUSes were mitigated, but that 19% of the 12m MOUSes were mitigated, which is consistent with our expectations from Figure 1. Figure 2 shows that a majority (2/3) of MOUSes are impacted by multiple mitigation factors, while Figure 3 shows the number of MOUSes triggering each type of mitigation. The top three mitigations employed are to reduce the FOV (imsize), to reduce the number of spws (spw), and to reduce the number of pixels per beam (cell). Reducing the number of fields (field) and binning the channels by a factor of two (nbin) are relatively rare. The former is by design to retain at least some images and cubes for as many sources as possible in the data set. The latter is because binning the number of channels by a factor of 2 only makes sense if the correlator has already not done this online.

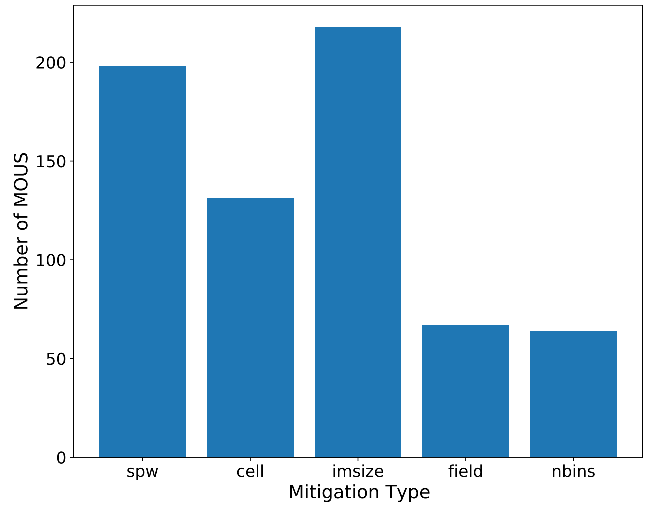


Figure 3. The number of MOUSes triggering each type of mitigation. Each MOUS can triggered multiple types of mitigation, so the total number of mitigations is greater than the total number of MOUSes.

While there are multiple different combinations of mitigation factors, we break them into five different categories. First are the cube size only mitigations. These mitigations are relatively benign in that all image products are produced, just with less than optimal quality, e.g., fewer pixels per beam, averaged channels, and/or smaller fields of view. These mitigations, however, only happen in 24% of cases. For the other 76% of the mitigated MOUSes, the effects of mitigation are more severe with the number of overall products reduced. The most commonly applied mitigation (in addition to the cube size) is a reduction in the number of spws imaged, although a significant fraction (26%) have both number of spws and number of fields imaged reduced. These results support our initial conclusion that the 12m array would be most affected and that while cube size mitigations would be important, product size mitigations would also play a significant role. We also note that of the mitigated MOUSes in our sample, 9% had mitigation limits lifted manually in order to allow the MOUSes to run through the pipeline. The mitigated and unmitigated 12m MOUSes have similar distributions of number of spws and number of sources. The unmitigated 12m MOUSes have twice the percentage of mosaics as the mitigated 12m MOUSes.

To investigate the combined effects of the different mitigations, we calculate the ratio of the mitigated product size to the initial product size, which we call the completion fraction, for all Cycle 7 12m MOUS in our sample. Figure 4 shows the completion fraction as a function of the 80th percentile baseline length (L80) with the notional ALMA configurations as indicated on the top axis of the plot. This figure confirms that longer baseline (L80 $>$ 1.0km) data are significantly impacted with many MOUSes showing a completion fraction less than 50%. In particular, data taken in configurations $\geq$ C43-8 often have a completion fraction under 30%. However, this is not just an effect that is confined to the longest baselines. Some MOUSes in configurations between C43-4 and C43-7 show completion fractions on the order of 10% or less.

Figure 5(a) compares the mitigated number of MOUSes (i.e., completion fraction less than one) to the total number of MOUSes in our sample as a function of 80th percentile baseline. It shows that the number of mitigated MOUSes increases as the 80th percentile baseline increases. The fraction of mitigated MOUSes as a function of 80th percentile baseline is given in Figure 5(b). From this figure, we can see that approximately 50% of the MOUSes are mitigated for configurations with an L80 of $\sim$1km (C43-5/6) and 90-100% of the MOUSes are mitigated for the longest baseline configurations (C43-8/9).

Ideally, the ALMA project would like to produce complete data products for all projects. We can estimate the computing time needed to process each Cycle 7 12m MOUS in our sample by assuming that imaging time scales linearly. For this assumption, the estimated time can be derived by dividing the Cycle 7 imaging time by the completion fraction calculated above. Figure 6 shows the estimated imaging time in days as a function of the 80th percentile baseline length (L80) for all Cycle 7 12m MOUSes. Mitigation caps the imaging runtime at about 10 days. If the migitation limits were completely lifted, then the imaging times would significantly increase beyond 10 days to up to 1.7 **years**. Even if cluster up-times could be maintained over the entire time[3], this is an extremely long time scale in terms of astronomical career times scales: it is equivalent to 1/3 of a Ph.D., an entire postdoc, and 2/3 of an NSF AST grant. This estimate suggests that the total time it would take to image all Cycle 7 12m MOUSes would be **41** years! This estimate, however, assumes that only one node is processing jobs at a time. In operations, JAO uses ten nodes with 8 cores and 256GB memory full time. Even with 10 nodes, however, it will still take 4.1 years to image all of Cycle 7 12m MOUSes, assuming no hardware failures, software limitiations, or reprocessing.

Previous experience shows that processing very large cubes can be highly unstable, often with stochastic failures (C. Brogan, private communication). The current mitigation limitation was empirically chosen to minimize these failures in operations. However, a subsequent theoretical analysis of the cube parallelization mechanism in the CASA `tclean` task shows that the mitigation limit matches the limit for efficient cube processing for the operational computing setup of one imaging run spread across 8 cores and 256GB even with the assumption that the software scaled perfectly.

Cube parallelization in `tclean` is achieved by splitting a cube into chunks of $n_{channels}$. Major and minor cycles run independently for each chunk with a synchronization step at each major-minor cycle boundary to update iteration control state using information from the entire cube. The number of channels per chunk is chosen to maximize the parallelization breadth given the memory required per channel. The amount of compute that is being done in the minor cycle is proportional to the amount of flux being deconvolved and the number of iterations. For a fixed data volume, the compute during the major cycle also scales with the number of image pixels and so the minor cycle iteration count may be used as a proxy for any computations that scale linearly with the number of pixels in the image. The total computing load per process is therefore proportional to the product of the number of channels per chunk and the number of iterations done per channel.

To estimate the computing load, we assume that there is flux in the central quarter of the image and 100 iterations are needed per pixel. This assumption may be an overestimate, but it does occur in practice for complex images. The `tclean` task needs the equivalent of 10 copies of the images per plane represented as floating point numbers (at its peak memory usage). This estimate is compared with the available memory per pro-

[3] Very unlikely based on the experience of these authors.

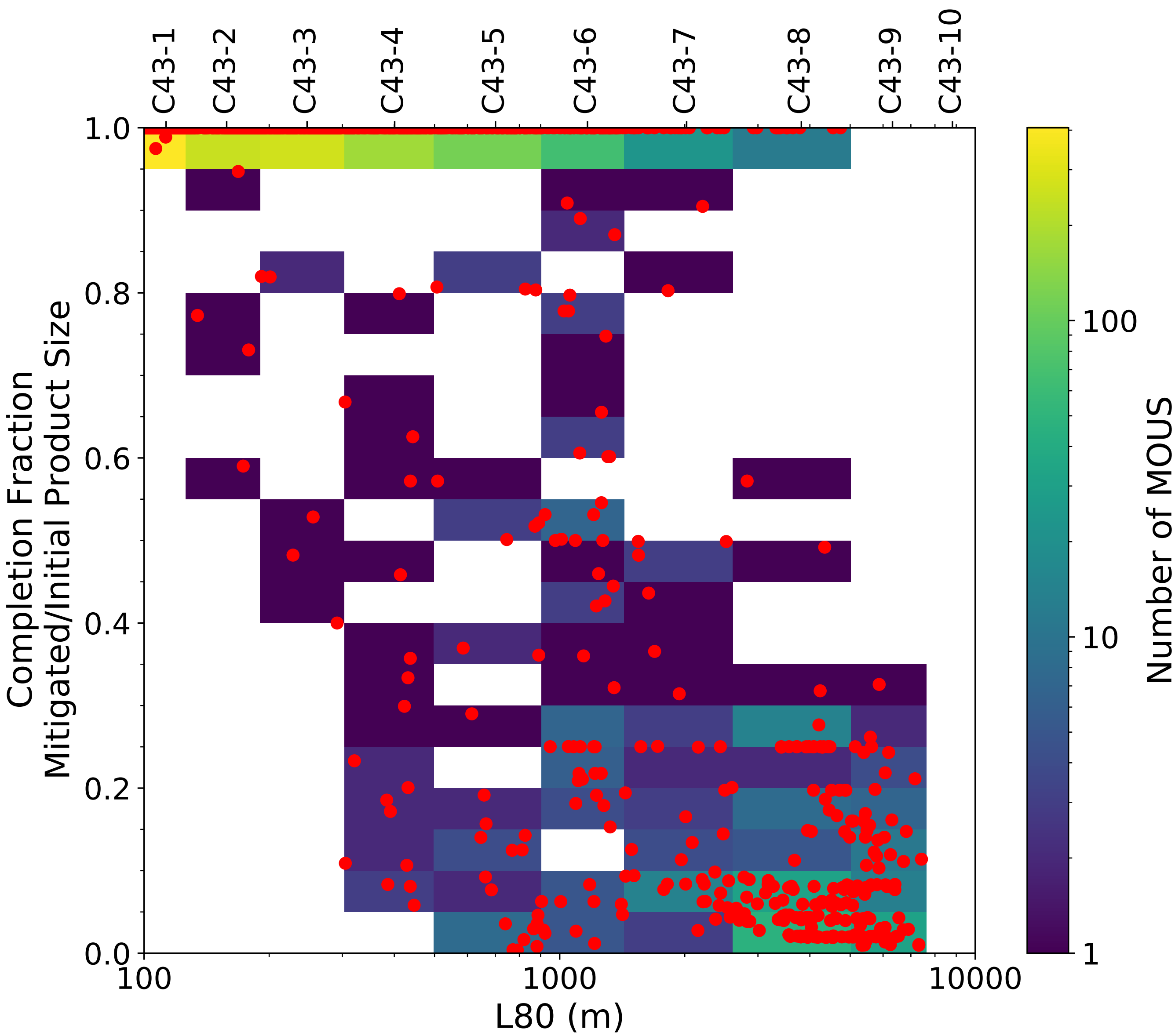
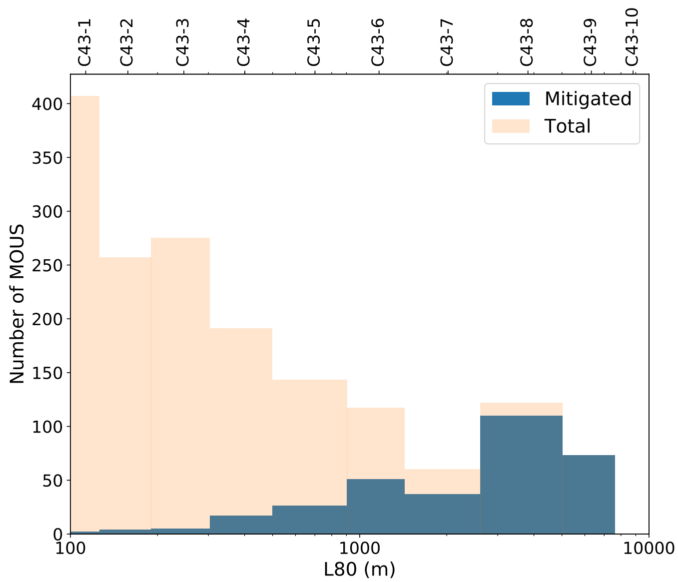


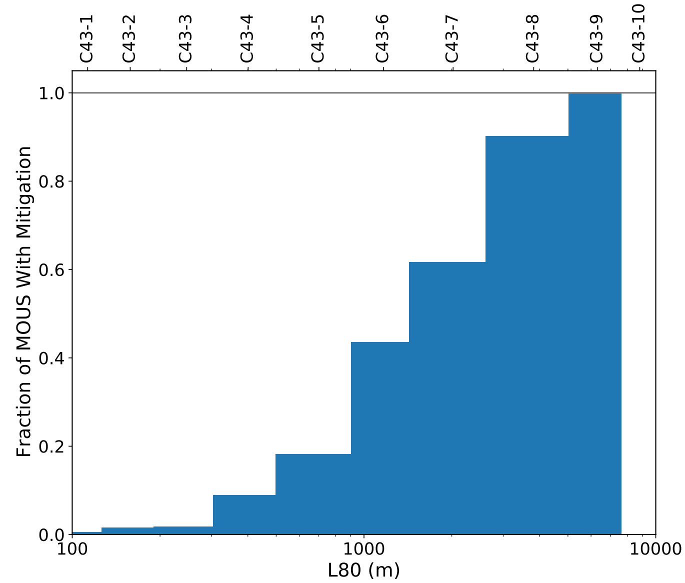
Figure 4. The completion fraction, or ratio of the mitigated product size to the initial product size, as a function of the 80th percentile baseline (L80) for all the Cycle 7 12m MOUSes in our sample. The corresponding notional configurations are listed on the top axis. The red circles show individual MOUSes and the image shows the number of MOUSes per two dimensional bin in L80 and completion fraction.

cess, to place an upper limit on the number of channels per chunk. Then, the number of available processes is used to determine if a smaller number of channels per chunk may allow for greater parallelization breadth than set by memory limits alone. These parallelization breadths are then compared and whichever produces the maximum parallelization breadth without exceeding the memory limits is selected. The compute load is then just the number of iterations per channel times this optimal number of channels per chunk.

In Figure 7, we show the result of the above calculation assuming we are processing using a standard JAO node with 8 cores and 256GB total memory. The computing load estimate shows that the computing load saturates in the upper right hand corner of the plot. In this region, there are more chunks than cores and thus chunks are required to queue up to be processed at every major-minor cycle boundary. The transition between the unsaturated and saturated region in the theoretical plot is just below the mitigation limit currently used in operations (60GB). We stress that this limit was chosen *empirically* to reduce the number of unreproducible failures in operations. One theory is that these failures are



(a)



(b)

Figure 5. (a) The total number and mitigated MOUSes as a function of the 80th percentile baseline length (L80) for all Cycle 7 12m MOUSes. (b) The fraction of mitigated MOUSes as a function of the 80th percentile baseline length (L80). The gray line indicates a mitigation fraction of 1, i.e., all MOUSes were mitigated. In both panels, the corresponding configurations for the given L80 values are given on the top axis of the plot.

more likely for cubes beyond the mitigation limit due to the increased wait time to process the queued up chunks.

Increasing the number of cores used by `tclean` would allow us to increase the mitigation line. However, we are currently limited to a single node for `tclean` because the current cube[4] parallelization implementation does not scale appropriately across multiple nodes (as found in benchmarking for the CASA Next Generation Infrastructure (CNGI) prototype testing[5] and Kepley, Madsen, Robnett, and Rowe, in prep). There is also no built-in fault-tolerance to allow for occasional hardware related (stochastic) failures.

However, the theoretical calculation also suggests that even if ALMA had access to cube imaging software that scaled perfectly to any number of processes but followed the same generic heuristics for calculating the number of channels per chunk that fit within reasonable memory limits, un-mitigated ALMA imaging runs would require 2 to 3 orders of magnitude more processing power than is currently used in JAO operations. Note that Figure 7 is a log-log plot and the compute cost function scales linearly with the number of pixels. Thus the upper right region in this figure actually represents a large range of linear space in compute costs.

Finally, in addition to the processing requirements, lifting the mitigation limits would require increased storage and would increase the difficulty of viewing and doing scientific analysis on the largest cubes. Figure 8 shows the distribution of maximum cube sizes and total product sizes for the data as currently delivered by the pipeline and what the distribution of the unmitigated maximum cube sizes and total product sizes would be.[6] The total estimated data volume for all 12m image products delivered in Cycle 7 is approximately 94TB. If we lifted the mitigation limits completely, the total estimated data volume for the 12m image products would increase by a factor of 10 to 950TB. Lifting the limits would require science users to view and analyze up to ∼1-2 TB cubes in some cases. While CARTA[7] can load cubes that large, performing analysis on these cubes is still difficult and requires significant computational resources, which decreases the accessibility of the data.

4. CONCLUSIONS

The ALMA Pipeline currently mitigates, or reduces the volume of imaging products produced, to allow

[4] We note that continuum imaging with CASA, however, follows a different parallelization mechanism and has been shown to scale to a few hundred processes, spread across nodes.

[5] https://cngi-prototype.readthedocs.io/en/latest/introduction.html

[6] These data sizes are estimated by the pipeline and should be close to the actual data sizes.

[7] https://cartavis.org/

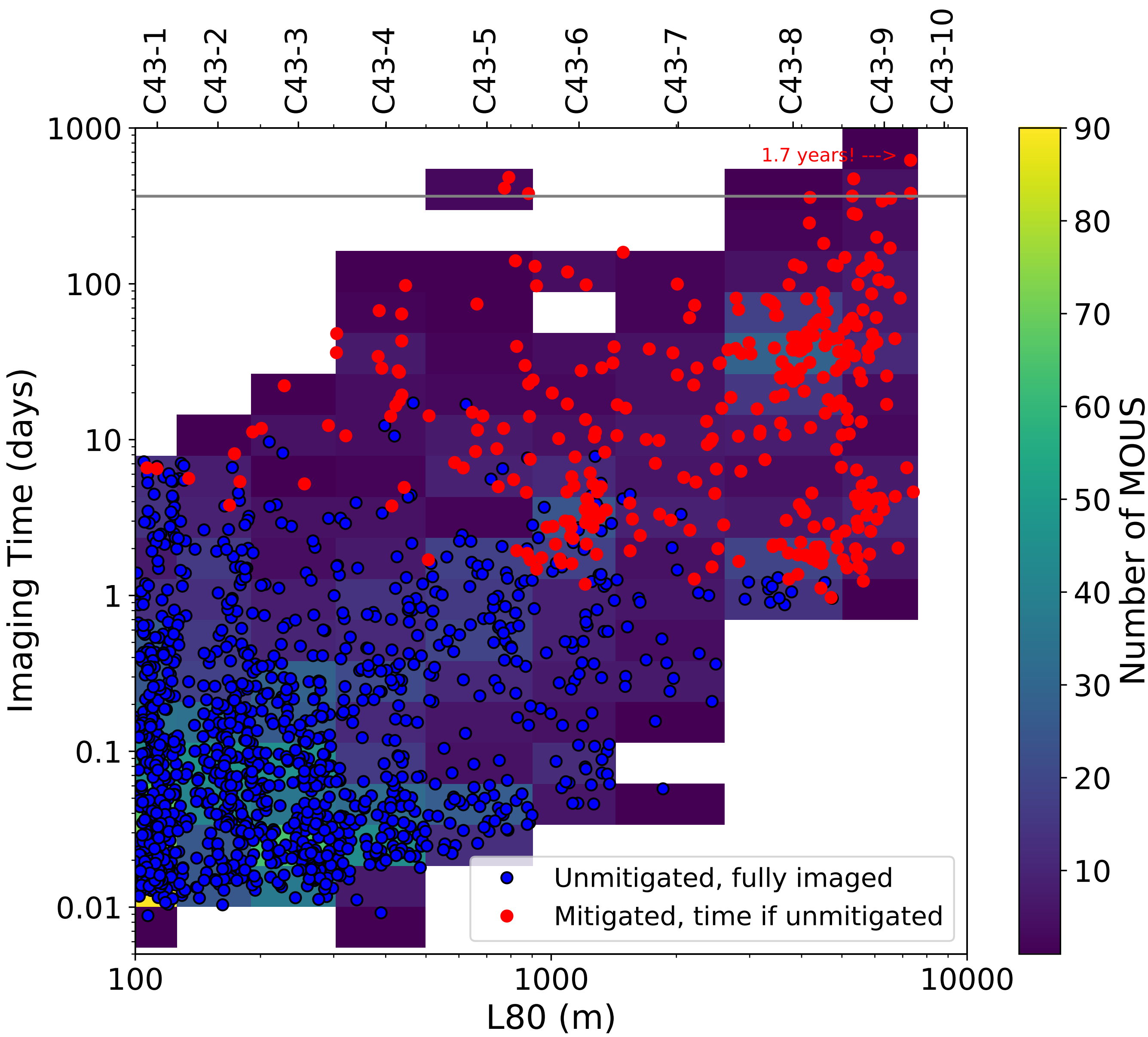


Figure 6. Imaging time in days as a function of the 80th percentile baseline length (L80) for all Cycle 7 12m MOUSes. The blue dots show the unmitigated data that was fully imaged by the pipeline and the red dots show the estimated imaging time for the mitigated data. This estimate was computed by dividing the Cycle 7 imaging time for a MOUS by its completion fraction. For example, projects with a 50% completion fraction are estimated to take twice as long to image if it were unmitigated.

MOUSes to be processed successfully and delivered to the PI. The mitigation strategies employed range from averaging two channels together, reducing the size of the imaged area, reducing the number of pixels per restoring beam, and/or reducing the number of sources and spws imaged, in order of most preferred to least preferred. In this memo, we show that mitigation most significantly affects data taken in configurations with an L80 greater than 1km (>C43-5) with 50% of observations in C43-5/6 mitigated and 100% of observations in C43-9 mitigated.

If we were to remove the mitigation limits from the pipeline, we estimate that some of the projects that are currently mitigated would take up to 1.7 **years** to image. This timescale is extremely long compared to the graduate student, postdoc, and grant timescales. This estimate is an optimistic one in that it does not take into account any fundamental hardware limitations (mem-

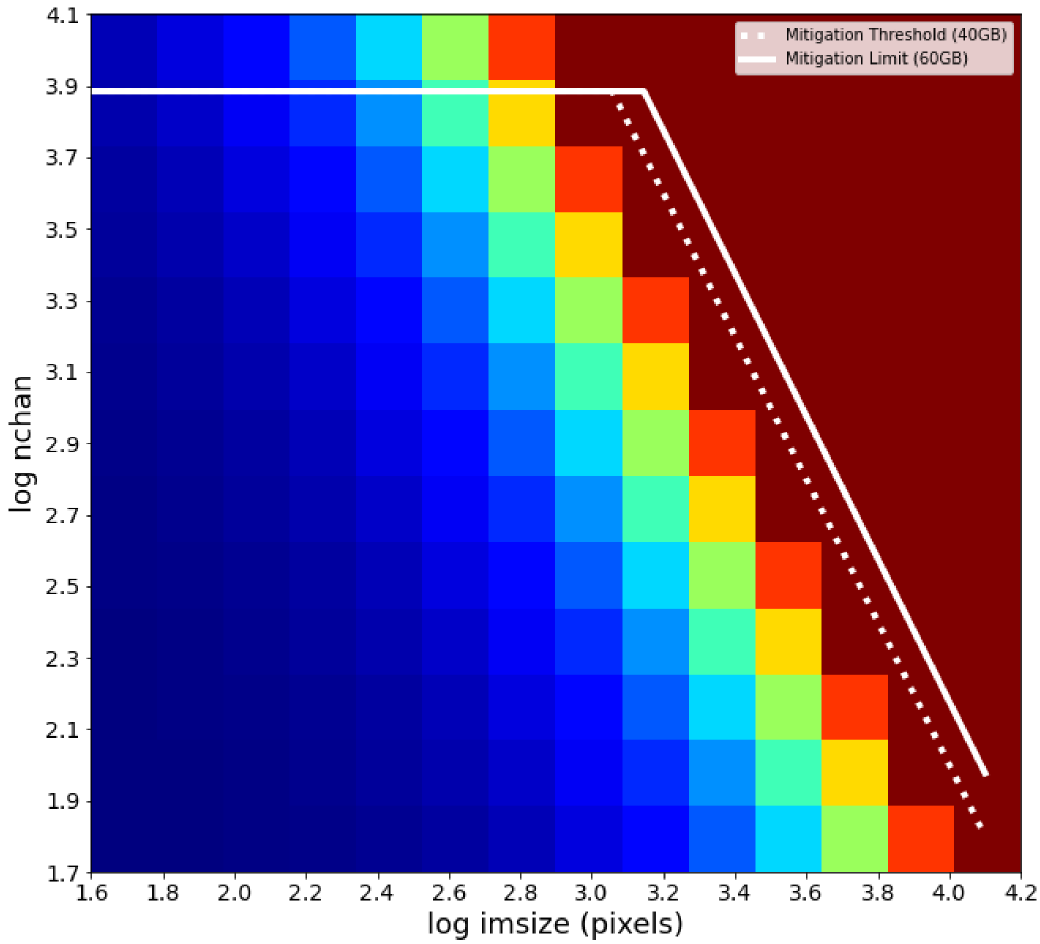


Figure 7. Theoretical compute load estimate for the cube parallelization mechanism in the CASA `tclean` task as a function of image size and number of channels. The current empirical mitigation threshold (40GB) and limit (60GB) are shown as a dotted and solid white lines, respectively.

ory, storage, data transfer, etc),[8] software limitations associated with CASA or the Pipeline, or the need to reprocess data to fix issues. In addition, theoretical estimates that assume perfectly scaling software suggest that our current mitigation limit is close to the theoretical maximum compute load for the chosen compute. Producing additional data products would also increase the total volume of imaging data by a factor of 10 and require science users to view and analyze cubes up to ∼1-2 TB in size.

[8] As an example, there is typically a issue with lustre every six months or so that requires a restart or downtime.

In a forthcoming memo (Kepley, Madsen, Robnett, and Rowe, in prep), we will report the results of tests of the performance of the ALMA Pipeline and CASA to compare to the estimates given above. Initial results largely confirm our prior experience that imaging very large cubes is difficult and is affected by a host of issues ranging from the software limitations to hardware uptime. Although producing larger cubes is currently prohibitive, we are investigating some potential modifications to the total product size mitigations at the expense of longer pipeline run times because they do not result in larger cubes, just larger overall data volumes.

Ultimately, mitigation limits the quality and quantity of imaging products provided to the PIs of long

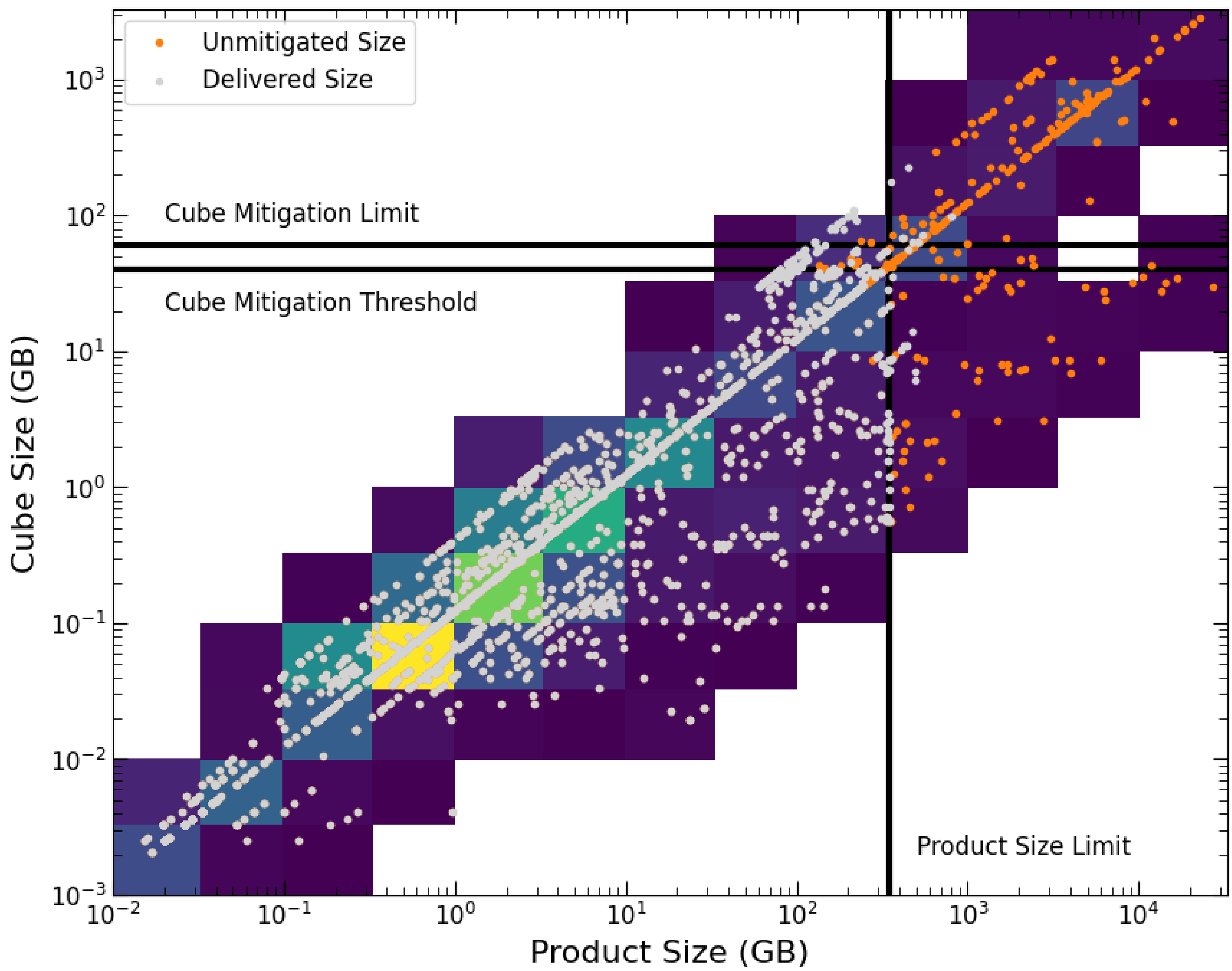


Figure 8. Total product size versus maximum cube size derived from the weblog for each Cycle 7 MOUS in our sample. The cube mitigation threshold (40GB) and mitigation limit (60GB) are indicated as well as the product size limit (350GB). Grey points show the size of the delivered products and orange points show the unmitigated size of the products. Recall that 9% of the mitigated MOUSes in our sample have had their mitigation limits lifted and thus appear beyond the mitigation thresholds and limits indicated on the plot. The product size limit was also raised from 350GB to 500GB between the Cycle 7 and 8 versions of the pipelines, so data taken in cycle 7 and processed in the cycle 8 pipeline would have a higher limit. The background color map represents the relative density of MOUSes in that bin.

baseline projects as well as the overall utility of the ALMA archive for this type of science. Long baseline (> 1.5km), high resolution observations form a key part of the ALMA science case. They are reflected in the original level 1 science directly as the "The ability to provide precise images at an angular resolution of 0.1″". Improving image processing for long baselines would further enhance the ability of this use case to produce significant scientific results. Based on our preliminary investigations, fully imaging long baseline data will likely require significant changes to the hardware and software used for processing, but it may be possible to produce more complete imaging products for entire MOUSes with less than optimal quality. The issues posed by long baseline cubes are only going to become more significant with increased channels provided by the proposed ALMA Wideband Sensitivity Upgrade.

APPENDIX

A. ALMA PIPELINE MITIGATION HEURISTIC

Below we have copied the mitigation heuristic used in the ALMA Pipeline as of PL2022 (Cycle 9 pipeline) from the ALMA Pipeline User's Guide.

Step 1: If cubesize > maxcubesize, for each spw that exceeds maxcubesize:

a. If (nchan == 3840) or (nchan in (1920, 960, 480) AND online channel averaging was NOT already performed), then set nbin=2.

b. If still too large, then calculate the Gaussian primary beam (PB) response level at which the largest cube size of all targets is equal to the maximum allowed cube size. The cube sizes are initially calculated at primary beam power level PB=0.2. For an image of width d, the response level at the edge will be PB=exp(-d^2*ln(2)/$FWHM^2$), the image size $d^2 \propto$ -ln(PB), and the required power level to create an image of size = maxcubesize is:

PB_mitigation = exp(ln(0.2) * maxcubesize / current_cubesize)

   i. Then account for imsize padding: PB_mitigation = 1.02 * PB_mitigation
   ii. Then limit the size reduction to PB=0.7: PB_mitigation = min(PB_mitigation, 0.7)
   iii. Then round to 2 significant digits: PB_mitigation = round(PB_mitigation, 2)

NOTE: this mitigation cannot be applied to mosaics, only single fields, and the same mitigated FoV is used for all science target image products.

c. If still too large, change the pixels per beam (cell size) from 5 to 3.25 (if robust=+2) or 3.0 otherwise.

d. If still too large, *stop with error*, the largest size cube(s) cannot be mitigated.

Step 2: If productsize > maxproductsize

a. If the number of science targets (single fields or mosaics) is greater than 1, reduce the number of targets to be imaged until productsize < maxproductsize. The representative target is always retained.

b. If productsize still too large, repeat steps 1a, 1b, and 1c, recalculating productsize each time.

c. If productsize is still large, *stop with error*, the productsize cannot be mitigated.

Step 3: For projects with large cubes that can be mitigated, restrict the number of large cubes that will be cleaned:

a. If there are cubes with sizes greater than 0.5 * maxcubelimit, limit the number of large cubes to be cleaned to 1. The spw encompassing the representative frequency shall always be among the cubes retained.

Step 4: For projects that have many science targets, limit the number to be imaged to 30, the representative target is always retained in the list.

REFERENCES

ALMA Pipeline Team, 2021, ALMA Science Pipeline User's Guide, ALMA Doc 2021.2v1.0

CASA Team, 2022, CASA, the Common Astronomy Software Applications for Radio Astronomy, arXiv:2210.02276

Cortes, P. C., Remijan, A., Hales, A., Carpenter, J., Dent, W., Kameno, S., Loomis, R., Vila-Vilaro, B.,Bigg, A, Miotello, A., Vlahakis, C., Rosen, R., Stoehr, F., Saini, K., 2022, ALMA Technical Handbook, ALMA Doc. 9.3, ver. 1.0ISBN 978-3-923524-66-2