1) ATOMIUM ALMA Large Programme

2018.1.00659.L, PIs Decin, Gottlieb 2021 December

CONTENTS

2) Data overview and introduction

The ATOMIUM Large Programme has the main goal of understanding the dynamics and the gas phase and dust formation chemistry in the winds of evolved asymptotic giant branch (AGB) and red supergiant (RSG) stars. See Decin et al. https://ui.adsabs.harvard.edu/#abs/2020Sci...369.1497D/abstract and Gottlieb et al. A&A accepted https://ui.adsabs.harvard.edu/#abs/2021arXiv211204399G/abstract (G21) for early results and more details. Figs. 1 and 2 give an overview of the stars and frequency coverage. This document applies to the data processing for all stars observed.

cube	$\nu_{\rm central}$ (GHz)	Δv (km/s)	δv (km/s)	SG
00	214.8	2598	1.36	a/e
01	217.0	2572	1.35	a/e
02	221.2	2523	1.32	b
03	224.6	2485	1.30	b
04	228.2	2445	1.28	a/e
05	230.5	2420	1.27	a/e
06	236.4	2360	1.24	b
07	239.7	1164	1.22*	b
08	244.5	1141	1.20*	c/f
09	246.3	2266	1.19	c/f
10	252.6	2209	1.16	d
11	254.9	2189	1.15	d
12	259.6	2149	1.13	c/f
13	262.6	1062	1.12*	c/f
14	266.5	2093	1.10	d
15	268.7	2076	1.09	d

Figure 1. (a) Frequency ranges. *Cube total frequency width 937.5 MHz (to fit in the tuning); all other cubes 1875 MHz. Δv is velocity span per cube, δv is channel spacing corresponding to 0.9765625 MHz. Also see Fig. 2.

Star$^{(a)}$	Variability type$^{(b)}$	$v_{\rm LSR}$ ALMA obs. (km/s)
S Pav	SRa	−20.0
T Mic	SRb	25.3
U Del	SRb	−6.4
RW Sco$^{(e)}$	Mira	−72.0
V PsA	SRb	−11.1
SV Aqr	LPV	8.5
R Hya$^{(f)}$	Mira	−11.0
U Her	Mira	−14.5
π^1 Gru$^{(f,g)}$	SRb	−13
AH Sco	SRc	−4.0
R Aql$^{(e)}$	Mira	47.0
W Aql$^{(f,g)}$	Mira	−25.0
GY Aql	Mira	34.0
IRC −10529$^{(e)}$	Mira	−18.0
KW Sgr	SRc	4.0
IRC +10011$^{(e)}$	Mira	10.0
VX Sgr	SRc	5.3

(b) ATOMIUM stars. Observed $v_{\rm LSR}$ is not necessarily the accurate stellar velocity, see G21 for inferred velocities.

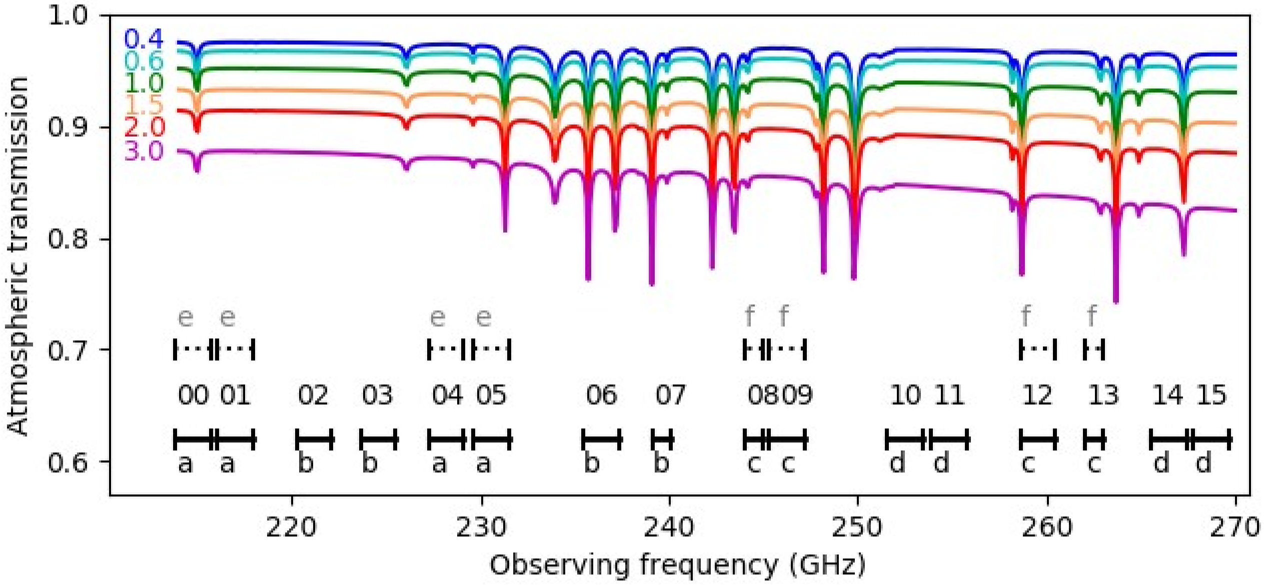

Figure 2. The frequency coverage of ATOMIUM, overlaid with atmospheric transmision at various values of precipitable water vapour in mm. The letters (a, b, c, d) and (e, f) denote the observing tunings (Science Goals, SG) for (MID and EXTENDED) and (COMPACT) configurations respectively. The original spw numbers in the MS are mostly different from the final, frequency-ordered image cube numbers 00 - 15.

The observations were initially processed by the ALMA observatory using the standard pipeline or scripts; we refer to the resulting datasets and cubes as archive data, which can be downloaded from the ALMA Science Archive. We extracted the target data and performed self-calibration on the stellar continuum, applied this to all data, subtracted the continuum and made image cubes, as described in G21. The half-width spw (Fig. 1(a)) originally have half the channel separation given but were averaged for consistency with the rest of the spw. There are two sets of visibility data, one at the high spectral resolution as in Fig. 1, and the other averaged to 15.25 MHz channels for continuum imaging. We refer to the products we have processed and returned to the ALMA archive as LP products. All measurement sets (MS) and some other products are in compressed tar form (*.tgz) and can be extracted using tar -zxvf <file>.tgz . Here, we use <> or * to denote a file name for you to fill in and we omit the .tgz when referring to files.

We assume a basic knowledge of CASA and ALMA data processing; please consult CASA Guides and your local ALMA Regional Centre if you need assistance.

3) LP products data delivery contents and naming conventions

Names of form <gous><lp><star>_<config>

<gous> = 'group.uid___A001_X133d_X11bc'

is the observing group ID for the (alphabetically) first group covering a star

<lp_ldecin> = 'lp_ldecin' denotes a Large Programme, P.I.s Leen Decin and Carl Gottlieb

<star> = 'W_Aql' is the target

<config> is the array configuration, i.e. 'extended' and 'mid' (and 'compact', if used) 'combined' denotes all configurations combined.

MEASUREMENT SETS

'cont' denotes the data averaged every 16 channels

In individual configurations the lines are still present

'line' denotes the data at full resolution of 0.9765625 MHz,

'contsub' are line data after continuum subtraction, as used for all image cubes.

Each measurement set here contains all the tunings for that configuration, with 8 spw for compact (none for AH Sco, KW Sgr) and 16 for the others and combined.

```
group.uid___A001_X133d_X11bc.lp_ldecin.W_Aql_extended_cont.ms.tgz
group.uid___A001_X133d_X11bc.lp_ldecin.W_Aql_extended_cont.ms.flagversions.tgz
group.uid___A001_X133d_X11bc.lp_ldecin.W_Aql_extended_line.ms.contsub.tgz
group.uid___A001_X133d_X11bc.lp_ldecin.W_Aql_extended_line.ms.tgz
group.uid___A001_X133d_X11bc.lp_ldecin.W_Aql_extended_line.ms.flagversions.tgz
group.uid___A001_X133d_X11bc.lp_ldecin.W_Aql_mid_cont.ms.tgz
group.uid___A001_X133d_X11bc.lp_ldecin.W_Aql_mid.cont.ms.flagversions.tgz
group.uid___A001_X133d_X11bc.lp_ldecin.W_Aql_mid_line.ms.contsub.tgz
group.uid___A001_X133d_X11bc.lp_ldecin.W_Aql_mid_line.ms.tgz
group.uid___A001_X133d_X11bc.lp_ldecin.W_Aql_mid_line.ms.flagversions.tgz
group.uid___A001_X133d_X11bc.lp_ldecin.W_Aql_compact_cont.ms.tgz
group.uid___A001_X133d_X11bc.lp_ldecin.W_Aql_compact_cont.ms.flagversions.tgz
group.uid___A001_X133d_X11bc.lp_ldecin.W_Aql_compact_line.ms.contsub.tgz
group.uid___A001_X133d_X11bc.lp_ldecin.W_Aql_compact_line.ms.tgz
group.uid___A001_X133d_X11bc.lp_ldecin.W_Aql_compact_line.ms.flagversions.tgz
group.uid___A001_X133d_X11bc.lp_ldecin.W_Aql_combined_cont.ms.tgz
group.uid___A001_X133d_X11bc.lp_ldecin.W_Aql_combined_line.ms.contsub.tgz
```

The data with self-calibration applied are in the CORRECTED column of the .ms and the DATA column of .ms.contsub.

CUBES

The line image cubes are denoted by 'cube' followed by numbers '00' to '15', denoting ascending frequency order (not the original spw numbering).
The channels are also numbered in ascending frequency which, in the case of cubes 00,01,02,03,08,09,10,11 is the reverse of the order originally created.
The apparent rest frequencies/velocities were set to optimise the whole spw span and do not necessarily correspond to a real transition.

compact image, image.pbcor and mask fits files
mid: image, image.pbcor and mask fits files
extended: image, image.pbcor and mask fits files
combined: image and image.pbcor fits files
e.g.
group.uid___A001_X133d_X11bc.lp_ldecin.W_Aql_combined_cube00.pbcor.fits

CONTINUUM IMAGES

Excluding line channels, all configurations: image and mask fits files;
'fin' denotes that the final state of calibration was applied.
e.g
group.uid___A001_X133d_X11bc.lp_ldecin.W_Aql_combined_cont_mfs.fin.fits

SPECTRA

all spectra in one tar file per star:
`group.uid___A001_X133d_X11bc.lp_ldecin.W_Aql_spec.tgz`
Extracted from each cube for each configuration at the resolution in the name of each spectrum after extraction; also an rms spectrum.
All in ascending frequency order in the LP products

SCRIPTS
all scripts in one tar file per star, explaining the data reduction steps and imaging:
`group.uid___A001_X133d_X11bc.lp_ldecin.W_Aql_scripts.tgz`
Separate configurations:
```
W_Aql_extended_getSpecAtom.py
W_Aql_extended_SelfCal.py
W_Aql_mid_getSpecAtom.py
W_Aql_mid_SelfCal.py
W_Aql_compact_getSpecAtom.py
W_Aql_compact_SelfCal.py
```

Combined:
```
W_Aql_combined_catalign.py
W_Aql_combined_CubesComb.py
W_Aql_combined_getSpecCombined.py
W_Aql_combined_ImagComb.py
```

The scripts used for data reduction are described in detail in
group.uid___A001_X133d_X11bc.lp_ldecin.description.pdf
See also Gottlieb et al, A&A, accepted,
https://ui.adsabs.harvard.edu/#abs/2021arXiv211204399G/abstract

Notes

1) All positions are in ICRS. The observing positions used Hipparcos proper motions so in some cases the stars are slightly off-centre. For EXTENDED where this was more than a beam the images have been centred and the EXTENDED positions were used to align the COMBINED data.

2) Masks
Masks for compact and mid configuration cubes were produced by automasking. Occasionally these were modified interactively during cleaning. The mid masks were used as a starting point for extended configuration cubes but may have been modified.
The masking process for combined cubes was complicated (see `W_Aql_CubesComb.py`) and requires interaction, so the final masks are not provided. Users are advised to tailor imaging parameters and masking for particular lines of interest as these differ depending on the line structure.

3) Continuum
In all cases the continuum was found to be confined within a few synthesised beams and pbcor cubes were not made.

4) a. Data processing: Quick start overview

See the README or Sec. 3 for naming conventions. The LP products for each star contain a set of scripts which are specific for that star and each configuration. These are different from the product names used during ATOMIUM data processing and analysis, notably by the group ID (GOUS) prefix and standardisation of continuum image names. For reprocessing using our scripts, un'tar the data and then remove the GOUS prefix from the MS (or edit the scripts accordingly).

(i) Ready-made images and cubes

The continuum images and cubes (all `fits` format) in the LP products are made using our standard scripts. In addition, they have been modified where necessary so all are in ascending frequency order. These are in general science-ready with some caveats. The standard EXTENDED configuration cubes do not cover the whole field of view (although most extended emission is resolved out). The standard COMBINED cubes also do not cover the total extent of species such as CO v=0 and, due to the large range of scales sampled in the data, it is often possible to improve greatly the images of a specific line by weighting optimised for its specific scales of emission and surface brightness.

(ii): Using LP products for re-imaging

For imaging only, download the MS and scripts for the relevant star and configuration: `*cont.ms` or `*line.ms.contsub` for continuum or line, respectively. See Secs. 5 and 8., scripts `SelfCal.py` (or `CubesComb.py` for COMBINED) for guidance in parameters for `tclean`. By default, cubes are in native frequency order, i.e. descending for the lower sideband. Note that for the individual configurations *cont.ms contains lines and a line-free selection must be made (as in the scripts) for continuum imaging. In `*combined*cont.ms` the line channels have been flagged.

The final masks we used are supplied for separate configurations. These were based on auto-masking for COMPACT and MID. If you do use the existing masks, import them into CASA format and edit script parameters in `tclean` accordingly.

The MID masks were used for EXTENDED but in some cases with interactive modification; see Sec. 4) d. Imaging COMBINED requires several stages depending on the line structure, see Sec. 4) e. In either case, you can use the LP product MID masks, but retain the script steps which map these to the current spw order. If you have re-processed MID and made your own masks, the script will find them if you are using the assumed directory structure. *However, for imaging individual lines in EXTENDED or COMBINED, it is probably best to make your own, tailored masks*.

(iii) Resubtracting continuum

We identified the line-free channels from the standard archive cubes and inspection of the visibility data. To re-do this, get `*.line.ms` (and `*line.ms.flagversions` is provided in case you want to restore any data flagged by previous `applycal`) and scripts and consult `SelfCal.py` if you want to use the initial continuum selection and `uvcontsub` parameters as a starting template. The COMBINED `*line.ms.contsub` is a concatenation of the separate configuration contsub's, so redo the individual subtractions first and then follow *ImagComb.py etc. to re-combine, see Sec. 4) e.

(iv) Redoing or adding to self-calibration

Our scripts assumed a standard directory structure like

/raid/ATOMIUM/<star>/<CONFIG>/Imaging_<config>/

for each star and separate configuration, or `/raid/ATOMIUM/<star>/COMBINED/` and file names without the group* prefix of the LP products. You can either adopt these conventions, or use our script `SelfCal.py` as a template to fit your structure. Download `*cont.ms` (and `*line.ms` and `*.flagversions` unless you only want low spectral resolution/continuum) and after you have finished calibration, apply the solutions to `*line.ms` and subtract the continuum. Sec. 4) b. describes this step by step.

(v) Redoing everything

If you want to modify the original pipeline/QA2 calibration or flagging or otherwise start from the archive data (prior to our reprocessing), to follow the structure assumed in our scripts, download the archive data (not the LP products) and process as wanted. See Sec. 4) b. for details and assumptions. We used a modified version of Lumberjack (github.com/adam-avison/LumberJack) to identify line-free channels in the archive cubes (in a few cases where these had not been made for all spw we made 512x512 cubes). It uses a 3" aperture to extract lines for MID and COMPACT. For EXTENDED, a 0".1 aperture is used as most lines detected are brightest, or seen in absorption, very close to the star. Please ask the Atomium contact if you want the scripts used. We suggest that instead, you check/modify the selection already copied into `SelfCal.py` . See Sec. 4) b. for a step by step description of self-calibration.

4) b. Data processing: Self-calibration details

Here is an annotated list of steps in `SelfCal.py.` This covers all steps for Compact and Mid configurations; the script is also used for Extended but see Sec. 4) d. for additional issues. Separate scripts, described in Sec. 4) e., are used for Combined.

Read through the script preamble and steps and note what and how you need to proceed depending on what you want to do. The scripts assume a standard directory structure (Sec. 4) a. (iv)) which you may need to modify. In the standard directory structure the archive directory starting `2018.1.00659.L/` should be in e.g. `/raid/ATOMIUM/<star>/<CONFIG>/`, and the unmodified LP scripts will then find the data and can be run producing products as per our original conventions.

`SelfCal.py` uses Analysis Utilities[1] but you could replace its use by setting values such as noise threshold manually.

0: 'split out with averaging for continuum self-cal and concatenate',

- Uses the pipeline-calibrated ms for each tuning
- Averages target data to TDM resolution for speed in continuum imaging
- Produces <star>_<config>_cont.ms and continuum listobs

1: 'split out at uniform high resolution and concatenate',

- Similarly for line data, averaging only the 1/2-width channels and correcting to constant LSRK
- Produces <star>_<config>_line.ms and line listobs

Steps 0. and 1. are only needed if you have modified the original archive products.

1 https://casaguides.nrao.edu/index.php/Analysis_Utilities

2: 'plotms continuum to identify line-free channels',

- Uses line-free continuum averaged channels output by
- Zoom in to check possible residual lines
- If necessary, edit `contchanscont` selection in `SelfCal.py` and re-run this step to check
- The data have been well-edited but if you see a few isolated, very discrepant visibilities, plot amp and/or phase v. time or uv distance with less averaging to identify the cause and if necessary flag. If it is a systematic mis-scaling or not pure noise, this may however be corrigible during self-calibration.

3: 'first image of continuum',

- Interactive, don't over-clean
- Gives most accurate astrometric position and model for self-cal
- Check beam size. If the cell size (variables `cell` and `cellline` near the start of the script, for each config) is such that there are less than 3-4 pixels across the minor axis, make the cell size smaller and/or change Robust from the default 0.5 to 1 or 1.5.
- Note that in each continuum clean cycle for separate configurations, the line-free channel selection must be entered.

4: 'statistics from image and self-cal solint estimates',

- Compare actual and predicted rms
- MID and COMPACT have scan length 5-6 min.
    - If solmin is much more than this, self-calibration is unlikely to be possible, but:
        - In some cases the actual rms is much better than predicted,
        - in which case a solint shorter than solmin can be used
        - Conversely, if the actual rms is higher, try to find out why (from the web log) - possibly e.g. a weak phase-ref, self-cal may improve S/N enough to use a shorter solint in a later step.

5: 'first phase-only self-cal',

- Check that not more than a few solutions fail at every step, or change solint/minsnr

6: 'apply and re-image',

- S/N should improve

7: 'next phase-only self-cal',

- Only if either or both of the model has improved (e.g. more extended emission), or you can reduce the solint.

8: 'apply and re-image',

- 2 Taylor series terms are used to prepare a model including spectral index in case amp self-cal is possible
- clean as deeply as possible to get all the flux for possible amp self-cal
- select the correct best continuum image
- edit step 11 to apply the intended tables only

9: 'amp self-cal applying phase solutions',

- Mask carefully. S/N should improve. You may not have enough S/N to do all the self-calibration steps, but amplitude self-calibration should be performed unless it makes the continuum image much worse, as it is important to align the flux scale across spw observed at different epochs:
- By default this step averages polarizations and normalises the gains, i.e. the flux scale will not be changed, only time-variable amplitude corrections.
- Check solutions; they should be within a few tenths of unity.

Steps 3. to 9. are only needed if you want to redo the self-calibration.

10: 'apply and re-image',

- S/N should improve or at least not deteriorate and sidelobes should decrease.
- It is usually better not to re-weight or flag by gain as failed solutions are usually due to low S/N, not bad data, and the phase-referencing corrections are adequate, i.e. `calwt=False, applymode='calonly'`.

Step 10. can be used as a template for re-imaging continuum

11: 'apply calibration to line data',

- Apply tables which you used for continuum with other parameters as in **Step 10**.
- If you run this step with `applymode='calflag'` and then go back to earlier steps or notice problems later, look for the `flagmanager` command in **step 11** and re-run it manually, `mode = 'restore'`. Beware, sometimes most of an spw or antenna can be flagged for weak stars, which appears to improve the continuum but the affected spw noise and/or resolution is badly degraded.

Do not run step 11. unless you have re-done self-calibration

12: 'plotms line data to id continuum',

- Uses line-free line channels identified as `contchansline`
- If necessary, edit selection in `SelfCal.py` and re-run to check
- See Step 2 notes on possibly flagging data (very rarely necessary)

Step 12 is only needed if you have re-done self-calibration or if you have downloaded the line.ms and want to check the line-free selection.

13: 'subtract continuum',

- Uses selection in `contchansline` for subtraction
- Check output .contsub MS in plotms

14: 'Make cubes one spw at a time, (change loop control to do just one)',

- Check loop control; we strongly suggest that you set the variable near the start of the script `interactmode=True` and initially clean just one cube interactively, to investigate whether automask parameters are optimised (or for Extended, that the Mid mask has been applied correctly). In any spw you might want to change the threshold (linethresh variable) and change masks interactively if needed.

Step 14 can be used as a template for customised imaging of (part of) a cube. See listobs or the table at the top of the script to identify which spw cover the frequencies of interest.

15: 'Export images to FITS format (see step to set destination variable) '

- `rename.py` will be run if `rename=True` to make target names consistent in all cubes (only needed for Compact, pi1_Gru, T_Mic, VX_Sgr). 'destination' can be blank/omit the subsection.

The original per-configuration scripts produce image cubes have names containing spw0, spw1 etc. but these are not in frequency order. See Sec. 5) for renaming to cubes in frequency order (this has already been done for LP products).

4) c. Data processing: Changing calibration cycles

If the star is fainter or brighter than the default assumption (~10 mJy), or conditions were different, then you might need less or more self-calibration. These guidelines were applied during the original calibration for each star.

I Fainter / lower S/N

e.g. after first image and imstat **steps 3, 4**:

```
Predicted continuum rms  0.060 mJy; achieved 0.055 mJy
Continuum peak  6.188 mJy; S/N   113
Minimum solint for S/N 3 per antenna, per spw, per polarization 103.4 sec
```

after phase-only self-cal, solint 'inf' making table *.p0, imaging and imstat **steps 5, 6**:

```
Predicted continuum rms p0 0.060 mJy; achieved 0.053 mJy
Continuum peak p0 6.358 mJy; S/N   120
```

A modest improvement

NB 150 s is close to 1/2 scan in this example

after phase-only self-cal, solint '150s', table *.p1, imaging and imstat **steps 7, 8**:

```
Predicted continuum rms p0 0.060 mJy; achieved 0.063 mJy
Continuum peak p0 6.970 mJy; S/N   110
```

Slightly worse S/N, and the increase in peak is due to the different spectral weighting (we have fitted for a spectral index).

Delete table *p1, use `gaintable=<table>.p0` in `gaincal` in **step 7** with `gaintype='T'`. This will use *.p0 to correct offsets between X and Y so they can be averaged to improve S/N in the shorter solution interval. In **step 8**, add table *.p0 to the list of gain tables in `applycal`. This gives

```
Predicted continuum rms p0 0.060 mJy; achieved 0.060 mJy
Continuum peak p0 6.970 mJy; S/N   116
```

In **steps 9 and 10**, add table *.p0 to the list of gain tables in `gaincal` and `applycal`. after amp self-cal, solint 'inf', tables *.p1:

```
Predicted continuum rms ap1 0.060 mJy; achieved 0.06 mJy
Continuum peak ap1 7.020 mJy; S/N   117
```

S/N slightly improved

VERDICT:

This suggests that the conditions were good enough that the phases were already quite stable, and just the first phase self-cal was needed to correct for directional differences between the phase-ref and target. The S/N is marginal for amp self-cal but it is important to try to achieve this as there can be ~7% amplitude scale uncertainties between observations of the different tunings.

Test for amplitude consistency:

```
plotms(vis=target+'_'+config+'_cont.ms', xaxis='freq, yaxis='amp',
     spw=contchanscont,avgtime='999999', avgscan=True,
     avgbaseline=True, coloraxis='observation')
```

Whenever possible, use amp self cal to align the flux scale across spw, especially if there are visible jumps, greater than the noise, not consistent with a thermal spectrum. Try more time averaging if necessary, reduce `minsnr` and check the model, the faintest stars are probably unresolved. It is OK as long as the S/N fall and peak change are not more than ~10% as long as the resulting image does not have conspicously worse (usually symmetric) artefacts and all spw have enough good solutions to correct ~all data when interpolated.

If many solutions fail and you ran step 11 with `applycal applymode='calflag'`, but you then achieve fewer failed solutions, look for the `flagmanager` command in step 11 and re-run it manually, `mode='restore'`, to replace any data flagged by failed solutions, then re-apply the improved tables.

If you want to apply more or fewer tables, e.g. only table `.p0`, for continuum, you can do this by running the `applycal` from **step 6** and no more.

For line, edit **step 11**:

```
 applycal(vis=target+'_'+config+'_line.ms',
           gaintable=[target+'_'+config+'_cont.p0'], # Just the good table
           calwt=False,          # at your discretion
           applymode='calonly',
           flagbackup=False)
```

Finally, put a comment in the script and comment out the unnecessary steps:

```
step_title = {0: 'split target with averaging for continuum self-cal and
   concatenate',
              1: 'split target at uniform high resolution and concatenate',
              2: 'plotms continuum to identify line-free channels ** see below',
              3: 'first image of continuum',
              4: 'statistics from image and self-cal solint estimation',
              5: 'first phase-only self-cal',
              6: 'apply and re-image',
              #7: 'next phase-only self-cal',
              #8: 'apply and re-image',
              #9: 'amp self-cal applying phase solutions',
              #10: 'apply and re-image',
              11: 'apply calibration to line data',
              12: 'plotms line data to id continuum',
              13: 'subtract continuum',
              14: 'Make cubes one spw at a time (change loop control to do one/all)',
              15: 'Export images to FITS format'}
```

II Brighter/better S/N

e.g. for a brighter star, in step 9, a `solinta1 ='inf'` was used and after step 10 e.g.:

```
W_Aql_compact_cont.cleanap1.image.tt0
Predicted continuum rms p1 0.050 mJy; achieved 0.052 mJy
Continuum peak p1 64.485 mJy; S/N  1249
```

S/N of more than a few hundred means that you could try additional self-calibration with a shorter solution interval. Step 4 reported

```
Minimum solint for S/N 3 per antenna, per spw, per polarization 2.9 sec
```

but the solint cannot be less than the integration time of 6.06s and plotting the visibilities suggests that might just sample noise, so we tried 18.2s, as below:

Near the top of the script after `solintp1`, insert

solintp2 = '18.2s'

and add new steps after **step 10**:

```
mystep = 107
if(mystep in thesteps):
  casalog.post('Step '+str(mystep)+' '+step_title[mystep],'INFO')
  print 'Step ', mystep, step_title[mystep]
```

```
  ft(vis=target+'_'+config+'_cont.ms',
    nterms=2,
    model=[target+'_'+config+'_cont.cleanap1.model.tt0',
           target+'_'+config+'_cont.cleanap1.model.tt1'], # Use latest model
    usescratch=True)

  os.system('rm -rf '+target+'_'+config+'_cont.p0')
  gaincal(vis=target+'_'+config+'_cont.ms',
          caltable=target+'_'+config+'_cont.p2', # New caltable
          gaintable=[target+'_'+config+'_cont.p1',
                    target+'_'+config+'_cont.a1'],# apply cumulative calibration
          spw=contchanscont,
          solnorm=True,
          solint=solintp2,        # New solint
          refant=antrefs,
          calmode='p')

# applycal and re-image
mystep = 108
if(mystep in thesteps):
  casalog.post('Step '+str(mystep)+' '+step_title[mystep],'INFO')
  print 'Step ', mystep, step_title[mystep]
  applycal(vis=target+'_'+config+'_cont.ms',
           gaintable=[target+'_'+config+'_cont.p1',  # All relevant tables
           target+'_'+config+'_cont.a1', target+'_'+config+'_cont.p2',],
           applymode='calonly',  # comment this out if final applycal
           calwt=False,          # could apply weights if multiple EBs
           flagbackup=False)

  os.system('rm -rf '+target+'_'+config+'_cont.cleanap2*') # update image name
  tclean(vis=target+'_'+config+'_cont.ms',
       imagename=target+'_'+config+'_cont.cleanap2',       # update image name
       spw=contchanscont,
       imsize=imsz_cont,
       deconvolver='mtmfs',
       nterms=2,
       cell=cell,
       weighting = 'briggs',
       robust=0.5,
       interactive=True,
       threshold=thresh,
       niter=500)                   # clean deeper
```

Also in this step, don't forget to change the input image in imstat etc. to

```
target+'_'+config+'_cont.cleanap2.image.tt0',
```

Add the new steps to the list of steps:

```
step_title = {0: 'split target with averaging for continuum self-cal and
    concatenate',
              1: 'split target at uniform high resolution and concatenate',
              2: 'plotms continuum to identify line-free channels ** see below',
              3: 'first image of continuum',
              4: 'statistics from image and self-cal solint estimation',
              5: 'first phase-only self-cal',
              6: 'apply and re-image',
              7: 'next phase-only self-cal',
              8: 'apply and re-image',
              9: 'amp self-cal applying phase solutions',
              10: 'apply and re-image',
              107: 'p self-cal shorter solint applying amp & phase solutions',
              108: 'apply and re-image',
              11: 'apply calibration to line data',
              12: 'plotms line data to id continuum',
              13: 'subtract continuum',
              14: 'Make cubes one spw at a time (change loop control to do one/all)',
              15: 'Export images to FITS format'}
```

After running **steps 107, 108** the script reports e.g.:

```
W_Aql_compact_cont.cleanap2.image.tt0
Predicted continuum rms p1 0.050 mJy; achieved 0.050 mJy
Continuum peak p1 65.267 mJy; S/N  1309
```

VERDICT:

<u>This shows a small but significant improvement, and as well as applying this table you could try a further round of amp self-cal with a solint or perhaps 30 or 60 s</u>

To do this, create a further pair of steps modelled on 9 and 10, and insert these and a new `solintap2` at the start. Take care to include the latest ap2 image model in `ft` and to apply tables `p1,a1,p2` in gaincal; apply all these plus the new table in the final applycal.

Edit **step 11**: (this assumes both new tables are worthwhile)

```
applycal(vis=target+'_'+config+'_line.ms',
           gaintable=[target+'_'+config+'_cont.p1',
                      target+'_'+config+'_cont.a1',
                      target+'_'+config+'_cont.p2'
                      target+'_'+config+'_cont.a2'], # add new table(s)
           applymode='calonly', #
           calwt=False,          # at your discretion
           flagbackup=False)
```

4) d. Data processing: Extended configuration

`SelfCal.py` recognises `config='extended'` and activates the required options; in a few cases these are hardwired into the scripts supplied with the extended configuration.

- Each input integration is 2.02 s (for initial calibration). Averaging to 6.05 s (the same as the other configurations) causes <5% smearing and allows a dynamic range >1000 at the primary beam FWHM radius for 16 km baselines. You will need to repeat all the steps if you want unaveraged data (but the line-free channel selections will not be affected).
- Each scan is under a minute, so solint 'inf' is about 50 s, there is no need to use a shorter solint unless the S/N is good enough to go to 25 s or shorter. If necessary iterate phase cal, solint 'inf', to improve the model, and try amp cal.
- When observations were prepared only *Hipparcos* proper motions were available and some stars are 100 mas or more offset from the pointing centre. After the first continuum image, the peak position is found and has been set as variable `peakpos`, used to define the phasecenter in imaging thereafter.
- The default robust 0.5 is probably best for continuum self-cal, using a 3-mas cell, but for optimised sensitivity and a larger image in angular size, and consistency, a higher robust and possibly a slight taper can be used in the final continuum image and for line cubes, to give ~25-mas beam, using a 5-mas cell.
  - See new variables at the start of the script to control these.
  - Optional step 20 is added to make the final continuum image, see script. You may need to experiment with the variables `R` for robust and `taper` for uvtaper, to get a beam such that sqrt(Bmaj*Bmin) ~ 25 mas. This will change from lowest to highest frequency.
- The default 200 pixel continuum image is mostly OK but check for extended dust. Much extended line emission will be resolved out, especially CO. You can look at the pipeline images to see if there is any extended emission. Currently the default for cubes is 600 pixels.
- The script uses the 'mid' configuration masks in imaging, as automasking at high resolution is slow and very hard to 'tune'.
  - This requires the Mid masks to have been created, or the LP products downloaded into the correct path structure. The script attempts to 'translate' but please check (e.g. look at frequencies in `imhead` or start `tclean` interactively).
  - Check and if necessary adapt masks by hand for very bright/compact emission, notably SiO masers.
- For W Aql, R Aql and GY Aql, it was realised later that the EXTENDED flux scale was >10% overestimated. The final continuum image and cubes were rescaled to make the continuum consistent with the Mid and Compact peaks and the visibility data were rescaled before combination (so they can be re-imaged without further adjustment). This is included in the relevant `SelfCal.py`. If you have reprocessed the data, check whether this is still necessary.

4) e. Data processing: Combining Configurations

If you only want to make images from the existing combined MS, to follow `CubesComb.py` you need Mid configuration masks for cubes but you could make your own mask. In order to re-do combination, the script `ImagComb.py` assumes that you are working in a directory like `/raid1/scratch/DecinATOMIUM/<star>/COMBINED/` with the individual configuration data in a structure as in Secs. 4(iv) and 5. Edit the paths as needed. Calibrated MS and listobs (continuum and contsub) must be present for all configurations.

Scripts `ImagComb.py, CubesComb.py, catalign.py`.

Start with continuum to test matching up and weighting:

- **Position** `ImagComb.py` steps 0~5, 7~8. The proper motions (and possibly astrometric errors) are, for at least some stars, high enough to cause a significant offset. If the a-priori proper motions are accurate then `concat` can allow for this but possibly the pointing centre positions will not match the proper motions exactly. This can be corrected by using `imstat` to find the peak positions, `fixvis` to centre each data set and `fixplanets` to shift mid and compact to the extended position. `fixplanets` only recognises J2000, which can be a few tens mas offset from ICRS, so the data have to be labelled in J2000 and then shifted back to ICRS. Please use the individual configuration data for astrometry.
- **Frequency alignment** In most cases spw observed at different epochs could be combined with less than 1 channel difference in velocity (due to the motion of the Earth with respect to the LSR). script `catalign.py` checks for differences and optionally extracts and regrids the overlapping channel ranges before combination.
- **Flux scale** `ImagComb.py` step 5. Compare continuum uv-amp on overlapping baseline lengths (after flagging line channels). In some cases the continuum peaks meaured per-configuration differ by tens%, Extended often being brighter. This could be due to:
    - noise errors especially in the Compact data (shortest observations, lowest S/N) depressing the measured peak flux;
    - phase errors on long baselines in the Extended data which affect the solutions derived from the phase reference and could result in over-estimating the corrections applied to the target;
    - variability (but few% or less expected for the star) or dust formation/cooling. We cannot do anything about variability, especially as the star and lines will be affected differently.
        - Some SiO and water masers are very variable between epochs and the combined data cannot be used for analysis.

  For some stars there was >10% discrepancy between configurations. In some cases this was mitigated by more rigorous self-calibration of mid or compact configurations but for W Aql, R Aql and SV Aql the extended flux was still 30-50% too high. This was addressed by rescaling the affected target visibilities (line and continuum) to the average of mid and compact flux densities, using `gencal`, before combination.

- **Weighting and continuum images** `ImagComb.py` steps 5, 6, 8. We decided to combine the data giving each configuration equal weight. Because there are more Extended data, this gives a resolution of a few tens of mas and thus tends to resolve out much emission and can produce blobby or 'basket-weaving' artefacts. Nonetheless, Extended also has much data on intermediate baselines so the best option seems to be to apply a slight uv taper in `tclean`, giving a synthesised beam of 40-70 mas. The continuum-subtracted line data are similarly combined.

- **Line Cubes**
    - Where it was necessary to extract the overlapping channel ranges before combination, the total velocity span of the spw affected is reduced.
    - `CubesComb.py` makes default images ~4” across, with a slight uv taper to give a resolution of 50-80 mas (chosen to avoid the artefacts mentioned above). In all the imaging steps, the script will force interactive cleaning for cubes containing bright masers (see notes at top of script), since it may be necessary to remove the mask from dynamic range channels after a few major cycles, to allow weaker emission to be above the clean threshold in subsequent cycles. Also, maser variability may lead to deep negative artefacts. **We do not recommend using combined cubes for maser analysis.**

  Parallelised cleaning is not used as it produces unreliable results for deep cleaning (different segments (channel ranges) are cleaned to different depths and it is not possible to stop and restart as the segment labelling gets scrambled).
    - Step 10 creates files containing the correspondance for the combined data spw with the final cubes in frequency order and with the spw number of the masks from Mid.
    - Step 11 images each cube with a small mask around the stellar position. This is to mitigate the effect of very bright masers on surrounding thermal emission. If there is absorption at the stellar position, check if Extended shows the same. The end channels may be trimmed to ensure the first/last channel is not a single configuration-only giving a the wrong beam - please check fitted beam sizes.
    - Step 12 regrids the Mid masks to the current image parameters. This is very fussy about exact matching of the reference freq. etc., hence setting/tweaking related parameters. The output frequencies should be correct but the apparent velocities may have arbitrary reference freqs.
    - Step 13 images each cube applying the Mid mask. It is worth checking that these are aligned correctly with the current emission, especially for masers, and removing masks if a dynamic-range-limited channel reaches the limit, i.e. the extrema in that channel seem to be artefacts. A list of cubes/channels most likely to need manual mask changes is given at the start of `CubesComb.py`.
    - Step 14 tries to catch large-scale emission by cleaning in a circle out to close to

the edges. Manual removal of the mask from dynamic-range-limited channels may be needed.

5) Extracting spectra

The script `getSpecAtom.py` is used for Compact, Mid and Extended configurations to extract spectra from the image cubes (numbered in ascending frequency order) in a set of apertures centred on the star. The script also re-names the original cubes (`*spw*`), which were in non-linear frequency order, to cubes (`*cube*`) numbered in ascending frequency order. It needs modification to extract spectra only from final cubes.

`getSpecCombined.py` extracts spectra (only) from combined cubes (which are already numbered in ascending frequency order).

Both scripts need modification to work on the LP product names.

The spectra we extracted are included in the LP products, modified where necessary so all are in ascending frequency order.

6) Quality control

In all cases the image rms noise reached or exceeded the specifications for the observing set-up and conditions.

The visibility data and images were inspected for amplitude consistency across the frequency and baseline length range, and additional calibration was applied if needed.

The extracted spectra were checked for excess, inexplicable noise and calibration or imaging improved if needed.

Imaging fidelity was compared between configurations. As noted in Sec. 4) e., we identified a need for customised imaging for some COMBINED lines.

The flux scale and astrometric accuracy are described in G21.

Acknowldgements:

We thank the ALMA staff for all their support from observations to data delivery.
Our data processing was carried out on data hosted at JBCA on IRIS[2] workstations, tracked using PBWorks.

2 We thank IRIS, http://iris.ac.uk, for provision of high-performance computing facilities.