

# I-TRAIN #6: Improving Image Fidelity Through Self-Calibration



E. Moravec - Czech ARC node  
A. M. S. Richards - UK ARC node  
A. Pérez-Sánchez - Allegro ARC node  
M.C. Toribio - Nordic ARC node



EUROPEAN ARC  
ALMA Regional Centre



**Astronomical  
Institute**

of the Czech Academy  
of Sciences



**CHALMERS**



# Q&A



## Q&A

The following slides are a compilation of Questions and Answers during the tutorial on self-calibration.


Some of the questions were discussed live and you can find them in the video available online around time ~51:01



## (1) Could you explain why you use `calwt=False`?


In some data sets some antennas might have worse errors, in this case the most distant antennas (or smaller antennas in VLBI, for example). So these will have noisier solutions, initially at least. If you downweight these, they will contribute less to an image and thus the model will not constrain so well the very antennas which most need correcting. So I tend only to use `calwt=True` at the end, for the final image. There are exceptions, of course - e.g. if one antenna is so noisy you cannot make a good initial image - or if noisier antennas have something wrong with them and do not contribute unique spacings.

You can find a follow-up on this question in the video ~51:01




(2) What is the minimum brightness of your source needed to do the most basic selfcal?

You can find this question discussed in the video ~52:13. This topic is also covered in the Advanced lecture.




(3) I understand that whatever the continuum channels are specified in `tclean spw`, the calibration is applied to all channels. Is it also applied to channels which might have been flagged in previous steps?

You can find this question discussed in the video ~56:18



(4) Follow-up on a couple of questions ago: is there a limit on the upper end as well, where self-calibration doesn't make sense if the source is too bright?


You can find this question discussed in the video ~59:17.



(5) Is it more robust to run the self-cal with antenna-based gain solutions or baseline-based solutions? Is it not dangerous to do it baseline-based, as to introduce substructures in the source?

Yes, you are right. Baseline-dependent solutions need a much higher  $S/N$  in the whole image (v. antenna based), by roughly a factor of the number of antennas. Also you are fitting a model to every detail of the image. So it is easier to distort your data to match incorrect details of the model. I would always start with antenna-based solutions. One situation where people do use baseline self-cal is a classic radio galaxy with an unresolved, very bright core and weak lobes which are too faint to be included in the model. Or a spectral channel with a very bright maser. So you can get an ideal point model which is very accurate and use that for baseline calibration. However, for modern interferometers, the causes of baseline errors (antenna positions, correlator/electronic faults) are very rare except for VLBI. If anyone has e.g. EHT experience they might know more...?





(6) I am surprised by your choice in the script's step 12 where the gaintable parameter has two arguments. Is it the same as sequential correction with the two gaintables?

Yes, that is exactly the idea. The two gain tables will be applied sequentially to the DATA column. Notice that in step 11 in gaincal we apply the table "ph1" on -the-fly to calculate table "ph2".

There are various ways that you can apply the solutions cumulatively. You can:

1. apply the gaintable for the current round of self-cal and then split off the corrected column and work with this MS for the next round but this creates a lot of data
2. apply the solutions 'on-the-fly' in which you apply the tables of solutions in a cumulative fashion by stacking them as we have in the tutorial
3. keep improving the model by iterative rounds of self-cal then applying the final table

These are all pretty much the same. The crucial thing is to apply the same gain tables in gaincal and applycal (plus the new one in applycal). If you decide to discard an earlier gaintable then omit it from both tasks.