ALMACAL project report

Table of Contents

[Collecting deep calibrators observations 1](#_Toc517947936)

[The longest integration time 1](#_Toc517947937)

[astroquery alma 2](#_Toc517947938)

[Browsing the ALMA archive for calibrators: 2](#_Toc517947939)

[DIRECT SEARCH IN ALMACAL 2](#_Toc517947940)

[HL TAU ALMACAL database 3](#_Toc517947941)

[cube formation code: 4](#_Toc517947942)

[J1924-2914 4](#_Toc517947943)

[CASA 5](#_Toc517947944)

[DUCHAMP 6](#_Toc517947945)

[Testing duchamp parameters 7](#_Toc517947946)

[CUBEFORM updates journal: 10](#_Toc517947947)

[ANGULAR SIZE ESTIMATIONS 11](#_Toc517947948)

[Marcels cube 12](#_Toc517947949)

[ALMACAL DEEP 12](#_Toc517947950)

[Matching the fake sources with detection + real detections 17](#_Toc517947951)

[Indexes for duchamp results file for fixed treshhold option 17](#_Toc517947952)

Collecting deep calibrators observations

## The longest integration time

Integration time describes how long (in total) the calibrator was observed. Singular exposure lasts several minutes but in the programs (especially large programs) observations of single calibrator can be repeated many times.

## astroquery alma

Astroquery.alma – interface to query alma archive   
<http://test-astroquery.readthedocs.io/en/latest/alma/alma.html>   
In the end the archive search proved not to be useful. The archive has issues, queries of the same parameters can differ, also the archive gives only one of the uids from the whole bunch of calibrations observations. What is missing are the other fiels with add up for the long integration time. You still have to search externally for the all other observations made with similar setup. We desided to use the existing archive for a start, and search through already reduced files in order to find similar uids [setups]. In this way we will be still missing some new observations, not yet incorporated in the ALMACAL archive. In the end I need to double check what files are missing, so we can reduce them.

## Browsing the ALMA archive for calibrators:

We need calibrators, ones with long integration time and also these for which at least one spw setup is similar. We are not interested in calibrators which are :

* Planets or other Solar System objects
* Have POLARIZATION setup
* Are POINTINGS
* Are TARGET or TARGET WVR or APPPHASE TARGET
* Have integration time smaller than 2500s
* Are form cycle zero

We are though allowed to use the new not published results, but not programs which are still ongoing.

Deep\_calib.py was quering the alma archive using the list of all ALMA calibrators. ­­Out of 7635 calibrators the program chosen 783 entries fulfilling the requirements. Among these entries we search for the ones fulfilling the frequency coverage criteria: no small spectral windows (< 0.5 GHz), and no repetition of the data.

29 of the entries had the “12m7m” configuration. These were not included in final sample – the number of entries down to 754 (calib\_deep.txt). If we ask for the ones which has more than one spw bigger than 0.5 GHz we get 683 entries. The list of the entries: deep\_calib\_2.txt

We need to compare the direct ALMACAL archive search with deep\_calib\_2.txt

## DIRECT SEARCH IN ALMACAL

To search through the archive I first created the list of all folders and files in them: almacal.list.   
Then I used the list of folders to create file list for each calibrator: almacal\_search.sh, each calibrator name would get the list of all files <calibrator>.txt. Then I run the script search\_ms.sh: it again reads the list of the calibrators,, creates the directory for each calibrator and then run get\_ms.py script in it

I created the list of the parameters of all files in the almacal archive. In the meantime I found several broken files, probably lost during copying. The full “catalog” of files for each calibrator is on hltau:in deep\_almacal folder <calib\_name>.list, calibrators.txt is a list of the calibrators names.

<calib\_name>.list file structure:

ms\_filename, project\_iuid, integration\_time[minutes],

Not working, asdm uid shows only one of the obserwations during whole program. Need to look for member ids which are the same, this way we get all data with the same setup

Member id – same setup

There is only one id in the ms fiels tables, and it is consisten with the member id

# HL TAU ALMACAL database

On hltau

For copying we need bash script

/science-ALMACAL/data/calibrator\_name

name od the ms file: uid\_\_\_<asdmuid>.ms.\*.<cal\_name>\_<band>

can be split.cal or anything else

Browsing files in casa.

Using casa, we browse throught ALMACAL archive, looking for long cubes:   
sum integratin time for all files with the same member id

Imaging – forming the deep cubes

testing

Before the mass-production of the deep cubes I run test on the cubeform\_single.py code which would be integrated in the genral script later. Under the directory below the test files can be found with cudeformsingle.py  
/scigarfs/opsw/work/ahamanow/deep/

Some ALMACAL files are repeated under slightly different name (no split.cal  extensiton or shorten name of calibrator)

# cube formation code:

Uvcontsub: The way to get rid of the residual continuum emission in the centre of the cubes, is to use uvcontsub before running clean. You can run it with the default values on the ms, but with fitorder=1, so that it makes  
a first order fit in frequency space. The output files will then have the postfix .contsub, which is then your input in clean.

Fitspw=the spws used for fitting, spw=applie fitting to spws. If yo choose individual spws these would be saved as output of contsub… The othewr would be neglected (so if fitting spw by spw you would loose the info about some).

Uvcontsb must work on of the spws on each of the measurement sets separately.  If we combine them to make a cube or make cubes of indivisual ms’s, doesn’t matter. We can then make smaller cubes, which would be easier to implement…But would be more of them. Within cubesingle.py will x What I managed to do?

exist the extra function, doing uvcontsub only on the spw’s we are interested in.

Basing on the literature search, we check what is the possible size of CO emission in the L\* glaixes. Typically observed were sizes ~10 kpc which gives about 1.6 – 1.2 “ for redshifts 0.5-2. We decided to apply the resolution of 0.5”.

Parameter uvtamper - This controls the radial weighting of visibilities in the uv-plane through the multiplication of the visibilities by the Fourier transform of an elliptical Gaussian  
in cleas uvtamper=True, sbparams: outertaper=[‘0.5arcsec’]  
Imsize defied by the FWHM of the primaty beam multiplied by 1.5 (au.primaryBeamArcsec(frequency=refFreq[0])), to cover less sensitive regions., Cellsize=1/3. Synthesized beam size (au.estimateSynthesizedBeam(names[0])/3.).

Before running cubesingle change the file directory in the \*.list file, create names.list

For cubesinge! Don’t’ divide cell by 3

Scigar problems with saving big files

 lfs setstripe -c -1 /lustre/opsw/work/ahamanow/ALMACAL/deep/J1058+0133/

lfs df -h

find 7m  
open casa and run ‘find\_7m.py’, remove from names.list uids no longer usable (u=you can chaeck intehgration times for ones partially removed)

cube formation  
open casa and run ‘cubesingle.py’

cube rejection reasons: 7m contaminations, interference patters, correlated noise

## J1924-2914

From the estimated 26 setups we ha to resign from several due to interference paters, dcorrelatied noise or the contamination from 7m antennas(!) . The latte ris most suprising since the ALMACAL database should be free of such data.

During the imaging process I found several datacubes which showed big ixels. Not knowing what causes such inconsistency (the cell,pix and beam were readed out by sophisticated functions), I tried to fine tune the parameters myself to get better result. In the end it appeard tha these particular member uidscointaned nbumerous ms firls from 7m a ntennas! So the beam estimation was wrong, because it depend on the antenna diameter. Closer investidaton showed that numerous ms files from 7m are present in ALMACAL datavase (although should cbe none), I had to verify if any other problematic cubes could be affected. For some memebrs uid the 7m and 12m observations were ixed, in one case the rescued 12m ms files counded to 2416s, a bit below the threshold but I moved it bask to be processed. (uid\_\_\_A001\_X5ac\_X7c4). Remainng cubes ans lists were deleted alongside the interference issued cubes.   
Assuming that all member ids are taken is same setup, I always used the firsgt ms on the list to get the arguments form celan etc. If the first on the list were 7m it affected whole imaging.  
The interference patters are not linked to 7m contamination. For some cubes the choice oflimiting spw separation was not efficient, gaps were to brad and I decidied to split them .

There are several further observations affected by some pattern background, we need ot investigate and some causing problems (like too big imsize).

Find\_7m.py is a code which would go through all l uid’s lists and check in any ms file is a 7m. Theiu it would print the 7m-ms files along with the uid . You can execute it in the calibrator foilder, the input is ile names.list including a list of uid lists. Preferably run it before cubeform [procedure: find 7m, do’t image them add later]

# CASA

Cut cubes – only on images

imsubimage(imagename="uid\_\_\_A001\_X10c\_X10c.cube\_01.J0423-0120\_B7.image",outfile="uid\_\_\_A001\_X10c\_X10c.cube\_01a.J0423-0120\_B7.image",box="",region="",chans="0~478"

Execfile(‘\*.py’)

Exportfits(‘\*.image’,’\*.fits’)

Rms for each frequency slice

stats=imstat('uid\_\_\_A001\_X1b1\_X74.cube\_03.J1924-2914\_B7.image',axes=[0,1])

and then stats[‘rms’] give rms per channel

Total elapsed time listed by listobs is just the time interval between first and last observatojn of the object (on certain setup). Itiis not an integration time and shouldnot be used for the statistics

# DUCHAMP

Usage:

Path to Duchamp in version 1.6.2 is:  
/lustre/opsw/software/Duchamp-1.6.2/bin/Duchamp-1.6.2  
  
root@scigarn1:/lustre/opsw/software/Duchamp-1.6.2/bin>./Duchamp-1.6.2  
Usage: Duchamp [OPTION] [FILE]  
Duchamp is an object finder for spectral-line FITS cubes.  
  
  -p FILE      Read in parameters from FILE, including FITS image location.  
  -f FILE      Use default parameters with imageFile=FILE  
  -t THRESH    Sets the detection threshold to THRESH, overriding that given by the parameter file.  
  -x           Do not use X-windows PGPLOT output  
               (equivalent to setting flagXOutput=false -- overrides the parameter file)  
  
  -v           Return version number and exit  
  -h           Display this help information and exit  
  
Before running Duchamp please run fallowing command:  
echo $LD\_LIBRARY\_PATH  
  
If you receive any output please use command:  
export LD\_LIBRARY\_PATH=/usr/local/lib:$LD\_LIBRARY\_PATH  
  
if not please run command:  
export LD\_LIBRARY\_PATH=/usr/local/lib

Run Duchamp (If you have a fits file)  
duchamp –f \*.fits [default parameters]  
Duchamp –p \*.pits \*.param [with your parameters]

Output of suchamp: list of the detectons, map of detections and moments, spectra of the detections

Export CASA image into fits file (requested by Duchamp)

Exportfits(imagename=, fitsimage=, )

flaggedChannels 0,1,2…12-20

plot\_rms.py – will read the input file ofg Duchamp ready images, calculate rms per channel, and choose outliers .output is a Duchamp.in input file to ducham script, with the list of flagged channels

Possible baseline removal

Smoothing the cube

Possible spectral and spatial smothing

Creating parameter file

Parameter names are not case-sensitive, and lines in the input file that start with # are ignored.

If a parameter is not listed, the default value is assumed

image file must be specified!

fits files for Duchamp are in each calibrator folder inside for\_duchamp

script\_duchamp.py – load the list of cubes (cubes.list) and perform search on it

running Duchamp on J1924-2914, not all cubes yet but most. Some are empty, oter shave few deftections and then 1000…

## Testing duchamp parameters

Cube X1b1\_X74\_cube\_03 populated with 20 detections. Testing the grid of parameters. Two channels flagged (0,13)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| X1b1\_X74\_cube\_03, 60 channels | | | | | | | | |  |
| RUN ID | Number of detections | Recovered objects | SN\_cut | Search Type | | Wavelet/ smooth | Snr/ smooth\_param | baseline | Multiple matches (detected more than once) |
| R1 | 904 | 3 | 1 | Spectral | | Wavelet | 3 | no | 0 |
| R2 | 102 | 17 | 2 | Spectral | | Wavelet | 3 | no | 2 |
| R3 | 13 | 12 | 3 | Spectral | | Wavelet | 3 | no | 1 |
| R4 | 904 | 3 | 1 | Spatial | | Wavelet | 3 | no | 0 |
| R5 | 102 | 17 | 2 | Spatial | | Wavelet | 3 | no | 2 |
| R6 | 13 | 12 | 3 | Spatial | | Wavelet | 3 | no | 1 |
| R7 | 908 | 6 | 1 | Spatial | | Wavelet | 3 | Yes | 0 |
| R8 | 8 | 0 | 2 | Spatial | | Wavelet | 3 | Yes | 0 |
| R9 | 0 | 0 | 3 | Spatial | | Wavelet | 3 | Yes | 0 |
| R10 | 908 | 6 | 1 | Spectral | | Wavelet | 3 | Yes | 0 |
| R11 | 8 | 0 | 2 | Spectral | | Wavelet | 3 | Yes | 0 |
| R12 | 0 | 0 | 3 | Spectral | | Wavelet | 3 | Yes | 0 |
| R13 | 9 | 0 | 1 | Spectral | | Smooth | 5 | Yes | 0 |
| R14 | 2128 | 14 | 2 | Spectral | | Smooth | 5 | Yes | 6 |
| R15 | 77 | 2 | 3 | Spectral | | Smooth | 5 | Yes | 0 |
| R16 | 27 | 0 | 1 | Spectral | | Smooth | 5 | No | 0 |
| R17 | 2118 | 17 | 2 | Spectral | | Smooth | 5 | No | 2 |
| R18 | 187 | 18 | 3 | Spectral | | Smooth | 5 | No | 1 |
| R19 | 9 | 0 | 1 | Spatial | | Smooth | 5 | Yes | 0 |
| R20 | 2128 | 14 | 2 | Spatial | | Smooth | 5 | Yes | 6 |
| R21 | 77 | 2 | 3 | Spatial | | Smooth | 5 | Yes | 0 |
| R22 | 27 | 0 | 1 | Spatial | | Smooth | 5 | No | 0 |
| R23 | 2128 | 17 | 2 | Spatial | | Smooth | 5 | No | 2 |
| R24 | 187 | 18 | 3 | Spatial | | Smooth | 5 | No | 1 |
| R25 | 3 | 0 | 1 | Spectral | | smooth | 3 | No | 0 |
| R26 | 1441 | 17 | 2 | Spectral | | Smooth | 3 | No | 1 |
| R27 | 59 | 17 | 3 | Spectral | | Smooth | 3 | No | 0 |
| R28 | 121 | 0 | 1 | Spectral | | Smooth | 7 | No | 0 |
| R29 | 2517 | 19 | 2 | Spectral | | Smooth | 7 | No | 3 |
| R30 | 301 | 18 | 3 | Spectral | | Smooth | 7 | No | 3 |
| Anne | 15 | 14 | 2.5 | Spatial | | Wavelet | 2.5 | No | 0 |
| Anne2 | 15 | 14 | 2.5 | spectral | | Wavelet | 2.5 | No | 0 |
| uid\_\_\_A001\_X1b1\_Xd.cube\_23.J1924-2914\_B7 250 channels | | | | | | | | |  |
| RUN ID | Number of detections | Recovered objects | SN\_cut | Search Type | Wavelet/ smooth | | Snr/ smooth\_param | baseline | Multiple matches (detected more than once) |
| R1 | 176 | 0 | 1 | Spectral | Wavelet | | 3 | no | 0 |
| R2 | 1313 | 18 | 2 | Spectral | Wavelet | | 3 | no | 1 |
| R3 | 117 | 17 | 3 | Spectral | Wavelet | | 3 | no | 1 |
| R4 | 176 | 0 | 1 | Spatial | Wavelet | | 3 | no | 0 |
| R5 | 1313 | 18 | 2 | Spatial | Wavelet | | 3 | no | 1 |
| R6 | 117 | 17 | 3 | Spatial | Wavelet | | 3 | no | 1 |
| R16 | 86 | 0 | 1 | Spectral | Smooth | | 5 | No | 0 |
| R17 | 2821 | 17 | 2 | Spectral | Smooth | | 5 | No | 1 |
| R18 | 355 | 18 | 3 | Spectral | Smooth | | 5 | No | 3 |
| R22 | 86 | 0 | 1 | Spatial | Smooth | | 5 | No | 0 |
| R23 | 2821 | 17 | 2 | Spatial | Smooth | | 5 | No | 1 |
| R24 | 355 | 18 | 3 | Spatial | Smooth | | 5 | No | 3 |

# CUBEFORM updates journal:

What cubesingle does?

20 XI ’17

cubesingle: decide which spw’s are important, then performs uvcontsub onthese selected ones. Important spws: not epeating frequency coverage, not to narrow ones, pairing ones laying close enough  
Still needs: pixelsize an dimsize better suited for the sourcefind

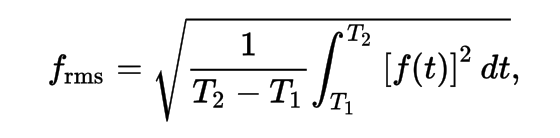
Pixel size and image size

27 XI ‘17

cell = beam/3.  
imsize <- fwhm\_beam \*1.5  
applying uv\_taper =True, outertaper=0.5”

Measuring rmsof the cube

Rms – root mean square



measure rms after the cube formation  
rms = imstat (’\*.image’)[‘rms’][0]

clean crashed due to large number of opened files provided in the list: introducing “concat” as the solution. For each spw selection the new concat\_spw file is created which is the cleaned and is forming a cube. The change is incorporated into the code.

Note: tclean provided the same error, due to large number of opened files

au.pickCellSize(vis='', spw='', npix=5, intent='OBSERVE\_TARGET#ON\_SOURCE', imsize=False,

maxBaselinePercentile=95, cellstring=False, roundcell=2, compare=False, pblevel=0.2,

config='', frequency=0, verbose=True)

Inputs:

Volume estimations

Usint Lambda CDM in python to get angular sizes, distances etc: astropy.cosmology

# ANGULAR SIZE ESTIMATIONS

For cube formaton 0.5 “ resolution was established - uvtaper parameter for clean

## Marcels cube

Run with regular Duchamp params fiel, the dectection was found only because of its narrow peak. Onbly the peak was detected. If the detection had ust boxy-shape not aboeve 4 simga would be oitted

# ALMACAL DEEP

77 calibrator fields will be used for further investigation

J0522-3627

J2056-4714

J1427-4206

J0334-4008

J1924-2914

J1058+0133

J0510+1800

J1256-0547

J0006-0623

J1229+0203

J0238+1636

J0342-3007

J0519-4546

J1000+0005

J1225+1253

J2148+0657

J0423-0120

J1337-1257

J1751+0939

J2258-2758

J0348-2749

J0854+2006

J1550+0527

J1744-3116

J2357-5311

J1733-1304

J1517-2422

J2253+1608

J1733-3722

J1107-4449

J1617-5848

J1342-2900

J0948+0022

J0607-0834

J0538-4405

J0237+2848

J0038-2459

J0040-3243

J0045-3705

J0102-7546

J0108+0135

J0132-1654

J0137-2430

J0152+2207

J0217+0144

J0217-0820

J0241-0815

J0253-5441

J0426+2327

J0440-6952

J0532+0732

J0541-0211

J0542-0913

J0552+0313

J0601-7036

J0635-7516

J0750+1231

J0909+0121

J1010-0200

J1036-3744

J1042-4143

J1044+0655

J1047-6217

J1103-3251

J1150+2417

J1245-1616

J1337-2951

J1650-5044

J1651+0129

J1717-3342

J1743-0350

J1832-2039

J1911-2006

J1924+1540

J1957-3845

J2025-2845

J2232+1143

J1924-2914

29 cubes ready for Duchamp

6.06.18

Cubes divided I to good and bad quality. Bad quality with thehunders/thousands of channels (why?), big jumps between spws etc. I will try to work on them later.

Matching and new detections procedure –

Current cubes/band count

band3 42 -hist

band4 10 - scat

band6 44 - hist

band7 48 - hist

band8 10 - scat

band9 5 - scat

35 fields, 137 cubes [rms only for the cubes that were formed of course and for onels with rms <0.5]

I found this way problematic cubes (voila) , with RMS > 1 . I remove them from the histograms.

8 cubes, 2 fields

**!! J1733-1304.rms 4.17869**

**!! J1733-1304.rms 7.8483**

**!! J1733-1304.rms 3.87343**

**!! J2148+0657.rms 7.1715**

**!! J2148+0657.rms 2.95048**

**!! J2148+0657.rms 1.26722**

**Two more excluded >0.5**

**!! J1733-1304.rms 0.937235**

**!! J1733-1304.rms 0.696348**

List of the cubes rms > 0.5

**uid\_\_\_A001\_X10f\_X459.cube\_01.J1733-1304\_B6.image | QSO**

**uid\_\_\_A001\_X10f\_X459.cube\_23.J1733-1304\_B6.image | QSO**

**uid\_\_\_A001\_Xa0\_X132a.cube\_23.J1733-1304\_B7.image | QSO**

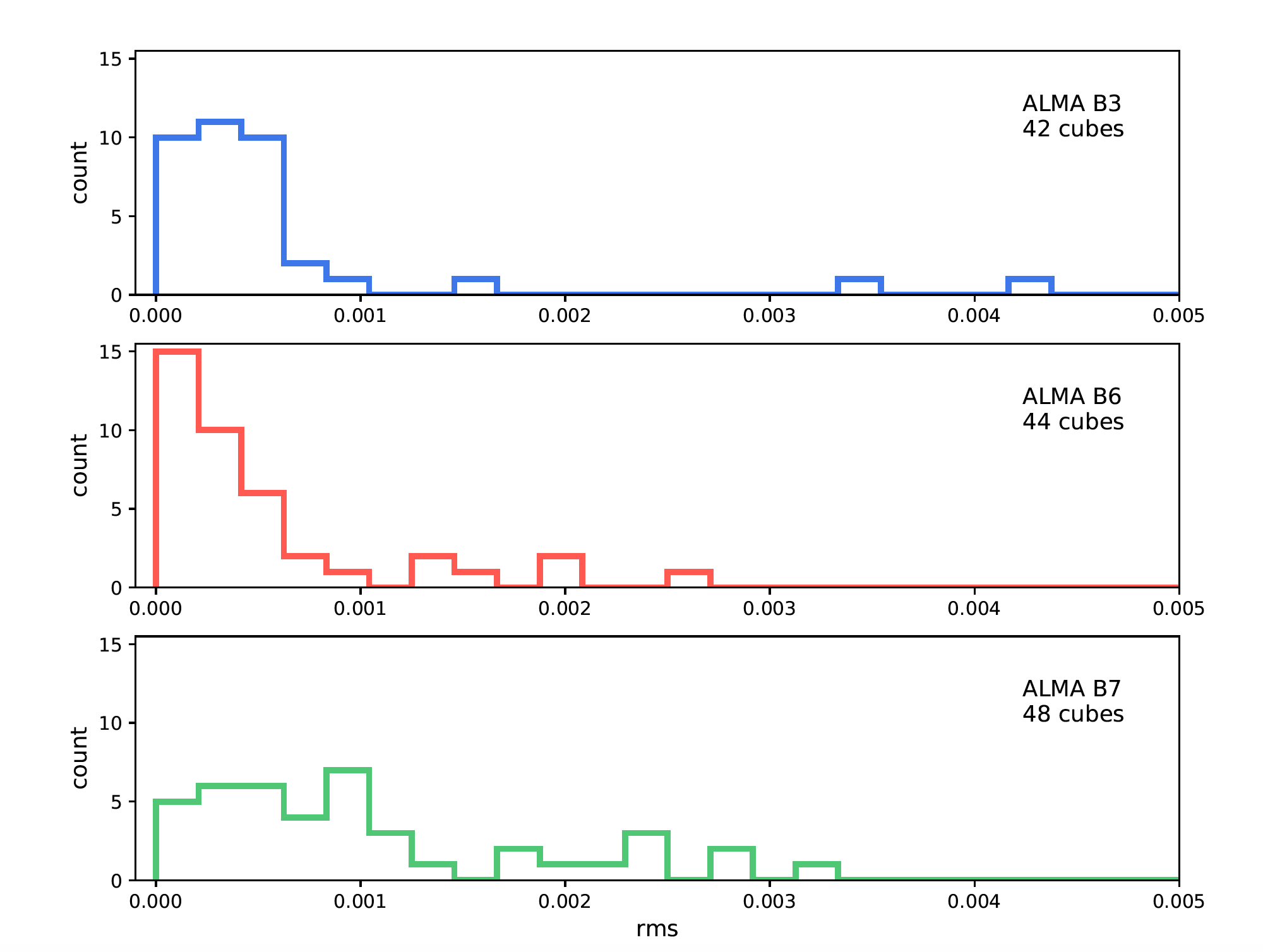
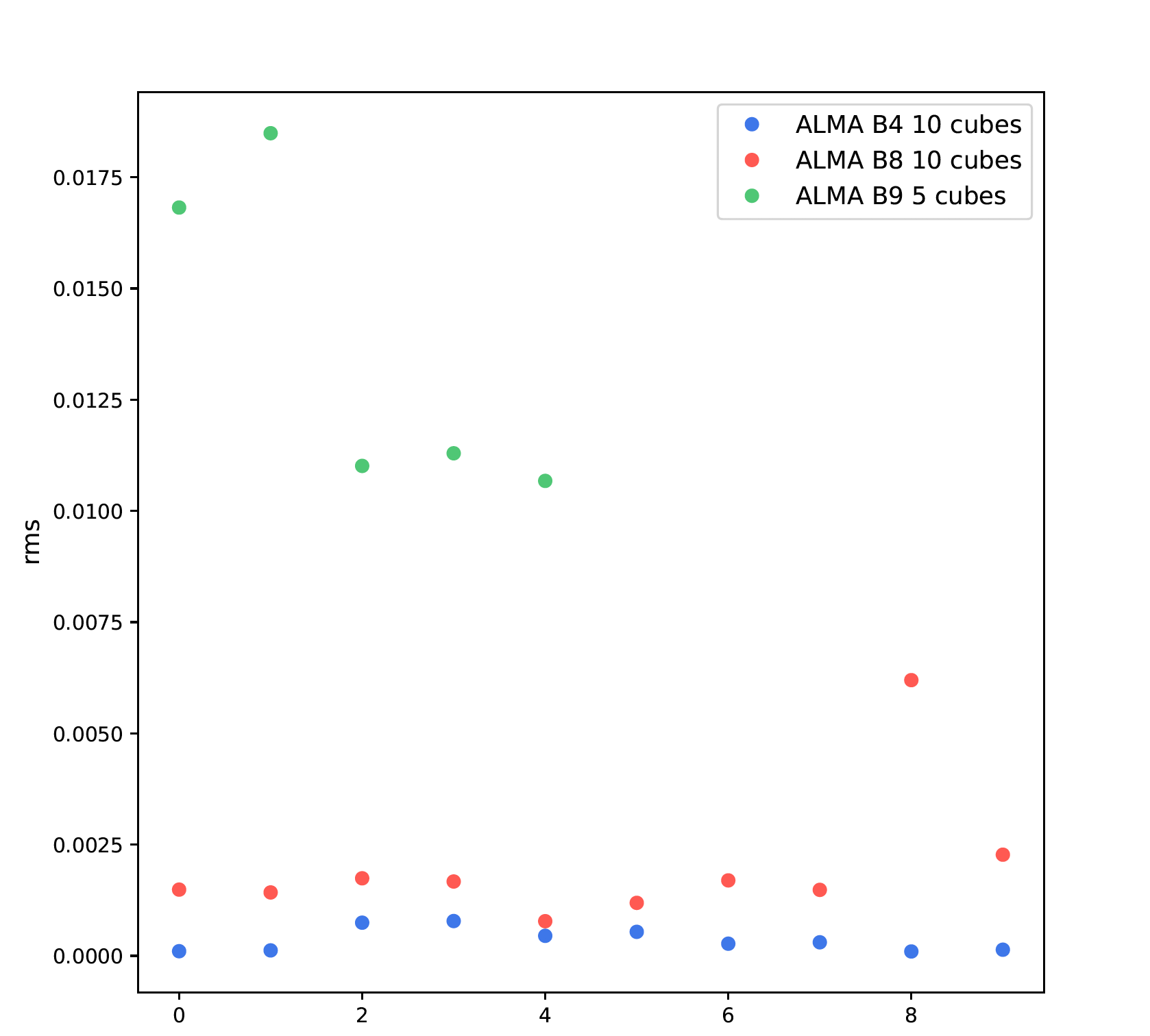
**uid\_\_\_A001\_Xa0\_X132a.cube\_0.J1733-1304\_B7.image QSO**

**uid\_\_\_A001\_Xa0\_X132a.cube\_1.J1733-1304\_B7.image ms > 0.5**

**uid\_\_\_A001\_X10d\_X5c.cube\_0.J2148+0657\_B7.image | QSO**

**uid\_\_\_A001\_X10d\_X5c.cube\_1.J2148+0657\_B7.image**

**uid\_\_\_A001\_X10d\_X5c.cube\_2.J2148+0657\_B7.image**

List  

# Matching the fake sources with detection + real detections

J1924 , has many cubes, ideal for testing the programs

Match\_all.py takes the Duchamp.in file with the input list fo r the Duchamp (filename and list of flagged channels). It requires the \*\_results.txt and \*\_mock.txt , \*stat fiels for each entry.It then compare the central pixels of the reported detection with plugged mock sources.

The output is ithe file \*out for each entry, witht the detections SN and widths and also corresponding matches if matched.

### Indexes for duchamp results file for fixed treshhold option

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 Obj ID | 1 – Name | 2 – X | 3 – Y | 4 – Z | 5 RA | 6 DEC | 7 RA [deg] |
| 9 FREQ | 10 Maj | 11 Min | 12 PA |  |  |  |  |
|  |  |  |  |  |  |  |  |

0 -ObjID

1- Name 8 DEC

9 FREQ

10 MAJ

11 MIN

12 PA

13 w\_RA

14 w\_DEC

15 w\_50

16 w\_20

17 w\_FREQ

18 F\_int

19 F\_tot

20 F\_peak

21 X1

22 X2

23 Y1

24 Y2

25 Z1

26 Z2

27 Nvoxel

28 Nchan

29 Nspatpix

30 Flag

31 X\_av

32 Y\_av

33 Z\_av

34 X\_cent

35 Y\_cent

36 Z\_cent

37 X\_peak

38 Y\_peak

39 Z\_peak

8 DEC [deg]

# Volume calculation for CO transitions in ALMACAL deep

CO line luminocity prime for different transition (from 1-0 to 7-6) and for different redshifts from 1 to 5. Assumed signal I= 1 v = 200 km/s. To apply for CO volume calculation one must multiply the LCO value with the value of I. Files 'Luminocity\_CO\_prime\_'+str(i)+'.txt' whete I form 0 to 6 corresponds to consecutive transitions.

**Co\_volume.py**  calculation s of the volume for ech CO transition