Mining the Science in ALMA Data

Peter Teuben, Lee Mundy, Marc Pound

28 February 2022

SUMMARY

This is the final report for the ALMA Study started in 2019 titled "Data Mining the Alma Science Archive using Admit in AWS". The title of the current report has been changed to better reflect the final direction and outcome of the study.

The study successfully implemented a database consisting of tables of general observational and scientific data harvested from the `astroquery.alma` database and the data products produced by the ALMA Data-MIning Toolkit (ADMIT). The database is queried using a new module `astroquery.admit` that we developed. The queries can be done from a Jupyter notebook and the query results are returned as a pandas dataframe. The implementation with Jupyter notebooks and pandas dataframes enables the user to conduct further analysis of the data with the powerful tools available in pandas and python.

We present examples of 7 science cases covering 4 different ALMA projects. We highlight a range of inquiries including source/line searches, light curves for an object with multiple epoch observations, and reporting measurements of line fluxes. We provide pdf's of the Juypter notebook sessions for each case.

We found that the study concept of a science query database derived from ADMIT products is viable and could be implemented as an ALMA-wide service for all ALMA archival data. We made recommendations for the improvements if the concept is developed for general use.

1 Introduction

This study[1] is a pathfinder development study to create a Science Query Database that allows science-driven queries of ALMA projects. Our goal is to enable science discovery with ALMA archival data by enhancing users' ability to identify and access relevant projects through creation of data-base tables of metadata extracted from processing the science images and cubes. We ran the ALMA Data Mining Toolkit (ADMIT) on four selected public Cycle 4,5,6 data to create a standard set of science products, and ingested the ADMIT science metadata (e.g., line identifications, line characteristics, source intensities, image statistics, source coordinates) into the Science Query Database. We merged these metadata with those already captured by the ALMA Science Archive. Combining these with the existing archive interface capability of searching project abstracts and science keywords allow investigators to make queries that lead to science-driven browsing of the archive.

The query engine is based on the Astroquery python package, adapted for ADMIT. The metadata available for query are gathered from a number of sources and are structured as nested relational tables (see Figure 1). With this implementation, queries can be integrated into many user environments. In this study we present examples of queries using Jupyter notebooks, which are familiar to many astronomers. The queries outcomes are returned as a pandas DataFrame which can then be further filtered and manipulated using the pandas package (also familiar to many astronomers) and, where applicable, plotted with matplotlib or other Python plotting packages.

In the original proposal, we outlined a proposed use of Amazon Web Services (AWS) for data processing and potentially as the remote location for relational database. We obtained a free grant of time on the AWS for testing. In the usage of this free time, we determined that AWS was not an appropriate development platform for the study. If, in the future, ALMA decides a full system for community usage, there should be a review of the cost efficiency of AWS at that time.

2 Database

The design and population of the database is at the core of this study. A well designed set of tables in a relational database is built around both the intended data content and the query requirements. For our goals of the study, the data content are the relevant facts about ALMA observations and the scientific data in the resulting image cubes. The query requirements are set out as a number of science-based questions about the observations and the measurements of continuum and spectral lines in the image cubes.

The root table for our database is a table created from te `astroquery.alma` database. This ALMA table is a simple 63 column table that characterized the observational parameters for any ALMA MOUS. We add to this three additional

[1] `https://github.com/teuben/study7` for supporting material

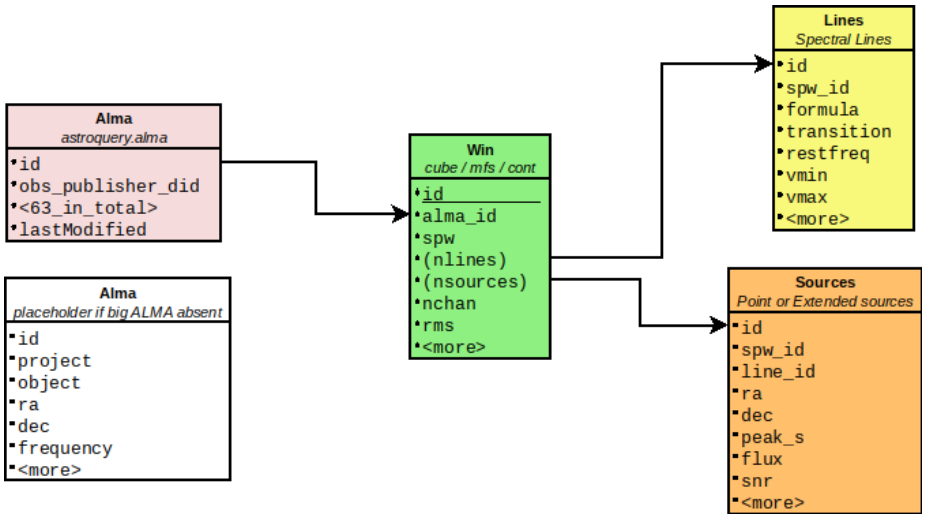


Figure 1: Tables used in this study (see text for explanation of A/W/L/S). For completeness we've drawn the Alma table twice, to accentuate that we keep our own (pink) copy since we don't have access to the full one. The Win, Lines, and Sources tables contain data produced by ADMIT

.

tables derived from ADMIT products that characterize the additional information and science results organized based on the spectral window, the spectral lines observed within a spectral window, and the emission sources detected. These tables are:

1. **ALMA (A)**: This mirrors the original ALMA table, but we only ingest the MOUS's for which we have processed ADMIT data, and for the purposes of this prototype we didn't ingest all columns, but focused on the columns which would be necessary for our example science use cases.

2. **Window (W)**: The W table contains information about the spectral window and summary information about the science content of the window. A MOUS with N (typically 4) spectral windows occupies N rows in the ALMA table, and will typically get processed by the ALMA imaging pipeline into 2N+1 FITS files[2]: N+1 will be continuum (N are "mfs" and 1 is "cont"), and N are spectral line cubes ("cube"). Each FITS file is treated as an entry in this table, In each cube there might be **nlines** detected, and **nsources**. We currently attempt to detect sources in the "CubeSum", which is the sum of all channels over a spectral window, and the "mfs/cont".

3. **Lines (L)**: For each spectral line detected by ADMIT we record its prop-

[2]technically there are several: pb/pbcor/mask

erties, such as the total moment-0 flux (in Jy.km/s), the moment-1 and moment-2 values at the peak of the emission, both in km/s, in the L table.

4. **Sources (S)**: For each source detected in the CubeSum and mfs/cont, we record characteristics such as the position, source size, peak brightness, SNR. We also record the characteristics of each spectral line at the position the each source in the CubeSum. Currently, the continuum sources do not have associated spectral line information but this is a simple upgrade to our processing which can be implemented in the future.

Figure 1 displays a schematic diagram of the table structure. The table structure is hierarchical because each MOUS in the A table will have entries for each spectral window in the W table. Each entry in the W table can have multiple lines displayed in the L table and multiple sources displayed in the S table.

In our proposal for this study, we suggested writing a simple query language based on `astroquery` to provide the user access to the database. However, we found that implementing a full `astroquery.admit` module in a fork of that Astropy affiliated package was more fruitful and in better keeping with the `astroquery` ecosystem.

3 Data and Workflow

3.1 ALMA data used

We used data from the following projects:

1. **2016.1.00650.S**: a typical single object, the barred galaxy NGC3504, with two 12m and one 7m observations, with CO, CS and some continuum. Of course we only query each dataset individually, we do not have access to the combined data. Archive data-size: 12 GB

2. **2017.1.00964.S**: a series of seven galaxies to study gas flow near black holes at high resolution. Some objects consist of multiple observations at different resolutions. NGC3504 is also conveniently part of this sample, and has two observations. Archive data-size: 451 GB

3. **2018.1.01055.L**: the MAPS ALMA large survey (2 TB). The MAPS team web page has processed data and downloadable science cubes. We processed this, but also picked one dataset from the teams website (MWC_480) to compare the results. There are five targets in this survey, with around 20 lines detected.

4. **2016.1.00875.S:** Since only a select number of data from Cycles 2,3,4 were used in the ARI-L project, we picked their example **uid://A011/X88f/X11a** for processing to get some idea how the data products compare. – some pipeline products in this tree had different filename conventions and got skipped – Archive data-size: 235 GB

The two first projects take about 3 hours of (single core) ADMIT processing together, and produce just over 100 admit datasets. We used the native resolution only, unlike our JAO "COVID" experiment where we used two resolutions per dataset. The large MAPS survey took about 11 hours to process with ADMIT.

When the ALMA observatory was shutdown for COVID, we were able to run all ALMA data prior to 2020 through ADMIT, in collaboration with the JAO and ESO. This uncovered some minor problems but was successful in producing ALMA data products for most observations. The current study utilizes the standard ADMIT products (only for the datasets enumerated above), with the addition of one new task for finding limits to line fluxes toward source locations detected in the cubesum map.

3.2 Workflow

We process the ALMA pipeline products on the MOUS level only through ADMIT, at the (usually native) resolution delivered. The processing included the target source, phase and bandpass calibrator images. We do not combine data on any level. We only process public data (PASS or SEMIPASS in QA2). We do not process Total Power data.

- Selected data are downloaded (this can now also be scripted up via astroquery.alma) and the pipeline data are placed in a read-only hierarchy of `Project/SOUS/GOUS/MOUS/product/`.

- The Admit After Pipeline (AAP) procedure runs ADMIT on all files in the product tree, but places them in a different writable location. We also included the calibrator images, as they were delivered with the scientific target. However, we kept the science and calibrator ADMIT products in a different database for later queries. For the science we used the aggregate continuum ("cont"), multi-frequency synthesis continuum images ("mfs"), and continuum subtracted spectral cubes ("cube").

- ADMIT results are processed and ingested into our relational database system (admit.db) that combines the original "63 alma fields" with the new "admit fields". This process revealed some problems in processing, due to issues in CASA, Imaging Pipeline or ADMIT. These problems were all solved or worked around.

- The `astroquery.admit` module is then used to query the data in admit.db. The user interface is to pass a series of *keyword=value* (the "payload" in astroquery lingo) that describe their science query to the `query()` method which translates the request to SQL and interrogates the database. They keywords are shown in Tables 1 – 4. We have assembled a number of use cases in the Appendix, in the form of reproducible Jupyter notebooks.

- The query returns a pandas DataFrame which is then available to the user in their Jupyter notebook. The user is then free to analyze and visualize the data using pandas, matplotlib, or and their favorite Python tools.

The query does in general not have direct access to the graphical ADMIT products as they are not included in admit.db. However, users who run the AAP procedure on their own products locally will have the ADMIT products available and can use the queries to identify the products of most interest to them. A future enhancement could allow fetching of user-selected ADMIT products from local disk or from the ALMA archive once ADMIT products are readily available there.

One aspect of the query design is that the returned pandas DataFrame schema depends on the query. The A and W tables are always returned, joined on the `obs_id`. If a query does not a request for information from an S or L table, the returned DataFrame does not include columns from that table or tables. For example, a request for MOUS that match a given target name will only return information from the A+W tables. A request for lines detected in a target will return information from the A, W, and L tables but not the S table.

3.3 Sample Session

In the attached notebooks you will find many different types of examples, but below we present a simple (i)python session, looking for all projects where the abstract mentions the words "Black Hole", and where CO was found. The returned pandas DataFrame includes all information from the A, W and L tables, 40 columns. We can then request from the DataFrame selected information to see: observation id, target name, resolution and VLSR in the example below; we could have also just type "p" to get the full table with a scrollbar. The simple printout below tell us where CO was found, and the VLSR derived from the ADMIT ingestion process. If you were interest in only high resolution, you could now re-do your search with a limit on s_resolution as a keyword. The Jupyter notebook Science Case #1 expands slightly on the example shown here.[3]

```
from astroquery.admit import ADMIT
import pandas as pd

a = ADMIT('admit.db')
p = a.query(formula="CO",project_abstract="*lack*ole")
ci = ['obs_id', 'target_name', 's_resolution', 'vlsr']
print(p[ci])
```

```
                obs_id target_name  s_resolution         vlsr
0  uid://A001/X1288/Xba2     NGC3049      0.332481  1447.989431
1  uid://A001/X1288/Xba6     NGC3504      0.034121  1521.106704
2  uid://A001/X1288/Xba8     NGC3504      0.200656  1521.106704
```

[3]Determining VLSR can be hazardous to your health

```
3  uid://A001/X1288/Xbae     NGC3593      0.289034   628.000000
4  uid://A001/X1288/Xbc0     NGC4414      0.223839   715.580697
5  uid://A001/X1288/Xbc4     NGC5253      0.074072   401.856222
6  uid://A001/X1288/Xbc6     NGC5253      0.258194   401.856222
7  uid://A001/X1296/X6f3   5-NGC3773      0.299770   978.913173
```

4 Recommendations

- Our study has been successful in creating a local prototype of the proposed database tables and query structure. While the study implementation is solely local, there is no significant barrier to utilizing a remote database (such as one at ALMA Science Center) and using TAP queries to retrieve the desired panda dataframe to the user's local Jupyter notebook. The implementation would be similar to that currently used for astroquery.alma.
- The ADMIT data products include images which have significant value to the user in exploring the science. This is an area where additional investigation of the technology options could provide a good solution for providing these to a user in real time.
- A properly designed set of relational database tables is needed to optimally make use of the information. Currently our "alma" table is not normalized.
- For software using the pipeline results it would be very useful if there was a selected MOUS for which pipeline results of different versions of the pipeline were kept online for a more robust testing of code evolution. Notably for ADMIT this has often been a struggle.
- A missing VLSR (we do our best to guess one) is a problem in line identification. Inclusion of velocity information somewhere on the ALMA data flow is highly desirable.
- We found it important that the ADMIT keywords are flexibly defined, in fact, if users could define their own, add these to the admit query. A "weakness" of the BDP structures that ADMIT uses is a very rigorously defined set of keywords with a DTD.

4.1 Items of Note during the Study

- ADMIT depends on CASA tasks and tools. It can be challenging to use this as an API, given that pre- and post-conditions are not always well defined. One example is grabbing the flux at a given pixel using imval, sometimes returning an array, sometimes a single value.
- Our workflow depended on an astroquery (or pyvo) to the ALMA science archive to success, but often out of a 100 queries we had a few small failures.

- There is a tight relationship between the way we mine the science data (e.g. if we use a continuum map to probe the line map),and the design of the admit tables, which then result in a set of queries. We have not fully explored the ramifications of this.
- The analytics used with queries has also proven to shake down some problems in admit/casa algorithms - e.g. detecting a line, but not a source
- For anybody with a large project collecting their own ADMIT products and making a local database and running queries can be an effective tool to data mine your own project. The github repo is set up to flexibly change the keywords to enhance the queries above those that ADMIT currently handles.

5 Science Case Studies

In order to showcase the capabilities of the science queries, we have assembled a number of jupyter notebooks, using some of the material we outlined in the proposal. We highlight a few:[4]

Case 1: short example CO near black holes

This is the actual notebook matching the example from section 3.3. The notebook goes into slightly more detail, adds a column and discusses the dreaded VLSR.

Case 2: query introduction

Showcasing lots of styles of query, via `admit.query()` and `admit.sql()`. This also shows examples of plotting search results with matplotlib.

Case 3: Find any project where *spectral line(s)* was detected

This is a relatively straightforward question, given a spectal line, find the projects where this was observed.

Case 4: NGC3504

A galaxy in two independent projects. CO and CS are detected. Continuum flux of the nuclear region can be monitored, a poor man's lightcurve. We do a better lightcurve in Case #7.

[4] See also: `https://github.com/teuben/study7/tree/master/notebooks`

Case 5: MAPS alma large program

We are still working through this. All 2TB data downloaded, takes 11 hours to process it through ADMIT, but only got about 330/377 datasets clean so far, plus perhaps some ingestion problems too. Very large datasets had problems.

Case 6: noise stats: comparing ALMA and ADMIT

Comparing what ALMA tells us (via the sensitivity calculator) and what ADMIT has measured.

Case 7: lightcurve – of calibrators

We don't have enough data sampled in time for a science source, but we do for the calibrators! We keep a separate notebook for this special case. ALMA keeps a database of calibrators online, we were able to download that and compare the two. See for yourself who you should believe.

6 Conclusion

We have shown that a science-based query system that leverages the science data products produced by ADMIT along with the archive metadata produced by the AAP can be a powerful way to extract new science from ALMA archival data. Our prototype shows how it can be implemented with well-thought out relational database tables layered with a Python API based on `astroquery`. It also works well within the now popular ecosystem of astropy and pandas dataframes for any further analysis.

On a more personal note: it was good to see that the labor we put into ADMIT is really showing off its fruits in this study.

7 References

1. astroquery: https://github.com/teuben/astroquery (branch: admit) and https://astroquery.readthedocs.io/
2. pyvo: https://pyvo.readthedocs.io/
3. alminer: https://alminer.readthedocs.io
4. ARI-L report (Stoehr et.al - 2022) - https://arxiv.org/pdf/2107.11071.pdf
5. ALMA science Archive Notebooks - https://almascience.eso.org/alma-data/archive/archive-notebooks

Ackowledgements

Peter Teuben acknowledges many fruitful discussions and hours of labor with Jorge Garcia (JAO) and Felix Stoehr (ESO) during and after our “COVID break”.

This paper makes use of the following ALMA data:
ADS/JAO.ALMA/#2016.1.00875.S,
ADS/JAO.ALMA/#2016.1.00650.S,
ADS/JAO.ALMA/#2017.1.00964.S, and
ADS/JAO.ALMA/#2018.1.01055.L

8 Table Keyword Descriptions

Table 1: Window Table Keywords (W)

Keyword	Description	Database Table
alma_id	ALMA ID	Window
win_id	Spectral window ID	Window
spw	Spectral window name	Window
nlines	Number of lines	Window
nsources	Number of sources	Window
nchan	Number of channels	Window
win_peak	Window peak flux	Window
win_rms	Window RMS noise	Window
win_snr	Window Signal to noise ratio (peak/rms)	Window
bmaj	Beam major axis	Window
bmin	Beam minor axis	Window
bpa	Beam PA	Window
freqc	Frequency center (GHz)	Window
freqw	Frequency width (GHz)	Window
vlsr	LSR Velocity (km/s)	Window
fcoverage	Frequency coverage?	Window

Table 2: Line Table Keywords (L)

Keyword	Description	Database Table
l_win_id	Spectral window ID(L)	Lines
restfreq	Rest frequency	Lines
formula	Formula	Lines
transition	Transition	Lines
velocity	Velocity	Lines
vmin	Minimum velocity	Lines
vmax	Maximum velocity	Lines
mom0flux	Moment zero flux	Lines
mom1peak	Moment one at peak	Lines
mom2peak	Moment two at peak (km/s)	Lines

Table 3: Source Table Keywords (S)

Keyword	Description	Database Table
s_win_id	Spectral Window ID(S)	Sources
lines_id	Line ID	Sources
ra	RA (Degrees)	Sources
dec	Dec (Degrees)	Sources
flux	Flux	Sources
source_snr	Source Signal to noise ratio	Sources
source_peak	Source Peak flux	Sources
smaj	Source major axis	Sources
smin	Source minor axis	Sources
spa	Source PA	Sources
region	Search Region	Sources

Table 4: ALMA Table Keywords (A)

Keyword	Description	Database Table
obs_id	Observation	Alma
source_name_resolver	Source name (astropy Resolver)	Alma
source_name_alma	Source name (ALMA)	Alma
ra_dec	RA Dec (Sexagesimal)	Alma
galactic	Galactic (Degrees)	Alma
spatial_resolution	Angular resolution (arcsec)	Alma
spatial_scale_max	Largest angular scale (arcsec)	Alma
fov	Field of view (arcsec)	Alma
frequency	Frequency (GHz)	Alma
bandwidth	Bandwidth (Hz)	Alma
spectral_resolution	Spectral resolution (KHz)	Alma
band_list	Band	Alma
start_date	Observation date	Alma
integration_time	Integration time (s)	Alma
polarisation_type	Polarisation type (Single, Dual, Full)	Alma
line_sensitivity	Line sensitivity (10 km/s) (mJy/beam)	Alma
continuum_sensitivity	Continuum sensitivity (mJy/beam)	Alma
water_vapour	Water vapour (mm)	Alma
project_code	Project code	Alma
project_title	Project title	Alma
pi_name	PI name	Alma
proposal_authors	Proposal authors	Alma
project_abstract	Project abstract	Alma
publication_count	Publication count	Alma
science_keyword	Science keyword	Alma
scientific_category	Scientific category	Alma
bibcode	Bibcode	Alma
pub_title	Title	Alma
first_author	First author	Alma
authors	Authors	Alma
pub_abstract	Abstract	Alma
publication_year	Year	Alma

Science Case #1: Black Holes and CO gas

Expanded example from the report, also shows how to add a new column with the linear scale in parsecs, and compare the VLSR and our rdz lookup table.

In [1]:
```
from astroquery.admit import ADMIT
import pandas as pd
```

In [2]:
```
a = ADMIT()
a.check()
```

```
Found  /home/teuben/ALMA/study7/query/admit.db
Checking db.... 0
71 71 71
Database version: 27-feb-2022. core.py version: 26-feb-2022
header    : 1 entries
alma      : 131 entries
win       : 130 entries
lines     : 33 entries
sources   : 801 entries
```

Find projects about black holes where CO was observed.

Note: we avoid B and H for because the underlying SQL search is case sensitive

In [3]:
```
p = a.query(formula="CO",project_abstract="*lack*ole*")
```

```
select * from alma inner join win on (win.a_id = alma.id)  inner join lin
es on (lines.w_id = win.id )  WHERE lines.formula='CO' AND alma.project_a
bstract LIKE '%lack%ole%'
```

In [4]:
```
p
```

Out[4]:

	id	obs_id	target_name	s_ra	s_dec	frequency	s_resolution	
0	6	uid://A001/X1288/Xba2	NGC3049	148.706950	9.271113	229.424396	0.332481	58
1	10	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	58
2	14	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	229.371216	0.200656	58
3	15	uid://A001/X1288/Xbae	NGC3593	168.654583	12.817675	230.055098	0.289034	58
4	26	uid://A001/X1288/Xbc0	NGC4414	186.612807	31.223649	229.988814	0.223839	58
5	30	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	230.225275	0.074072	58
6	34	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	230.225281	0.258194	58
7	42	uid://A001/X1296/X6f3	5-NGC3773	174.553958	12.111899	229.781272	0.299770	58

8 rows × 40 columns

In [5]:
```
ci=['obs_id','target_name','s_resolution', 'vlsr']
print(p[ci])
```

```
                obs_id target_name  s_resolution         vlsr
0  uid://A001/X1288/Xba2     NGC3049      0.332481  1447.989431
1  uid://A001/X1288/Xba6     NGC3504      0.034121  1521.106704
2  uid://A001/X1288/Xba8     NGC3504      0.200656  1521.106704
3  uid://A001/X1288/Xbae     NGC3593      0.289034   628.000000
4  uid://A001/X1288/Xbc0     NGC4414      0.223839   715.580697
5  uid://A001/X1288/Xbc4     NGC5253      0.074072   401.856222
6  uid://A001/X1288/Xbc6     NGC5253      0.258194   401.856222
7  uid://A001/X1296/X6f3   5-NGC3773      0.299770   978.913173
```

This looks interesting. But we like to know the resolution in physical pc, not in arcsec.

Note: the current **s_resolution** is in arcsec, which is in error, it should have been degrees. This is a fix the archive group will make sometime in 2022

In [6]:
```
H0 = 75           # just to get something....
ascale = 1.0      # this will be 3600 once s_resolution is in degrees agai
n

p['size_pc'] = p['s_resolution']/ascale * p['vlsr']/H0 * 1e6/206265
```

In [7]:
```
ci=['obs_id','target_name','s_resolution', 'vlsr','size_pc']
print(p[ci])
```

```
                obs_id target_name  s_resolution         vlsr    size_p
c
0  uid://A001/X1288/Xba2     NGC3049      0.332481  1447.989431  31.12037
6
1  uid://A001/X1288/Xba6     NGC3504      0.034121  1521.106704   3.35502
5
2  uid://A001/X1288/Xba8     NGC3504      0.200656  1521.106704  19.72987
0
3  uid://A001/X1288/Xbae     NGC3593      0.289034   628.000000  11.73335
2
4  uid://A001/X1288/Xbc0     NGC4414      0.223839   715.580697  10.35396
9
5  uid://A001/X1288/Xbc4     NGC5253      0.074072   401.856222   1.92415
8
6  uid://A001/X1288/Xbc6     NGC5253      0.258194   401.856222   6.70703
3
7  uid://A001/X1296/X6f3   5-NGC3773      0.299770   978.913173  18.96901
7
```

We don't have information here on the expected black hole mass, which should be ~0.01 * bulge_mass

In [8]:
```
p[ci].describe()
```

Out[8]:

	s_resolution	vlsr	size_pc
count	8.000000	8.000000	8.000000
mean	0.214021	952.051144	12.986600
std	0.107696	487.073056	9.806514
min	0.034121	401.856222	1.924158
25%	0.169010	571.464055	5.869031
50%	0.241016	847.246935	11.043660
75%	0.291718	1466.268749	19.159231
max	0.332481	1521.106704	31.120376

Accuracy of VLSR?

The VLSR that ADMIT uses is not reliable. Since the PI information of VLSR is optional, and not made available in the fits headers, ADMIT uses the RESTFREQ in the fits header, and compares that to the frequency in the middle of the spw. The PI can easily use another algorithm, though often this method works.

We are using a table lookup provided by Felix Stoehr (ESO) that we added as an option to a special query in out admit astroquery. This query_rdz() returns the 'z' of the object, e.g.

In [9]:
```
a.query_rdz('uid://A001/X1288/Xba6','NGC3504')
```

```
Found 49765 entries in rdz
Searching for  uid://A001/X1288/Xba6 NGC3504
uid://A001/X1288/Xba6.source.NGC3504|165.796629167|27.9724361111|0.005104
|NGC  3504|
```

Out[9]:
```
0.005104
```

an alternative method would be a lookup via NED , at least for galaxies. Here's the example for NGC3504

In [10]:
```
from astroquery.ipac.ned import Ned
Ned.query_object("NGC3504")
```

Out[10]: *Table length=1*

No.	Object Name	RA	DEC	Type	Velocity	Redshift	Redshift Flag	Magnitude and Filter	Separation
		degrees	degrees		km / s				arcmin
int32	bytes30	float64	float64	object	float64	float64	object	object	float64
1	NGC 3504	165.79671	27.9725	G	1525.0	0.005087		11.4B	--

Looping over all galaxies in our query, we grab the redshift in a new array, and add it to the dataframe.

In [11]:
```
vz = np.zeros(len(p))
for i in range(len(p)):
    vz[i] =  a.query_rdz(p['obs_id'][i],p['target_name'][i])
```

```
Searching for  uid://A001/X1288/Xba2 NGC3049
uid://A001/X1288/Xba2.source.NGC3049|148.706517455|9.2710935803|0.0048540
0017|NGC 3049|

Searching for  uid://A001/X1288/Xba6 NGC3504
uid://A001/X1288/Xba6.source.NGC3504|165.796629167|27.9724361111|0.005104
|NGC  3504|

Searching for  uid://A001/X1288/Xba8 NGC3504
uid://A001/X1288/Xba8.source.NGC3504|165.796629167|27.9724361111|0.005104
|NGC  3504|

Searching for  uid://A001/X1288/Xbae NGC3593
uid://A001/X1288/Xbae.source.NGC3593|168.654166667|12.8176666667|0.002094
99989|NGC 3593|

Searching for  uid://A001/X1288/Xbc0 NGC4414
uid://A001/X1288/Xbc0.source.NGC4414|186.612870833|31.2235444444|0.002395
|NGC  4414|

Searching for  uid://A001/X1288/Xbc4 NGC5253
uid://A001/X1288/Xbc4.source.NGC5253|204.983179583|-31.6401077778|0.00135
799998|NGC 5253|

Searching for  uid://A001/X1288/Xbc6 NGC5253
Searching for  uid://A001/X1296/X6f3 5-NGC3773
uid://A001/X1296/X6f3.source.5-NGC3773|174.553647831|12.1120476073|0.0032
7600003|NGC 3773|
```

In [12]:
```
p['rdz'] = vz*300000
```

In [13]:
```
ci=['obs_id','target_name', 'vlsr','rdz','size_pc']
print(p[ci])
```

```
                 obs_id target_name         vlsr          rdz    size_pc
0  uid://A001/X1288/Xba2     NGC3049  1447.989431  1456.200051  31.120376
1  uid://A001/X1288/Xba6     NGC3504  1521.106704  1531.200000   3.355025
2  uid://A001/X1288/Xba8     NGC3504  1521.106704  1531.200000  19.729870
3  uid://A001/X1288/Xbae     NGC3593   628.000000   628.499967  11.733352
4  uid://A001/X1288/Xbc0     NGC4414   715.580697   718.500000  10.353969
5  uid://A001/X1288/Xbc4     NGC5253   401.856222   407.399994   1.924158
6  uid://A001/X1288/Xbc6     NGC5253   401.856222          NaN   6.707033
7  uid://A001/X1296/X6f3   5-NGC3773   978.913173   982.800009  18.969017
```

One of the obsid's for NGC5253 was apparently not in the "rdz" database, luckily another one did.

Science Case #2: Query Introduction

Find all *source-type* within a given area of the sky with emission from *molecule(s)* detected.

The science motivation might be to find all young stellar objects (YSOs) in Taurus with ALMA detections of CO J=2-1. The return from this query would include images of the moment maps of CO emission, peak intensity, resolution, noise, correlator setting, and other relevant information from any Taurus YSOs (as identified by science keywords, abstract text, SIMBAD coordinate matches) in the publicly accessible ALMA Science Archive. The user could identify which data are appropriate for their study an pull the u,v data or image cubes of interest from the archive.

Optionally, the user could then create a query to retrieve the full set of science products for all sources, or the sources of interest. This would allow exploration of continuum and other lines detected in the same observations. The quantitative data are returned as tables in the user's Python environment (for example Jupyter notebook) which can then be manipulated (see Section 4), for example to produce a plot of continuum flux versus CO integrated intensity or CO intensity versus [13CO/12CO] intensity ratio.

NB on nomenclature I am using below: Targets are object names, sources are results of ADMIT source find

In [1]:
```
from astroquery.admit import ADMIT, ADMIT_FORM_KEYS
from astroquery.alma import Alma, tapsql
from astropy.coordinates import SkyCoord
from astropy import units as u
import pandas as pd
import matplotlib.pyplot as plt
# Display options
# display the whole table inthe notebook
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
pd.set_option('display.width', None)
pd.set_option('display.max_colwidth',25)
cols_to_exclude = ['id','project_abstract','obs_title','fcoverage','scienc
e_keyword']
```

In [2]:
```
a = ADMIT()
a.check()
```

```
Found  /home/teuben/ALMA/study7/query/admit.db
Checking db.... 0
71 71 71
Database version: 27-feb-2022. core.py version: 26-feb-2022
header    : 1 entries
alma      : 131 entries
win       : 130 entries
lines     : 33 entries
sources   : 801 entries
```

Show what search keywords are available

In [3]:
```
a.key_description.show_in_notebook()
```

Out[3]: *Table length=71*

Show 50 entries Search:

idx	Keyword	Description	Database Table
0	alma_id	ALMA ID	Window
1	win_id	Spectral window ID	Window
2	spw	Spectral window name	Window
3	nlines	Number of lines	Window
4	nsources	Number of sources	Window
5	nchan	Number of channels	Window
6	win_peak	Window peak flux	Window
7	win_rms	Window RMS noise	Window
8	win_snr	Window Signal to noise ratio (peak/rms)	Window
9	bmaj	Beam major axis	Window
10	bmin	Beam minor axis	Window
11	bpa	Beam PA	Window
12	freqc	Frequency center (GHz)	Window
13	freqw	Frequency width (GHz)	Window
14	vlsr	LSR Velocity (km/s)	Window
15	fcoverage	Frequency coverage?	Window
16	l_win_id	Spectral window ID(L)	Lines
17	restfreq	Rest frequency	Lines
18	formula	Formula	Lines
19	transition	Transition	Lines
20	velocity	Velocity	Lines
21	vmin	Minimum velocity	Lines
22	vmax	Maximum velocity	Lines
23	mom0flux	Moment zero flux	Lines
24	mom1peak	Moment one peak	Lines
25	mom2peak	Moment two peak (km/s)	Lines
26	s_win_id	Spectral Window ID(S)	Sources
27	lines_id	Line ID	Sources
28	ra	RA (Degrees)	Sources
29	dec	Dec (Degrees)	Sources
30	flux	Flux	Sources
31	source_snr	Source Signal to noise ratio	Sources
32	source_peak	Source Peak flux	Sources
33	smaj	Source major axis	Sources

idx	Keyword	Description	Database Table
34	smin	Source minor axis	Sources
35	spa	Source PA	Sources
36	region	Search Region	Sources
37	header_key	Key	Header
38	header_val	Value	Header
39	obs_id	Observation	Alma
40	source_name_resolver	Source name (astropy Resolver)	Alma
41	source_name_alma	Source name (ALMA)	Alma
42	ra_dec	RA Dec (Sexagesimal)	Alma
43	galactic	Galactic (Degrees)	Alma
44	spatial_resolution	Angular resolution (arcsec)	Alma
45	spatial_scale_max	Largest angular scale (arcsec)	Alma
46	fov	Field of view (arcsec)	Alma
47	frequency	Frequency (GHz)	Alma
48	bandwidth	Bandwidth (Hz)	Alma
49	spectral_resolution	Spectral resolution (KHz)	Alma

Showing 1 to 50 of 71 entries First Previous 1 2 Next Last

Find projects about black holes. Search keywords can use unix wildcards (*,?)

Note: we avoid B and H for because the underlying SQL search is case sensitive

In [4]:
```
result = a.query(project_abstract="*lack*ole*")
#len(result)
lst = set(zip(result['obs_id'],result['target_name']))
print("Found these projects and targets:",
      *('{}: {}'.format(*k[1]) for k in enumerate(lst)), sep="\n")
```

```
select * from alma inner join win on (win.a_id = alma.id)  WHERE alma.pro
ject_abstract LIKE '%lack%ole%'

Found these projects and targets:
uid://A001/X1288/Xba0: NGC3049
uid://A001/X1288/Xba2: NGC3049
uid://A001/X1288/Xbc4: NGC5253
uid://A001/X1288/Xbc0: NGC4414
uid://A001/X1288/Xbc6: NGC5253
uid://A001/X1288/Xba8: NGC3504
uid://A001/X1288/Xbba: NGC4402
uid://A001/X1288/Xbae: NGC3593
uid://A001/X1288/Xbbe: NGC4414
uid://A001/X1296/X6f3: 5-NGC3773
uid://A001/X1296/X6f1: 5-NGC3773
uid://A001/X1288/Xba6: NGC3504
```

Limit the result to those with sources found in the Cubesum

In [5]:
```
result[result['nsources']>0]
```

Out[5]:

	id	obs_id	target_name	s_ra	s_dec	frequency	s_resolution
2	3	uid://A001/X1288/Xba0	NGC3049	148.706950	9.271113	243.826966	0.039484
5	6	uid://A001/X1288/Xba2	NGC3049	148.706950	9.271113	229.424396	0.332481
9	10	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121
12	13	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	243.760022	0.200656
13	14	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	229.371216	0.200656
14	15	uid://A001/X1288/Xbae	NGC3593	168.654583	12.817675	230.055098	0.289034
19	20	uid://A001/X1288/Xbbe	NGC4414	186.612807	31.223649	242.171714	0.032896
21	22	uid://A001/X1288/Xbbe	NGC4414	186.612807	31.223649	229.988806	0.032896
25	26	uid://A001/X1288/Xbc0	NGC4414	186.612807	31.223649	229.988814	0.223839
27	28	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	242.420288	0.074072
29	30	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	230.225275	0.074072
33	34	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	230.225281	0.258194
37	38	uid://A001/X1296/X6f1	5-NGC3773	174.553958	12.111899	229.781745	0.054512
41	42	uid://A001/X1296/X6f3	5-NGC3773	174.553958	12.111899	229.781272	0.299770
46	59	uid://A001/X1288/Xba2	NGC3049	148.706950	9.271113	227.609012	0.332481
48	61	uid://A001/X1288/Xba2	NGC3049	148.706950	9.271113	243.816538	0.332481
49	62	uid://A001/X1288/Xba2	NGC3049	148.706950	9.271113	229.424396	0.332481

	id	obs_id	target_name	s_ra	s_dec	frequency	s_resolution
51	64	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	241.511633	0.034121
54	67	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	227.554761	0.200656
55	68	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	241.511635	0.200656
56	69	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	243.760022	0.200656
57	70	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	229.371216	0.200656
71	84	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	228.400349	0.074072
72	85	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	242.420288	0.074072
73	86	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	244.667658	0.074072
74	87	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	230.225275	0.074072
75	88	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	228.399516	0.258194
76	89	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	242.420716	0.258194
77	90	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	244.667664	0.258194
78	91	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	230.225281	0.258194
88	117	uid://A001/X1288/Xba2	NGC3049	148.706950	9.271113	241.566693	0.332481
89	118	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	241.511633	0.034121
90	119	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	241.511635	0.200656
95	125	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	242.420716	0.258194

Limit the result to those with sources found in the Cubesum and the lines were identified

In [6]:
```
result[(result['nsources']>0) & (result['nlines']>0)]
```

Out[6]:

	id	obs_id	target_name	s_ra	s_dec	frequency	s_resolution	
2	3	uid://A001/X1288/Xba0	NGC3049	148.706950	9.271113	243.826966	0.039484	5
5	6	uid://A001/X1288/Xba2	NGC3049	148.706950	9.271113	229.424396	0.332481	5
9	10	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	5
12	13	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	243.760022	0.200656	5
13	14	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	229.371216	0.200656	5
14	15	uid://A001/X1288/Xbae	NGC3593	168.654583	12.817675	230.055098	0.289034	5
19	20	uid://A001/X1288/Xbbe	NGC4414	186.612807	31.223649	242.171714	0.032896	5
25	26	uid://A001/X1288/Xbc0	NGC4414	186.612807	31.223649	229.988814	0.223839	5
27	28	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	242.420288	0.074072	5
29	30	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	230.225275	0.074072	5
33	34	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	230.225281	0.258194	5
41	42	uid://A001/X1296/X6f3	5-NGC3773	174.553958	12.111899	229.781272	0.299770	5

Black hole projects with CO 2-1 detections

In [7]:
```
result = a.query(formula="CO", transition="2-1", project_abstract="*lack*ole*")
lst = set(zip(result['obs_id'],result['target_name']))
print("Found these projects and targets:",
      *('{}: {}'.format(*k[1]) for k in enumerate(lst)), sep="\n")
result
```

```
select * from alma inner join win on (win.a_id = alma.id)  inner join lines on (lines.w_id = win.id )  WHERE lines.formula='CO' AND lines.transition='2-1' AND alma.project_abstract LIKE '%lack%ole%'

Found these projects and targets:
uid://A001/X1288/Xba2: NGC3049
uid://A001/X1288/Xbc4: NGC5253
uid://A001/X1288/Xbc0: NGC4414
uid://A001/X1288/Xbc6: NGC5253
uid://A001/X1288/Xba8: NGC3504
uid://A001/X1288/Xbae: NGC3593
uid://A001/X1296/X6f3: 5-NGC3773
uid://A001/X1288/Xba6: NGC3504
```

Out[7]:

	id	obs_id	target_name	s_ra	s_dec	frequency	s_resolution	
0	6	uid://A001/X1288/Xba2	NGC3049	148.706950	9.271113	229.424396	0.332481	58
1	10	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	58
2	14	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	229.371216	0.200656	58
3	15	uid://A001/X1288/Xbae	NGC3593	168.654583	12.817675	230.055098	0.289034	58
4	26	uid://A001/X1288/Xbc0	NGC4414	186.612807	31.223649	229.988814	0.223839	58
5	30	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	230.225275	0.074072	58
6	34	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	230.225281	0.258194	58
7	42	uid://A001/X1296/X6f3	5-NGC3773	174.553958	12.111899	229.781272	0.299770	58

Black hole projects within a square region

In [8]:
```
# Center coordinate
c = SkyCoord('11h15m30s', '+12d45m00s', frame='icrs')
# Square box side size
size = 40*u.degree
result = a.query(project_abstract="*lack*ole*", region = [c,size])
lst = set(zip(result['obs_id'],result['target_name']))
print("Found these projects and targets:",
      *('{}: {}'.format(*k[1]) for k in enumerate(lst)), sep="\n")
print(f"The unique targets  are: {result['target_name'].unique()}")
```

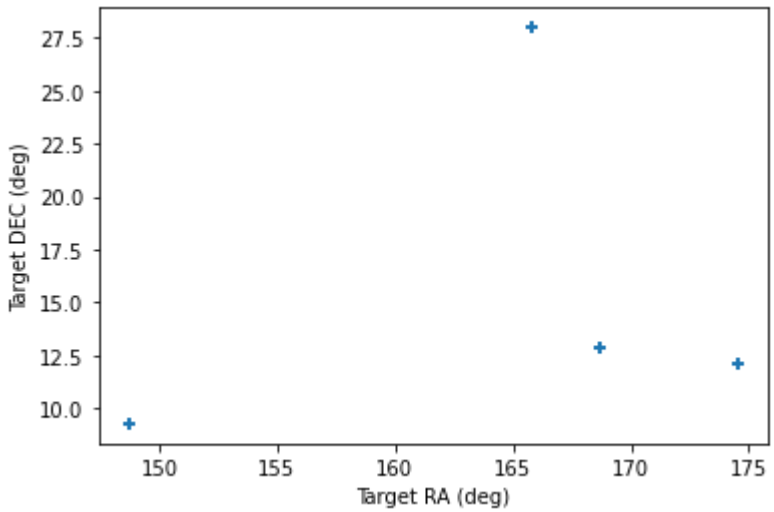
```
select * from alma inner join win on (win.a_id = alma.id)  inner join sou
rces on (sources.w_id = win.id)   WHERE alma.project_abstract LIKE '%lac
k%ole%' AND  sources.ra > 128.87499999999997 AND sources.ra < 208.8749999
9999997 AND sources.dec > -27.25 AND sources.dec < 52.75  AND sources.l_i
d = 0

Found these projects and targets:
uid://A001/X1288/Xba0: NGC3049
uid://A001/X1288/Xba2: NGC3049
uid://A001/X1288/Xba8: NGC3504
uid://A001/X1288/Xbae: NGC3593
uid://A001/X1296/X6f3: 5-NGC3773
uid://A001/X1296/X6f1: 5-NGC3773
uid://A001/X1288/Xba6: NGC3504
The unique targets  are: ['NGC3049' 'NGC3504' 'NGC3593' '5-NGC3773']
```

Plot them on the sky

In [9]:
```
plt.scatter(result['s_ra'],result['s_dec'],marker="+")
plt.xlabel("Target RA (deg)")
plt.ylabel("Target DEC (deg)")
```

Out[9]: Text(0, 0.5, 'Target DEC (deg)')



Black hole projects within a square region and CO2-1 detected

In [10]:
```
c = SkyCoord('11h15m30s', '+12d45m00s', frame='icrs')
size = 40*u.degree
result = a.query(project_abstract="*lack*ole*", region = [c,size], formula
="CO", transition="2-1")
print(f"Targets found: {result['target_name'].unique()}")
```

```
select * from alma inner join win on (win.a_id = alma.id)  inner join sou
rces on (sources.w_id = win.id)   inner join lines on (lines.w_id = win.i
d )  WHERE alma.project_abstract LIKE '%lack%ole%' AND  sources.ra > 128.
87499999999997 AND sources.ra < 208.87499999999997 AND sources.dec > -27.
25 AND sources.dec < 52.75  AND lines.formula='CO' AND lines.transition
='2-1' AND sources.l_id > 0

Targets found: ['NGC3049' 'NGC3504' 'NGC3593' 'NGC4414' '5-NGC3773']
```

In [11]:
```
#columns to include
ci = [x for x in result.columns if x not in cols_to_exclude]
result[ci]
```

Out[11]:

	obs_id	target_name	s_ra	s_dec	frequency	s_resolution	
0	uid://A001/X1288/Xba2	NGC3049	148.706950	9.271113	229.424396	0.332481	58379
1	uid://A001/X1288/Xba2	NGC3049	148.706950	9.271113	229.424396	0.332481	58379
2	uid://A001/X1288/Xba2	NGC3049	148.706950	9.271113	229.424396	0.332481	58379
3	uid://A001/X1288/Xba2	NGC3049	148.706950	9.271113	229.424396	0.332481	58379
4	uid://A001/X1288/Xba2	NGC3049	148.706950	9.271113	229.424396	0.332481	58379
5	uid://A001/X1288/Xba2	NGC3049	148.706950	9.271113	229.424396	0.332481	58379
6	uid://A001/X1288/Xba2	NGC3049	148.706950	9.271113	229.424396	0.332481	58379
7	uid://A001/X1288/Xba2	NGC3049	148.706950	9.271113	229.424396	0.332481	58379
8	uid://A001/X1288/Xba2	NGC3049	148.706950	9.271113	229.424396	0.332481	58379
9	uid://A001/X1288/Xba2	NGC3049	148.706950	9.271113	229.424396	0.332481	58379
10	uid://A001/X1288/Xba2	NGC3049	148.706950	9.271113	229.424396	0.332481	58379
11	uid://A001/X1288/Xba2	NGC3049	148.706950	9.271113	229.424396	0.332481	58379
12	uid://A001/X1288/Xba2	NGC3049	148.706950	9.271113	229.424396	0.332481	58379
13	uid://A001/X1288/Xba2	NGC3049	148.706950	9.271113	229.424396	0.332481	58379
14	uid://A001/X1288/Xba2	NGC3049	148.706950	9.271113	229.424396	0.332481	58379
15	uid://A001/X1288/Xba2	NGC3049	148.706950	9.271113	229.424396	0.332481	58379
16	uid://A001/X1288/Xba2	NGC3049	148.706950	9.271113	229.424396	0.332481	58379
17	uid://A001/X1288/Xba2	NGC3049	148.706950	9.271113	229.424396	0.332481	58379
18	uid://A001/X1288/Xba2	NGC3049	148.706950	9.271113	229.424396	0.332481	58379
19	uid://A001/X1288/Xba2	NGC3049	148.706950	9.271113	229.424396	0.332481	58379
20	uid://A001/X1288/Xba2	NGC3049	148.706950	9.271113	229.424396	0.332481	58379
21	uid://A001/X1288/Xba2	NGC3049	148.706950	9.271113	229.424396	0.332481	58379
22	uid://A001/X1288/Xba2	NGC3049	148.706950	9.271113	229.424396	0.332481	58379

	obs_id	target_name	s_ra	s_dec	frequency	s_resolution	
23	uid://A001/X1288/Xba2	NGC3049	148.706950	9.271113	229.424396	0.332481	58379
24	uid://A001/X1288/Xba2	NGC3049	148.706950	9.271113	229.424396	0.332481	58379
25	uid://A001/X1288/Xba2	NGC3049	148.706950	9.271113	229.424396	0.332481	58379
26	uid://A001/X1288/Xba2	NGC3049	148.706950	9.271113	229.424396	0.332481	58379
27	uid://A001/X1288/Xba2	NGC3049	148.706950	9.271113	229.424396	0.332481	58379
28	uid://A001/X1288/Xba2	NGC3049	148.706950	9.271113	229.424396	0.332481	58379
29	uid://A001/X1288/Xba2	NGC3049	148.706950	9.271113	229.424396	0.332481	58379
30	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	58050
31	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	58050
32	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	58050
33	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	58050
34	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	58050
35	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	58050
36	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	58050
37	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	58050
38	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	58050
39	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	58050
40	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	58050
41	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	58050
42	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	58050
43	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	58050
44	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	58050
45	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	58050

	obs_id	target_name	s_ra	s_dec	frequency	s_resolution	
46	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	5805(
47	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	5805(
48	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	5805(
49	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	5805(
50	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	5805(
51	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	5805(
52	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	5805(
53	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	5805(
54	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	5805(
55	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	5805(
56	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	5805(
57	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	5805(
58	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	5805(
59	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	5805(
60	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	229.371216	0.200656	5811(
61	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	229.371216	0.200656	5811(
62	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	229.371216	0.200656	5811(
63	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	229.371216	0.200656	5811(
64	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	229.371216	0.200656	5811(
65	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	229.371216	0.200656	5811(
66	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	229.371216	0.200656	5811(
67	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	229.371216	0.200656	5811(
68	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	229.371216	0.200656	5811(

	obs_id	target_name	s_ra	s_dec	frequency	s_resolution	
69	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	229.371216	0.200656	5811
70	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	229.371216	0.200656	5811
71	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	229.371216	0.200656	5811
72	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	229.371216	0.200656	5811
73	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	229.371216	0.200656	5811
74	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	229.371216	0.200656	5811
75	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	229.371216	0.200656	5811
76	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	229.371216	0.200656	5811
77	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	229.371216	0.200656	5811
78	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	229.371216	0.200656	5811
79	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	229.371216	0.200656	5811
80	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	229.371216	0.200656	5811
81	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	229.371216	0.200656	5811
82	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	229.371216	0.200656	5811
83	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	229.371216	0.200656	5811
84	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	229.371216	0.200656	5811
85	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	229.371216	0.200656	5811
86	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	229.371216	0.200656	5811
87	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	229.371216	0.200656	5811
88	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	229.371216	0.200656	5811
89	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	229.371216	0.200656	5811
90	uid://A001/X1288/Xbae	NGC3593	168.654583	12.817675	230.055098	0.289034	5838
91	uid://A001/X1288/Xbae	NGC3593	168.654583	12.817675	230.055098	0.289034	5838

	obs_id	target_name	s_ra	s_dec	frequency	s_resolution	
92	uid://A001/X1288/Xbae	NGC3593	168.654583	12.817675	230.055098	0.289034	58384
93	uid://A001/X1288/Xbae	NGC3593	168.654583	12.817675	230.055098	0.289034	58384
94	uid://A001/X1288/Xbae	NGC3593	168.654583	12.817675	230.055098	0.289034	58384
95	uid://A001/X1288/Xbae	NGC3593	168.654583	12.817675	230.055098	0.289034	58384
96	uid://A001/X1288/Xbae	NGC3593	168.654583	12.817675	230.055098	0.289034	58384
97	uid://A001/X1288/Xbae	NGC3593	168.654583	12.817675	230.055098	0.289034	58384
98	uid://A001/X1288/Xbae	NGC3593	168.654583	12.817675	230.055098	0.289034	58384
99	uid://A001/X1288/Xbae	NGC3593	168.654583	12.817675	230.055098	0.289034	58384
100	uid://A001/X1288/Xbae	NGC3593	168.654583	12.817675	230.055098	0.289034	58384
101	uid://A001/X1288/Xbae	NGC3593	168.654583	12.817675	230.055098	0.289034	58384
102	uid://A001/X1288/Xbae	NGC3593	168.654583	12.817675	230.055098	0.289034	58384
103	uid://A001/X1288/Xbae	NGC3593	168.654583	12.817675	230.055098	0.289034	58384
104	uid://A001/X1288/Xbae	NGC3593	168.654583	12.817675	230.055098	0.289034	58384
105	uid://A001/X1288/Xbae	NGC3593	168.654583	12.817675	230.055098	0.289034	58384
106	uid://A001/X1288/Xbae	NGC3593	168.654583	12.817675	230.055098	0.289034	58384
107	uid://A001/X1288/Xbae	NGC3593	168.654583	12.817675	230.055098	0.289034	58384
108	uid://A001/X1288/Xbae	NGC3593	168.654583	12.817675	230.055098	0.289034	58384
109	uid://A001/X1288/Xbae	NGC3593	168.654583	12.817675	230.055098	0.289034	58384
110	uid://A001/X1288/Xbae	NGC3593	168.654583	12.817675	230.055098	0.289034	58384
111	uid://A001/X1288/Xbae	NGC3593	168.654583	12.817675	230.055098	0.289034	58384
112	uid://A001/X1288/Xbae	NGC3593	168.654583	12.817675	230.055098	0.289034	58384
113	uid://A001/X1288/Xbae	NGC3593	168.654583	12.817675	230.055098	0.289034	58384
114	uid://A001/X1288/Xbae	NGC3593	168.654583	12.817675	230.055098	0.289034	58384

	obs_id	target_name	s_ra	s_dec	frequency	s_resolution	
115	uid://A001/X1288/Xbae	NGC3593	168.654583	12.817675	230.055098	0.289034	58384
116	uid://A001/X1288/Xbae	NGC3593	168.654583	12.817675	230.055098	0.289034	58384
117	uid://A001/X1288/Xbae	NGC3593	168.654583	12.817675	230.055098	0.289034	58384
118	uid://A001/X1288/Xbae	NGC3593	168.654583	12.817675	230.055098	0.289034	58384
119	uid://A001/X1288/Xbae	NGC3593	168.654583	12.817675	230.055098	0.289034	58384
120	uid://A001/X1288/Xbc0	NGC4414	186.612807	31.223649	229.988814	0.223839	58123
121	uid://A001/X1288/Xbc0	NGC4414	186.612807	31.223649	229.988814	0.223839	58123
122	uid://A001/X1288/Xbc0	NGC4414	186.612807	31.223649	229.988814	0.223839	58123
123	uid://A001/X1288/Xbc0	NGC4414	186.612807	31.223649	229.988814	0.223839	58123
124	uid://A001/X1288/Xbc0	NGC4414	186.612807	31.223649	229.988814	0.223839	58123
125	uid://A001/X1288/Xbc0	NGC4414	186.612807	31.223649	229.988814	0.223839	58123
126	uid://A001/X1288/Xbc0	NGC4414	186.612807	31.223649	229.988814	0.223839	58123
127	uid://A001/X1288/Xbc0	NGC4414	186.612807	31.223649	229.988814	0.223839	58123
128	uid://A001/X1288/Xbc0	NGC4414	186.612807	31.223649	229.988814	0.223839	58123
129	uid://A001/X1288/Xbc0	NGC4414	186.612807	31.223649	229.988814	0.223839	58123
130	uid://A001/X1288/Xbc0	NGC4414	186.612807	31.223649	229.988814	0.223839	58123
131	uid://A001/X1288/Xbc0	NGC4414	186.612807	31.223649	229.988814	0.223839	58123
132	uid://A001/X1288/Xbc0	NGC4414	186.612807	31.223649	229.988814	0.223839	58123
133	uid://A001/X1288/Xbc0	NGC4414	186.612807	31.223649	229.988814	0.223839	58123
134	uid://A001/X1288/Xbc0	NGC4414	186.612807	31.223649	229.988814	0.223839	58123
135	uid://A001/X1288/Xbc0	NGC4414	186.612807	31.223649	229.988814	0.223839	58123
136	uid://A001/X1288/Xbc0	NGC4414	186.612807	31.223649	229.988814	0.223839	58123
137	uid://A001/X1288/Xbc0	NGC4414	186.612807	31.223649	229.988814	0.223839	58123

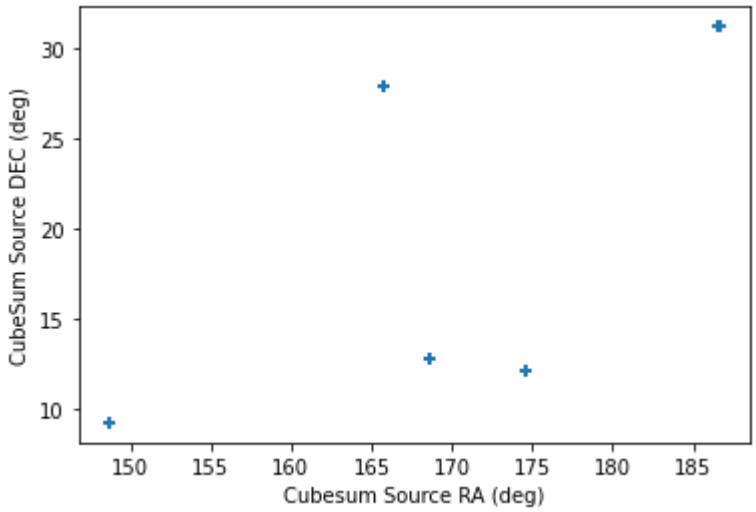
	obs_id	target_name	s_ra	s_dec	frequency	s_resolution	
138	uid://A001/X1288/Xbc0	NGC4414	186.612807	31.223649	229.988814	0.223839	58123
139	uid://A001/X1288/Xbc0	NGC4414	186.612807	31.223649	229.988814	0.223839	58123
140	uid://A001/X1288/Xbc0	NGC4414	186.612807	31.223649	229.988814	0.223839	58123
141	uid://A001/X1288/Xbc0	NGC4414	186.612807	31.223649	229.988814	0.223839	58123
142	uid://A001/X1288/Xbc0	NGC4414	186.612807	31.223649	229.988814	0.223839	58123
143	uid://A001/X1288/Xbc0	NGC4414	186.612807	31.223649	229.988814	0.223839	58123
144	uid://A001/X1288/Xbc0	NGC4414	186.612807	31.223649	229.988814	0.223839	58123
145	uid://A001/X1288/Xbc0	NGC4414	186.612807	31.223649	229.988814	0.223839	58123
146	uid://A001/X1288/Xbc0	NGC4414	186.612807	31.223649	229.988814	0.223839	58123
147	uid://A001/X1288/Xbc0	NGC4414	186.612807	31.223649	229.988814	0.223839	58123
148	uid://A001/X1288/Xbc0	NGC4414	186.612807	31.223649	229.988814	0.223839	58123
149	uid://A001/X1288/Xbc0	NGC4414	186.612807	31.223649	229.988814	0.223839	58123
150	uid://A001/X1296/X6f3	5-NGC3773	174.553958	12.111899	229.781272	0.299770	58384
151	uid://A001/X1296/X6f3	5-NGC3773	174.553958	12.111899	229.781272	0.299770	58384
152	uid://A001/X1296/X6f3	5-NGC3773	174.553958	12.111899	229.781272	0.299770	58384
153	uid://A001/X1296/X6f3	5-NGC3773	174.553958	12.111899	229.781272	0.299770	58384
154	uid://A001/X1296/X6f3	5-NGC3773	174.553958	12.111899	229.781272	0.299770	58384
155	uid://A001/X1296/X6f3	5-NGC3773	174.553958	12.111899	229.781272	0.299770	58384
156	uid://A001/X1296/X6f3	5-NGC3773	174.553958	12.111899	229.781272	0.299770	58384
157	uid://A001/X1296/X6f3	5-NGC3773	174.553958	12.111899	229.781272	0.299770	58384
158	uid://A001/X1296/X6f3	5-NGC3773	174.553958	12.111899	229.781272	0.299770	58384
159	uid://A001/X1296/X6f3	5-NGC3773	174.553958	12.111899	229.781272	0.299770	58384
160	uid://A001/X1296/X6f3	5-NGC3773	174.553958	12.111899	229.781272	0.299770	58384

	obs_id	target_name	s_ra	s_dec	frequency	s_resolution	
161	uid://A001/X1296/X6f3	5-NGC3773	174.553958	12.111899	229.781272	0.299770	58384
162	uid://A001/X1296/X6f3	5-NGC3773	174.553958	12.111899	229.781272	0.299770	58384
163	uid://A001/X1296/X6f3	5-NGC3773	174.553958	12.111899	229.781272	0.299770	58384
164	uid://A001/X1296/X6f3	5-NGC3773	174.553958	12.111899	229.781272	0.299770	58384
165	uid://A001/X1296/X6f3	5-NGC3773	174.553958	12.111899	229.781272	0.299770	58384
166	uid://A001/X1296/X6f3	5-NGC3773	174.553958	12.111899	229.781272	0.299770	58384
167	uid://A001/X1296/X6f3	5-NGC3773	174.553958	12.111899	229.781272	0.299770	58384
168	uid://A001/X1296/X6f3	5-NGC3773	174.553958	12.111899	229.781272	0.299770	58384
169	uid://A001/X1296/X6f3	5-NGC3773	174.553958	12.111899	229.781272	0.299770	58384
170	uid://A001/X1296/X6f3	5-NGC3773	174.553958	12.111899	229.781272	0.299770	58384
171	uid://A001/X1296/X6f3	5-NGC3773	174.553958	12.111899	229.781272	0.299770	58384
172	uid://A001/X1296/X6f3	5-NGC3773	174.553958	12.111899	229.781272	0.299770	58384
173	uid://A001/X1296/X6f3	5-NGC3773	174.553958	12.111899	229.781272	0.299770	58384
174	uid://A001/X1296/X6f3	5-NGC3773	174.553958	12.111899	229.781272	0.299770	58384
175	uid://A001/X1296/X6f3	5-NGC3773	174.553958	12.111899	229.781272	0.299770	58384
176	uid://A001/X1296/X6f3	5-NGC3773	174.553958	12.111899	229.781272	0.299770	58384
177	uid://A001/X1296/X6f3	5-NGC3773	174.553958	12.111899	229.781272	0.299770	58384
178	uid://A001/X1296/X6f3	5-NGC3773	174.553958	12.111899	229.781272	0.299770	58384
179	uid://A001/X1296/X6f3	5-NGC3773	174.553958	12.111899	229.781272	0.299770	58384

Plot them on the sky

In [12]:
```
plt.scatter(result['ra'],result['dec'],marker="+")  # note ra,dec = sourc
e; s_ra,s_dec = target
plt.xlabel("Cubesum Source RA (deg)")
plt.ylabel("CubeSum Source DEC (deg)")
```

Out[12]:
```
Text(0, 0.5, 'CubeSum Source DEC (deg)')
```



Find where NGC3504 has lines detected with moment 0 flux greater than one

In [13]:
```
result=a.query(source_name_alma="NGC3504",mom0flux=">1")
result
```

```
select * from alma inner join win on (win.a_id = alma.id)  inner join lin
es on (lines.w_id = win.id )  WHERE alma.target_name='NGC3504' AND lines.
mom0flux>=1.0
```

Out[13]:

	id	obs_id	target_name	s_ra	s_dec	frequency	s_resolution	
0	9	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	243.760018	0.034121	580
1	10	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	580
2	13	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	243.760022	0.200656	581
3	14	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	229.371216	0.200656	581
4	44	uid://A001/X87a/X706	NGC3504	165.796629	27.972436	229.366802	0.565398	577
5	46	uid://A001/X87a/X706	NGC3504	165.796629	27.972436	243.689957	0.565398	577
6	48	uid://A001/X87a/X708	NGC3504	165.796629	27.972436	229.366820	1.782565	578
7	50	uid://A001/X87a/X708	NGC3504	165.796629	27.972436	243.690784	1.782565	578
8	51	uid://A001/X87a/X70a	NGC3504	165.796629	27.973328	229.366824	5.503584	577
9	53	uid://A001/X87a/X70a	NGC3504	165.796629	27.973328	243.689992	5.503584	577

Find sources in NGC3504 with signal to noise ratio>3

In [14]:
```
result=a.query(source_name_alma='NGC3504',source_snr=">3")
print(f'{len(result)} sources found.')
result.head() # first 5
```

```
select * from alma inner join win on (win.a_id = alma.id)  inner join sou
rces on (sources.w_id = win.id)   WHERE alma.target_name='NGC3504' AND so
urces.snr_s>=3.0 AND sources.l_id = 0

130 sources found.
```

Out[14]:

	id	obs_id	target_name	s_ra	s_dec	frequency	s_resolution	
0	10	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	580
1	10	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	580
2	10	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	580
3	10	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	580
4	10	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	580

Find sources in NGC3504 with signal to noise ratio>3 and CO was detected

In [15]:
```
result=a.query(source_name_alma='NGC3504',source_snr=">3",formula="CO")
print(f'{len(result)} sources found.')
result.head() # first 5
```

```
select * from alma inner join win on (win.a_id = alma.id)  inner join sou
rces on (sources.w_id = win.id)   inner join lines on (lines.w_id = win.i
d )  WHERE alma.target_name='NGC3504' AND sources.snr_s>=3.0 AND lines.fo
rmula='CO' AND sources.l_id > 0

97 sources found.
```

Out[15]:

	id	obs_id	target_name	s_ra	s_dec	frequency	s_resolution	
0	10	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	580
1	10	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	580
2	10	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	580
3	10	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	580
4	10	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	580

Sources in NGC5253 with peak flux > 0

In [16]:
```
result=a.query(source_name_alma='NGC5253',source_peak=">0")
print(f'{len(result)} sources found.')
result.head() # first 5
```

```
select * from alma inner join win on (win.a_id = alma.id)  inner join sou
rces on (sources.w_id = win.id)   WHERE alma.target_name='NGC5253' AND so
urces.peak_s>=0.0 AND sources.l_id = 0

70 sources found.
```

Out[16]:

	id	obs_id	target_name	s_ra	s_dec	frequency	s_resolution	
0	28	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	242.420288	0.074072	58(
1	30	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	230.225275	0.074072	58(
2	30	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	230.225275	0.074072	58(
3	30	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	230.225275	0.074072	58(
4	30	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	230.225275	0.074072	58(

Example of how to exclude columns listed in the first cell

In [17]:
```
#columns to include
ci = [x for x in result.columns if x not in cols_to_exclude]
result[ci]
```

Out[17]:

	obs_id	target_name	s_ra	s_dec	frequency	s_resolution	
0	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	242.420288	0.074072	58076.
1	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	230.225275	0.074072	58076.
2	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	230.225275	0.074072	58076.
3	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	230.225275	0.074072	58076.
4	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	230.225275	0.074072	58076.
5	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	230.225275	0.074072	58076.
6	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	230.225275	0.074072	58076.
7	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	230.225275	0.074072	58076.
8	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	230.225275	0.074072	58076.
9	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	230.225275	0.074072	58076.
10	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	230.225275	0.074072	58076.
11	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	230.225275	0.074072	58076.
12	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	230.225275	0.074072	58076.
13	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	230.225275	0.074072	58076.
14	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	230.225275	0.074072	58076.
15	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	230.225275	0.074072	58076.
16	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	230.225275	0.074072	58076.
17	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	230.225275	0.074072	58076.
18	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	230.225275	0.074072	58076.
19	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	230.225275	0.074072	58076.
20	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	230.225275	0.074072	58076.
21	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	230.225281	0.258194	58382.
22	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	230.225281	0.258194	58382.

	obs_id	target_name	s_ra	s_dec	frequency	s_resolution	
23	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	230.225281	0.258194	58382.
24	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	230.225281	0.258194	58382.
25	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	230.225281	0.258194	58382.
26	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	230.225281	0.258194	58382.
27	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	230.225281	0.258194	58382.
28	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	230.225281	0.258194	58382.
29	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	230.225281	0.258194	58382.
30	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	230.225281	0.258194	58382.
31	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	230.225281	0.258194	58382.
32	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	230.225281	0.258194	58382.
33	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	230.225281	0.258194	58382.
34	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	230.225281	0.258194	58382.
35	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	230.225281	0.258194	58382.
36	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	230.225281	0.258194	58382.
37	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	230.225281	0.258194	58382.
38	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	230.225281	0.258194	58382.
39	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	230.225281	0.258194	58382.
40	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	230.225281	0.258194	58382.
41	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	230.225281	0.258194	58382.
42	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	230.225281	0.258194	58382.
43	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	230.225281	0.258194	58382.
44	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	230.225281	0.258194	58382.
45	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	230.225281	0.258194	58382.

	obs_id	target_name	s_ra	s_dec	frequency	s_resolution	
46	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	230.225281	0.258194	58382.
47	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	230.225281	0.258194	58382.
48	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	230.225281	0.258194	58382.
49	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	230.225281	0.258194	58382.
50	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	230.225281	0.258194	58382.
51	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	228.400349	0.074072	58076.
52	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	228.400349	0.074072	58076.
53	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	242.420288	0.074072	58076.
54	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	242.420288	0.074072	58076.
55	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	244.667658	0.074072	58076.
56	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	230.225275	0.074072	58076.
57	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	230.225275	0.074072	58076.
58	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	228.399516	0.258194	58382.
59	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	228.399516	0.258194	58382.
60	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	228.399516	0.258194	58382.
61	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	242.420716	0.258194	58382.
62	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	244.667664	0.258194	58382.
63	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	244.667664	0.258194	58382.
64	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	230.225281	0.258194	58382.
65	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	242.420716	0.258194	58382.
66	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	242.420716	0.258194	58382.
67	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	242.420716	0.258194	58382.
68	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	242.420716	0.258194	58382.

	obs_id	target_name	s_ra	s_dec	frequency	s_resolution	
69	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	242.420716	0.258194	58382.

CS detected in NGC3504

In [18]:
```
result=a.query(source_name_alma='NGC3504',formula="CS")
result
```

```
select * from alma inner join win on (win.a_id = alma.id)  inner join lin
es on (lines.w_id = win.id )  WHERE alma.target_name='NGC3504' AND lines.
formula='CS'
```

Out[18]:

	id	obs_id	target_name	s_ra	s_dec	frequency	s_resolution	
0	9	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	243.760018	0.034121	580
1	13	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	243.760022	0.200656	581
2	46	uid://A001/X87a/X706	NGC3504	165.796629	27.972436	243.689957	0.565398	577
3	50	uid://A001/X87a/X708	NGC3504	165.796629	27.972436	243.690784	1.782565	578
4	53	uid://A001/X87a/X70a	NGC3504	165.796629	27.973328	243.689992	5.503584	577

Find all projects with a particular scientific category.

Note case-sensitive!

In [19]:
```
result=a.query(scientific_category="Galaxy evolution")
lst = set(zip(result['obs_id'],result['target_name']))
print("Found these projects and targets:",
      *('{}: {}'.format(*k[1]) for k in enumerate(lst)), sep="\n")
print(f"The unique targets  are: {result['target_name'].unique()}")
```

```
select * from alma inner join win on (win.a_id = alma.id)  WHERE alma.sci
entific_category='Galaxy evolution'

Found these projects and targets:
uid://A001/X1288/Xba0: NGC3049
uid://A001/X1288/Xba2: NGC3049
uid://A001/X1288/Xbc4: NGC5253
uid://A001/X1288/Xbc0: NGC4414
uid://A001/X1288/Xbc6: NGC5253
uid://A001/X1288/Xba8: NGC3504
uid://A001/X1288/Xbba: NGC4402
uid://A001/X1288/Xbae: NGC3593
uid://A001/X1288/Xbbe: NGC4414
uid://A001/X1296/X6f3: 5-NGC3773
uid://A001/X1296/X6f1: 5-NGC3773
uid://A001/X1288/Xba6: NGC3504
The unique targets  are: ['NGC3049' 'NGC3504' 'NGC3593' 'NGC4402' 'NGC441
4' 'NGC5253' '5-NGC3773']
```

Find results where the Cubesum SNR is between 3 and 10

In [20]:
```
result=a.query(win_snr="3..10",source_name_alma="NGC3049")
```

```
select * from alma inner join win on (win.a_id = alma.id)  WHERE (3.0<=(
win.peak_w / win.rms_w ) AND ( win.peak_w / win.rms_w )<=10.0) AND alma.t
arget_name='NGC3049'
```

In [21]:
```
result
```

Out[21]:

	id	obs_id	target_name	s_ra	s_dec	frequency	s_resolution	
0	1	uid://A001/X1288/Xba0	NGC3049	148.70695	9.271113	227.595275	0.039484	580
1	2	uid://A001/X1288/Xba0	NGC3049	148.70695	9.271113	241.577407	0.039484	580
2	3	uid://A001/X1288/Xba0	NGC3049	148.70695	9.271113	243.826966	0.039484	580
3	4	uid://A001/X1288/Xba2	NGC3049	148.70695	9.271113	241.566693	0.332481	583
4	5	uid://A001/X1288/Xba2	NGC3049	148.70695	9.271113	243.816538	0.332481	583
5	55	uid://A001/X1288/Xba0	NGC3049	148.70695	9.271113	227.595275	0.039484	580
6	56	uid://A001/X1288/Xba0	NGC3049	148.70695	9.271113	241.577407	0.039484	580
7	57	uid://A001/X1288/Xba0	NGC3049	148.70695	9.271113	243.826966	0.039484	580
8	58	uid://A001/X1288/Xba0	NGC3049	148.70695	9.271113	229.411041	0.039484	580
9	59	uid://A001/X1288/Xba2	NGC3049	148.70695	9.271113	227.609012	0.332481	583
10	60	uid://A001/X1288/Xba2	NGC3049	148.70695	9.271113	241.566693	0.332481	583
11	61	uid://A001/X1288/Xba2	NGC3049	148.70695	9.271113	243.816538	0.332481	583
12	62	uid://A001/X1288/Xba2	NGC3049	148.70695	9.271113	229.424396	0.332481	583
13	116	uid://A001/X1288/Xba0	NGC3049	148.70695	9.271113	241.577407	0.039484	580

Science Case #3: find any project where <spectral line(s)> was detected or observed

The science case here would be to search the archive for instances where a rare line was observed and/or detected. Desired lines could be logically ANDed to narrow the results to coincident detections. For example, the search could be for Si18O observations. One could even limit results to be above a certain S/N or line ratio (peak or integrated). This is a straightforward search of ADMIT's line identifications and line strengths. Additional constra ints can of course be given, e.g., a frequency range, or ALMA band. The information returned would allow the user to see what sources were observed, which transitions, { and examine moment maps of the detection }. This same pattern could be used to find sources where a large fraction of the CO-ladder was observed.

In [1]:
```
from astroquery.admit import ADMIT
import pandas as pd
import numpy as np
# display the whole table inthe notebook
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
pd.set_option('display.width', None)
pd.set_option('display.max_colwidth',25)
```

In [2]:
```
a = ADMIT()
```

```
Found  /home/teuben/ALMA/study7/query/admit.db
Checking db.... 0
71 71 71
Database version: 27-feb-2022. core.py version: 26-feb-2022
```

Find any project where any CO transition was detected in a LineCube

In [3]:
```
a.query(formula="CO")
```

```
select * from alma inner join win on (win.a_id = alma.id)  inner join lin
es on (lines.w_id = win.id )  WHERE lines.formula='CO'
```

Out[3]:

	id	obs_id	target_name	s_ra	s_dec	frequency	s_resolution	
0	6	uid://A001/X1288/Xba2	NGC3049	148.706950	9.271113	229.424396	0.332481	5
1	10	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	5
2	14	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	229.371216	0.200656	5
3	15	uid://A001/X1288/Xbae	NGC3593	168.654583	12.817675	230.055098	0.289034	5
4	26	uid://A001/X1288/Xbc0	NGC4414	186.612807	31.223649	229.988814	0.223839	5
5	30	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	230.225275	0.074072	5
6	34	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	230.225281	0.258194	5
7	42	uid://A001/X1296/X6f3	5-NGC3773	174.553958	12.111899	229.781272	0.299770	5
8	44	uid://A001/X87a/X706	NGC3504	165.796629	27.972436	229.366802	0.565398	5
9	48	uid://A001/X87a/X708	NGC3504	165.796629	27.972436	229.366820	1.782565	5
10	51	uid://A001/X87a/X70a	NGC3504	165.796629	27.973328	229.366824	5.503584	5

Find any project where CO(J=2-1) transition was detected in a LineCube

In [4]:
```
a.query(formula="CO",transition="2-1")
```

```
select * from alma inner join win on (win.a_id = alma.id)  inner join lin
es on (lines.w_id = win.id )  WHERE lines.formula='CO' AND lines.transiti
on='2-1'
```

Out[4]:

	id	obs_id	target_name	s_ra	s_dec	frequency	s_resolution	
0	6	uid://A001/X1288/Xba2	NGC3049	148.706950	9.271113	229.424396	0.332481	5
1	10	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	5
2	14	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	229.371216	0.200656	5
3	15	uid://A001/X1288/Xbae	NGC3593	168.654583	12.817675	230.055098	0.289034	5
4	26	uid://A001/X1288/Xbc0	NGC4414	186.612807	31.223649	229.988814	0.223839	5
5	30	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	230.225275	0.074072	5
6	34	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	230.225281	0.258194	5
7	42	uid://A001/X1296/X6f3	5-NGC3773	174.553958	12.111899	229.781272	0.299770	5
8	44	uid://A001/X87a/X706	NGC3504	165.796629	27.972436	229.366802	0.565398	5
9	48	uid://A001/X87a/X708	NGC3504	165.796629	27.972436	229.366820	1.782565	5
10	51	uid://A001/X87a/X70a	NGC3504	165.796629	27.973328	229.366824	5.503584	5

Find any project where any CO or CS transition was detected in a LineCube

In [5]:
```
a.query(formula="CO|CS")
```

```
select * from alma inner join win on (win.a_id = alma.id)  inner join lin
es on (lines.w_id = win.id )  WHERE (lines.formula='CO' OR lines.formula
='CS')
```

Out[5]:

	id	obs_id	target_name	s_ra	s_dec	frequency	s_resolution	
0	6	uid://A001/X1288/Xba2	NGC3049	148.706950	9.271113	229.424396	0.332481	5
1	9	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	243.760018	0.034121	5
2	10	uid://A001/X1288/Xba6	NGC3504	165.796595	27.972404	229.371212	0.034121	5
3	13	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	243.760022	0.200656	5
4	14	uid://A001/X1288/Xba8	NGC3504	165.796595	27.972404	229.371216	0.200656	5
5	15	uid://A001/X1288/Xbae	NGC3593	168.654583	12.817675	230.055098	0.289034	5
6	21	uid://A001/X1288/Xbbe	NGC4414	186.612807	31.223649	244.416354	0.032896	5
7	25	uid://A001/X1288/Xbc0	NGC4414	186.612807	31.223649	244.416362	0.223839	5
8	26	uid://A001/X1288/Xbc0	NGC4414	186.612807	31.223649	229.988814	0.223839	5
9	30	uid://A001/X1288/Xbc4	NGC5253	204.983216	-31.640107	230.225275	0.074072	5
10	34	uid://A001/X1288/Xbc6	NGC5253	204.983216	-31.640107	230.225281	0.258194	5
11	42	uid://A001/X1296/X6f3	5-NGC3773	174.553958	12.111899	229.781272	0.299770	5
12	44	uid://A001/X87a/X706	NGC3504	165.796629	27.972436	229.366802	0.565398	5
13	46	uid://A001/X87a/X706	NGC3504	165.796629	27.972436	243.689957	0.565398	5
14	48	uid://A001/X87a/X708	NGC3504	165.796629	27.972436	229.366820	1.782565	5
15	50	uid://A001/X87a/X708	NGC3504	165.796629	27.972436	243.690784	1.782565	5
16	51	uid://A001/X87a/X70a	NGC3504	165.796629	27.973328	229.366824	5.503584	5

	id	obs_id	target_name	s_ra	s_dec	frequency	s_resolution	
17	53	uid://A001/X87a/X70a	NGC3504	165.796629	27.973328	243.689992	5.503584	5

Sources where both CO and CS have been detected

In [6]:
```
result_co = a.query(formula="CO")
result_cs = a.query(formula="CS")
co_targets = set(result_co['target_name'])
cs_targets = set(result_cs['target_name'])
co_targets.intersection(cs_targets)
```

```
select * from alma inner join win on (win.a_id = alma.id)  inner join lin
es on (lines.w_id = win.id )  WHERE lines.formula='CO'

select * from alma inner join win on (win.a_id = alma.id)  inner join lin
es on (lines.w_id = win.id )  WHERE lines.formula='CS'
```

Out[6]:
```
{'NGC3504', 'NGC4414'}
```

Projects where both CO and CS have been detected

In [7]:
```
co_projects = set(result_co['obs_id'])
cs_projects = set(result_cs['obs_id'])
```

In [8]:
```
co_projects.intersection(cs_projects)
```

Out[8]:
```
{'uid://A001/X1288/Xba6',
 'uid://A001/X1288/Xba8',
 'uid://A001/X1288/Xbc0',
 'uid://A001/X87a/X706',
 'uid://A001/X87a/X708',
 'uid://A001/X87a/X70a'}
```

In [ ]:
```

```

Science Case #4: NGC3504

The galaxy NGC3504 has been observed in two unrelated ALMA projects, both in band 6 at at 230 GHz, cause for an interesting comparison.

1. **2016.1.00650.S** - one 7m and two 12m observations of just NGC3504, to study flow in a bar
2. **2017.1.00964.S** - a collection of 7 galaxies with the purpose of measure gas flow near the central black hole. For NGC3504 two datasets were collected.

Here we are focusing on the commonalities and differences between these two observations.

In [1]:
```
%matplotlib inline
import matplotlib.pyplot as plt
```

In [2]:
```
source = "NGC3504"
```

astroquery.alma

First we should query the science archive, we can do https://almascience.nrao.edu/aq/ (https://almascience.nrao.edu/aq/) as well, but we want also show this via the notebook.

In [3]:
```
from astroquery.alma import Alma
import pandas as pd
# display the whole table in the notebook
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
pd.set_option('display.width', None)
pd.set_option('display.max_colwidth',25)
# more to come here.  here we just want to show how much you can do with t
he current query, before science
alma = Alma()
```

In [4]:
```
n = Alma.query_object(source)
```

In [5]:
```
print(pd.unique(n['proposal_id']))
print(pd.unique(n['obs_id']))
```

```
[b'2017.1.00964.S' b'2016.1.00650.S']
[b'uid://A001/X1288/Xba6' b'uid://A001/X1288/Xba8' b'uid://A001/X87a/X70
a'
 b'uid://A001/X87a/X708' b'uid://A001/X87a/X706']
```

Thus we have indeed two projects,and five observations across those two.

Lets print how many beams we have across the image

In [6]:
```
ci=['obs_id','s_fov','s_resolution']
n['nres'] = 3600*n['s_fov']/n['s_resolution']
print(n[ci])
```

```
        obs_id              s_fov            s_resolution
                             deg                 deg
-------------------- ------------------- --------------------
uid://A001/X1288/Xba6 0.006863718695429476 0.034121085407438106
uid://A001/X1288/Xba6 0.006863718695429476 0.034121085407438106
uid://A001/X1288/Xba6 0.006863718695429476 0.034121085407438106
uid://A001/X1288/Xba6 0.006863718695429476 0.034121085407438106
uid://A001/X1288/Xba8  0.00686371854646795  0.20065563262611036
uid://A001/X1288/Xba8  0.00686371854646795  0.20065563262611036
uid://A001/X1288/Xba8  0.00686371854646795  0.20065563262611036
uid://A001/X1288/Xba8  0.00686371854646795  0.20065563262611036
 uid://A001/X87a/X70a 0.017853666133604822    5.503583966319726
 uid://A001/X87a/X70a 0.017853666133604822    5.503583966319726
 uid://A001/X87a/X70a 0.017853666133604822    5.503583966319726
 uid://A001/X87a/X70a 0.017853666133604822    5.503583966319726
 uid://A001/X87a/X708  0.01305128673735267   1.7825652390462614
 uid://A001/X87a/X708  0.01305128673735267   1.7825652390462614
 uid://A001/X87a/X708  0.01305128673735267   1.7825652390462614
 uid://A001/X87a/X708  0.01305128673735267   1.7825652390462614
 uid://A001/X87a/X706 0.013051297132328152    0.565397979165851
 uid://A001/X87a/X706 0.013051297132328152    0.565397979165851
 uid://A001/X87a/X706 0.013051297132328152    0.565397979165851
 uid://A001/X87a/X706 0.013051297132328152    0.565397979165851
```

Notice there appears to be a units issue with s_resolution: they appear to be in arcsec. There is also a 'spatial_resolution', but it has the same issue.

astroquery.admit

In [7]:
```
from astroquery.admit import ADMIT
import pandas as pd

pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
pd.set_option('display.width', None)
pd.set_option('display.max_colwidth',25)


a = ADMIT()
a.check()
```

```
Found  /home/teuben/ALMA/study7/query/admit.db
Checking db.... 0
71 71 71
Database version: 27-feb-2022. core.py version: 26-feb-2022
header    : 1 entries
alma      : 124 entries
win       : 123 entries
lines     : 33 entries
sources   : 769 entries
```

Continuum

First we want to see if any continuum is detected, so we select all windows with one channel.

In [8]:
```
p = a.query(source_name_alma=source,nchan=1,flux='>0')
print(p.shape)
```

```
select * from alma inner join win on (win.a_id = alma.id)  inner join sou
rces on (sources.w_id = win.id)   WHERE alma.target_name='NGC3504' AND wi
n.nchan=1 AND sources.flux>=0.0 AND sources.l_id = 0

(23, 40)
```

In [9]:
```
a.key_description.show_in_notebook(display_length=20)
```

Out[9]:

Table length=71

Show 20 entries Search:

idx	Keyword	Description	Database Table
0	alma_id	ALMA ID	Window
1	win_id	Spectral window ID	Window
2	spw	Spectral window name	Window
3	nlines	Number of lines	Window
4	nsources	Number of sources	Window
5	nchan	Number of channels	Window
6	win_peak	Window peak flux	Window
7	win_rms	Window RMS noise	Window
8	win_snr	Window Signal to noise ratio (peak/rms)	Window
9	bmaj	Beam major axis	Window
10	bmin	Beam minor axis	Window
11	bpa	Beam PA	Window
12	freqc	Frequency center (GHz)	Window
13	freqw	Frequency width (GHz)	Window
14	vlsr	LSR Velocity (km/s)	Window
15	fcoverage	Frequency coverage?	Window
16	l_win_id	Spectral window ID(L)	Lines
17	restfreq	Rest frequency	Lines
18	formula	Formula	Lines
19	transition	Transition	Lines

Showing 1 to 20 of 71 entries First Previous 1 2 3 4 Next Last

We collect a few observables: observing time, as well as peak and flux and the resolution

Note we need to clean up the units

In [10]:
```
ci=['obs_id','spw','nsources','t_min', 'flux', 'peak_s','fop','bmaj_arcse
c','smaj_arcsec']
p['fop'] = p['flux']/p['peak_s']
p['bmaj_arcsec'] = p['bmaj'] * 3600
p['smaj_arcsec'] = p['smaj'] * 3600
print(p[ci])
```

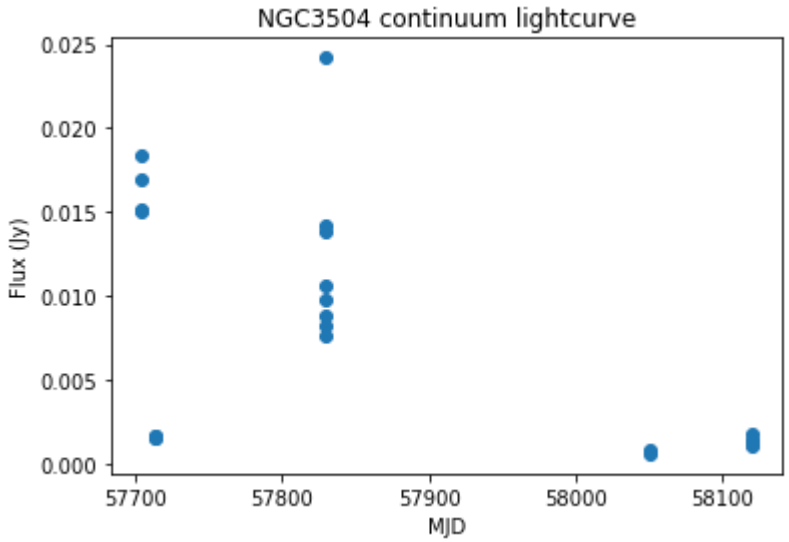
	obs_id	spw	nsources	t_min	flux
0	uid://A001/X1288/Xba6	spw21	1	58050.543677	0.000860
1	uid://A001/X1288/Xba8	spw19	1	58119.297186	0.001110
2	uid://A001/X1288/Xba8	spw21	1	58119.297186	0.001240
3	uid://A001/X1288/Xba8	spw23	1	58119.297186	0.001790
4	uid://A001/X1288/Xba8	spw25	1	58119.297186	0.001450
5	uid://A001/X87a/X706	spw23	1	57713.452563	0.001730
6	uid://A001/X87a/X706	spw25	1	57713.452563	0.001510
7	uid://A001/X87a/X706	spw27	1	57713.452563	0.001560
8	uid://A001/X87a/X706	spw29	1	57713.452563	0.001730
9	uid://A001/X87a/X708	spw23	3	57830.136178	0.007700
10	uid://A001/X87a/X708	spw23	3	57830.136178	0.013800
11	uid://A001/X87a/X708	spw23	3	57830.136178	0.024200
12	uid://A001/X87a/X708	spw25	2	57830.136178	0.010600
13	uid://A001/X87a/X708	spw25	2	57830.136178	0.009770
14	uid://A001/X87a/X708	spw27	2	57830.136178	0.008230
15	uid://A001/X87a/X708	spw27	2	57830.136178	0.014200
16	uid://A001/X87a/X708	spw29	1	57830.136178	0.008840
17	uid://A001/X87a/X70a	spw16	1	57704.450990	0.015200
18	uid://A001/X87a/X70a	spw18	1	57704.450990	0.015000
19	uid://A001/X87a/X70a	spw20	1	57704.450990	0.018400
20	uid://A001/X87a/X70a	spw22	1	57704.450990	0.016900
21	uid://A001/X1288/Xba6	spw19_21_23_25	1	58050.543677	0.000618
22	uid://A001/X1288/Xba8	spw19_21_23_25	1	58119.297186	0.001290

	peak_s	fop	bmaj_arcsec	smaj_arcsec
0	0.000151	5.695364	0.049721	0.0
1	0.000512	2.167969	0.218953	0.0
2	0.000580	2.137931	0.208199	0.0
3	0.000594	3.013468	0.206176	0.0
4	0.000706	2.053824	0.219428	0.0
5	0.001090	1.587156	0.617609	0.0
6	0.000968	1.559917	0.660397	0.0

7	0.001020	1.529412	0.655523	0.0
8	0.001260	1.373016	0.623399	0.0
9	0.002080	3.701923	1.810933	3.6
10	0.002040	6.764706	1.810933	7.2
11	0.001940	12.474227	1.810933	7.2
12	0.002120	5.000000	1.919427	7.2
13	0.001710	5.713450	1.919427	3.6
14	0.002290	3.593886	2.262012	3.6
15	0.002310	6.147186	2.262012	7.2
16	0.002280	3.877193	1.812930	3.6
17	0.009540	1.593291	6.312150	7.2
18	0.009410	1.594049	6.296568	7.2
19	0.010600	1.735849	5.772073	7.2
20	0.009970	1.695085	5.714493	7.2
21	0.000160	3.862500	0.041953	0.0
22	0.000570	2.263158	0.212872	0.0

It's a little surprising that flux/peak is 1.5 for the lowest and highest resolution array data, but there clearly is something very odd about the middle resolution (X708) data.

In [11]:
```
plt.scatter(p['t_min'],p['flux']);
plt.title(source + " continuum lightcurve")
plt.xlabel('MJD')
plt.ylabel('Flux (Jy)');
```



well, the fluxes are somewhat all over the place..... averaging 10-15 mJy.

The other dataset of NGC3504 at mjd > 58000 seems to have lost a lot of flux.

Here is a figure of the continuum source, as seen in the three different ALMA confirmation. Figures have been taking from ADMIT, including where sources were detected. ADMIT was using CASA's ia.findsources()

NGC3504

Spectral Lines

In [12]:
```
p = a.query(source_name_alma=source,nchan='>1',mom0flux='>0')
```

```
select * from alma inner join win on (win.a_id = alma.id)  inner join lin
es on (lines.w_id = win.id )  WHERE alma.target_name='NGC3504' AND win.nc
han>=1.0 AND lines.mom0flux>=0.0
```

In [13]:
```
p = a.query(nchan='>1',mom0flux='>0')
```

```
select * from alma inner join win on (win.a_id = alma.id)  inner join lin
es on (lines.w_id = win.id )  WHERE win.nchan>=1.0 AND lines.mom0flux>=0.
0
```

In [14]:
```
print(p.shape)
print(p.columns)
```

```
(33, 39)
Index(['id', 'obs_id', 'target_name', 's_ra', 's_dec', 'frequency', 't_mi
n',
       'cont_sensitivity_bandwidth', 'sensitivity_10kms', 'project_abstra
ct',
       'obs_title', 'science_keyword', 'scientific_category',
       'proposal_authors', 'id', 'a_id', 'spw', 'freqc', 'freqw', 'vlsr',
       'nlines', 'nsources', 'nchan', 'peak_w', 'rms_w', 'bmaj', 'bmin',
'bpa',
       'fcoverage', 'id', 'w_id', 'formula', 'transition', 'restfreq', 'v
min',
       'vmax', 'mom0flux', 'mom1peak', 'mom2peak'],
      dtype='object')
```

In [15]:
```
ci=['obs_id','spw','restfreq','formula','mom0flux','mom1peak','mom2peak']
ci=['spw','restfreq','formula','mom0flux','mom1peak','vlsr','mom2peak','nl
ines']
print(p[ci])
```

	spw	restfreq	formula	mom0flux	mom1peak	vlsr	mom2peak
0	spw21	243.48293	H2COH+	1863.8500	1457.430	1447.989431	31.018
1	spw23	244.59816	HCOOH	4235.2400	1427.340	1447.989431	31.755
2	spw23	244.59816	HCOOH	313.3270	1420.510	1447.989431	38.154
3	spw23	244.63395	CH3CH2OH	309.1810	1448.800	1447.989431	33.364
4	spw25	230.53800	CO	2698.2200	1475.480	1447.989431	20.864
5	spw23	244.93556	CS	3429.8800	1831.720	1521.106704	33.667
6	spw25	230.53800	CO	3388.7800	1516.190	1521.106704	34.980
7	spw23	244.93556	CS	1090.5600	1549.750	1521.106704	80.493
8	spw25	230.53800	CO	18313.7000	1547.690	1521.106704	58.095
9	spw16	230.53800	CO	11534.2000	639.522	628.000000	70.282
10	spw21	242.49769	CH3COOH	477.7500	241.190	232.006873	37.102
11	spw21	242.50962	CH3COOH	0.0000	0.000	232.006873	0.000
12	spw21	242.90447	CH3CH2CN	4678.6600	755.657	715.580697	32.757
13	spw23	244.93556	CS	1365.7300	969.082	715.580697	47.377
14	spw21	242.90447	CH3CH2CN	1217.1900	751.218	715.580697	29.585
15	spw23	244.93556	CS	405.3630	989.835	715.580697	49.216
16	spw25	230.53800	CO	9306.8000	710.181	715.580697	62.409
17	spw21	242.61847	CH2DCCH	1438.9200	393.771	401.856222	32.588
18	spw21	242.63925	H2NCH2CN	1496.2700	408.513	401.856222	34.565
19	spw25	230.53800	CO	436.5640	387.738	401.856222	10.908
20	spw25	230.53800	CO	894.7710	397.203	401.856222	15.017
21	spw19	228.60363	CH2CHCHO	500.2180	1008.250	978.913173	52.920
22	spw21	243.08765	SO2	833.0270	974.240	978.913173	33.245
23	spw23	244.22213	HCCCH	646.3020	980.356	978.913173	30.189
24	spw21	243.12925	CH2OHCHO	370.0540	991.099	978.913173	31.064
25	spw23	244.23979	CH3CH2CN	264.3150	949.060	978.913173	36.270
26	spw25	230.53800	CO	323.7160	980.293	978.913173	6.998
27	spw25	230.53800	CO	24782.4000	1548.350	1521.106704	60.991

50							
28	spw29	244.93556	CS	140.5190	1483.460	1521.106704	15.873
80							
29	spw25	230.53800	CO	32461.2000	1539.720	1521.106704	59.098
90							
30	spw29	244.93556	CS	107.7540	1538.800	1521.106704	31.050
40							
31	spw16	230.53800	CO	26393.2000	1537.470	1521.106704	40.224
90							
32	spw20	244.93556	CS	91.1055	1544.310	1521.106704	24.968
80							

	nlines
0	1
1	1
2	2
3	2
4	1
5	1
6	1
7	1
8	1
9	1
10	2
11	2
12	1
13	1
14	1
15	1
16	1
17	2
18	2
19	1
20	1
21	1
22	1
23	1
24	1
25	1
26	1
27	1
28	1
29	1
30	1
31	1
32	1

To note here in the current set of tables the mom0flux values are probed at the location where in the CubeSum a source was detected. There is no mom0flux value generated for the CubeSum, this might have been useful here to compare to.

One caveat: the CO line appears to be missing (Xba6,8), whereas the CS line is there.... which turned out to be because the CO cubes crashed in ADMIT (some I/O error).

In [ ]:

In [ ]:

Science Case #5: MAPS: an ALMA large program

The Molecules with ALMA at Planet-forming Scales (MAPS) (http://alma-maps.info) survey, also known as 2018.1.01055.L

See also https://almascience.eso.org/alma-data/lp/MAPS (https://almascience.eso.org/alma-data/lp/MAPS)

Data in **2018.1.01055.L** is about 2TB

In [1]:
```
import pandas as pd
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
pd.set_option('display.width', None)
pd.set_option('display.max_colwidth',25)

from astroquery.admit import ADMIT
```

For now a private database, just for MAPS

In [2]:
```
a = ADMIT('../query/admit_maps.db')
a.check()
```

```
Found  ../query/admit_maps.db
Checking db.... 0
71 71 71
Database version: 27-feb-2022. core.py version: 26-feb-2022
header    : 1 entries
alma      : 368 entries
win       : 335 entries
lines     : 94 entries
sources   : 2403 entries
```

In [3]:
```
p = a.query(nchan='>0')
s = p['target_name'].unique()
print(len(s),' unique sources: ',s)
```

```
select * from alma inner join win on (win.a_id = alma.id)  WHERE win.ncha
n>=0.0

5  unique sources:  ['GM_Aur' 'MWC_480' 'IM_Lup' 'HD163296' 'AS_209']
```

In [4]:
```
p = a.query(nchan='>2',mom0flux='>0')
print(len(p))
f=p['formula'].unique()
print(len(f),'unique lines: ',f)
```

```
select * from alma inner join win on (win.a_id = alma.id)  inner join lin
es on (lines.w_id = win.id )  WHERE win.nchan>=2.0 AND lines.mom0flux>=0.
0

94
10 unique lines:  ['HCO+' 'HCN' 'CCH' 'HC3N' 'C18O' '13CO' 'CO' 'H2CO' 'D
CN' 'HCCCHO']
```

Note that their website reports 19 lines

13CN **13CO** C17O **C18O** C2H c-C3H2 CH3CN CN **CO** CS **DCN** H13CN H13CO+ **H2CO** HC15N **HC3N HCN HCO+** N2D+

Should list VLSR used....

In [5]:
```
p1 = a.query(formula="CO")
```

```
select * from alma inner join win on (win.a_id = alma.id)  inner join lin
es on (lines.w_id = win.id )  WHERE lines.formula='CO'
```

In [6]:
```
ci=['obs_id','target_name','s_resolution', 'vlsr']
print(s)
for s1 in s:
    p1 = a.query(source_name_alma=s1)
    print(s1,len(p1))
```

```
['GM_Aur' 'MWC_480' 'IM_Lup' 'HD163296' 'AS_209']
select * from alma inner join win on (win.a_id = alma.id)  WHERE alma.tar
get_name='GM_Aur'

GM_Aur 57
select * from alma inner join win on (win.a_id = alma.id)  WHERE alma.tar
get_name='MWC_480'

MWC_480 60
select * from alma inner join win on (win.a_id = alma.id)  WHERE alma.tar
get_name='IM_Lup'

IM_Lup 89
select * from alma inner join win on (win.a_id = alma.id)  WHERE alma.tar
get_name='HD163296'

HD163296 75
select * from alma inner join win on (win.a_id = alma.id)  WHERE alma.tar
get_name='AS_209'

AS_209 54
```

In [ ]:

Science Case #6: noise stats: comparing ALMA and ADMIT

Since we have noise estimates for both line and continuum from the ALMA query, we can cross-check that with the value ADMIT obtained from the images. The ALMA values are obtained from the raw data and sensitivity calculator, so really theoretical.

In [1]:
```
from astroquery.admit import ADMIT
import pandas as pd

pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
pd.set_option('display.width', None)
pd.set_option('display.max_colwidth',25)


a = ADMIT()
a.check()
```

```
Found  /home/teuben/ALMA/study7/query/admit.db
Checking db.... 0
71 71 71
Database version: 27-feb-2022. core.py version: 26-feb-2022
header    : 1 entries
alma      : 124 entries
win       : 123 entries
lines     : 33 entries
sources   : 769 entries
```

Line

Let's pick one selected MOUS first, the 7m data for NGC3504. This is where one of the spw's has no signal, the CO and CS line are in two other spw's,and the fourth spw has fewer channels.

In [2]:
```
p1 = a.query(nchan='>2',obs_id='uid://A001/X87a/X70a')
```

```
select * from alma inner join win on (win.a_id = alma.id)  WHERE win.ncha
n>=2.0 AND alma.obs_id='uid://A001/X87a/X70a'
```

In [3]:
```
# indeed there are 4 spw's
# p1
print(p1)
```

```
   id                obs_id target_name        s_ra      s_dec   frequenc
y  \
0  51  uid://A001/X87a/X70a     NGC3504  165.796629  27.973328  229.36682
4
1  52  uid://A001/X87a/X70a     NGC3504  165.796629  27.973328  230.73321
6
2  53  uid://A001/X87a/X70a     NGC3504  165.796629  27.973328  243.68999
2
3  54  uid://A001/X87a/X70a     NGC3504  165.796629  27.973328  245.56481
4

         t_min  cont_sensitivity_bandwidth  sensitivity_10kms  \
0  57704.45099                    0.213557           6.807179
1  57704.45099                    0.213557           6.514320
2  57704.45099                    0.213557           6.754302
3  57704.45099                    0.213557           7.155698

           project_abstract                 obs_title  \
0  We propose to establi...  Pattern Speed in the ...
1  We propose to establi...  Pattern Speed in the ...
2  We propose to establi...  Pattern Speed in the ...
3  We propose to establi...  Pattern Speed in the ...

           science_keyword scientific_category          proposal_authors
id  \
0  Galactic centres/nuclei     Active galaxies  Trejo, Alfonso; Jiang...
51
1  Galactic centres/nuclei     Active galaxies  Trejo, Alfonso; Jiang...
52
2  Galactic centres/nuclei     Active galaxies  Trejo, Alfonso; Jiang...
53
3  Galactic centres/nuclei     Active galaxies  Trejo, Alfonso; Jiang...
54

   a_id    spw       freqc     freqw         vlsr  nlines  nsources  ncha
n  \
0    51  spw16  229.366865  1.867088  1521.106704       1         1    47
8
1    52  spw18  230.733263  1.867089  1521.106704       0         0    47
8
2    53  spw20  243.690031  1.867093  1521.106704       1         1    47
8
3    54  spw22  245.564853  1.843657  1521.106704       0         0    11
8

     peak_w     rms_w      bmaj      bmin        bpa  fcoverage
0  3.110140  0.005382  0.001761  0.001609  66.427072   0.150628
1  0.033664  0.004655  0.001757  0.001602  62.536886   0.000000
2  0.043824  0.005212  0.001621  0.001531  51.806499   0.056485
3  0.007329  0.001668  0.001602  0.001536  48.256371   0.000000
```

Notes

1. alma says cont has rms 0.214 mJy/beam, but this is really over 4 spw's, each 1.9 GHz
2. alma says line10 should have 6.8 mJy.km/s; channels are 4.8 km/s
3. ADMIT measured RMS in the spw's (rms_w) each at about 5 mJy/beam in a 4.8 km/s channel
4. See also https://almascience.eso.org/dataPortal/member.uid___A001_X87a_X70a.qa2_report.html (https://almascience.eso.org/dataPortal/member.uid___A001_X87a_X70a.qa2_report.html)

Continuum

In [4]:
```
p2 = a.query(nchan=1,obs_id='uid://A001/X87a/X70a')
```

```
select * from alma inner join win on (win.a_id = alma.id)  WHERE win.ncha
n=1 AND alma.obs_id='uid://A001/X87a/X70a'
```

In [5]:
```
# there are 4, but there should be 5, the aggregative spw_19_21-23_25 is missing?
# p2
print(p2)
```

```
    id               obs_id target_name        s_ra      s_dec   frequency  \
0  108  uid://A001/X87a/X70a     NGC3504  165.796629  27.973328  229.366824
1  109  uid://A001/X87a/X70a     NGC3504  165.796629  27.973328  230.733216
2  110  uid://A001/X87a/X70a     NGC3504  165.796629  27.973328  243.689992
3  111  uid://A001/X87a/X70a     NGC3504  165.796629  27.973328  245.564814

         t_min  cont_sensitivity_bandwidth  sensitivity_10kms  \
0  57704.45099                    0.213557           6.807179
1  57704.45099                    0.213557           6.514320
2  57704.45099                    0.213557           6.754302
3  57704.45099                    0.213557           7.155698

           project_abstract                 obs_title  \
0  We propose to establi...  Pattern Speed in the ...
1  We propose to establi...  Pattern Speed in the ...
2  We propose to establi...  Pattern Speed in the ...
3  We propose to establi...  Pattern Speed in the ...

           science_keyword scientific_category          proposal_authors  id  \
0  Galactic centres/nuclei     Active galaxies  Trejo, Alfonso; Jiang...  108
1  Galactic centres/nuclei     Active galaxies  Trejo, Alfonso; Jiang...  109
2  Galactic centres/nuclei     Active galaxies  Trejo, Alfonso; Jiang...  110
3  Galactic centres/nuclei     Active galaxies  Trejo, Alfonso; Jiang...  111

   a_id    spw       freqc     freqw  vlsr  nlines  nsources  nchan    peak_w  \
0   108  spw16  229.370762  1.843670   0.0       0         1      1  0.009543
1   109  spw18  230.735202  1.831962   0.0       0         1      1  0.009410
2   110  spw20  243.684167  1.839761   0.0       0         1      1  0.010758
3   111  spw22  245.564849  1.781170   0.0       0         1      1  0.010040

      rms_w      bmaj      bmin        bpa  fcoverage
0  0.000363  0.001753  0.001600  64.721130        0.0
1  0.000318  0.001749  0.001591  61.463028        0.0
2  0.000435  0.001603  0.001523  53.963772        0.0
3  0.000356  0.001587  0.001525  47.179459        0.0
```

In [6]:
```
#  rough guess of what the aggregate cont rms would be from the 4 spw's
p2['rms_w'].mean()/2
```

Out[6]:
```
0.0001840875
```

Notes

1. alma claims cont sens is 0.213 mJy (but over 4 spw's)
2. admit measures about 0.37 per spw, which would imply about 0.18 for the aggregrate. A bit lower than what alma claimed.
3. Note that for this continuum record, the line sensitivity is still reported because they share the same MOUS.

All data?

The obvious thing to try is do this for all data, and get a ratio of what ADMIT measured and what ALMA claims. This would be a good check on any remaining problems in the ADMIT results. We make two separate DataFrame's for continuum and line.

In [7]:
```
p2c = a.query(nchan=1)
p2l = a.query(nchan='>2')
```

```
select * from alma inner join win on (win.a_id = alma.id)  WHERE win.ncha
n=1

select * from alma inner join win on (win.a_id = alma.id)  WHERE win.ncha
n>=2.0
```

And showing the tail end of both:

In [8]:
```
ci=['cont_sensitivity_bandwidth','rms_w','nchan']

p2c[ci].tail()
```

Out[8]:

	cont_sensitivity_bandwidth	rms_w	nchan
64	0.021794	0.000028	1
65	0.038697	0.000042	1
66	0.017613	0.000025	1
67	0.013186	0.000014	1
68	0.015109	0.000021	1

In [9]:
```
p2l[ci].tail()
```

Out[9]:

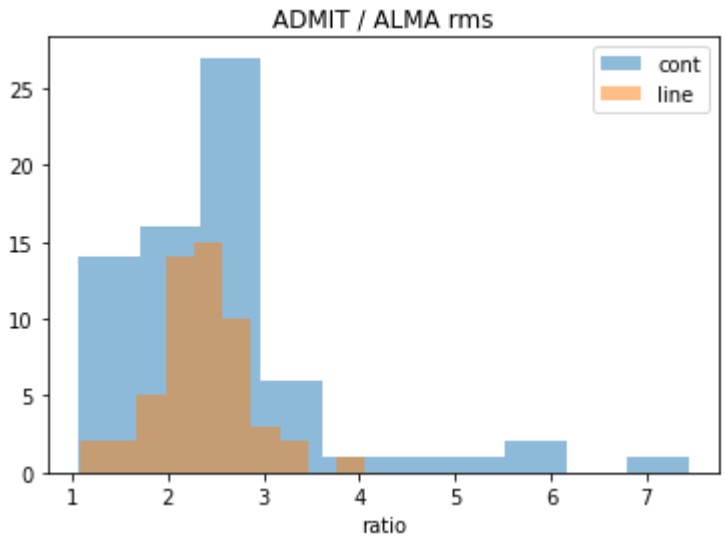
	cont_sensitivity_bandwidth	rms_w	nchan
49	0.062059	0.002596	478
50	0.213557	0.005382	478
51	0.213557	0.004655	478
52	0.213557	0.005212	478
53	0.213557	0.001668	118

Taking the ratio, and recalling that one is in Jy the other mJy....

In [10]:
```
ratio_c = 1000*p2c['rms_w']/p2c['cont_sensitivity_bandwidth']/np.sqrt(p2c[
'nchan'])
ratio_l = 1000*p2l['rms_w']/p2l['cont_sensitivity_bandwidth']/np.sqrt(p2l[
'nchan'])
# hanning factor
ratio_l = ratio_l / 0.612
print("ratio_c:",ratio_c.min(),ratio_c.max(),ratio_c.mean())
print("ratio_l:",ratio_l.min(),ratio_l.max(),ratio_l.mean())
```

```
ratio_c: 1.0650495121132497 7.434821768271884 2.5360411286504574
ratio_l: 1.0767907288171514 4.062892204315444 2.3803743775390096
```

In [11]:
```
plt.hist(ratio_c,label="cont",alpha=0.5)
plt.hist(ratio_l,label="line",alpha=0.5)
plt.xlabel("ratio")
plt.title("ADMIT / ALMA rms")
#plt.xlim([0,4])
plt.legend();
```



The ratio should be 2, if a MOUS has 4 spw's. For this we need the number of identical obsid's. It certainly seems that the cubes measure a lower rms than expected from the ALMA records. This is probably the hanning smoothing, since we used $\sqrt{N_{chan}}$. This is a factor sqrt(3/8) ~ 0.612

Science Case #7: Lightcurve (of a calibrator)

The phase and bandpass calibrators are also included in the product tree of the pipeline results. We developed a alternate ingestion of putting the calibrators in another database with the same schema. This could the be queried much like the science data, except there is only continuum available. Stacking of calibrator maps has already given interesting results (Zwaan, priv.comm.) The use of this notebook is not scientific, as we have no control over how the flux scale was determined (usually via a planet or a moon of a planet). With that caveat we are going to plot the flux as function of time for the calibrators we have available.

See also https://almascience.eso.org/alma-data/calibrator-catalogue (https://almascience.eso.org/alma-data/calibrator-catalogue)

Note: We also check the flux of the continuum source in NGC3504 in Science Case 4 as we have a few measurements in two unrelated projects

In [1]:
```
%matplotlib inline
import matplotlib.pyplot as plt
```

In [2]:
```
from astroquery.admit import ADMIT
import pandas as pd
a = ADMIT('../admit_cal.db')
a.check()
```

```
Found  ../admit_cal.db
Checking db.... 0
71 71 71
Database version: 27-feb-2022. core.py version: 26-feb-2022
header    : 1 entries
alma      : 123 entries
win       : 123 entries
lines     : 0 entries
sources   : 123 entries
```

First we need to find the unique names of the sources. This is probably easiest done with a special **SELECT DISTINCT** query in sql. We find 18 sources.

In [3]:
```
p=pd.DataFrame(a.sql('select distinct alma.target_name from alma'))
print(p)
```

```
             0
0   J1002+1216
1   J1102+2757
2   J1118+1234
3   J1215+1654
4   J1221+2813
5   J1336-3357
6   J1130+0846
7   J1103+3014
8   J1150+2417
9   J1159+2914
10  J1744-3116
11  J1058+0133
12  J0854+2006
13  J1037-2934
14  J1337-1257
15  J1229+0203
16  J1427-4206
17  J1924-2914
```

Let's just pick a random one, and review the fluxes, and plot fluxes as function of time (ALMA keeps the MJD in the database. There are functions to change these back to a human readable time, but we leave this is as an exercize for the reader! NOTE: we might need to ask for if flux > 0 or so, to trigger the source column we need for this.

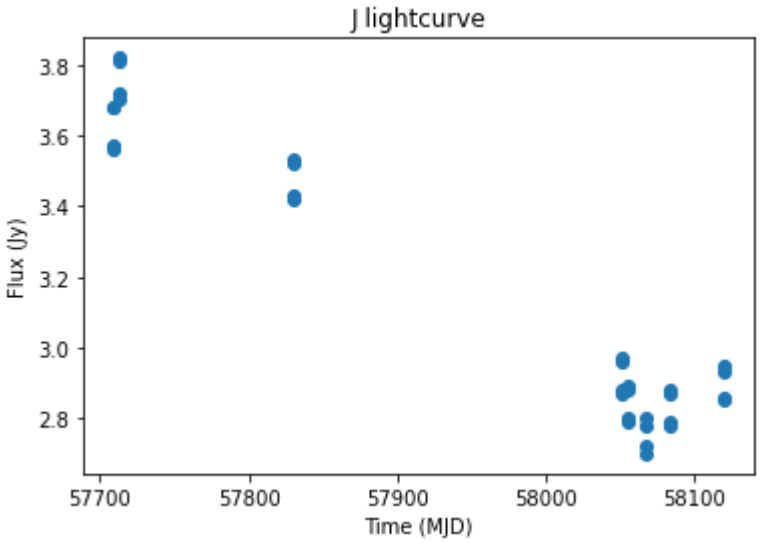
In [4]:
```
source = "J1058+0133"
t1 = a.query(source_name_alma=source, flux='>0')
```

```
select * from alma inner join win on (win.a_id = alma.id)  inner join sou
rces on (sources.w_id = win.id)   WHERE alma.target_name='J1058+0133' AND
sources.flux>=0.0 AND sources.l_id = 0
```

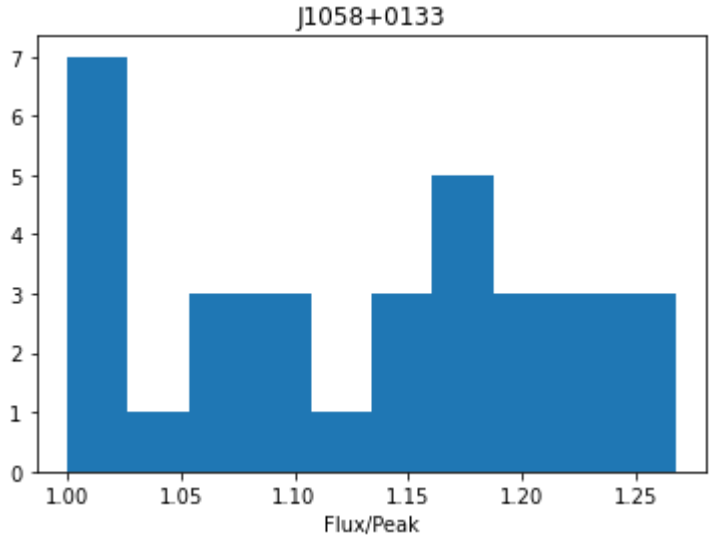
In [5]:
```
ci=['t_min', 'flux', 'peak_s']
print(t1[ci])
```

```
          t_min  flux  peak_s
0   58067.381452  2.80    2.26
1   58067.381452  2.72    2.15
2   58067.381452  2.70    2.13
3   58067.381452  2.78    2.24
4   58050.537257  2.97    2.97
5   58050.537257  2.88    2.88
6   58050.537257  2.87    2.87
7   58050.537257  2.96    2.96
8   58119.290160  2.95    2.47
9   58119.290160  2.86    2.35
10  58119.290160  2.85    2.33
11  58119.290160  2.93    2.45
12  58055.534294  2.89    2.47
13  58055.534294  2.80    2.36
14  58055.534294  2.79    2.34
15  58055.534294  2.88    2.46
16  58083.495681  2.88    2.50
17  58083.495681  2.79    2.40
18  58083.495681  2.78    2.40
19  58083.495681  2.87    2.49
20  57713.447599  3.70    3.42
21  57713.447599  3.82    3.56
22  57713.447599  3.81    3.55
23  57713.447599  3.72    3.16
24  57830.124871  3.42    3.42
25  57830.124871  3.53    3.52
26  57830.124871  3.52    3.52
27  57830.124871  3.43    3.18
28  57709.478662  3.68    3.36
29  57709.478662  3.68    3.36
30  57709.478662  3.57    3.22
31  57709.478662  3.56    3.40
```

In [6]:
```
plt.scatter(t1['t_min'],t1['flux']);
plt.title(source[0] + ' lightcurve')
plt.xlabel('Time (MJD)')
plt.ylabel('Flux (Jy)');
```



In [7]:
```
fop = t1['flux']/t1['peak_s']
plt.hist(fop)
plt.xlabel('Flux/Peak')
plt.title(source);
```



With a quick plotting function we can easily loop over all the sources and be bedazzled?

In [8]:
```
def qplot(a, source):
    """ plot flux(time) and histogram(flux/peak)"""
    t1 = a.query(source_name_alma=source,flux='>0')
    fop = t1['flux']/t1['peak_s']
    plt.scatter(t1['t_min'],t1['flux']);
    plt.title(source)
    plt.xlabel('MJD')
    plt.ylabel('Flux (Jy)')
    plt.show()
    plt.hist(fop)
    plt.xlabel('Flux/Peak')
    plt.title(source);
```

In [9]:
```
for source in p.values:
    print('====',source[0],'====')
    # qplot(a,source[0])
```

```
==== J1002+1216 ====
==== J1102+2757 ====
==== J1118+1234 ====
==== J1215+1654 ====
==== J1221+2813 ====
==== J1336-3357 ====
==== J1130+0846 ====
==== J1103+3014 ====
==== J1150+2417 ====
==== J1159+2914 ====
==== J1744-3116 ====
==== J1058+0133 ====
==== J0854+2006 ====
==== J1037-2934 ====
==== J1337-1257 ====
==== J1229+0203 ====
==== J1427-4206 ====
==== J1924-2914 ====
```

Comparing with ALMA calibrator catalog

On https://almascience.nrao.edu/sc/# (https://almascience.nrao.edu/sc/#) we pick J1058+0133, set band=6, and pick dates between 2014 and 2022. Download the table as "csv" and plot and compare with what we obtained via admit.

Sadly the "csv" file could not be read by astropy.table, but one quick awk produced a table that was readable.

In [10]:
```
!grep -v ^# alma_sourcecat_searchresults.csv | awk -F, '{print $3,$10,$1
1}' > cal.tab
```

In [11]:
```
from astropy.table import Table
t = Table.read('cal.tab', format="ascii")
```

In [12]:
```
print(t)
```

```
        col1          col2  col3
------------------- ----- -----
2021-08-01T18:42:30Z  1.84  0.08
2021-05-19T00:52:12Z   2.2  0.06
2021-04-17T01:30:20Z  2.19  0.05
2021-04-10T02:53:50Z   2.2  0.05
2021-04-08T01:51:55Z  2.08  0.06
2021-04-07T22:50:25Z  2.08  0.08
2021-04-06T04:00:28Z  2.14  0.06
2021-03-25T22:22:51Z  2.03  0.05
2021-03-18T00:00:00Z  2.01  0.06
2021-02-25T03:24:21Z  2.02  0.05
                 ...   ...   ...
2015-01-18T00:00:00Z  3.23   0.1
2014-12-01T00:00:00Z  2.96  0.13
2014-07-26T00:00:00Z  2.64  0.09
2014-06-11T00:00:00Z 2.315 0.185
2014-05-03T00:00:00Z  2.72  0.06
2014-04-28T00:00:00Z  2.54  0.06
2014-03-09T00:00:00Z  3.25  0.15
2014-03-09T00:00:00Z 2.938 0.206
2013-12-21T00:00:00Z 1.815 0.113
2012-10-06T00:00:00Z  1.58  0.08
2012-05-03T00:00:00Z  2.29  0.12
Length = 50 rows
```

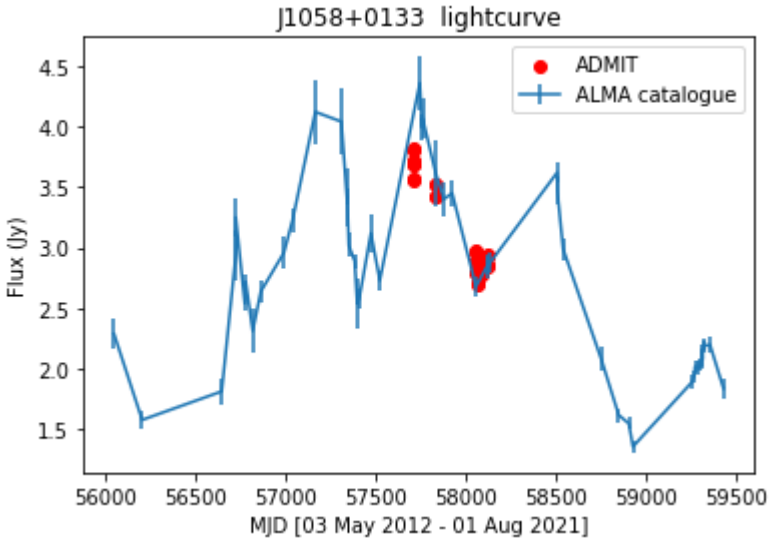
The first column is in standard DATE-OBS format, since we have our earlier data mdj, that's the easiest to compare

In [13]:
```
from astropy.time import Time
```

In [14]:
```
tcal = Time(t['col1']).mjd
#    also grab the min and max time for plotting in the label
tcal_min = tcal.min()
tcal_max = tcal.max()

u0=Time(tcal_min, format='mjd')
u1=Time(tcal_max, format='mjd')
```

In [15]:
```
plt.errorbar(tcal,t['col2'],t['col3'],label="ALMA catalogue")
plt.scatter(t1['t_min'],t1['flux'],c='red',label='ADMIT')
plt.xlabel('MJD [%s - %s]' % (u0.strftime('%d %b %Y'),u1.strftime('%d %b %
Y')))
plt.ylabel('Flux (Jy)')
plt.title('J1058+0133  lightcurve');
plt.legend();
```



well, the 2nd and 3rd clump of data seem to agree quite well, but the first clump seems to deviate.