



Development Upgrades of the Atacama Large Millimeter/submillimeter Array (ALMA)

Study Proposal

Science Mining the ALMA Archive

PRINCIPAL INVESTIGATOR: **PETER TEUBEN**

INSTITUTION: ASTRONOMY DEPARTMENT

ADDRESS: UNIVERSITY OF MARYLAND, COLLEGE PARK, MD 20742

PI CONTACT INFORMATION:

Telephone Number 301-405-1540

Email Address teuben@astro.umd.edu

ABSTRACT

We propose a study to create a prototype Science Query Database that enables broad science-driven queries of ALMA projects. Our goal is to enable science discovery with ALMA archival data by enhancing users' ability to identify, access, and examine relevant data sets through database access to scientific and observational metadata. This study proposes a design and prototype implementation as a pathfinder for a full ALMA implementation.

We will design and construct a Science Query Database on an Amazon Web Services (AWS) testbed using selected public Cycle 5 data. We will image, as necessary, and run the ALMA Data Mining Toolkit (ADMIT) on full projects to create a standard set of science products, and ingest the ADMIT science metadata (e.g., line identifications, line characteristics, source intensities, image statistics, source coordinates) into the Science Query Database. We will merge these metadata with metadata harvested from the ALMA Science Archive system and the u,v and image data files. Combining these with the existing archive interface capability of searching project abstracts and science keywords will allow investigators to make queries that dig through the data rich archive to facilitate new science and explore new ideas.

We will create a new AstroQueryLite Python package to showcase how this implementation can be integrated into many user environments. We will use remote Jupyter notebooks for our study, which are familiar to many astronomers. The outcomes of this study will be: the design framework for including science metadata in future ALMA archive upgrades, a prototype implementation of a Science Query Database and associated access tools, and a test of the viability of AWS as an archival database server for public use.

CONTENTS

1.0 CO-INVESTIGATOR(S) AND COLLABORATING INSTITUTION(S).....	1
2.0 SUBCONTRACTORS.....	1
3.0 SCIENCE CASE.....	1
4.0 STUDY SCOPE.....	3
4.1 Objectives.....	3
4.2 Approaches.....	4
4.2.1 Overview of ADMIT.....	4
4.2.2 Determining the Optimal Metadata.....	6
4.2.3 Creating the Science Query Database.....	7
4.2.4 A Simple User Interface.....	9
4.2.5 Amazon Web Services.....	10
4.3 Work Plan.....	11
5.0 STUDY DELIVERABLES.....	13
5.1 Hardware.....	13
5.2 Software.....	13
5.3 Services.....	13
5.4 Documents.....	13
6.0 INTERFACES TO ALMA.....	13
7.0 SITE LOCATION IMPACT STATEMENT (IF APPLICABLE).....	13
8.0 PERIOD OF PERFORMANCE.....	13
9.0 STAFFING.....	14
9.1 Offerer’s Staffing.....	14
9.2 External Staffing (if applicable).....	14
10.0 STUDY SCHEDULE.....	15
11.0 STUDY MANAGEMENT.....	15
11.1 Systems/Configuration Control.....	15
11.1.1 Systems Requirement and Specification Control.....	15
11.1.2 Documentation Control.....	15
11.1.3 Product & Quality Assurance Control.....	15
11.2 Performance to Schedule.....	16
11.3 Performance to Budget.....	16
11.4 Measures of Success.....	16

11.5 Risk Management.....	16
11.6 Communication Plan and Progress Reporting.....	17
12.0 STUDY CLOSEOUT.....	17
13.0 COMMITMENT.....	17
APPENDIX A - REFERENCE DOCUMENTS.....	18

1.0 Co-Investigator(s) and Collaborating Institution(s)

Table 1.0: Co-Investigator(s) and Collaborating Institution(s).

NAME	INSTITUTION	EMAIL	TELEPHONE
Marc Pound	University of Maryland	mpound@umd.edu	301-405-1520
Lee Mundy	University of Maryland	lgm@astro.umd.edu	301-405-1529

2.0 Subcontractors

None.

3.0 Science Case

We propose to investigate improvements to the ALMA Science Archive metadata and query methodology that will dramatically increase its potential for **science discovery**. The archive interface and metadata as currently implemented are good for certain types of queries, for instance finding a specific source or project. It is less effective for doing science-driven queries like: “What protostars in Taurus have been detected in CO?” or “What observations have been made of CH₃CN?” More importantly, the current archive is not capable of giving the user immediate feedback on the science content of the data. This immediate feedback is vital to effective use of the archive’s scientific potential.

The goals of this study are to design and implement a pathfinder tool which enables science driven queries which return immediate science data products to the user for further exploration. The science data products, both metadata and pre-made images, will be available through queries to a Science Query Database that we will create. Results from queries will be formatted and displayed for the user in an accessible format. Based on their query results, the user will have an overview of the science content and links to detailed science products and the basic data themselves.

Our pathfinder implementation will utilize the ALMA Data Mining Toolkit (ADMIT; <http://admit.astro.umd.edu>), previously developed by our team under an ALMA Development Proposal, to create scientifically useful metadata and images from the data cubes produced by archive pipeline imaging scripts (see e.g. Friedel et al. 2014 and Teuben et al. 2015). ADMIT can automatically produce source lists, noise statistics, line identifications, line strengths and widths, position-velocity diagrams, and moment maps. All have harvestable, quantitative data that can be turned into searchable metadata. Combining these with the existing archive interface capability of searching project abstracts and science keywords will allow investigators to make queries leading to broader range of science outcomes. Below we describe some example science cases that can be enabled by our proposed work.

Science Case #1: 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 and 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.

Science Case #2: 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 constraints 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 detections. This same pattern could be used to find sources where a large fraction of the CO-ladder was observed.

Science Case #3: Find all ALMA continuum detections of a selected source to construct a light curve or a spectral energy distribution.

The keyword search here would be on an RA/Dec, source name, and continuum data. Optionally the user could restrict the search to an ALMA band or frequency range. The returned information would be a table of qualifying observations. The user would sub-select the desired data and pull the desired science information into a table which can be manipulated in Python. The images of the continuum for each dataset can be viewed to verify data quality. We also include calibrator data, which are particularly interesting to monitor flux as function of time and/or frequency.

Science Case #4: Gather all (or selected) data on a specified source

The user can search on a source name (resolved to RA-Dec by SIMBAD) or an RA-Dec location. Since the database has access to source detections from the ADMIT science products,

the search scope would include any detected source in an observed field. The results of the search will be summarized in a tabular format. The user can ask for concatenation of the ADMIT science products for all observations and thumb through web pages to view moment maps and sample spectra. There are pointers to the ADMIT FITS products, and to the ALMA archive u,v data and pipeline data cubes.

These, and more, are the types of science-based queries that we would like to see supported by the ALMA Science Archive in the future. As ALMA continues to improve its pipeline archive images, and expands the archive to include key (large) projects and user curated science images, the immediate science potential will grow exponentially. We are proposing here a pathfinder study that would prototype the structure for such a system. Due to the limited nature of study proposals, we will create a curated mini-archive of 30-60 Cycle 5 publicly available projects as our “archive.” However, the design and implementation will be done with the goal of providing a working prototype that can serve as a starting point for a future ALMA implementation.

Why is the proposed system uniquely different from the current ALMA archive access, or simple extrapolations of the current archive? The key differences are:

- 1 the creation of a wide range of science metadata and the inclusion of those metadata in the searchable database, and
- 2 Immediate access to scientific results in the ALMA datacubes

Immediate access and the ability to view, to easily “run your fingers through the data,” enable the discovery of new science and strongly enhance the mining of meaningful science for the large and rich ALMA archive, as we will show in the next Section.

4.0 Study Scope

4.1 Objectives

The technical goal of this study is to produce a working prototype system that enables queries into the ALMA archive based on the combination of observational and scientific metadata, and returns to the user quantitative information about the data and images for immediate evaluation and scientific use. The system will be SQL, Python, and Jupyter notebook based, compatible with CASA, and designed as a phase-zero implementation that can serve as a pathfinder for a future full implementation by ALMA. We intend to show: the workability of creating and utilizing a rich science metadata environment, the new user views into the archive enabled by this environment, and the high value of the system in optimizing the scientific return from ALMA.

The full ALMA archive implementation of science-based queries assumes that the archive is populated by full, uniform science-quality data cubes for all observations. This has been a stated goal of ALMA from its inception and continues to be the goal of the Observatory. We assume that this archive level will exist in the future; but it does not yet exist for publicly released data. For this study, we will therefore create a mini-archive of 30-60 Cycle 5 publicly available projects with complete, curated data cubes. The prototype Science Query Database system will be implemented for the mini-archive, and the scalability to the entire archive in a future implementation will be evaluated.

Support for queries based on science results, like whether a line was detected, or a combination of a spatial resolution and a given sensitivity was achieved, is one of our objectives; but it is not possible by simply imaging the uv data into data cubes. It requires sophisticated processing such as that offered by the ALMA Data Mining Toolkit (ADMIT) . We developed ADMIT in 2014-2016 under the ALMA Development Program and delivered it to NRAO upon completion. The ALMA Pipeline group has committed to installing ADMIT as an add-on package that will run on all pipeline-produced ALMA science images to automatically create enhanced data products for ingestion into the ALMA Science Archive. ADMIT produces science products which include: cube noise statistics, molecular line identifications, moment maps of lines, identification of emission peaks in continuum, line maps, and more. All of these science data are available as tables or images and ADMIT produces a webpage that displays the products. We will produce ADMIT data products for the mini-archive and utilize their scientific metadata.

The scope of our study includes: definition of a scientific metadata dictionary, design and implementation of a database and query system integrated in a Python environment, tools for harvesting metadata to support queries, and simple tools for displaying results of queries. The sections that follow present an overview of ADMIT, details of our approach and implementation, a description of the cloud computing resources we will use, and our work plan.

4.2 Approaches

4.2.1 Overview of ADMIT

The ALMA Data-Mining Toolkit is an execution environment and set of tools for analyzing image data cubes. ADMIT is based on Python and designed to be fully compliant with CASA and to utilize CASA routines where possible. ADMIT has a flow-oriented approach, executing a series of ADMIT Tasks in a sequence to produce a set of science data products (see Figure 1). ADMIT can be driven by simple scripts that can be run at the Unix level or from inside of CASA. ADMIT provides a simple browser interface for looking at the data products (much like the QA2 weblog), and all major data products are on disk as CASA images and graphics files. For the advanced user, ADMIT is a Python environment for data analysis and for creating new tools for analyzing data. The primary operations supported by ADMIT are focused on analysis of images commonly produced by ALMA and similar radio telescopes.

In ADMIT, we designed and implemented a robust, comprehensive, and well-documented Python package for science that integrates with CASA and will be included in a future CASA release. We worked closely with the CASA team, NRAO and ALMA personnel to ensure that ADMIT code and data products would integrate well into CASA, the ALMA Pipeline, and the ALMA Science Archive. For the purposes of this proposal, a standard ADMIT script will be run on all data cubes to enable quality evaluation of the imaging products and to produce science metadata. We understand well the computing resources needed to run ADMIT on data cubes of all sizes (Figure 2), and have scoped our work here accordingly.

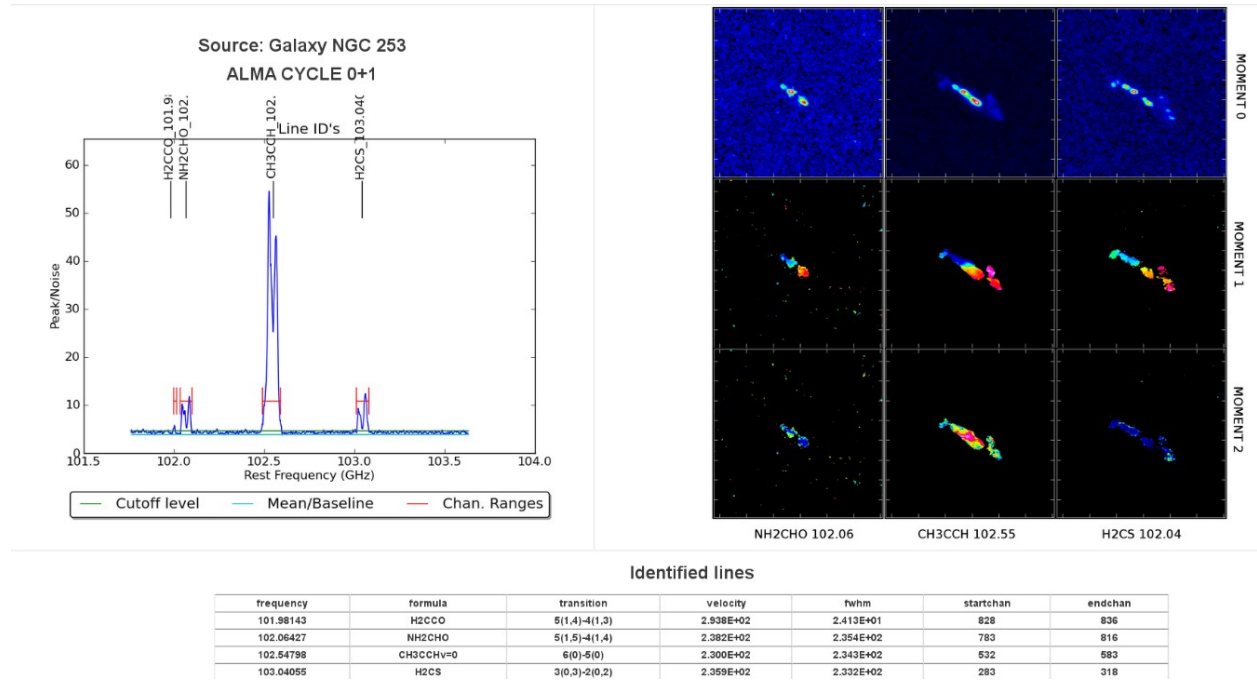
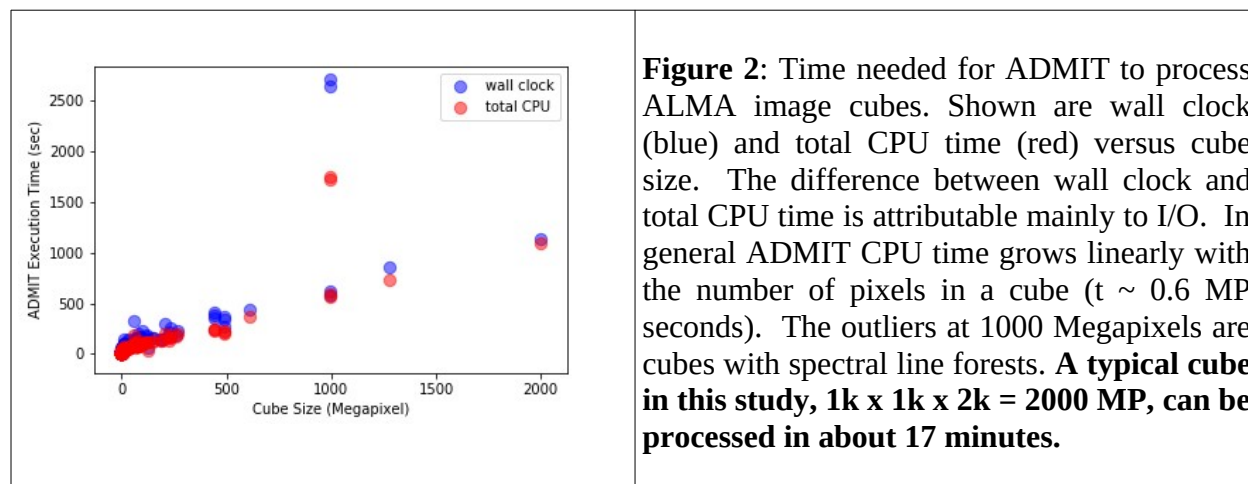


Figure 1: Examples of science data products from ADMIT: The left figure shows a spectrum in blue with the intervals of detected emission indicated by the red bars; the line identifications are shown along the top. The right figure shows the zero, first, and second moment maps for three lines. The first and second moment maps are auto-clipped to minimize noise in areas with low signal in the zero moment map. The table at the bottom show information about the lines detected. All of these data products are automatically produced by ADMIT with no human intervention and can be captured as metadata for queries.



4.2.2 Determining the Optimal Metadata

One aspect of this project is determining which metadata will provide the most utility in science-driven searches. We anticipate harvesting metadata from u,v measurement sets, data cubes, and ADMIT science products. The science cases in Section 3 provide a first guide toward defining the metadata set. During the course of the study, we will define additional science cases. Below we outline broad science categories and the type of metadata required for the listed cases.

Science	Required Metadata
Source type search	sky coordinates, continuum or line intensity, source type (from abstract text, science keywords, or SIMBAD), error estimates based on image statistics
Coverage of data	correlator settings, u,v coverage, pointing centers
Excitation Studies	Line and transition identification, peak intensity, total intensity, error estimates, moment maps
Mass Spectra	Source count, source size, source type, continuum or line total intensity, line width, moment maps, error estimates
Light Curves	sky coordinates or source name, source type, continuum frequency and intensity, ALMA band, observation date/time, error estimates
SEDs	sky coordinates or source name, source type, continuum frequency and intensity, error estimates
Rotation Curves	PV diagram, source size, source type, moment maps
Astrochemistry	line and transition identification, line profiles, line width, peak intensity, total intensity, error estimates

The product from this activity will be the defined set of metadata which will go into the Science Query Database to enable searches.

4.2.3 Creating the Science Query Database

Creating the Science Query Database to support science-driven queries requires a number of steps because the metadata that will populate the database come from different sources: the ALMA measurement sets (MS) and pipeline spectral and continuum cubes, the ALMA Archive database (which is separate from the ALMA data), and ADMIT.

The first thing we do is fetch the information from ALMA archive. We will use the Python package AstroQuery (AQ) [<https://astroquery.readthedocs.io>] to gather initial metadata about the selected projects from the ALMA Archive, and then to download the actual project data. When an astronomer uses the standard web interface (<http://almascience.nrao.edu/aq>) to query the ALMA archive, the response is the Results Table, which has metadata columns describing the projects (e.g., Project code, Source name, RA, Dec, Angular Resolution, abstract text). We refer to these as Results metadata. The only way to programmatically get the Results metadata is through AstroQuery; except for a few columns they do not exist in the MS or data cubes. AstroQuery returns a handle to a project object which contain the Results metadata and associated Python methods to subsequently download the ALMA data (ASDM, MS, and pipeline images). We anticipate that a significant fraction of the projects will require re-imaging to create full cubes and we have costed for that in our computing budget. The NRAO/NA ALMA is planning to have an automated service providing calibrated MS files working this summer (Jeff Kern, private communication); we will utilize this service when available.

Once we have the Results metadata, we can initialize a *local* relational database that forms the root table of our Science Query Database. The Results metadata is not sufficient for sophisticated science queries, so we must build additional database tables, using ADMIT and CASA to extract metadata. These three database tables are: Spectral Window Table, Source Table, and Line Table. They are related to each other and the Results (root) table via indices as shown in Figure 3. Note in Figure 3, some of the metadata are shown as PNG images produced by ADMIT. In reality what would be stored would be a link to the PNG image file on the filesystem. However, as noted below, the display of the table in a Jupyter notebook can actually include the thumbnail images.

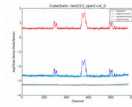
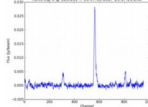
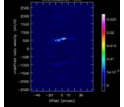
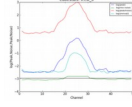
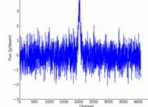
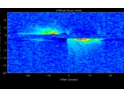
After ADMIT has been run on the images, we append a number of new columns to Results Table, such as the spectral windows present in each Project Code and number of ADMIT Tasks run. Other ADMIT results are added to the three new database tables. The Source and Line Tables are essentially already produced by ADMIT, but the Spectral Window Table is new, and makes the connection between the Results Table and ADMIT products (Figure 3). These four database tables for the base for science-driven queries.

For the database management, we will use SQLite [<https://www.sqlite.org>], a mature software package with a good Python interface. To access the Science Query Database, we will write new lightweight Python package called AQLite (AstroQuery + SQLite), which at the highest level will be compatible with AQ.

Combined ALMA and ADMIT Results Table

ALMA Science Archive Results						ADMIT Results		
index	Project Code	Source Name	RA	Dec	...	SPW	Tasks	...
0	2017.1.005.S	CenA	12:34:56	01:02:03	...	23,25,27,29	6	...
1	2018.3.040.S	NGC 1234	16:17:18	-19:20:21	...	1,2,3,4,5,6	8	...

Spectral Window Table

index	ALMA index	SPW	# lines	# sources	CubeStats	CubeSpectrum	PV Slice	...
10	0	23	4	10				...
11	0	25	1	1				...

Line Table

index	Spectral Win. index	Frequency	Formula	Transition	velocity	startchan	endchan	...
9	10	110.20137	13CO	1-0	238.238	786	812	...
8	11	102.54798	CH3CCHv=0	6(0)-5(0)	179.740	529	586	...

Source Table

index	Spectral Win. index	Line index	Ra	Dec	Peak	Flux	S/N	...
41	10	9	12:34:56	01:02:03	12.3	20.1	30	...
42	10	5	12:34:49	01:03:00	3.6	7.5	5	...

Figure 3: Examples of the metadata tables we create from ALMA and ADMIT products for sophisticated science queries. The tables are linked via their indices (color highlights), allowing combined queries like “Sources with 13CO detected at S/N >10.” Note we can also insert visual cues for the user such as thumbnail images of ADMIT products [Spectral Window Table].

4.2.4 A Simple User Interface

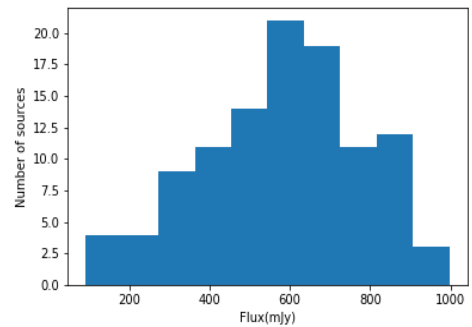
The purpose of this study is not to create a fancy graphical user interface, but to show how the combination of ALMA and ADMIT metadata can allow sophisticated science queries. To that end, we will leverage the ease of use and built-in functionality of Jupyter notebooks [<https://jupyter.org/>] by creating a Jupyterhub [<https://jupyterhub.readthedocs.io/>] connected with our Amazon Web Services (AWS) platform. The high-performance data structure Python package *pandas* [<https://pandas.pydata.org/>] has an API for reading SQL tables such as those in our Science Query Database, allows programmatic manipulation of the table objects (called DataFrames), and can display them neatly in Jupyter notebooks. *Pandas* can even display in table columns thumbnail images such as those created by ADMIT for its products (e.g., moment maps, spectral line plots; see Spectral Window Table in Figure 3), allowing the user to immediately visually identify interesting results. With AWS built-in “serverless computing”, the user can view the web pages produced by ADMIT that summarize in detail the ADMIT Tasks run on each project and their outputs. AWS also supports direct access to product files, such as FITS images, through the boto3 library [<https://boto3.readthedocs.io/>].

As an example, the AQLite query for Science Case #1 might look like:

```
payload = {
    "RA": "04 41 45.9",
    "Dec": "+25 41 27",
    "Radius": "1 degree",
    "Abstract": ["YSO", "young stellar object", "protostar",
    "protostellar"],
    "Lines": ["CO(2-1)"],
    "S/N": [ ">4" ]
}
df = database.query(payload)
```

The query results would be returned as a Dataframe for the user to analyze with *pandas* techniques such as filtering by value. Inside the notebook, the user has access to all the Python packages now familiar to astronomers. For instance, here’s how one could use *pandas* and *matplotlib* to make a histogram of the source fluxes above 100 mJy and with line widths greater than 2 km/s:

```
%matplotlib inline
import matplotlib.pyplot as plt
# create a new DataFrame
# by selecting on flux AND fwhm
df2 = df[(df["flux"]>100) & (df["fwhm"]>2)]
# plot the histogram
plt.hist(df2["flux" ])
plt.xlabel("Flux (mJy)")
plt.ylabel("Number of sources")
plt.show()
```



4.2.5 Amazon Web Services

In order to provide us with the requisite storage and compute power to process enough interesting data to create the full-cube dataset for our mini-archive, we obtained a \$10,000 grant from Amazon to set up a series of AWS machines to match our problem. For this study, use of AWS serves two purposes. First, it gives us compute power on demand and more compute power than is easily available to us locally. Second, it tests possible usage of AWS by ALMA or NA ALMA in the future. Other major astronomical archives are also considering cloud storage and services. The Milkulski Archive for Space Telescopes (MAST) has copied their HST archive over to AWS already (Momcheva & Smith 2019).

The CASA team provides AWS virtual machine images that include all the packages required to run CASA. Our standard compute engine is 128GB of RAM with 16 x “AMD EPYC 7571” 2.5 GHz CPUs with 16TB of storage. This configuration can run the CASA task *tclean* on line data using 11/16 CPUs at peak (fewer at non-peak). However, the advantage of AWS is that memory and compute power can be balanced to match the problem. For smaller cubes, less powerful (and less expensive) machines can be deployed, and vice versa, for larger problems up to 768GB RAM and 96 CPUs is easy to obtain. The largest machine currently available has nearly 4TB memory and 128 cores! The variety of machine instantiations that AWS makes available allows us to use the \$10,000 grant wisely.

Assuming a typical project contains about 100 GB of data (ASDM), we can estimate the total compute resources and cost assuming typical cubes (1k x 1k x 2k) of 8GB each, as summarized below.

AWS Cost Breakdown per Source
(16 CPUs, 128GB memory)

Task	Time (hr)	Cost
Download data using AQ	2	\$0
Re-image 4 spectral windows	16	\$30
Run ADMIT	4	\$10
Ingest ADMIT metadata	1	\$2
TOTAL	23	\$42

Rounding the total hours to 1 project/day, we estimate we can process 30-60 projects in 1-2 months at a cost of \$1500-\$3000 to download, process, and ingest all data. For the study we would keep all data cubes, ADMIT products, Python code, and the Science Query Database, but not the u,v data, on AWS, at a cost of about \$2000/year. Retaining the u,v data would incur an additional ~\$5000/year storage charge. It should be noted that uploading data to AWS is free, however storage and CPU are not, nor is downloading data from AWS (which we do not include here). The overhead of running remote Jupyter notebooks are small, and we don't plan to support large downloads from the cloud during a user session.

One of the outcomes of this study will be an evaluation of AWS services for processing ALMA data. The selection of machines to get the task done gives us a chance to evaluate the cost and usefulness of this type of cloud computing for ALMA projects. Regardless, the code we will distribute can be used for anyone to download their own projects, use ADMIT, and interact with the data through AQLite and *pandas* DataFrames, either locally if they have enough resources or on a cloud platform.

4.3 Work Plan

This description of our work plan closely matches the Gantt chart in Section 11.

First we select and download a number of publicly available projects from Cycle 5 (2017.x) using AstroQuery (AQ). This immediately enables us to capture the already present metadata in the ALMA archive. Since downloads can be slow (10-15 MB/s) we expect it could take several months to download all the projects. We will work with pipeline products if present. However, if project data are not yet fully imaged, we will download either the calibrated Measurement Sets (available Summer 2019 [Jeff Kern, private communication]), or download the ASDM re-calibrate with the ScriptForPI and re-image the data.

We will design the additional tables needed for the ADMIT metadata in the database (Figure 3). These already largely match ADMIT's current science data products. We do not envision having

to add new tasks in ADMIT; we will fine tune the production of the metadata in ADMIT tasks as needed. As projects come in from the archive, we will run ADMIT on them and populate our database. It can be anticipated that there will be developments in the database and metadata during the first 6 months of the project. Fortunately, ADMIT is fully scriptable; ADMIT products can be recreated and re-digested with modest CPU cost and little human time.

We build a new lightweight package AstroQueryLite (AQLite) for the Science Query, which will be based on the SQLite software [<https://www.sqlite.org>]. We are going to a separate package from AQ to have more freedom in design and implementation during the study. AQLite will run on the AWS platform and most queries will return a *pandas* DataFrame.

The user interface will be a Jupyter notebook. We will set up a JupyterHub connected to our AWS cloud resources, allowing users to access AWS remotely via a local Jupyter notebook, which allows them to work with their other favorite science packages they may be running, e.g. SciPy, matplotlib, Glue, Bokeh.

In building new software, adoption by testers and interested users is typically a bottleneck. We will employ two strategies to facilitate the user testing. In the initial selection of projects from Cycle 5, we will contact project PIs to gauge their willingness to be testers. Second, once our prototype is ready for significant user testing, we will talk with selected PIs about inclusion of their Project in our mini-archive. Documentation will be kept up to date throughout the whole performance period, and a final report and users guide will be delivered at the end.

Dr. Teuben, study PI, will be responsible for overall management, data collection and processing, implementation of our Science Database, and AQLite. Dr. Pound will be responsible for JupyterHub implementation and integration of pandas and admit webpages into the system. Dr. Mundy will assist in the imaging and creation of ADMIT products. He will lead quality control and user testing. All three will work together on science cases, defining metadata, documentation, and top-level design for all aspects of the study.

5.0 Study Deliverables

5.1 Hardware

No deliverables.

5.2 Software

The testbed will be delivered in the form of a public github repository, with scripts that will ingest the sample data from the ALMA archive into the final state that we achieved in this study (querying the data). A snapshot of the repository in the form of a tar file will be delivered as well, as well as the final sqlite database.

5.3 Services

No deliverables

5.4 Documents

- Monthly Progress Reports
- Final Report
- User Guide (usage and examples)

6.0 Interfaces to ALMA

We plan to make a simplified version of AstroQuery (AQ) available, dubbed AstroQueryLite (AQL) that is able to process the archive and science queries exemplified in this project. Since this is a proof of concept study, we decided an exact replica of AQ would be overkill.

7.0 Site Location Impact Statement *(if applicable)*

We anticipate no impact on existing facilities.

8.0 Period of Performance

October 1, 2019 – September 30, 2020

9.0 Staffing

9.1 Offerer's Staffing

Table 2.0: Labor Estimate.

TITLE (EXAMPLES)	KEY PERSONNEL	FTE	DURATION (MONTHS)
Principle Investigator	Peter Teuben	0.25	12
Engineering Lead	Peter Teuben		
Engineering Lead	Marc Pound	0.0833	12
Scientific Lead	Lee Mundy	0.03	12
<i>TOTALS</i>		0.00	0.00

9.2 External Staffing (if applicable)

Table 3.0: External Staffing and Contact Information.

TITLE	NAME	INSTITUTION	EMAIL	TELEPHONE
Consultant	Jeff Kern	NRAO	jkern@nrao.edu	434-296-0260
Vendor Point of Contact	Jianjun Xu	Amazon	jianjx@amazon.com	202-361-5585

10.0 Study Schedule

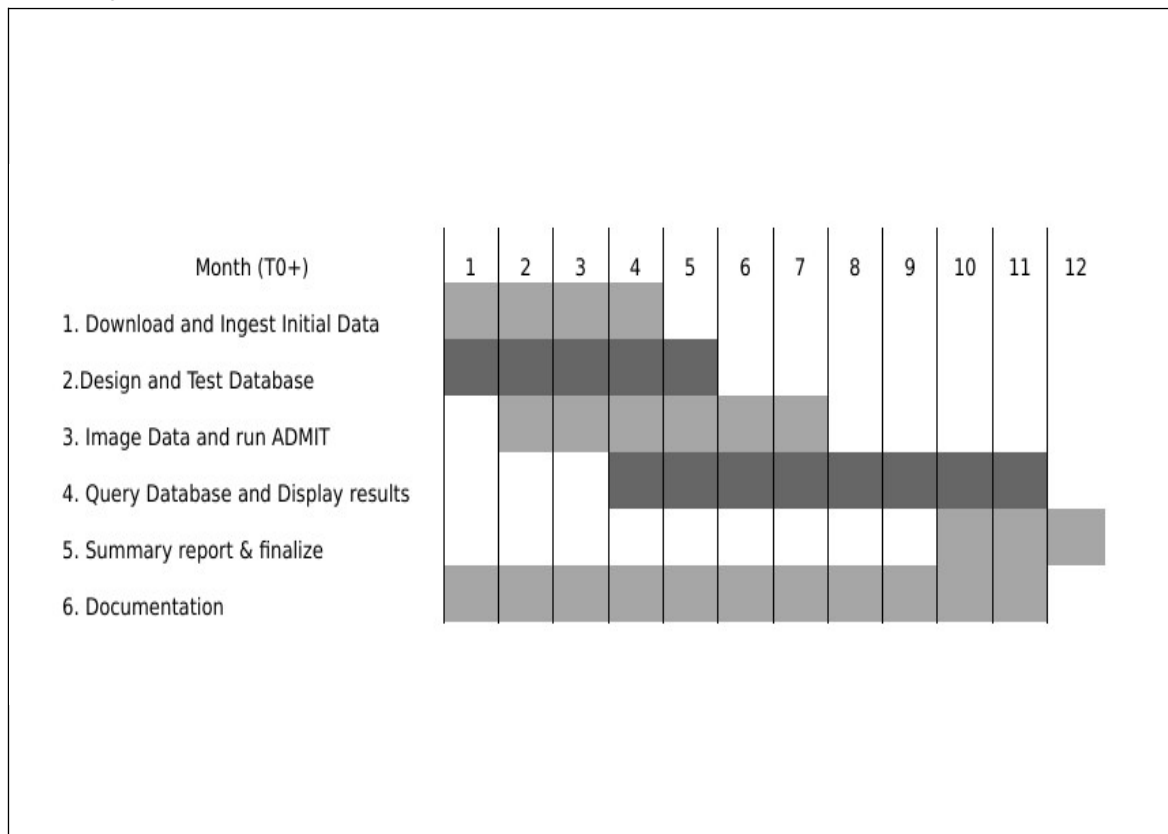


Figure 1.0: Study Schedule.

11.0 Study Management

11.1 Systems/Configuration Control

11.1.1 Systems Requirement and Specification Control

Refer to [RD1].

Development engineering and design activities shall be conducted in accordance with established Systems Engineering policies, practices and procedures.

11.1.2 Documentation Control

All shared documents shall be dated and bear a revision level number.

11.1.3 Product & Quality Assurance Control

Refer to [RD2]. A unique Product Assurance Plan is unnecessary.

Development engineering and design activities shall be conducted in accordance with established ALMA PA/QA policies, practices and procedures.

11.2 Performance to Schedule

The Principal Investigator has primary responsibility for schedule development and performance to schedule. The NA ALMA Development Program Office will, if requested, provide support to the PI in establishment of a revision-controlled Study Schedule and monthly preparation of performance to schedule status. In the event of a schedule variance, the PI and the NA ALMA Development Program Manager will assess the impact and develop the appropriate recovery action(s).

11.3 Performance to Budget

The Principal Investigator has primary responsibility for intra-Study budget allocation and cost performance. The NA ALMA Development Program Office, if requested, will provide support to the PI in establishment of cost accounts, budget load, and the preparation of a revision-controlled, monthly Budget Status Report. In the event of a cost variance, the PI and the NA ALMA Development Program Manager will assess the impact and develop the appropriate recovery action(s).

11.4 Measures of Success

Apart from generally working software, we measure success when all the data has been properly ingested, and successfully queried with a variety of science cases as presented in Section 3.0

For a successful study, we expect this to be a scaleable implementation for a future ALMA science archive.

11.5 Risk Management

Table 9.0: Project Risk Assessment.

No.	PRIMARY RISK(S)	PROB. (%)	IMPACT (\$)	MITIGATION
0	Download speed too low	10	0	USE HARD DRIVE AND UPS
1	AWS not sufficient	10	0	USE LUSTRE AT NRAO
2	No calibrated MS	25	0	USE ASDM OR REFERENCE IMAGES
3	Cycle5 not enough public	10	0	USE CYCLE4
TOTAL STUDY CONTINGENCY (\$)			0.00	

11.6 Communication Plan and Progress Reporting

A monthly Progress Report shall be prepared by the Principal Investigator in accordance with NRAO Program Management practices and procedures. Informal reviews will be conducted by the NA ALMA Development Program Manager upon the completion of project milestones.

12.0 Study Closeout

Upon conclusion of this Study, the NA ALMA Development Program Office will coordinate the orderly closeout of activities; or, the transition of activities to a continuing Study or Project. At a minimum, this will include the following:

- verification of compliance with established procurement policies and procedures;
- verification of Purchase Order final payments;
- cost and schedule variance analysis;
- inactivation of cost accounts;
- preparation of a Final Report;
- preparation of an Outcome Report (if applicable); and
- archiving of Study records.

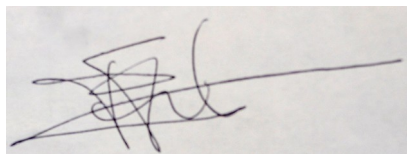
13.0 Commitment

Having read all documents listed in and annexed to the Call for Development Study Proposals, and having assessed the situation and the nature and difficulties of the proposed services, the undersigned hereby offers the “Science Mining the ALMA Archive” in accordance with the provisions of the present Call for Development Study Proposals and, if awarded the Agreement, undertakes to carry out the work required according to best trade practices, within the prescribed time limits, and at the price set out in this Proposal.

Name: Peter Teuben

Institution: University of Maryland

Signature:

A handwritten signature in dark ink, appearing to be 'P. Teuben', written on a light-colored background.

Date: May 1, 2019

Appendix A - Reference Documents

- ADMIT: <http://admit.astro.umd.edu>
- ALMA memo 613: Liuzzo et al. (Dec 2018)
- ALMA memo 614: Massardi et al. (Jan 2019)
- ALMA QA2 Data Products for Cycle 5 (May 2018)
- ALMA Science Archive Manual (March 2019)
- AstroQuery: <https://astroquery.readthedocs.io>
- Momcheva & Smith (2019) – ADASS XXVIII, p.223
- Friedel et al. (2014) – ADASS XXIII, p.151
- sqlite: <https://www.sqlite.org>
- Teuben et al. (2015) – ADASS XXIV, p.305