UNIVERSIDADE DE LISBOA

FACULDADE DE CIÊNCIAS

DEPARTAMENTO DE INFORMÁTICA

Better Science Through an Enhanced User Interface with the ALMA Archive

António Antunes Dias

Mestrado em Engenharia Informática

Especialização em Engenharia de Software

Trabalho de Projeto orientado por:

Sara Madeira

Israel Matute

2022

Dedicatória e agradecimentos

Em primeiro lugar, a Israel Matute e Sara Madeira, pelo seu acompanhamento e confiança ao longo deste projecto. Um agradecimento especial por nunca terem deixado de acreditar no seu sucesso e pelo auxílio pessoal nas alturas mais difíceis.

Ao pessoal do Instituto de Astrofísica pela hospitalidade, curiosidade que sempre mostraram pelo projecto e cordialidade nunca diminuída pela minha azelhice com VPNs. Um agradecimento especial a José Afonso e Ciro Papallardo pelo *feedback* valioso ao longo do processo de desenvolvimento.

A Felix Stoehr e restante equipa do ESO pelo entusiasmo, disponibilidade e pelo fornecimento de dados vitais ao funcionamento desta ferramenta. Espero que o desenvolvimento continuado da mesma represente uma mais-valia para a comunidade do ALMA.

À minha família, pelo apoio que sempre prestaram – mas também pelos empurrões que fazem de mim uma pessoa melhor todos os dias.

À Bianca e à Margarida, cuja amizade indispensável me deu motivação para trazer este projecto a bom porto e a quem devo uma quantidade indescritível de tostas de salmão por estarem sempre presentes.

Ao Marcelo e ao Zé, por fazerem parte das melhores memórias da minha vida.

Ao Pedro e à Cláudia da STRÓ, pelos cachecóis sempre confortáveis, pela oportunidade única de trabalho que me deram no distante ano de 2019 e por se terem sempre interessado nesta tese.

Aos meus colegas e amigos dos Hypixel Studios, a quem agradeço a confiança que depositaram em mim no último ano e com quem eu espero continuar a trabalhar num jogo estupendo.

Aos meus animais: Alice, Zé, Tobias e Amélia, por toda a confusão que criam em casa, por taparem o monitor de 5 em 5 segundos, pelos momentos de brincadeira e por me ajudarem a treinar a arte da fotografia.

*

Dedico esta tese à memória da minha avó materna, Noémia Martins Lopes, e do meu cão, Sebastião.

Resumo

O Atacama Large Millimetre Array, localizado no deserto Chileno homónimo, é um dos observatórios interferométricos mais avançados do mundo. Composta por um conjunto de 66 antenas que actuam com um único instrumento, esta instalação especializa-se na observação de luz com comprimentos de onda na ordem dos milímetros, equivalente à radiação infravermelha e microondas de alta frequência: uma região do espectro electromagnético de grande interesse para a ciência extragaláctica, mas tanbém de relevo no estudo da formação planetária e estelar. Tal como outros interferómetros, o ALMA faz uso do processo de síntese de abertura: uma técnica de observação e processamento de dados em que vários receptores produzem imagens com a resolução angular equivalente à de um telescópio com um diâmetro igual ao espaçamento máximo entre cada antena, podendo esta distância, no ALMA, chegar aos dezasseis quilómetros: uma configuração que, dependendo das frequências observadas, produz imagens de resolução até cerca de 20 milliarco-segundos (ALMA Basics, s.d.). Consequentemente, os dados obtidos pelo observatório servem as necessidades de uma comunidade crescente de utlizadores que se enquadram em diversos grupos de investigação de astronomia e astrofísica numa grande variedade de projectos, dos quais se pode destacar o Event Horizon Telescope, responsável pela obtenção da primeira imagem directa de um buraco negro em 2019 (The Event Horizon Telescope Collaboration, 2019).

Contudo, e tal como acontece com qualquer instrumentação do género, o armazenamento e exploração de dados constituem desafios de grande relevo; enquanto que o primeiro pode ser directamente colmatado através da actualizações técnicas à infrastrutura do observatório, o último depende, em larga maioria, dos meios de acesso, análise e apresentação dos dados obtidos. Actualmente, as observações registadas pelo ALMA são publicadas na plataforma web do ASA (ALMA Science Archive). Esta ferramenta contém uma lista de observações obtidas pelo ALMA, assim como um histograma de cobertura de frequência de cada observação e uma representação da sua localização espacial e campo de visão. Adicionalmente, o utilizador pode aceder a um portal de armazenamento onde pode descarregar as imagens obtidas em qualquer observação – deve-se referir que as observações são publicadas após um período de doze meses contados a partir do fim da segunda fase de controlo de qualidade (QA2), estando antes disso exclusivamente disponibilizadas aos autores do projecto a que dizem respeito. Contudo, os metadados de uma observação são publicados assim que a mesma passa pela primeira fase (QA0) (Cox, 2017). Apesar do ASA permitir aos utilizadores a filtração de observações com base em metadados – por exemplo, através da localização, resolução angular ou tempo de integração das mesmas –, considera-se que a plataforma poderia beneficiar de ferramentas complementares que, através de componentes de visualização de dados, permitam a avaliação mais directa tanto do estado global do arquivo do ALMA como de tendências que possam existir em *clusters* de observações densos. Estas ferramentas deverão servir um conjunto de diferentes utilizadores e objectivos:

- Considera-se que o arquivo do ALMA contém um grande potencial científico que pode ser extraído através de processos de combinação de observações. Sucintamente, a qualidade dos dados obtidos pelo observatório depende principalmente do nível de ruído das imagens, que é altamente variável e depende de um grande número de factores. Embora a combinação de imagens seja um processo geralmente dispendioso em recursos computacionais e temporais, é possível obterem-se estimativas da sensibilidade resultante da combinação de duas ou mais imagens apenas recorrendo aos seus metadados; esta opção, significamente mais rápida, pode informar o processo de selecção de observações a combinar. Ademais, as estimativas com base em metadados podem ser efectuadas em observações que ainda não foram publicadas, aumentando assim o conjunto de dados sobre os quais se pode trabalhar em qualquer instante.

- Enquanto instituição científica, o ALMA procura sempre promover o envolvimento e o conhecimento do público em geral dos resultados científicos do observatório. Para além das usuais actividades de divulgação, pretende-se ainda facilitar o uso do arquivo de dados por parte de utilizadores menos experientes. Entende-se que, para esse fim, qualquer ferramenta a ser desenvolvida terá de colocar um ênfase especial na interface de utilização e no apelo visual das ferramentas de análise fornecidas.

Estes desafios levaram à criação de uma proposta de projecto por parte do Instituto de Astrofísica e Ciências do Espaço que, ao invés de se focar num conjunto bem definido de funcionalidades, tem por objectivo final o desenvolvimento de protótipos para ferramentas de análise e visualização dos dados do ALMA. Geralmente, o projecto final deverá permitir que os utilizadores obtenham informações respeitantes tanto ao estado actual do arquivo – principalmente no que diz respeito à distribuição da área de cobertura pelas frequências observadas – como ao potencial científico por explorar em agrupamentos de observações. Deve-se, porém, destacar a necessidade de se poderem obter estimativas rápidas do ganho obtido através da combinação de observações, visto este processo representar uma enorme mais-valia para a comunidade. Esta tese descreve a motivação, âmbito e implementação de um projecto em curso pelo Instituto de Astrofísica e Ciências do Espaço que tem por objectivo final o desenvolvimento e produção de um *website*, separado do ALMA, que forneça aos utilizadores um conjunto de ferramentas de análise e visualização de dados do observatório, pretendendo assim complementar as ferramentas existentes com uma plataforma dedicada à visualização e análise estatística do arquivo. O projecto em curso, temporariamente denominado *ASH* (ALMA Science Hub), consiste numa plataforma web que disponibiliza duas componentes principais de visualização de dados: um diagrama de dispersão de *clusters* de observações e uma ferramenta de geração de mapas de calor numa região definida pelo utilizador: a primeira representa uma abordagem inicial à exploração científica do arquivo – destacando regiões do céu onde existe uma maior área de sobreposição de observações – enquanto que a última foca-se na estimativa do ganho de sensibilidade obtido nas áreas de sobreposição presentes num determinado campo de observações (por motivos de desempenho, e dada a complexidade geométrica de observações em mosaico, esta estimativa é calculada num conjunto limitado de pontos, representados como píxeis). De futuro, pretende-se que estas duas ferramentas estejam sincronizadas uma com a outra, permitindo que o utilizador, por exemplo, consiga obter os parâmetros para o mapa de região que cubra a totalidade de um determinado *cluster*.

A longo prazo, espera-se que a plataforma seja útil tanto para as redes de investigação do ALMA como para uma base de utilizadores amadora que, que de uma maneira ou outra, se queira envolver nas actividades do observatório. Em paralelo, os algoritmos de combinação de dados são constantemente melhorados pelo grupo do IA, e as limitações das ferramentas actuais são cada vez melhor conhecidas. Ademais, também estão a ser exploradas potenciais sinergias com outras ferramentas científicas do ALMA orientadas a metadados, estando duas das quais descritas em maior detalhe no capítulo 2.

Palavras-chave: web, computação científica, visualização de dados, astronomia

Abstract

The Atacama Large Millimetre Array, located on the homonymous Chilean desert, constitutes one of the world's most advanced interferometric observatories. Covering the millimetre and sub-millimetre wavelengths, the facility supports the needs of an ever-growing community within astronomy and astrophysics research groups. As it happens with all ground-based astronomical instruments, however, data storage and exploitation remain an outstanding challenge; while the former can be managed through technical upgrades to the Array instrumentation, the latter largely depends on how the archival data is accessed, analysed and presented to the user. Currently, ALMA observations are made publicly available through the ASA (ALMA Science Archive) online platform. Despite offering native, metadata-based filtering that can be applied to an observation's various fields, such as location, spectral windows or integration time, it is understood that the ASA would benefit from supplementary tools improving its visualization components, allowing for a more direct assessment of both the archive's global state and its more localized clusters (e.g., field density distribution). This thesis describes the status of an ongoing project within the Institute of Astrophysics and Space Sciences (IA), foreseeing the development and production of a separate website that provides the user with a set of data analysis and visualization tools; while other functionalities are predicted to be included, the platform will mainly focus on the visual plotting of specific regions containing ALMA observations, enabling the users to directly identify particularly deep regions of the sky. Apart from allowing for a better scientific exploitation of pre-existing astronomical data through the combination of interrelated observations, the visualization toolkit may also increase the archive's appeal and utility among non-expert users, promoting a larger public engagement with the ESO's activities; on a long-term basis, it is intended that this platform will come to represent a valuable asset to the ALMA community.

Keywords: web, scientific computation, data visualization, astronomy

Índice

Lista de quadros e figuras

*

Lista de abreviaturas

ALMA – Atacama Large Millimeter Array

ASH – ALMA Science Hub

ARC – ALMA Regional Centre

ARTEMIX – ALMA RemoTE Mining eXperiment

ASA – ALMA Science Archive

CRUD – Create-Read-Update-Delete

CSS – Cascading Style Sheets

CSV – Comma-Separated Values

DOM – Document Object Model

ESO – European Southern Observatory

FITS – Flexible Image Transport System

FWHM – Full-Width-At-Half-Maximum

GHz – Gigahertz

HTML – HyperText Markup Language

IA – Instituto de Astrofísica e Ciências do Espaço

JSON – JavaScript Object Notation

KAFE – Key-analysis Automated FITS-images Explorer

LOD – Levels-Of-Detail

mJy – millijanskys

MVT – Model View Template

ORM – Object-Relational Mapping

SVG – Scalable Vector Graphics

UI – User Interface

URL – Uniform Resource Locator

1 Introduction

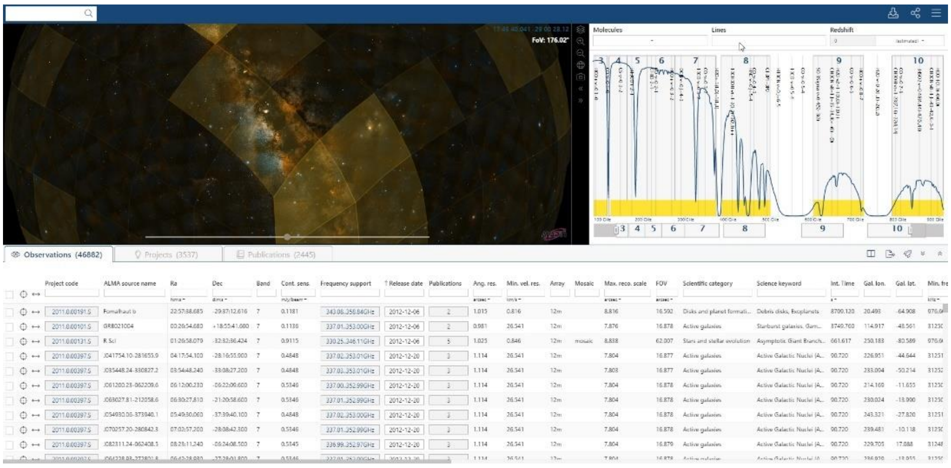
This chapter contains a brief history of the ALMA observatory (almaobservatory.org/en/home), the motivation and drivers behind this project and an introduction to the data combination process and its scientific potential.

1.1 Context

Located high on the homonymous Chilean desert, the Atacama Large Millimetre Array (ALMA) is the world's largest millimetre/submillimetre telescope complex. Composed of sixty-six antennas acting together as a single dish through a process called interferometry, the ALMA facility can resolve images on the 84-950 GHz band, which corresponds to millimetre and submillimetre wavelengths. In astronomy, this region of the electromagnetic spectrum is commonly observed in respect to infrared emissions from cold gas nebulae - which can be used to study and characterize the background stars responsible for heating up these clouds - and planetary formation. Adding to their importance in the study of local radio sources (e.g., sky objects located within the Milky Way), this frequency range is also of vital importance for cosmology and extra-galactic science, with diverse branches such as galactic formation and distant quasars, whose emission sources are redshifted into the infrared and near microwave regions.

The project traces back to the late 90's, where multiple concepts for a millimetre-wavelength observatory were merged into a central project, with the final agreement being signed in 2003 between European and American organizations, with Japan joining later. ALMA has contributed to thousands of articles and citations across a wide range of astronomy and cosmology studies, from near-Earth objects to some of the most distant – and youngest – galaxies ever found. Recently, ALMA was highlighted as one of the chief contributors to the Event Horizon Telescope project (eventhorizontelescope.org), which famously produced the first direct image of a black hole (The Event Horizon Telescope Collaboration, 2019). The Array is an ever-evolving infrastructure with further capabilities being provided each year, whether in terms of data processing capabilities or spectral coverage.

ALMA data is made available through the ALMA Science Archive (almascience.eso.org/aq), a web-based tool that allows users to explore and download all available observations to date through their metadata. As it happens with many scientific platforms, the ASA's usage of rich visualization tools fills a critical goal on both scientific research and community engagement, not only facilitating data queries but also engaging non-expert users with the work being done by the observatory. A screenshot of the Archive's main page can be found in Figure 1.1.



Project code	ALMA source name	Ra	Dec	Band	Cont. sens.	Frequency support	Release date	Publications	Ang. res.	Min. vel. res.	Array	Mosaic	Max. reco. scale	FOV	Scientific category	Science keyword	Int. Time	Gal. lon.	Gal. lat.	Min. fre
2011.0.00191.S	Fomalhaut b	22:57:38.685	-29:37:12.616	7	0.1181	343.08..358.84GHz	2012-12-06	2	1.015	0.816	12m		8.816	16.592	Disks and planet formati...	Debris disks, Exoplanets	8709.120	20.493	-64.908	976.0
2011.0.00101.S	GRB021004	00:26:54.680	+18:55:41.600	7	0.1136	337.01..353.00GHz	2012-12-06	2	0.981	26.541	12m		7.876	16.878	Active galaxies	Starburst galaxies, Gam...	3749.760	114.917	-43.561	3123
2011.0.00131.S	R Scl	01:26:58.079	-32:32:36.424	7	0.9115	330.25..346.11GHz	2012-12-06	5	1.025	0.846	12m	mosaic	8.838	62.007	Stars and stellar evolution	Asymptotic Giant Branch...	661.617	230.183	-80.589	976.0
2011.0.00397.S	J041754.10-281655.9	04:17:54.100	-28:16:55.900	7	0.4848	337.02..353.01GHz	2012-12-20	3	1.114	26.541	12m		7.804	16.877	Active galaxies	Active Galactic Nuclei (A...	90.720	226.951	-44.644	3125
2011.0.00397.S	J035448.24-330827.2	03:54:48.240	-33:08:27.200	7	0.4848	337.03..353.01GHz	2012-12-20	3	1.114	26.541	12m		7.803	16.877	Active galaxies	Active Galactic Nuclei (A...	90.720	233.004	-50.214	3125
2011.0.00397.S	J061200.23-062209.6	06:12:00.230	-06:22:09.600	7	0.5346	337.00..352.99GHz	2012-12-20	3	1.114	26.541	12m		7.804	16.878	Active galaxies	Active Galactic Nuclei (A...	90.720	214.169	-11.655	3123
2011.0.00397.S	J063027.81-212058.6	06:30:27.810	-21:20:58.600	7	0.5346	337.01..352.99GHz	2012-12-20	3	1.114	26.541	12m		7.804	16.878	Active galaxies	Active Galactic Nuclei (A...	90.720	230.024	-13.990	3123
2011.0.00397.S	J054930.06-373940.1	05:49:30.060	-37:39:40.100	7	0.4848	337.02..353.00GHz	2012-12-20	3	1.114	26.541	12m		7.804	16.878	Active galaxies	Active Galactic Nuclei (A...	90.720	243.321	-27.820	3125
2011.0.00397.S	J070257.20-280842.3	07:02:57.200	-28:08:42.300	7	0.5346	337.01..352.99GHz	2012-12-20	3	1.114	26.541	12m		7.804	16.878	Active galaxies	Active Galactic Nuclei (A...	90.720	239.481	-10.118	3123
2011.0.00397.S	J082311.24-062408.5	08:23:11.240	-06:24:08.500	7	0.5345	336.99..352.97GHz	2012-12-20	3	1.114	26.541	12m		7.804	16.879	Active galaxies	Active Galactic Nuclei (A...	90.720	229.705	17.088	3124

Figure 1.1: A screenshot of the ALMA Science Archive web platform. The page features a table containing all ALMA observations, a spatial viewport that shows observation density and footprints and a frequency coverage histogram.

1.2 Project concept and motivation

Ever since its first light in 2011, the ALMA observatory has seen a continuous increase in technical and scientific capabilities, and its data archive contained more than 45 000 individual observations at the time of writing of this thesis. As the facility's data output increases, so does the need for more capable data linkage and storage infrastructure. However, and as understood by the Institute of Astrophysics and the European Southern Observatory (eso.org/public) community, scientific exploitation can also be improved with existing observation metadata, which is made public even before the observation undergoes quality assurance processes (in contrast, the data cubes themselves enjoy a 12-month proprietary period).

The IA's proposal aims to develop a web-based application that interacts with the ASA and, through its metadata, allows both expert and amateur users to better understand the state of the facility's extensive archive, while also providing them with a varied and user-friendly visualization and filtering toolset. Many of the design decisions that have been taken concerning their development are explained in detail on their respective chapters.

The development process was highly agile, as new features were continuously proposed by both the IA stakeholders and the ESO community and their technical aspects better understood over time. The first page to be developed concerned a field plot tool that displays any subsection of the celestial sphere as it is covered by ALMA observations. This remains the main tool through which expert users can estimate the combined sensitivity between any group of overlapping observations and their mutual frequency coverage in respect to specific emission lines.

Two more tools were developed on the second half of the project and were kept as prototypes: a sky map which provides visual information on regional trends (e.g., which sections of the sky contain a higher density of observations and/or overlapping areas) and a homepage which provides general archive information and aims to engage amateur users with the work done by ALMA.

2 Background

2.1 Combining observations

One of the project's earliest drivers was the need for a toolset that could allow users to seek out regions of the sky where overlapping observations can be found and combined with each other. This process results in an image with a potential better sensitivity, from which more and better scientific data can be extracted. Since such an operation requires direct manipulation of the data cubes themselves, it is outside the scope of this project; regardless, and given the high resource and time costs involved, the final achieved sensitivity can and should be estimated beforehand, therefore allowing the user to pre-select which data to analyse.

The notion of overlapping data, in the context of ALMA, can be applied to spectral overlap, where two or more observations happen to observe the same regions of the electromagnetic spectrum, or geometric overlap, where the same region of the sky is imaged by multiple observations. While providing information regarding spectral overlap, this tool focuses on the latter.

As will all antennas, the power response of ALMA's reaches a maximum in the azimuth of the receptor but starts to decrease as the radial distance to the centre of the image increases. At the same time, sensitivity decreases, which directly correlates with signal degradation and noise. In interferometers, the field-of-view of an image corresponds to the angular size of the full-width-at-half-maximum (FWHM) beam. This threshold represents the radial distance at which the power response of the antenna falls below 50% of that of the azimuth. Additionally, as shown on Figure 2.1, the power response function is different for 7-meter and 12-meter antennas.

It is possible to decrease the noise from the higher-sensitivity regions of two overlapping observations, resulting in higher signal-to-noise data that, due to its improved quality, could potentially yield new scientific targets and analysis pathways. The process of analysing and combining observations is, however, computationally expensive; at the same time, it requires ASA users to download the raw data, calibrate and image the entire set of data cubes that are to be combined.

Achieving accurate estimates for the final, combined sensitivity depends on information that's not currently included in the ASA metadata. The noise level for any observation is usually calculated with the following formula:

$$S = \frac{k * T_{sys}}{A * N^2 \sqrt{N_p * \Delta v * \Delta t}}$$

2.1

where T_{sys} is the system temperature, A the area of each antenna, N the number of antennas, N_p the number of polarizations, Δv the available bandwidth, Δt the observation time of the image and k the brightness temperature for extended sources (ALMA Basics, s.d.). Two of these variables are not included in the metadata: the system temperature and the number of antennas; at the same time, the maximum spatial resolution depends on the used baseline configuration, which isn't provided either.

Rough estimates of the final sensitivity can be nonetheless directly obtained through the (currently publicized) continuum sensitivities of all observations that contribute to a common area: a method which, while not yielding accurate results, may result in efficient exploitation of existing data by quickly providing the user with insight on which datasets could result in higher-quality images.

The resulting sensitivity on each point – equivalently, pixel – with observation overlaps is therefore calculated with the metadata from all covering observations, according to the following process:

1. Each observation covering the point has its sensitivity normalized to a value that depends on the distance to the observation's centre (primary beam), where the power response is at its maximum, and the covered frequencies. The power response function, as seen in Figure 2.1, was approximated by a Gaussian function and, as such, ignores the antenna's side lobes.

2. Scale all observations to a predefined resolution in mJy/beam.

3. Sum the corrected sensitivity of each combined observation, according to the formula

$$S_f = \frac{1}{\sqrt{\sum_{i=0}^{N-1}\left(\frac{1}{S_i}\right)^2}}$$

2.2

where S_i is each observation's sensitivity and S_f the final combined sensitivity.

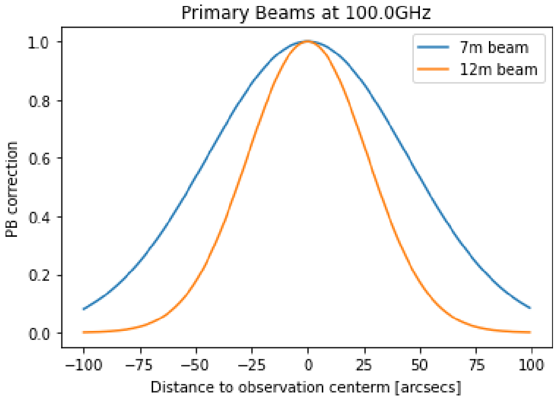


Figure 2.1: The approximated power response function for ALMA's antennas at the frequency of 100 GHz, highlighting how signal degradation increases with the angular distance to the observation's center. Note the difference between the power responses of 7m and 12m antennas: the reason each is handled differently for sensitivity calculations.

2.2 Ongoing projects

From its conception, this project was designed to complement other existing tools that, in one way or another, aim to facilitate data exploitation of the ALMA archive through visualization suites. This section describes two such initiatives.

2.2.1 ARTEMIX

The ALMA RemoTE MIning eXperiment (ARTEMIX, Consulted 2021), in development by the Paris Observatory, consists of a data exploration web service that strives to improve ASA queries based on spatial coordinates, search radii and chemical species, among others. While having a similar scope to ASH (both rely on archive metadata and focus on archive accessibility by less-experienced users), it does not contain the necessary visualization tools to fulfil the IA's goals, namely the ability to query intersections between observations or their combined sensitivities.

2.2.2 KAFE

The Key-analysis Automated FITS-image Explorer (Burkutean, et al., 2018), in development by the Italian ALMA Regional Center (ARC) with researchers from the University of Bologna, the Bologna Observatory and the Radio Astronomical Institute, is an online post-processing tool for FITS images. The project was designed around the visualization of data cubes taken on the millimeter and sub-millimeter regions of the electromagnetic spectrum and seeks to provide an elaborate and visually appealing data plotting toolset. Despite much of its background processing relying on observation metadata, it is better suited to image analysis of the data itself and, as such, can be understood to fulfil a different role.

3 The ALMA Science Hub

3.1 Overview

The ALMA Science Hub (ASH) is a web-based data visualization tool that aims to complement the (also web-based) ALMA Science Archive by focusing on user experience, meaningful and engaging data exploration tools and the needs of a non-specialist userbase interested in ALMA's operations and scientific output. Three separate webpages have been designed and implemented for a variety of functional and non-functional requirements:

- **Homepage:** basic information on the ALMA archive, overall spectral coverage and an embedded feed of the observatory's Twitter account used for outreach, job postings and operational updates,

- **Sky map:** an overview of the ALMA coverage of the celestial sphere, represented by a scatterplot of observation clusters that direct the user to dense regions (both in terms of total and overlapping area)

- **Field plot:** a data visualization tool that generates a pixelized render of the observations that cover a given region and their overlaps, properties and spectral coverages. This tool mainly aims to facilitate data combination processes by providing the user with an estimate of the achievable sensitivities – a process which is described in further detail in section 2.1.

3.1.1 Technical description

The project was built upon the following languages, frameworks, and technologies:

3.1.1.1 Front-end

- HTML (HyperText Markup Language) is a standard declarative language used for documents that are rendered and displayed on a web browser. An HTML file organizes page elements, like text, images, and containers, via a hierarchical tagging system, but doesn't describe control flows, interaction, or user presentation. These page elements are organized in a tree structure known as the Document Object Model (hereafter DOM), which can be manipulated and read through languages such as JavaScript.

- CSS (Cascading Style Sheets) is a style sheet language that provides styling information to a web page, such as size, position, color, or alignment. Any HTML element can implement any number of CSS classes that describe its visual presentation, either through class tags or a direct reference to the element's ID.

- JavaScript is a web scripting language used in tandem with markup languages, such as HTML, to provide user interactivity and control to web pages.

Given its reliance on web-based data visualization, the project makes heavy use of D3.js (Data-Driven Documents, 2011), a JavaScript library built for the rendering, visualization, and analysis of datasets through a multitude of plotting tools. Each data point belonging to the dataset is generally represented by a SVG (Scalable Vector Graphics) object and, as such, is easily manipulated and interacted with – for instance, assigning a circle's diameter to the size of any property of its respective data node is trivial.

However, and as the datasets increase in length, so does the number of individual DOM objects, which may impact performance in terms of data updating, user interaction and rendering responsiveness. As such, a number of rendering strategies are commonly cited as alternatives to D3's SVG-based approach. Two of these have been implemented in different parts of the project and are discussed in detail further ahead.

Additionally, DOM manipulation is also assisted by the jQuery (2006) library, vastly facilitating event handling, input access and validation, as well as providing a framework for the simple implementation of user controls contained within the jQuery UI (2007) widget collection.

3.1.1.2 Back-end

The web service is implemented in Django (2005), a Python-based framework that follows a model-template-view (MTV) pattern. This solution was chosen over the alternatives for a couple of reasons:

- The usage of Python as the backend language allowed the leverage of pre-existing scripts and libraries catered towards astronomical data processing such as astropy, apart from being a language the team was already comfortable with.

- Django takes on a "plug-and-play" approach that facilitates the integration between data storage, access, and processing routines due to an in-built object-relational mapping (ORM) that can interact with multiple database frameworks such as SQLite and PostgreSQL.

As with all Django projects, the ASH is composed of multiple applications. An application is a self-contained module that is designed to provide all the necessary utilities required by a specific task; applications, which are directly mapped to the file structure of the project, can contain their own routing rules, static files, HTML templates, Python scripts and even their own database models. Applications can also freely import functions and classes from each other, which allows for common files, models, and scripts to be placed in top-level applications that act as a project "library".

The site's root folder is designed to contain static files and templates that are used by multiple applications. For this project, the static root folder was used to store basic styling – most visible on the platform's top navbar – and two external libraries: jQuery UI and Font Awesome (Fonticons, Consulted 2021), the latter being an open-source icon repository currently used on the *field_plot* application.

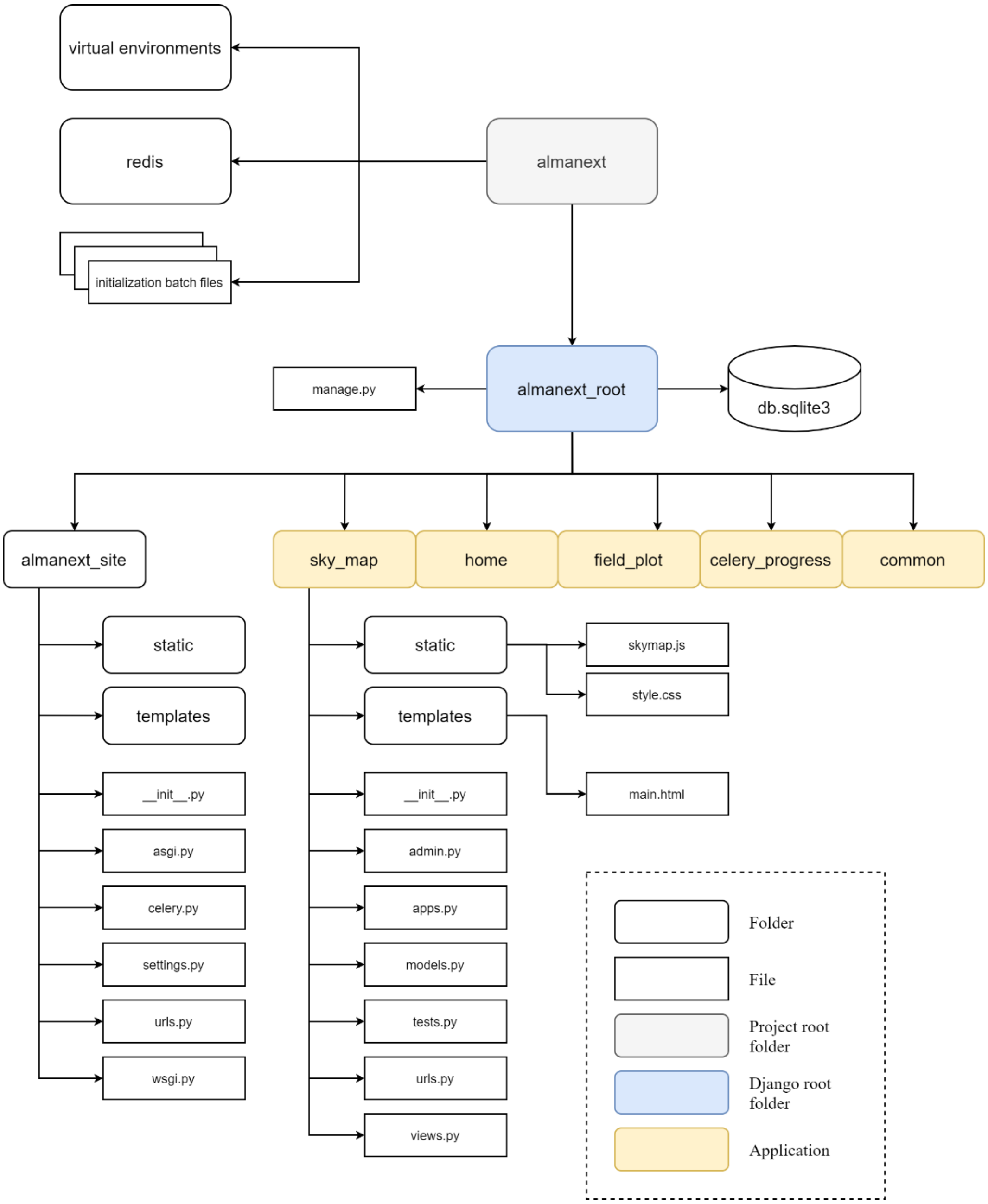


Figure 3.1: The folder structure of the ASH platform. Apart from Django's standard file organization tree and the developed applications, the project also includes a Windows build for the Redis server (3.2.2.4.2) and batch files that initialize all the required components, such as the Celery worker that will build the field plot heatmap (3.2.2.4.1). Only the sky_map application was expanded for illustrative purposes.

Table 3.1 contains a brief description of the applications that currently compose the ASH platform:

Table 3.1: A condensed list of all applications that were developed (or otherwise used) for this project.

common	Contains the models which are used across multiple applications and management scripts.
home	The landing page application.
sky_map	The sky map application.
field_plot	The field plot application.
celery_progress	An application that implements JS progress bars for Celery tasks. Developed by Cory Zue and used in accordance with the MIT license.

3.1.2 The Django web framework

As described in the previous section, Django follows a model-template-view logic that, despite the different nomenclature, is equivalent to that of the well-known model-view-controller pattern: any database object ("model") can be represented by a Python class, which can be interacted with by callable functions known as "views" responsible for rendering HTML responses through a template system. These views are called by a URL dispatcher which acts as the "controller". A simplified schematic of the data flow between these components is presented in Figure 3.2.

3.1.2.1 Model

Django's object-relation-mapper (ORM) works as an adapter between a relational database's objects and Python classes, allowing for a high level of abstraction both for basic CRUD operations and more complex queries. Additionally, data access is agnostic to the database software being used, allowing for the seamless switch between different management systems to accommodate development and production needs.

Object models are defined as Python classes inside an application's models.py file, from where they can be collected and migrated to the database. Django's ORM supports a multitude of field types and cardinal relationships (e.g., one-to-many) through foreign keys.

3.1.2.2 Template

One of Django's most recognizable features is its use of HTML templates that allow for a modular approach to HTML rendering. Templates are text files or string variables that, apart from the typical web markup, include tokens that are interpreted and written to by the template engine. The information that is rendered in these tokens at any given time is known as the page's context. However, and in many cases – such as this project – templates can also be used to split common page elements across multiple files which are then concatenated by the engine. For this project, basic CSS styling and the top navbar, which is present across all applications, were included in the base.html file that can be found in the *almanext_site/templates* folder.

3.1.2.3 View

A Django view is, essentially, a callable function that returns a rendered HTML page. When a request is sent to the server and matched to a URL pattern, it can either be redirected to another dispatcher (such as one belonging to a specific application) or result in a view being called.

Apart from generating HTML files to the client, views also support a variety of return types that allows for the server to send serialized data back to the client in the JSON format. This capability was used throughout the project to obtain data from the backend, which was then rendered in the visualization plots implemented in JavaScript.

Like static files, routing rules can either belong to a specific application or the entire site. While the latter focus on redirecting users to an application's webpage, the former are normally employed to fetch webpage templates or data from the back-end.

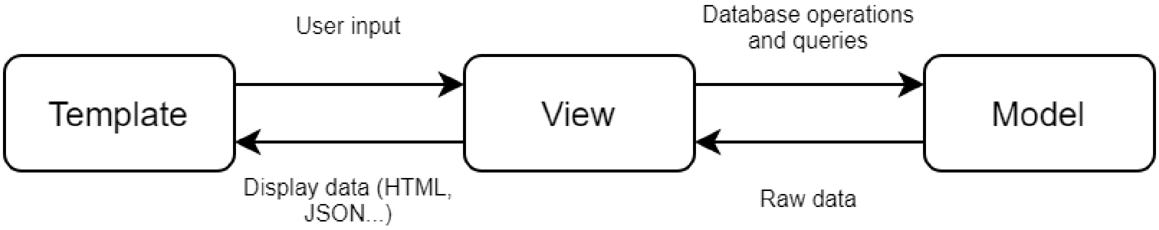


Figure 3.2: Simplified data flow diagram of a Django application. Despite the different nomenclature, the model-view-template architecture is functionally similar to the model-view-controller design pattern.

3.1.3 Data model and requirements

The project's data is obtained from the ALMA Science Archive in the CSV format and converted into a proper database scheme. In general, column names map directly to database fields, but some require additional processing (one such example is an observation's spectral coverage, which is encoded in a string value and then broken down into its separate frequency windows).

While it is possible to automate data updates, the archive does not currently contain the information for an observation's traces, which is vital for the field plot application to work; this data is currently handed over by a member of the European ARC network. As such, to keep information coherent and to test all applications, the project currently uses an ASA snapshot dating back to October of 2019 – a dataset for which a traces file was provided at the time. Making the traces file publicly available is a matter of ongoing discussion between the Instituto de Astrofísica and other ARCs.

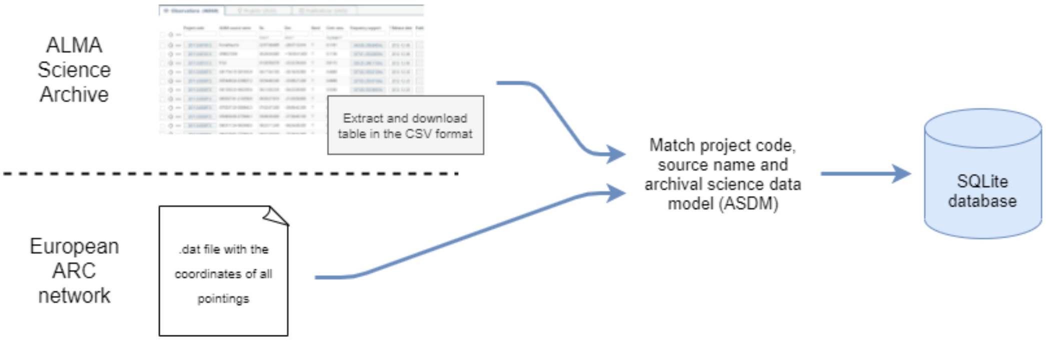

ALMA Science Archive
Extract and download table in the CSV format
European ARC network
.dat file with the coordinates of all pointings
Match project code, source name and archival science data model (ASDM)
SQLite database

Figure 3.3: The two data sources that ASH relies on. As described in the paragraph above, the pointings file must be obtained externally, whereas the rest of the data is fetched from the ALMA Science Archive in a CSV file.

Database tables are stored under the models of each application. Models that are accessed by more than one application were placed under the *common* app, whereas those that are only used internally are placed on their specific app – on this project, only the *sky_map* application required an internal data model.

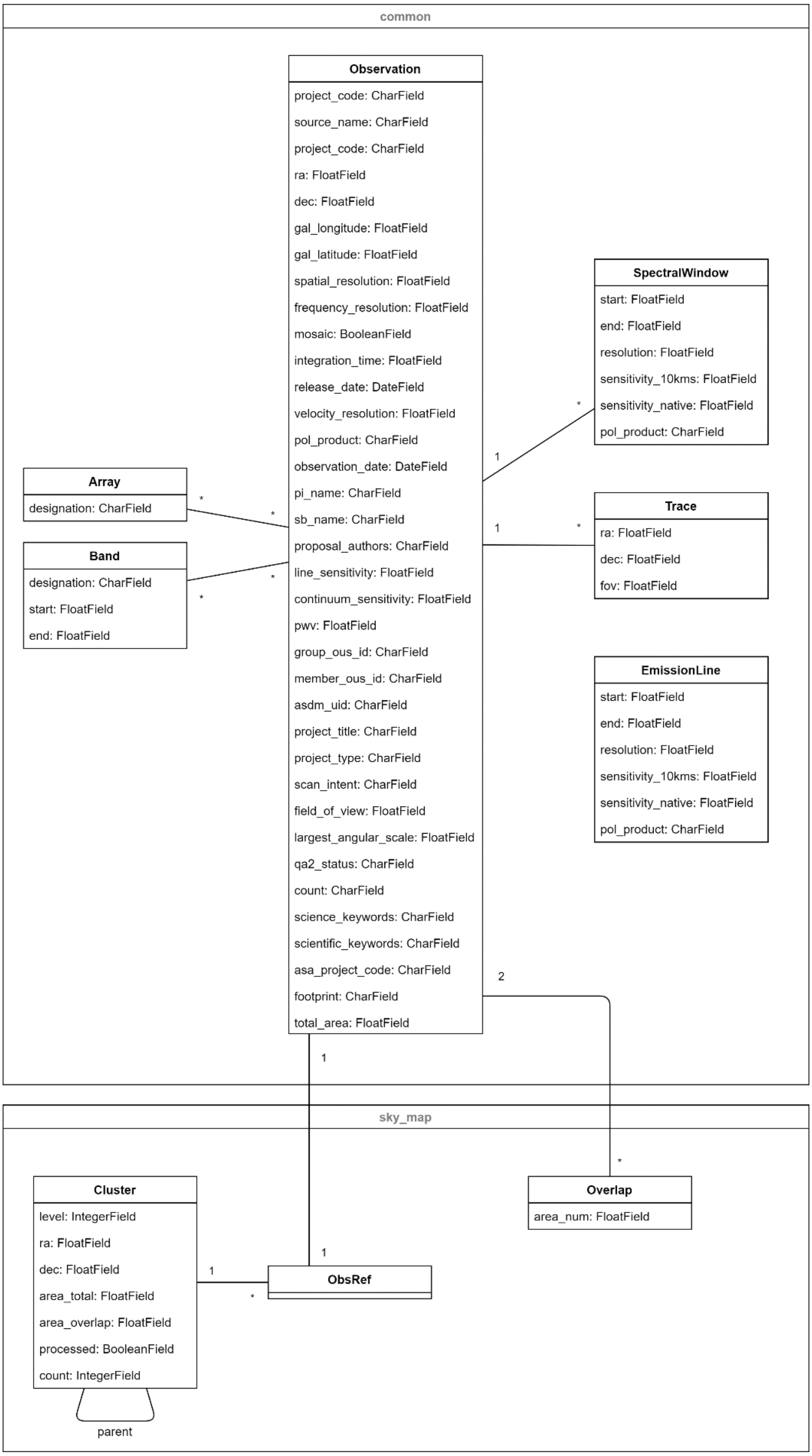


Figure 3.4: A diagram of the data models belonging to the common and sky map applications. Field types refer to the class names used by Django's ORM, which are then mapped to their respective field type on the chosen database framework. The overlap table is currently unused and was created to allow users to quickly obtain the list of observations each of them overlaps with.

Figure 3.4 contains a diagram of the data models created for the *common* and *sky_map* applications. As previously mentioned, many of the created entities are directly loaded from the CSV file obtained through the ASA, whereas certain fields and relationships require special handling.

3.1.3.1 Common

- **Observation**: this object represents an ALMA observation and includes all metadata that is exported from the Archive. Two of its properties, like the array configuration and the observed bands, are implemented through many-to-many relationships, as any observation can contain any number of either.

- **SpectralWindow**: this object, which is assigned to a single observation, encodes a frequency band that the observation was taken at, and includes the start and end frequencies, the resolution, the sensitivity (both native and normalized to 10km/s) and the polarization product. Since each spectral window is only assigned to one observation, the relationship is implemented through a foreign key field.

- **Array**: the array configurations used by the observation. It is expressed in a CharField object that encodes a string and is constricted to the following values: “12”, “7” and “TP” – respectively, the 12-meter array, the 7-meter array or the total power array.

- **Bands**: the frequency bands this observation was taken at. On the field plot tool, the user can query observations based on their covered bands; if the redshift range is left at zero (e.g., the user wants to query the rest frame frequency interval of the selected bands) this field is directly used to filter bands.

- **Trace**: the pointings an observation is composed of. Single observations only contain one pointing, while mosaics have multiple. The former will reuse the right ascension and declination values from the observation, and the latter will fetch the pointings from the traces file.

- **EmissionLine**: the different emission lines that are displayed to the user in the field plot tool. These were provided in a file created by the IA, which includes the most studied emission lines in respect to extragalactic science.

3.1.3.2 Sky map

- **Cluster**: any of the observation clusters that can be viewed on the sky map page. These represent groups of spatially close ALMA observations and are created with a recursive method based on the k-means algorithm, with a bottom-up approach that starts on the highest level-of-detail (observations) and ends on the lowest. As such, each cluster points to its parent on the next level-of-detail cluster set and contains information on the total coverage and overlapping area of its children.

- **ObsRef**: the relationship between an observation and the cluster it belongs to. This entity was added in to prevent a cyclical dependency between both data models, as the one-to-one relationship would otherwise need to be implemented through a foreign key on the Observation object.

- **Overlap**: the overlapping area between two observations. While currently unused, this table was added in to support a future revision of the sky map that, for instance, would allow users to quickly identify the observations that overlap with any other.

3.1.4 Virtual environments and execution

The ALMA Science Hub runs on a Python environment containing all required libraries, which makes the project easy to setup on any machine. During its development, ASH was tested on two separate Windows systems and will be deployed on a machine using the Fedora Linux distribution, for which a virtual environment already exists. Due to some of its dependencies not being yet available for newer releases, the project runs on Python version 3.1.7.

To facilitate setup, the project folder includes a *requirements.txt* file that contains a list of all required libraries and their versions. Virtual environments can be created with the following command:

```
py -m venv <path-to-environment>\<environment-name>
```

or, if the system has other Python versions installed,

```
py -m virtualenv -p=<python-executable> <path-to-environment>\<environment-name>
```

Afterwards, the environment can be activated with

```
<path-to-environment>\Scripts\activate
```
(Windows)

```
. <path-to-environment>\bin\activate
```
(Linux)

and its dependencies installed from the requirements file through

```
pip install -r <path-to-requirements>\requirements.txt
```

Running the project requires the execution of the Django server as well as the Redis message broker and the Celery worker server (3.2.2.4). For convenience, all necessary virtual environment activations and executions are handled by a set of callable batch files found in the project's root folder, as illustrated in Figure 3.1.

3.1.5 Project management

The proposal for this project did not envision a particular set of tools or design requirements, instead being driven by the need for a general toolset that would allow users to engage with the ALMA Science Archive and facilitate data exploration. Therefore, and considering the initial lack of experience in web development, the first phase focused on exploratory programming on webpage design, JavaScript and metadata-driven data plots generated and rendered by Python.

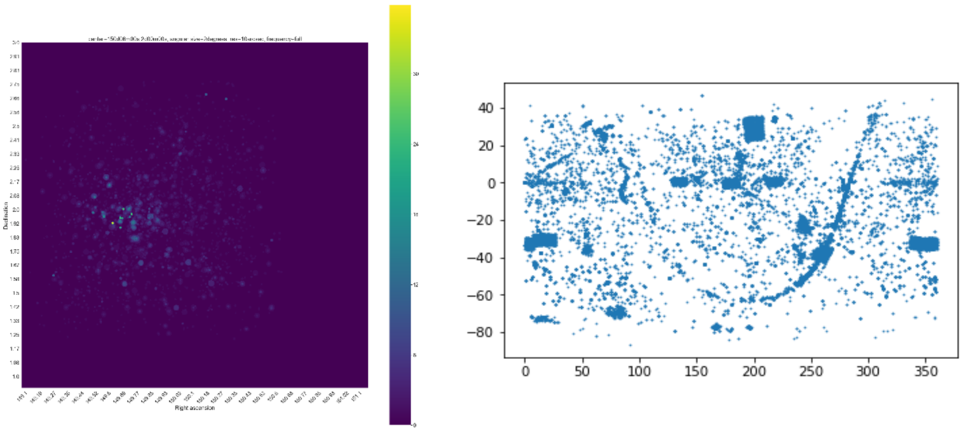


Figure 3.5: Early data analysis prototypes. Left: a pixelized rendering of the COSMOS field, highlighting overlapping observations. Right: observation density map of the whole sky. Both plots were created in Python with the Mathplotlib library.

The first application to be developed was the field plot tool, as its visualization suite – mainly as concerns the pixelated heatmap – was considered a direct follow-up to the previous Python-based prototypes. Additionally, this tool was considered the one with the most scientific potential.

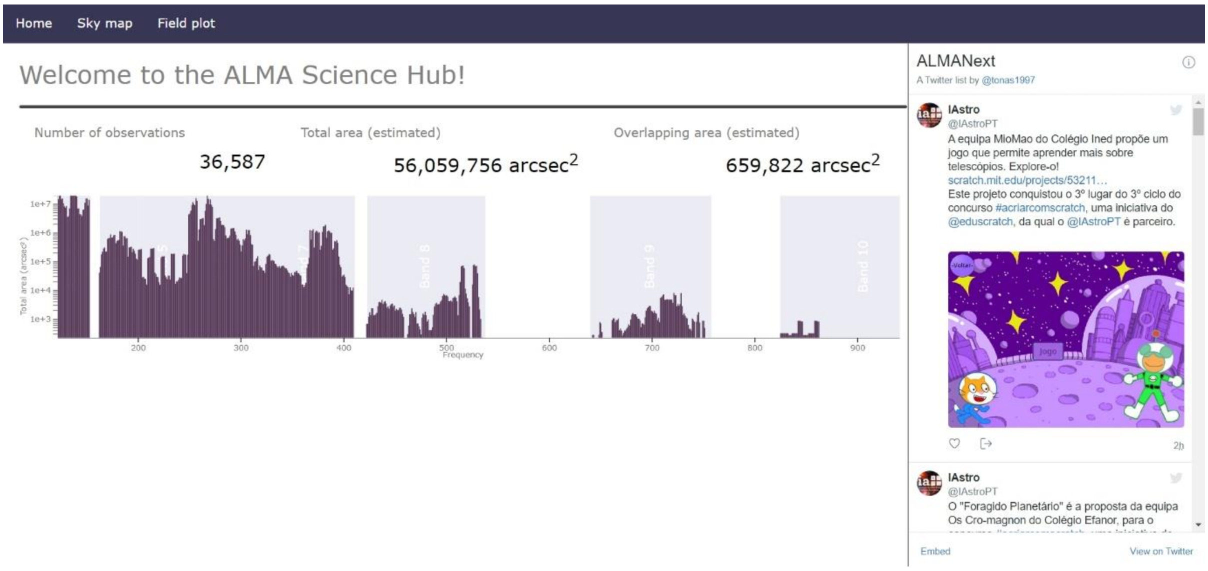
While the sky map application was originally planned to be implemented on the second development stage, the homepage ended up being integrated into the project before the sky map, as the latter required heavy data processing during which it was possible to work on both applications simultaneously.

The final phase of the project encompassed the deployment and testing of the web service, as well as the writing of the final report.

3.2 Web pages

This chapter explains in detail each web page that composes the ASH in terms of purpose, implementation, and technical aspects. These can be accessed through the top horizontal menu, as seen in the following screenshots.

3.2.1 Home



Welcome to the ALMA Science Hub!

Number of observations	Total area (estimated)	Overlapping area (estimated)
36,587	56,059,756 arcsec^2	659,822 arcsec^2

ALMANext

A Twitter list by @tonas1997

IAstro
@IAstroPT
A equipa MioMao do Colégio Ined propõe um jogo que permite aprender mais sobre telescópios. Explore-o!
scratch.mit.edu/projects/53211…
Este projeto conquistou o 3º lugar do 3º ciclo do concurso #acriarcomscratch, uma iniciativa do @eduscratch, da qual o @IAstroPT é parceiro.

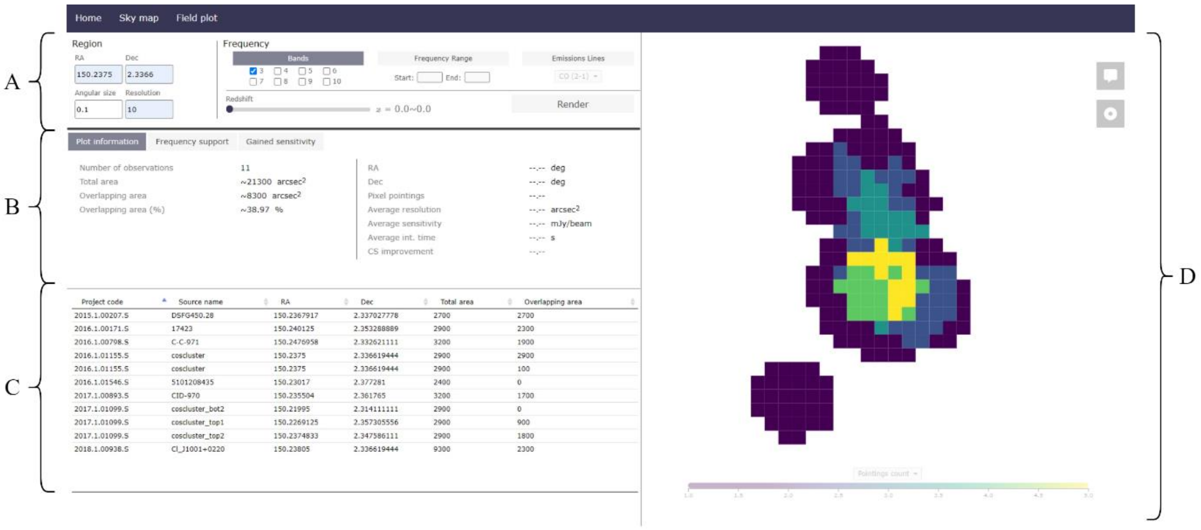
IAstro
@IAstroPT
O "Foragido Planetário" é a proposta da equipa Os Cro-magnon do Colégio Efanor, para o

Figure 3.6: The ALMA Science Hub main page. The main panel contains information on the archive's number of observations and total observed and overlapping areas. The histogram shows the total coverage area as a function of the observed frequencies. The rightmost panel contains a feed for the Twitter accounts of both ALMA and IA, which frequently engage in scientific outreach.

3.2.1.1 Purpose

This application serves as the tool's landing page. As shown in Figure 3.6, the user is greeted by metrics showing the archive's number of observations, total coverage area and total overlapping area. Additionally, a histogram displays total area coverage per frequency and band, and the right-hand panel features a Twitter feed for both ALMA and the Instituto de Astrofísica accounts. This widget, implementer was included to introduce users to the work being done by both institutions, as their feeds include regular updates on operational activity, job postings and scientific outreach to the community.

3.2.2 Field plot



Region

RA 150.2375 Dec 2.3366

Angular size 0.1 Resolution 10

Frequency

☑ 3 □ 4 □ 5 □ 6
□ 7 □ 8 □ 9 □ 10

Start: End:

Redshift z = 0.0~0.0

Number of observations	11
Total area	~21300 arcsec^2
Overlapping area	~8300 arcsec^2
Overlapping area (%)	~38.97 %

RA	--.-- deg
Dec	--.-- deg
Pixel pointings	--.--
Average resolution	--.-- arcsec^2
Average sensitivity	--.-- mJy/beam
Average int. time	--.-- s
CS improvement	--.--

Project code	Source name	RA	Dec	Total area	Overlapping area
2015.1.00207.S	DSFG450.28	150.2367917	2.337027778	2700	2700
2016.1.00171.S	17423	150.240125	2.353288889	2900	2300
2016.1.00798.S	C-C-971	150.2476958	2.332621111	3200	1900
2016.1.01155.S	coscluster	150.2375	2.336619444	2900	2900
2016.1.01155.S	coscluster	150.2375	2.336619444	2900	100
2016.1.01546.S	5101208435	150.23017	2.377281	2400	0
2017.1.00893.S	CID-970	150.235504	2.361765	3200	1700
2017.1.01099.S	coscluster_bot2	150.21995	2.314111111	2900	0
2017.1.01099.S	coscluster_top1	150.2269125	2.357305556	2900	900
2017.1.01099.S	coscluster_top2	150.2374833	2.347586111	2900	1800
2018.1.00938.S	Cl_J1001+0220	150.23805	2.336619444	9300	2300

Figure 3.7: The field plot tool, showing the user input, information and render panels. The input area, where the user sets the parameters for the field plot, can be found in A). The plot information panels, split across a tabbed menu, are placed in the frame marked with B). An observation table can be found in C). The plot itself in rendered in the D) pane.

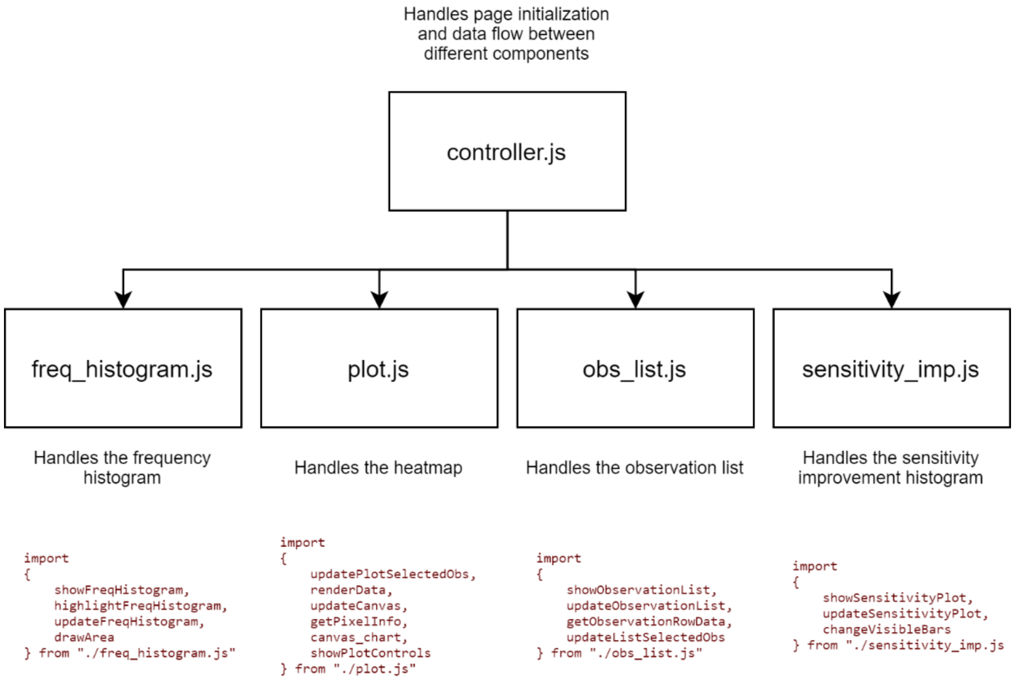
3.2.2.1 Purpose

The field plot tool was designed to support data analysis on specific regions observed by ALMA that can lead into the spatial or frequency combination of different observations. This is achieved by sampling discreet points in the region – vastly facilitating the sensitivity improvement calculations for both single-observation traces and complex mosaic shapes – which are then rendered as pixels on an HTML canvas. Figure 3.7 shows the layout of each component of this application.

Clicking on any pixel will select all observations that cover it, allowing the user to obtain estimates on the outcome of data combination on that region of the plot. Apart from the sensitivity improvement, it is also possible to visualize the frequency coverage of the selected pixel in respect to particular emission lines, which can also be redshifted by a small amount.

3.2.2.2 Code organization

During development and considering the number and complexity of semi-independent JavaScript/D3 components, the page's behavior was implemented through the ES6 standard, which allows for code to be split across multiple files. This is accomplished with import and export statements that can read and modify variables from other files. Initially, the page's JavaScript was organized as presented in Figure 3.8.


Handles page initialization and data flow between different components
controller.js
freq_histogram.js
plot.js
obs_list.js
sensitivity_imp.js
Handles the frequency histogram
Handles the heatmap
Handles the observation list
Handles the sensitivity improvement histogram

```
import
{
    showFreqHistogram,
    highlightFreqHistogram,
    updateFreqHistogram,
    drawArea
} from "./freq_histogram.js"
```

```
import
{
    updatePlotSelectedObs,
    renderData,
    updateCanvas,
    getPixelInfo,
    canvas_chart,
    showPlotControls
} from "./plot.js"
```

```
import
{
    showObservationList,
    updateObservationList,
    getObservationRowData,
    updateListSelectedObs
} from "./obs_list.js"
```

```
import
{
    showSensitivityPlot,
    updateSensitivityPlot,
    changeVisibleBars
} from "./sensitivity_imp.js
```

Figure 3.8: The previous relationship between the different modules of the field plot application. The central "controller" file imports variables and methods from other JS files, allowing for the page's visualization components to interact with eachother.

However, and as user interactivity became more complex over time, this approach became increasingly harder to maintain and develop, as visualization components became more tightly coupled with each other. In the end, the first two modules (*freq_histogram.js* and *plot.js*) were refactored and merged into the page's controller. The final two standalone modules (*obs_list.js* and *sensitivity_imp.js*) are expected to undergo a similar process before deployment.

3.2.2.3 User input parameters

Regions are rendered according to several properties:

Right ascension/declination – the coordinates, in the equatorial frame, of the plot's center point. The former supports both decimal degrees (e.g., 181.9708) and HH:MM:SS notations (e.g., 12:07:53). The latter accepts any float number between -90 and 90 degrees.

Region size – the region's angular size in degrees, equivalent to the side of the squared region which the user wants to render.

Resolution – the angular size of each pixel, in arcseconds.

Frequency – the frequency coverage that observations will be filtered against. The tool supports three query modes:

- Bands: one or more frequency bands, as defined by ALMA.
- Range: a custom frequency range as defined by start and end frequencies.

- Emission line: any of the provided emission lines in respect to their rest frame frequency.

Regardless of the chosen option, each and any observation that falls within the given parameters, no matter how small the spectral coverage overlap, will be included in the final plot.

The redshift slider found underneath the frequency modes panel allows the user to select a z-factor interval. Ranging from a factor of 0 (rest frame) up to 12 (an arbitrary distance somewhat depicting the limitations of current 8-10m ground based telescopes), this widget automates the calculation of the frequency shift associated with distant radio sources that, due to the expansion of the universe, are observed on Earth at lower frequencies. This phenomenon is known as *cosmological redshift*.

3.2.2.4 Rendering

When the user clicks on the "render" button – and the input is deemed valid – the plot's parameters are serialized in the JSON format and sent to the server, where they are used as query arguments to fetch observations from the database. While most fields are directly included in the query, frequency coverage parameters require special treatment; searching by bands without any redshift offset, for instance, allows the server to simply query observations through their band column, whereas user-defined frequency ranges and emission lines, with or without a non-zero redshift, require a more complex analysis of an observation's spectral windows.

After the search parameters are converted into a set of all needed database objects, observations are processed one-by-one and converted into a 2D pixel grid. Depending on user traffic and the size of the query set, this process can take up to a few dozen seconds; as such, the user is presented with a progress bar implemented through Celery, Redis and a standalone application implementing both front and back-end logic for progress bars. However, a separate JavaScript progress bar library was chosen over the one provided by the aforementioned application for its simplicity (Brunfeldt, 2014).

3.2.2.4.1 Celery

Celery (Solem, 2009) is an asynchronous task queue implemented in Python, allowing for concurrent jobs to be assigned to working units called *tasks*. In this project, however, only one task is used to build the plot to avoid concurrency errors caused by two tasks modifying the same pixel; nevertheless, moving the plot build algorithm to an asynchronous worker allows for its progress to be evaluated during execution. Since Django's QuerySet object format – which contains the result of a database query – can't be deserialized by Celery tasks, the latter obtain the plot's observations through a list of IDs that are handed over by the application's view.

3.2.2.4.2 Redis

Redis (2009) is a multifunctional data storage system that is usually employed as a key-value in-memory database, cache, or message broker. In the context of this project, Redis acts as a message broker between Django and Celery workers, therefore allowing the client to periodically obtain the status of the plot building task.

The relationship between client, server, Redis and Celery is outlined in Figure 3.9:

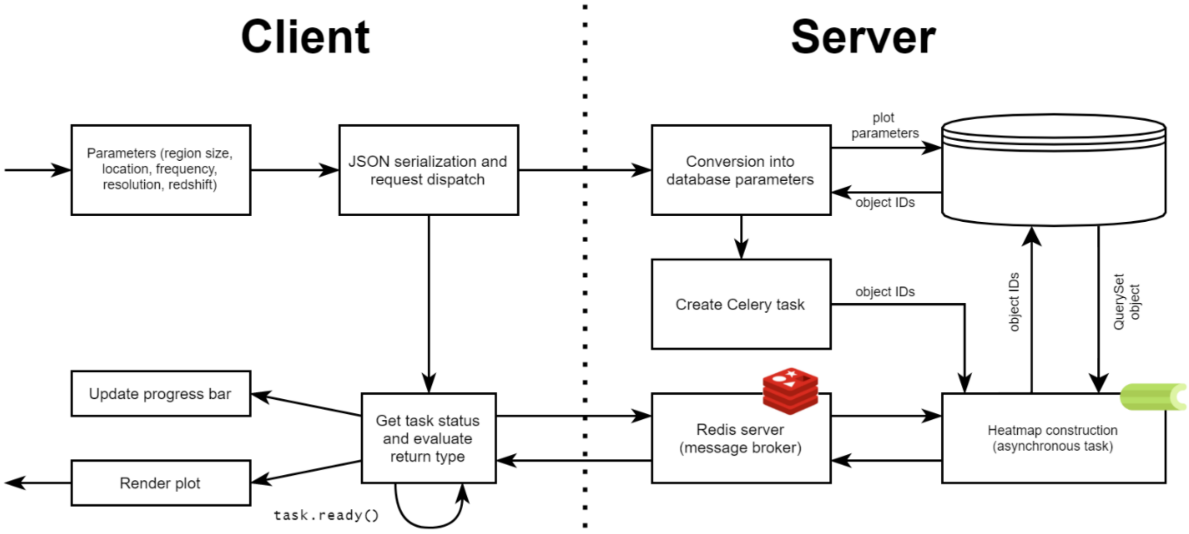

Client
Server
Parameters (region size, location, frequency, resolution, redshift)
JSON serialization and request dispatch
Conversion into database parameters
plot parameters
object IDs
Create Celery task
object IDs
object IDs
QuerySet object
Update progress bar
Get task status and evaluate return type
Redis server (message broker)
Heatmap construction (asynchronous task)
Render plot
task.ready()

Figure 3.9: The client-server interactions triggered by requesting a render of a given region. After a request is sent by the client, the server fetches data from the backend and kickstarts the plot construction task. In the meanwhile, the client starts sending progress information requests and updates the progress bar; when it receives the heatmap data, no further requests are made, and the visualization panes are initialized.

To keep argument processing and plot building logic separated, Celery tasks do not handle server requests on their own, even though this architecture necessitates two database accesses for the same client request. The performance impact of this second query, however, is minimal, as ID-based queries are relatively fast.

The progress bars were implemented on the front-end by the Celery Progress project (Zue, 2018), which handles the presentation layer of the bars.

3.2.2.5 Data serialization

As mentioned above, that data that is displayed on the front-end is encoded in a JSON file that includes the information that is contained in each of the plot’s pixels, apart from the metadata of all observations that are covered by the user’s input parameters and the list of spectral windows which will be displayed on the frequency coverage panel (3.2.2.7.2). The structure of this file is presented in Figure 3.10.

The *properties* object, containing metadata such as the number of observations or maximum combined sensitivities, was created to provide the page’s elements with information on the maximum and minimum values for each metric, which proved useful for the initialization of many of the page’s D3 graphs. More recent additions, such as the sensitivity improvement histogram, are able to calculate these dynamically and, as such, much of the information contained in the *properties* object is to be removed in the future.

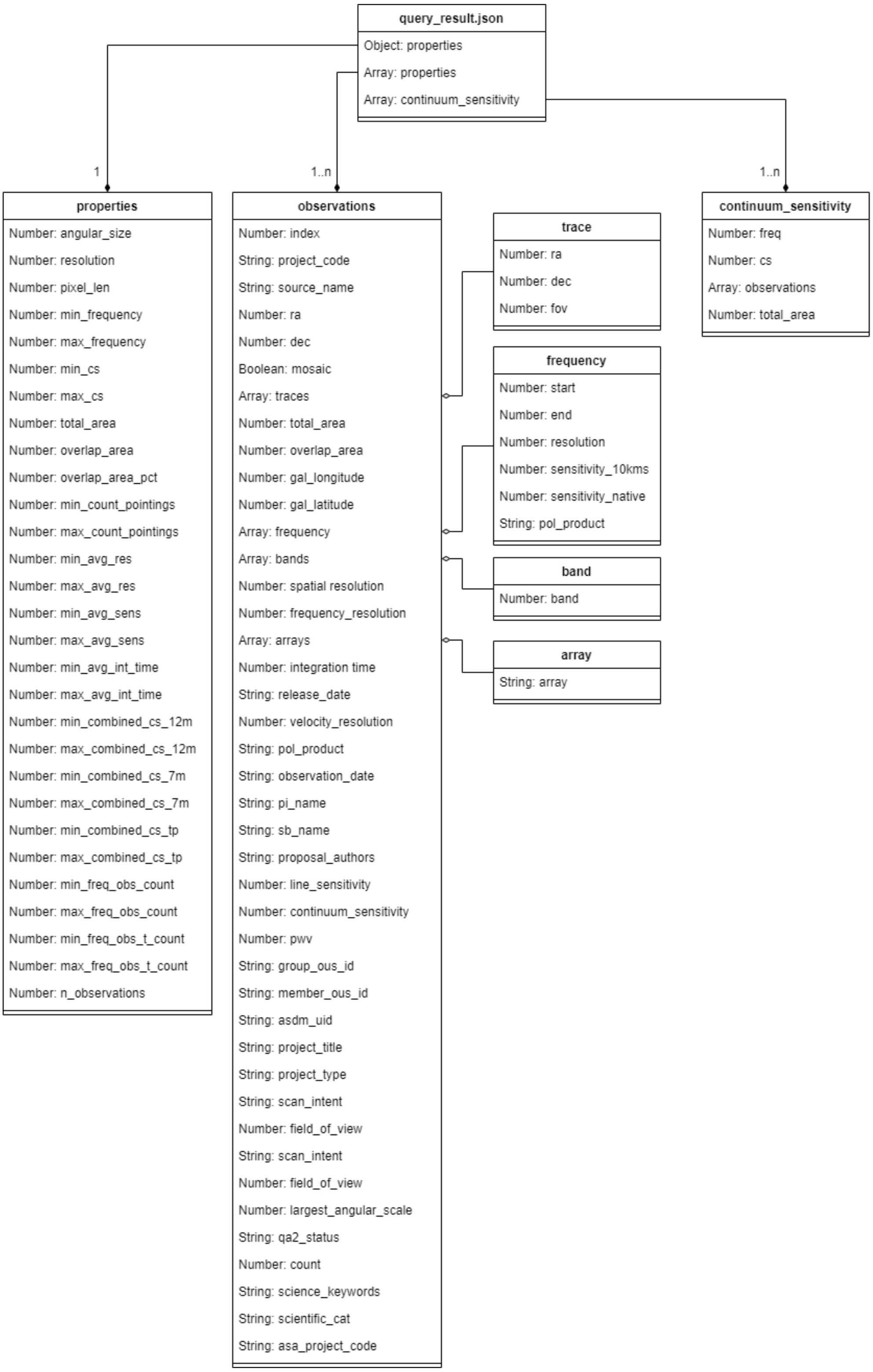


Figure 3.10: The structure of the field plot's JSON object. While this application ultimately only makes use of a subset of any observation's metadata, the file includes information from all the ASA table columns in case other fields are to be included in the field plot and displayed to the user in the future.

3.2.2.6 Canvas

The main visualization component of the page renders the result as a grid of pixels representing the relative areas, positions and overlaps of all found ALMA images for the given parameters. The plot supports zooming, panning, two visualization options shown on the upper-right corner of the frame and multiple render modes.

Unlike D3's native heatmap implementation, which consists of a multidimensional array of SVG objects bound to their specific datapoints, this plot is instead rendered as an HTML canvas. This alternative approach was chosen because of its better scalability to dataset length, therefore supporting larger plots – both in terms of angular size and resolution – without significant performance impacts. However, this improvement comes at the cost of implementation complexity: while D3's standard SVG object approach allows for easy interaction and data manipulation with each datapoint, regardless of whatever transformations are applied to the plot (e.g., zooming), canvas pixels require for dedicated functions that can, for instance, translate between the mouse's position and the corresponding pixel in the untransformed grid.

The upper-right buttons allow for the user to highlight pixels being covered by more than one footprint and to enable/disable a pixel tooltip that displays its position, combined sensitivity, average integration time and other such values. Clicking on a pixel will select and highlight any observations that happen to cover it.

3.2.2.6.1 Render modes

The plot supports five distinct render modes, each with a different color scale. The maximum and minimum values are calculated from each rendered dataset, and all scales go from the worst value of that metric to the best. All color scales were made different to improve the recognizability of each plot and are displayed in Figure 3.11.

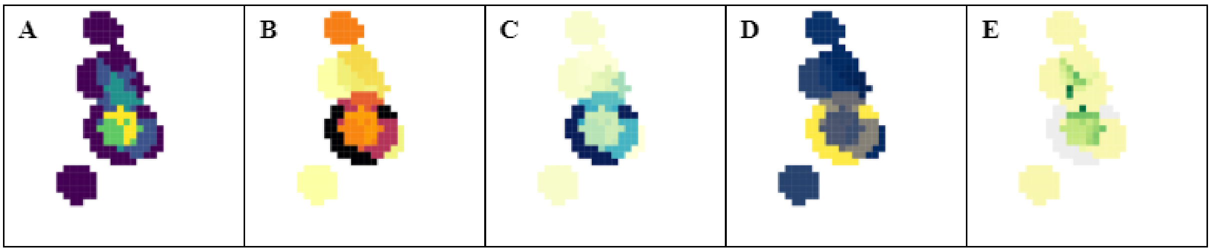


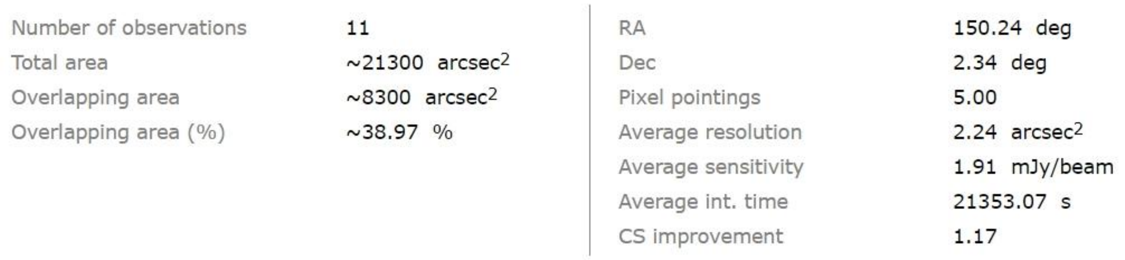
Figure 3.11: All five render modes for the plotting tool. A) shows observation count, B) average resolution, C) average sensitivity, D) average integration time and E) the sensitivity improvement factor for the selected array configuration.

The sensitivity improvement factor map uses a diverging color scale, since it is possible for combined observations to yield worse results that those that can be achieved by just considering the observation with the best sensitivity taken with a different array configuration. Red represents an increase in noise, yellow means there's little-to-no improvement and green signifies a decrease in noise.

3.2.2.7 Plot information panels

On the left side of the screen, underneath the input parameters panel, the user can select between three visualization tools. These are implemented by jQuery's Tabs widget.

3.2.2.7.1 Plot information



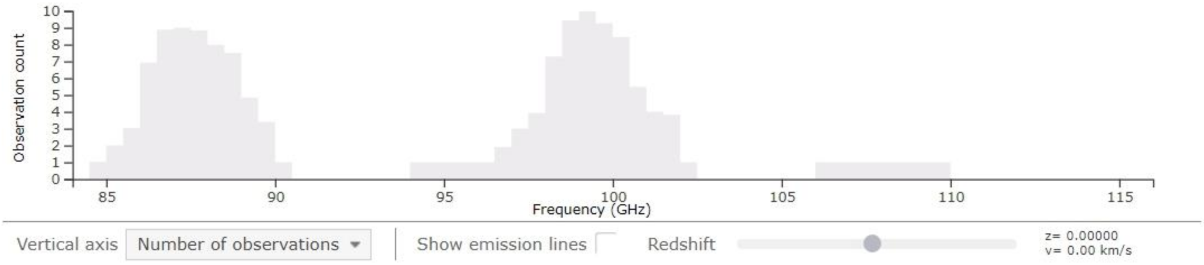
Number of observations	11
Total area	~21300 arcsec^2
Overlapping area	~8300 arcsec^2
Overlapping area (%)	~38.97 %

RA	150.24 deg
Dec	2.34 deg
Pixel pointings	5.00
Average resolution	2.24 arcsec^2
Average sensitivity	1.91 mJy/beam
Average int. time	21353.07 s
CS improvement	1.17

Figure 3.12: The plot information panel. The left pane contains information on the full plot: total number of observations, covered area and overlapping area, both in absolute and relative values. The right pane shows information pertaining to the currently selected pixel, including location, the average values for resolution, sensitivity and integration time measured between all covering observations and the sensitivity improvement factor that was estimated for the selected array configuration.

This panel, as shown in Figure 3.12, displays general properties for both the plot (left side) and the pixel the user is hovering with the mouse, if any (right side). The latter is updated as the user moves the mouse around the canvas.

3.2.2.7.2 Frequency coverage



Vertical axis Number of observations ▾

z= 0.00000
v= 0.00 km/s

Figure 3.13: The frequency coverage panel. The user can plot either the number of observations or coverage area as a function of frequency. Additionally, it is possible to display any covered emission lines within the frequency range and offset them by changing the redshift factor. This allows the user to assess whether a particular emission line maintains the same degree of coverage within this field across a particular redshift range.

The second panel, displayed in Figure 3.13, shows the frequency coverage for the generated plot as a histogram, including any emission lines found within the set frequency interval, in terms of either observation count or coverage area.

As it happens with the field heatmap, the frequency coverage histogram is also potentially vulnerable to the performance issues that come with larger datasets. Since coverage is measured in 0.01 GHz increments, wider bands might result in thousands of buckets being rendered at once, which can result in low responsiveness to user interaction – a problem which is obviously aggravated when multiple, widely spaced bands are being queried. One possible solution is to aggregate data points in levels-of-detail which are rendered according to the user's desired magnification level and scale range, not only improving performance by constraining the maximum number of SVG elements being loaded into the page's DOM (Plotting 50 Million Points with D3, 2018) but also decreasing visual noise on charts with a significant number of datapoints.

The adopted levels-of-detail approach keeps data binding between SVG elements and data points, saving considerable development time that would otherwise be spent adding user interactivity through reverse transforming of the plot's zoom and panning levels, as it had happened with the plot's canvas implementation.

This solution consists of limiting the maximum number of rendered SVG objects by choosing which parts of the dataset ought to be visible on the plot's current viewport, which changes with zoom and panning actions. If the plot's number of 0.01 GHz frequency buckets (corresponding to the maximum detail level) exceeds a threshold, these are instead placed into equally sized bins in a recursive process that builds a levels-of-detail hierarchy, as explained in Figure 3.14:

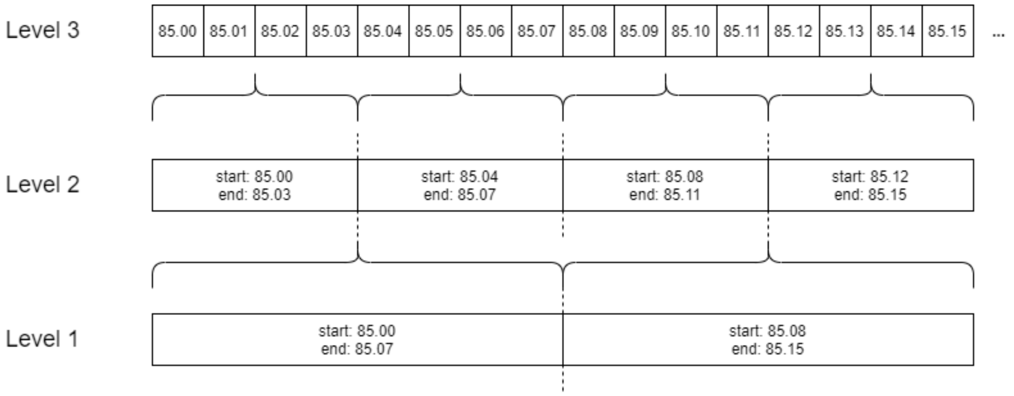


Figure 3.14: A simplified diagram explaining the LOD approach to rendering frequency buckets.

The number of levels-of-detail is given by the expression:

$$l(n) = \lfloor log_{10} n \rfloor$$

3.1

where *n* is the number of 0.01 GHz increments that fit within the plot's frequency range. The level-of-detail dataset that is rendered at a given transform scale factor *k* is therefore defined by the formula:

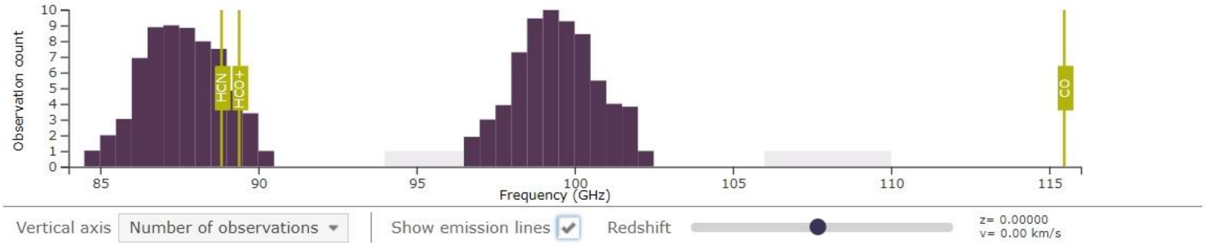
$$d(k) \approx (\log_{10} k + 1)$$

3.2

with the added constant offsetting the zoom factor at which a LOD will transition to the next.

The bottom row contains visualization controls that allow the user to plot frequency coverage against either observation count or area and to assess spectral line coverage at a particular Doppler redshift (as selected on the right-hand slider). Doppler redshift is usually of interest when studying rotating galactic disks and other objects with high radial speeds and, as such, are limited to ±100 km/s.

Figure 3.15 shows how any selected observations will also have their frequency windows highlighted on the plot:



Vertical axis: Number of observations

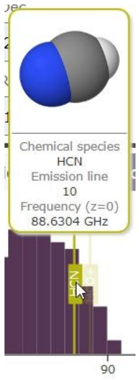
Show emission lines ☑

Redshift

z= 0.00000
v= 0.00 km/s

Figure 3.15: The frequency coverage panel, with selected frequency buckets. These can be selected by clicking on the bars themselves, a pixel on the plot or an observation on the list.

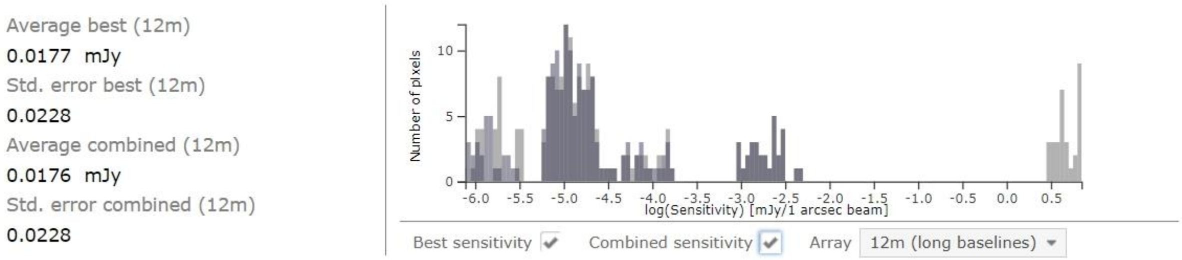
To avoid information overload on the UI, only the designation of the chemical species related to each emission line is displayed; placing the mouse over any of them will show an info card that shows a 3D render of the molecule, its Doppler redshift, and its line designation. The 3D renderings of the molecules were produced with the web based MolView tool (molview.org). Figure 3.16 shows the info card that is displayed to the user if an HCN emission line on band 3 is hovered with the cursor.



Chemical species
HCN
Emission line
10
Frequency (z=0)
88.6304 GHz

Figure 3.16: The information card that is shown when an emission line is hovered. The "frequency" field responds to the user's configured Doppler redshift.

3.2.2.7.3 Sensitivity improvement



Average best (12m)
0.0177 mJy
Std. error best (12m)
0.0228
Average combined (12m)
0.0176 mJy
Std. error combined (12m)
0.0228

Best sensitivity ✓

Combined sensitivity ☑

Array: 12m (long baselines)

Figure 3.17: The sensitivity improvement panel. The left pane shows the average and error values for both the best and estimated combined sensitivities. The histogram on the right plots pixel count against sensitivity, with the user being able to toggle the visibility of either value sets. Different array configurations are chosen with the selection box on the lower right corner of the histogram.

The third and final pane of the field plot tool, displayed on Figure 3.17, was designed to facilitate data combination between different observations. The histogram shows the relationship between coverage area and the logarithm of the line sensitivity, with each pixel contributing with two values:

- The highest (equivalently, best) sensitivity found across all observations that cover that pixel.

- The highest sensitivity that can be achieved by combining a subset of all observations that cover that pixel, depending on the chosen array configuration.

While sensitivity can theoretically be calculated among observations taken by different array configurations, the estimates for the combined sensitivity of 7m and 12m-array data will generally be worse than the sensitivity of the latter, since the compact 7m array has a higher overall sensitivity that translates to noisier images. Therefore, each pixel contains information for the best sensitivity, the combined sensitivity among all 12m observations that cover it and the combined sensitivity for all 7m observations that cover it. The array configurations can be switched on the panel's selection menu on the right.

The sensitivity improvement factor can also be visualized on the pixelated heatmap if the "CS improvement factor" render mode is selected on the plot. The color scale will show the combined sensitivity improvement factor among all observations taken with the chosen array configuration, as seen on Figure 3.18:

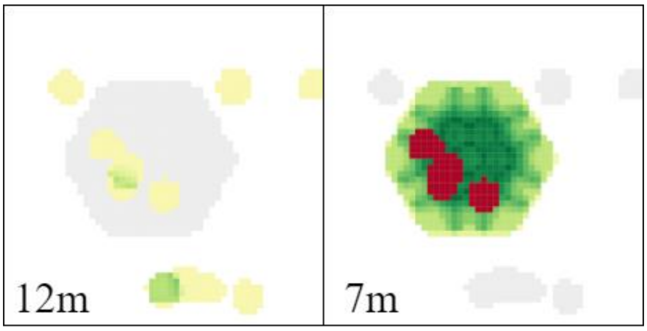
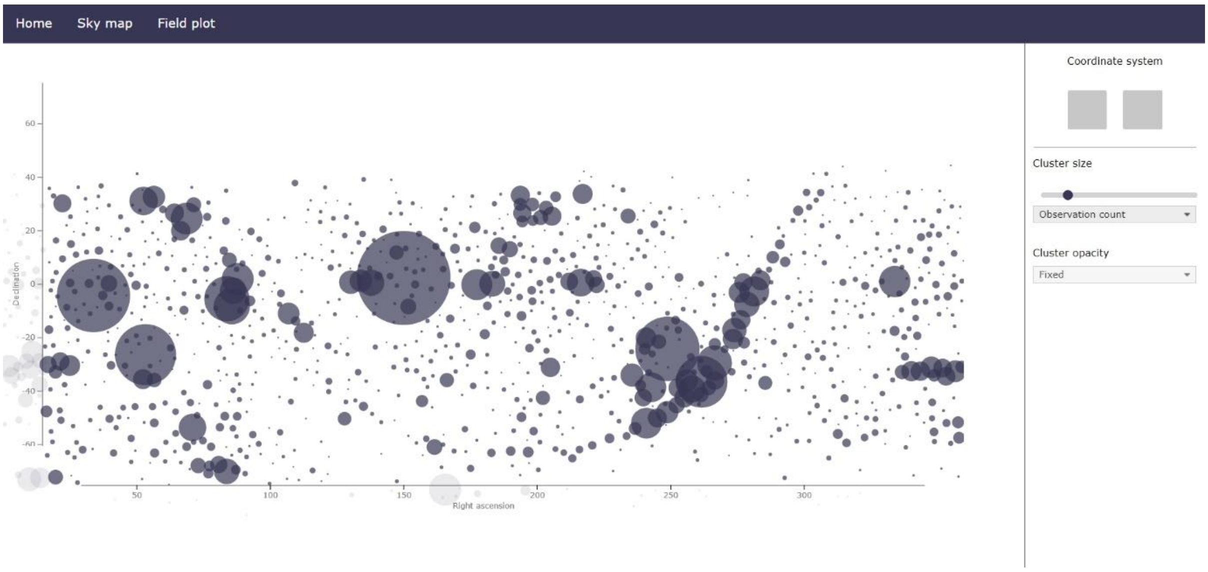


Figure 3.18: The sensitivity improvement heatmap with two different array configurations. The central mosaic was observed with 7-m antennas, while the surrounding observations were taken with the 12-m array. Green symbolizes an improvement in sensitivity if all observations on the selected array configuration are combined, red signifies a degradation in image quality and grey indicates coverage by observations on a different array configuration.

3.2.2.8 Observation table

The bottom section of the field plot main page contains a list of observations implemented through the DataTables (2014) jQuery plugin, which implements fully customizable dataset tables. This interface element was included to allow users to quickly obtain and select any observations through their total coverage and overlap areas, as well as their project codes and source names. While it is possible to include an arbitrary number of columns on the widget – limited only by the information that can be obtained on the ASA – the current implementation offers a more condensed set of metadata that is of particular interest to the field plot tool.

3.2.3 Sky map



Cluster size
Observation count

Cluster opacity
Fixed

Figure 3.19: The sky map application. The "coordinate system" buttons on the top-right corner are currently non-functional but are expected to eventually allow the user to switch between galactic and equatorial coordinate frames.

3.2.3.1 Purpose

The sky map tool was thought of as corresponding to the highest level of data exploration provided by the ASH. It consists of a scatterplot representing the celestial sphere – rendered under rectangular projection – where the user can easily identify which regions contain a higher level of ALMA coverage. These regions are represented by clusters of observations in increasing levels-of-detail, aiming to improve zoom/panning performance by limiting the number of simultaneous SVG objects being rendered by the DOM. Figure 3.19 shows the application's page.

The user can choose to render each cluster's opacity and scale, allowing for the simultaneous visualization of a region's observation count and overlapping area, for instance. This feature is intended to highlight the tool's role as the first step in the data analysis workflow, providing users with a broad view of which regions have the most potential for data combination.

3.2.3.2 Clustering

As mentioned above, observations (corresponding to the lowest LOD) are aggregated in clusters which, in turn, are also clustered and placed in a LOD dataset. This strategy was driven by the performance limitations that come with D3's standard data visualization techniques: since each datapoint – in this case, observations – corresponds to an SVG object encoding position and style information, rendering a large dataset at once results in diminishing responsiveness to user interactions such as panning and zooming; thus, data reduction becomes a useful approach by both improving performance and increasing clarity by lowering the number of simultaneous datapoints being displayed at any given time. The position of each cluster was calculated with the unsupervised *k-means* algorithm.

As such, clusters are organized in levels-of-detail where the area values of each cluster correspond to the sum of the areas of its children. Due to the high number of datapoints, it was possible to perform area calculations with a multithread pool, with each worker calculating the area of a single cluster at any given time. This is illustrated in Figure 3.20.

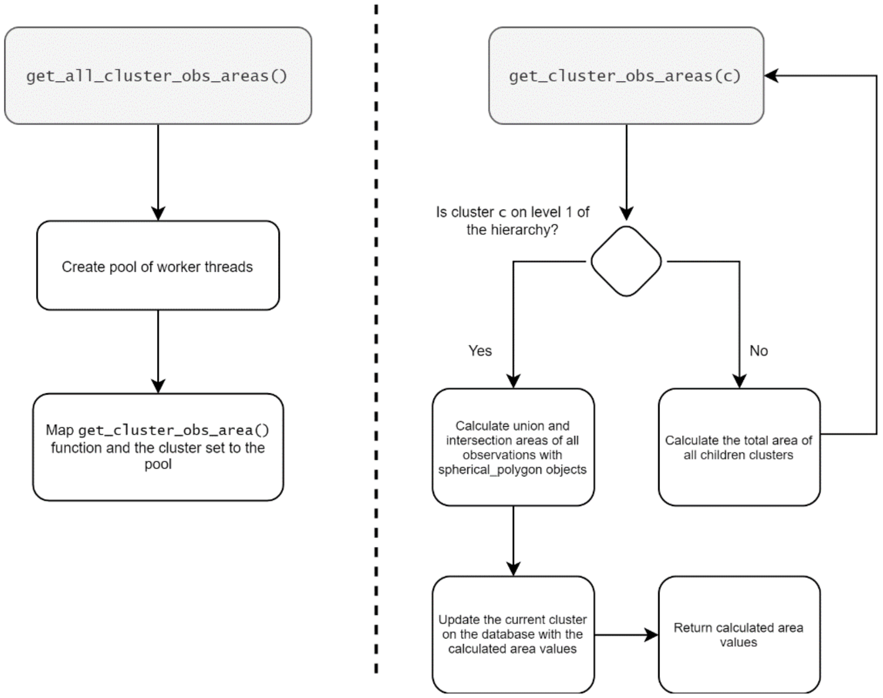

get_all_cluster_obs_areas()
Create pool of worker threads
Map get_cluster_obs_area() function and the cluster set to the pool
get_cluster_obs_areas(c)
Is cluster c on level 1 of the hierarchy?
Yes
No
Calculate union and intersection areas of all observations with spherical_polygon objects
Calculate the total area of all children clusters
Update the current cluster on the database with the calculated area values
Return calculated area values

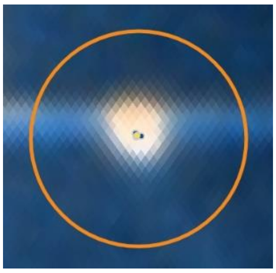
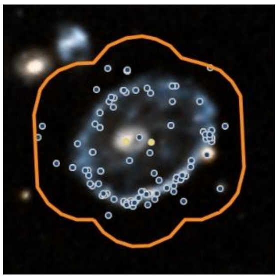
Figure 3.20: The procedure through which the total and overlap areas of each cluster is calculated. Level-1 corresponds to the cluster dataset for which there are no children, corresponding to the highest zoom factor and, consequently, highest detail: at this level, observations are directly mapped to clusters – and vice-versa – through the ObsRef *object (3.1.3.2).*

As mentioned above, each cluster contains an aggregate of the total coverage and overlapping area of its children. Calculating these values on the observation level posed an outstanding challenge as, apart from the absence of simple analytical solutions for area calculation in spherical geometry even for simple cases such as the intersection area of two spherical caps, e.g., two overlapping single-pointing observations, mosaic observations typically have highly complex shapes, often resulting from the intersection of dozens of pointings.

Due to time and processing constraints, exact solutions were precluded in favor of area approximations that could be obtained with the *spherical_geometry* (Droettboom, 2015) Python library. This toolkit features functions that calculate union and intersection areas of a set of spherical polygons which, in turn, can be constructed with either a cone, which is defined by right ascension and declination of the center of the observation plus its field-of-view, or a list of spatial locations that define the polygon.

While the former can be directly obtained from each observation's metadata, the latter must be constructed with individual points that, when connected, form the outline of a mosaic. Fortunately, such a structure is *also* provided by the ALMA Science Archive: the site's viewport, that makes use of the ALADIN (ALADIN Sky Atlas, Consulted 2021) plugin, builds a spatial representation of each observation through a spherical polygon defined by a set of points. Therefore, it becomes possible to leverage the *spherical_geometry* library regardless of the type of observation, as shown in Table 3.2.

Table 3.2: The different polygon data formats and conversion processes for both single pointings and mosaic observations.

Type	ALADIN trace	Polygon data	Conversion process
Single pointing (Lup_9, project code 2011.0.00733.S)		UNION (Circle ICRS 242.118375 -39.092278 0.003480)	Extract numerical values, respectively right ascension, declination and radius, and call *SphericalPolygon.from_cone()*
Mosaic (Cartwheel Galaxy, project code 2012.1.00720.S)		UNION (Polygon ICRS 9.426943 -33.727705 9.425231 -33.729525 9.422860 -33.730865 9.420107 -33.731531 9.417241 -33.731456 9.413572 -33.730135 9.410974 -33.727705 9.408057 -33.727636 9.404864 -33.726585 9.403094 -33.725366 9.401745 -33.723821 9.400612 -33.720149 9.401257 -33.709705 9.402717 -33.707653 9.404867 -33.706076 9.407498 -33.705130 9.411547 -33.704213 9.413574 -33.702527 9.416128 -33.701444 9.418383 -33.701086 9.421236 -33.701311 9.424343 -33.702527 9.426941 -33.704958 9.429857 -33.705025 9.433049 -33.706076 9.434819 -33.707294 9.436169 -33.708839 9.437303 -33.712511 9.436662 -33.722955 9.434822 -33.725366 9.432557 -33.726828 9.429860 -33.727636)	Extract numerical values, create a list of tuples containing the right ascension and declination values of each point and call *SphericalPolygon.from_radec()*

When calculating the area values for a cluster, two new polygons must be created, both operating on the cluster's assigned observation set:

- Union polygon of all observations, whose area corresponds to the cluster's total area,
- Union polygon of all intersection areas, whose area corresponds to the cluster's total overlapping area

As each observation belongs to one and only one cluster, this approach doesn't yield exact results, as intersections between observations belonging to different clusters won't be accounted for. However, and given the purpose of the sky map tool – which is designed to highlight dense observation areas in relation to one another, not necessarily providing exact values upfront – this discrepancy was deemed acceptable.

4 Use case examples

This chapter presents use cases that aim to guide the user in answering scientific questions through the ASH platform. Additionally, a general walkthrough of the tool – leveraging both the sky map and the field plot tools – is also provided.

4.1 Toolset overview

The core functionality of the developed toolset is to provide a direct path towards data exploration. This naturally involves both the sky map and field plot applications, where the user will usually follow a general workflow that can be roughly split into two steps:

1. Find a region (equivalently, cluster) in the sky with a high percentage of overlapping area, obtained through the sky map application.

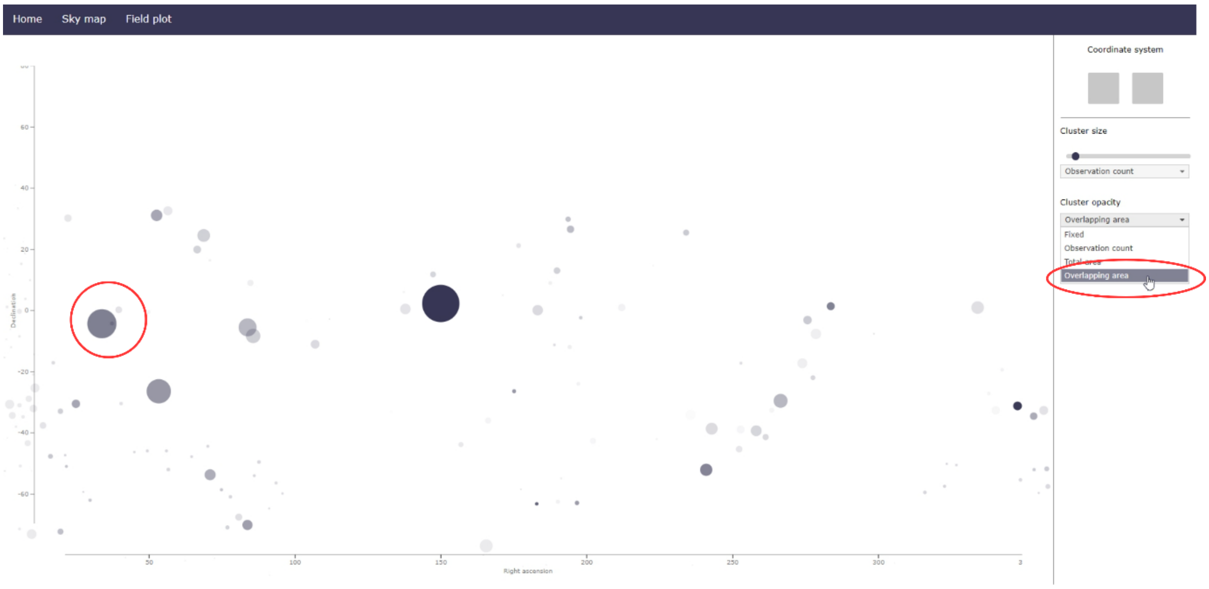
2. Obtain sensitivity gain estimates on a plot containing the previously identified cluster and observe coverage on different emission lines and/or spectral bands.

The following walkthroughs guide the user in identifying a cluster of overlapping observations covering bands 3 and 6, which can be downloaded at a later stage from the ALMA Science Archive.

4.1.1 Sky map

Data exploration takes on a top-down approach, where the user starts by working on the largest possible scale (full sky coverage) and then obtains more detailed insights into regions of their particular interest. As such, the user starts by switching to the sky map application and selecting the “overlapping area” opacity plotting option – as shown in Figure 4.1 – which will promptly highlight regions with higher observation densities. The central, darker blob corresponds to the COSMOS field[1]: an equatorial, multispectral survey studying the formation and evolution of galaxies across multiple redshift ranges.

[1] https://cosmos.astro.caltech.edu/



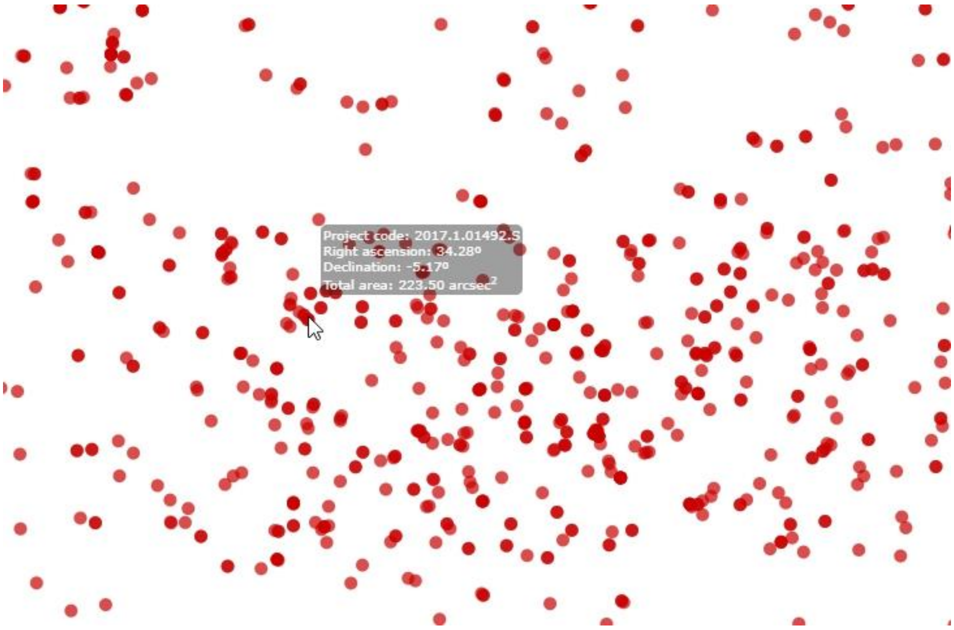
Cluster size

Observation count

Cluster opacity

Overlapping area

Figure 4.1: The sky map application can be used to highlight regions of the sky of particular interest to data combination. In this case, cluster opacity was plotted against the total overlapping area found within (right red shape), which highlighted a cluster that will be explored in further detail (left red shape).



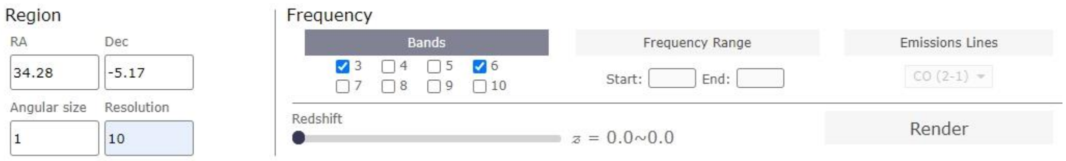
Project code: 2017.1.01492.S
Right ascension: 34.28º
Declination: -5.17º
Total area: 223.50 arcsec^2

Figure 4.2: Zooming into the chosen cluster will eventually reveal the relative positions of each of its observations. Hovering the mouse over one of the observation groupings displays a tooltip showing the coordinates of that particular observation. These coordinates can later be used to generate a query on the field plot tool.

As the user zooms in on the identified cluster, it will start to break down into smaller and more spatially accurate groupings until observations are rendered individually as red dots. More opaque regions indicate regions with a high number of nearby observations and, as shown in Figure 4.2, a small tooltip will present the user with that observation's project code, coordinates, and total area. This information will then be fed into the field plot tool.

4.1.2 Field plot

The sky map application provided the user with the spatial coordinates of an observation cluster with scientific potential. After switching to the field plot tool on the top navigation bar, the location parameters are input in the left side of the panel shown in Figure 4.3; to provide some leeway, the plot will generate on a 1-square degree field, while resolution is set to 10 arcseconds. Since the initial scientific use case aimed to explore bands 3 and 6 of the electromagnetic spectrum, the frequency coverage options were filled in as displayed in Figure 4.3:



Region

RA: 34.28 Dec: -5.17

Angular size: 1 Resolution: 10

Frequency

Bands: ☑ 3 □ 4 □ 5 ☑ 6 □ 7 □ 8 □ 9 □ 10

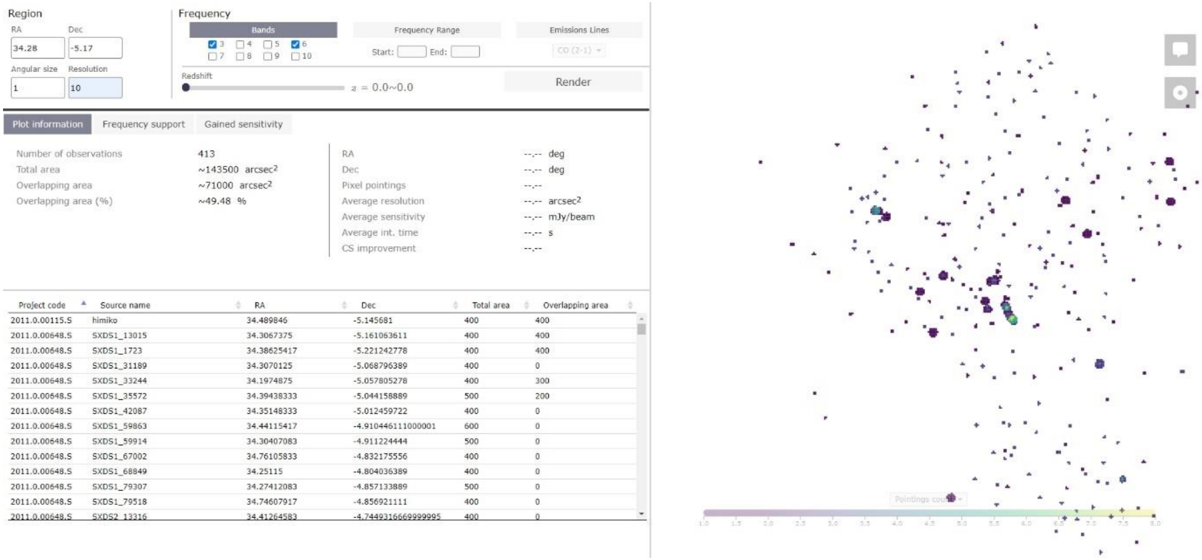
Frequency Range: Start: End:

Emissions Lines: CO (2-1)

Redshift: z = 0.0~0.0

Render

Figure 4.3: The coordinates for the previously selected observation are then input in the field plot tool, which was opened on a separate tab. While the sky map tool doesn't provide with frequency coverage information yet, bands 3 and 6 were chosen for illustrative purposes.



Region

RA: 34.28 Dec: -5.17

Angular size: 1 Resolution: 10

Frequency

Bands: ☑ 3 □ 4 □ 5 ☑ 6 □ 7 □ 8 □ 9 □ 10

Frequency Range: Start: End:

Emissions Lines: CO (2-1)

Redshift: z = 0.0~0.0

Render

Plot information | Frequency support | Gained sensitivity

Number of observations	413
Total area	~143500 arcsec^2
Overlapping area	~71000 arcsec^2
Overlapping area (%)	~49.48 %

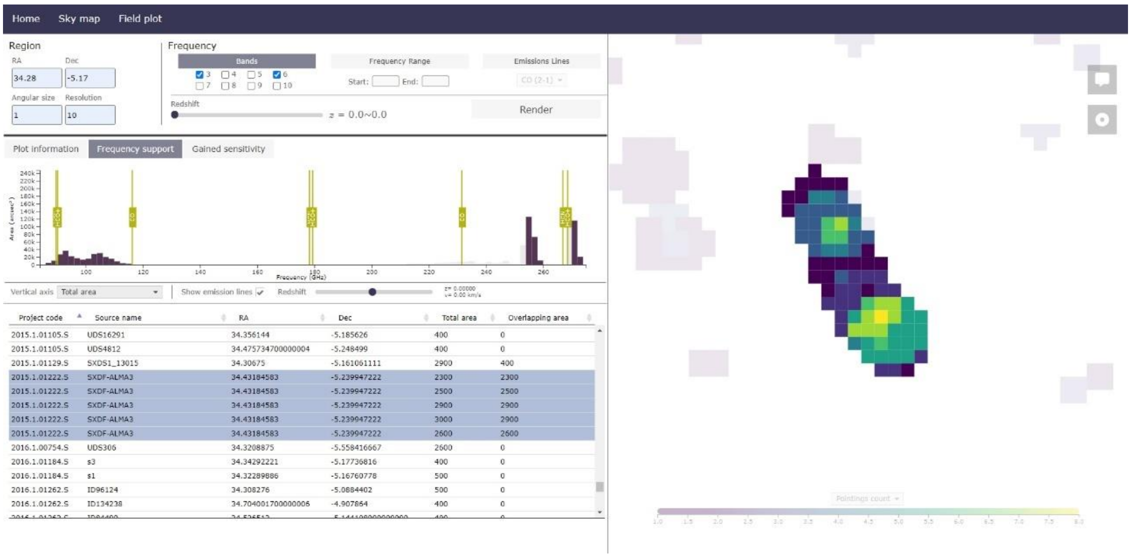
RA	--.-- deg
Dec	--.-- deg
Pixel pointings	--.--
Average resolution	--.-- arcsec^2
Average sensitivity	--.-- mJy/beam
Average int. time	--.-- s
CS improvement	--.--

Project code	Source name	RA	Dec	Total area	Overlapping area
2011.0.00115.S	himiko	34.489846	-5.145681	400	400
2011.0.00648.S	SXDS1_13015	34.3067375	-5.161063611	400	400
2011.0.00648.S	SXDS1_1723	34.38625417	-5.221242778	400	400
2011.0.00648.S	SXDS1_31189	34.3070125	-5.068796389	400	0
2011.0.00648.S	SXDS1_33244	34.1974875	-5.057805278	400	300
2011.0.00648.S	SXDS1_35572	34.39438333	-5.044158889	500	200
2011.0.00648.S	SXDS1_42087	34.35148333	-5.012459722	400	0
2011.0.00648.S	SXDS1_59863	34.44115417	-4.910446111000001	600	0
2011.0.00648.S	SXDS1_59914	34.30407083	-4.911224444	500	0
2011.0.00648.S	SXDS1_67002	34.76105833	-4.832175556	400	0
2011.0.00648.S	SXDS1_68849	34.25115	-4.804036389	400	0
2011.0.00648.S	SXDS1_79307	34.27412083	-4.857133889	500	0
2011.0.00648.S	SXDS1_79518	34.74607917	-4.856921111	400	0
2011.0.00648.S	SXDS2_13316	34.41264583	-4.7449316669999995	400	0

Figure 4.4: After the plot is rendered, the user is presented with the list of all queried observations and information on the plot's dataset. The heatmap also immediately reveals a grouping of overlapping observations near the center, which will be explored in more detail. The default color scale, as presented on the bottom of the field heatmap, shows the number of observations covering any given pixel.

After the "render" button is clicked on – and the plot is rendered – the remaining interface elements will be initialized, and the plot will appear on the right side of the page. From this point onwards, the user can delve deeper into any region of the plot without further assistance from the sky map tool. The first information panel that is presented to the user on the left side of the page shows information about the plot's coverage area; furthermore, mousing over a pixel on the plot will display its information on the same panel.

The user then clicks on a pixel that is connected to a series of overlapping observations - selecting them - and switches to the “frequency support tab” The result of these actions is shown in Figure 4.5.



Region

RA: 34.28 Dec: -5.17

Angular size: 1 Resolution: 10

Frequency

Bands: ☑ 3 □ 4 □ 5 ☑ 6 □ 7 □ 8 □ 9 □ 10

Start: End:

Redshift: z = 0.0~0.0

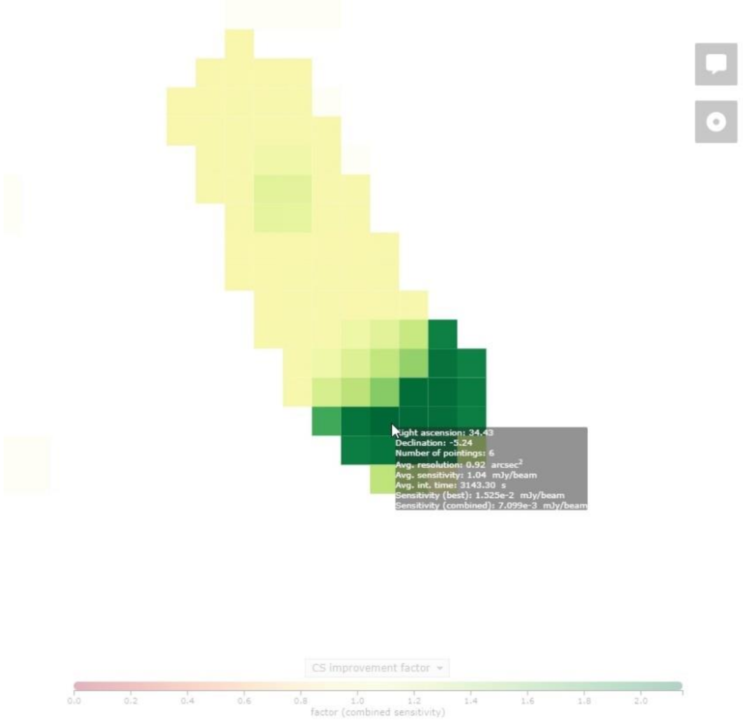
Vertical axis: Total area

Show emission lines ✓

Redshift: z= 0.00000 v= 0.00 km/s

Project code	Source name	RA	Dec	Total area	Overlapping area
2015.1.01105.S	UDS16291	34.356144	-5.185626	400	0
2015.1.01105.S	UDS4812	34.475734700000004	-5.248499	400	0
2015.1.01129.S	SXDS1_13015	34.30675	-5.161061111	2900	400
2015.1.01222.S	SXDF-ALMA3	34.43184583	-5.239947222	2300	2300
2015.1.01222.S	SXDF-ALMA3	34.43184583	-5.239947222	2500	2500
2015.1.01222.S	SXDF-ALMA3	34.43184583	-5.239947222	2900	2900
2015.1.01222.S	SXDF-ALMA3	34.43184583	-5.239947222	3000	2900
2015.1.01222.S	SXDF-ALMA3	34.43184583	-5.239947222	2600	2600
2016.1.00754.S	UDS306	34.3208875	-5.558416667	2600	0
2016.1.01184.S	s3	34.34292221	-5.17736816	400	0
2016.1.01184.S	s1	34.32289886	-5.16760778	500	0
2016.1.01262.S	ID96124	34.308276	-5.0884402	500	0
2016.1.01262.S	ID134238	34.704001700000006	-4.907864	400	0

Figure 4.5: Zooming in and clicking on the identified observation grouping and selecting the "frequency support" tab reveals that this set of observations fully covers HCN and HCO+ emission lines at rest frame (z=0), both lying inside band 3. The selected observations appear highlighted on the table.



Right ascension: 34.43
Declination: -5.24
Number of pointings: 6
Avg. resolution: 0.92 arcsec^2
Avg. sensitivity: 1.04 mJy/beam
Avg. int. time: 3143.30 s
Sensitivity (best): 1.525e-2 mJy/beam
Sensitivity (combined): 7.099e-3 mJy/beam

Figure 4.6: Switching to the "CS improvement factor" on the plot's scale selector shows that data combination on the bottom region of this cluster is predicted to result in a significant sensitivity improvement. The tooltip was enabled with the first of the top-right buttons. At this point, the user can make use of the tool to decide which observations to download from the ALMA Science Archive.

The final step of this use case consists of selecting the “CS improvement panel” render option, which will then plot each pixel according to the gained sensitivity that is potentially obtainable through combining all its covering observations. This gain is expressed as a ratio between the lowest sensitivity across the covering observations and the theoretical best that can be achieved through combination of those observations – if the latter is lower the ratio will be positive, meaning it is possible to reduce the image's sensitivity and thus reduce its noise. Greener pixels, as shown in Figure 4.6, accentuate regions with higher degrees of improvement.

While this archetypical workflow provides a clear pathway towards data exploitation of the ALMA archive, it nonetheless has a couple of limitations that will be addressed soon:

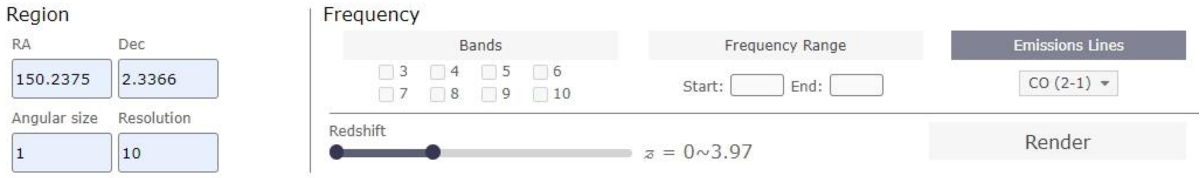
1. As mentioned in Figure 4.3, the sky map tool doesn't provide any insight on the spatial distribution of each cluster's frequency coverage. This can be especially useful if the user wants to study observations covering a particular emission line or a band: a functionality that was suggested by an attendee of the ENAA XXI presentation (5.2.2).

2. While it is possible to obtain an observation's location through a tooltip in the sky map as shown in Figure 4.2 and use that information to generate a regional plot, it would be desirable for the user to have the option for the platform to generate the right parameters directly and initiate the rendering process on its own.

4.2 Emission line queries

Due to the expansion of the Universe, incoming radiation from distant radio sources – "distant" here referring to the cosmological scales usually employed in extragalactic studies – is measured at lower frequencies than they were emitted at, a phenomenon known as *redshift.* To study these emissions, it is necessary to account for their associated redshift; on the field plot tool, it is possible to query a field by selecting an emission line and a redshift range, as mentioned in 3.2.2.3.

For this example, the user will be querying the COSMOS field that was identified in the previous use case. The input parameters are shown in Figure 4.7:



Region

RA: 150.2375

Dec: 2.3366

Angular size: 1

Resolution: 10

Frequency

Bands: □ 3 □ 4 □ 5 □ 6 □ 7 □ 8 □ 9 □ 10

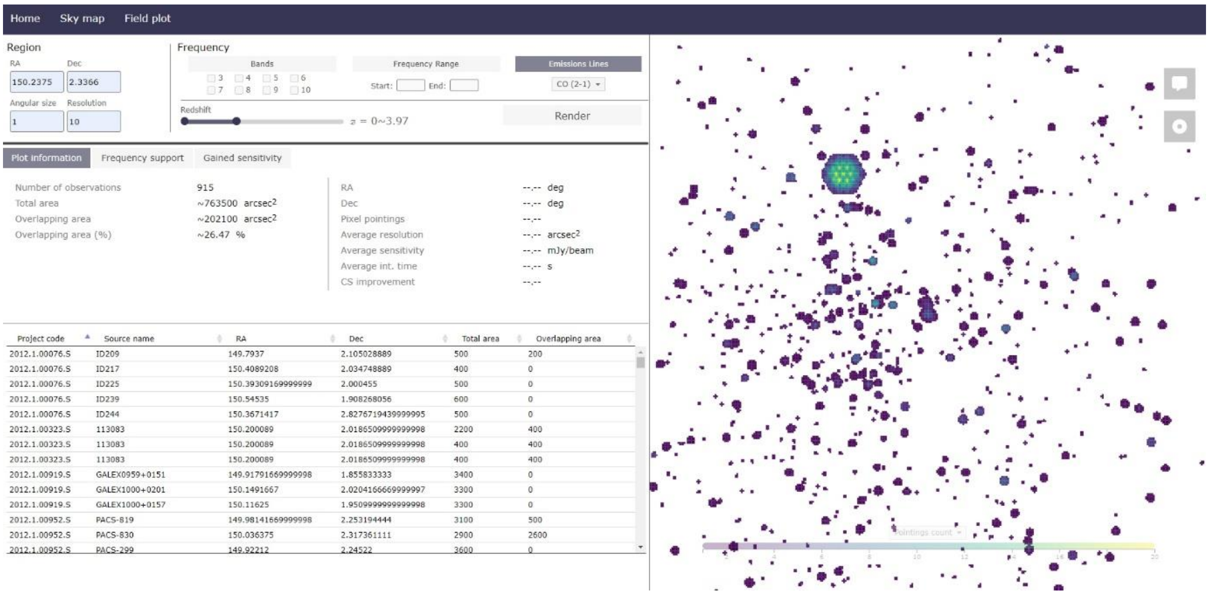
Frequency Range: Start: End:

Emissions Lines: CO (2-1)

Redshift: z = 0~3.97

Figure 4.7:The user input panel for an emission line query. The drop-down menu on the right side allows the user to select one of the many emission lines observed in extragalactic studies. As shown by the slider values, this query will select all observations with (at least) partial coverage of the CO (2-1) line at a redshift (z) range between 0 and ~4.

The plot is then rendered as shown in Figure 4.8. Just like in the previous use case, the multi-band coverage of this query can be assessed by the angular size of each observation, with smaller footprints corresponding to band 6, inside which the selected line is detected at rest frame, and larger ones corresponding to band 3, where the line can be detected at high redshift values. The tool guarantees that every observation captured by the query will have some coverage of the chosen emission line somewhere inside the redshift range.



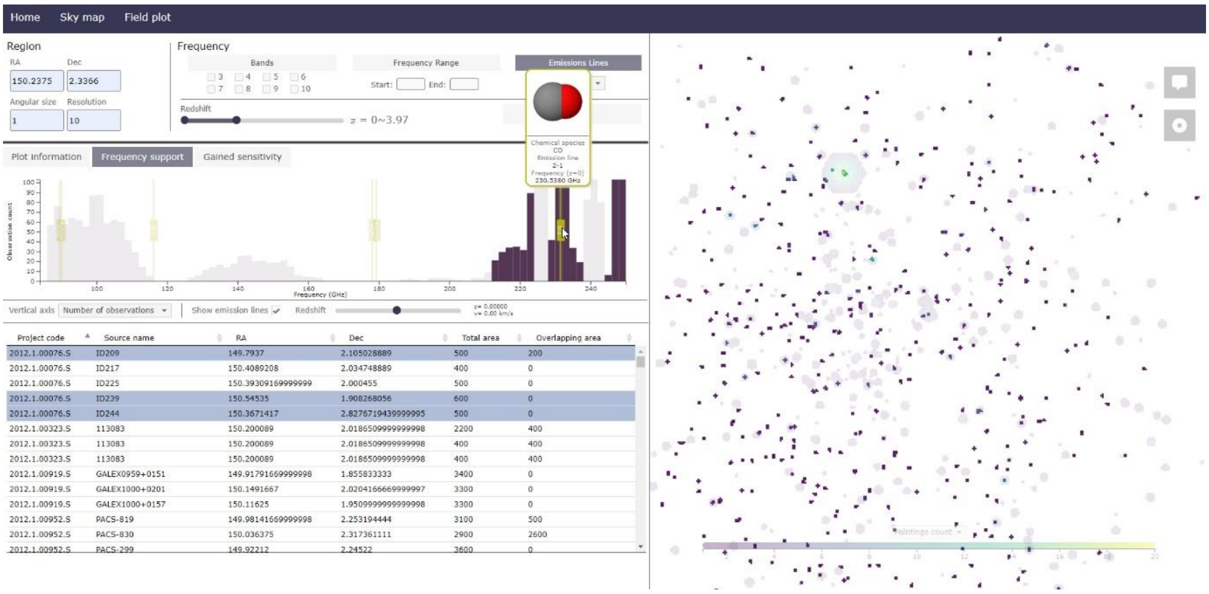
RA 150.2375 Dec 2.3366
Angular size 1 Resolution 10
CO (2-1)
z = 0~3.97

Number of observations	915	RA	--.-- deg
Total area	~763500 arcsec^2	Dec	--.-- deg
Overlapping area	~202100 arcsec^2	Pixel pointings	--.--
Overlapping area (%)	~26.47 %	Average resolution	--.-- arcsec^2
		Average sensitivity	--.-- mJy/beam
		Average int. time	--.-- s
		CS improvement	--.--

Project code	Source name	RA	Dec	Total area	Overlapping area
2012.1.00076.S	ID209	149.7937	2.105028889	500	200
2012.1.00076.S	ID217	150.4089208	2.034748889	400	0
2012.1.00076.S	ID225	150.39309169999999	2.000455	500	0
2012.1.00076.S	ID239	150.54535	1.908268056	600	0
2012.1.00076.S	ID244	150.3671417	2.8276719439999995	500	0
2012.1.00323.S	113083	150.200089	2.0186509999999998	2200	400
2012.1.00323.S	113083	150.200089	2.0186509999999998	400	400
2012.1.00323.S	113083	150.200089	2.0186509999999998	400	400
2012.1.00919.S	GALEX0959+0151	149.91791669999998	1.855833333	3400	0
2012.1.00919.S	GALEX1000+0201	150.1491667	2.0204166669999997	3300	0
2012.1.00919.S	GALEX1000+0157	150.11625	1.9509999999999998	3300	0
2012.1.00952.S	PACS-819	149.98141669999998	2.253194444	3100	500
2012.1.00952.S	PACS-830	150.036375	2.317361111	2900	2600
2012.1.00952.S	PACS-299	149.92212	2.24522	3600	0

Figure 4.8: The result of the redshifted emission line query. Every found observation is guaranteed to have coverage of the CO (2-1) line at rest frame (z=0) somewhere within the [0, 3.97] redshift range.

Switching to the frequency support panel will reveal the frequency coverage distribution of this observation set. The emission line can be observed on its rest frame frequency on the right side, and the info card pop-up confirms that it corresponds to the 2-1 transition. The pane's redshift slider can be currently used to add or remove an extra (and much smaller) redshift factor, usually correlated with the Doppler redshift observed in rotating galactic disks or other objects with a relatively low radial velocity. Furthermore, the user can click on any of the frequency bars to select and highlight the observations that cover them, as shown in Figure 4.9:



RA 150.2375 Dec 2.3366
Angular size 1 Resolution 10
z = 0~3.97
Vertical axis: Number of observations
Show emission lines ✓
Redshift: z= 0.00000, v= 0.00 km/s

Project code	Source name	RA	Dec	Total area	Overlapping area
2012.1.00076.S	ID209	149.7937	2.105028889	500	200
2012.1.00076.S	ID217	150.4089208	2.034748889	400	0
2012.1.00076.S	ID225	150.39309169999999	2.000455	500	0
2012.1.00076.S	ID239	150.54535	1.908268056	600	0
2012.1.00076.S	ID244	150.3671417	2.8276719439999995	500	0
2012.1.00323.S	113083	150.200089	2.0186509999999998	2200	400
2012.1.00323.S	113083	150.200089	2.0186509999999998	400	400
2012.1.00323.S	113083	150.200089	2.0186509999999998	400	400
2012.1.00919.S	GALEX0959+0151	149.91791669999998	1.855833333	3400	0
2012.1.00919.S	GALEX1000+0201	150.1491667	2.0204166669999997	3300	0
2012.1.00919.S	GALEX1000+0157	150.11625	1.9509999999999998	3300	0
2012.1.00952.S	PACS-819	149.98141669999998	2.253194444	3100	500
2012.1.00952.S	PACS-830	150.036375	2.317361111	2900	2600
2012.1.00952.S	PACS-299	149.92212	2.24522	3600	0

Figure 4.9: The result of selecting the frequency bar that contains the CO (2-1) line at z=0. As expected, this line's rest frame frequency falls within band 6, with its covering observations displaying a field-of-view in the order of 10 arcseconds. It should be noted that, at z = 4, the frequency of the CO (2-1) line will be redshifted to a frequency of approximately 46 GHz, which falls outside ALMA's detection capabilities. A z range of [0-2] allows the user to assess the coverage of this line on local galaxies (z=[0,0.5], falling inside band 6) and on emissions dating back to half of the Universe's age (z=[1-2], falling inside band 3).

5 Conclusion and future work

This section contrasts the status of the project against ideas for future development of the ASH. At present, the web application is deemed to fulfil the original proposal in multiple fronts; however, there are areas of improvement that have been outlined as the focus of near and mid-term development efforts.

5.1 Status

As of the writing of these thesis, all applications, features and capabilities of the ASH have been implemented in a development environment. The website's server is run on a virtual environment that contains all necessary Python libraries and that is provided for both Windows and Linux systems.

In the near future, the webpage is expected to be deployed on a machine belonging to the Institute of Astrophysics that runs on a Fedora, a Linux distribution. Getting the project ready for production, however, entails a few steps that are listed below (this list is not extensive and can be modified in the future):

1. **Changing to a PostgreSQL database**: while the current SQLite database management system is more than suitable for development purposes, it contains several built-in limitations that limit much of the tool's potential. One notable example is the hard-coded upper bound for the number of variables that can be contained in an SQL query, which limits the number of observations that can be obtained by the field plot tool - a restriction that doesn't exist in PostgreSQL. However, the final decision on which framework to switch to will be informed by further investigation in the short term.

2. **Testing on monitors with different resolutions:** by coincidence, the project was exclusively developed on computers with monitors with a resolution of 1920x1080 pixels; while the usage of Flexbox containers on most pages makes them responsive and adaptable to changes in screen size, aspect ratio and resolution, more extensive tests are planned as well.

3. **Deployment checklist:** any project that's built on the Django framework is subject to a few configuration changes before it can be safely deployed to a production state. These entail static file management, debug modes and performance improvements, among others (Deployment Checklist, Consulted 2021).

5.2 Presentations

To collect early feedback on the tool by the ALMA community, this project was the subject of multiple talks and presentations.

5.2.1 EAS 2020 – European Astronomical Society Annual Meeting

Formerly known as the European Week of Astronomy and Space Sciences (EWASS), this event features presentations on work being developed by European institutions linked to space sciences, from observation tools and instrumentation to scientific investigations on planetary sciences, star formation and cosmology, among others. Due to the COVID-19 pandemic, the meeting took place online for the first time ever. This project was submitted, approved, and presented on July the 3rd of 2020 as part of the *Surveys & Instruments* category.

The presentation mainly focused on the project's motivation and scientific potential, with its technical aspects being explained in a more elementary manner. Feedback from the project was positive; the Q&A session that followed, albeit short, allowed for one question on the field plot's tool processing time.

5.2.2 ENAA XXI – Encontro Nacional de Astronomia e Astrofísica 2021

The project was presented on the 7th of September 2021 as part of the event's *Astronomical Instruments* category. The 15-minute session focused on the project's purpose, namely as it concerns the scientific gains obtained through data combination, the existing toolset and plans for future development of the tool. All posed questions and answers are listed below:

- *"Instead of looking for all observations available for a region of the sky, will it be possible to also look for observations in different regions that have similar observations e.g., similar spectral setups, bands, etc?"* (Elisabete da Cunha) – Yes, this functionality is expected to be included in the sky map tool to assist users in finding regions of their particular interest.

- *"Are you in contact with ALMA/ESO? Are they helping in the development of this tool?"* (João Calhau) – Yes, we've been in contact with the European ARC network, and they've provided much of the necessary data (e.g., individual trace locations for mosaics). We are expected to increase this engagement in the following months as the project is deployed and we start requiring more data to be made available that's not currently on the ASA.

5.3 Current issues

This section lists a couple of outstanding issues with the webpage, as well as their possible solutions.

5.3.1 Field plot geometry

In its current implementation, the heatmap for the field plot application uses a simple rectangular projection to render the traces of the field's observations, pixel-by-pixel. This technique was initially chosen for its simplicity, as the centers of each pixel are directly mapped to grid positions in accordance with the plot's angular size and resolution. While this approach yields good results for regions on lower latitudes (as is the case with the COSMOS field, which was used to test the visualization tools throughout development), queries taken on regions with higher inclinations will display increasing visual degeneration – as shown in Figure 5.1. While the sensitivity estimations remain correct, as they are based on accurate angular distance measuring, both the visual fidelity of the plot and the area calculations are significantly affected.

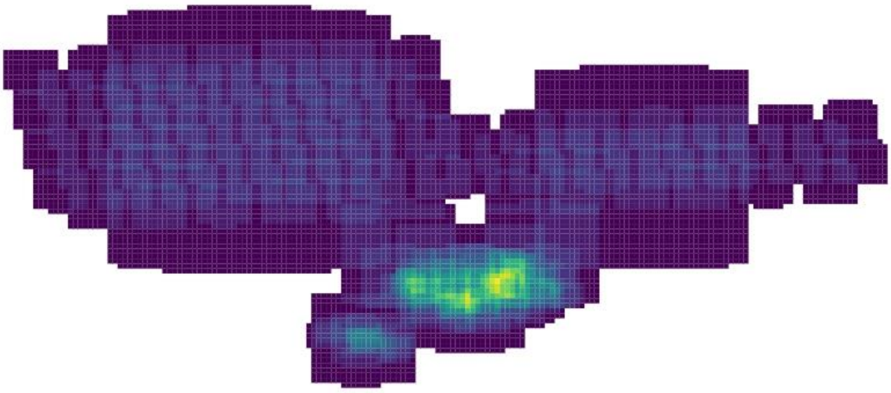


Figure 5.1: A series of observation mosaics with declinations around -70 degrees. Individual pointings are visible and display heavy area and shape distortion.

This distortion happens because the angular distance between two points at the same latitude (declination) but kept at a constant longitudinal separation (right ascension) decreases near the poles, which means the bottom side of the queried region will be "stretched" horizontally:

Therefore, this problem will be fixed in two steps:

1. Change the observation query to calculate proper maximum and minimum right ascension values based on the selected declination, so that the resulting set includes all observations that would end up inside the plotted squared region,

2. Create a new mapping function that calculates the horizontal offset of each pixel considering its declination – in other words, the position of pixels in relation to one another will reflect their actual angular separation, rather than the difference between their right ascension angles.

5.3.2 Data exploration pipeline

The two main data visualization applications (field plot and sky map) were designed to complement each other, allowing for the user to visualize the archive on different scale and detail levels. However, at present, these two tools are disjoined from one another; for instance, if a user wants to generate a plot for a specific cluster of the sky map, it'll have to obtain the coordinates for the center of the cluster and attempt to guess a sufficiently large radius that can encompass all the cluster's observations. Additionally, the project doesn't provide a direct way to view an observation's metadata, requiring users to search for the observation on the ALMA Science Archive.

These two issues are expected to drive a future revision of the sky map application which would include a new visualization window for both clusters and single observations that, apart from including the entirety of their metadata, would also provide the user with the appropriate field plot parameters.

6 References

ALADIN Sky Atlas. (Consulted 2021). Retrieved from Centre de Données astronomiques de Strasbourg: https://aladin.u-strasbg.fr/aladin.gml

ALMA Basics. (n.d.). Retrieved from ALMA Science Portal: https://almascience.nrao.edu/about-alma/alma-basics

ALMA Observatory. (n.d.). Retrieved from https://www.almaobservatory.org/en/home/

ARTEMIX. (Consulted 2021). Retrieved from ALMA RemoTE MIning eXperiment: http://artemix.obspm.fr/

ASA. (Consulted 2021). Retrieved from ALMA Science Archive: https://almascience.eso.org/aq/

Brunfeldt, K. (2014, October 13). *ProgressBar.js*. Retrieved from progressbar.js: https://github.com/kimmobrunfeldt/progressbar.js

Burkutean, S., Giannetti, A., Liuzzo, E., Massardi, M., Rygl, K., Brand, J., . . . Smareglia, R. (2018). KAFE: the Key-analysis Automated FITS-images Explorer. *Journal of Astronomical Telescopes, Instruments, and Systems*.

COSMOS. (2015). Retrieved from Caltech: https://cosmos.astro.caltech.edu/

Cox, P. C. (2017, March). *ALMA Users' Policies*. Retrieved from https://almascience.nrao.edu/documents-and-tools/cycle5/alma-user-policies

Data-Driven Documents. (2011, February 18). Retrieved from d3.js: https://d3js.org/

DataTables. (2014, May 1). Retrieved from DataTables: https://datatables.net/

Deployment Checklist. (Consulted 2021). Retrieved from Django Project: https://docs.djangoproject.com/en/3.2/howto/deployment/checklist/

Django. (2005, July 21). Retrieved from Django Project: https://www.djangoproject.com/

Droettboom, M. (2015). *Spherical Geometry Toolkit*. Retrieved from astropy: https://spacetelescope.github.io/spherical_geometry/spherical_geometry/

European Southern Observatory. (n.d.). Retrieved from https://www.eso.org/public/

Event Horizon Telescope. (n.d.). Retrieved from https://eventhorizontelescope.org/

Fonticons, I. (Consulted 2021). *FontAwesome*. Retrieved from FontAwesome: https://fontawesome.com/

jQuery. (2006, August 26). Retrieved from jQuery: https://jquery.com/

jQuery UI. (2007, September). Retrieved from jQuery UI: https://jqueryui.com/

MolView. (n.d.). Retrieved from https://molview.org/

Plotting 50 Million Points with D3. (2018, September 2). Retrieved from int21.io: https://int21.io/post/50-million-points/index.html

Redis. (2009, May 10). Retrieved from Redis: https://redis.io/

Solem, A. (2009). *Celery*. Retrieved from Celery: https://docs.celeryproject.org/en/stable/

The Event Horizon Telescope Collaboration. (2019). First M87 Event Horizon Telescope Results. I. The Shadow of the Supermassive Black Hole. *The Astrophysical Journal Letters*, 2-3.

Zue, C. (2018, January 20). *Celery Progress*. Retrieved from celery_progress: https://github.com/czue/celery-progress