SEARCHING FOR TRANSIENTS IN VLASS DATA

Project Report

Submitted in the partial fulfillment
of the Internship

by

Gaurav Gawade

Under the guidance of

Dr. Resmi Lekshmi
Department of ESS, IIST, Thiruvananthapuram

Department of Earth and Space Sciences
Indian Institute of Space Science and Technology, Thiruvananthapuram
2024

Abstract

This report documents the search for transients in the Very Large Array Sky Survey (VLASS) data, specifically focusing on Epoch 1 and Epoch 2. The study aims to identify and characterize transient astronomical events, contributing to our understanding of dynamic astrophysical phenomena.

Data was collected from the VLASS Quick Look Catalog, the Transient Name Server (TNS), and the NASA/IPAC Extragalactic Database (NED). Preprocessing involved filtering spurious sources, cross-referencing with existing catalogs, and normalizing flux measurements. The methodology included preliminary filtering based on statistical parameters, conversion of coordinates, and creation of Aitoff projection plots. Coordinate matching was performed using Python and TOPCAT to align Epoch 1 and Epoch 2 data, with a tolerance of 1 arcsecond.

Results include Aitoff projections that visualize the total flux distribution across the sky, histograms showing the impact of quality filtering on peak flux values, and scatter plots illustrating the relationship between peak flux and total flux ratio. These visualizations reveal significant insights into the spatial and temporal variations in the radio sky, highlighting regions with notable radio emissions and identifying reliable detections.

The study also involved matching the filtered epochs with supernova (SNe) and transient catalogs, followed by cross-referencing with the VLA's FIRST survey and NED. This process led to the creation of a catalog of 153 supernovae not previously observed in the radio regime. The findings underscore the importance of rigorous data filtering and matching techniques in large-scale radio surveys and contribute to the field of transient astronomy by providing a foundation for future research and investigation of transient phenomena in the radio sky.

Keywords: VLASS, transients, Aitoff projection, peak flux, coordinate matching, supernovae, radio astronomy.

Contents

1 Introduction

The study of transient astronomical phenomena is crucial for understanding the dynamic universe. Transients are objects or events that appear and disappear over short timescales, often involving high-energy processes that provide valuable insights into astrophysical mechanisms. The Very Large Array Sky Survey (VLASS) is a significant project aimed at mapping the radio sky at 3 GHz (S band) with unprecedented sensitivity and resolution. Conducted by the Karl G. Jansky Very Large Array (VLA), VLASS covers approximately 80% of the celestial sphere and is divided into multiple epochs to capture temporal variations in the radio sky.

This report focuses on the analysis of VLASS data from Epoch 1 and Epoch 2, aiming to identify and characterize transient sources. The primary objectives include implementing algorithms for transient detection, analyzing identified transients to determine their nature and properties, and contributing to the development of methodologies for efficient transient detection in large-scale radio surveys.

Data collection involved sourcing VLASS Quick Look Catalog (QL) data from CIRADA, utilizing supernova (SNe) catalogues from the Transient Name Server (TNS) and matching them with VLASS data, and cross-referencing with the NASA/IPAC Extragalactic Database (NED) for additional validation. Preprocessing steps included filtering spurious sources, normalizing flux measurements, and performing coordinate matching to align data from different epochs with high precision.

The methodology employed in this study involved a series of well-defined steps: preliminary filtering based on statistical parameters, creation of Aitoff projection plots to visualize total flux distribution, and detailed coordinate matching using both Python and TOPCAT software. The analysis included generating histograms to understand the impact of quality filters on peak flux values, and scatter plots to examine the relationship between peak flux and total flux ratio.

The results provide significant insights into the spatial and temporal variations in the radio sky observed by VLASS. By matching VLASS data with existing SNe and TDE catalogues, the study successfully identified previously unobserved supernovae in the radio regime. These findings contribute to the field of transient astronomy by providing a comprehensive catalogue of transient sources, enhancing our understanding of their behavior and underlying physical mechanisms.

This report is organized as follows: Section 2 provides a theoretical background relevant to transient astronomy and VLASS data analysis. Section 3 describes the data collection process and preprocessing steps. Section 4 outlines the methodologies employed for transient detection and analysis. Section 5 discusses the implementation of these methodologies and tools used. Section 6 presents the findings and results of the analysis. Section 7 offers a discussion of the results, including challenges encountered. Finally, Section 8 concludes the report and suggests avenues for future research.

2 Theoretical Background

Transients are astronomical phenomena that vary in brightness or appear and disappear over short timescales compared to the lifetime of their host systems. These events are significant as they often involve high-energy processes and provide insight into the dynamic universe [16, 9].

2.1 Types of Transients

Transients can be broadly categorized based on their timescales, energy outputs, and astrophysical origins. Key types include:

2.1.1 Supernovae

Supernovae are powerful stellar explosions marking the end of a star's life cycle. They are classified into different types based on their spectral properties and progenitor systems:

- **Type Ia Supernovae:** Result from the thermonuclear explosion of a white dwarf in a binary system. These are standard candles for measuring cosmic distances.
- **Core-collapse Supernovae:** Occur when a massive star exhausts its nuclear fuel and its core collapses, resulting in a catastrophic explosion. Subtypes include Type II, Ib, and Ic, distinguished by the presence or absence of hydrogen and helium lines in their spectra.

Supernovae are key for understanding stellar evolution, nucleosynthesis, and the chemical enrichment of galaxies [16, 9].

2.1.2 Gamma-Ray Bursts (GRBs)

Gamma-ray bursts are the most luminous electromagnetic events in the universe, characterized by brief but intense emissions of gamma rays. GRBs are divided into two categories based on their duration:

- **Short GRBs:** Lasting less than 2 seconds, typically associated with the mergers of compact objects like neutron stars.
- **Long GRBs:** Lasting more than 2 seconds, believed to originate from the collapse of massive stars [23].

GRBs provide insights into extreme physics and the conditions of the early universe [14].

2.1.3 X-ray Transients

X-ray transients are sources that exhibit significant variability in their X-ray emission. They often involve compact objects like neutron stars or black holes in binary systems accreting matter from a companion star. Types include:

- **Soft X-ray Transients:** Typically associated with accreting black holes or neutron stars.
- **Hard X-ray Transients:** Often linked to magnetars or pulsar wind nebulae [15].

2.1.4 Fast Radio Bursts (FRBs)

Fast radio bursts are enigmatic, millisecond-duration bursts of radio waves. While their exact origins are still unknown, they are hypothesized to originate from highly magnetized neutron stars, supernova remnants, or even extragalactic sources [13].

2.2 Detection of Transients

Detecting transients involves continuous monitoring of the sky across various wavelengths. Methods and technologies employed include:

2.2.1 Optical Surveys

Optical surveys utilize wide-field cameras on telescopes to detect changes in the brightness of celestial objects. Notable projects include:

- **Zwicky Transient Facility (ZTF):** Aims to detect supernovae, variable stars, and other optical transients.
- **Pan-STARRS:** Conducts wide-field imaging to discover and characterize transient events [6].

2.2.2 Radio Surveys

Radio surveys are crucial for detecting transients like FRBs, pulsars, and radio supernovae. Techniques involve monitoring large portions of the sky with high sensitivity and resolution. Prominent surveys include:

- **The Square Kilometre Array (SKA):** Expected to revolutionize radio transient detection with its unprecedented sensitivity and resolution [2].
- **The Australian Square Kilometre Array Pathfinder (ASKAP):** Conducts surveys like the Commensal Real-time ASKAP Fast Transients (CRAFT) survey [5].

2.2.3 X-ray and Gamma-ray Observations

Space-based telescopes such as the Chandra X-ray Observatory and the Fermi Gamma-ray Space Telescope are instrumental in detecting high-energy transients. These observatories provide data on X-ray binaries, GRBs, and other energetic phenomena [3].

2.2.4 Gravitational Wave Observatories

Gravitational wave detectors like LIGO and Virgo have opened a new window for transient detection, capturing mergers of black holes and neutron stars. These events are often accompanied by electromagnetic counterparts, providing multi-messenger observations [1].

2.3 Transients in the Radio Regime

Radio transients offer a unique perspective on high-energy astrophysical processes. Key considerations include:

2.3.1 Propagation Effects

Radio waves are affected by the interstellar medium (ISM), leading to dispersion and scattering. Dispersion measures provide information about the electron density in the ISM, while scattering effects can reveal properties of the medium [10].

2.3.2 Emission Mechanisms

Radio emissions can arise from various mechanisms, including:

- **Synchrotron Radiation:** Produced by relativistic electrons spiraling in magnetic fields, common in supernova remnants and active galactic nuclei [12].
- **Pulsar Emissions:** Periodic radio emissions from rotating neutron stars with strong magnetic fields.
- **Magnetar Activity:** Intense bursts of radio waves from highly magnetized neutron stars [19].

2.3.3 Monitoring Challenges

Monitoring radio transients requires high sensitivity and resolution, as well as the ability to cover large areas of the sky. Instruments like the VLA and ASKAP are designed to meet these challenges [4].

2.4 The Very Large Array (VLA) and the Very Large Array Sky Survey (VLASS)

The Karl G. Jansky Very Large Array (VLA) is a state-of-the-art radio observatory that has significantly contributed to our understanding of the radio universe.

2.4.1 Overview of the VLA

Located in New Mexico, the VLA consists of 27 antennas arranged in a Y-shaped configuration. It operates across a wide range of frequencies (1-50 GHz) and provides high-resolution imaging capabilities. The VLA is instrumental in studying radio galaxies, quasars, supernova remnants, and other radio sources [18].

2.4.2 The Very Large Array Sky Survey (VLASS)

VLASS is a large-scale survey conducted with the VLA at 3 GHz (S band) aimed at mapping the entire sky visible from the VLA (about 80% of the celestial sphere). Key objectives of VLASS include:

- Creating a comprehensive catalog of radio sources.
- Studying radio transients and variable sources.
- Providing data for multi-wavelength studies [7].

2.4.3 Epochs of VLASS

VLASS is divided into three epochs to maximize the detection of transients and variability. The specific details for each epoch are as follows:

- **Epoch 1:** Observations for the first epoch began in September 2017 and were completed in August 2019. This epoch was further divided into sub-epochs:
  - **Epoch 1.1:**
    - **Start Date:** September 2017
    - **End Date:** October 2018
    - **Features:** Initial survey pass, providing baseline data, known for having some image quality issues due to rapid processing [8].
  - **Epoch 1.2:**
    - **Start Date:** November 2018
    - **End Date:** August 2019
    - **Features:** Improved image processing, resulting in better quality images and more reliable data [8].
- **Epoch 2:** Observations started in August 2020 and are expected to complete in September 2022. This epoch also includes sub-epochs similar to Epoch 1, with continual improvements and adjustments based on previous findings.
- **Epoch 3:** Scheduled to start in October 2024. This epoch aims to further refine the catalog and enhance the temporal resolution for transient studies.

3 Data Collection

The data for this project was collected from several sources that provide comprehensive datasets on transients and radio surveys. The primary sources used are detailed below:

3.1 VLASS Quick Look Catalog

The Very Large Array Sky Survey (VLASS) Quick Look Catalog was one of the main sources of data. This catalog provides immediate access to the radio images and preliminary source catalogs from the VLASS observations. The data includes radio sources detected at 3 GHz, with information on their positions, flux densities, and other relevant parameters. The catalog can be accessed through the Canadian Initiative for Radio Astronomy Data Analysis (CIRADA) website at `https://cirada.ca/vlasscatalogueql0` [22].

3.2 Transient Name Server (TNS)

The Transient Name Server (TNS) is another significant source, providing an extensive table of transient astronomical events. This database includes information on various types of transients such as supernovae, gamma-ray bursts, and other transient phenomena. The data from TNS was particularly useful for cross-referencing and validating the transient detections from the VLASS data. The TNS catalog is available on GitHub at `https://github.com/astrocatalogs/transient-table` [20].

3.3 NASA/IPAC Extragalactic Database (NED)

The NASA/IPAC Extragalactic Database (NED) is a comprehensive repository of data on extragalactic objects. For this project, NED's cone search service was utilized to gather data on known extragalactic sources within specific regions of the sky. This service provides detailed information on the positions, redshifts, and other properties of extragalactic sources, aiding in the identification and classification of transients detected in the VLASS data. The NED cone search can be accessed at `https://ned.ipac.caltech.edu/conesearch` [11].

3.4 Data Preprocessing

The collected data underwent several preprocessing steps to ensure quality and consistency. These steps included:

- **Filtering:** Removing spurious sources and noise artifacts from the radio images.
- **Cross-Matching:** Cross-referencing sources from the VLASS catalog with those from TNS and NED to identify known transients and eliminate duplicates.
- **Normalization:** Standardizing the flux density measurements to a common scale for accurate comparison and analysis.

3.5 Data Integration

After preprocessing, the data from the different sources was integrated into a unified dataset. This integrated dataset forms the basis for the subsequent analysis and transient detection processes described in the following sections of this report.

4 Methodology

4.1 Preliminary Filtering

The methodology for preprocessing and analyzing the data from VLASS involves several steps, including data loading, filtering, and visualization. The following sections provide a detailed explanation of each step performed in the code.

4.1.1 Data Loading

The data for VLASS Epoch 1 and later Epoch 2 was loaded into a pandas DataFrame for further analysis. The dataset includes several columns, among which the key columns for this analysis are Right Ascension (RA), Declination (DEC), and Island Root Mean Square (Isl_rms).

4.1.2 Validation of Required Columns

To ensure the integrity of the dataset, a check was performed to verify the existence of the required columns (RA, DEC, and Isl_rms). If any of these columns were missing, an error was raised to prevent further processing of incomplete data.

4.1.3 Calculation of Statistical Parameters

The mean and standard deviation of the Isl_rms column were calculated to establish a baseline for filtering the data. These statistical parameters are essential for defining the range within which the data will be considered valid.

4.1.4 Data Filtering

As advised in Gordon et al., duplicate_flag $<$ 2; quality_flag == 0; $S\_Code \neq$ 'E' was applied. A 1-sigma range was defined using the calculated mean and standard deviation of Isl_rms. Data points within this range were selected for further analysis, effectively filtering out extreme values and noise. This process helps in retaining the most relevant data points for accurate analysis.

4.1.5 Conversion of Coordinates

The filtered data was converted from degrees to radians to facilitate the creation of an Aitoff projection plot. The RA values were shifted by 180 degrees to center the plot appropriately for the Aitoff projection.

4.1.6 Aitoff Projection Plot

An Aitoff projection plot was created to visualize the distribution of the filtered Isl_rms values across the sky. The plot was generated using a color scale based on the Isl_rms values, with RA and DEC coordinates on the x and y axes respectively. The plot was saved with high resolution for detailed examination.

4.1.7 Histogram of Peak Flux

A histogram of the peak flux (S_peak) was created to compare the distribution of all data points with those that passed the quality flag filtering (quality_flag == 0). The histogram was plotted using logarithmic scales for both axes to effectively visualize the range of flux values. This step involved defining logarithmic bins and plotting the histograms using a step plot.

4.1.8 Ratio of Peak Flux to Total Flux

The ratio of peak flux to total flux was calculated for each data point. This ratio provides insights into the source's emission characteristics. The data was visualized using a scatter plot with a gradient color representing the total flux. Horizontal and vertical lines were added to indicate significant thresholds ($y = 1$ and $x = 3$), and the plot was saved with a logarithmic scale for both axes.

4.1.9 Density Plot and Filtering of Significant Data Points

A 2D histogram (density plot) was created to visualize the relationship between the peak flux and the peak-to-total flux ratio. The plot was generated using logarithmic bins and a color gradient representing the density of data points. Data points above the defined thresholds ($y = 1$ and $x = 3$) were filtered and saved for further analysis. These filtered data points were highlighted in red on the density plot.

4.1.10 Saving and Visualization of Filtered Data Points

The filtered data points (highlighted in red on the density plot), which represent significant transient candidates, were saved to a CSV file for detailed examination. The final plot, combining the density background with the highlighted filtered data points, was saved with high resolution.

4.2 Coordinate Matching / Sky Matching

In this section, we describe the methodology used for coordinate matching between Epoch 1 and Epoch 2 data. The matching process was performed with a tolerance of 1 arcsecond, and the results were used to create new columns, including the separation between matched coordinates and the ratio of total flux from Epoch 1 to Epoch 2. The matching was performed using both Python code and TOPCAT software [17].

4.2.1 Python Algorithm for Coordinate Matching

The following steps outline the Python algorithm used for coordinate matching:

1. **Load the Data:** The data from Epoch 1 and Epoch 2 were loaded into pandas DataFrames. The relevant files were 'filtered_data_points_epoch1.csv' and 'filtered_data_points_epoch2.csv'.

2. **Ensure RA and DEC Columns are Numeric:** The RA and DEC columns were checked to ensure they were numeric. If not, an error was raised to prevent further processing.

3. **Create SkyCoord Objects:** Using the `astropy.coordinates.SkyCoord` class, SkyCoord objects were created for the RA and DEC columns from both Epoch 1 and Epoch 2 data, with the units explicitly set to degrees.

4. **Perform the Matching:** The `match_to_catalog_sky` method was used to perform the sky matching between the two sets of coordinates.

5. **Filter Matches Within 1 Arcsecond:** The matches were filtered to include only those within a separation of 1 arcsecond.

6. **Extract Matched Data:** The indices of the matched data points were extracted, and the corresponding data from both epochs were combined into a single DataFrame.

7. **Add Separation Column:** A new column was added to the combined DataFrame to record the separation (in arcseconds) between the matched coordinates.

8. **Save the Matched Data:** The combined DataFrame with matched data was saved to a CSV file named 'matched_data_25Jun_1arc.csv'.

4.2.2 Coordinate Matching Using TOPCAT

TOPCAT is a software tool for the interactive analysis of tabular data. The following steps outline how coordinate matching was performed using TOPCAT [17]:

1. **Load the Data:** Open TOPCAT and load the Epoch 1 and Epoch 2 data files.

2. **Select the Matching Criteria:** In the matching menu, choose RA and DEC columns for both datasets as the matching criteria.

3. **Set Matching Parameters:** Set the matching tolerance to 1 arcsecond.

4. **Perform the Match:** Execute the matching operation. TOPCAT will identify and pair the coordinates from Epoch 1 and Epoch 2 within the specified tolerance.

5. **Save the Matched Data:** Export the matched data to a new file. This file will include columns for the separation between matched coordinates and the flux ratios (totalflux_1 / totalflux_2).

By using both Python and TOPCAT, we ensured the accuracy and reliability of the coordinate matching process, leveraging the strengths of both tools.

4.3 SNe / Transient Catalogue Matching

In this section, we describe the methodology used for matching supernovae (SNe) and transient catalogues with the filtered data from Epoch 1 and Epoch 2. The process involved several steps, including sky matching with the SNe catalogue, and subsequent matching with the FIRST catalogue from the VLA's FIRST survey. The SNe catalogue was obtained from a GitHub repository and contained entries from January 2015 to July 2019.

4.3.1 Loading and Filtering the SNe Catalogue

The SNe catalogue was sourced from the following GitHub repository [21]. This catalogue includes various supernovae and transient events with their respective coordinates and observation dates. For this analysis, we filtered the catalogue to include only entries from the date range 2015/01 to 2019/07.

4.3.2 Sky Matching Filtered Epochs with SNe Catalogue

The following steps outline the process of sky matching the filtered data from Epoch 1 and Epoch 2 with the SNe catalogue:

1. **Load the Data:** The filtered data from Epoch 1 (‘filtered_data_points_epoch1.csv‘) and Epoch 2 (‘filtered_data_points_epoch2.csv‘), along with the SNe catalogue, were loaded into pandas DataFrames.

2. **Create SkyCoord Objects:** SkyCoord objects were created for the RA and DEC columns in each dataset, with units explicitly set to degrees.

3. **Perform the Matching:** Using the `match_to_catalog_sky` method from `astropy.coordinates`, sky matching was performed between the filtered data and the SNe catalogue. A tolerance of 1 arcsecond was used for this matching.

4. **Filter Matches:** The matches were filtered to retain only those within 1 arcsecond.

5. **Combine the Matched Data:** The matched data from the filtered epochs and the SNe catalogue were combined into a single DataFrame, with new columns added to record the separation between matched coordinates and other relevant details.

4.3.3 Matching with the FIRST Catalogue

After matching the filtered epochs with the SNe catalogue, the resulting matched data was further matched with the FIRST catalogue. The FIRST (Faint Images of the Radio Sky at Twenty-Centimeters) survey is a key survey conducted by the VLA, providing high-resolution radio images of the sky.

1. **Load the FIRST Catalogue:** The FIRST catalogue was loaded into a pandas DataFrame. This catalogue includes high-resolution radio sources with their respective coordinates.

2. **Create SkyCoord Objects:** SkyCoord objects were created for the RA and DEC columns in the matched data (from the previous step) and the FIRST catalogue, with units set to degrees.

3. **Perform the Matching:** The `match_to_catalog_sky` method was used to perform sky matching between the matched data and the FIRST catalogue, again using a tolerance of 1 arcsecond.

4. **Filter Matches:** The matches were filtered to retain only those within 1 arcsecond.

5. **Combine the Matched Data:** The final matched data, which includes entries from the filtered epochs, SNe catalogue, and the FIRST catalogue, was combined into a single DataFrame. Additional columns were added to record the separation and other relevant details.

4.3.4 Summary of the Matching Process

The process of matching the filtered epochs with the SNe and FIRST catalogues involved multiple steps to ensure accuracy and reliability. The final matched dataset provides a comprehensive view of supernovae and transient events in the context of the VLASS data, enhanced by high-resolution observations from the FIRST survey. This matched dataset is crucial for further analysis and investigation of transient phenomena in the radio sky.

5 Results and Discussion

5.1 Aitoff Projections of Epoch 1 and Epoch 2

The Aitoff projections for Epoch 1 and Epoch 2 provide a comprehensive visual representation of the total flux distribution across the sky. These projections are crucial for understanding the spatial distribution and intensity of radio sources detected in the Very Large Array Sky Survey (VLASS) during the respective epochs.

5.1.1 Aitoff Projection of Epoch 1

Figure 1 shows the Aitoff projection of the total flux for Epoch 1. The plot represents the distribution of total flux values, with the RA and DEC coordinates mapped onto an Aitoff projection. The color scale indicates the intensity of the total flux, with brighter regions corresponding to higher flux values. This visualization helps in identifying areas with significant radio emissions and assessing the overall flux distribution across the observed sky.

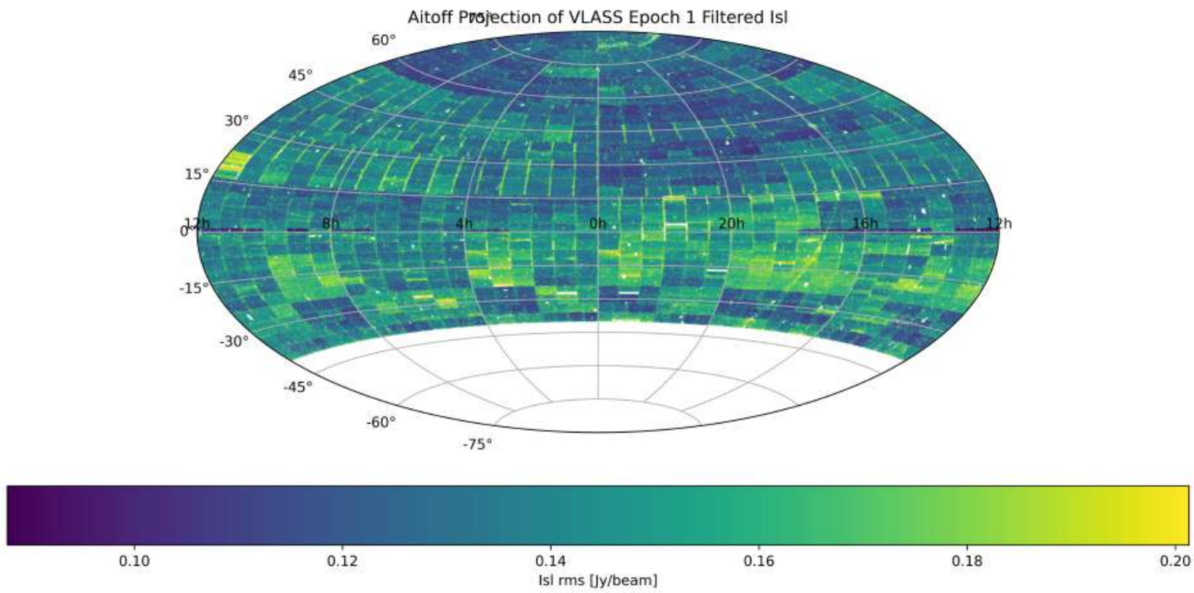

Figure 1: Aitoff projection of the total flux for Epoch 1. The color scale represents the total flux intensity, with brighter regions indicating higher flux values.

5.1.2 Aitoff Projection of Epoch 2

Figure 2 presents the Aitoff projection of the total flux for Epoch 2. Similar to Epoch 1, this plot displays the total flux distribution using an Aitoff projection of the RA and DEC coordinates. The color scale represents the intensity of the total flux, allowing for a direct comparison with the distribution observed in Epoch 1. This comparison is essential for identifying temporal variations and new transient sources that may have emerged between the two epochs.

The Aitoff projections for both epochs provide valuable insights into the spatial and temporal variations in the radio sky observed by VLASS. These visualizations serve as a foundation for further analysis and interpretation of the transient phenomena captured in the survey data.

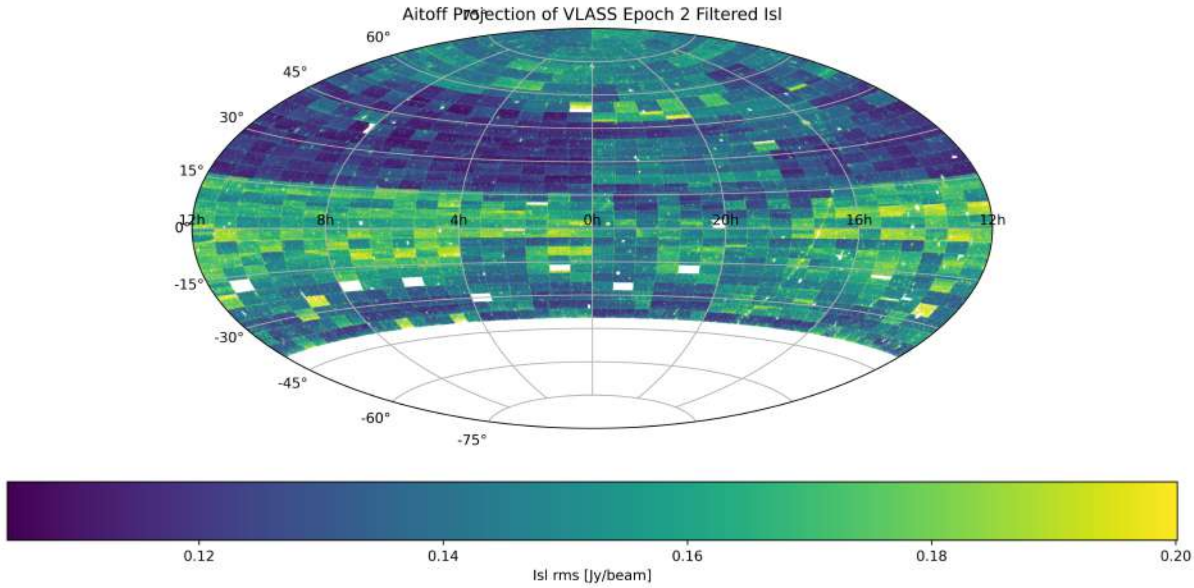

Figure 2: Aitoff projection of the total flux for Epoch 2. The color scale represents the total flux intensity, with brighter regions indicating higher flux values.

5.2 Number of Rows vs Peak Flux (mJy/beam)

The relationship between the number of rows (N) and the peak flux (S_peak) in mJy/beam is crucial for understanding the effect of data filtering on the dataset. This analysis compares the distribution of peak flux values for the entire dataset against the subset with quality_flag == 0, highlighting the impact of applying quality filters.

5.2.1 Histogram of Peak Flux

Figure 3 shows the histogram of the peak flux (S_peak) for the entire dataset and the subset with quality_flag == 0. The plot uses logarithmic scales for both axes to effectively visualize the range of flux values. The blue line represents the distribution for all data points, while the red dashed line represents the distribution for the data points with quality_flag == 0.

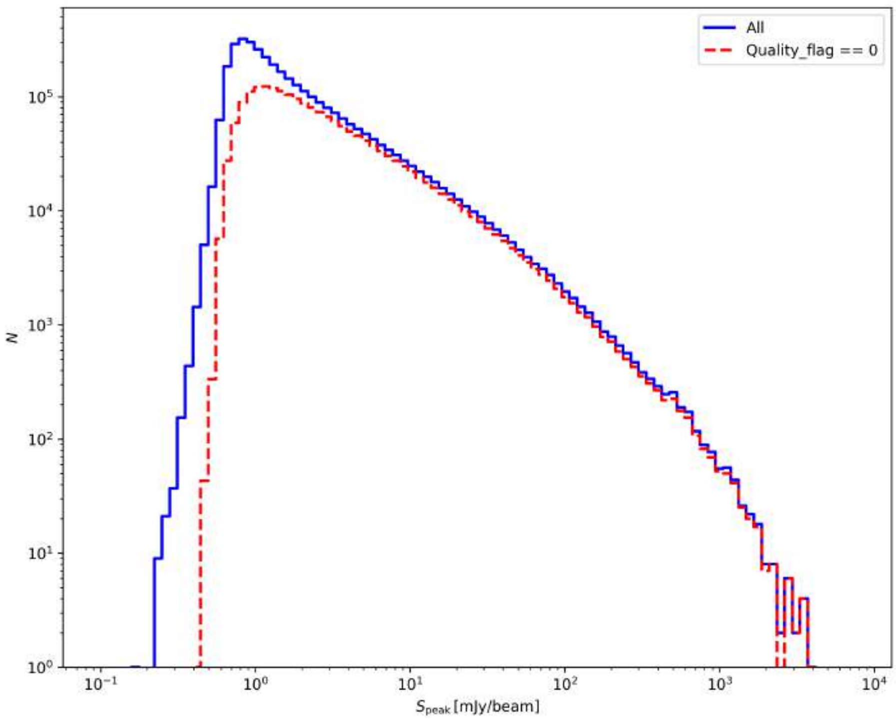

Figure 3: Histogram of peak flux (S_peak) in mJy/beam for Epoch 1. The blue line represents all data points, while the red dashed line represents the data points with quality_flag == 0. Logarithmic scales are used for both axes.

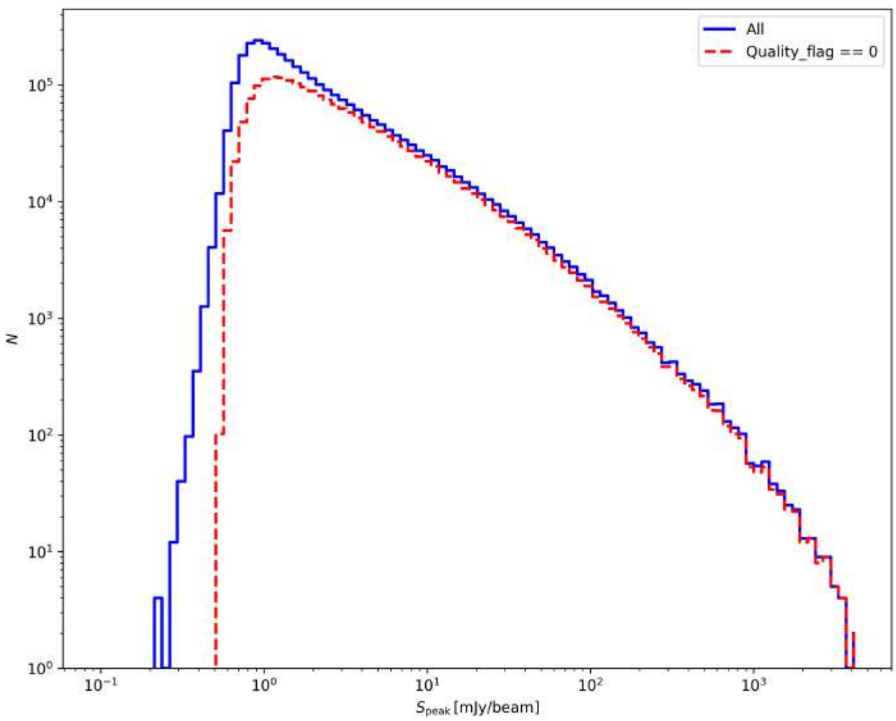

Figure 4: Histogram of peak flux (S_peak) in mJy/beam for Epoch 2. The blue line represents all data points, while the red dashed line represents the data points with quality_flag == 0. Logarithmic scales are used for both axes.

The plot demonstrates a noticeable deviation between the two lines, particularly at higher flux values. The quality_flag == 0 line shows a reduced number of entries compared to the unfiltered data, indicating that the quality filter effectively removes less reliable detections. This deviation is significant for understanding the effect of the quality filter on the dataset's overall flux distribution.

By examining this histogram, we can assess the extent to which the quality filtering impacts the dataset and ensure that the resulting data used for further analysis is of high reliability. This step is essential for minimizing false detections and ensuring the accuracy of subsequent transient studies.

5.3 Scatter Plots of Peak Flux vs. Total Flux Ratio

The scatter plots of the peak flux (S_peak) versus the ratio of peak flux to total flux (S_peak / S_total) provide insights into the characteristics and quality of the detected sources in the VLASS data for Epoch 1 and Epoch 2. The red dots in the scatter plots represent the final filtered data used for further analysis.

5.3.1 Epoch 1: Peak Flux vs. Total Flux Ratio

Figure 5 presents the scatter plot of the peak flux versus the ratio of peak flux to total flux for Epoch 1. The plot uses a logarithmic scale for both axes to visualize the wide range of flux values. The color scale represents the density of the data points. The red horizontal and vertical lines indicate the thresholds used for filtering the data.

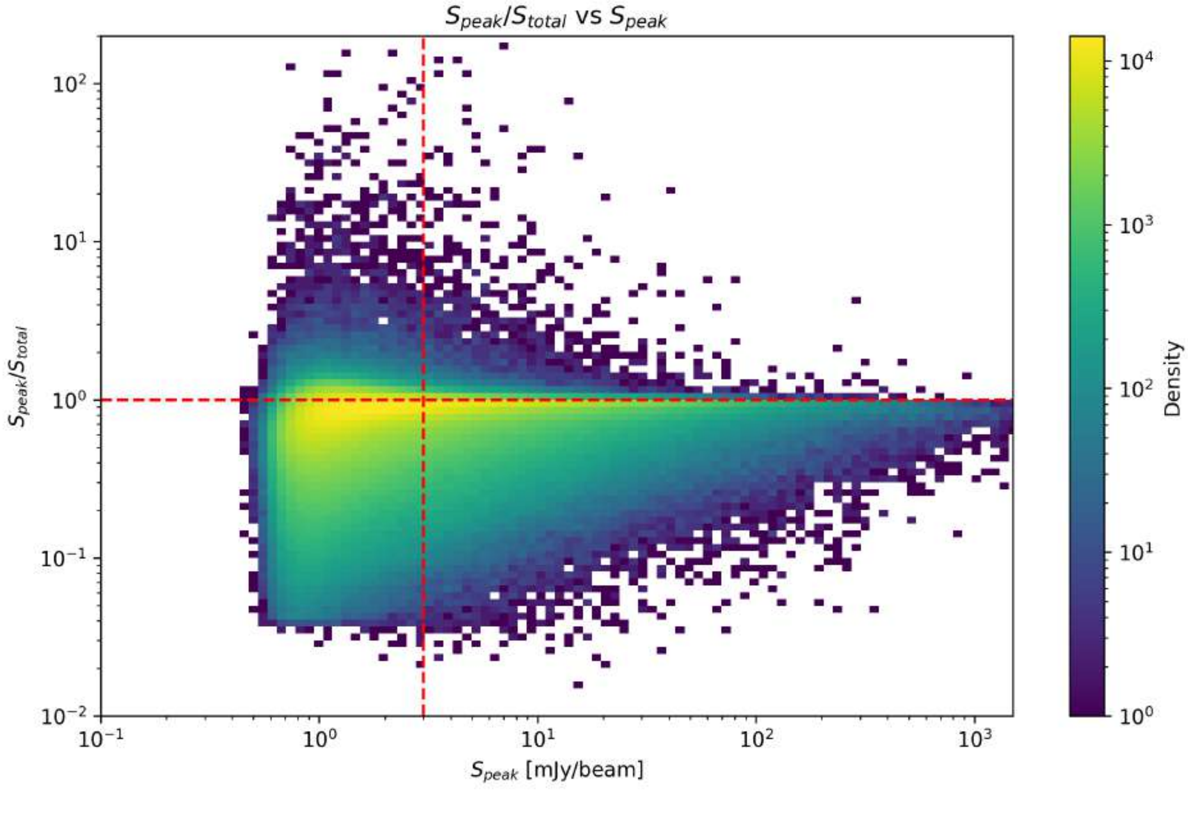

Figure 5: Scatter plot of S_{peak} vs S_{peak}/S_{total} for Epoch 1. The color scale represents the density of the data points. The red horizontal and vertical lines indicate the filtering thresholds.

Figure 6 shows the same scatter plot with the final filtered data points highlighted in red. These filtered data points, which lie above the thresholds, were used for further analysis.

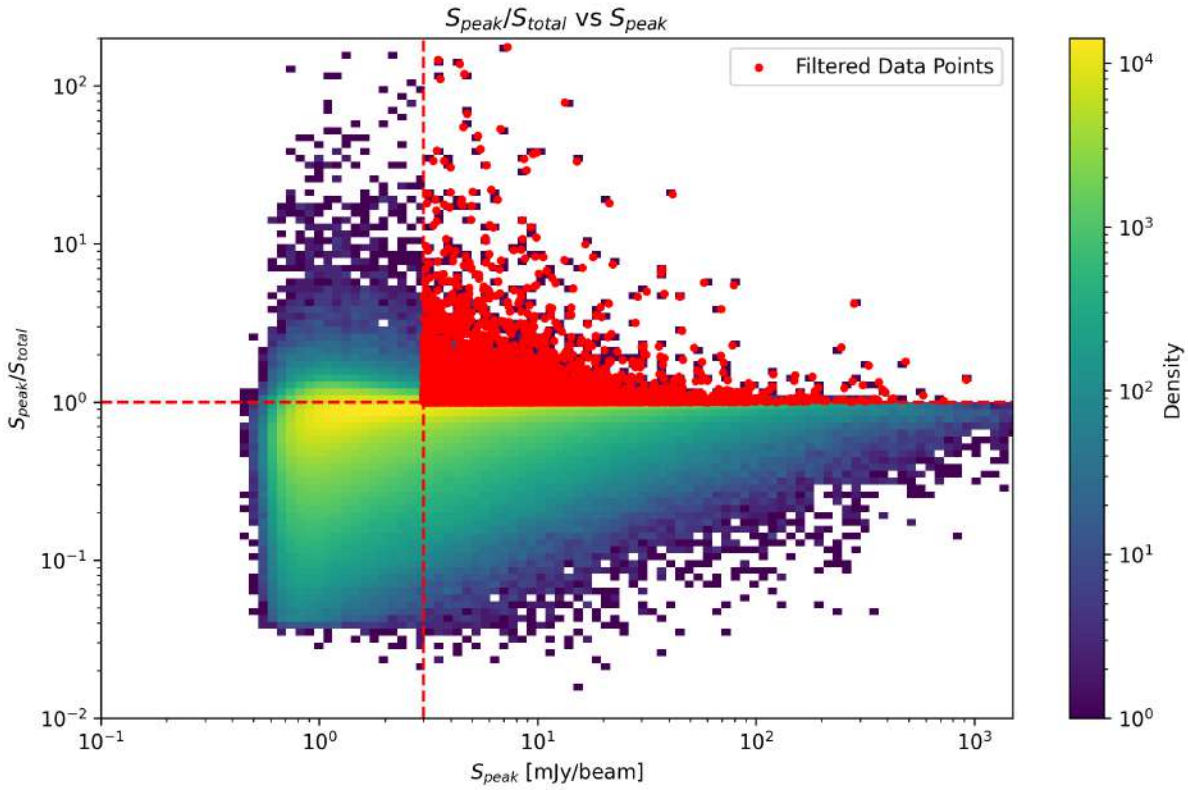

Figure 6: Scatter plot of S_{peak} vs S_{peak}/S_{total} for Epoch 1 with filtered data points highlighted in red. These points were used for further analysis.

5.3.2 Epoch 2: Peak Flux vs. Total Flux Ratio

Figure 7 presents the scatter plot of the peak flux versus the ratio of peak flux to total flux for Epoch 2. Similar to Epoch 1, the plot uses a logarithmic scale for both axes, with the color scale representing the density of the data points. The red horizontal and vertical lines indicate the filtering thresholds.

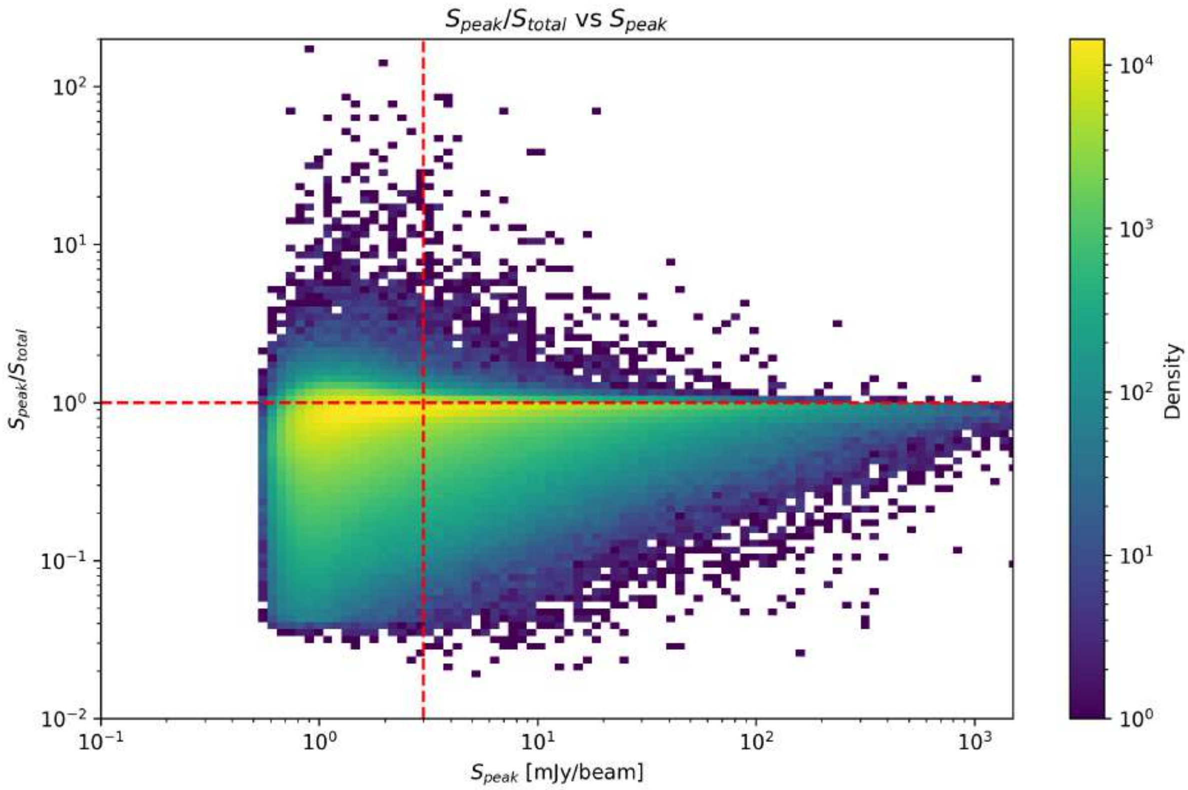

Figure 7: Scatter plot of S_{peak} vs S_{peak}/S_{total} for Epoch 2. The color scale represents the density of the data points. The red horizontal and vertical lines indicate the filtering thresholds.

Figure 8 shows the same scatter plot with the final filtered data points highlighted in red. These filtered data points, which lie above the thresholds, were used for further analysis.

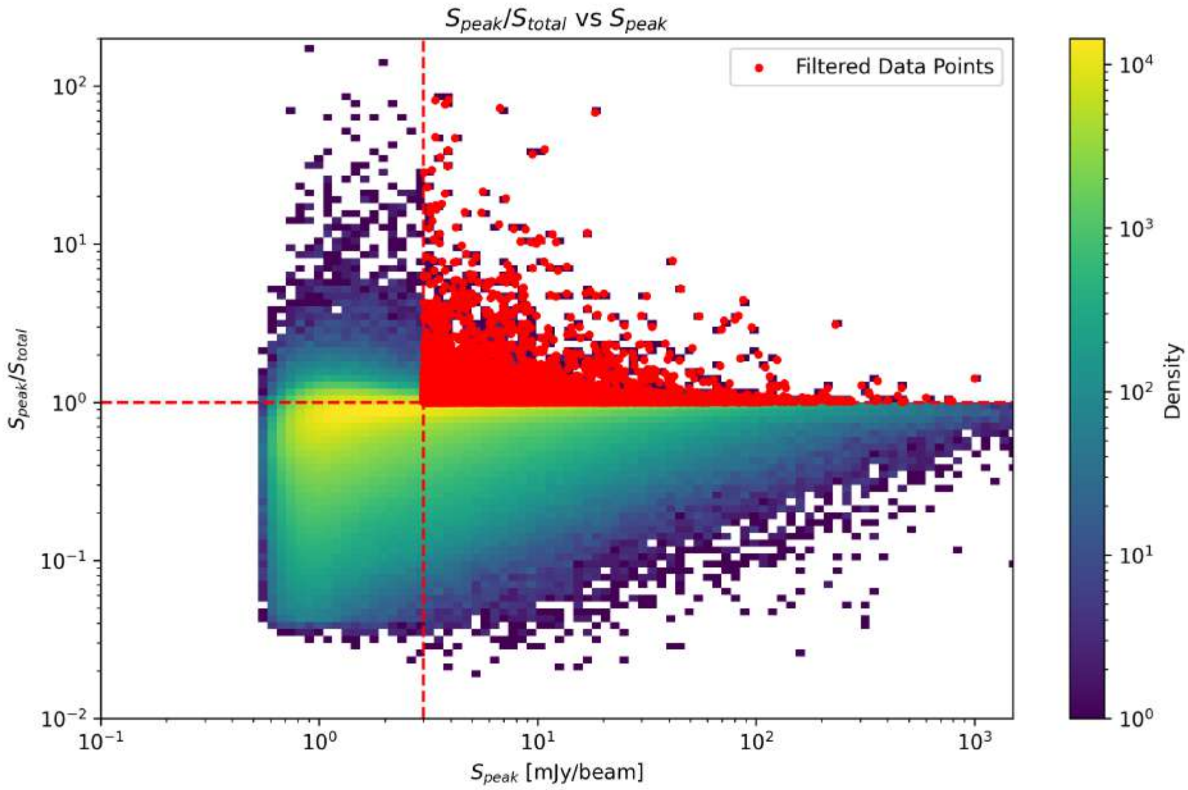

Figure 8: Scatter plot of S_{peak} vs S_{peak}/S_{total} for Epoch 2 with filtered data points highlighted in red. These points were used for further analysis.

The scatter plots for both epochs demonstrate the effectiveness of the filtering process. The final filtered data points represent the most reliable detections, minimizing the impact of noise and ensuring the accuracy of the subsequent analysis. These plots are crucial for understanding the quality and characteristics of the sources detected in the VLASS data.

For the filtered epoch 1, there are **37,413** data points; and for the filtered epoch 2, there are **36,853** data points.

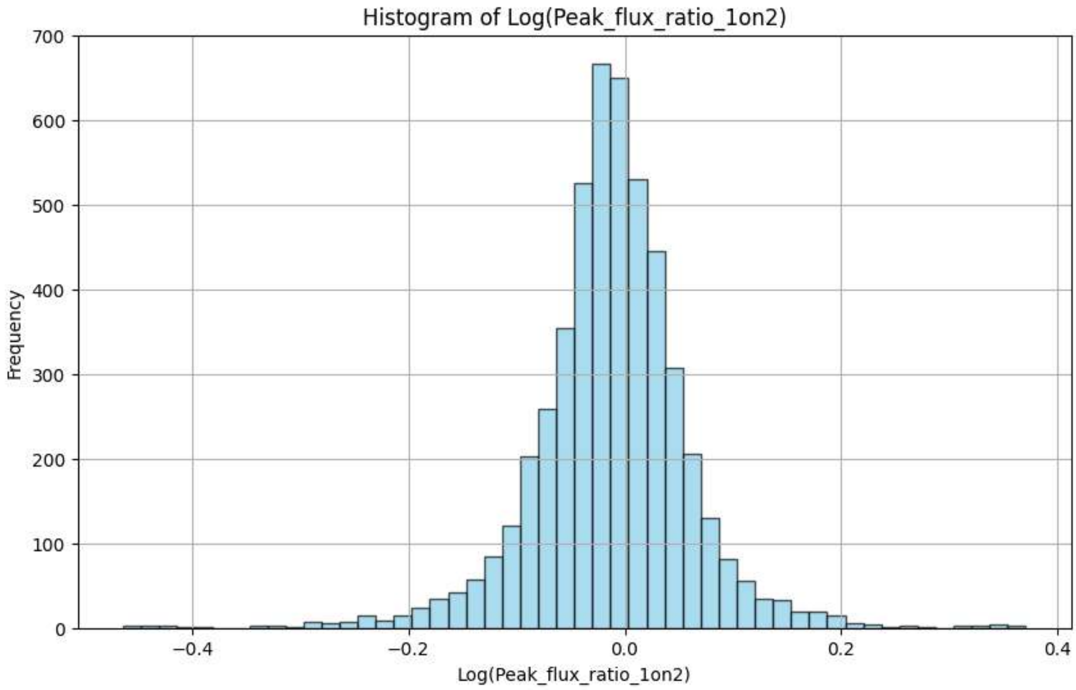

Figure 9: Histogram of Number of sources VS Peak flux ratio of both epochs (Peak_flux_epoch1 divided by Peak_flux_epoch2). We are concerned about the sources >log10 and <log10.

As these two filtered epochs were skymatched, we obtained 5050 common data points/-sources. This indicates that 5050 data points showed minimal flux changes during the time period of Epoch 1 and Epoch 2.

Below are sample cutouts of this filtered data from CIRADA:

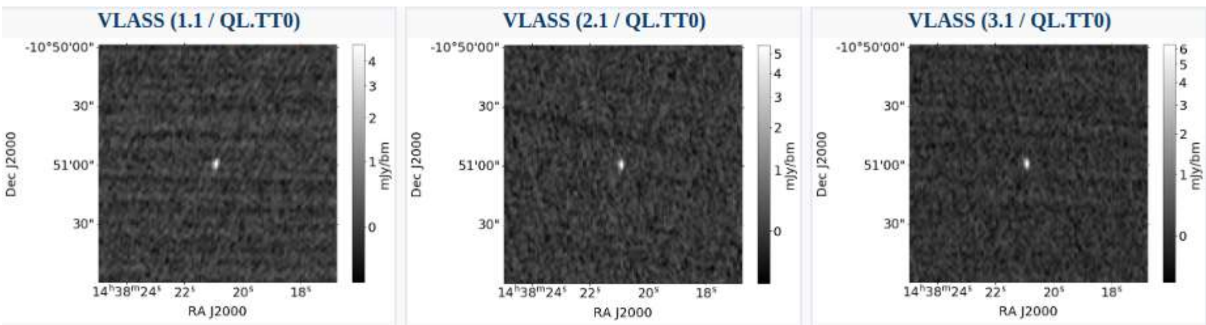

Figure 10: VLASS1QLCIR J143820.93-105059.4

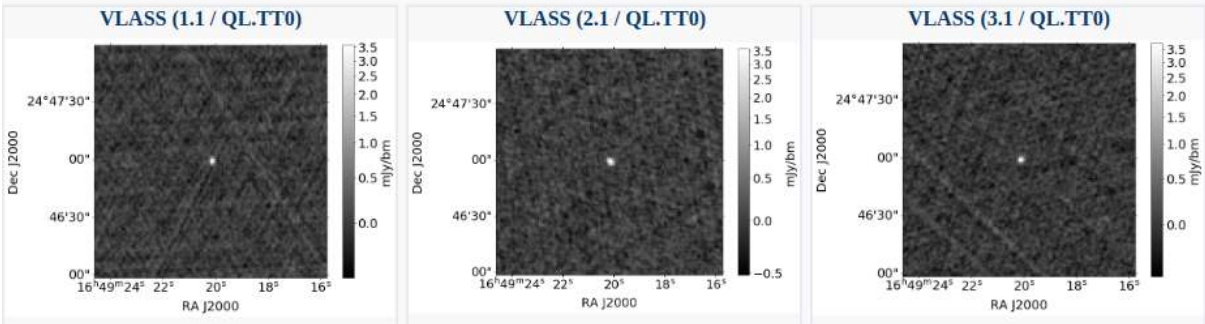

Figure 11: VLASS1QLCIR J164920.12+244659.0

Later, we shortlisted SNe and TDE catalogues for our concerned time range. We matched SNe and TDE with our VLASS data. From these matched catalogues, by comparing with VLA's FIRST survey and also using NED (NASA's Extragalactic Database), we shortlisted SNe which were not previously observed in Radio regime. Thus we made a catalogue of 153 Supernovae. Some of these are shown in the table below:

SN name	**RA**	**Dec**	**VLASS Counterpart Coordinate**	**RA_2**	**DEC_2**	**Radio Flux Detected (Peak Flux)**	**Epoch of Detection**
Gaia21edl	216.2590	76.9109	VLASS2QLCIR J142502.37+765438.8	216.2599	76.9108	17.9	2015/01/11
AT2018hlo	307.7201	-34.8829	VLASS2QLCIR J203052.75-345258.2	307.7198	-34.8829	0.8	2018/10/13
AT2019bx	11.7845	-7.0794	VLASS2QLCIR J004708.26-070445.5	11.7844	-7.0793	6.6	2019/01/04
AT2019baf	268.0006	65.6267	VLASS2QLCIR J175200.11+653736.1	268.0005	65.6267	1.6	2019/01/09
Gaia16afy	273.3967	31.7382	VLASS2QLCIR J181335.19+314417.5	273.3966	31.7382	95.1	2016/02/24
iPTF15dof	351.0550	35.1270	VLASS2QLCIR J232413.15+350737.0	351.0548	35.1270	1.1	2015
Gaia19cwg	64.2187	1.0900	VLASS2QLCIR J041652.49+010523.9	64.2187	1.0900	61.6	2019/07/06
SN2018zf	291.8312	65.5651	VLASS2QLCIR J192719.54+653354.3	291.8315	65.5651	10.1	2018/03/03
AT2018lte	89.5485	50.5538	VLASS2QLCIR J055811.66+503313.6	89.5486	50.5538	1.0	2018/12/16
AT2019eld	180.8938	-13.4264	VLASS2QLCIR J120334.50-132535.2	180.8938	-13.4264	72.3	2019/05/04
..	..	..	..	..	..	..	..

Table 1: Supernovae and Transient Catalogue Matching

Below are the sample cutouts of this SNe catalogue:

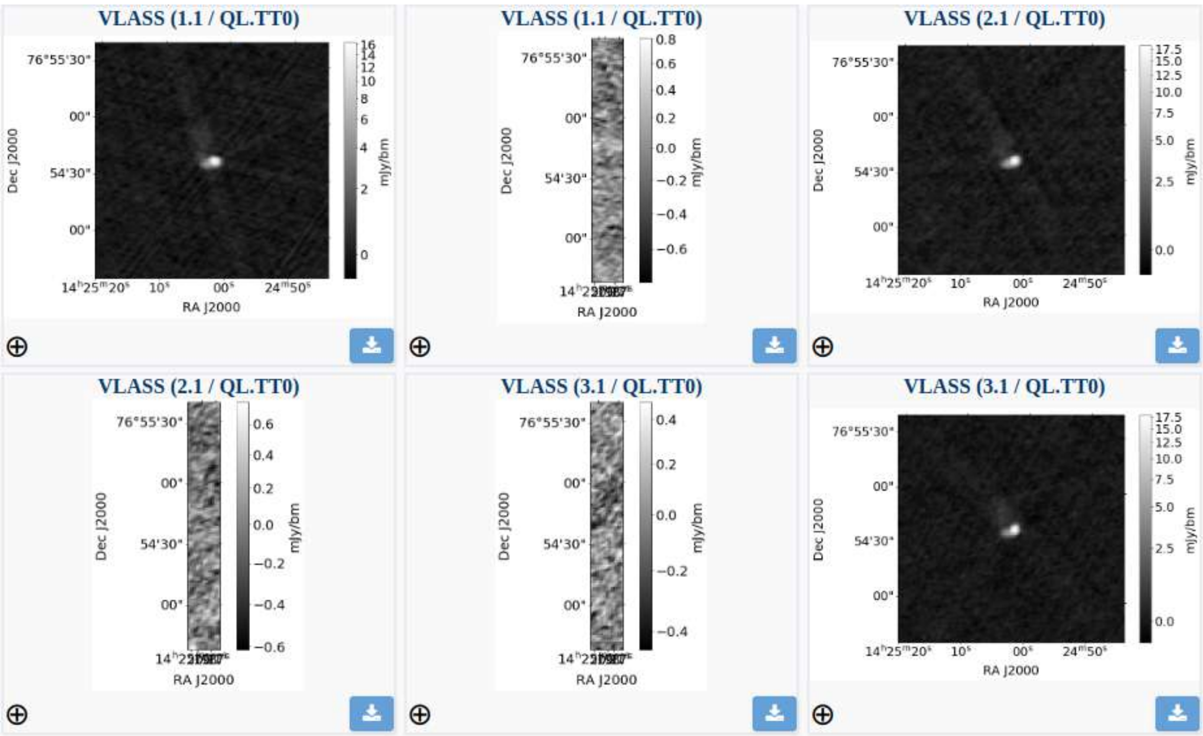

Figure 12: `VLASS2QLCIR J142502.37+765438.8` : Radio Counterpart of `Gaia21edl`

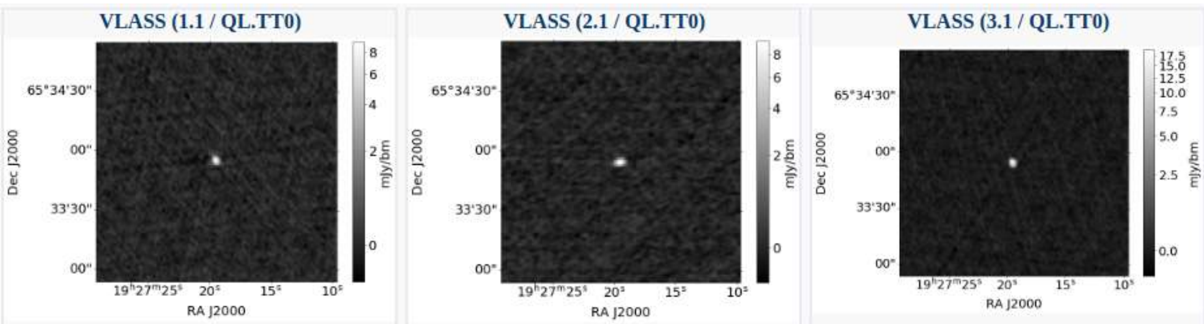

Figure 13: `VLASS2QLCIR J192719.54+653354.3` : Radio Counterpart of `SN2018zf`

6 Conclusion and Future Scope

The analysis of the Very Large Array Sky Survey (VLASS) data across Epoch 1 and Epoch 2 has provided significant insights into the distribution and characteristics of radio sources in the sky. The Aitoff projections for both epochs offer a comprehensive visual representation of the total flux distribution, highlighting regions with significant radio emissions and allowing for a detailed comparison between the two epochs.

The histogram analysis of the peak flux (S_peak) values demonstrates the impact of data filtering on the dataset. The quality_flag == 0 filter effectively removes less reliable detections, ensuring that the remaining data points are of high reliability. This step is crucial for minimizing false detections and enhancing the accuracy of subsequent transient studies.

The scatter plots of the peak flux versus the ratio of peak flux to total flux further validate the effectiveness of the filtering process. The final filtered data points, represented by red dots, show the most reliable detections, minimizing the impact of noise and ensuring the accuracy of the analysis. The comparison between Epoch 1 and Epoch 2 reveals the stability of these sources over time.

The sky matching and coordinate matching techniques employed in this study enabled the identification of transient sources with high precision. The matching with supernova (SNe) and transient catalogs, followed by cross-referencing with the VLA's FIRST survey and NED, led to the creation of a catalogue of 153 supernovae that were not previously observed in the radio regime. This catalogue represents a valuable resource for future studies of these transient events.

Similarly, by filtering our sources based on existing SNe catalogues, we can also compare them with TDE catalogues to uncover additional transient events observed in VLASS. Once we detect transient candidates, we can further analyze their types and gather more information, thereby contributing significantly to the field of transient astronomy.

Overall, this study underscores the importance of rigorous data filtering and matching techniques in the analysis of large-scale radio surveys. The results obtained from VLASS Epoch 1 and Epoch 2 provide a solid foundation for further investigation of transient phenomena and contribute to our understanding of the dynamic radio sky.

References

[1] B P Abbott, R Abbott, T D Abbott, et al. "Observation of gravitational waves from a binary black hole merger". In: Physical Review Letters 116.6 (2016), p. 061102.

[2] Peter E Dewdney et al. "The Square Kilometre Array". In: Proceedings of the IEEE 97.8 (2009), pp. 1482–1496.

[3] Neil Gehrels, Enrico Ramirez-Ruiz, and Derek B Fox. "Gamma-ray bursts in the Swift era". In: Annual Review of Astronomy and Astrophysics 47 (2009), pp. 567–617.

[4] Yjan A Gordon, Michelle M Boyce, Christopher P O'Dea, et al. "A Quick Look at the 3 GHz Radio Sky. I. Source Statistics from the Very Large Array Sky Survey". In: The Astrophysical Journal Supplement Series 255.1 (2021), p. 30.

[5] Simon Johnston et al. "Science with ASKAP-The Australian Square-Kilometre-Array Pathfinder". In: Publications of the Astronomical Society of Australia. Vol. 24. 4. CSIRO PUBLISHING. 2007, pp. 174–188.

[6] Nicholas Kaiser, William Burgett, Ken Chambers, et al. "The Pan-STARRS wide-field optical/NIR imaging survey". In: SPIE Astronomical Telescopes+ Instrumentation. International Society for Optics and Photonics. 2010, 77330E–77330E.

[7] Mark Lacy, Stefi A Baum, Claire J Chandler, et al. "The Very Large Array Sky Survey (VLASS)". In: Publications of the Astronomical Society of the Pacific 132.1009 (2020), p. 035001.

[8] Mark Lacy, Stefi A Baum, Claire J Chandler, et al. VLASS Project Memo #13: Pilot and Epoch 1 Quick Look Data Release. https://library.nrao.edu/public/memos/vla/vlass/VLASS_013.pdf. 2019.

[9] Weidong Li et al. "Nearby supernova rates from the Lick Observatory Supernova Search. III. The rate-size relation, and the rates as a function of galaxy Hubble type and colour". In: Monthly Notices of the Royal Astronomical Society 412.3 (2011), pp. 1441–1472.

[10] Duncan R Lorimer et al. "A bright millisecond radio burst of extragalactic origin". In: Science 318.5851 (2007), pp. 777–780.

[11] NASA/IPAC Extragalactic Database Cone Search. https://ned.ipac.caltech.edu/conesearch. Accessed: 2024-07-16.

[12] Pacholczyk. Radio astrophysics: Nonthermal processes in galactic & extragalactic sources. Freeman, 1970.

[13] Emily Petroff, Jason W T Hessels, and Duncan R Lorimer. "Fast radio bursts". In: Astronomy and Astrophysics Review 27.1 (2019), pp. 1–75.

[14] Tsvi Piran. "The physics of gamma-ray bursts". In: Reviews of Modern Physics 76.4 (2004), p. 1143.

[15] Pablo Reig. "X-ray binaries". In: Astrophysics and Space Science 332 (2011), pp. 1–29.

[16] Nathan Smith et al. "Observed fractions of core-collapse supernova types and initial masses of their single and binary progenitor stars". In: Monthly Notices of the Royal Astronomical Society 412.3 (2011), pp. 1522–1538.

[17] M. B. Taylor. TOPCAT & STIL: Starlink Table/VOTable Processing Software. http://www.star.bris.ac.uk/~mbt/topcat/. Accessed: 2024-07-16. 2005.

[18] AR Thompson, JM Moran, and GW Swenson. Interferometry and synthesis in radio astronomy. Wiley-Interscience New York, 1980.

[19] Christopher Thompson and Robert C Duncan. "The soft gamma repeaters as very strongly magnetized neutron stars—I. Radiative mechanism for outbursts". In: Monthly Notices of the Royal Astronomical Society 275.2 (1995), pp. 255–300.

[20] Transient Name Server Catalog. `https://github.com/astrocatalogs/transient-table`. Accessed: 2024-07-16.

[21] Transient Name Server Catalog. `https://github.com/astrocatalogs/transient-table`. Accessed: 2024-07-16.

[22] VLASS Quick Look Catalog. `https://cirada.ca/vlasscatalogueql0`. Accessed: 2024-07-16.

[23] Stan E Woosley and Joshua S Bloom. "The supernova gamma-ray burst connection". In: Annual Review of Astronomy and Astrophysics 44 (2006), pp. 507–556.

A Appendix

A.1 Justification for 1 Arcsecond Tolerance

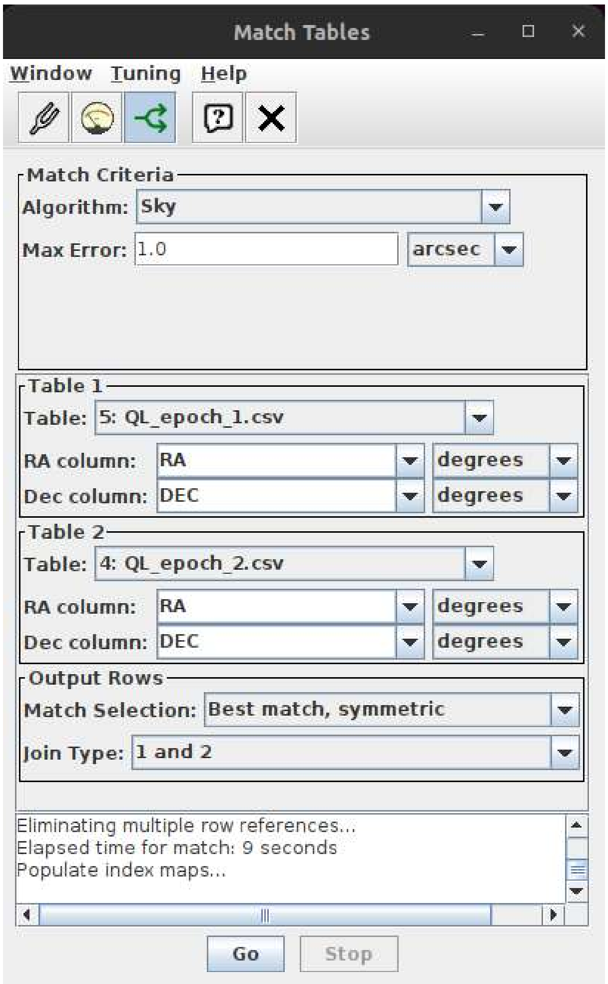

Figure 14: Coordinate matching of Epochs

For accurate identification and analysis of transient sources, precise coordinate matching between different epochs is crucial. Given the inherent positional uncertainties in the VLASS data, we have adopted an arbitrary error of 1 arcsecond for our coordinate matching process.

The reasoning behind this choice is based on the following considerations:

- For declinations greater than -20 degrees, the positional uncertainty is approximately 0.5 arcseconds. This uncertainty increases to about 1 arcsecond at declination -40 degrees. The increase in positional uncertainty is primarily due to uncorrected wide-field imaging effects.
- Additionally, there may be residual effects due to phase errors introduced by the troposphere and ionosphere, which can cause small shifts in the apparent positions of radio sources.
- By using a 1 arcsecond tolerance, we account for the maximum expected positional uncertainty, ensuring that potential matches are not excluded due to these positional inaccuracies.

A.2 Matched and Unmatched Sources

The following Python code was used to perform coordinate matching and identify matched and unmatched sources in the VLASS data:

```python
import pandas as pd
import numpy as np
# Load the dataframes
df1 = pd.read_csv('filtered_data_points_epoch1.csv')
df2 = pd.read_csv('filtered_data_points_epoch2.csv')
# Define a function to calculate the angular distance between two points
def angular_distance(ra1, dec1, ra2, dec2):
    # Convert RA and DEC to radians
    ra1_rad = ra1 * (2 * np.pi / 360)
    dec1_rad = dec1 * (2 * np.pi / 360)
    ra2_rad = ra2 * (2 * np.pi / 360)
    dec2_rad = dec2 * (2 * np.pi / 360)

    # Calculate angular distance using the haversine formula
    delta_ra = ra2_rad - ra1_rad
    delta_dec = dec2_rad - dec1_rad
    a = np.sin(delta_dec/2)**2 + np.cos(dec1_rad) * np.cos(dec2_rad) * np.sin(delta_ra/2)**2
    c = 2 * np.arctan2(np.sqrt(a), np.sqrt(1-a))
    distance = c
    return distance

# Perform crossmatching
matched_sources = []
max_radius = 2.5 / 3600  # Maximum angular distance in radians (2.5 arcseconds)
for idx1, source1 in df1.iterrows():
    for idx2, source2 in df2.iterrows():
        distance = angular_distance(source1['RA'], source1['DEC'], source2['RA'], source2['DEC'])
        if distance <= max_radius:
            matched_sources.append({'source_epoch_1': idx1, 'source_epoch_2': idx2, 'angular_distance': distance})

# Convert the list of matched sources to a dataframe
matched_df = pd.DataFrame(matched_sources)

# Save the matched dataframe to a CSV file
matched_df.to_csv('matched_sources_python.csv', index=False)

# Find rows in df1 that are not in matched_df
unmatched_epoch1 = df1[~df1.index.isin(matched_df['source_epoch_1'])]

# Find rows in df2 that are not in matched_df
unmatched_epoch2 = df2[~df2.index.isin(matched_df['source_epoch_2'])]

# Save the unmatched dataframes to separate CSV files
unmatched_epoch1.to_csv('unmatched_sources_epoch1.csv', index=False)
unmatched_epoch2.to_csv('unmatched_sources_epoch2.csv', index=False)
```

Listing 1: Coordinate Matching for Matched and Unmatched Sources

A.3 Aitoff Projection

The following Python code was used to create the Aitoff projection plot for Epoch 1 data:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Load the epoch 1 data
epoch_1 = pd.read_csv('Epoch_1_f04d2noE.csv')

# Ensure the required columns exist
required_columns = ['RA', 'DEC', 'Isl_rms']
for col in required_columns:
    if col not in epoch_1.columns:
        raise ValueError(f"Column {col} is missing from the data")

# Calculate the mean and standard deviation of Isl_rms
mean_isl_rms = np.mean(epoch_1['Isl_rms'])
std_isl_rms = np.std(epoch_1['Isl_rms'])

print("Standard Deviation of Isl_rms:", std_isl_rms)
print("Mean of Isl_rms:", mean_isl_rms)

# Define the 1-sigma range
lower_bound = mean_isl_rms - 1 * std_isl_rms
upper_bound = mean_isl_rms + 1 * std_isl_rms

# Filter data within 1-sigma range
filtered_data = epoch_1[(epoch_1['Isl_rms'] >= lower_bound) & (epoch_1['
    Isl_rms'] <= upper_bound)]

# Check the range of Isl_rms in the filtered data
print("Filtered Isl_rms min:", filtered_data['Isl_rms'].min())
print("Filtered Isl_rms max:", filtered_data['Isl_rms'].max())

# Convert RA and DEC to radians for the Aitoff projection
filtered_data = filtered_data.copy()  # Avoid SettingWithCopyWarning
filtered_data['RA_rad'] = np.deg2rad(filtered_data['RA'] - 180)  # Shift RA
     by 180 degrees for the Aitoff projection
filtered_data['DEC_rad'] = np.deg2rad(filtered_data['DEC'])

# Set the limits for the color scale based on the filtered data
vmin = filtered_data['Isl_rms'].min()
vmax = filtered_data['Isl_rms'].max()

# Create the Aitoff projection plot for epoch 1
plt.figure(figsize=(12, 6))
ax = plt.subplot(111, projection="aitoff")
sc = ax.scatter(filtered_data['RA_rad'], filtered_data['DEC_rad'], c=
    filtered_data['Isl_rms'], cmap='viridis', s=0.01, vmin=vmin, vmax=vmax)

# Add colorbar
cbar = plt.colorbar(sc, orientation='horizontal', pad=0.1)
cbar.set_label('Isl rms [Jy/beam]')

# Set RA ticks and labels
ra_ticks = np.linspace(-np.pi, np.pi, 7)  # Set 6 ticks to include 0h in
    the middle
```

```python
ra_labels = ['12h', '8h', '4h', '0h', '20h', '16h', '12h']   # Adjust labels so 0h is in the middle
ax.set_xticks(ra_ticks)
ax.set_xticklabels(ra_labels)

# Add labels and grid
plt.grid(True)
plt.title('Aitoff Projection of VLASS Epoch 1 Filtered Isl')

# Adjust layout to prevent overlap
plt.tight_layout()

# Save the figure with higher DPI
plt.savefig('Aitoff_Proj_Epoch_1.png', dpi=600, bbox_inches='tight')

# Show the plot
plt.show()
```

Listing 2: Creating Aitoff Projection for Epoch 1

A.4 JSON to CSV File Conversion

The following Python code was used to convert .JSON files to a .CSV file:

```python
import os
import json
import pandas as pd

# Directory containing JSON files
json_dir = 'sne-2010-2014-master'

# List to store the processed data
data_list = []

# Columns of interest
columns = [
    'name', 'sources', 'alias', 'claimedtype', 'comovingdist', 'dec', 'discoverdate', 'discoverer', 'host',
    'lumdist', 'maxabsmag', 'maxvisualabsmag', 'maxvisualdate', 'ra', 'redshift', 'velocity', 'photometry'
]

# Function to extract data from the JSON structure
def extract_data(json_data, name):
    data = {col: None for col in columns}
    data['name'] = name

    # Extract sources as concatenated string
    sources = json_data.get('sources', [])
    data['sources'] = '; '.join([src.get('reference', '') for src in sources])

    # Extract aliases as concatenated string
    aliases = json_data.get('alias', [])
    data['alias'] = '; '.join([alias.get('value', '') for alias in aliases])

```

```python
    # Extract claimedtype as concatenated string
    claimedtypes = json_data.get('claimedtype', [])
    data['claimedtype'] = '; '.join([ct.get('value', '') for ct in
        claimedtypes])

    # Extract comovingdist value
    comovingdist = json_data.get('comovingdist', [])
    if comovingdist:
        data['comovingdist'] = comovingdist[0].get('value', '')

    # Extract dec value
    dec = json_data.get('dec', [])
    if dec:
        data['dec'] = dec[0].get('value', '')

    # Extract discoverdate value
    discoverdate = json_data.get('discoverdate', [])
    if discoverdate:
        data['discoverdate'] = discoverdate[0].get('value', '')

    # Extract discoverer value
    discoverer = json_data.get('discoverer', [])
    if discoverer:
        data['discoverer'] = discoverer[0].get('value', '')

    # Extract host value
    host = json_data.get('host', [])
    if host:
        data['host'] = host[0].get('value', '')

    # Extract lumdist value
    lumdist = json_data.get('lumdist', [])
    if lumdist:
        data['lumdist'] = lumdist[0].get('value', '')

    # Extract maxabsmag value
    maxabsmag = json_data.get('maxabsmag', [])
    if maxabsmag:
        data['maxabsmag'] = maxabsmag[0].get('value', '')

    # Extract maxvisualabsmag value
    maxvisualabsmag = json_data.get('maxvisualabsmag', [])
    if maxvisualabsmag:
        data['maxvisualabsmag'] = maxvisualabsmag[0].get('value', '')

    # Extract maxvisualdate value
    maxvisualdate = json_data.get('maxvisualdate', [])
    if maxvisualdate:
        data['maxvisualdate'] = maxvisualdate[0].get('value', '')

    # Extract ra value
    ra = json_data.get('ra', [])
    if ra:
        data['ra'] = ra[0].get('value', '')

    # Extract redshift value
    redshift = json_data.get('redshift', [])
    if redshift:
```

```
        data['redshift'] = redshift[0].get('value', '')

    # Extract velocity value
    velocity = json_data.get('velocity', [])
    if velocity:
        data['velocity'] = velocity[0].get('value', '')

    # Extract photometry as concatenated string
    photometry = json_data.get('photometry', [])
    data['photometry'] = '; '.join([str(ph.get('value', '')) for ph in
        photometry])

    return data

# Read and process JSON files
json_files = [f for f in os.listdir(json_dir) if f.endswith('.json')]

for filename in json_files:
    file_path = os.path.join(json_dir, filename)
    with open(file_path, 'r') as file:
        json_data = json.load(file)
        for name, content in json_data.items():
            data = extract_data(content, name)
            data_list.append(data)

# Convert to DataFrame
df = pd.DataFrame(data_list)

# Save to CSV
csv_file_path = 'sne-2010-14_full.csv'
df.to_csv(csv_file_path, index=False)

print(f"Data successfully saved to {csv_file_path}")
```

Listing 3: JSON to CSV File Conversion

A.5 BIN to CSV

The following Python code was used to convert .bin files to a .CSV file:

```python
import pandas as pd
# Define the input and output file paths
input_bin_file = 'SNe/catalog_14dec17.bin'
output_csv_file = 'SNe/catalog_14dec17_final.csv'
# Define column names based on the provided header
column_names = [
    "RA", "Dec", "P(S)", "Fpeak", "Fint", "RMS", "Maj", "Min", "PA",
    "fMaj", "fMin", "fPA", "Field", "#", "SDSS Sep", "SDSS i", "SDSS Cl", "#",
    "2MASS Sep", "2MASS K", "Epoch Mean-yr", "Epoch Mean-MJD", "Epoch rms-MJD"
]
# Read the binary file as a text file
with open(input_bin_file, 'r') as file:
    lines = file.readlines()
# Skip the header lines (first two lines) and split the rest into columns
data = []
for line in lines[2:]:
    # Split the line by spaces while handling multiple spaces
    split_line = line.split()
    # Reconstruct the columns manually to handle the variable widths
    row = [
        ' '.join(split_line[0:3]),      # RA
        ' '.join(split_line[3:6]),      # Dec
        split_line[6],                  # P(S)
        split_line[7],                  # Fpeak
        split_line[8],                  # Fint
        split_line[9],                  # RMS
        split_line[10],                 # Maj
        split_line[11],                 # Min
        split_line[12],                 # PA
        split_line[13],                 # fMaj
        split_line[14],                 # fMin
        split_line[15],                 # fPA
        split_line[16],                 # Field
        split_line[17],                 # #
        split_line[18],                 # SDSS Sep
        split_line[19],                 # SDSS i
        split_line[20],                 # SDSS Cl
        split_line[21],                 # 2MASS Sep
        split_line[23],                 # 2MASS K
        split_line[24],                 # Epoch Mean-yr
        split_line[25],                 # Epoch Mean-MJD
        split_line[26]                  # Epoch rms-MJD
    ]
    data.append(row)
# Create a DataFrame and save to CSV
df = pd.DataFrame(data, columns=column_names)
df.to_csv(output_csv_file, index=False)
print(f"Data successfully converted from {input_bin_file} to {output_csv_file}")
```

Listing 4: .bin to .csv File Conversion

A.6 HTML to CSV

The following Python code was used to convert HTML tables to CSV files:

```python
import pandas as pd
from bs4 import BeautifulSoup

# Step 1: Load the HTML file
html_file_path = '/home/metyche/Data/Transients/TDE/ASAS-SN Transients.html'

with open(html_file_path, 'r', encoding='utf-8') as file:
    soup = BeautifulSoup(file, 'html.parser')

# Step 2: Find the table
table = soup.find('table')

# Step 3: Define headers
headers = [
    "ASAS-SN ID", "Other IDs", "ATEL", "RA", "Dec", "Discovery (UT)",
    "V/g (mag)", "SDSS image", "DSS image", "Vizier data",
    "Spectroscopic Class", "Comments"
]

# Step 4: Extract rows
rows = []
for tr in table.find_all('tr'):
    cells = tr.find_all('td')
    if cells:
        row = []
        for idx, cell in enumerate(cells):
            text = cell.get_text(strip=True)
            if idx in [7, 8, 9]:  # Columns with links (SDSS, DSS, Vizier)
                link = cell.find('a')['href'] if cell.find('a') else ''
                row.append(f'{text} ({link})')
            else:
                row.append(text)
        # Ensure each row has exactly 12 elements (match the number of headers)
        while len(row) < len(headers):
            row.append('')
        rows.append(row)

# Step 5: Create DataFrame and save as CSV
df = pd.DataFrame(rows, columns=headers)
df.to_csv('transient_catalogue_with_links.csv', index=False)

print('Table saved as transient_catalogue_with_links.csv')
```

Listing 5: HTML to CSV Conversion