NenuFAR LT02 Imaging: From Observation to Detection

Author: Xiang Zhang
Date: 18 August 2025

---

1. Overview

Finding exoplanets in NenuFAR images sounds exciting - but how exactly do we attempt it? This document introduces the current methods used in the LT02 (star/exoplanet) project, from observation to data processing and candidate verification. You're welcome to provide feedback or build your own pipeline based on these foundations!

Currently, there is no dedicated publication describing the full pipeline. To acknowledge the developers and the work behind it, please cite the following paper if you use this pipeline in your research:

Zhang, X., P. Zarka, J. N. Girard, C. Tasse, A. Loh, E. Mauduit, F. G. Mertens et al. "A circularly polarized low-frequency radio burst from the exoplanetary system HD 189733." Astronomy & Astrophysics 700 (2025): A140.

---

2. Observations

If you are interested in using NenuFAR to observe your favorite exoplanet or star (hereafter referred to as the **target**), the first step is to request observing time.

There are two options:

- **Submit a NenuFAR proposal** through the standard time allocation process.

- **Join the LT02 key project**, which already includes a program for exoplanetary and stellar observations. Contact the LT02 team to request that your TARGET be added to their observing list - this is often the easier route.

Piggyback Mode

NenuFAR supports "piggyback" observations, meaning you can record **both imaging and beamformed data simultaneously**. Since beamformed observations are relatively low-cost in terms of resources and disk space, we strongly recommend enabling them by default during imaging sessions.

Choosing Observing Parameters

Once observing time is allocated, the next step is defining the configuration:

- **Resolution**: We refer here to **L1 data**—the pre-processed visibilities. Raw L0 data are not retained due to their size. The standard resolution used in LT02 imaging is:
    - **1 second** temporal resolution
    - **90 kHz** frequency resolution (from 2 channels per 195 kHz subband)
- **Frequency Range**: This depends on the scientific goals. NenuFAR can observe in the 10–88 MHz range, but:
    - Frequencies **below 20 MHz** are affected by ionospheric turbulence
    - Frequencies **above 80 MHz** suffer from strong RFI (e.g., FM radio)
    - **Optimal range:** 50–60 MHz

  The full system allows observation of up to **244 subbands**, corresponding to a **47.5 MHz bandwidth**. Suggested configurations:
    - **Exoplanets**: 19.0–66.5 MHz (choose low frequencies due to CMI cutoff)
    - **Stars**: 30.8–78.3 MHz (best data quality)

You may also consider synchronizing frequency coverage with other instruments if conducting **multi-wavelength campaigns**.

🔍 *Don't Forget the Calibrator*

While in-field calibration is technically possible using sky models, it's not generally reliable. NenuFAR's short baselines and the proximity of many targets to the **Galactic plane** often result in images dominated by diffuse emission.

We therefore strongly recommend including a dedicated **calibrator** observation. Use one of the four primary calibrators, in order of preference: 1. Cyg A 2. Cas A 3. Vir A 4. Tau A

Calibrators should ideally be:

- Observed **immediately before or after** the target (or placed between two halves of the target session)

- Configured with the **same frequency range and resolution** as the target

- Observed for **about 15 minutes**

Example observations can be found on NenuFAR VCR:

- **GJ687 (exoplanet)** with calibrator Cyg A – Observations on 2025-07-06 22:00

- **LP212-62 (star)** with calibrator Vir A – Observations on 2025-01-15 00:00

⏱ *Choosing the Time of Observation*

The best observation times are typically during **late night hours**, when the temperature is low and ionospheric conditions are most stable. Observations during **early night** or **early morning** are also possible, though less favorable:

- **Early night** tends to suffer from RFI

- **Early morning** may include contamination from solar activity

For exoplanetary targets, **orbital phase coverage** is another critical consideration. Depending on the scientific goal, you may wish to:

- Observe the full orbital phase

- Target specific key phases (e.g., eclipse, periastron, conjunction)

To predict orbital phase coverage for any given time and date, use the orbital coverage calculator: https://github.com/zhangxiang-planet/orbital_coverage

---

3. Automatic Processing Pipeline

All LT02 imaging observations—regardless of who requested them—are processed centrally using the **automatic imaging pipeline**, available here:

https://github.com/zhangxiang-planet/exo_img_pipe

The pipeline is designed to run continuously on a **Nançay CEP** user account managed by a designated person (hereafter referred to as the **HOST**). The HOST is responsible for keeping the system operational and should ensure that:

- At least **2 TB of local disk space** is available for processing and temporary files

- All required dependencies and environmental configurations are properly installed

Once configured, the pipeline automatically detects new observations and handles data processing without the need for manual intervention.

3.1 Dependencies

The automatic imaging pipeline relies on a variety of software packages and tools. Due to the complexity of the environment, we strongly recommend that the **HOST** use a package or environment manager such as **Conda** to ensure reproducibility and ease of installation.

Below is a categorized list of required dependencies:

🧰 *Public Python Packages*

- **Numpy**, **Scipy**, **Astropy**, **Matplotlib**
  Standard scientific packages used for numerical operations, astronomical calculations, and plotting.
- **Dask**
  For efficient parallel processing, particularly when handling large datasets.
- **H5py**
  Used for reading and writing HDF5 files.

- **Astroquery**
  Enables querying external databases such as SIMBAD and the NASA Exoplanet Archive.

Workflow Orchestration

- **Prefect**
  A modern workflow management library used to schedule and manage the pipeline's modular tasks.

LOFAR Tools (available on the Nançay machine)

- **DP3** (Default Pre-Processing Pipeline)
  Used for averaging, flagging, and calibration of visibilities.
- **aoflagger**
  Automatic RFI flagging tool used during pre-processing.
- **WSClean**
  Wide-field interferometric imager used for producing sky maps.

NenuFAR-Specific Tools

- **Nenupy**
  Python interface for NenuFAR beamformed and imaging data access and manipulation.
- **Nenucal**
  Used for calibration and metadata handling specific to NenuFAR data products.

CASA-Compatible Tools

- **Casacore**
  A lightweight set of CASA-compatible libraries (installed on the Nançay system).
- **Casatools**
  Provides CASA functionalities accessible directly via Python.

Proprietary / Internal LOFAR Tools

These tools are not (entirely) publicly distributed. Please contact C. Tasse for access and support.

- **DDFacet**
Direction-dependent calibration and imaging tool.

- **KillMS**
Direction-dependent solver for gain calibration.

- **DynspecMS**
Used to generate dynamic spectra from calibrated Measurement Sets.

3.2. File Structure

At the top level, the pipeline consists of several Python scripts and supporting folders. Here's a detailed breakdown of the structure and their roles:

Top-Level Python Scripts:

- autopilot.py: Main script that runs the automatic pipeline. This is the primary script used by the HOST.

- data_storage.py: Used periodically (e.g., each semester) to archive processed data into /databf as Level 2 (L2) products.

- manual_inspect.py: Script for re-imaging selected detections.

- manual_inspect_fullband.py: Improved version of the above with full-band capability; recommended for most re-imaging tasks.

- period_search.py: Script to search for periodic signals from interesting targets.

- manual_drive*.py: Custom scripts used for specific targets or one-off analyses; not considered part of the core pipeline.

Folder: cal_models/

- CYG_A_lcs.skymodel, CAS_A_lcs.skymodel, TAU_A_lcs.skymodel, VIR_A_lcs.skymodel: Sky models for the main calibrators, derived from LOFAR observations. Used for absolute flux calibration.

Folder: catalogs/

- gaia_dr3_ucd.fits: Catalog of ultra-cool dwarfs from Gaia DR3. Used to locate scientifically interesting targets within the field of view.

Folder: regions/

- Ateam.reg: DS9 region file marking positions of bright A-team sources. Used for direction-dependent calibration and A-team subtraction.

- CasA.reg, CygA.reg: Individual region files for specific A-team members. These are legacy files and not actively used in the current version of the pipeline.

Folder: templates/

- DPPP*.parset: Parset files used by DP3 for averaging, flagging, and calibration steps.

- Nenufar64C1S.lua: RFI flagging strategy file used by aoflagger.

- Nenufar64C1S_FRB.lua: A variation of the above, optimized for specific science targets such as FRBs.

- bad_MA.toml, cali_tran.toml, calibrator.toml: Configuration templates used by nenucal to define calibration procedures and apply solutions.

- skip.txt: List of known-bad observations that should be skipped by the automatic pipeline. This list is updated continuously.

- template_DI.parset: A template parset for direction-independent imaging, used by DDFacet.

- Find_Bad_MAs_template.py: Script to analyze calibration solutions and identify malfunctioning mini-arrays.

- Make_Target_List_template.py: Generates a list of pencil beam directions for dynamic spectrum generation.

- Plot_target_distri_template.py: Creates a visual layout of pencil beam pointings overlaid on a multi-frequency synthesis (MFS) image.

- Noise_esti_template.py: Estimates noise across the dynamic spectra, used to normalize the data before source-finding.

3.3. Running the Pipeline

Once the environment is properly set up, running the pipeline is straightforward. Here are the recommended steps for the HOST to follow:

1. **Clone the Repository**
   Use git clone to copy the pipeline repository into your home directory:

   ```
   git clone https://github.com/zhangxiang-planet/exo_img_pipe
   ```

2. **Configure**

   Open the autopilot.py script and edit lines 22–30 to reflect the correct paths and filenames for your system. These include:

   - Path to the L1 data directory (i.e., watch_dir)
   - Output directory for intermediate and final products

3. **Launch a Persistent Terminal Session**
   Since the pipeline runs continuously in the background, it is recommended to use a terminal multiplexer like tmux or screen to keep the process alive:

   ```
   tmux new -s nenupipe
   ```

4. **Start the Pipeline**
   Navigate to the pipeline directory and launch the main driver:

   ```
   cd ~/exo_img_pipe
   python autopilot.py
   ```

5. **Relax and Monitor**
   Once started, the pipeline will automatically monitor for new L1 data, process it, and handle archiving and diagnostics. You can sip your coffee or tea while the pipeline does the heavy lifting.

You may detach from the terminal session using Ctrl+b d (if using tmux) and reattach later with:

```
tmux attach -t nenupipe
```

It's recommended to check logs and Prefect dashboard periodically to ensure tasks are running smoothly.

3.4. Pipeline Workflow

The core of the automatic pipeline is organized around a task-based dataflow using the **Prefect** orchestration framework. It operates in two main stages:

1. **Monitoring for New Data**
   The pipeline continuously runs a task called check_flow, which scans a designated watch_dir—the folder containing newly produced Level 1 (L1) data. This check is performed every six hours. When new L1 data is detected, check_flow triggers the main processing task exo_pipe, and temporarily pauses itself until that data is fully processed.

2. **Main Processing Workflow (exo_pipe)**
   The exo_pipe task executes a sequence of modular subtasks to process each observation. Here is a step-by-step breakdown:

   - **copy_calibrator_data and copy_target_data**
     Copies the raw calibrator and target datasets into the working directory for processing.

   - **identify_bad_mini_arrays**
     Performs an initial calibration on the calibrator. The resulting calibration solutions are analyzed to identify malfunctioning mini-arrays (MAs), such as those with discontinuous or abnormally low response. A list of bad MAs is saved for future reference.

   - **calibration_Ateam**
     Executes a second calibration round on the calibrator, now excluding the identified bad MAs for more robust gain solutions.

   - **apply_Ateam_solution**
     Applies the refined calibration solutions from the second round to the target field.

   - **subtract_Ateam**
     Performs direction-dependent calibration and subtraction of strong off-axis sources (A-team), which can contaminate the field. This improves dynamic range and reduces artefacts.

   - **dynspec**
     Generates dynamic spectra (Stokes I, Q, U, V) for a few hundred "pencil beams"

that sample both the locations of scientific targets and field grids. These spectra form the basis for transient searches.

- **source_find_v**
  Scans the Stokes V dynamic spectra for transient burst candidates.
- **source_find_i**
  Performs the same operation on Stokes I spectra.
- **clear_up**
  Cleans up intermediate files to reduce storage use, leaving only the images and dynamic spectra needed for archiving and analysis.

3.5. Data Products

For each observation, the pipeline generates two main output directories: - One for the **calibrator** - One for the **target**

These contain all intermediate and final data products needed for quality control, reprocessing, and scientific analysis.

Calibrator Folder

This directory includes diagnostic files and calibration solutions:

- bad_MA.txt
  List of malfunctioning mini-arrays (MAs) identified during initial calibration.
- Amp_sol_highlighted.png, ratio_sol_highlighted.png, Phase_sol_highlighted.png, Diff_phase_sol_highlighted.png
  Diagnostic plots visualizing the amplitude and phase of the calibration solutions. Problematic MAs are highlighted for easy review.
- GSB.MS/instrument_ddecal.h5
  The final calibration solution file (direction-independent) generated from the calibrator data. This is applied to the corresponding target.

Target Folder

This directory contains beam layouts, diagnostic plots, and dynamic spectra:

- Target.txt
  List of pencil beam directions used in the observation, covering both scientific targets and grid points.
- target_distri.png
  Visual map of pencil beam positions overlaid on a multi-frequency synthesis (MFS) image of the target field.

- Archive_images/
  MFS images produced from the target visibilities. Useful for assessing calibration and imaging quality.

- dynamic_spec_DynSpecs_GSB.MS/
  Core directory for all dynamic spectrum products:

    - TARGET/
      Full-Stokes (IQUV) dynamic spectra for each pencil beam.

    - mean_std_i.fits, mean_std_v.fits
      Mean and standard deviation across the field for Stokes I and V. Used to normalize the dynamic spectra.

    - weighted_dynamic_spec_i/, weighted_dynamic_spec_v/
      Normalized Stokes I and V dynamic spectra, weighted by background statistics.

    - detected_dynamic_spec_i/, detected_dynamic_spec_v/
      Subset of dynamic spectra where candidate transients have been flagged in I or V.

    - ${observation}_png_i/, ${observation}_png_v/
      PNG visualizations of detected bursts, useful for manual review and presentations.

---

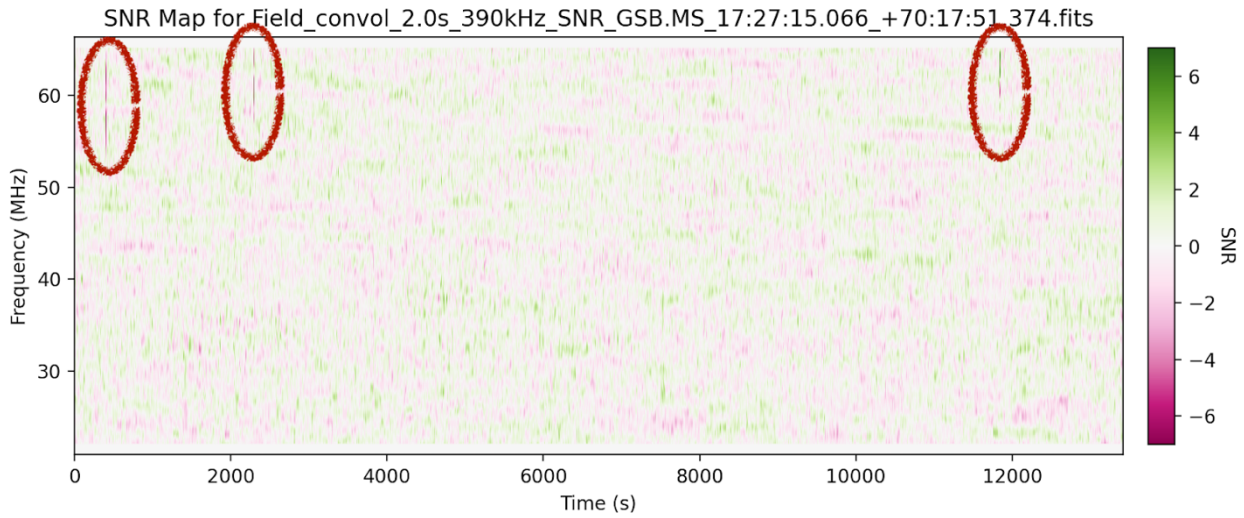
4. Verification

What should you do if potential transient detections are made?

First—don't get too excited just yet! Many types of contamination can mimic real signals. Based on our experience up to July 2025, **over 95% of transient candidates** in our data turn out to be false positives, with **Starlink satellites** being the common culprit. 😣

🚀 *Satellite Contamination*

Satellites, particularly Starlink, often leave clear signatures in dynamic spectra. Because they move across the sky, they appear in **multiple pencil beams at different time stamps**. Even faint satellites that only show up in one beam often produce **short, broadband bursts**—usually centered near **60 MHz**.



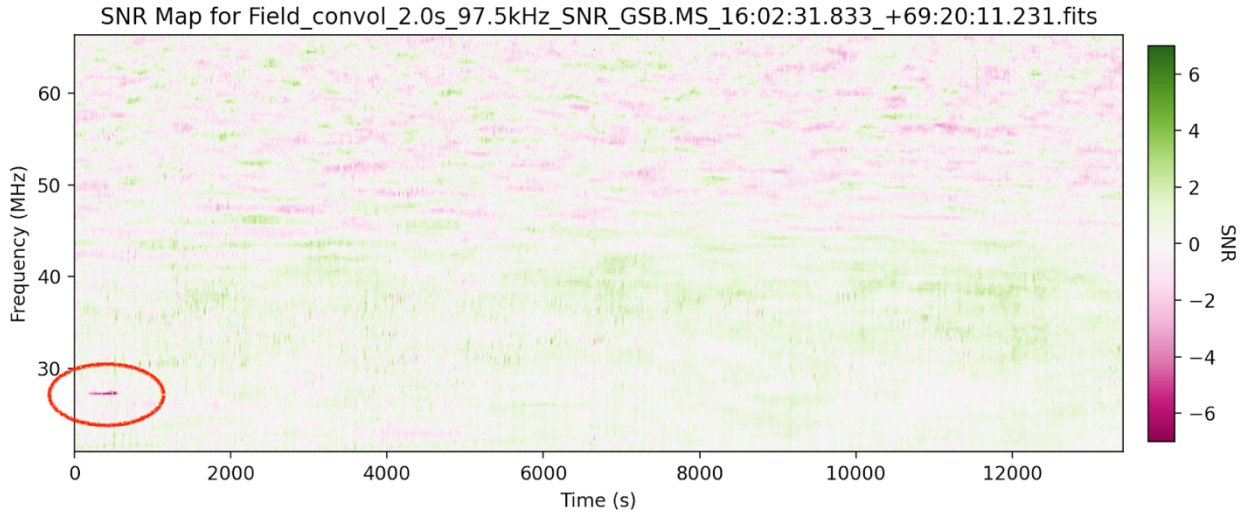
Example: A Stokes V dynamic spectrum from July 2025 showing multiple bursts caused by Starlink satellites passing by.

✈️ *Aircraft Contamination*

Aircraft can produce similar effects to satellites: **broadband, short-duration bursts** visible in multiple beams at slightly staggered times. These are usually easy to identify with experience.

📡 *Residual RFI*

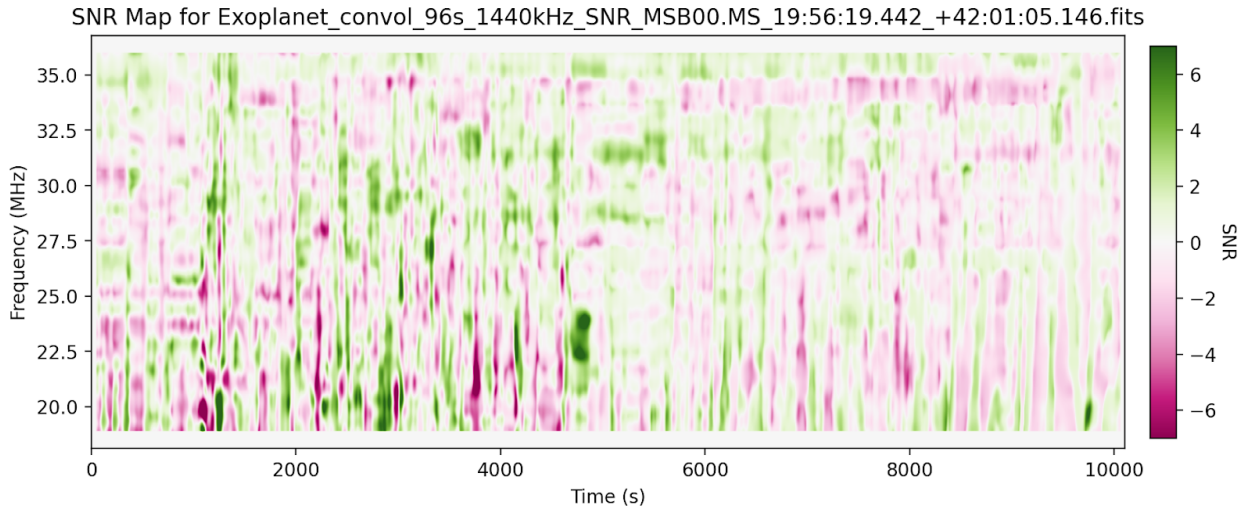
Not all RFI is successfully flagged. Unflagged RFI typically manifests as **bright, narrowband signals** affecting many beams simultaneously. These signals are often persistent across multiple time steps and easily mistaken for bursts if not carefully vetted.



Example: Residual RFI after imperfect flagging, showing narrow dashes in the dynamic spectrum.

⛈️ *Bad Weather Conditions*

Storms or unstable ionospheric conditions can produce bursts across the field. These show up in **multiple beams** and across different times. In some cases, beamformed data will show clear signs of **lightning strikes**.



Example: Dynamic spectra collected on 2023-09-03 during a thunderstorm event. The data was heavily contaminated.

🧠 *General Tips for Inspection*

Be cautious of candidates that:

- Are narrowband or extremely short in duration
- Appear during known periods of bad weather or poor data quality
- Affect many beams simultaneously without an astrophysical explanation

If the data quality looks good and the burst is neither narrowband nor short-lived, can we trust it?

The answer is still: **not immediately.** One of the most deceptive contamination types is due to **bright background sources**.

✨ *Artefacts from Background Sources*

Due to: - **Imperfect calibration** - **Ionospheric turbulence**

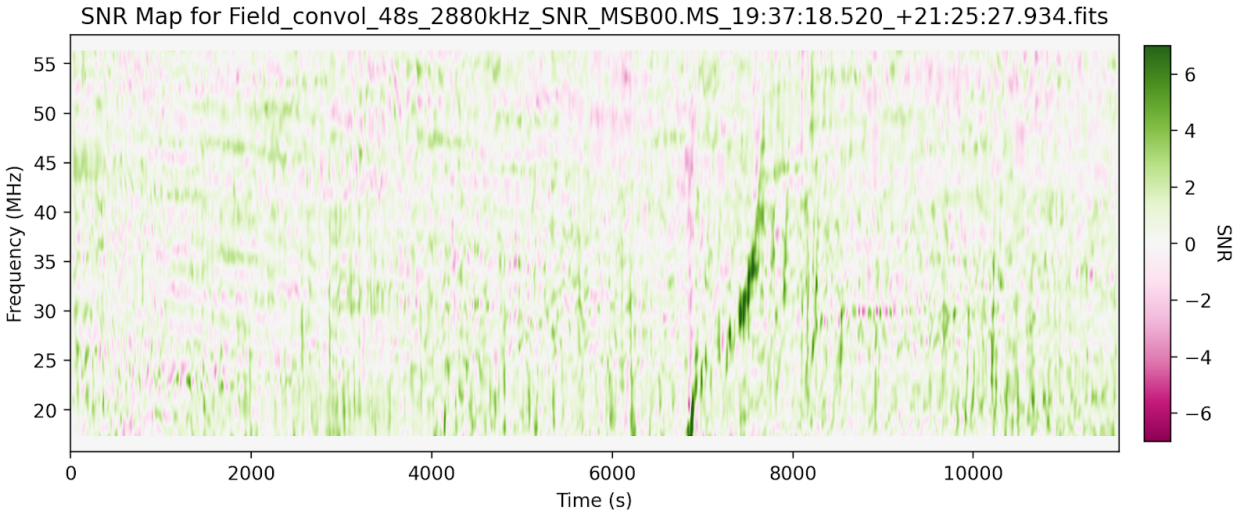
The flux of bright sources in the field can vary or drift in position, leading to **artificial variability in Stokes I**, and **leakage into Stokes V**.

This can create **false transient candidates** that mimic real astrophysical signals.

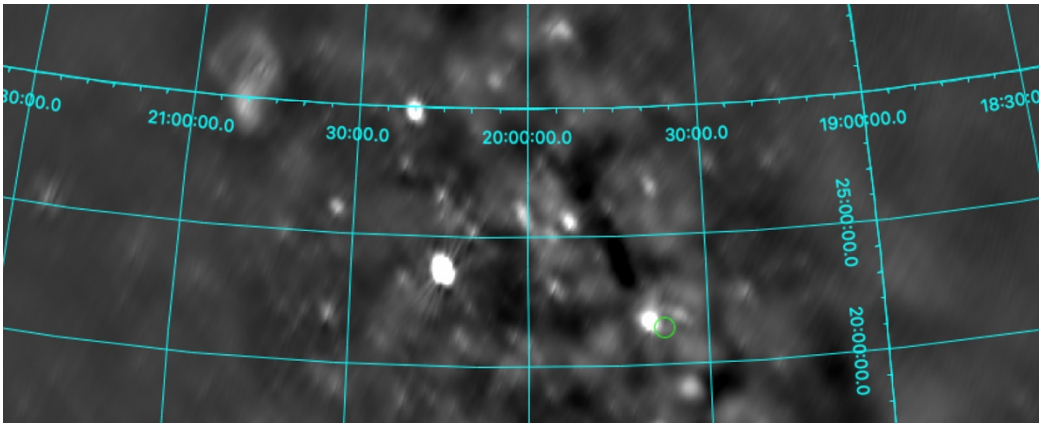
🔍 *Re-imaging as Verification*

We strongly recommend re-imaging all interesting candidates. Specifically:

- Image the **time–frequency window** of the burst
- Include a **buffer window before and after** the burst
- Inspect whether the burst aligns spatially with a known bright source
- Compare the source shape to the expected PSF of the array



Example: A transient candidate appeared promising, but upon re-imaging, it was found to lie adjacent to a **20-sigma background source**. Its morphology and alignment suggested the burst was a **leakage artefact**.



---

5. Troubleshooting

What should you do if the pipeline crashes or fails to complete an observation?

🧪 *5.1 Common Cause: Input Data Quality*

The **most frequent reason** for pipeline crashes is poor input L1 data. This may happen due to:

- Pre-processing failures (e.g., processing stuck or incomplete)
- Subbands not fully processed
- Severe weather conditions (e.g., thunderstorms) corrupting visibility quality

Recommended action:

- Contact the operations team (Cédric) as soon as possible to request reprocessing or recovery of the L1 data.
- In many cases, this is feasible **within a few days of observation**, while the raw L0 data is still stored and accessible.

If recovery is **not possible** (e.g., confirmed weather-related contamination), then:

- **Option A:** Add the observation to the skip.txt list if it can be re-scheduled.
- **Option B:** Attempt **manual processing** if re-observation is not feasible.

⚙️ *5.2 Other Causes: Pipeline Logic*

Occasionally, crashes arise from the pipeline logic itself. For instance:

- The pipeline previously failed when **calibrator and target observations spanned different calendar months**, which wasn't expected by the code.

In such cases:

- Read the output logs displayed in the virtual terminal window
- Prefect also provides a basic logging interface

Note: Prefect's log history is **only retained for 7 days** unless a premium plan is used.

🔍 *5.3 Best Practices*

To ensure smooth operation:

- The HOST should monitor the **Prefect dashboard** at least **once per week**
- Promptly report anomalies, especially if they involve recent observations

Maintaining this level of vigilance will help reduce downtime and ensure high-quality, reliable data products from the pipeline.

Feel free to reach out or contribute improvements to the pipeline. Together, we can advance NenuFAR's capabilities in exoplanetary science.