RAS Techniques and Instruments

RASTI **5,** 1–27 (2026) https://doi.org/10.1093/rasti/rzag004
Advance Access publication 2026 January 14

# Evaluating the effectiveness of radio frequency interference removal algorithms for single pulse searches

R. S. Hombal [1★] L. Levin,[1] B. W. Stappers [1] M. Droog,[2] A. Karastergiou,[3] D. Lumbaa,[4] M. B. Mickaliger,[1] A. Naidu,[3] K. M. Rajwade,[3] J. Sepulveda,[1] B. Shaw [1] S. Singh[1] and T. Prabu [5]

[1]*Jodrell Bank Centre for Astrophysics, Department of Physics and Astronomy, The University of Manchester, Manchester M13 9PL, UK*
[2]*Covnetics Ltd, Eliot Park Innovation Centre, 4 Barling Way, Nuneaton CV10 7RH, UK*
[3]*Department of Astrophysics, University of Oxford, Denys Wilkinson Building, Keble Road, Oxford OX1 3RH, UK*
[4]*Akkodis UK Limited, New Filton House, Filton, Bristol BS34 7QQ, UK*
[5]*Raman Research Institute, Sadashivanagar, Bengaluru 560080, India*

Accepted 2026 January 12. Received 2025 November 28; in original form 2025 February 28

## ABSTRACT

Radio frequency interference (RFI), the presence of artificial and/or terrestrial signals in astronomical data, poses a great challenge to the search for pulsars and radio transients, such as rotating radio transients (RRATs) and fast radio bursts (FRBs), by obscuring or distorting the signal of interest and resulting in large numbers of erroneous detections. RFI mitigation algorithms aim to remove this interference and improve the chance of detection of transients, but with the growing number of techniques, selecting the most appropriate method for a given survey can be problematic. The choice of method is particularly important in real-time searches planned for next-generation telescopes such as those of the SKAO, where there is no possibility to reprocess the data. In this paper, we explore the algorithm selection problem by injecting pulses into data which simulates several RFI environments. A set of these files is then cleaned using RFI mitigation algorithms and run through a single pulse search pipeline to analyse the recovery of the injected pulses. We examine the recovery of the injected single pulses with an emphasis on a number of cases spanning a range of pulse brightness, width, and dispersion measure. The efficacy and side effects of a few popular RFI excision methods, namely IQRM, SKF, and ZDMF are evaluated.

**Key words:** Algorithms – Radio Frequency Interference – Fast Radio Bursts – Pulsars – RRATs.

## 1 INTRODUCTION

The detection of single short-duration pulses of radio emission has led to the discovery of many astronomical objects such as pulsars and rotating radio transients (A. Hewish et al. 1968; M. A. McLaughlin et al. 2006). Pulsars are highly magnetized, rapidly rotating neutron stars emitting radio waves from their magnetic poles, detected as a pulse of emission if the beam crosses the observer's line of sight. Rotating radio transients (RRATs) are a subclass of neutron stars which emit short but infrequent radio bursts compared to the general pulsar population. J. M. Cordes & M. A. McLaughlin (2003) proposed a method for detecting short-duration radio pulses that is now widely used to explore various regions of the transient parameter space, spanning a range of different time-scales and luminosities. The use of this technique serendipitously led to the discovery of a distinct class of phenomena known as fast radio bursts (FRBs; D. R. Lorimer et al. 2007). FRBs are characterized by short-duration but very bright radio bursts, whose origin remains unknown. Recent discoveries (C. D. Bochenek et al. 2020; CHIME/FRB Collaboration 2020) and the identification of a population of repeaters (e.g. CHIME/FRB Collaboration 2023) suggest connections between FRBs and magnetars might explain at least some of the population. However whether this is the only type of source that can produce FRBs is still unknown. Many models for the origin of FRBs can be found in E. Platts et al. (2019) and more details on FRBs can be found in these reviews E. Petroff, J. W. T. Hessels & D. R. Lorimer (2022), B. Zhang (2023), and D. R. Lorimer, M. A. McLaughlin & M. Bailes (2024). Discoveries like FRBs motivate astronomers to explore all parts of the parameter space of possible radio transients, not only unexplored regions but also to revisit the previously studied regime with improved sensitivity and algorithms. However, with increased radio frequency interference (RFI) in the observable radio spectrum, this becomes a challenge as signals from transients can be obscured. Any artificial, naturally occurring or any non-white-noise-like signal that can negatively impact astronomical observations is often referred to as RFI. The observed signal at the end of the telescope signal chain is the sum of the contributions of the astronomical source, thermal noise contributed by the parts of the instrument itself, backgrounds such as the sky and the ground, and RFI. Typical power densities of astronomical sources are in the range of $-220$ to $-150$ dB W m$^{-2}$

★ E-mail: raghuttamshreepadraj.hombal@manchester.ac.uk

(equivalent to 0.1 mJy − 100 Jy when observed with a frequency bandwidth of 100 MHz). Radio telescopes are therefore required to be extremely sensitive, as even the strongest astronomical signal at a frequency of 300 MHz, $10^5$ Jy, is still approximately $10^6$ times weaker at the telescope than the interference from a typical communication transmitter (see J. M. Ford & K. D. Buch 2014).

In addition to their strength, RFI can exhibit combinations of duty cycle, bandwidth variations, unusual shapes, chirp-like structures, complex modulations, and frequency-dependent variations. Natural phenomena like lightning can affect wide frequency bands, whereas transmitters such as mobile and telecommunication systems, frequency-modulated and Amateur (HAM) radio transmitters, radars, and others occupy designated frequency bands. These narrowband transmitters usually transmit modulated signals to facilitate longer propagation ranges, which may affect adjacent frequency bands to those intended for communication. In addition to this, there can also be signatures of Doppler-shifted RFI, such as that from satellites directly or reflected off aeroplanes.

With the growing number of RFI sources, it is becoming paramount to mitigate the effects of RFI. When searching for radio transients, the relative strength and transient, and/or modulated, nature of RFI could lead to the RFI being reported as real astrophysical sources (false positives) or result in measuring inaccurate source parameters, such as strength and/or width, or complete obscuration of a real astrophysical signal (E. Petroff et al. 2015). Sometimes, they might also imitate spectral lines (Fridman, P. A. & Baan, W. A. 2001).

RFI can be mitigated using several methods, such as frequency rejection and spatial filtering (e.g. Fridman, P. A. & Baan, W. A. 2001). Each method has its own advantages and limitations in its ability to excise RFI. The frequency rejection method excises RFI by applying a mask or notch filter to eliminate frequency channels that are either known a priori or predicted to be contaminated with RFI. The spatial filtering method localizes the RFI emission using a reference antenna pointing off-source. It suppresses the unwanted signal by nulling the synthesized antenna pattern that coincides with incoming RFI (see A. Ardenne, B. Smolders & G. Hampson 2000). These methods are effective, but in this paper, we wanted to consider the methods that are directly applied to the time-frequency space, henceforth called dynamic spectra.

Several RFI Mitigation (RFIM) techniques have been designed to remove RFI signatures from dynamic spectra: Sum-threshold algorithm (A. R. Offringa et al. 2010), Zero-DM filter (R. P. Eatough, E. F. Keane & A. G. Lyne 2009), Spectral Kurtosis filter (G. M. Nita et al. 2007; G. Nita & D. Gary 2010), Inter Quartile Range mitigation (V. Morello, K. M. Rajwade & B. W. Stappers 2022), and Zero-DM Matched filtering (Y. P. Men et al. 2019) are a few popular algorithms that are often used (details in Section 2.2). Recently, deep learning algorithms have also been explored to excise RFI (e.g. A. Vafaei Sadr et al. 2020; H.-F. Wang et al. 2020; Z. Yang et al. 2020; B. R. B. Saliwanchik & A. Slosar 2022).

Generally, the appropriate RFIM algorithms and parameters are chosen based on the performance they achieve when applied to observational data, but to the best of our knowledge, a comparison using a set of controlled parameters and input data has not been explored in the literature. The enormous data rates and the need for rapid follow-up of fast transients, means that telescopes have to run data processing pipelines in real-time, e.g. for the SKAO and its precursors (see e.g. J.-P. Macquart et al. 2010; L. Levin et al. 2017; S. Sanidas et al. 2017), therefore, care must be taken when choosing the most appropriate algorithm as the raw data is no longer available. In addition to this, RFIM algorithms not only have to remove RFI but also have to preserve the intrinsic properties of the single pulse events as much as possible. In this work, we present a method for optimizing the set of RFIM algorithms and their parameters to select an appropriate combination of RFIM algorithms in order to minimize missing candidates and the number of false positives.

Z. Cao et al. (2024) evaluates the effectiveness of algorithms capable of cleaning channelized voltage data (using Median Absolute Deviation) and power spectral density (Spectral Kurtosis), by comparing the resulting signal-to-noise (S/N) of the folded pulse profile of a test pulsar. They compared the algorithms that work on pre-detection data (stage where the data are raw voltages), whereas we are working with post-detection data (a stage where the data are in units of power), a constraint common to many transient searches using total power or Stokes data, where the ability to access and clean raw voltages is lost. V. Morello et al. (2022) compared their proposed algorithm to other existing ones using post-detection observational data from telescopes. Using observational data is useful for comparing algorithms, but it is more difficult to determine whether one has recovered the expected signals and parameters of the astrophysical sources that might be included. One could inject pulses into real data, however, there may be uncertainty because the data may include underlying RFI, baseline variations, and perhaps other instrumental effects that we are not in control of.

We proceed with the assumption that if all the unknown underlying RFI instances mentioned above were absent, the data would be noise-like. In our approach, we therefore conduct tests on dynamic spectra whose contents are completely under our control with minimal uncertainties. This provides a way to directly measure the effectiveness of the algorithms at detecting the known input pulses. This work was in part motivated by that of E. Heerden, A. Karastergiou & S. J. Roberts (2017), who assessed the effect of non-stationary Gaussian noise and RFI on standard pulsar search pipelines and their ability to detect pulsars.

## 2 METHODOLOGY

Our approach to evaluating the effectiveness of RFI mitigation algorithms comprises three stages: generating test vectors, applying the RFIM algorithm to remove the injected RFI, and performing a search to recover the pulses injected. A test vector is a controlled representation of the data that would be presented to a search pipeline and can be used to evaluate the response of the search pipeline. We note that we use detection fraction as our metric here, as we are interested in seeing whether the pulses are recovered during the real-time search rather than investigating the accuracy of the pulse parameters detected. Typically, real-time search pipelines for fast transients will preserve a small amount of complex voltage data at the time of the detected pulse, and these data can be used at a later stage to get the best possible parameters for the detected pulses. We also run the searches over a range of dispersion measures (DMs) that are representative of a typical single pulse search campaign. This is because we want to test whether the presence, or imperfect removal of RFI can affect the detectability of the injected pulses and/or result in them being detected at the incorrect DM, width, time of arrival, and S/N. We note that the imperfect removal of RFI might also lead to a large number of false positives, which results in the search pipelines

**Table 1.** Single pulse parameters used to create test vector filterbanks.

| Parameter | Values |
|---|---|
| Integration time | 60 s |
| Sampling time | 64 μs |
| Number of frequency channels | 4096 |
| Frequency of highest channel | 1670 MHz |
| Channel bandwidth | 78.125 kHz |
| DM (pc cm$^{-3}$) | 10 |
| | 20 |
| | 100 |
| | 150 |
| | 300 |
| | 500 |
| | 1000 |
| | 3000 |
| Pulse widths (ms) | 8 |
| | 40 |
| | 80 |
| | 800 |
| Signal-to-noise | 9.1 |
| | 14.1 |
| | 42.4 |
| | 84.9 |
| | 141.4 |

missing the astrophysical pulses, or becoming non-real-time (see Appendix A).

### 2.1 Test vectors

The process of generating test vectors is summarized in the following steps:

(i) Generating a noise file,
(ii) Injecting pulses of various DMs, widths, and S/N as detailed in Table 1, which results in 160 test vectors,
(iii) Injecting 21 realizations of RFI into each of these test vectors, leading to a total of 3360 test vectors.

This three-step process is explained in detail in the sections below.

*Generating a noise file*

Table 1 contains the telescope parameters and the properties of the pulse used to create these test vectors. The telescope parameters were chosen to be similar to those expected for band 2 of SKA-Mid. For the experiment described in this paper, as discussed above, we assume that the noise in the generated data, prior to insertion of any RFI or astrophysical pulses, is described by a Gaussian distribution.[1] The test vectors are therefore generated using 8-bit unsigned integers drawn from normally distributed noise with fixed mean and standard deviation. These noise files are saved in the `SIGPROC` filterbank format, which is used as a standard data format for transient search pipelines globally (see D. R. Lorimer 2011).

[1]While we model the noise as Gaussian for simplicity, the actual underlying distribution could be different, such as $\chi^2$. This choice does not impact the study's conclusions, as the RFI mitigation algorithms employed for our tests are agnostic to the form of the noise distribution.

*Pulse injection*

A pulsar of desired S/N, with a DM and a Gaussian-shaped pulse of chosen width, is injected into the filterbank noise file using a package called `filtools` (M. J. Keith 2021) for each combination of these pulse parameters. The pulsar is injected periodically with a fixed S/N, determined by the requested S/N of the integrated pulsar signal. The way `filtools` works means that it injects pulses that are identical to each other. Thus, they will differ only due to the contribution from the underlying noise. The pulses are injected with a fixed period of 8 s and without intrachannel DM smearing. The period is chosen so that each test vector contains six individual pulses. We have injected pulses with a constant amplitude across all frequencies, and without any temporal variation, because we wanted to perform a first-order experiment, and including more free parameters such as spectral features, scintillation, and temporal variation would mean additional complexity. We can use the known arrival time of the pulse when determining whether the search process has correctly identified the input pulse.

The ranges of parameters are chosen in such a way that they cover as many cases as possible for a real astrophysical pulse that might be detected in the future. The range is sampled on a log scale to capture as wide a set of parameters as we can in a definite and manageable number of possible values. Three sets of test vectors were generated that contained different realizations of RFI. The first set of test vectors contains white noise with pulses and no RFI is injected (referred to as TVS-1 henceforth). The test vectors of sets 2 and 3 were generated by adding RFI along with the pulse parameters as shown in Table 1.

*RFI injection*

The RFI instances we inject into the filterbank are designed to mimic real-life RFI sources in a simplified but representative way, which are sufficient to test how well an algorithm will excise corresponding real-world RFI. We chose to do this to have control over the nature of RFI in the data. Capturing the complete range of possible RFI manifestations in a finite set of simulated observations is a challenging task. Any instance of RFI in our test vectors can be defined by an amplitude and the number of frequency channels and time samples it spans. Additionally, we use a period and duty cycle for some of these RFI instances to introduce periodicity and make the injection process easier. As most RFI does not have sharp edges in time and/or frequency that would match our time and frequency sampling, the edges of the RFI instances are smoothed by approximately 0.8 MHz in frequency and 640 μs in time. This is done by convolving the RFI instances with a mask defined in a 2D array (P. Virtanen et al. 2020).

As mentioned in Section 1, RFI can manifest in different forms from various sources. Despite the complexity, RFI can generally be categorized into four types: constant broad-band, constant narrowband, periodic broad-band, and periodic narrowband. We note that we consider periodic broad-band RFI to capture the nature of constant broad-band RFI as well, because the algorithm that we use for broad-band RFI mitigation is insensitive to the temporal width of the RFI. All types of RFI we inject are represented in Fig. 1. It shows an example of a piece of a filterbank file as a dynamic spectrum (time against frequency with colour scale corresponding to the strength of the signal) with simulated RFI and an injected pulse.

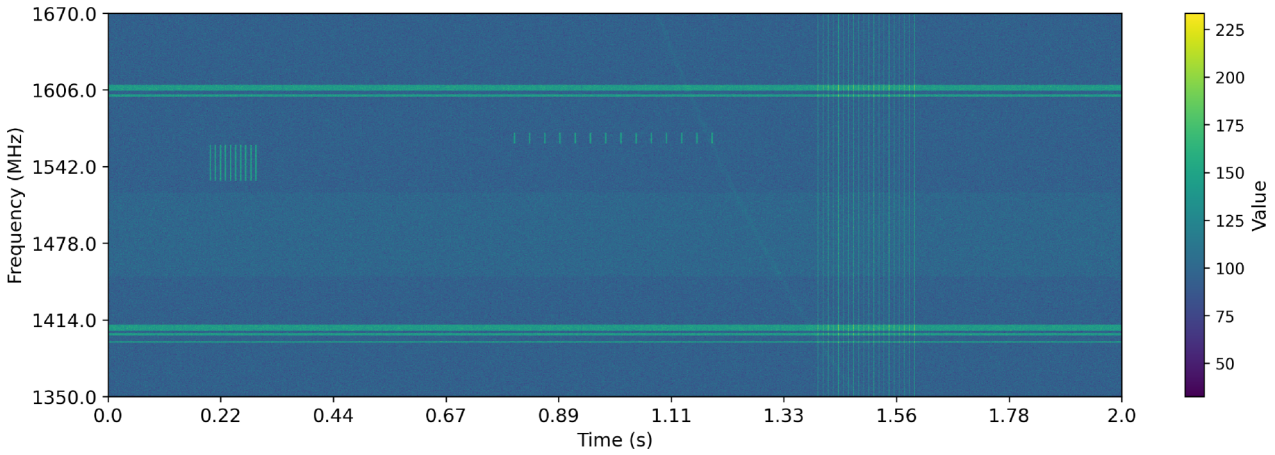


**Figure 1.** Data in a filterbank file that contains a simulated radio transient with a DM of 500 pc cm$^{-2}$ with a S/N of 141 and added RFI. This test vector shows strong RFI ($2\sigma$) of all the types, i.e. narrowband constant, narrowband periodic and broadband periodic RFI. A Gaussian-shaped, dispersed pulse can be seen sweeping from around 1.11 s in time in the highest frequency channel (1670 MHz) up to 1.5 s in the lowest frequency channel (1350 MHz). The structures around 1606 and 1410 MHz mimic the constant narrowband RFI that can be seen from fixed-frequency terrestrial transmitters, which affect the entire observation. The periodic structures affecting the full bandwidth around 1.5 s in time are similar to those of lightning or some other periodic broad-band signal. The periodic structures affecting a narrowband of frequencies from 0.22 to 0.7 s are like those produced by a communication satellite or mobile phones. The data has been downsampled by a factor of 2 in time and frequency to make the pulse stand out.

**Table 2.** RFI environments simulated for the TVS-2 test vectors. The symbols show the strength of RFI, where ✦ represents $2\sigma$ (Strong RFI) and ○ represents $0.5\sigma$ (Weak RFI).

| Type | Narrowband | Broadband | Periodic |
|---|---|---|---|
| 1 | ○ | ○ | ○ |
| 2 | ○ | ○ | ✦ |
| 3 | ○ | ✦ | ○ |
| 4 | ○ | ✦ | ✦ |
| 5 | ✦ | ○ | ○ |
| 6 | ✦ | ○ | ✦ |
| 7 | ✦ | ✦ | ○ |
| 8 | ✦ | ✦ | ✦ |

**Table 3.** The RFI parameters used to simulate the RFI environment used to create TVS-3 test vectors. The percentage of affected samples refers to the fraction of affected frequency channels in the case of narrowband RFI and time samples in the case of broad-band RFI. TVS-3A comprises the narrowband RFI realizations, and TVS-3B the broad-band RFI ones.

| Type of RFI | Per cent of affected samples | Strength |
|---|---|---|
| Narrowband | 25 per cent | $2\sigma$ |
| Narrowband | 25 per cent | $1\sigma$ |
| Narrowband | 25 per cent | $0.5\sigma$ |
| Narrowband | 10 per cent | $2\sigma$ |
| Narrowband | 10 per cent | $1\sigma$ |
| Narrowband | 10 per cent | $0.5\sigma$ |
| Broadband | 25 per cent | $2\sigma$ |
| Broadband | 25 per cent | $1\sigma$ |
| Broadband | 25 per cent | $0.5\sigma$ |
| Broadband | 10 per cent | $2\sigma$ |
| Broadband | 10 per cent | $1\sigma$ |
| Broadband | 10 per cent | $0.5\sigma$ |

The second set of test vectors (referred to as TVS-2 henceforth) contains a combination of these narrowband, broad-band, and periodic RFI. A combination can be made with each type with varied amplitudes, thus creating a combination of eight test vectors for a given DM, S/N, and pulse width as shown in Table 2. When an instance of RFI is injected, it has a uniform strength which corresponds to an increase in the mean by n$\sigma$ per frequency channel per time sample, where n can be different for different types of RFI in a given test vector. A total of 5 per cent of the frequency channels and time samples in TVS-2 are contaminated by RFI.

**Table 4.** Truth table of contents in each set of test vectors

| | Pulses | Periodic broadband | Periodic narrowband | Persistent narrowband |
|---|---|---|---|---|
| TVS-1 | ✓ | × | × | × |
| TVS-2 | ✓ | ✓ | ✓ | ✓ |
| TVS-3A | ✓ | ✓ | × | × |
| TVS-3B | ✓ | × | ✓ | ✓ |

The third set of test vectors has two subsets, referred to as TVS-3A and TVS-3B henceforth. TVS-3A contains only evenly spaced narrow-band periodic RFI, and TVS-3B contains only broad-band RFI, contaminating all frequency channels of a few time samples randomly repeated over time. The maximum number of affected frequency channels and time samples was restricted to 25 per cent of the total number of frequency channels or time samples. Note that each of the test vectors in a subset contains the same kind of RFI, for instance, in TVS-3B, every test vector contains RFI affecting the same group of frequency channels (see Figs C1 and C2). Table 3 shows all the combinations of values of RFI parameters used to generate test vector sets TVS-3A and TVS-3B containing RFI.

Table 4 summarizes all the test vector sets and their contents. A Python-based script, `Generator.py`.[2] was used to automate the tasks mentioned in the section above.

[2] https://gitlab.com/ska-telescope/pss/ska-pss-test-vector-generator/. The Gaussian noise generator uses `123123` as the seed value to generate white noise to inject pulses and RFI into.

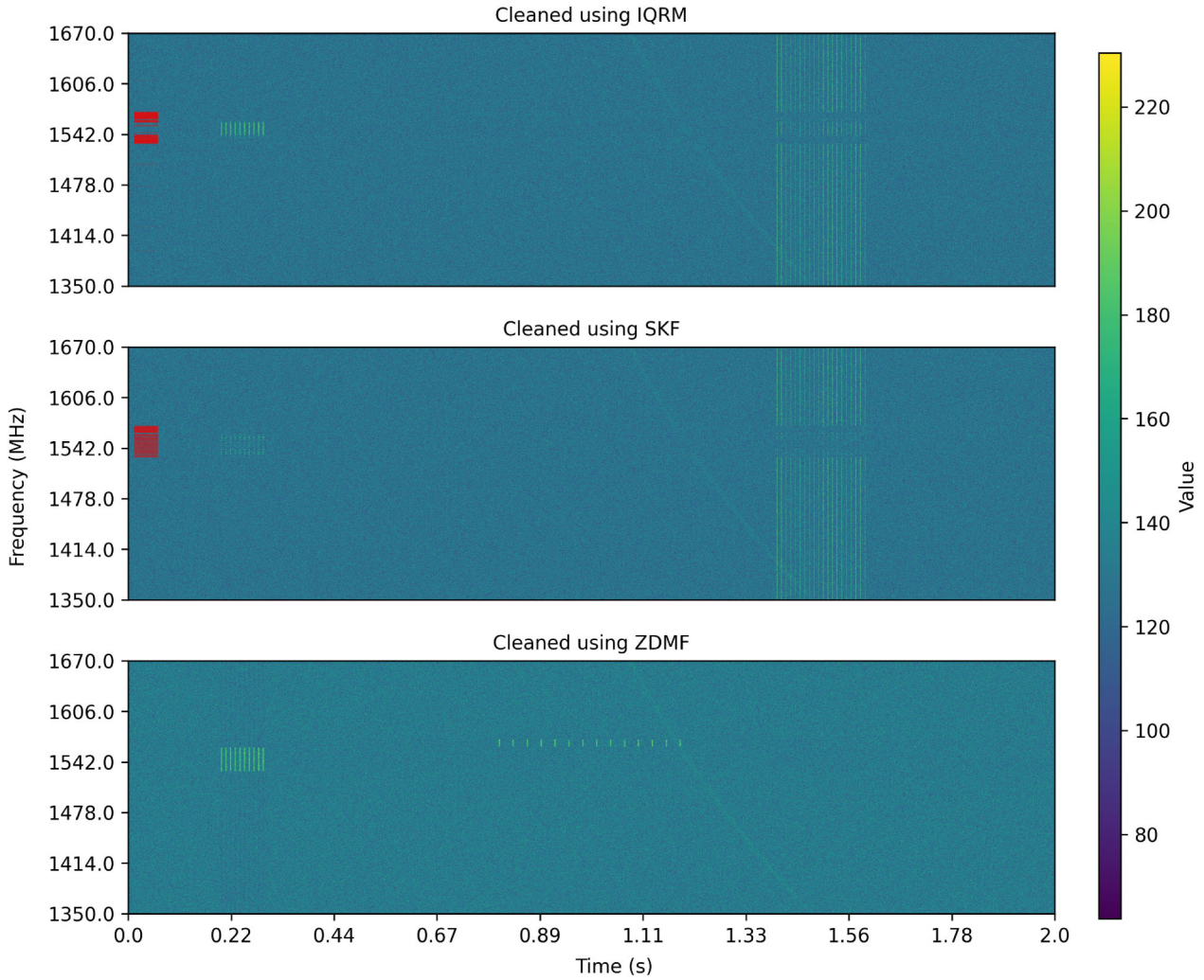


**Figure 2.** Demonstration of the chosen RFI removal algorithms, run individually, on the data shown in Fig. 1. The red lines at the left of the top two panels indicate the channels that are flagged as RFI-affected by the respective algorithms. `filtool` (used for SKF and ZDMF) corrects for the bandshape of the chunk of data, whereas `iqrm-apollo` does not. To present the data comparable to each other, the topmost plot in the figure showing data cleaned by IQRM is, therefore also shown after correcting for its bandshape. Since ZDMF acts on the time samples, it does not mask any data but changes every time sample, and so only removes broad-band RFI. One can also notice the patterns extending the residual RFI across all the frequency channels.

### 2.2 RFIM algorithms

Software implementations of the selected algorithms are installed on the machine that runs the tests. These are capable of reading a filterbank file, cleaning the RFI, and writing out a cleaned filterbank file, which is then used to search for the injected pulses. The algorithms being examined are:

(i) Spectral Kurtosis Filter (SKF) (see G. Nita & D. Gary 2010)
(ii) Inter Quartile Range Mitigation (IQRM) (see V. Morello et al. 2022)
(iii) Zero DM Matched Filtering (ZDMF) (see Y. P. Men et al. 2019)

We refer the readers to the papers listed above for the details of the respective algorithms. SKF and ZDMF are implemented in `filtool`, which is a part of the `PulsarX`[3] package (see Y. Men et al. 2023) and IQRM is implemented by `iqrm-apollo`[4]. The efficacy of the individual RFIM algorithm is shown in Fig. 2. The flagged data is replaced with noise samples drawn from a normal distribution. In the case of `filtool`, the mean and standard deviation of the entire data (both flagged and unflagged) is used to generate these noise samples.[5] However, in `iqrm-apollo`, the median and standard deviation of the data block are used to generate the distribution of samples to replace the flagged data. `filtool` and `iqrm-apollo` are configured to read and clean 2 s (i.e. 31 250 time samples) duration subsets of data, and these subsets are used to calculate the statistics and spectral moments (mean, standard deviation, skewness, and kurtosis). In a real-time search pipeline, we might not know the nature of

[3] https://github.com/ypmen/PulsarX; not same as `filtools`

[4] https://gitlab.com/kmrajwade/iqrm_apollo

[5] In `filtool`, the data is normalized to have 0 mean and 1 standard deviation, thus using the same mean and standard deviation to replace the data. These values have to be scaled up to values appropriate for the 8-bit data type used in the filterbank file format before exporting.

the RFI, but in our case, we have prior information about the RFI. This could create a bias in optimizing parameters of RFI excision algorithms. To avoid this, we chose to use the default subsets time from `filtool (PulsarX)` to clean the data. The side effects of this can be seen in Fig. 2, where some residual RFI is visible, which may not be expected when cleaned using a perfect RFI excision algorithm. However, this reproduces a real observation scenario, where some RFI usually leaks through the filtering stage even after RFI removal.

An important parameter in the SKF and IQRM algorithms is the threshold, which decides whether a slice of data is affected by RFI or not. It is set to $3\sigma$ as this is a commonly used value for this parameter. In the case of SKF, $\sigma$ is the interquartile range of the skewness and kurtosis values calculated for every frequency channel, whereas for IQRM it is the ratio of the interquartile range and inverse cumulative distribution at both quartiles. In the case of IQRM, another parameter needs to be set for optimal RFI rejection. This is the channel radius which is set to approximately 10 per cent of the total number of frequency channels (410 in our case). ZDMF has no configurable parameters and instead performs the same action on any data provided to it. Due to specializations of these algorithms to excise a particular kind of RFI, we use them in groups. However, the performance of each of the mentioned algorithms is heavily dependent on the input provided to it, hence, we do not use every possible permutation of these algorithms. We classify SKF and IQRM to be good at cleaning narrowband periodic RFI and cannot excise RFI that alters noise baselines, such as broad-band interference, as described by V. Morello et al. (2022). This is because the injected broad-band RFI does not cause a significant deviation in the per-channel statistics over the time ranges used for calculating the statistics. Therefore, a combination of IQRM–ZDMF or SKF–ZDMF, in theory, should be able to clean most RFI, and this is why TVS-2 was cleaned once with each combination. In addition to this, we also used IQRM alone to clean TVS-2 to get an idea of how important broad-band RFI removal techniques are.

This paper focuses only on a limited number of mitigation algorithms. The chosen algorithms are widely used in the fast transient searching community and a systematic comparison of their efficacies will inform their use in future surveys. However, our approach can evaluate any algorithm that can clean dynamic spectra.

### 2.3 Single pulse search

A Python-based testing framework, `ProTest`[6] is used to carry out the task of iteratively running the RFI cleaned test vectors through the search pipeline using the `ska-pss-cheetah` pipeline framework and verifying the candidate output. As previously described, since we are interested in the influence of RFI mitigation approaches on the detectability of single pulses in an untargeted search, we emulate a real single pulse search. Therefore, the post-RFI excision data is dedispersed at multiple DM trials before being searched for single pulses (shown in Table 5).

Dedispersion is a process of correcting for the delays caused by the interaction of electromagnetic waves with electrons in the interstellar medium (ISM) along the line of sight from the telescope to the source of the pulse. This is done by correcting for these delays in each frequency channel and then adding all the frequency channels to increase the S/N of the pulse. These operations are performed by an AVX-512-based tree dedispersion algorithm in an implementation called Klotski[7] (see A. Naidu et al. 2024). Emulating real single pulse search process also allows us to see if incomplete RFI removal, for example, results in too many false positive detections[8] and/or recovery of pulses with incorrect pulse parameters.

**Table 5.** Dedispersion plan used for dedispersing time-frequency data.

| Start DM | End DM | DM step |
|---|---|---|
| 0.0 | 100.0 | 0.1 |
| 100.0 | 300.0 | 0.2 |
| 300.0 | 700.0 | 0.4 |
| 700.0 | 1500.0 | 0.8 |
| 1500.0 | 3100.0 | 1.6 |

**Table 6.** The parameters of sifting and clustering used in the search.

| Module | Parameter | Values |
|---|---|---|
| Clustering | Time tolerance | 100 ms |
| | Pulse width tolerance | 100 ms |
| | DM tolerance | 5 pc cm$^{-3}$ |
| Sifting | S/N threshold | $< 6$ |
| | Pulse width threshold | $> 1$ s |
| | DM threshold | $< 5$ pc cm$^{-3}$ |

The dedispersed time series are formed and convolved with a set of box-car filters of various widths. When the S/N of the convolved product crosses a defined threshold, it is considered a detection. The S/N of a pulse is the integrated power of the pulse, normalized by the square root of the width of the pulse divided by the standard deviation of the off-pulse data. When a detection occurs, the corresponding timestamp of the event, the value of the DM used to dedisperse the data, the width of the box-car and the S/N are recorded. The range of widths of box-car filters begins with 2 samples and increases in powers of 2 up to 8192 samples and then a final filter of 15 000 samples (i.e. 960 ms) to be a near match to the maximum input pulse width.

The detections are passed through a clustering process to reduce the number of detections corresponding to the same event. We use the friends-of-friends algorithm explained in J. S. Deneva et al. (2009). In this algorithm, detections are grouped based on the proximity of two detections defined by a range of arrival times, DMs, and pulse widths called tolerances. The clustering parameters are listed in Table 6, and are set carefully to cluster single pulses of the same events.

The sifting algorithm employed here is a simple method of sifting false detections. A candidate is removed from the list of detections if either DM or S/N are below a defined threshold, or the width is above a threshold. Table 6 shows the default values of parameters used for sifting. A list of sifted detections is then obtained with corresponding metadata.

### 2.4 Recovery of pulse

The pulse arrival time, DM, and width from the list of detections are cross-checked with a list of the expected parameters from the injected pulses which are known for each test vector. As the input

[6] https://gitlab.com/ska-telescope/pss/ska-pss-protest/-/releases/4.1.2

[7] https://gitlab.com/ska-telescope/pss/ska-pss-cheetah/-/tags/0.4.0

[8] See Appendix A for a description of the detrimental effects of too many candidates on the single pulse search pipeline.

parameters describing the simulated pulses may not be recovered exactly, a range of tolerances is defined for each parameter based on the known value. These tolerances are used to classify if the detected pulse was the one which was originally injected and not a false positive.

The tolerance on the arrival time of the pulse is taken to be the $1\sigma$ width of the input Gaussian pulse. The tolerance on the width is taken to be one box-car width narrower/wider than the expected pulse width. For example, for a 10-bin-wide input pulse, 8 and 16 bins were also considered to be a detection.

The DM tolerance ($\Delta$DM) is defined as,

$$\Delta \mathrm{DM} = \frac{n\mathrm{w}}{\mathcal{D}\left(f_{\mathrm{low}}^{-2} - f_{\mathrm{high}}^{-2}\right)}, \tag{1}$$

where w is the true width of the pulse and $\mathcal{D}$ is the dispersion constant ($\approx 4.15$ GHz$^2$ cm$^3$ pc$^{-1}$ ms). The value of $n$ sets the tolerance to a level with an acceptable recovered S/N when dedispersed at a wrong DM and here $n = 2$ allows 85 per cent of S/N recovery (see J. M. Cordes & M. A. McLaughlin 2003).

Results are summarized in the section below and the fractions of recovery for every test vector are available in Appendices B and C. It is to be noted that the width of the injected pulse heavily influences the tolerances on arrival time and the DM, hence the tolerances for the case of the widest pulse in the test vector set could lead to defining tolerances so large that it is likely for a false positive to be considered as a true detection. This is the reason why, in some extreme cases, we do not include detections from the widest pulses in the analysis.

We also note that now and again there might be an occasional pulse missed in our analysis, see e.g. C10, where we would expect to get six out of six pulses. After inspection, we conclude that this happens when the pipeline returns a time of arrival and/or width that is outside of the aforementioned tolerances. This likely happens as the best S/N occurs at a wider width because of the noise properties and the pulse shape. This reflects the challenges of putting in place methods to check the performance of these pipelines without manual inspection. Such methods are necessary when trying to test across thousands of scenarios that might arise in a real search like those presented here. This loss of an occasional pulse does not affect our interpretation of the algorithms under test, as the overall trends are what we are most interested in. We also note that the pulses would not remain undiscovered in a real search, but the discovered width and possibly DM would differ slightly from the true value. The results for the no-RFI case presented in Appendices B and C can be used for comparison with the cases where RFI was included if there is any doubt.

## 3 RESULTS

### 3.1 RFI scenario 1

The use of RFIM has advantages and disadvantages. Although RFI mitigation algorithms may be effective at removing interference, they may affect data quality, leading to unintended distortion or weakening of the signal of interest. Many statistical parameters may also change, there might be an observed reduction in S/N of the pulse and sometimes cleaning techniques remove parts or the whole of the pulse if it is bright enough. In addition to this, it costs additional computing power and time. However, in the presence of RFI, the observed data cannot be used for most science cases because RFI might interfere with the process of extracting astrophysically useful and accurate information from it.

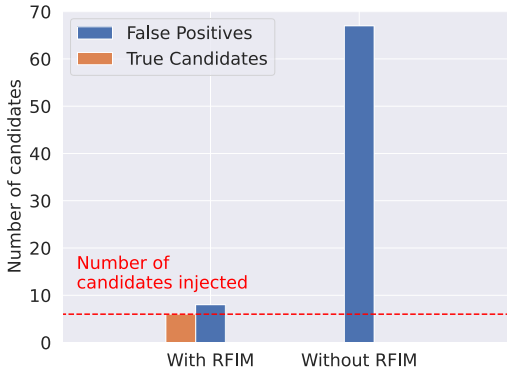


**Figure 3.** Histogram of the number of true and false positive candidates detected with and without RFIM in a single pulse search. A test vector containing six pulses with a DM of 300 pc cm$^{-3}$, a width of 40 ms, a single-pulse S/N of 42.4, and a combination of all three types of RFI ($2\sigma$) is searched for single pulses. Once without RFI cleaning and once after cleaning using IQRM and ZDMF.

As an example to further demonstrate the need to use RFIM, a test vector containing pulses of width 40 ms and a DM of 300 pc cm$^{-3}$ and added RFI was searched without cleaning by any RFI excision method. Fig. 3 shows the number of true and false positive detections. The same test vector was cleaned using IQRM followed by the ZDMF algorithm, before searching for single pulses and we could recover all six pulses. From Fig. 3, we can infer that, although there are a significant number of false positives, we could still detect injected pulses, contrary to the case of not using RFIM, where none of the pulses are detected. Therefore, as expected, it becomes clear that using RFIM during the search for transients is essential.

We used a combination of RFIM algorithms to clean the contaminated test vectors from TVS-2 as well as a complementary set with no RFI injected for comparison. The test vectors were cleaned using IQRM, IQRM with ZDMF, and SKF with ZDMF. In each case the resultant filterbank files were searched for single pulses and the results were analysed. Ideally, we should recover all the pulses for all the DMs, S/Ns, and pulses of all widths. Fig. 4 shows the fraction of recovered pulses using all three RFI mitigation strategies on test vectors from TVS-2 with strong and weak RFI of all types (Type 1 and Type 8 from Table 2). These are representative numbers from just 2 realizations of RFI, more results on the entire set, including those with no RFI, can be found in Appendix B.[9] In the first panel of Fig. 4, the fraction of recovered pulses is zero because the search pipeline timed out in all the cases. This is due to residual RFI creating so many candidates that the search pipeline stalls and is eventually killed by `ProTest`, leading to no pulses being recovered (see Appendix A for more details).

Our simulations included test vectors with widths of 800 ms, and the results for those widths are presented in the Appendix B.

[9]We note that for a handful of test vectors with no RFI injected (both in Appendices B and C), the pipeline returned zero candidates when the S/N was very high. We identified a corner case in our pipeline where this is because a lower DM reports a higher S/N. In these cases, we have instead reported the results after manual inspection. We have also confirmed that this does not affect any of the other results.

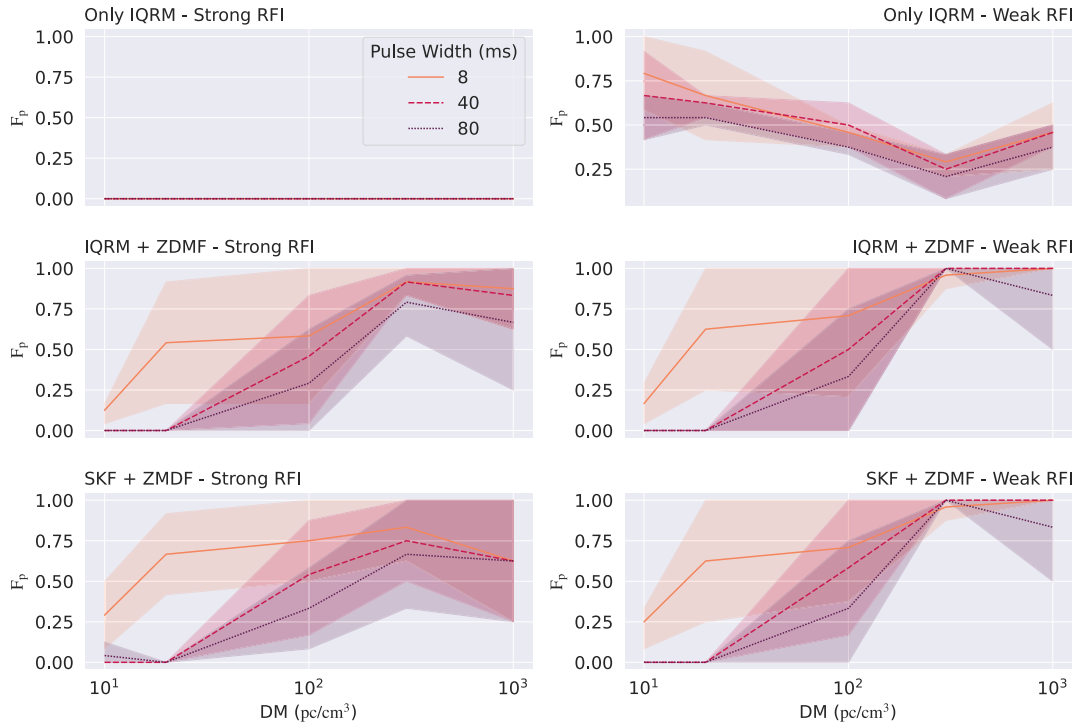


**Figure 4.** Detection rates for different combinations of RFIM applied to test vectors containing realistic RFI as a function of DM are shown. The fractions of recovered pulses ($F_p$) are plotted against DM for multiple widths. The combinations of RFIM used are – IQRM, IQRM + ZDMF and SKF + ZDMF. The plots on the left side are obtained from results when test vectors from TVS-2 containing RFI of type 8 are cleaned with respective RFIM, and the ones on the right side contain RFI of type 1 (see Table 2). The shaded region indicates the range of $F_p$ for different S/N (ranging from 14.1, corresponding to the lower edge, to 141.4, corresponding to the upper edge) at a given DM for a given pulse width. In the first panel, where the test vectors containing RFI of type 8 are cleaned using IQRM, no pulses are recovered, but the fraction of recovery increases when the RFI is weaker and/or cleaned by ZDMF as well.

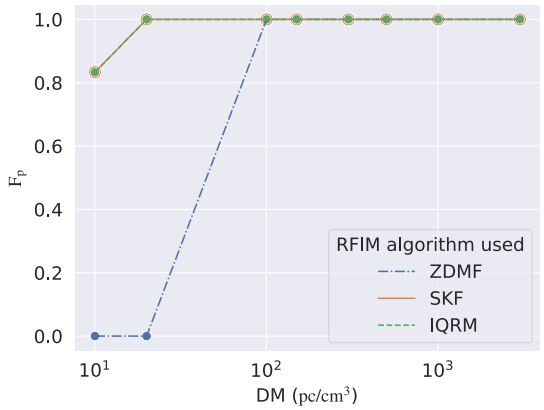


**Figure 5.** The impact of each algorithm on the fraction of recovered pulses ($F_p$) across a range of DMs for test vectors without added RFI. The test vectors contain pulses with a width of 40, ms and a (S/N) of 42.4. Note that the $F_p$ curves for SKF and IQRM overlap in the figure.

However, as briefly mentioned above, in the presence of incompletely removed RFI, the wide box car needed to recover these wide pulses can result in low-S/N false positive detections because of the larger tolerances on DM, width, and arrival time. We therefore do not include it in further analysis. However, we do present these results in the appendices to highlight the challenges that single pulse searches in the time domain will have in detecting such wide pulses in the presence of RFI. This also suggests that other approaches to RFI removal might be needed to enable their detection.

In the presence of all three kinds of RFI, we fail to recover a large portion of the injected pulses from the single pulse search when IQRM alone is used to clean the RFI. This is especially true in the presence of strong levels of RFI of all types ($2\sigma$), where we recover the fewest pulses. The fraction of recovered pulses increases when ZDMF is used in sequence with IQRM. This shows that IQRM is not good at cleaning broad-band RFI and should always be used in combination with an RFI excision method that is good at removing broad-band RFI (as explained by V. Morello et al. 2022). Another observation is that when ZDMF is used in sequence with SKF or IQRM, either combination of RFIM algorithms works well, and the total number of pulses recovered increases when compared to the case of not using ZDMF, but the fraction of recovered pulses was not 1. We could however, recover more pulses at higher DMs when compared to the fraction recovered at lower DMs. It appeared as if there was a function governing the process due to which pulses were not detected. To gain a better understanding of what could be causing this, we decided to use RFI-free test vectors containing pulses and apply the different mitigation methods individually.

### 3.2 RFI scenario 2

To examine the cause of the failure to detect some pulses, each of the RFI mitigation algorithms was applied to test vector data free of RFI contamination (TVS-1). Fig. 5 shows an example of test vectors containing pulses of width 40 ms and a S/N of ≈

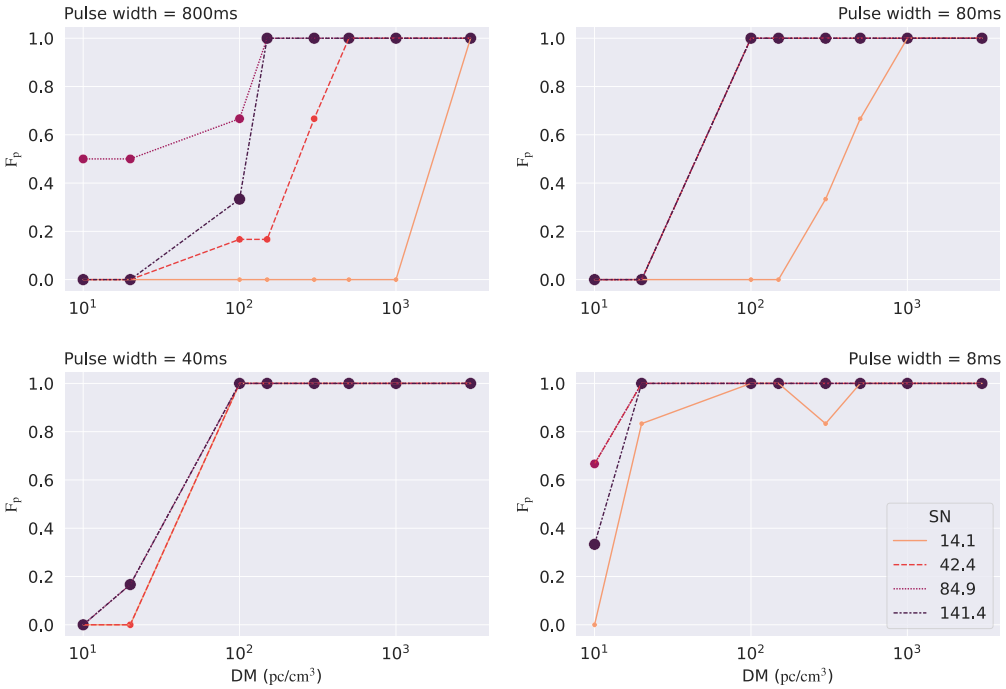


**Figure 6.** The effect of ZDMF filtering on RFI-free test vectors containing pulses of various pulse widths and S/N is shown by plotting the fraction of recovery of pulses ($F_p$) as a function of DM. At low DM, it becomes more difficult to recover wider pulses, however, this difficulty diminishes as the S/N increases. We have included the 800 ms widths here as no RFI was injected (see discussion in Section 3.1).

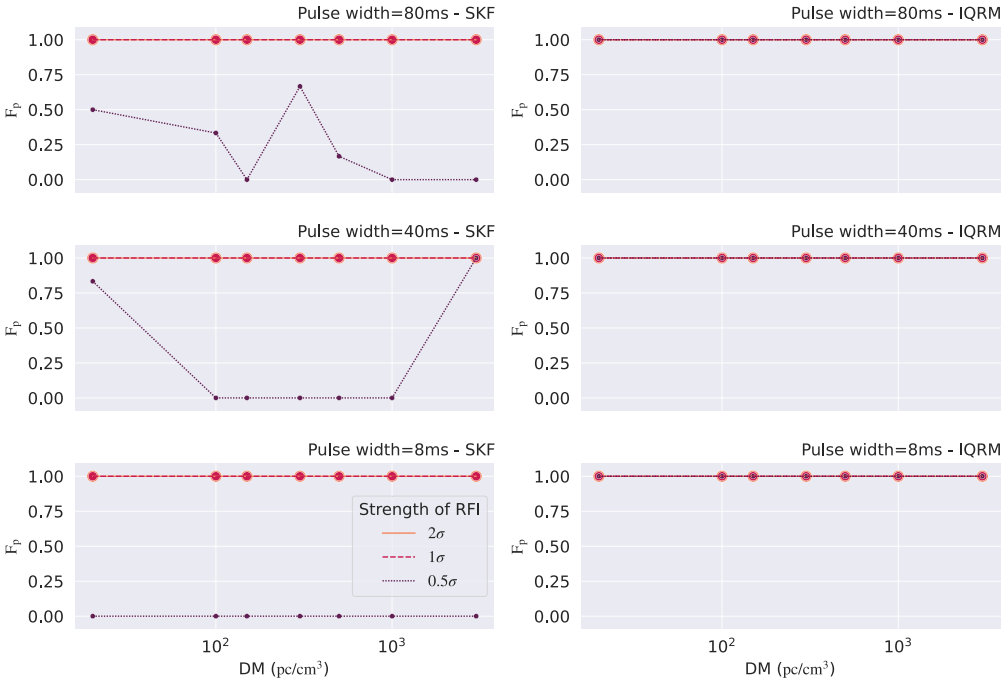


**Figure 7.** Recovered fractions of single pulses ($F_p$) as a function of DM for the case of S/N $\approx$ 85 for multiple widths when IQRM and SKF algorithms are used for various strengths of narrowband periodic RFI. Note that the red and orange dashes follow each other in all cases and for the plots showing $F_p$ for IQRM, all three lines follow each other and the value is unity at every DM.

42, where the fraction of pulses recovered from the single pulse search are plotted as a function of the DM of the pulses (the full results can be found in the left most panel of the figures in Appendices B and C). We observed that, in our sample space, the single pulse search did not recover pulses with a DM below 100 pc cm$^{-3}$ in all scenarios that included ZDMF filtering. With SKF and IQRM on their own, the fraction of recovered pulses is almost complete for all the DM trials when either of the algorithms is used to clean the RFI-free test vectors (see Fig. 5).

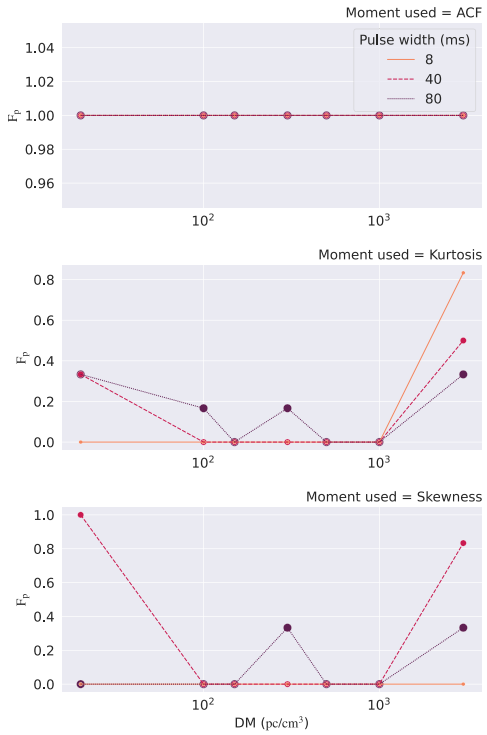


**Figure 8.** The effect on the fraction of recovered pulse ($F_p$) at a range of DMs for using various moments for RFI excision by IQRM is shown in the figure. The test vectors used contain periodic narrowband RFI of strength $0.5\sigma$ and pulses with S/N $\approx$ 42. Note that in the plot showing $F_p$ versus DM when the autocorrelation factor (ACF1) spectral metric is used, all three lines (orange, red, and purple line) follow each other for all the values of DM.

To investigate the lack of low-DM detections in the ZDMF case, we looked at the results for all the test vectors cleaned by ZDMF as shown in Fig. 6. We see a clear loss of pulses at low-DM when ZDMF is used. It is to be noted that we did not generate test vectors that contain pulses with DMs between 20 and 100 pc cm$^{-3}$ as our intention was not to test ZDMF alone. We therefore do not know exactly where this transition might happen, and as we can see in Fig. 6, there is a dependence on pulse width and S/N. This loss of low-DM pulses was already identified as an issue by R. P. Eatough et al. (2009) when zero-DM filtering (ZDF) was first introduced, and is due to the influence of the dispersed pulse on the estimation of the mean. The ZDMF algorithm is an extended version of the ZDF algorithm, which subtracts only the corresponding contribution from each channel at a given time sample (see Y. P. Men et al. 2019). As described by R. P. Eatough et al. (2009), the smaller the fractional bandwidth (the ratio of the bandwidth to the central frequency), the greater the prominence of the effect. The fractional bandwidth that we are using for this paper is similar to that for band-2 of SKA-Mid, and thus our results are representative of what can be expected in that band.

In the first panel of Fig. 6, the number of recovered pulses at lower DMs is greater for pulses with S/N of 85 than for pulses with S/N of 140. Upon investigating individual pulses with different DMs, we conclude that the detections reported by the search pipeline for an input S/N of 85 were due to the residual of the pulse after cleaning using the ZDMF algorithm. The candidates have S/N of approximately 6 or 7, and their corresponding pulse residuals had barely managed to cross the threshold to be detected by the search pipeline.

### 3.3 RFI scenario 3

*Excision of narrowband periodic RFI*

A set of test vectors containing narrowband periodic RFI (from TVS-3B, see Table 3) was cleaned using both SKF and IQRM (see Section 2.2), and the results can be found in Appendix C. Fig. 7 shows the effect of DM on the recovery of pulses with S/N $\approx$ 85 and a range of widths. In both cases, when the data is cleaned by either IQRM or SKF, there is a complete recovery of single pulse events in an environment containing RFI of strength $\geq 1\sigma$, where $\sigma$ is the RMS of the noise. However, when the strength of the injected RFI is fairly low (here $0.5\sigma$ per frequency channel per time sample, which is low in our test vector sets, because, when individual time samples or frequency channels are considered, the RFI is indistinguishable from the noise), IQRM can mitigate the RFI well enough that all the pulses are recovered, but SKF leaves behind residual RFI due to which some pulses cannot be detected during the single pulse search. This can be seen in the left panels of Fig. 7, where we can get zero candidates in the two ways described in Appendix A. We note that in the middle-left panel of Fig. 7 there are some pulses detected at the lowest and highest DMs. In the former case this corresponds to the pulses being detected, although at an incorrect DM, but which is still within the tolerances, while the latter case corresponds to low-S/N residual RFI appearing within the tolerances. It is to be noted that the spectral metric used by IQRM is the ACF1, but SKF uses skewness and kurtosis, which are statistical measures for the shape of a probability distribution (here, the data in a filterbank file), as its spectral moments to flag RFI-affected channels. When IQRM is run with the chosen spectral moment set to be kurtosis and skewness, the fraction of pulses recovered is no longer complete, as shown in Fig. 8. This is consistent with the SKF results and shows that the spectral moment used for the mitigation process plays a vital role in mitigating low levels of RFI, as also discussed in V. Morello et al. (2022).

*Excision of broad-band RFI*

As described in Section 2.2, broad-band interference alters the noise baseline, and the SKF and IQRM algorithms are specialized to remain unaffected by such baseline variations. All the test vectors containing broad-band RFI (TVS-3A, see Table 3) were therefore cleaned only using ZDMF, and the results can be found in Appendix C. Fig. 9 demonstrates the response of ZDMF by showing the fraction of recovered pulses as a function of DM for all widths in an environment containing RFI of strength equal to $2\sigma$ and affecting 10 per cent of time samples. The trends in the plot are similar to those demonstrated in the earlier section under no RFI conditions (see Fig. 6). The fraction of recovery increases as the pulse width decreases, and increases as the DM increases, but the RFI environment does not show any trend. This is because the operations performed by ZDMF rely on the undispersed nature of RFI rather than its strength. Another interesting observation is that, in the case of high DM but smaller S/N, the recovery of pulses is inversely proportional to the width of the injected pulse. For a given fractional bandwidth, the broader the pulse, the greater will be the degradation in S/N thus suppressing it enough not to cross the detection thresholds (mentioned earlier in Section 3.2).

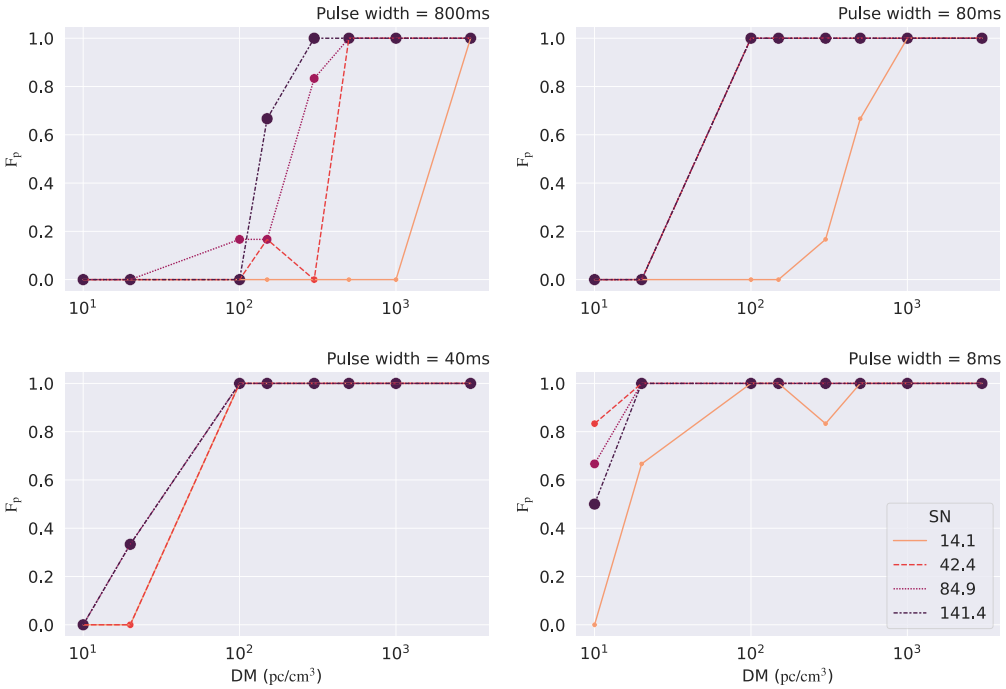


**Figure 9.** The fraction of recovery of single pulses ($F_p$) as a function of DM for multiple widths in an RFI environment which has a $2\sigma$ level of broad-band RFI affecting 10 per cent of total time samples observed. The result is similar to Fig. 6 which indicates that the ZDMF algorithm is unaffected by the amount of broad-band RFI in the data.

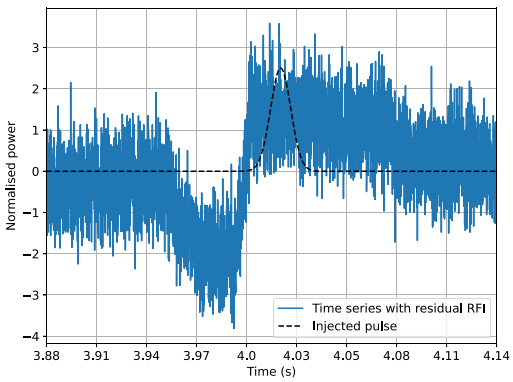


**Figure 10.** Dedispered time series of a test vector where a pulse with a width of 40 ms, S/N of 14.1, and DM 10 pc cm$^{-3}$ has been injected. Narrowband RFI of $0.5\sigma$ affecting 10 per cent of frequency channels (see Table 3) was injected and cleaned using SKF. The dashed black line is where the pulse was expected to be but is obscured by the residual RFI.

## 4 DISCUSSION

As discussed above, our aim was to generate a range of RFI scenarios which included simulated RFI with properties that were drawn from the underlying properties of real-world RFI, namely broad-band, narrowband, and then periodic versions of both of these, to test some common RFI mitigation algorithms. The highly variable and varied nature of RFI means that even though we considered a large number of different scenarios, it isn't possible to capture all possibilities here. We note that future work should consider the inclusion of more random variability of the amplitude of the RFI when it is injected, including an increase in the noise for example. The nature of the periodic RFI we introduced, especially in the narrowband case, was challenging, but when it was strong, both SKF and IQRM were able to deal with it well, but when weaker, SKF struggled more. Future work should also include a wider range of timescales for the periodic RFI, and compare the performance with the timescales used in calculations of the relevant statistics for each tested RFIM algorithm. Although not directly related to the RFI algorithms under consideration here, the issue of the large number of false-positives that can be produced in a single pulse search if the RFI is particularly bad and thus cannot be adequately removed, needs consideration. This is true whether you are doing an online or offline search, as it can result in your search pipeline stalling. In the real-time search, it may result in you missing real signals, and/or not being able to trigger on them.

Some cases with 0 detections were investigated to determine the cause, especially those cleaned by SKF (see Fig. C14). The test vectors were searched over a smaller number of DM trials, and the output candidate file from the search pipeline was manually inspected. Fig. 10 shows an example of a dedispersed time series of a test vector containing a pulse, dispersed with a DM of 10 pc cm$^{-3}$, a width of 40 ms, and a S/N of 14.1, along with narrowband RFI of $0.5\sigma$ affecting 10 per cent of frequency channels (refer Table 3). The plot also contains a dashed line, which is where the pulse would be expected to be found. Instead, the residual RFI has led to a detection of a false positive. As the parameters of this false positive lie outside of the tolerances for the real pulse, it is rightly not counted. This is evidence that residual RFI can corrupt pulses, and it can lead to detecting candidates with incorrect pulse parameters, as stated in the introduction.

## 5 CONCLUSIONS

We present a method to evaluate the effectiveness of an RFI removal technique by defining a number of test cases that one

wants to test for any RFIM algorithm capable of cleaning dynamic spectra. This method is demonstrated using combinations of three algorithms: IQRM, SKF, and ZDMF. Testing of the algorithm independently confirms that a single algorithm is insufficient. We find that across the range of strengths of RFI investigated here, the combination of IQRM and ZDMF works best when the ACF1 spectral metric is used for IQRM. As expected, the use of ZDMF does have a negative impact on the recovery of low-DM pulses for the fractional bandwidth used here. In the future, the investigation of other robust RFIM techniques (including spatial filtering, frequency rejection, and those which use deep learning) using the proposed method for radio transient search is strongly encouraged. Studying the efficacy of algorithms for wider and/or finer sampling ranges for all the parameters that were assumed to be constant or were restricted (such as spectral index and temporal features of pulses, limited realizations of RFI instances) is also encouraged.

## ACKNOWLEDGEMENTS

The authors acknowledge funding from the United Kingdom's Research and Innovation (UKRI) Science and Technology Facilities Council (STFC), project reference [ST/Z510439/1, ST/Z510440/1, and ST/W001950/1]. They would like to thank Yunpeng Men for discussions regarding the ZDMF algorithm, Michael Keith for supporting RFI injection with `filtools`, Sergio Belmonte Díaz for the discussions on the S/N for single pulses and Aristeidis Noutsos for their comments. The authors would also like to thank the reviewers of the paper, whose comments significantly improved the manuscript.

## CONFLICT OF INTEREST

The authors declare no conflict of interest.

## DATA AVAILABILITY

Steps to reproduce the test vectors and the results have been described in the paper along with all the software used.

## REFERENCES

van Ardenne A., Smolders B., Hampson G., 2000, in Butcher H. R., ed., Proc. SPIE Conf. Ser. Vol. 4015, Radio Telescopes. SPIE, Bellingham, p. 420
Bochenek C. D., Ravi V., Belov K. V., Hallinan G., Kocz J., Kulkarni S. R., McKenna D. L., 2020, Nature, 587, 59
CHIME/FRB Collaboration, 2020, Nature, 587, 54
CHIME/FRB Collaboration, 2023, ApJ, 947, 83
Cao Z. et al., 2024, RAS Techn. Instrum., 3, 535
Cordes J. M., McLaughlin M. A., 2003, ApJ, 596, 1142
Deneva J. S. et al., 2009, ApJ, 703, 2259
Eatough R. P., Keane E. F., Lyne A. G., 2009, MNRAS, 395, 410
Ford J. M., Buch K. D., 2014, in IEEE Geoscience and Remote Sensing Symposium. IEEE, p. 231
Fridman P. A., Baan W. A., 2001, A&A, 378, 327
van Heerden E., Karastergiou A., Roberts S. J., 2017, MNRAS, 467, 1661
Hewish A., Bell S. J., Pilkington J. D. H., Scott P. F., Collins R. A., 1968, Nature, 217, 709
Keith M. J., 2021, Filtools, Zenodo, available at: https://doi.org/10.5281/zenodo.5751985
Levin L. et al., 2017, Proc. IAU Symp. 385, Pulsar Searches with the SKA, 13, Cambridge Univ. Press, Cambridge, p. 171
Lorimer D. R., 2011, Astrophysics Source Code Library, record ascl:1107.016
Lorimer D. R., Bailes M., McLaughlin M. A., Narkevic D. J., Crawford F., 2007, Science, 318, 777
Lorimer D. R., McLaughlin M. A., Bailes M., 2024, Ap&SS, 369, 59
Macquart J.-P. et al., 2010, Publ. Astron. Soc. Aust., 27, 272
McLaughlin M. A. et al., 2006, Nature, 439, 817
Men Y. P. et al., 2019, MNRAS, 488, 3957
Men Y., Barr E., Clark C. J., Carli E., Desvignes G., 2023, A&A, 679, A20
Morello V., Rajwade K. M., Stappers B. W., 2022, MNRAS, 510, 1393
Naidu A. et al., 2024, 4th URSI AT-RASC, Gran Canaria
Nita G., Gary D., 2010, MNRAS, 406, L60
Nita G. M., Gary D. E., Liu Z., Hurford G. J., White S. M., 2007, PASP, 119, 805
Offringa A. R., de Bruyn A. G., Biehl M., Zaroubi S., Bernardi G., Pandey V. N., 2010, MNRAS, 405, 155
Petroff E. et al., 2015, MNRAS, 451, 3933
Petroff E., Hessels J. W. T., Lorimer D. R., 2022, Astron. Astrophys. Rev., 30, 2
Platts E., Weltman A., Walters A., Tendulkar S. P., Gordin J. E. B., Kandhai S., 2019, Phys. Rep., 821, 1
Saliwanchik B. R. B., Slosar A., 2022, PASP, 134, 114503
Sanidas S., Caleb M., Driessen L., Morello V., Rajwade K., Stappers B. W., 2017, Proc. IAU Symp. 385 will focus on Astronomy and Satellite Constellations: Pathways Forward., 13, Cambridge Univ. Press, Cambridge, p. 406
Vafaei Sadr A., Bassett B. A., Oozeer N., Fantaye Y., Finlay C., 2020, MNRAS, 499, 379
Virtanen P. et al., 2020, Nature Methods, 17, 261
Wang H.-F., Yuan M., Yin Q., Guo P., Zhu W.-W., Li D., Feng S.-B., 2020, Res. Astron. Astrophys., 20, 114
Yang Z., Yu C., Xiao J., Zhang B., 2020, MNRAS, 492, 1421
Zhang B., 2023, Rev. Mod. Phys., 95, 035005

## APPENDIX A: INVESTIGATING THE CASES OF MISSING CANDIDATES AND ERRORS

We have identified two scenarios where the pipeline falsely reports no candidates (see Fig. A1). The first occurs when the pipeline cleanly exits, but the output file containing the information on the candidates gets corrupted. This happens when there are large numbers of candidates and many asynchronous processes are writing simultaneously. This leads to `ProTest` not

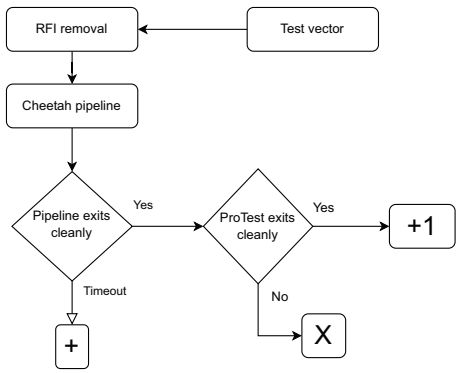


**Figure A1.** Flowchart describing the single pulse search and validation process. There are three results of a pipeline run – The cheetah pipeline may timeout due to a huge number of candidates, denoted by +, or the pipeline may run fine, but the exported file maybe corrupted due to a recognized pipeline bug, denoted by ×, or cleanly exit resulting in a number of recovered pulses.

being able to ingest the file. Such cases are marked with × in the plots in the Appendices. The second scenario occurs when a large number of candidates are produced by `ska-pss-cheetah` and the clustering algorithm takes too long. This leads to a timeout in the pipeline. These cases are marked with + in the plots in the Appendices. Both cases were revisited and searched over a smaller DM range and all candidates were recovered, supporting the arguments stated in the paper. Both aforementioned cases cause the pipeline to fail, primarily due to a large number of candidates, mainly due to residual RFI, which is the problem.

## APPENDIX B: RFI SCENARIO 2

The plots below contain all the detections from all the test vectors used from TVS-2. The injected RFI is a combination of Narrow-band, Periodic and Broadband RFI, with all possible cases of each type being either stronger ($2\sigma$) or weak ($0.5\sigma$). Figs B1, B2, B3, and B4 shows the results of using IQRM on different pulse widths of the injected pulses. Figs B5, B6, B7, and B8 shows the results of using IQRM in sequence with ZDMF. Figs B9, B10, B11, and B12 shows the results of using SKF with ZDMF. Every cell in the plots also displays the number of detections made. The marker × represents a case of an error by the software in exporting the candidates into a file, and + represents the case of the pipeline getting timed out because of an enormous number of candidates being detected, but both ×s and +s, have not affected our interpretation of the results (refer to Appendix A). The latter situation is prevalent when strong broad-band RFI is not fully removed and can mimic a pulse at many DMs and for many widths. The title of the subplots indicates the strength of the corresponding type of RFI in the test vectors used to generate the plot. For instance, if 'PNb' is the title of the subplot then the test vectors used contain periodic and narrowband RFI of strength $2\sigma$ and broad-band RFI of $0.5\sigma$ (more in Table 2).

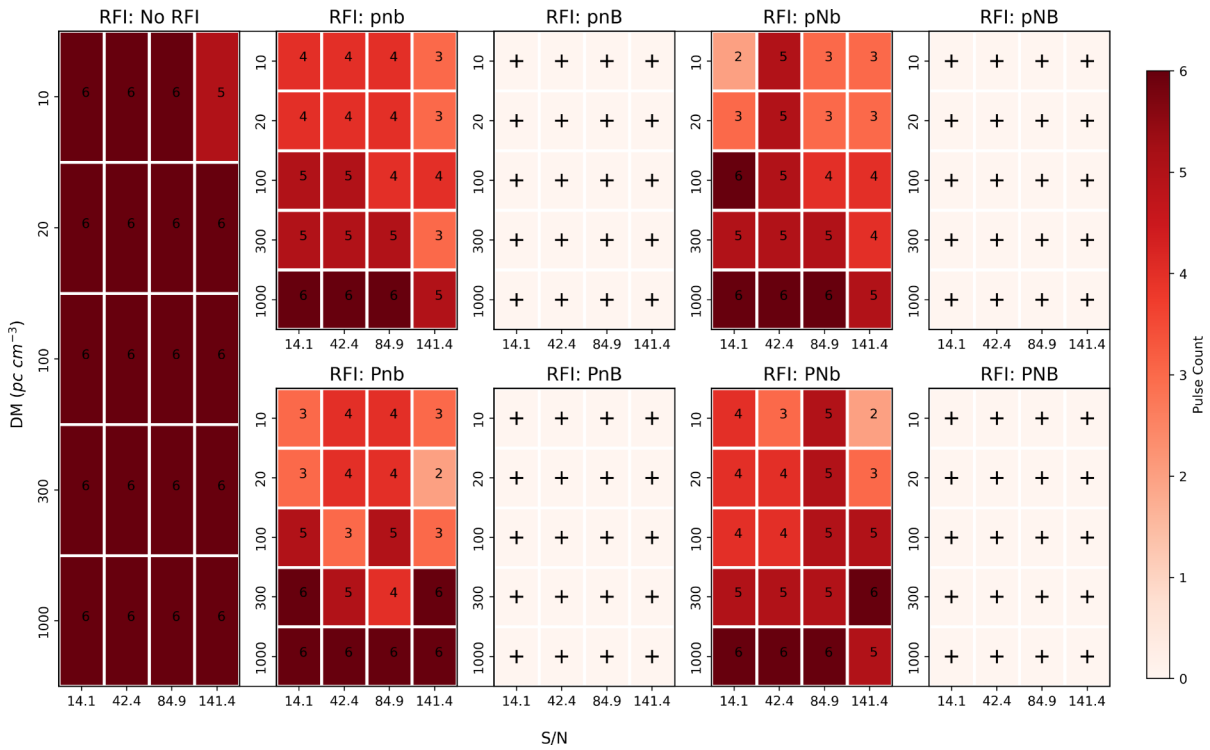


**Figure B1.** Number of detections of the pulses with a width of 800 ms cleaned using IQRM as a function of DM and S/N. The left-most panel shows the no RFI cases while other plots are for all the RFI combinations given in Table 2, with small letters (e.g. p) indicating weak RFI and capital letters (e.g. P) indicating strong RFI of that type. See Appendix A for explanation of the +'s and ×'s.

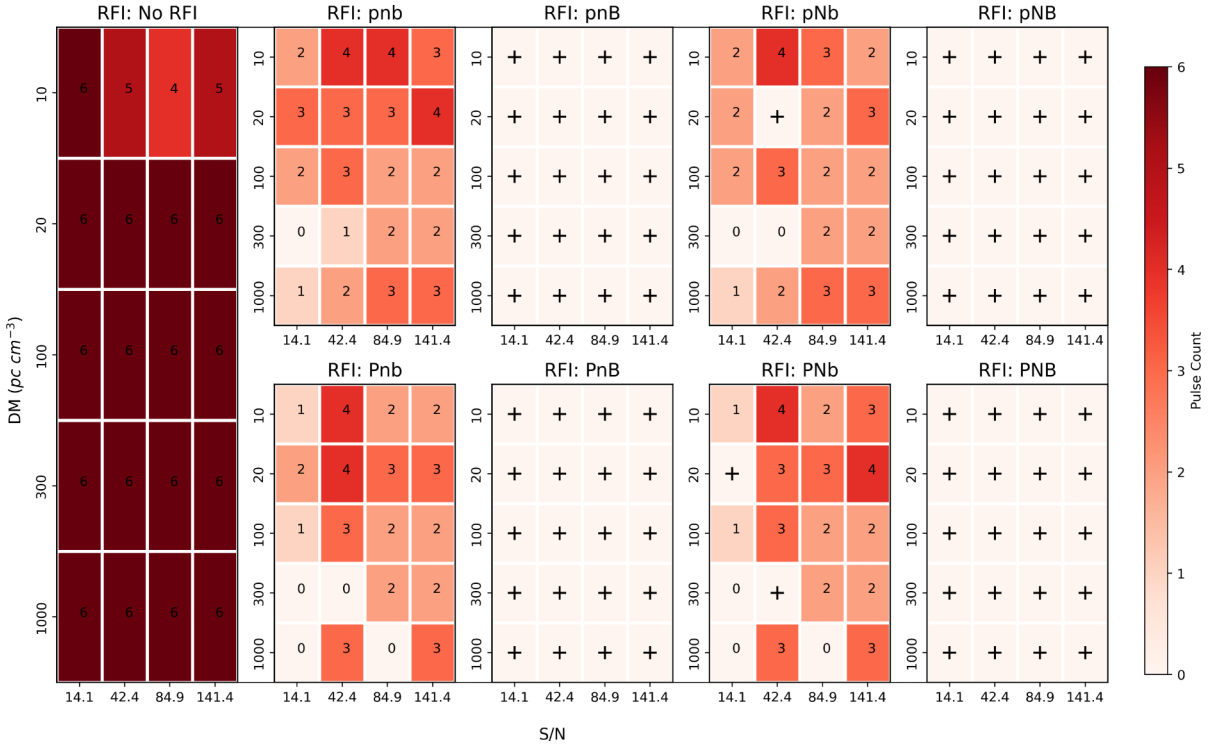


**Figure B2.** As for B1 but for a pulse width 80 ms.

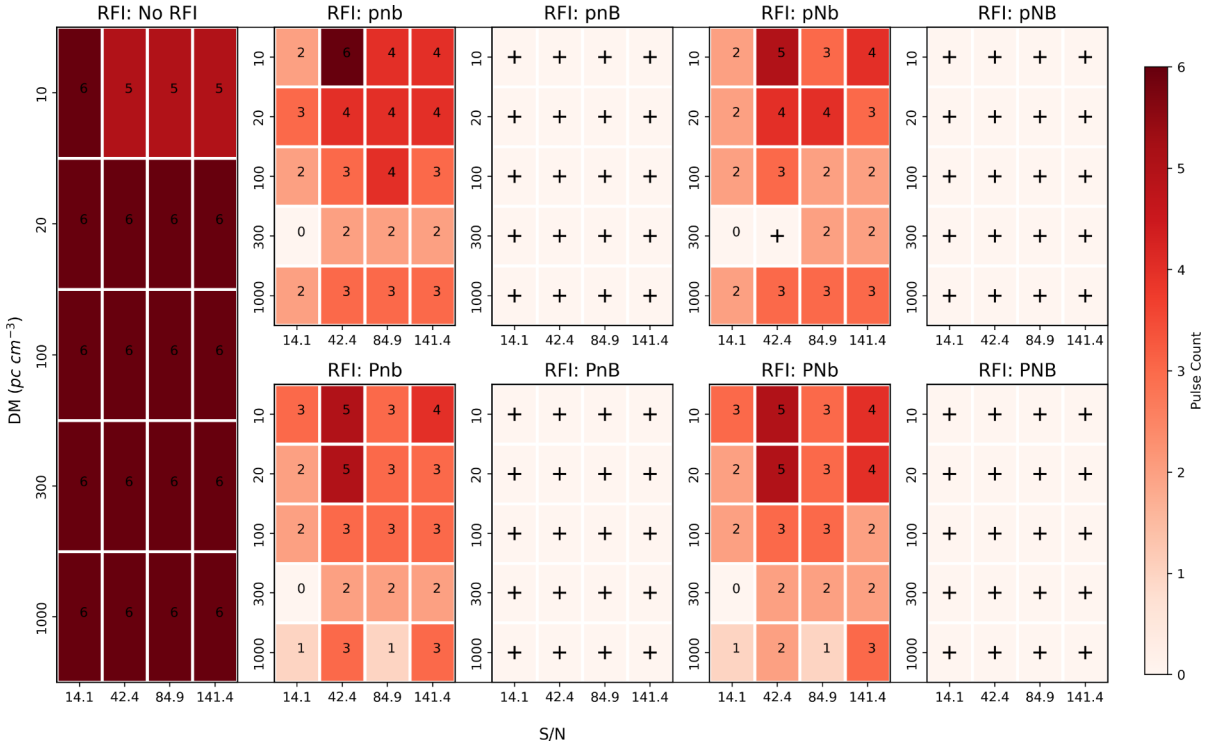


**Figure B3.** As for B1 but for a pulse width 40 ms.

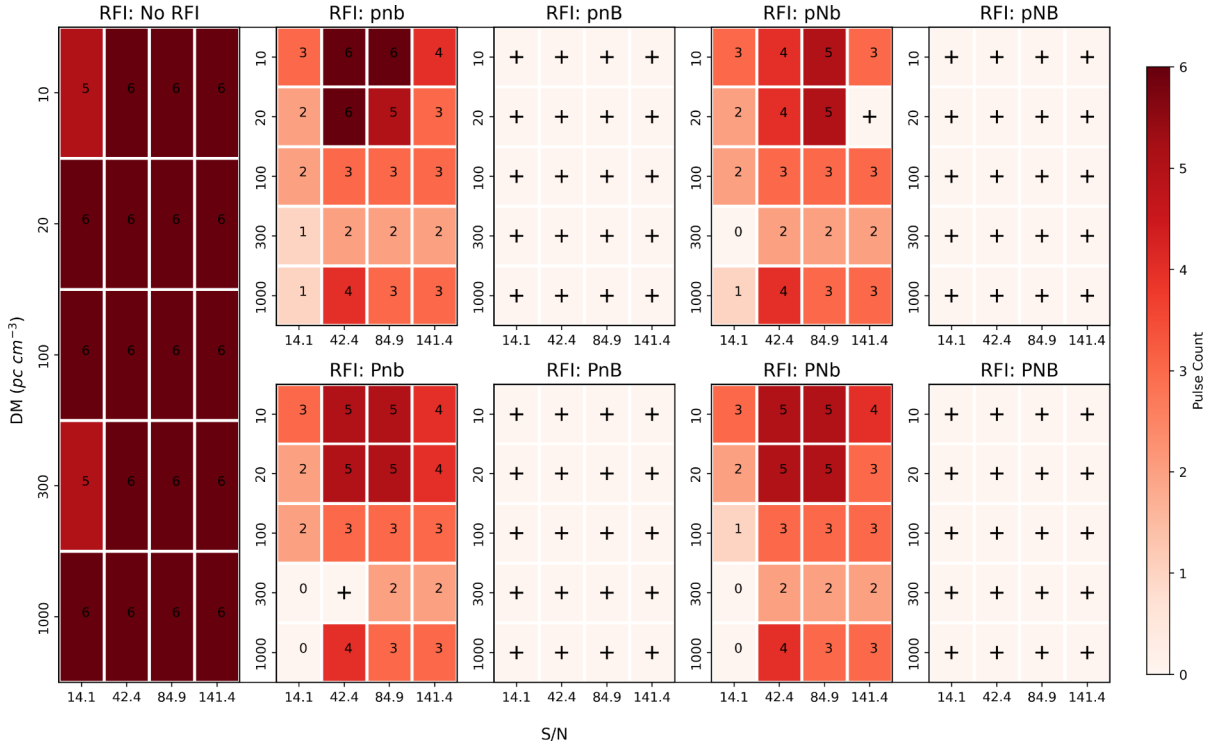


**Figure B4.** As for B1 but for a pulse width 8 ms.

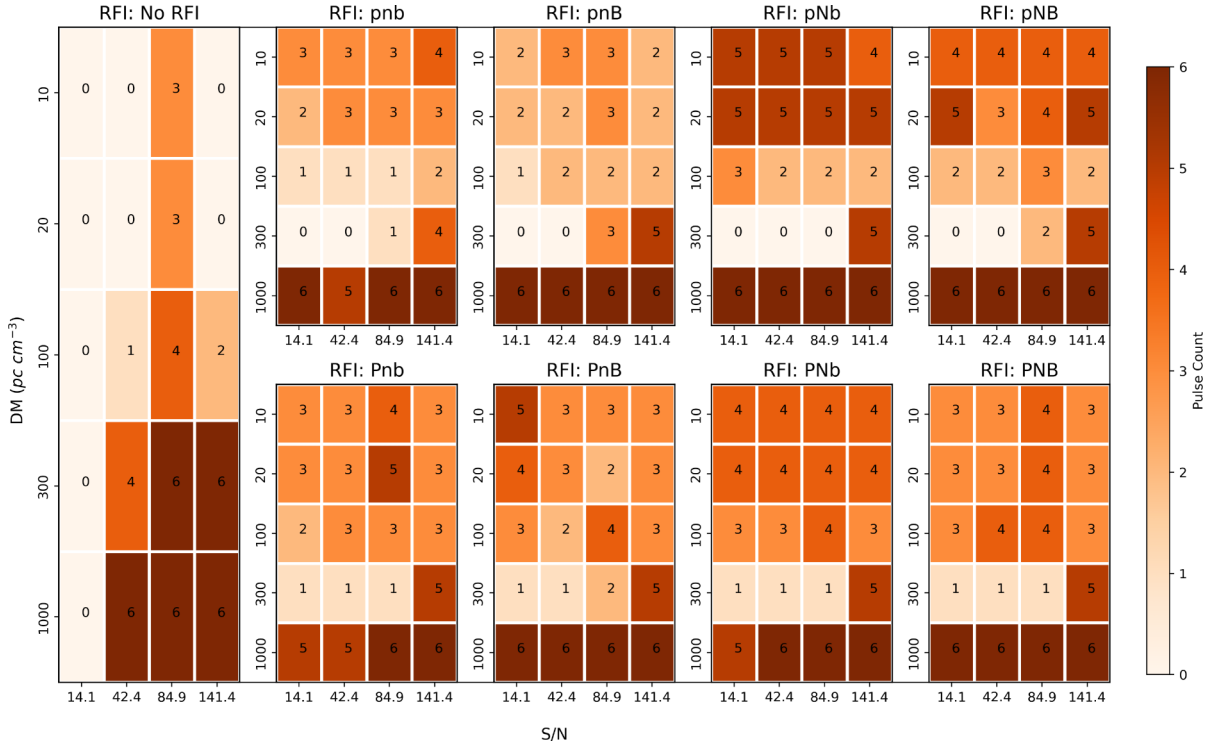


**Figure B5.** As for B1 but IQRM and ZDMF were used, and for a pulse width of 800 ms.

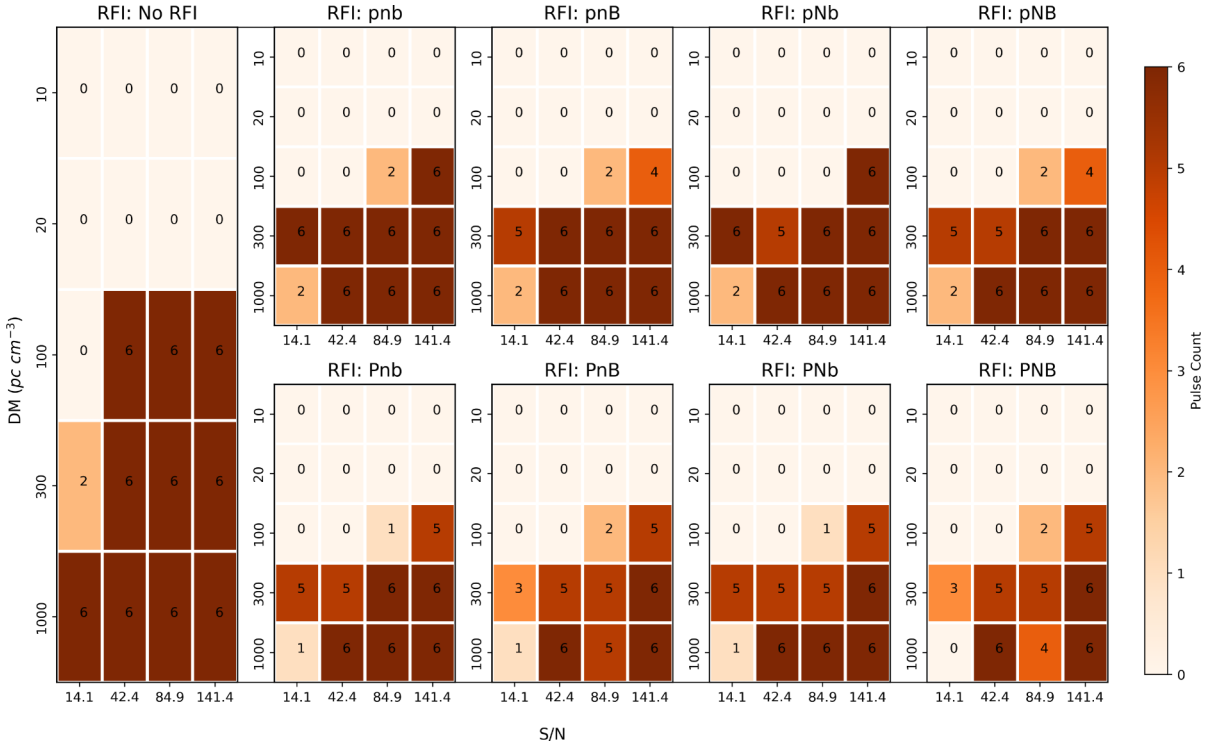


**Figure B6.** As for B5 but for a pulse width 80 ms.

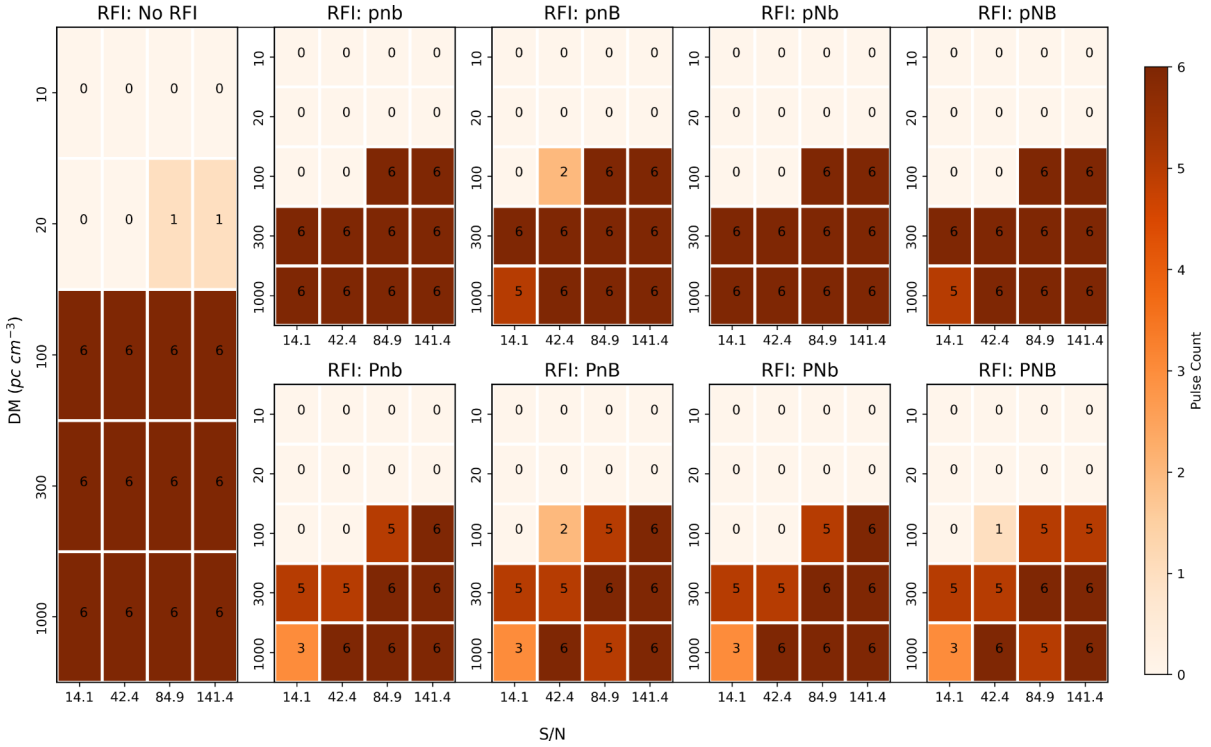


**Figure B7.** As for B5 but for a pulse width 40 ms.

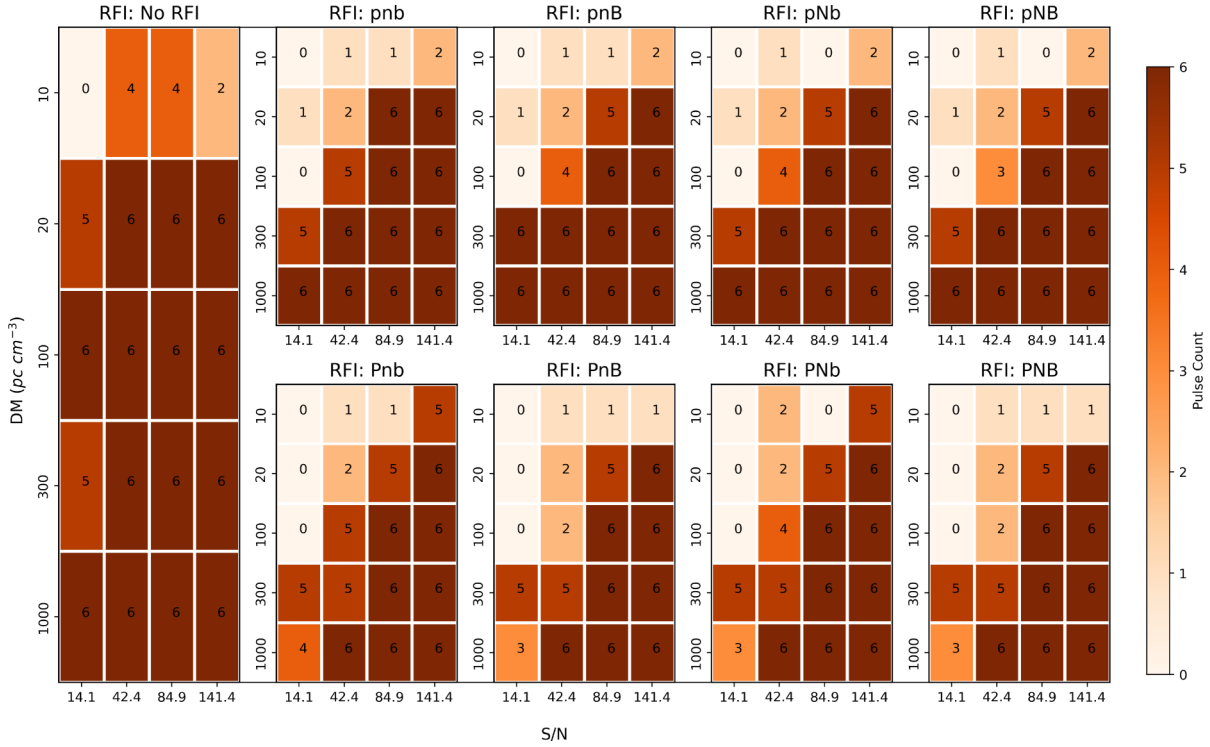


**Figure B8.** As for B5 but for a pulse width 8 ms.

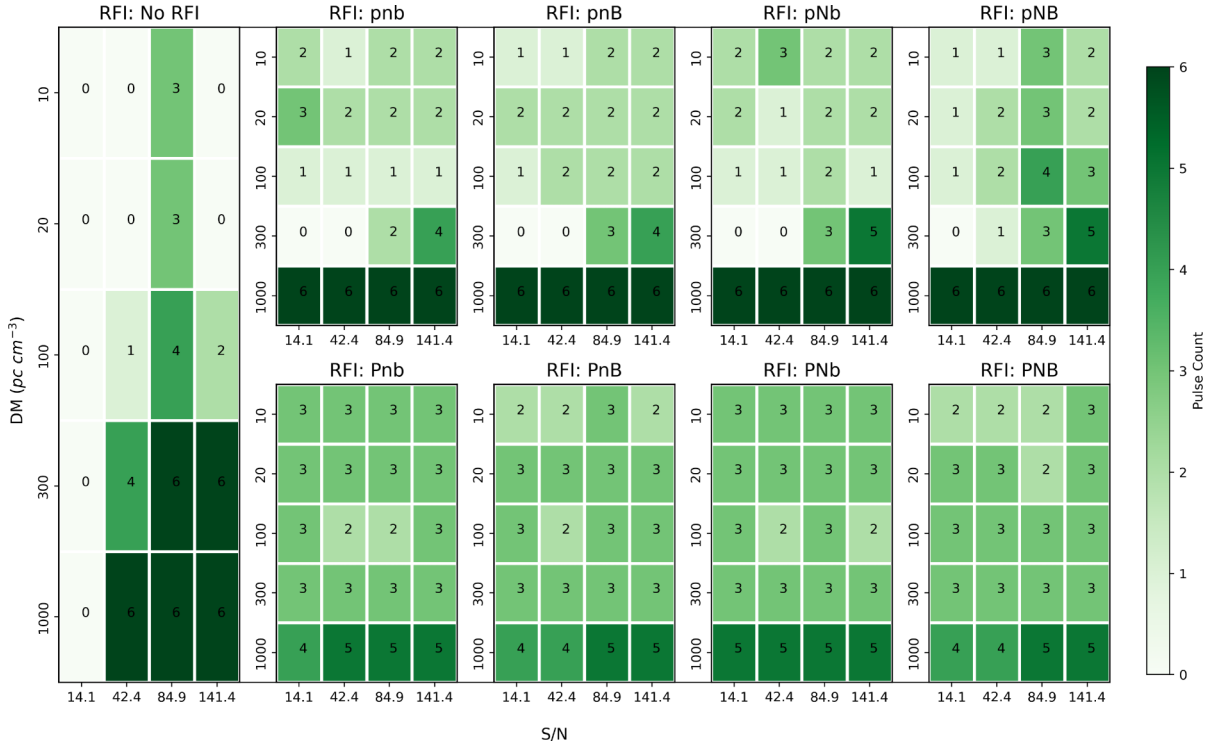


**Figure B9.** As for B1 but SKF and ZDMF were used, and for a pulse width of 800 ms.

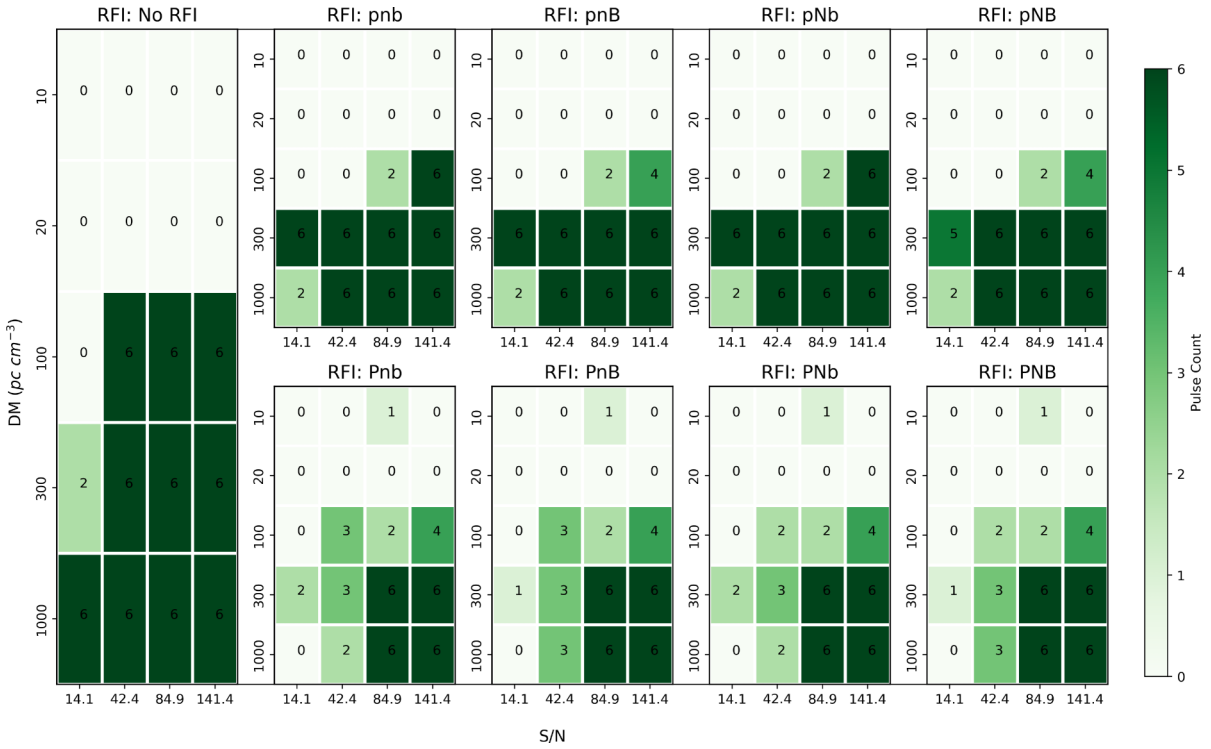


**Figure B10.** As for B9 but for a pulse width 80 ms.

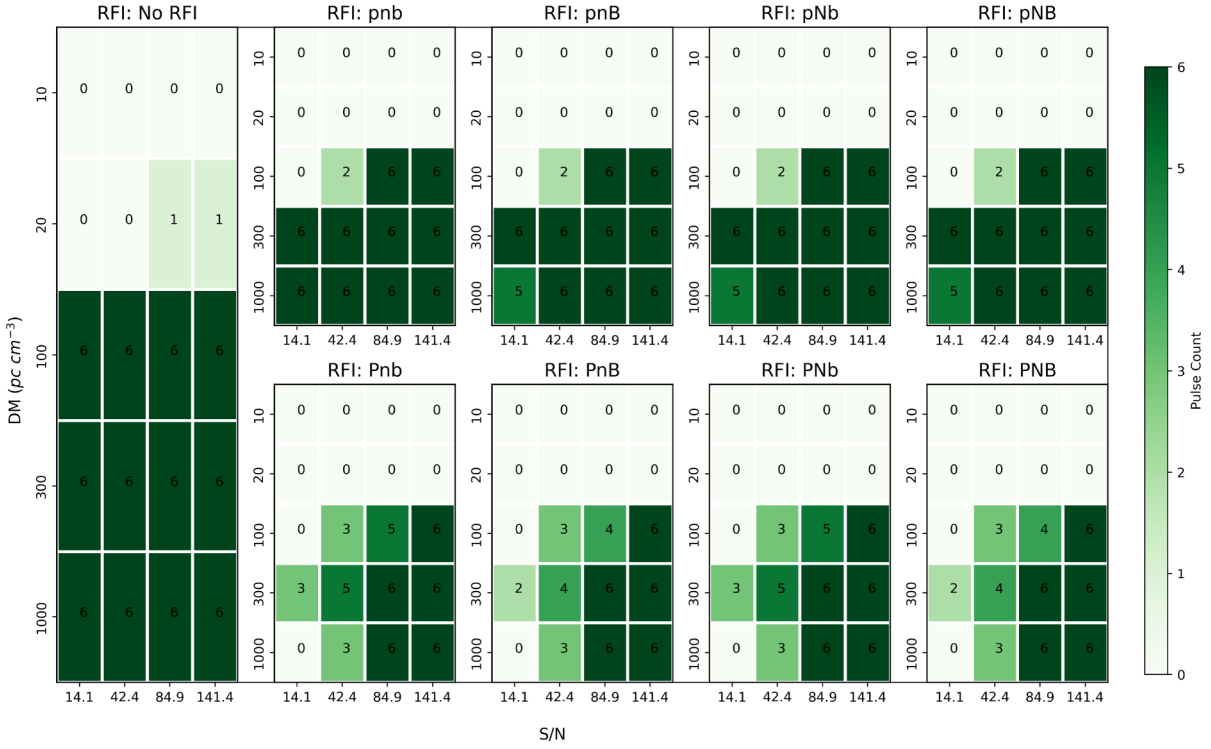


**Figure B11.** As for B9 but for a pulse width 40 ms.

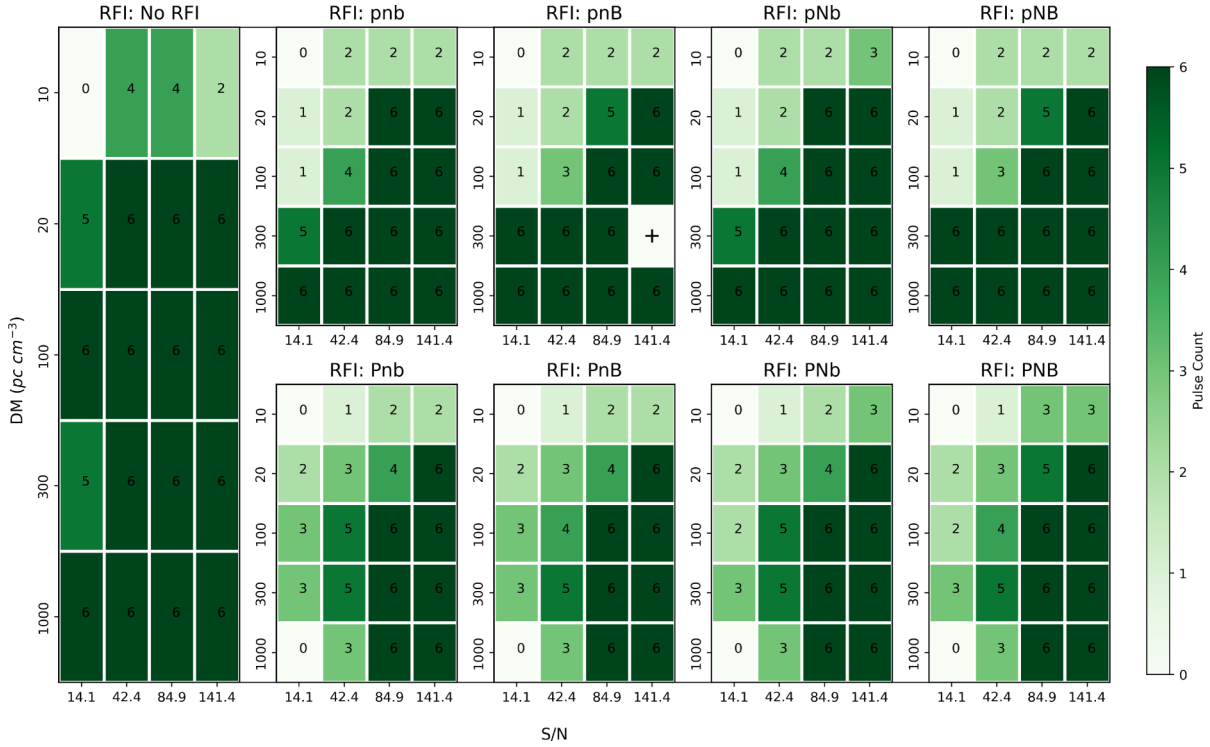


**Figure B12.** As for B9 but for a pulse width 8 ms.

## APPENDIX C: RFI SCENARIO 1 AND 3

Test vectors from TVS-1 are filterbank files containing no RFI, whereas test vectors from TVS-3 contain either only narrowband RFI or broad-band RFI. Figs C1 and C2 show a chunk of data in test vectors of TVS-3 used for the tests performed. Fig. C3 shows the same chunk of data when cleaned using ZDMF, IQRM, and SKF individually. Figs C4, C5, C6, and C7 show the number of pulses recovered when ZDMF is used to clean the test vectors containing broad-band RFI. Figs C12, C13, C14, and C15 show the results of the number of detections when SKF is used to clean the test vectors containing narrowband RFI. Figs C8, C9, C10, and C11 show the results of the number of detections when IQRM is used to clean the test vectors containing narrowband RFI. The × here represents the same as defined in the Appendix B.

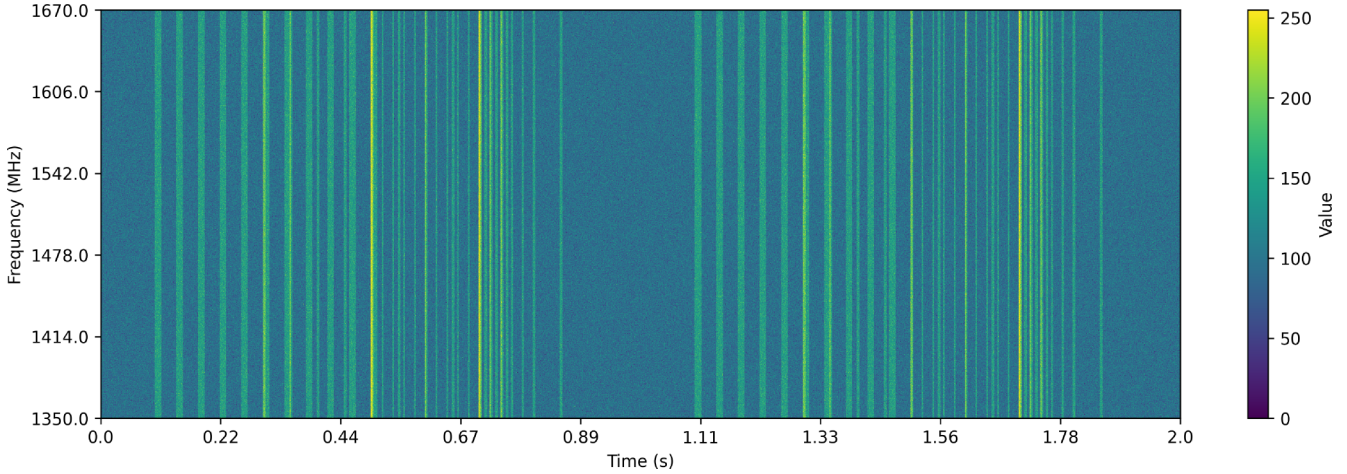


**Figure C1.** 1 s long data in a test vector filterbank used, containing only periodic broad-band RFI affecting 25 per cent of the total number of time samples in the filterbank file with a strength of $2\sigma$ (see Table 3).

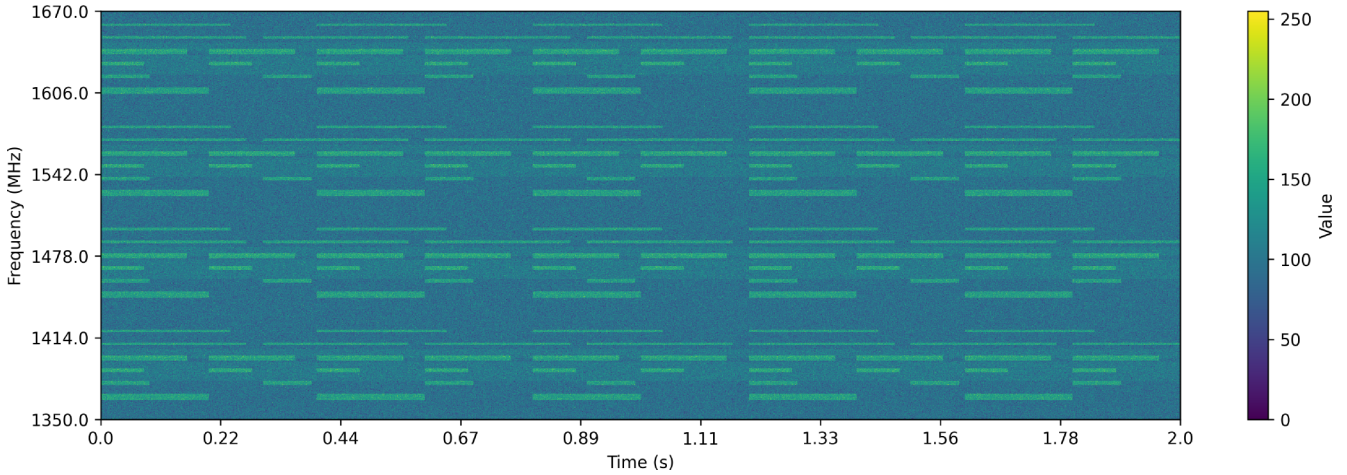


**Figure C2.** 1 s long data in a test vector filterbank used, containing only narrowband RFI affecting 25 per cent of the total number of frequency channels in the filterbank file with a strength of $2\sigma$ (see Table 3). This represents the worst-case scenario for periodic RFI.

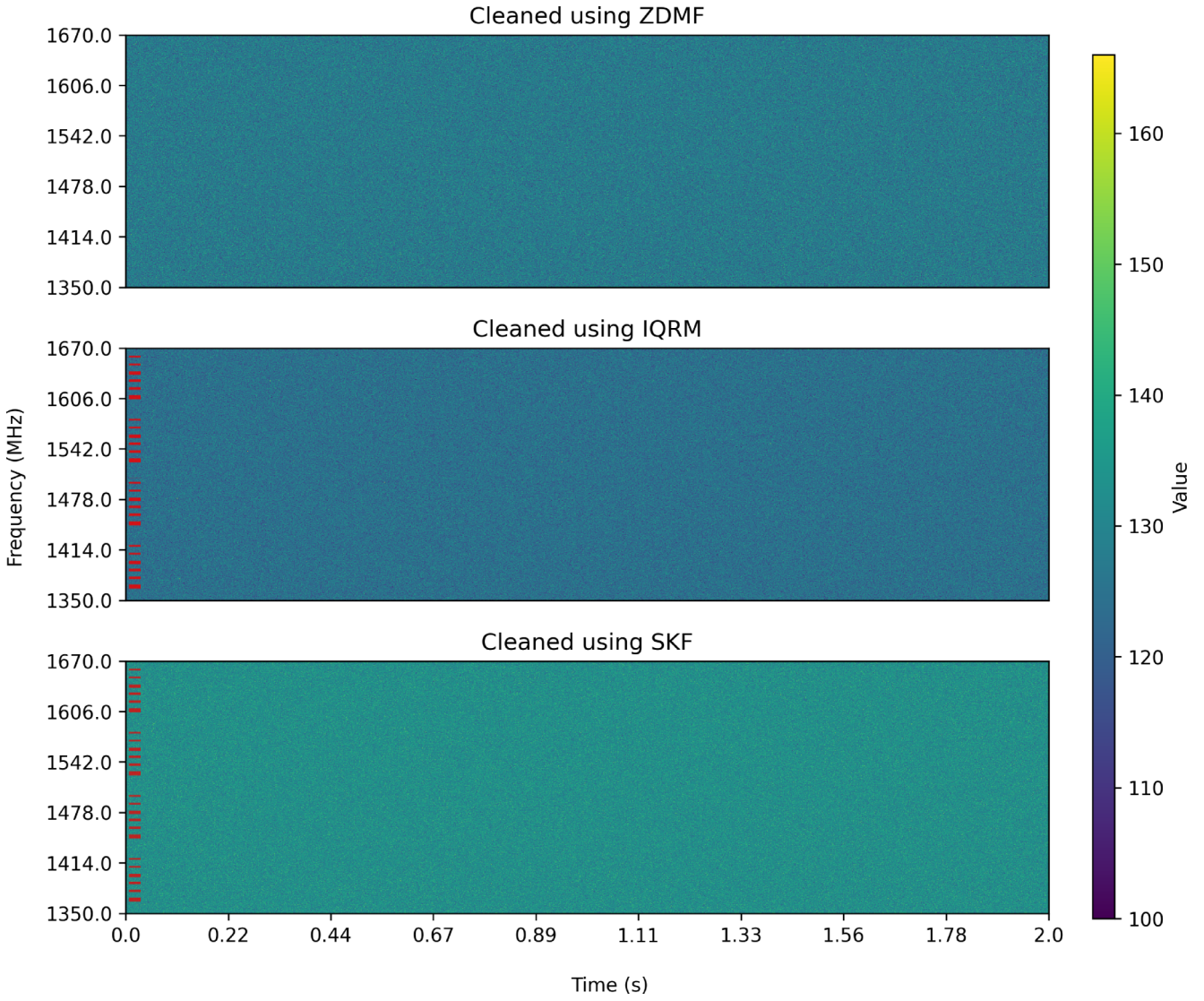


**Figure C3.** Demonstration of the chosen RFI removal algorithms, run individually, on the data shown in Fig. C1 (top panel) i.e. periodic broad-band RFI of $2\sigma$, cleaned using ZDMF, and Fig. C2 (middle and bottom panel) i.e. periodic narrowband RFI of $2\sigma$ cleaned using IQRM and SKF, similar to Fig. 4. The red lines at the left of the top two panels indicate the channels that are flagged as RFI-affected by the respective algorithms. filtool (used for SKF and ZDMF) corrects for the bandshape of the subset of data, whereas iqrm-apollo does not. To present the data comparable to each other, the middle panel in the figure showing data cleaned by IQRM is therefore also shown after correcting for its bandshape. Since ZDMF acts on the time samples, it does not mask any data but changes every time sample.

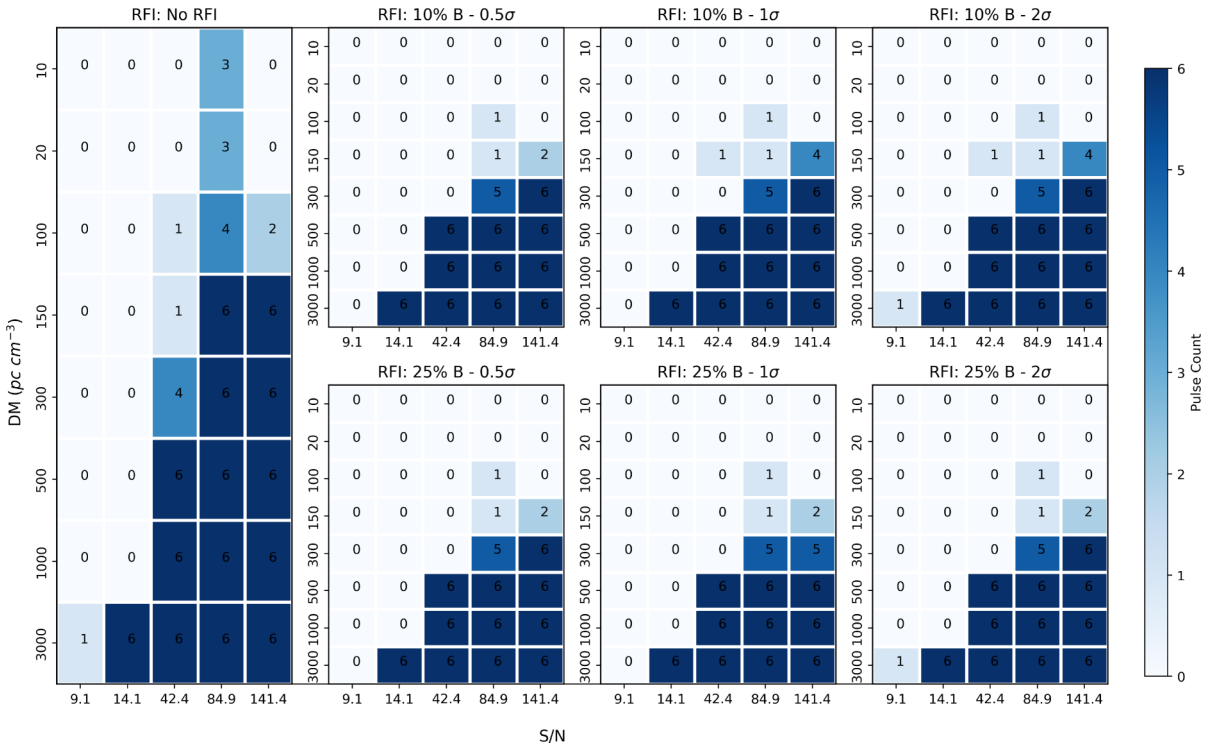


**Figure C4.** Number of detections of the pulses with a width of 800 ms cleaned using ZDMF as a function of DM and S/N. The left-most panel shows the no RFI cases while the other plots are for all the RFI combinations given in Table 3. See Appendix A for the explanation of any ✕'s.

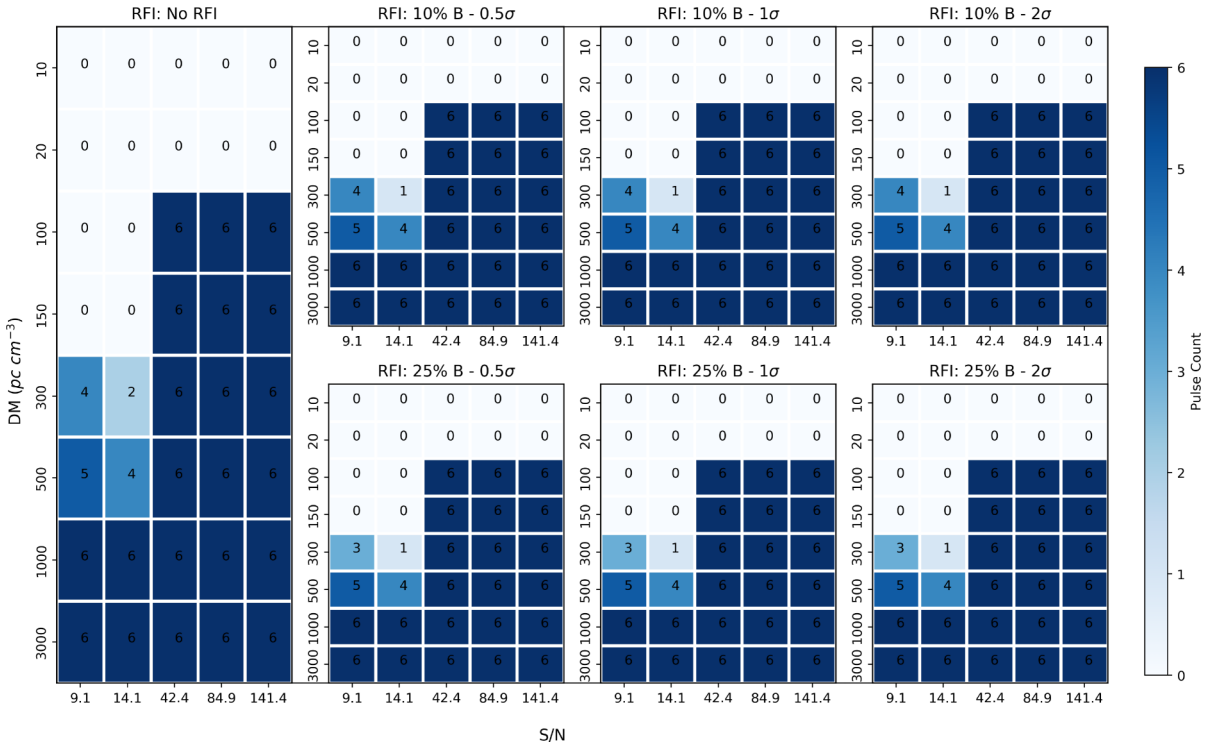


**Figure C5.** As for C4, but for a pulse width 80 ms.

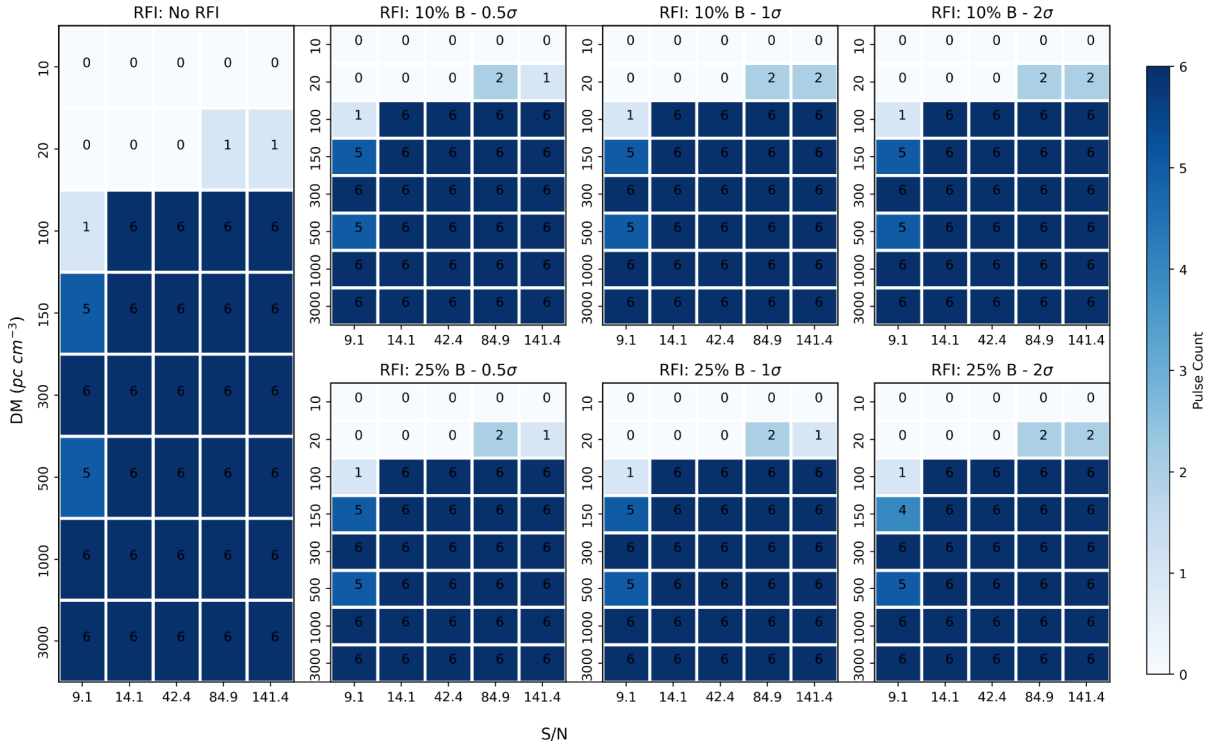


**Figure C6.** As for C4, but for a pulse width 40 ms.

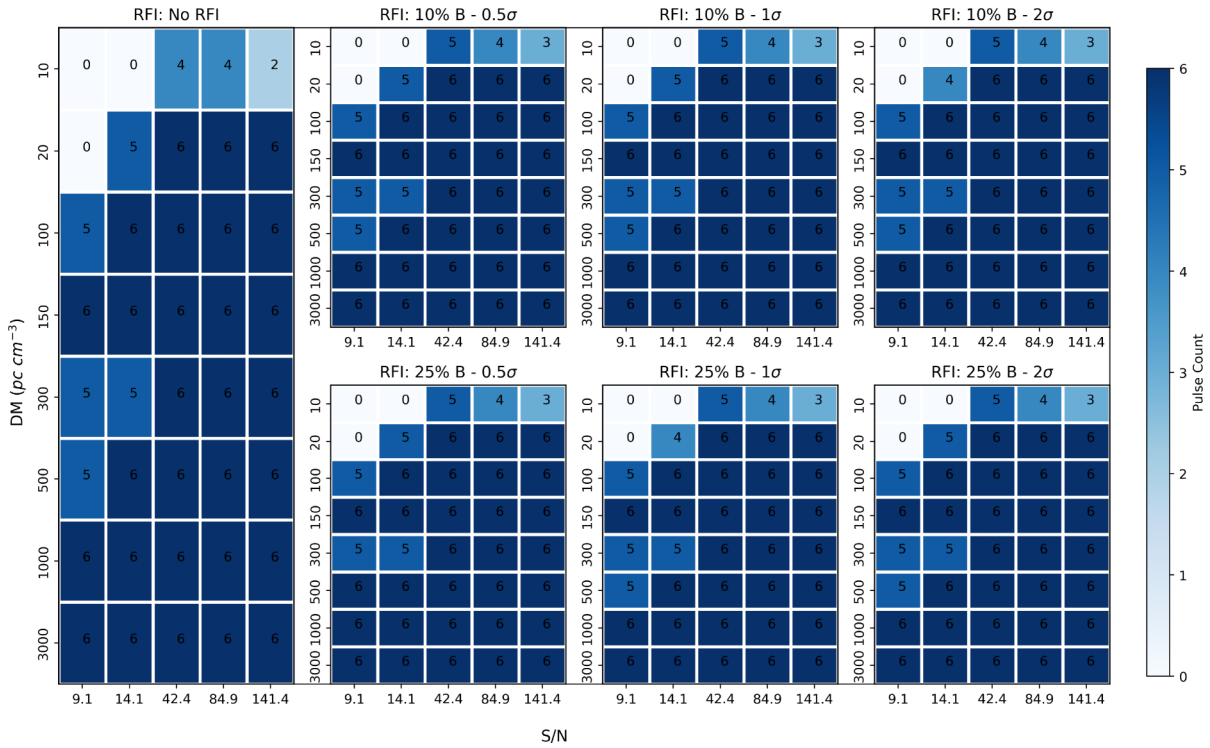


**Figure C7.** As for C4, but for a pulse width 8 ms.

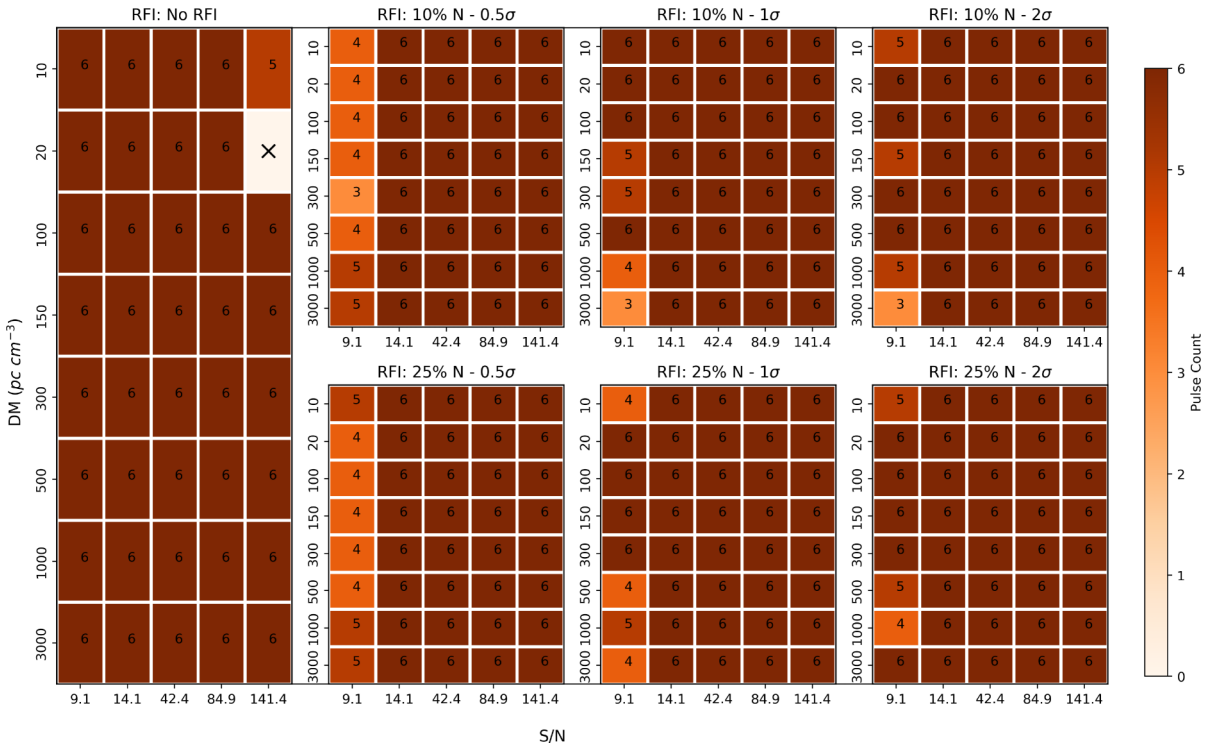


**Figure C8.** As for C4, but cleaned using IQRM, and for a pulse width of 800 ms.

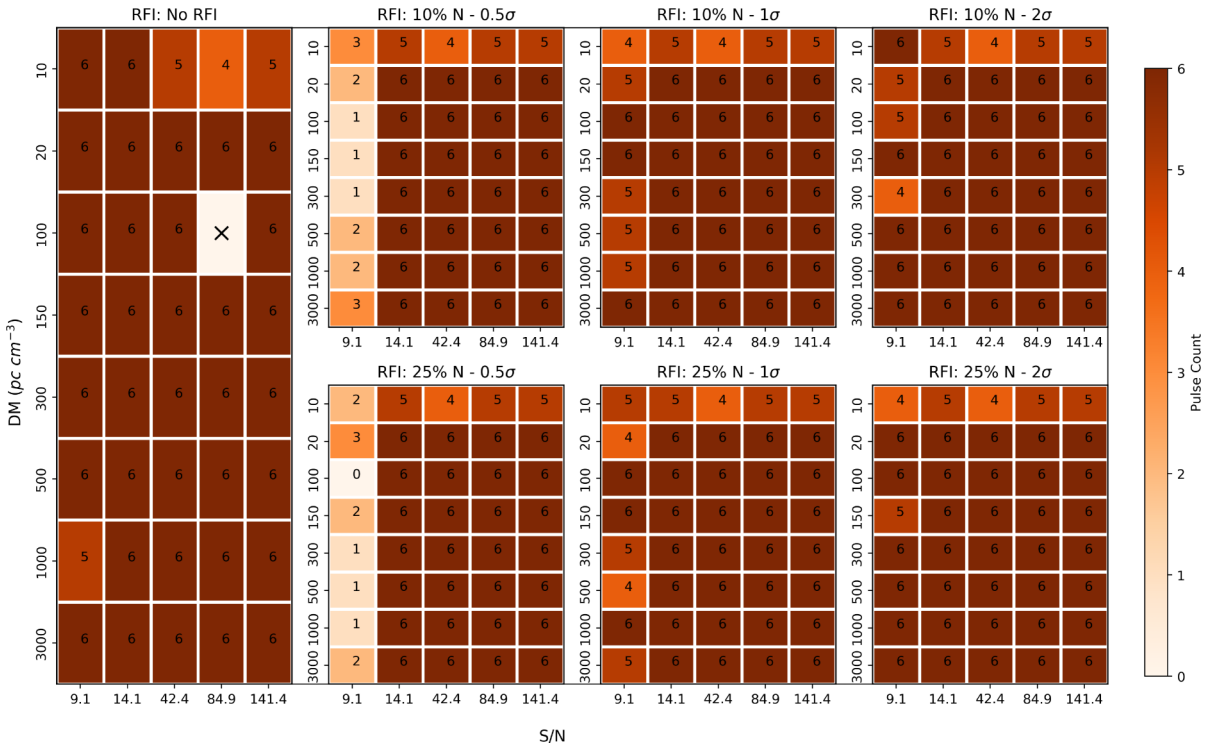


**Figure C9.** As for C8, but for a pulse width 80 ms.

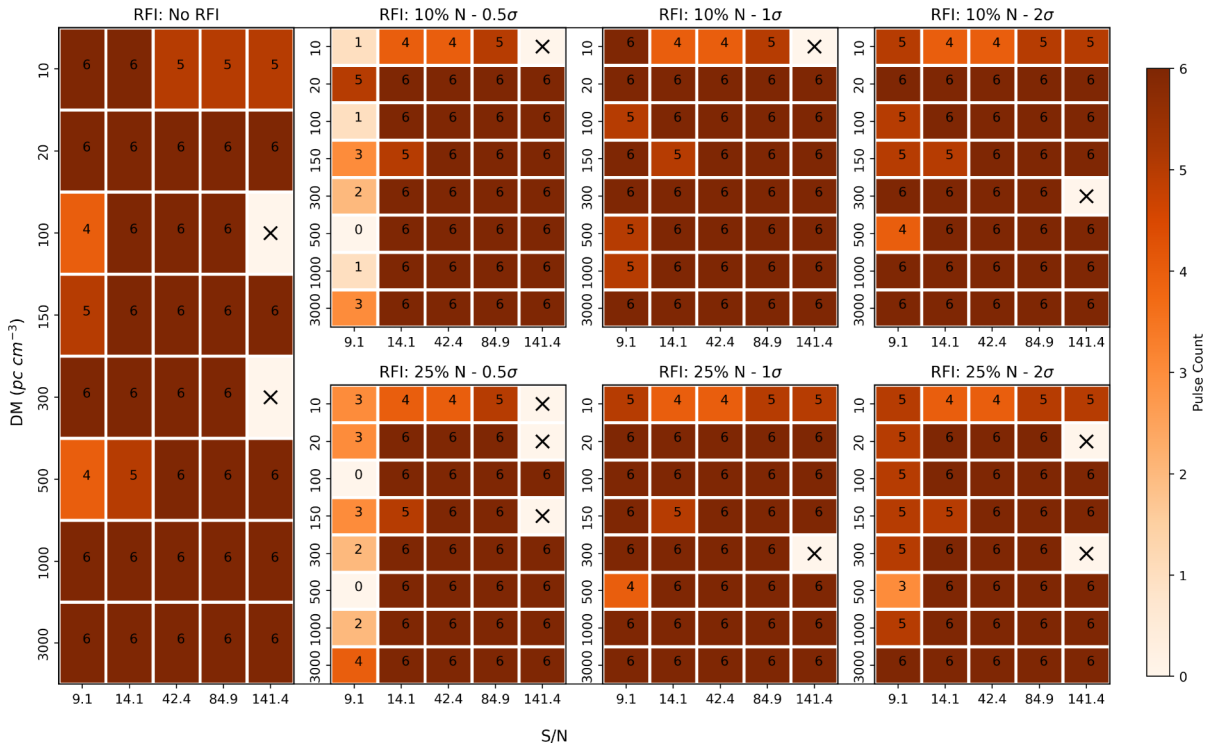


**Figure C10.** As for C8, but for a pulse width 40 ms.

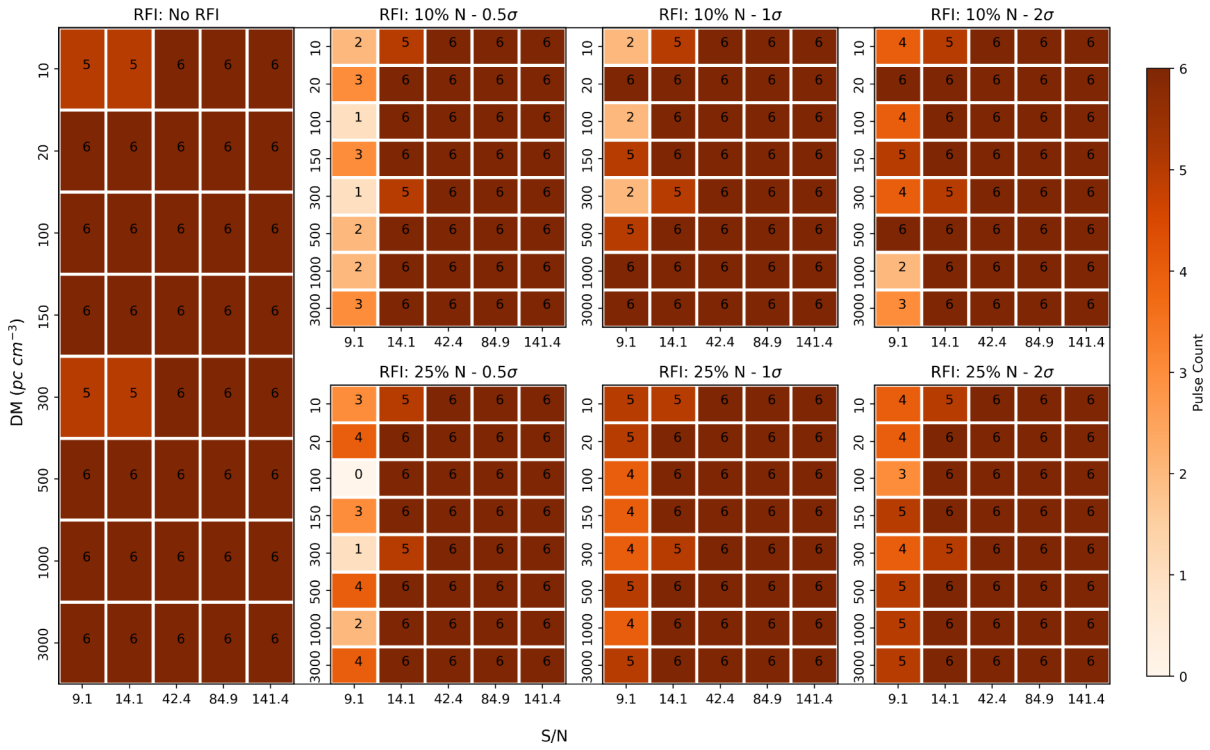


**Figure C11.** As for C8, but for a pulse width 8 ms.

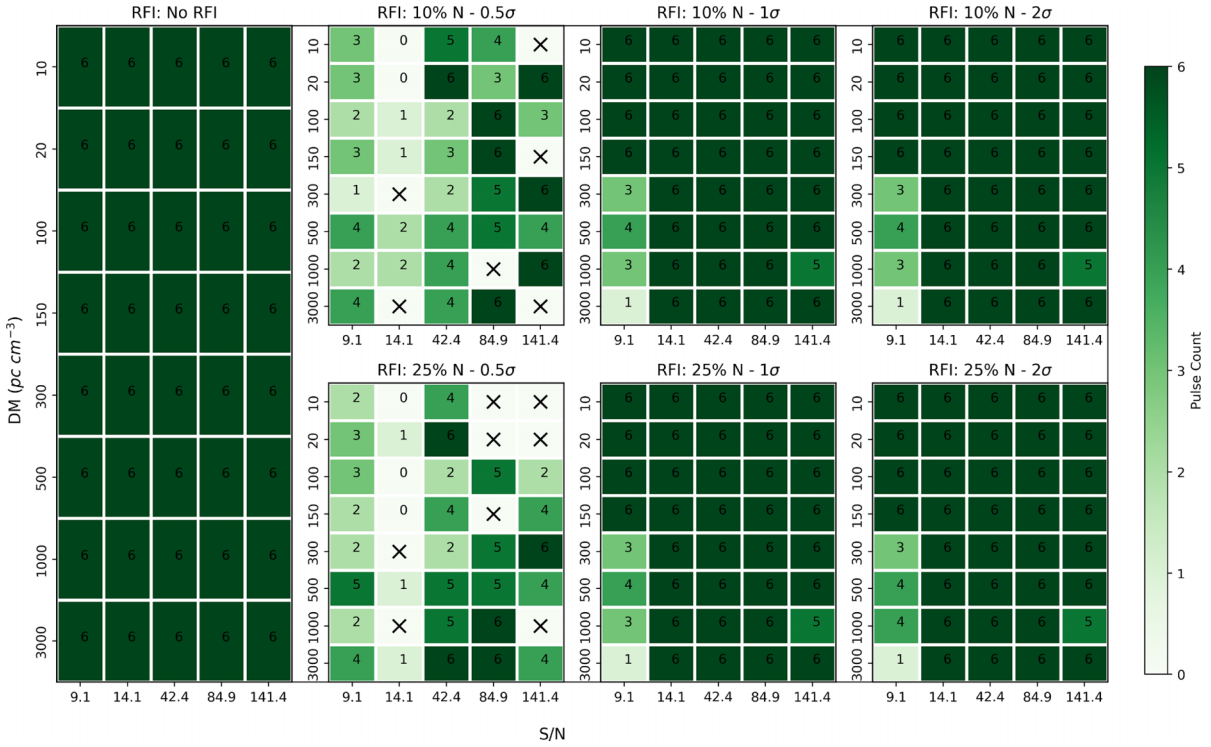


**Figure C12.** As for C4, but cleaned using SKF, and for a pulse width of 800 ms.

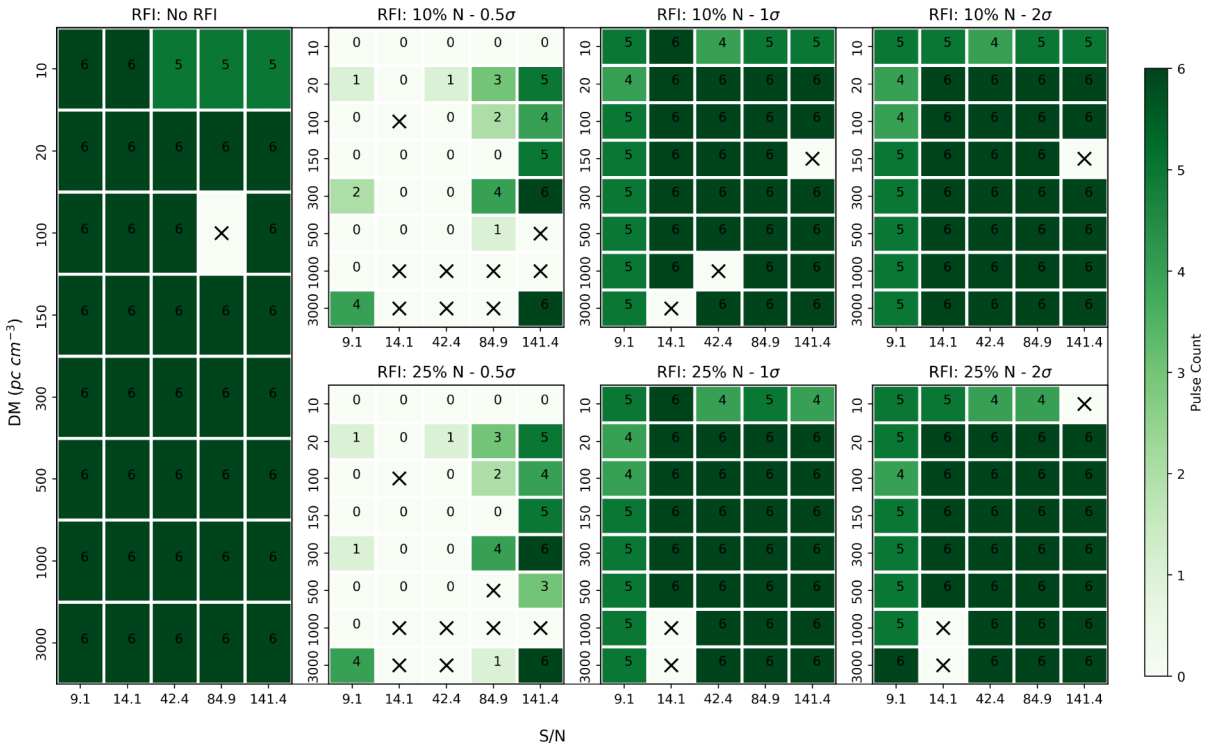


**Figure C13.** As for C12, but for a pulse width 80 ms.

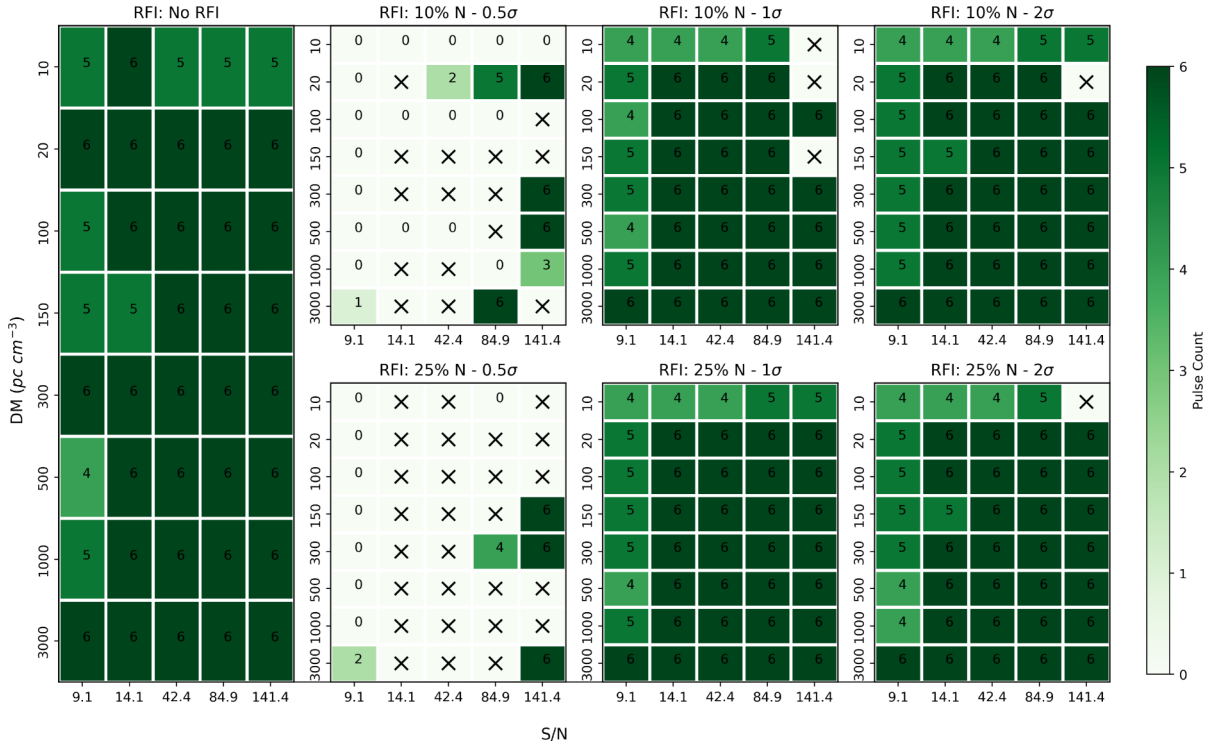


**Figure C14.** As for C12, but for a pulse width 40 ms.

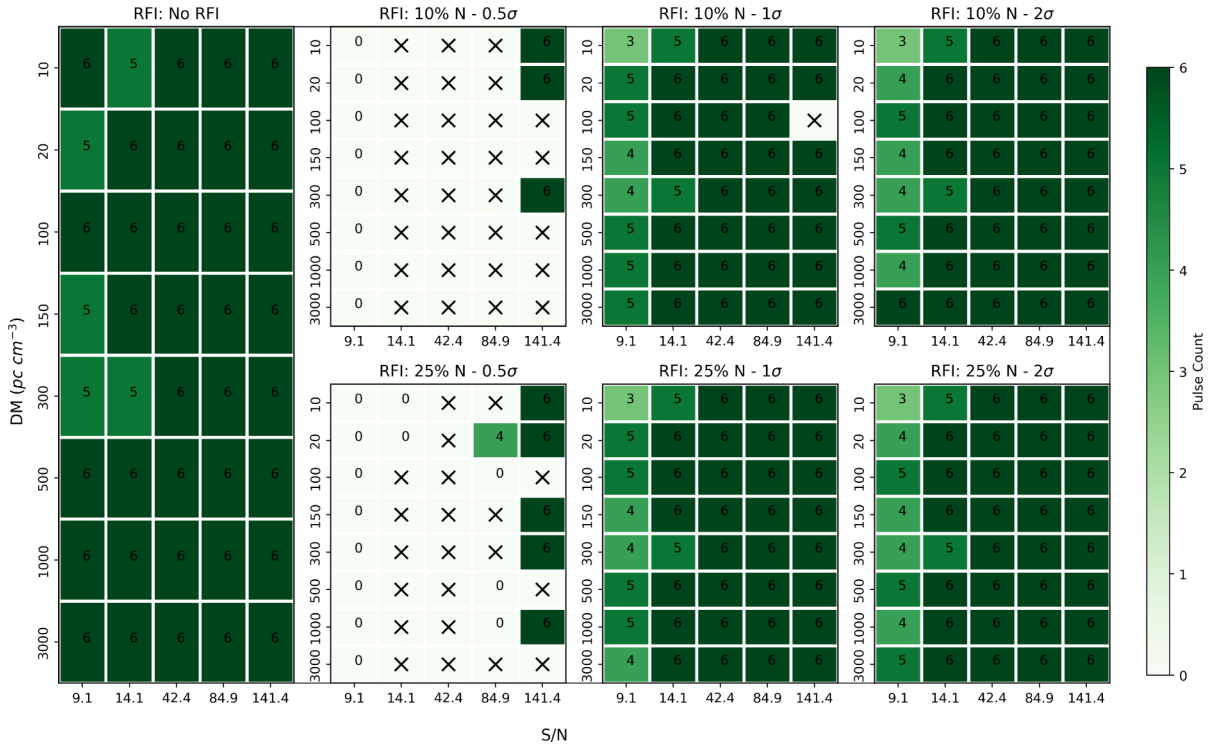


**Figure C15.** As for C12, but for a pulse width 8 ms.

This paper has been typeset from a TEX/LATEX file prepared by the author.