# WALLABY/DINGO Source Finding Busy Week 4 Review

## 1 Source finding overview

### 1.1 Source finding tasks

The term 'source finding' is generally used to refer to a sequence of four tasks. These tasks are:

1. Data pre-processing
2. Source detection
3. Source parameterisation
4. Catalogue post-processing

The simplest source finders only carry out the source detection task. A minimally useful source finder carries out the source parameterisation task as well. We will use these task labels to categorise the work done during the 4th WALLABY/DINGO Source Finding Busy Week. It is important to note that various algorithms and techniques can (and do) accomplish multiple tasks at the same time. An example of this is achieving good source parameterisation because your source detection algorithm has found most of your source.

### 1.2 Task glossary

Various methods have been developed (or are in development) for each source finding task. For this reason, we will define the tasks in terms of their goals rather than processes.

*Data pre-processing:* Source detection algorithms necessarily make assumptions about the noise properties of datasets. One of the goals of data pre-processing is to check that a dataset's noise properties satisfy these assumptions, and to adjust the dataset if required. The other goal of dataset pre-processing is to enhance the signal-to-noise ratios of sources. Wavelet denoising methods (such as the 2D-1D transform) are an example of data pre-processing that tries to improve the signal-to-noise ratio of sources. It is important to note that the two data pre-processing goals do not have to be done simultaneously.

*Source detection:* Source detection is any method or algorithm that identifies regions in a dataset as containing signal of some sort. Source detection is evaluated using two metrics, completeness and reliability. Completeness is the fraction of sources in a dataset that are found. Reliability is the fraction of identified sources that are real. A good source detection method/algorithm has both high completeness and high reliability.

The identification of dataset artifacts is usually counted against the source detection reliability, but technically, this is a region in the dataset that is not pure noise. For this reason, some source finders do not consider detection of artifacts as a source detection failure.

*Source parameterisation:* Source parameterisation is any method/algorithm that measures properties of sources from the dataset. Good source parameterisation produces parameters that are both accurate (unbiassed) and precise for all sources, independent of their properties. In practice, low signal-to-noise sources are mathematically unconstrained, and accurate, precise source parameterisation is very difficult. This is compounded by source detection biases. For example, using intensity thresholding for source detection preferentially finds low signal-to-noise sources overlapping with one or more large positive noise spikes. Source parameterisation is also made more difficult by convolution effects. Within a convolved interferometric dataset, sources do not have well-defined edges (technically their spatial extent is infinite) or global properties (such as total flux, which sums to zero over the image).

*Catalogue post-processing:* The goal of post-processing a catalogue is to refine the catalogue by removing and/or flagging dataset artifacts and false source detections. In practise, catalogue post-processing is the act of trying to assign a reliability value to individual sources or small groups of sources in a catalogue.

## 2 Noise scaling

*Task:* Data pre-processing

Most source finders apply a pre-defined or calculated noise value when searching for objects. Using a single noise estimate works fine if the noise is perfectly Gaussian. This is usually the case in simulations and can occur on small datasets, however it is unrealistic for large data-cubes. Typically noise variations occur at the edges of cubes, for example because a telescope receiver is less sensitive at the edge of the bandwidth. What is less predictable however are variations due to RFI, solar interference, continuum artefacts, etc. When all these features are left in the data-cubes it is very hard to determine a good noise estimate, and the source-finding will result in many false detections. A possible way to improve upon this is to make data-cubes with sigma values that estimate the local noise, and essentially filtering out or weighting down the problematic regions.

We have created sigma-scaled cubes for all the HIPASS cubes, by calculating the "MAD"

statistic (median absolute deviation) in each of the three dimensions. Starting with the spectral dimension we scaled each plane by the MAD value and did the same for the two spatial dimensions. The MAD statistic gives better results than normal rms values, as it is less sensitive to strong outliers.

In Figure 1 we show the results obtained with a similar Duchamp parameter file to the original HIPASS cube 19 and a noise-scaled version. Within Duchamp we use a search threshold of three sigma and a GrowthThreshold of 2 sigma. The top two panels show the results for the normal HIPASS cube where all detections are plotted as a function of position in the X, Y and Z planes. The bottom panels show similar plots for the noise-scaled cubes. The noise-scaled cubes give a much cleaner result with significantly less (false) detections. Bad regions with a lot of RFI that are especially visible in the X-Z plane are clearly filtered out.

The noise scaling has been applied to all HIPASS cubes. When doing an initial Duchamp test-run, without trying to optimise the search parameters, the HIPASS catalogue was recovered with a completeness of more than 90% and a reliability of more than 20%. Although these are not the final benchmark numbers to be aiming for, they are a significant improvement on previous results. This example highlights the importance of pre-processing datasets.

# 3 Fitting of Spectral Profiles of Galaxies

*Task:* Source parameterisation

## 3.1 Context

One method of parametrising galaxies is by fitting suitable functions to the integrated spectral profile. In the simplest case, the function could be a single Gaussian component. However, many galaxies exhibit a distinct, double-peaked profile for which a more complex function will be required.

Previous attempts to fit double-peaked profiles to galaxy spectra involved Gaussian components. Obreschkow et al. (2009) used a combination of two different functions to model the outer flanks and the inner dip of the profile separately, namely

$$S(v) \propto \begin{cases} \exp\left[-\frac{(v-v_0)^2}{c_1}\right] & \text{for } |v-v_0| \geq \Delta v_\mathrm{p}/2, \\ \frac{1}{\sqrt{c_2-v^2}} & \text{for } |v-v_0| < \Delta v_\mathrm{p}/2, \end{cases} \tag{1}$$

where $\Delta v_\mathrm{p}$ is the separation between the two peaks. The problem with this approach is that the profile gets broken up into two separate functions that would have to be fitted separately. In addition, the Gaussian flanks of the profile do not describe the general steepness of real galaxy profiles very well.

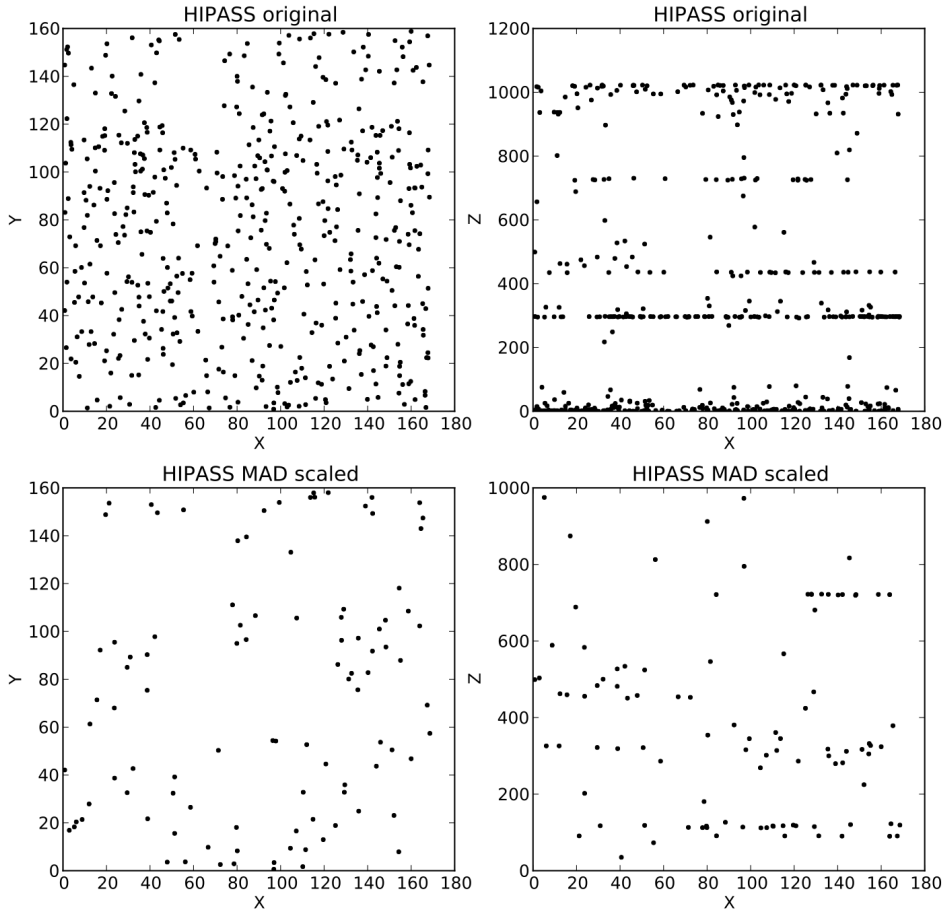


Figure 1: Duchamp results for HIPASS cube 19 before and after noise scaling.

A different approach was chosen by Saintonge (2007) for the ALFALFA survey on the Arecibo telescope. She selected a linear combination of two Hermite functions of the form

$$S(v) = a_0\Psi_0(v,\sigma) + a_2\Psi_2(v,\sigma), \tag{2}$$

but the problem again is that in this case the flanks of the spectrum are not steep enough to reflect the sharp rise in the spectra of real galaxies.

### 3.2 The "Busy Function"

In order to improve on previous attempts to model or fit double-peaked profiles to galaxy spectra we were looking for a function that would allow us to describe the steep flanks that are observed the spectra of real galaxies while also recovering the characteristic dip and sharp, narrow peaks of the spectrum. Such a function can indeed be constructed in a relatively simple fashion by multiplying two error functions with a simple parabola:

$$\begin{aligned} S(v) = a_0 &\times [\mathrm{erf}(a_1(v - v_0 + \Delta v)) + 1] \\ &\times [\mathrm{erf}(a_1(v_0 - v + \Delta v)) + 1] \\ &\times [a_2(v - v_0)^2 + 1]. \end{aligned} \tag{3}$$

Here, the two error functions, $\mathrm{erf}(x)$, generate the steep rise and drop of the spectrum, while multiplication with a parabola produces the central dip with relatively sharp peaks. Given its characteristics and flexibility, we decided to call this function the "busy function", and the motivation for this name will become clear in a moment.

The busy function is characterised by five free parameters: the centroid of the spectrum, $v_0$, the overall width of the spectrum, $\Delta v$, the total scaling factor, $a_0$, and two additional parameters, $a_1$ and $a_2$. The parameter $a_1$ controls the steepness of the two error functions describing the flanks of the spectrum. In the case of $a_1 \to \infty$ the flanks will become infinitely steep, whereas for $a_1 \to 0$ the slope of the flanks will approach zero. The parameter $a_2$ controls the emphasis of the parabola and hence the amplitude of the central dip. Values of $a_2 > 0$ imply increasing amplitudes of the dip, whereas for $a_2 = 0$ the dip will disappear altogether. Negative values of either $a_1$ or $a_2$ are not physically sensible in the case of spectral profile fitting of galaxies.

One advantage of the busy function, and the motivation for its name, is its flexibility when it comes to fitting spectral profiles of different shapes. Two examples of the busy function mimicking double-peaked profiles of different width are presented in the top panels of Fig. 2. By charefully choosing appropriate values for $a_1$, $a_2$, and $\Delta v$, almost any shape of (symmetric) double-peaked profile can be reproduced by the function. The bottom panels of Fig. 2 show two actual examples of HIPASS galaxies with fitted busy functions that are similar to the two examples shown in the top panels.

The flexibility of the busy function goes well beyond the fitting of double-peaked profiles, as is illustrated in the two panels of Fig. 3. In the left-hand panel, the parameter $a_2$

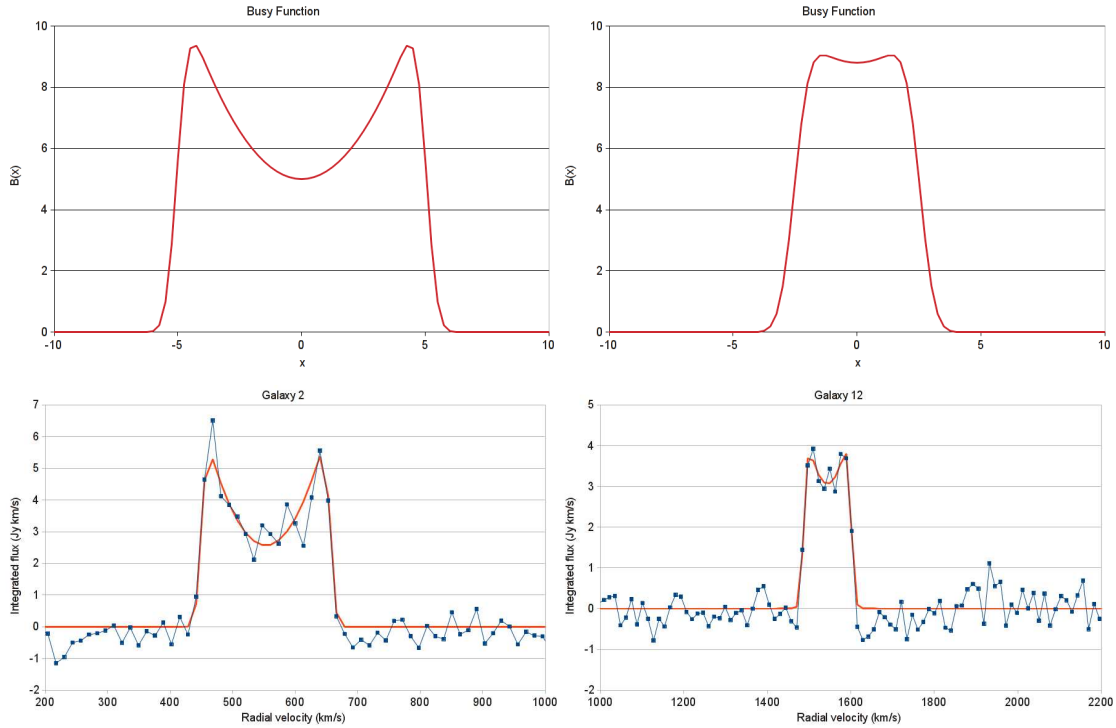


Figure 2: Two examples of the busy function imitating a wide (top-left) and narrow (top-right) double-peaked profile. The example on the left-hand side was created using $a_0 = 1.25$, $a_1 = 2$, $a_2 = 0.05$, and $\Delta v = 5$. The example on the right-hand side uses $a_0 = 2.2$, $a_1 = 1.5$, $a_2 = 0.02$, and $\Delta v = 2.5$. In both cases, $v_0 = 0$. The bottom panels show the integrated spectra of two galaxies extracted from HIPASS (blue) with fitted busy functions (orange) similar in shape to the two examples shown in the top panels.

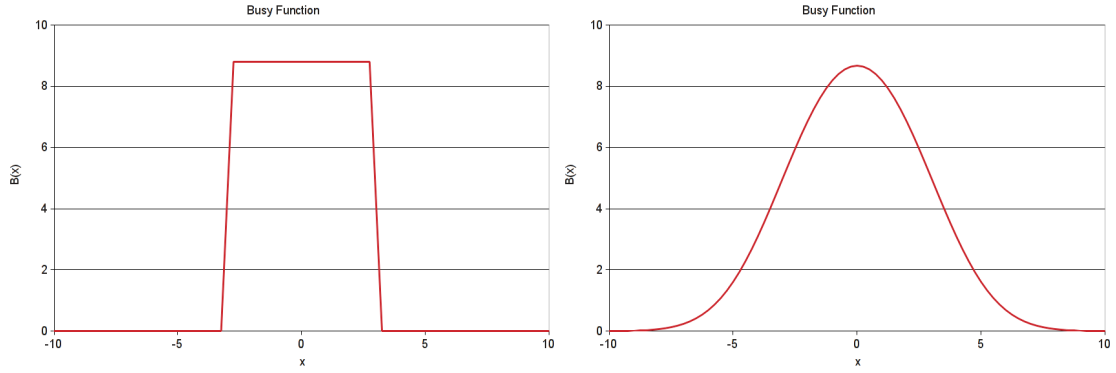


Figure 3: Two examples of the busy function imitating a top-hat function (left) and a Gaussian function (right). The example on the left-hand side was created using $a_0 = 2.2$ and $a_1 = 10$. The example on the right-hand side uses $a_0 = 2.5$ and $a_1 = 0.35$. In both cases, $v_0 = a_2 = 0$ and $\Delta v = 3$.

was set to zero to entirely remove the parabolic component and, in combination with a large value of $a_1$, produce an almost rectangular top-hat function. The example in the right-hand panel of Fig. 3 uses almost the same parameters, but this time taking a smaller value of $a_1$ to form a perfectly Gaussian shape with gradual slopes on both sides.

In its current form of Eq. 3, the busy function is symmetric, but at the cost of one or two additional free parameters it can easily be turned into an asymmetric double-peaked profile. One form of asymmetry would be to introduce a separate $v_0$ for the parabola, resulting in a shift of the central depression with respect to the line centroid and peaks of different height. Another option would be to provide each error function with its own $a_1$ parameter to achieve different slopes for the two flanks of the line. However, introducing these two changes would increase the number of free parameters to seven, which would potentially compromise the robustness of the fit.

### 3.3 Fitting Algorithm

As the busy function is composed of only two basic functions, the error function and a polynomial, it can be easily fitted to real spectra using a non-linear $\chi^2$ minimisation algorithm, such as the commonly used Levenberg–Marquardt algorithm. Many data fitting programs that allow the user to define custom functions to be fitted, e.g. QtiPlot, can be readily used for this purpose.

During the source finding busy week in Sydney we implemented a separate fitting algorithm for the busy function in the C++ programming language, using the Levenberg–Marquardt algorithm. As input the algorithm expects the data to be fitted, the corresponding uncertainties, and an initial estimate of the five free parameters of the fit. The algorithm will then carry out the $\chi^2$ minimisation and report the reduced $\chi^2$ as well as the five parameters and their uncertainties back to the user. All floating point variables have been implemented using double precision.

## 4 Flux error and reliability estimation from mask shifting

*Task:* Source parameterisation & Catalogue post-processing

Given the mask of a source candidate, an estimate for the error in the calculated total flux can be obtained by shifting the mask to multiple positions in the vicinity of the position of the source candidate and measuring the total flux obtained at the shifted position.

By repeating the process several times, an estimate for the local noise level can be obtained by calculating the dispersion over the obtained noise levels. This allows to assign

a meaningful error to the total flux value determined for a given source candidate. Also, if one takes the median of the differences between the candidate flux and the flux measured at the shifted positions, the total flux can be determined without contamination from any local offset, e.g. baselines.

Using robust statistics, this method also works if the source candidate is in the vicinity of a much brighter source.

Using the determined total flux and scatter and assuming Gaussian noise, one can assign a reliability to the source candidate. More accurately, one can calculate the probability of obtaining the measured flux from noise only, as opposed to a true underlying signal.

Figures 4 and 5 show the results for a simulation. In all cases the source of interest is a perfect point source that extends over 40 channels and has a signal-to-noise ratio of 0.5 in each channel.

In both figures the top-left panel shows the histogram of the differences between the measured source flux and the flux of the shifted masks. The top-right panel shows the moment map in which the source mask is shown as the black contour line. The bottom-left panel shows the position of the mask shifts and the measured difference at each location. The bottom-right panel is a spectrum through the peak of the source with its spectral extent indicated by the red lines.

In the first case (Figure 4), one can easily see that this method is very sensitive to the presence of the source. Even though the source would not be distinguishable from the background as seen in the moment map, the histogram shows a clear trend.

In the second case (Figure 5), a much brighter, close-by source is present in the field of interest. The effect from the source is clearly visible in the histogram as well as the shift map. Since robust statistics are employed, the measured median and dispersion are only marginally affected.

# 5 Refining astronomical catalogues with data mining

*Task:* Catalogue post-processing

We have observed that real sources and noise sources are separated in n-dimensional parameter space. This allows us to draw cuts in parameter space that separate the high and low reliability sources. There are however two complications. First, there is not a distinct separation between real sources and noise sources, but a fuzzy one. We have worked on improved source optimisation to make the distinction between the real sources and noise sources clearer. Second, it is very hard to find an optimal cut in an n-dimensional parameter space.

During the 4th WALLABY/DINGO Source Finding Busy Week, we investigated the possibility of using data mining techniques to address this second problem. We identified

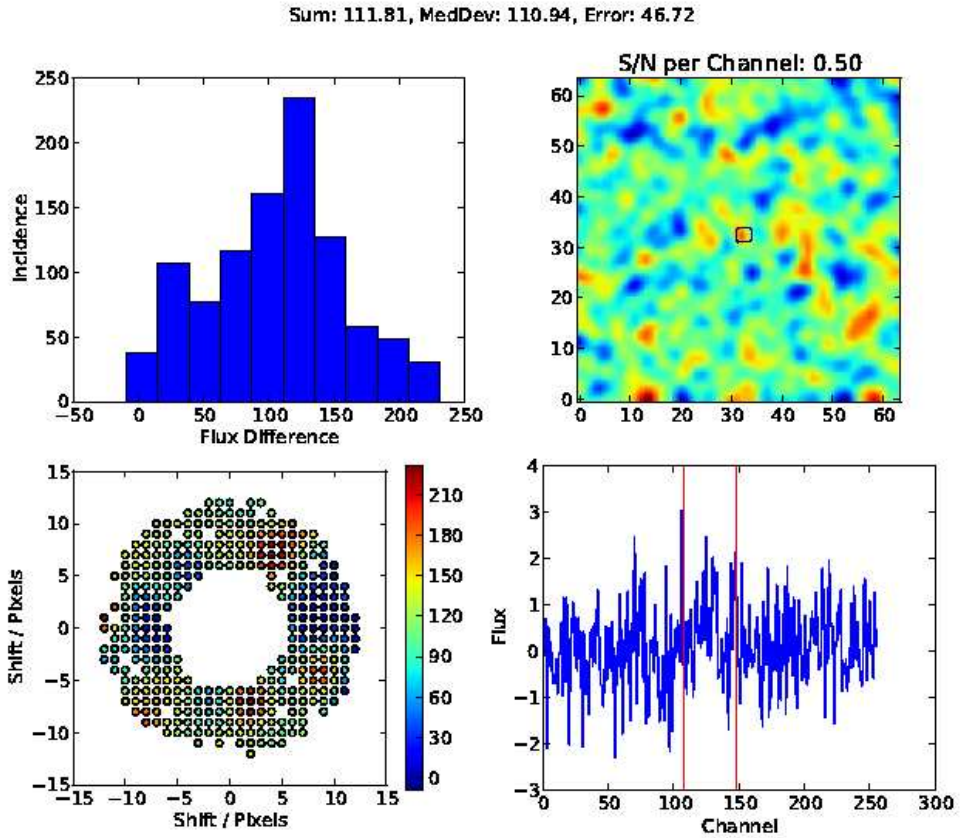


Figure 4: The mask shifting results for a simulation of a single source.

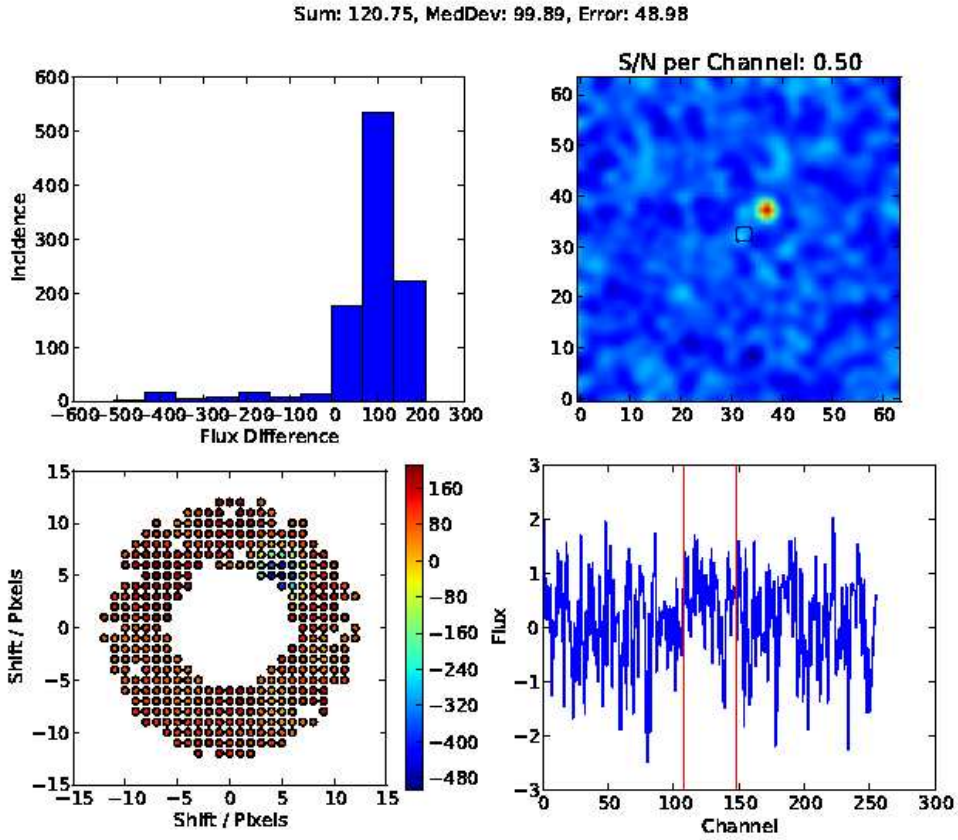


Figure 5: The mask shifting results for a simulation of a source with a brighter source nearby.

the Unsupervised Fuzzy Clustering Optimal Number of Partitions algorithm as the best for our purposes. The UFC-ONP algorithm assigns partial membership of every object to K classes. These classes are defined by their 'prototypes'. A cluster prototype is the n-dimensional mean[1] of the properties of the cluster objects.

The UFC-ONP algorithm iterates upon the cluster prototypes until prototypes are maximally similar to the objects that constitute their cluster, and maximally different from each other. The clusters are assumed to be maximally different, when each cluster prototype is maximally similar to that cluster's objects. The similarity of a cluster prototype to the cluster objects is measured by the cluster variance. This is not an n-dimensional variance. To calculate it a metric must be assumed that transforms distances between the various dimensions. This is compensated by using the mahalanobis distance instead of the euclidean distance. The mahalanobis distance applies an independent linear transform to each of the n-dimensions. This allows it to find clusters with hyper-ellipsoid shapes instead of being restricted to finding clusters that are hyper-spheres. Conceptually, applying an incorrect linear metric distorts a hyper-sphere into a hyper-ellipsoid. If the correct metric for an n-dimensional dataset is linear, then it is absorbed into the mahalanobis distance.

We created an implementation of the UFC-ONP algorithm in C++, and tested it on several datasets. My initial testing of the UFC-ONP algorithm identified the following issues:

- The number of classes needs to be specified.
- If too many classes are specified, the UFC-ONP algorithm can produce spurious results.
- The classification results are sensitive to the initial conditions (class prototypes).

We created an algorithm that mitigates the UFC-ONP limitations, UFC-ONP with pruning (UFC-ONP-wP). The number of classes, K, is varied until the classification results are no longer useful. Useful classification are defined as:

- The K class prototypes are unique.
- The K classes all contain at least M objects. M is a user specified limit. The practical lower limit of M is 1.

Having identified the maximum number of useful classes, the classification process is repeated multiple times (user specified) to identify an optimal classification. Repetition mitigates the initial condition sensitivity. The 'Clustering Balance statistic (Jung et al. 2002) is used to choose the best classification results out of the repeats.

Application of the UFC-ONP-wP application to several test datasets has been promising. Using our simulated point source and extended source datasets, we have confirmed that the classes define regions of similar reliability. Visual inspection of these classification

[1] A n-dimensional mean is a set of mean values. One for each of the n dimensions.

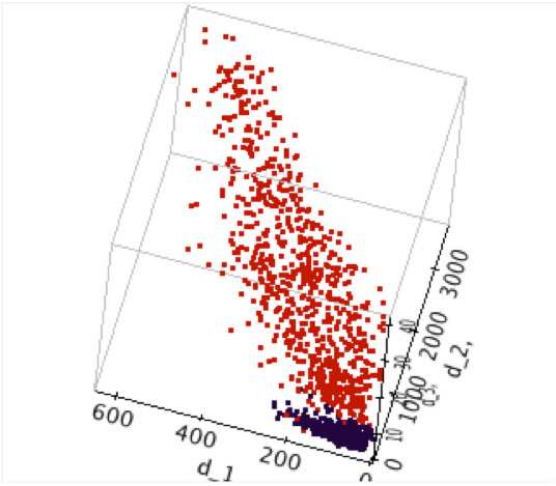


Figure 6: The classification results of the UFC-ONP-wP algorithm (for 2 classes) applied to a CNHI source finder catalogue of the simulated point source dataset.

results shows that they are consistent with the kind of cuts that we would make. As expected these cuts are superior to our cuts though, because they are n-dimensional hypersurfaces. Figure 6 illustrates a particularly good example of the classification results. When the UFC-ONP-wP was forced to identify two classes in a CNHI source finder catalogue of the simulated point source dataset, it identified a group (in red) with 99% reliability.

We also tried testing the UFC-ONP-wP algorithm on HIPASS, HIPASS BGC and Duchamp results for HIPASS datacubes 19, 24 and 511. The classification results correctly identify the HIPASS BGC as a subset of HIPASS. There is also evidence that classifying objects on their pixel co-ordinates may be useful as an RFI detection method.

## 6 Testing reliability based on negative detections with real data

*Task:* Catalogue post-processing

If the source finding is done on both the normal cube and the inverted cube (i.e. searching for negative detections) a reliability value can be given to a detection, assuming the noise is Gaussian and that RFI features that occur in the data are both positive and negative. This method is described in Serra, Jurek & Floeer (2012).

Positive and negative detections can be compared in different combinations of parameter space. In our testing we used the noise-scaled Duchamp catalogue for HIPASS cube 19. We compared the positive and negative detections in two 3D grids, the first one

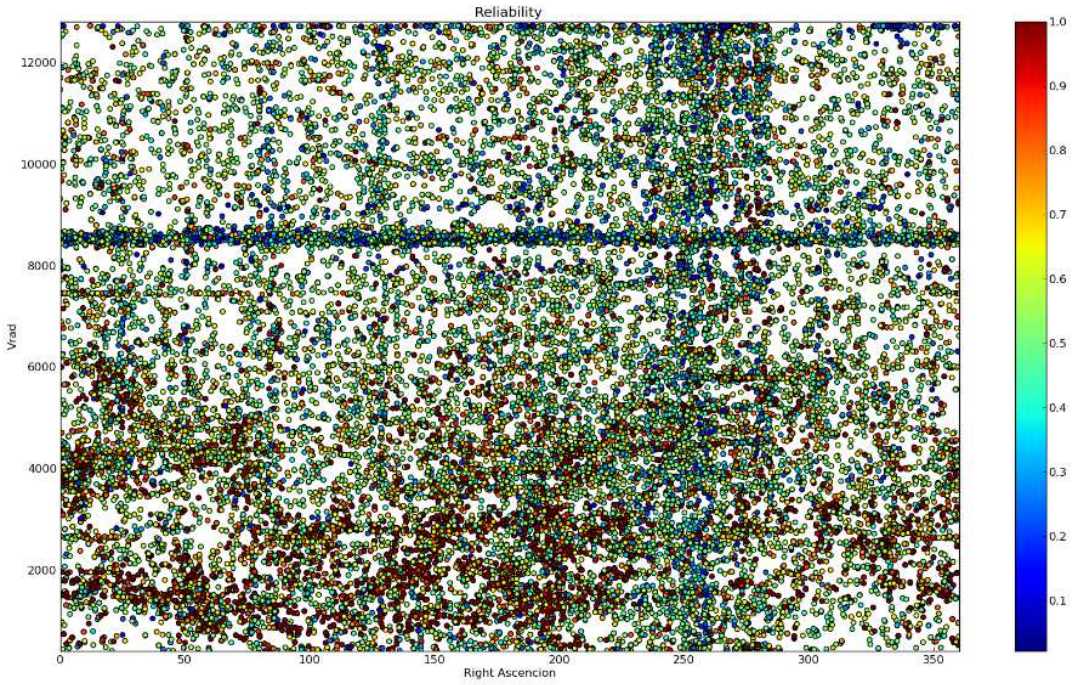


Figure 7: The reliability in the RA-frequency plane of the Duchamp HIPASS cube 19 catalogue after noise scaling.

containing the positions of the source (RA, DEC and Vrad). In the second grid we compared W50, peak Flux and Vrad. These three parameters were chosen as they are the most simple physical parameters of a detection that can be determined relatively easily by any source finder, without large discrepancies. Other useful values such as integrated flux and the total number of pixels are more sensitive to the parameterisation capabilities of a source finder. Note that when this technique is used on the noise scaled cubes as described before, certain values such as integrated flux do not make physical sense.

For each grid-point in the parameter space the reliability can be determined by dividing the positive detections by the number of positive plus negative detections. Hence each detection has a reliability, based on its position in the grid. The reliability of the detection in each grid was multiplied to determine a combined reliability. The results are plotted in Figure 7, where the reliability based on negative detections is plotted as function of Right Ascension and Radial velocity. The points coloured red are most likely true sources, while regions of RFI can be identified by an over density of blue/green sources.