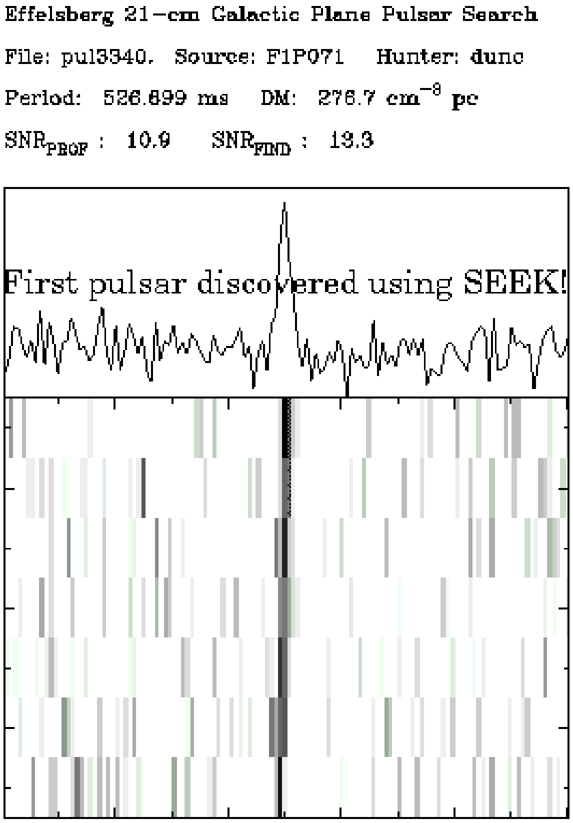
SEEK - Pulsar Search Software

D.R. Lorimer (drl@jb.man.ac.uk)



About SEEK

The SEEK package searches for radio pulsars in noisy data sets. Since its initial incantation in 1997, SEEK has found a total of 30 pulsars in a number of projects with different telescopes and data acquisition systems:

First pulsar discoveries with Effelsberg telescope (Lorimer et al. 2000)
Eleven binary millisecond pulsars in the globular cluster 47 Tucanae (Camilo et al. 2000)
First pulsar discovered with the upgraded Arecibo telescope (Lorimer & Xilouris 2000)
A faint pulsar in the supernova remnant G292.0+1.8 (Camilo et al. 2002a)
Youngest ever radio pulsar in supernova remnant 3C58 with GBT (Camilo et al. 2002b)
A young pulsar in the supernova remnant G54.1+0.3 (Camilo et al. 2002c)
A high-velocity pulsar associated with the Mouse nebula (Camilo et al. 2002d)
Ten pulsars discovered in Arecibo drift searches (Lorimer, McLaughlin et al. 2003, in preparation)

The point of SEEK is simply to find pulsar candidates. Follow-up programs (e.g. folding the data modulo the best pulse period as shown in the above figure) are NOT included here - since these usually exist at the various observatories (e.g. pdm at Parkes). The main programs of SEEK are:

- FIND/HUNT/ACCN: Standard Pulsar Search (FFT+harmonic summing) and Acceleration Searching
- BEST: Sifting through the output from above FFT and/or acceleration searches
- SPEC/MASK: Rudimentary spectrum analysis and interference excision

This guide is meant to be used primarily as a cookbook for standard search reduction, acceleration searches and interference excision.

Installation

STEP 1: The software is currently tested on solaris or linux machines. In order to prepare your computer for installation, you must set the OSTYPE environment variable to either "solaris" or "linux", depending on your system.

STEP 2: Click here to download the latest version of SEEK, or go to the following URL: http://www.jb.man.ac.uk/~drl/seek/seek.tar.gz.

STEP 3: Gunzip the tar file and move it to your source code directory. Unpacking the file will produce a seek subdirectory which you should cd into.

STEP 4: Edit the relevant makefile: "makefile.solaris" for Solaris users or "makefile.linux" for Linuxers. The relevant fields to edit are LPGPLOT - the link to the PGPLOT libraries (ask your system administrator for details) and BIN - the directory where the executables will be placed.

STEP 4: Type "make" to compile and place the executables in your BIN area. You should now be ready to use the programs. Email me (drl@jb.man.ac.uk) if there are any problems with compilation.

STEP 5: Because the program "find" is also a Unix command, you will need to make an alias to the SEEK version when you are using it!

Data formats

The SEEK programs work on dedispersed time series which are written as binary files in a simple format. In order to get your data into this format, you'll either have to adapt your dedispersion code, or use the SIGPROC package which currently handles data from a selection of pulsar machines based at Parkes, Jodrell, Arecibo and GBT. Click here for full details on installing this package and the data format.

In brief, SIGPROC provides a frontend program FILTERBANK to interface with the raw data and produce data in a standard format (which we term filterbank format) ready for dedispersion. For example, let's consider an imaginary raw datafile "rawdata.dat". To get this into filterbank format, type

```
filterbank rawdata.dat > rawdata.fil
```

Once the data are in filterbank format, SIGPROC provides a program DEDISPERSE to convert this into a dedispersed time series. For example:

```
dedisperse rawdata.fil -d 25.6 > rawdata.tim
```

will produce a time series (rawdata.tim) dedispersed for DM=25.6. These data are now in the correct format for SEEK to work on.

Cookbook

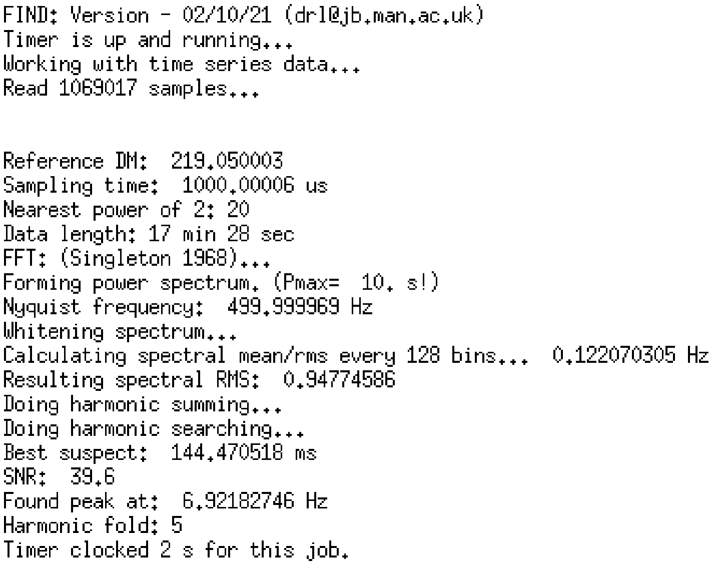
The following recipes show how to use SEEK by running the programs on example time series. These are binary files which can be downloaded for Big Endian Machines (e.g. Solaris) or Little Endian Machines (e.g. Linux). These data were produced using SIGPROC described schematically in the previous section. Click for further details about SIGPROC.

Search of single-DM time series

Finding a known pulsar in a time series that has been dedispersed at the nominal DM is usually the first thing you’d want to check out before searching other data sets. The example time series file 1930+22.tim is a 17.8-min observation using the 76-m Lovell telescope on the 144-ms pulsar B1930+22. To search it for periodicities, type:

% find 1930+22.tim

which will run the program find resulting in the following screen output:



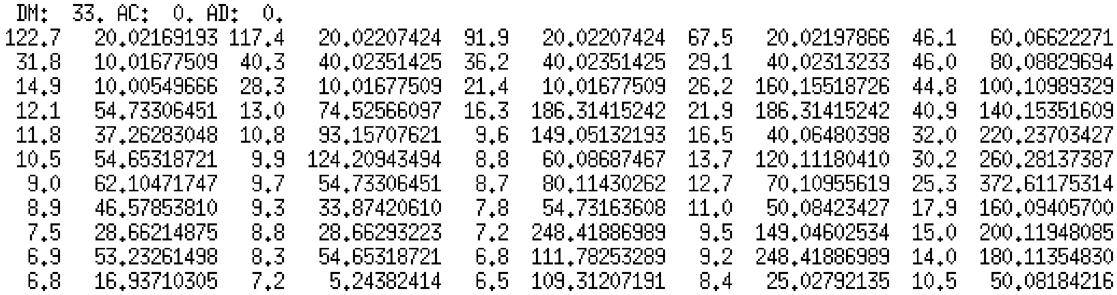
```
FIND: Version - 02/10/21 (drl@jb.man.ac.uk)
Timer is up and running...
Working with time series data...
Read 1069017 samples...


Reference DM:  219.050003
Sampling time:  1000.00006 us
Nearest power of 2: 20
Data length: 17 min 28 sec
FFT: (Singleton 1968)...
Forming power spectrum. (Pmax=  10. s!)
Nyquist frequency:  499.999969 Hz
Whitening spectrum...
Calculating spectral mean/rms every 128 bins...  0.122070305 Hz
Resulting spectral RMS:  0.94774586
Doing harmonic summing...
Doing harmonic searching...
Best suspect:  144.470518 ms
SNR:  39.6
Found peak at:  6.92182746 Hz
Harmonic fold: 5
Timer clocked 2 s for this job.
```

which shows a clear detection of the pulsar with a signal-to-noise ratio (S/N) of about 40. For this system (frequency 1400 MHz, bandwidth 64 MHz, system equivalent flux density ~35 Jy), the observed S/N is in good agreement with that expected from B1930+22 (a 1.2 mJy pulsar with a duty cycle of 8%).

The above output to the screen is just the candidate from the search with the highest S/N. In the above example, where the pulsar is clearly detected, this output is satisfactory. Other pulsars may not be the strongest signal in the search, as in the following 9-min observation of the 372-ms pulsar B2022+50. Download the time series 2022+50.tim and search it in the usual way:

```
% find 2022+50.tim
```

FIND will report a 20-ms signal with S/N~123. This is, of course, the mains frequency (50 Hz) power line. To dig deeper into the candidate list requires examination of the output files from find. The main file of interest is the complete list of S/Ns and periods for each harmonic fold: the so-called "period file". In this example, the period file produced by find is "2022+50.prd". The first few lines of the file look like this:



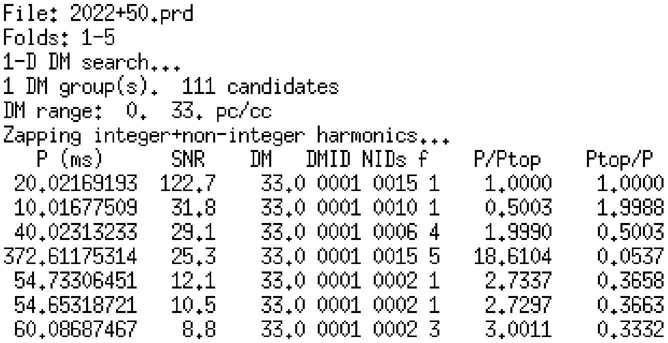
```
 DM:  33. AC:  0. AD:  0.
122.7   20.02169193 117.4   20.02207424  91.9   20.02207424  67.5   20.02197866  46.1   60.06622271
 31.8   10.01677509  40.3   40.02351425  36.2   40.02351425  29.1   40.02313233  46.0   80.08829694
 14.9   10.00549666  28.3   10.01677509  21.4   10.01677509  26.2  160.15518726  44.8  100.10989329
 12.1   54.73306451  13.0   74.52566097  16.3  186.31415242  21.9  186.31415242  40.9  140.15351609
 11.8   37.26283048  10.8   93.15707621   9.6  149.05132193  16.5   40.06480398  32.0  220.23703427
 10.5   54.65318721   9.9  124.20943494   8.8   60.08687467  13.7  120.11180410  30.2  260.28137387
  9.0   62.10471747   9.7   54.73306451   8.7   80.11430262  12.7   70.10955619  25.3  372.61175314
  8.9   46.57853810   9.3   33.87420610   7.8   54.73163608  11.0   50.08423427  17.9  160.09405700
  7.5   28.66214875   8.8   28.66293223   7.2  248.41886989   9.5  149.04602534  15.0  200.11948085
  6.9   53.23261498   8.3   54.65318721   6.8  111.78253289   9.2  248.41886989  14.0  180.11354830
  6.8   16.93710305   7.2    5.24382414   6.5  109.31207191   8.4   25.02792135  10.5   50.08184216
```

from left to right, the 10 columns are in pairs of S/N and period (ms) for harmonic folds 1-5. In this notation, fold 1 refers to the raw spectrum, fold 2 is the spectrum with the one harmonic stretch and add, fold 3 is a further stretch and add, and so on. Fold 5 then refers to addition of 16 harmonics. An inspection of this list shows the pulsar as the 372.6-ms signal in fold 5.

Obviously one does not want to look at period files like this one very often! What is required is a program to sift through the list and produce a summary of the best candidates. This is what the program BEST does:

```
% best 2022+50.prd
```

scans the file, sorts out harmonic relationships and produces the following output:



```
File: 2022+50.prd
Folds: 1-5
1-D DM search...
1 DM group(s).  111 candidates
DM range:  0.  33. pc/cc
Zapping integer+non-integer harmonics...
   P (ms)      SNR    DM   DMID NIDs f    P/Ptop    Ptop/P
 20.02169193  122.7    33.0 0001 0015 1    1.0000    1.0000
 10.01677509   31.8    33.0 0001 0010 1    0.5003    1.9988
 40.02313233   29.1    33.0 0001 0006 4    1.9990    0.5003
372.61175314   25.3    33.0 0001 0015 5   18.6104    0.0537
 54.73306451   12.1    33.0 0001 0002 1    2.7337    0.3658
 54.65318721   10.5    33.0 0001 0002 1    2.7297    0.3663
 60.08687467    8.8    33.0 0001 0002 3    3.0011    0.3332
```

Listed here are the period, S/N, DM of each candidate. The other columns list DM index (DMID; there is only one DM in this case), the number of identifications of this period including harmonics (NIDs), the harmonic fold that the candidate peaked in (f). The P/Ptop column lists ratio the period of this candidate to the period of the best candidate. Ptop/P is the inverse of this.

Search over range of DM values

Given a datafile in filterbank format, the script HUNT will dedisperse and run FIND on the data over a range of DM values. To use this, create an ASCII file "dmlist" with the DM values you wish to search and then run HUNT. As an example, try downloading the Arecibo drift-scan data file aobeam.fil - a beam containing a recently discovered 5.79-ms pulsar with DM~14. To create simple dmlists, the program STEP is available. For example:

```
% step 10 20 0.2 > dmlist
```

would create a list in the DM range 10-20 in steps of 0.2. To search this file for periodicities, run HUNT on the filestem:

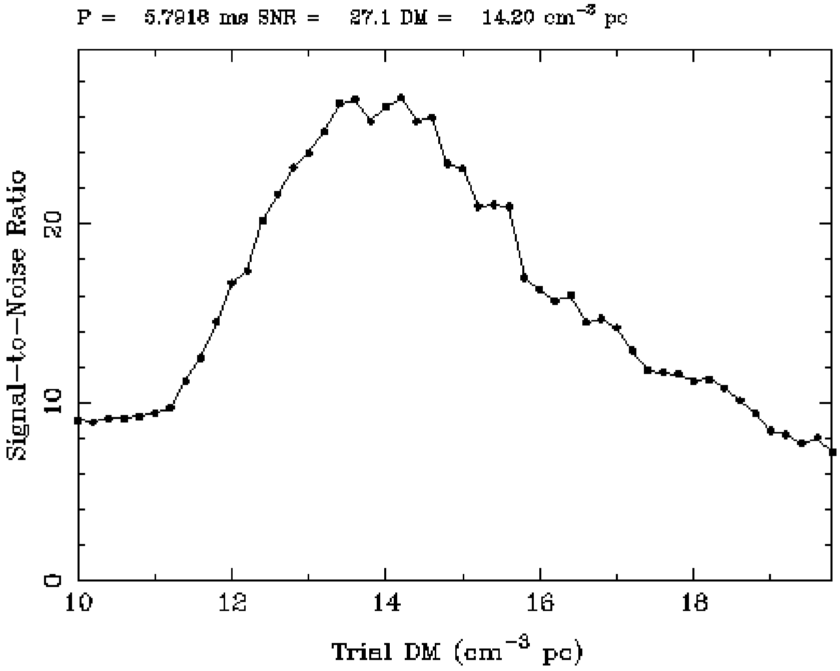
```
% hunt aobeam
```

when the search is complete, the script runs BEST on the file aobeam.prd to search for the best suspects. You should see the pulsar with S/N~27 in fold f=3.

A very useful diagnostic is the plot of S/N versus trial DM. To produce this plot, run best in the viewing mode:

```
% best aobeam.prd -v
```

which will cycle through the candidates displaying each plot in turn. The plot for the pulsar in this example should look like this:



Acceleration Search for a given DM

One of the first uses of SEEK was to perform coherent acceleration searches on dedispersed time series at a fixed DM. This application is tuned to globular clusters where the detection of the first pulsar limits the search so that the computing power can be applied to searching in acceleration (AC) space, rather than DM space. Our ongoing search of 47 Tuc with SEEK in this manner has so far turned up 11 binary pulsars.

Given a dedispersed time series, the acceleration search is controlled by the ACCN script which works much in the same way as the HUNT script described above. Instead of the "dmlist" file, you now need to supply an "aclist" file with the range of acceleration values (in m s^{-2}) to search over. In addition, if you have time you can also search in acceleration derivative (AD) by creating a file "adlist" with the range of values (in cm s^{-3}) to search.

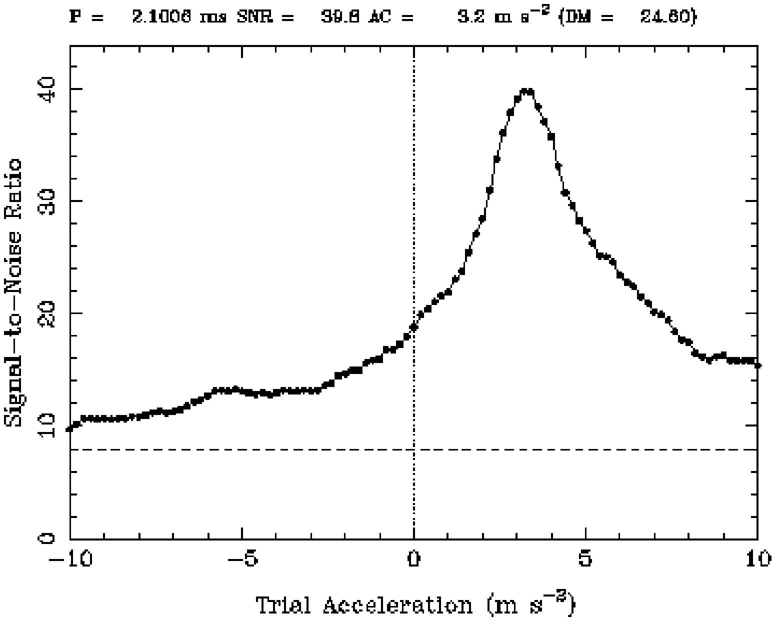
Consider for now a 1-D acceleration search (AD=0 throughout). Download the example 18-min Parkes observation of 47 Tuc (47tuc.tim) and run ACCN for a range of AC values between -10 m s^{-2} and +10 m s^{-2} with a step size of 0.2 m s^{-2}:

```
% step -10 10 .2 > aclist
% accn 47t
```

on completion, running BEST in interactive mode you should see a clear detection of the 2.1-ms pulsar 47TucJ at a best acceleration of 3.2 m s^{-2}, as well as the 4.34-ms pulsar 47TucU at an acceleration of -3.8 m s^{-2}. Running best interactively as before with the -v option

```
% best 47t.prd -v
```

will show the variation of S/N in acceleration space. For J this looks like this:



Interference Excision

Radio Frequency Interference (RFI) can produce a lot of spurious pulsar candidates seen as spikes in the amplitude spectrum. In extreme cases, this can bury potentially real pulsar candidates in a sea of RFI making them difficult to pick out. A simple technique to remove them is to censor certain bins in the spectrum so that they do not appear in the candidate lists. SEEK provides two methods for doing this: zap tables and spectral masks.

Zap Tables

A zap table is a list of frequency ranges and harmonics to be censored in the spectrum. One common entry in such a table is the mains power line and its harmonics visible most prominently in the DM=0 time series. As an example, download the Arecibo time series 0.tim. Running find on the data shows up a very strong signal from the mains (60 Hz at Arecibo) which would be good to zap. Before zapping, it is a good idea to examine the amplitude spectrum to see how many harmonics are present. To do this you will first need to produce binary spectrum files using the -s option in FIND:

```
% find 0.tim -s
```

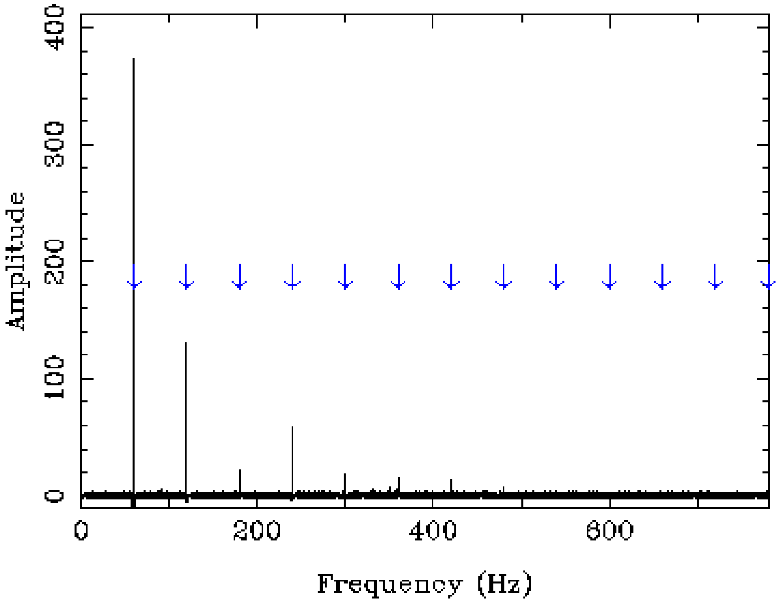
which will produces files fold0.spc, fold1.spc .... fold5.spc corresponding to the raw spectrum (fold0.spc), the spectrum after normalization (fold1.spc) and the spectrum after each of the four stages of harmonic summing (fold2-5.spc). To view the spectra, use the SPEC program. For example, to look at fold1.spc, you need to type:

```
% spec fold1.spc
```

which will produce a plot from 0 Hz up to the Nyquist frequency (~780 Hz in this case). To help identify spectral features, there is a -F option in SPEC which will display the fundamental and harmonics of a given frequency. So, to mark the 60-Hz line, type

```
% spec fold1.spc -F60
```

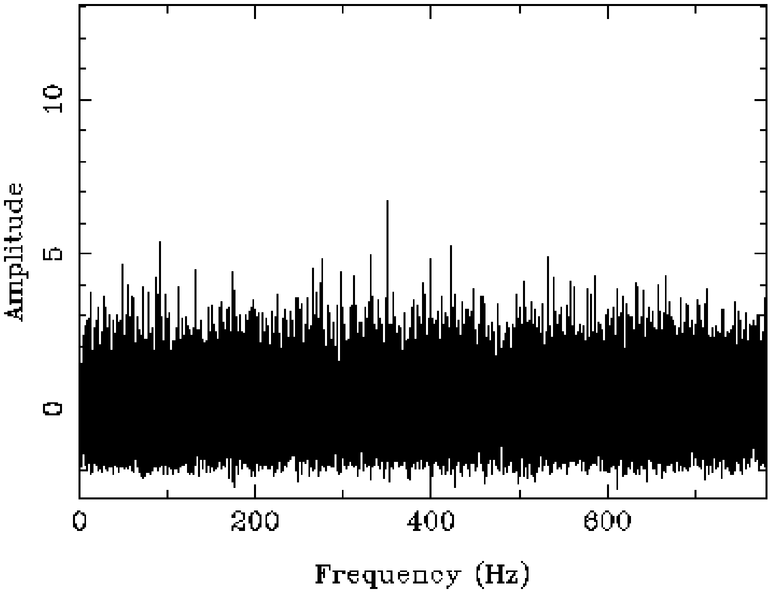
which should look like this:



which shows the fundamental plus about 7 harmonics. You can zap these lines by creating an ASCII file called "birdies" with the frequency ranges and total number of harmonics (including fundamental) to zap. So to zap these 8 lines, try creating a birdies file with the following line:

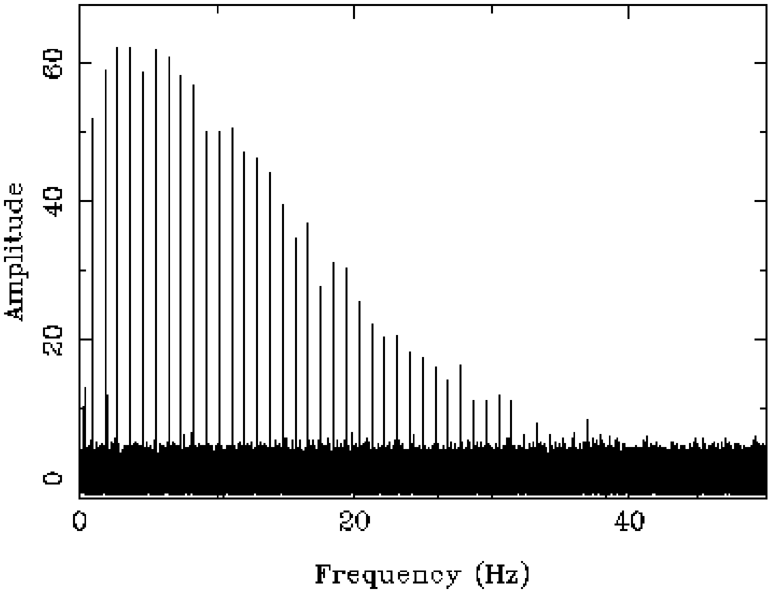
```
59.9 60.1 8
```

and now run FIND with the zap (-z) option to read the file. Looking at fold1.spc as before you should see a much cleaner spectrum like this.



Spectral Mask

Sometimes it is helpful to censor specific spectral bins of interest. For example we might want to mask the following spectrum containing many harmonics from the strong pulsar B1550-54:



To mask these harmonics, which are all predominantly above S/N=10, use the mask command:

```
% mask fold1.spc -t10
```

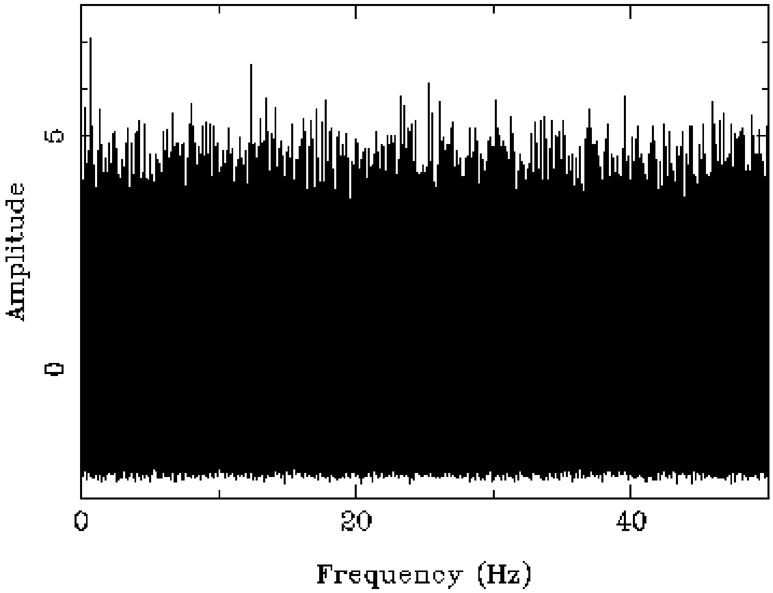
to display the spectrum after clipping these bins. To produce a mask file, add the -f option:

```
% mask fold1.spc -t10 -f
```

which will write out an ASCII file "mask.out" which is of the format

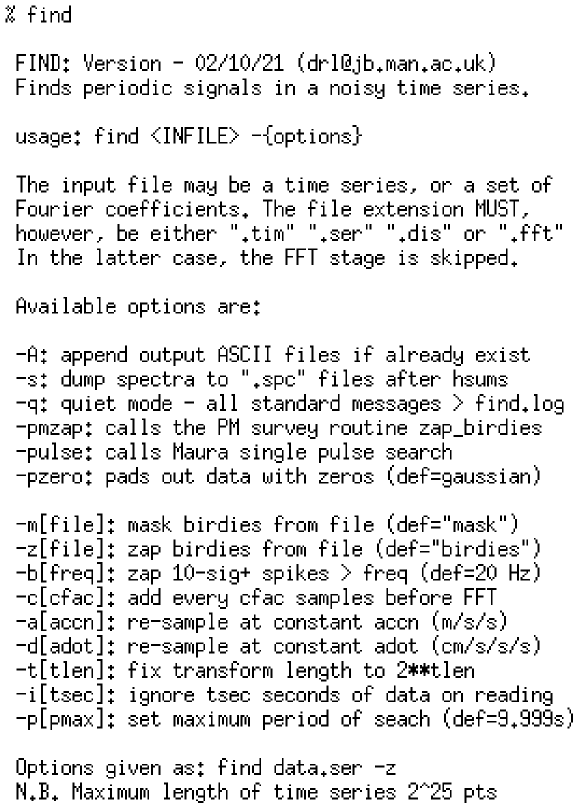
```
bin_number frequency_hz S/N
```

scan through this file and make sure that just the spectral frequencies of interest are being masked (you may need to delete unrelated features by hand!). When you are happy with the file, rename "mask.out" to simply "mask" and run FIND with the -m option. The resultant output is:



Miscellaneous Features

I will add some "advanced" and "obscure" features of the SEEK programs here when time permits. In the meantime, some help can be obtained by just typing the name of the program you are interested in, for example:



```
% find

 FIND: Version - 02/10/21 (drl@jb.man.ac.uk)
 Finds periodic signals in a noisy time series.

 usage: find <INFILE> -{options}

 The input file may be a time series, or a set of
 Fourier coefficients. The file extension MUST,
 however, be either ".tim" ".ser" ".dis" or ".fft"
 In the latter case, the FFT stage is skipped.

 Available options are:

 -A: append output ASCII files if already exist
 -s: dump spectra to ".spc" files after hsums
 -q: quiet mode - all standard messages > find.log
 -pmzap: calls the PM survey routine zap_birdies
 -pulse: calls Maura single pulse search
 -pzero: pads out data with zeros (def=gaussian)

 -m[file]: mask birdies from file (def="mask")
 -z[file]: zap birdies from file (def="birdies")
 -b[freq]: zap 10-sig+ spikes > freq (def=20 Hz)
 -c[cfac]: add every cfac samples before FFT
 -a[accn]: re-sample at constant accn (m/s/s)
 -d[adot]: re-sample at constant adot (cm/s/s/s)
 -t[tlen]: fix transform length to 2**tlen
 -i[tsec]: ignore tsec seconds of data on reading
 -p[pmax]: set maximum period of seach (def=9.999s)

 Options given as: find data.ser -z
 N.B. Maximum length of time series 2^25 pts
```

Closing Remarks

SEEK is by no means the be all and end all of pulsar search packages, and should be thought of as a first attempt at making pulsar searcher’s equivalent of TEMPO: i.e. a general use package that can be applied to data from different telescopes. Current plans are towards rewriting the package in C, with a view to improving some of the algorithms and making better use of memory, and merging it into a larger effort of general pulsar data analysis (SIGPROC).

Several people have contributed to the software development so far, particular thanks are due to Michael Kramer, Fernando Camilo, Maura McLaughlin, Ingrid Stairs and Paulo Freire who have provided code and/or comments on SEEK. Please let me know your comments (good and bad!) on your SEEK experiences, I will try to incorporate these into future versions. I’d also be very pleased to hear of any new pulsars you discover with SEEK... happy hunting!