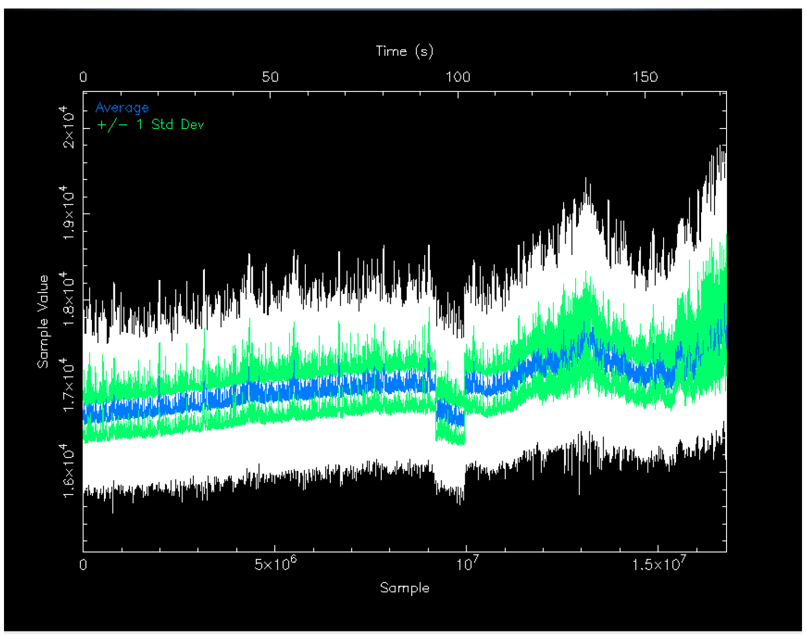
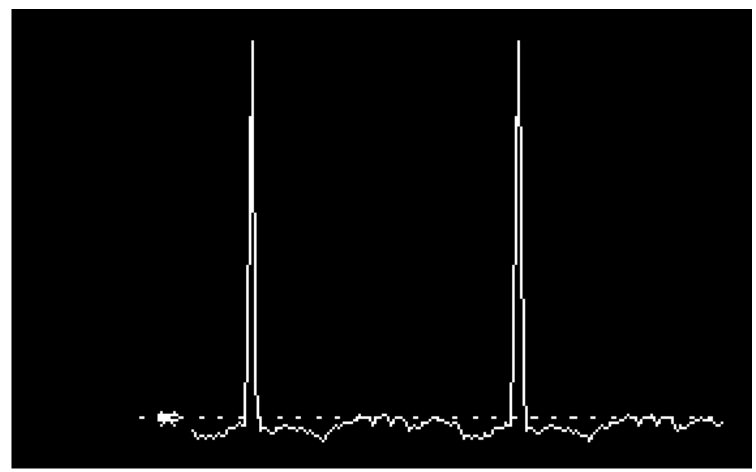
Time Domain Methods II: (Mostly) periodic signals

Ingrid Stairs, UBC
University of Arizona PIRE
Feb. 16, 2021





Sometimes periodic signals jump out at you!

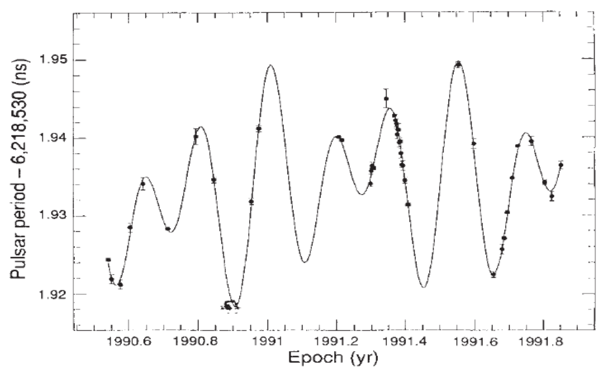
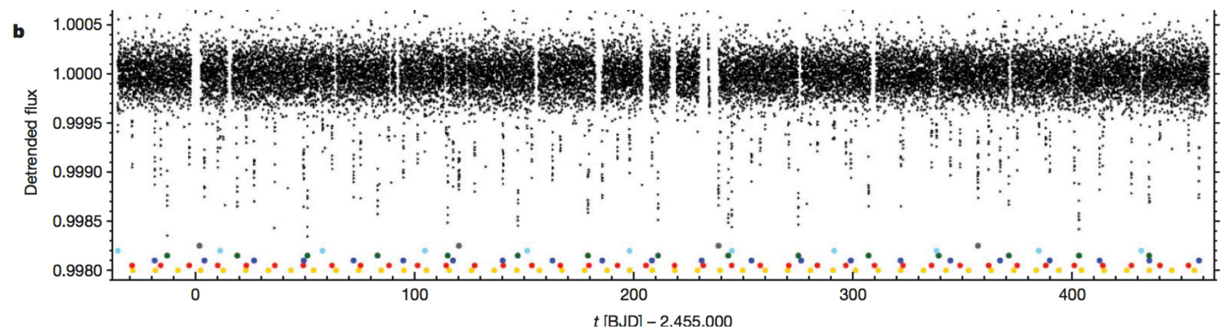


FIG. 3 Period variations of PSR1257+12. Each period measurement is based on observations made on at least two consecutive days. The solid line denotes changes in period predicted by a two-planet model of the 1257+12 system.

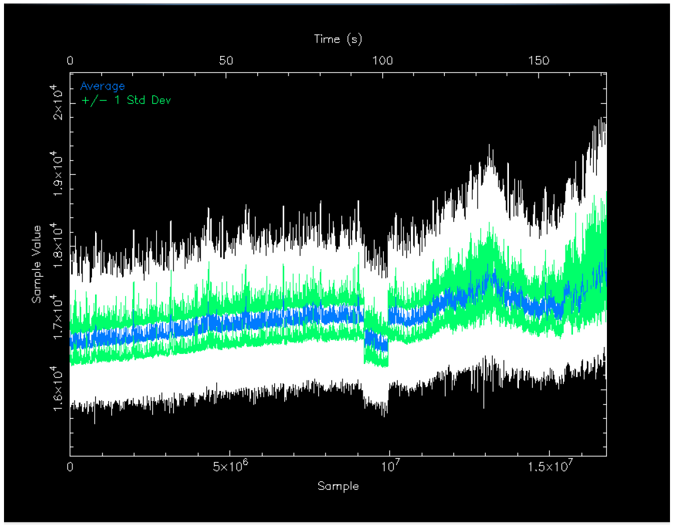


Transit data from Kepler-11

PSR B1257+12
Wolszczan & Frail, Nature 355, 145 (1992)

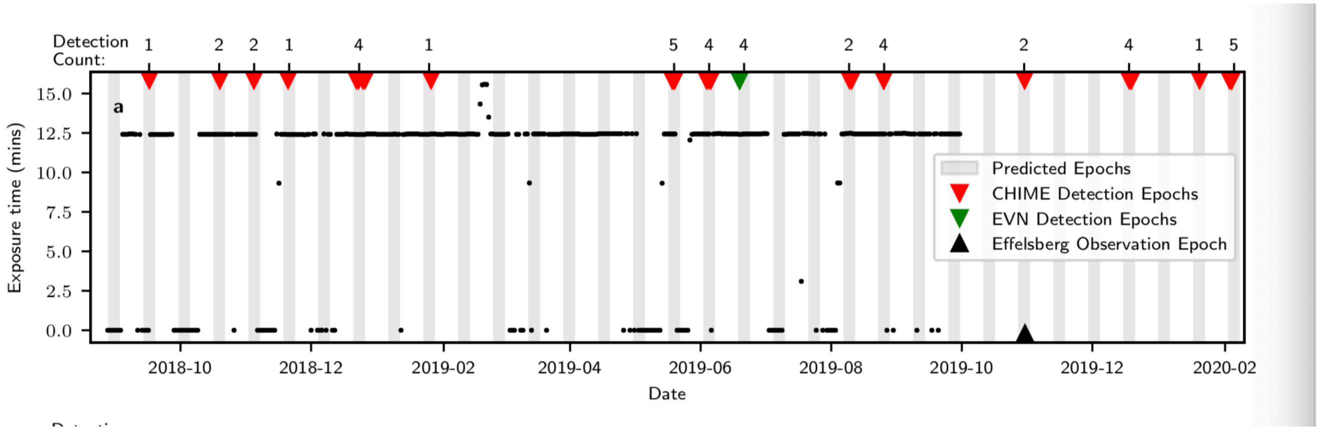
Lissauer et al., Nature **470**, 53 (2011).

But more often we are faced with data like this:



GBT data on PSR B1828-11

Or this:



CHIME/FRB Collaboration, Nature **582**, 351 (2020)

Outline of talk:

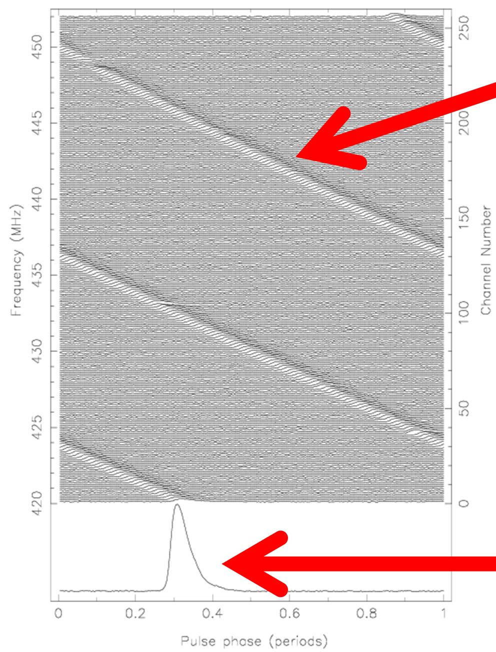
1. Searching for radio pulsars
    - Dispersion
    - Periodicity
    - Acceleration

3. Searching in sparsely-sampled data
    - Epoch-folding, Bayesian methods, high-energy pulsars
    - CLEAN, Lomb-Scargle
    - Repeating Fast Radio Bursts

4. Timing Pulsars
    - Template matching
    - Template construction

What do we have to look for when trying to find pulsars?

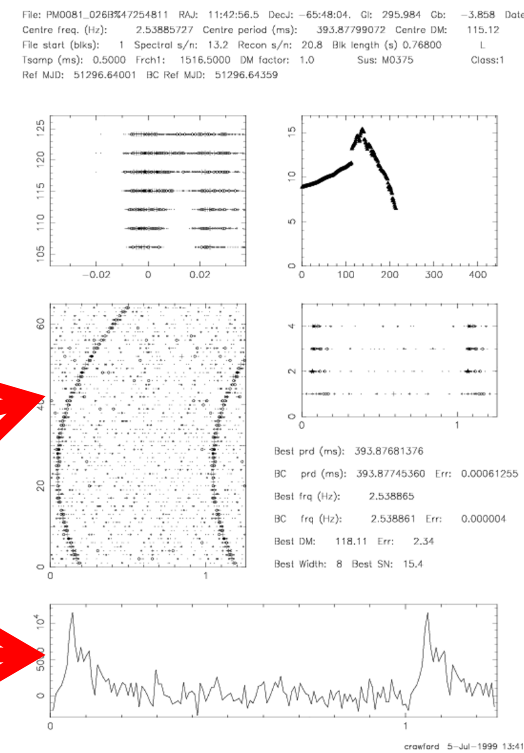
Pulsar search parameters:



Dispersion

Acceleration

Periodicity

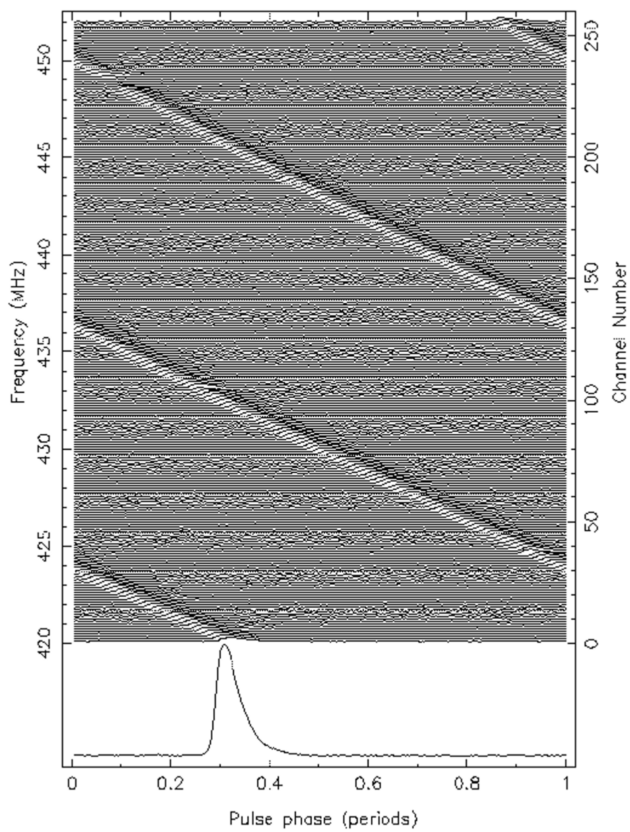


This takes a few CPU cycles...

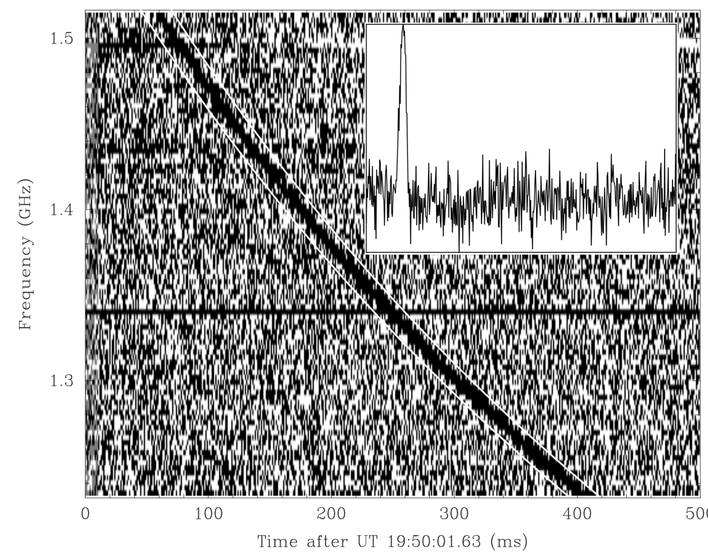
Dispersion: due to partially ionized gas in the interstellar medium. The pulse is delayed from infinite frequency by

$$t = \frac{\mathrm{DM}}{2.41 \times 10^{-4}} \frac{1}{f^2}$$

for t in seconds, f in MHz, and DM in pc cm^{-3}.



Vela pulsar at Parkes



Lorimer et al., 2007, Science 318, 777.

For data from a radio telescope that can be searched for pulsars:

- Wide bandwidth is divided into many small "filterbank" frequency channels with width $\Delta\nu$
- Filterbank data streams are then "detected" → total power and
- Rapidly sampled, at rate Δt (typically tens of μsec)
- Observation time T seconds (typically hundreds)

This quickly leads to PB of data.

What frequencies are we sensitive to?

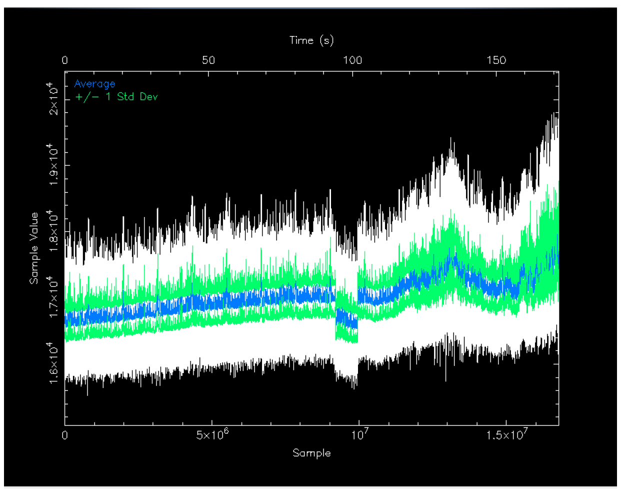
The total number of samples is $T/\Delta t = N$, and it's real data, not complex.

If we take a Fourier Transform, we'll have N/2 frequency bins, each with amplitude and phase.

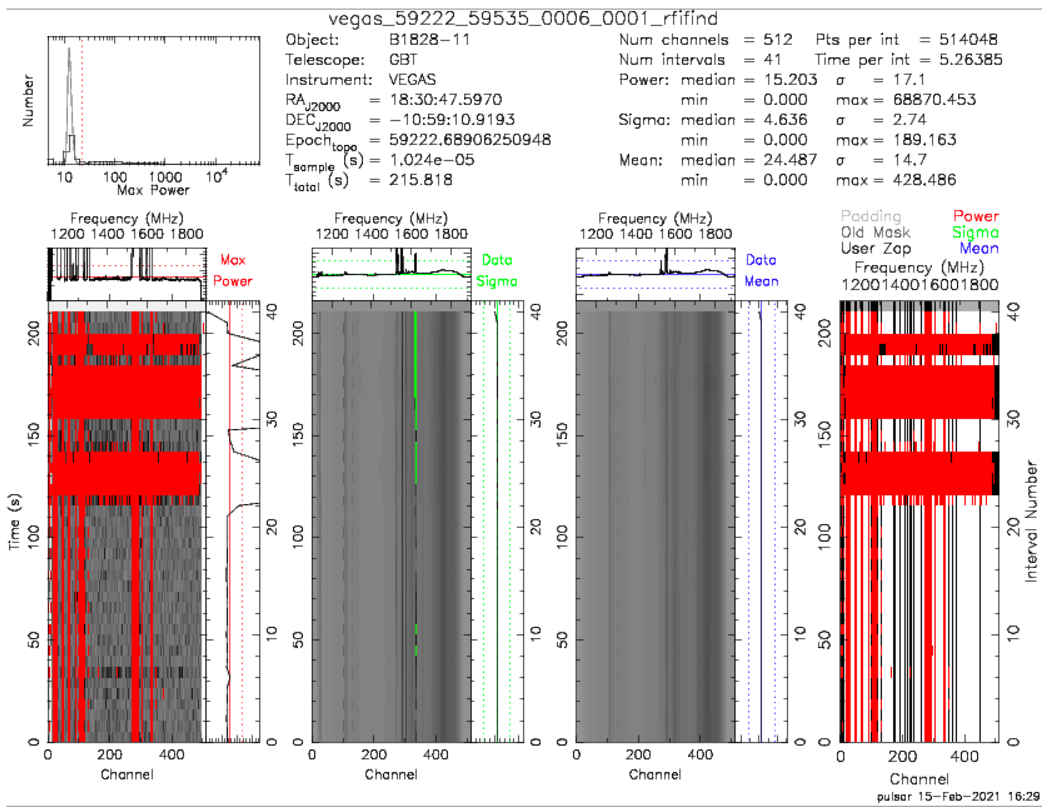
Each frequency bin will have width 1/T, making the highest frequency accessible $N/(2T) = 1/(2\,\Delta t)$. This is the **Nyquist frequency.**

We have data…. Let's start checking different Dispersion Measures (DMs) and periods!

Not so fast! First we need to get rid of Radio Frequency Interference (RFI). RFI can be impulsive, periodic, broadband, narrowband…. The challenge is to identify and zap it without zapping interesting signals!



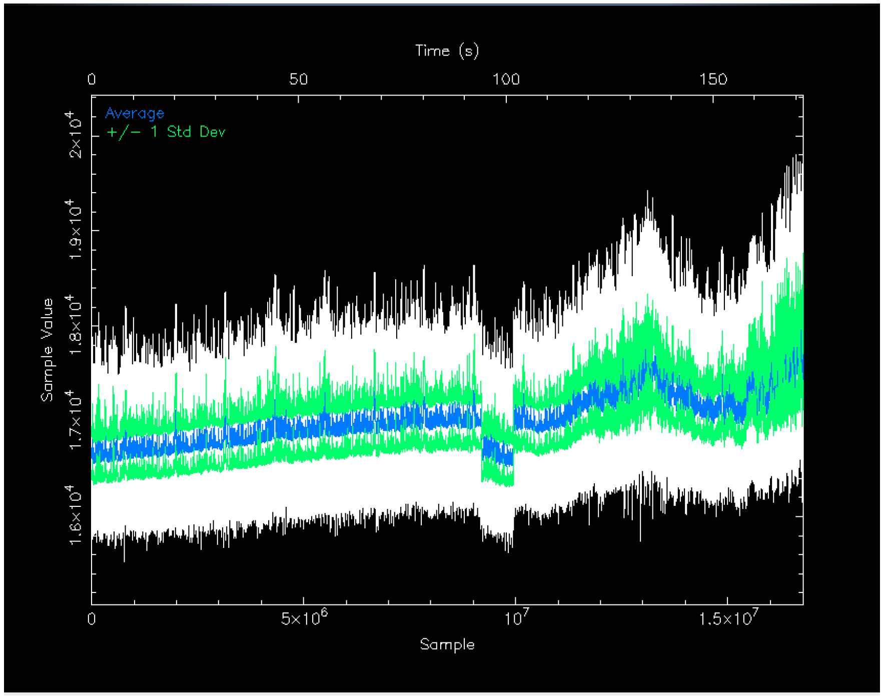
Rfifind routine from presto code:
https://github.com/scottransom/presto

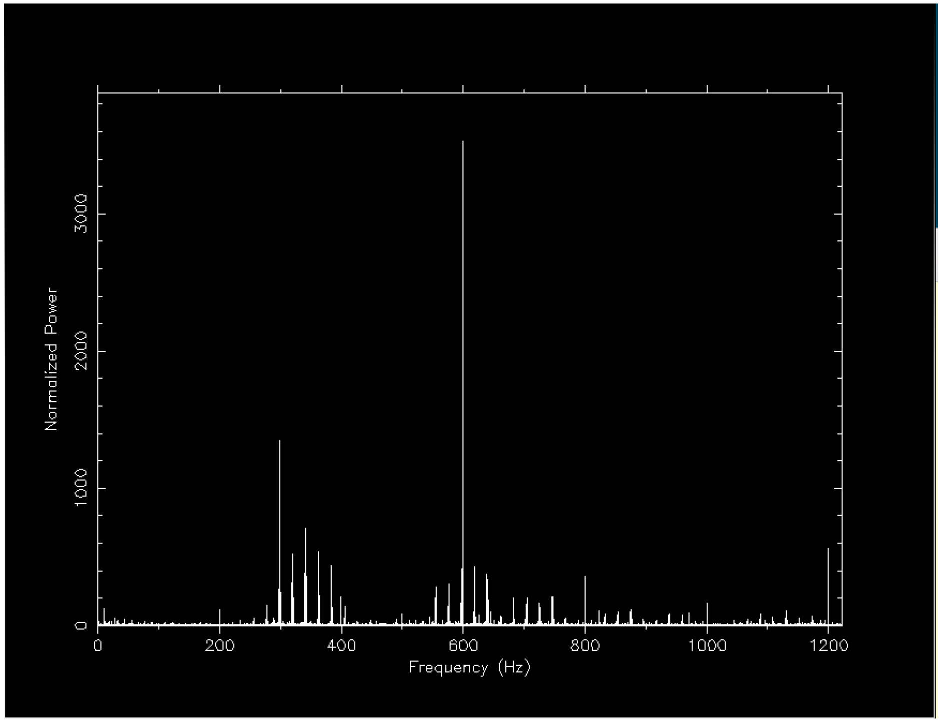


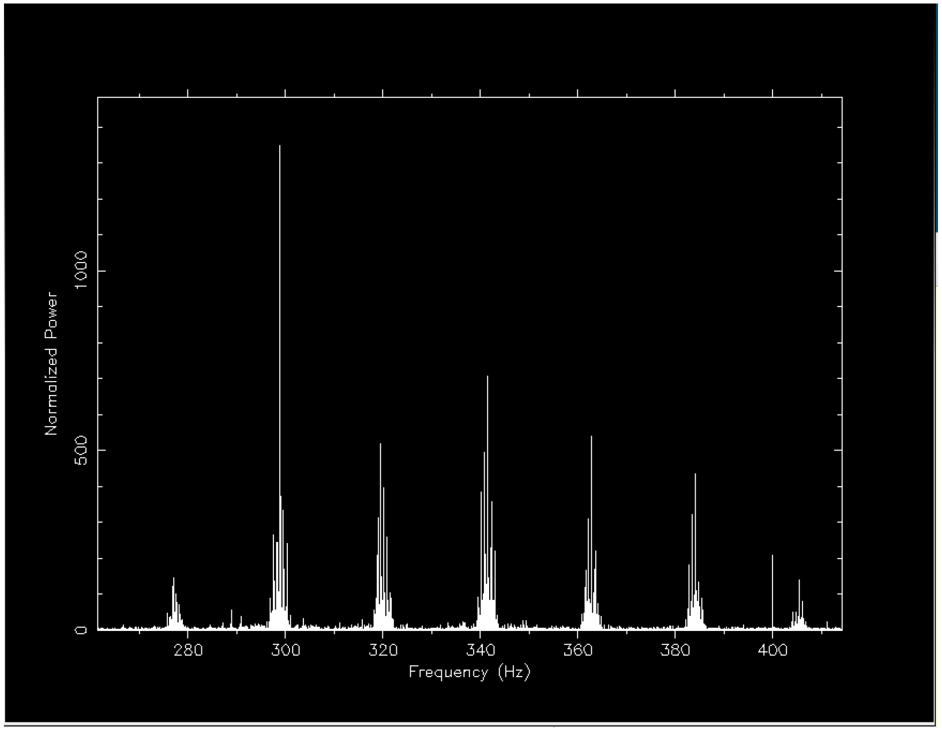
Mask

Does zapping RFI help?

Let's take an FFT of the unmasked data, collapsed to DM=0.

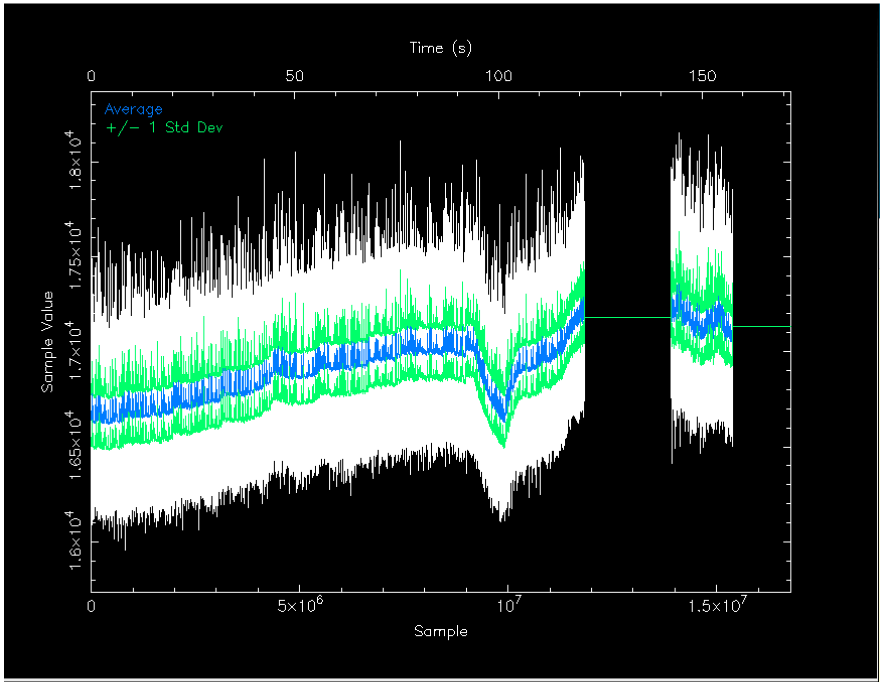


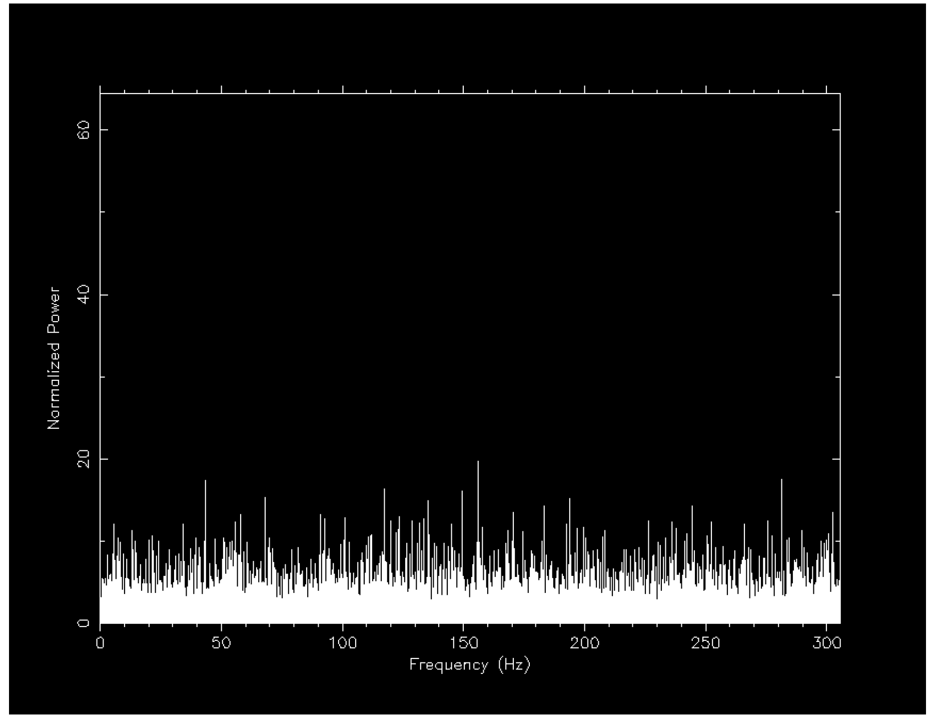


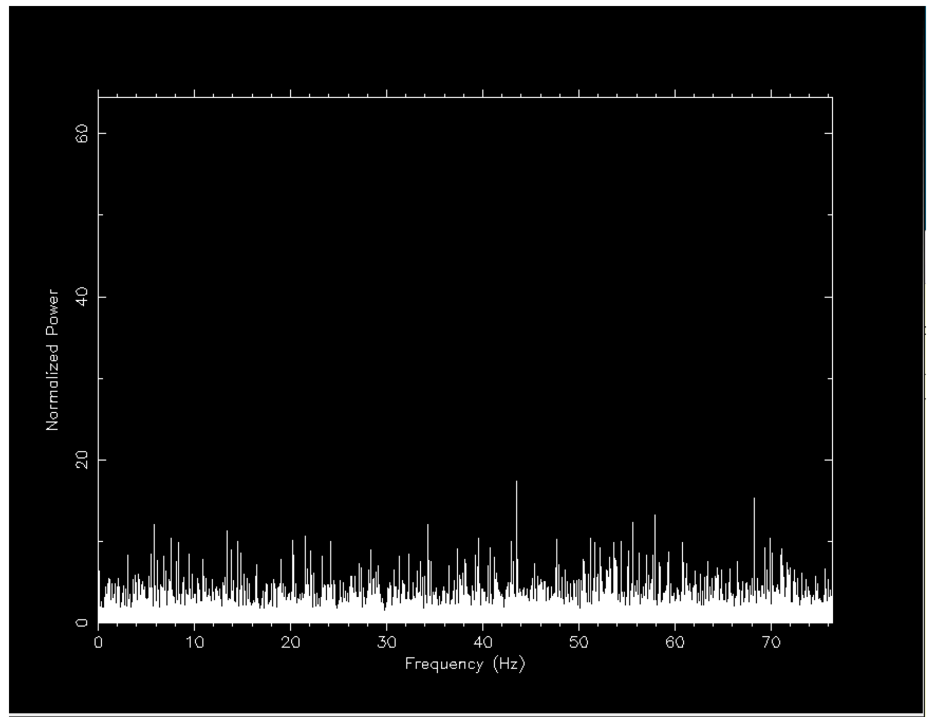


Does zapping RFI help?

Now an FFT of the masked data, collapsed to DM=0.







The first scientific step in any pulsar searching is forming frequency-collapsed time series for each of many possible **trial DMs**.

The Taylor tree algorithm (Astron. Astrophys. Suppl. **15**, 367 (1974)) is still the classic, most-used algorithm, making the process NlogN instead of N^2.

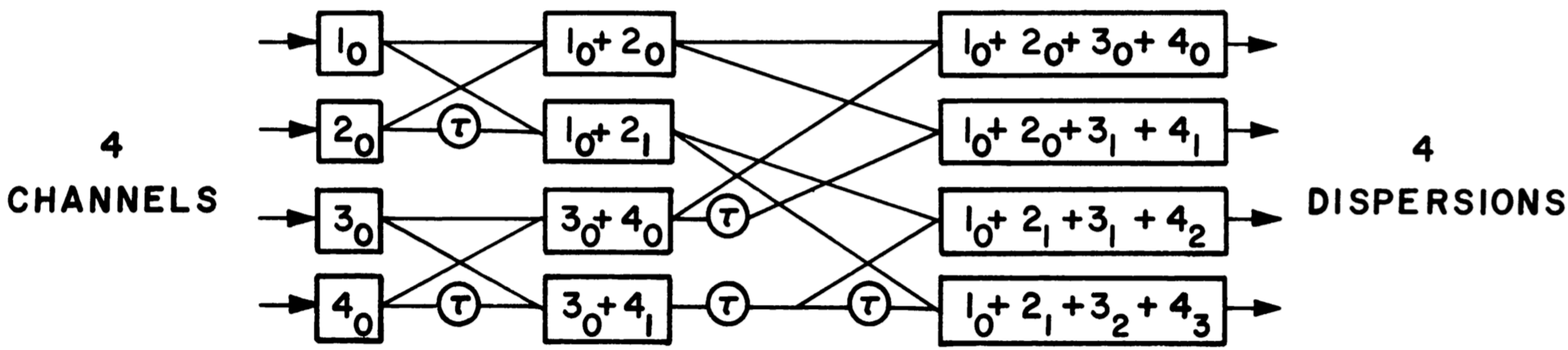
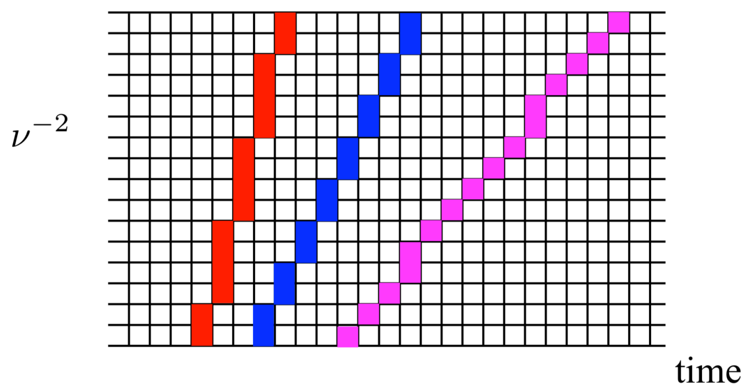
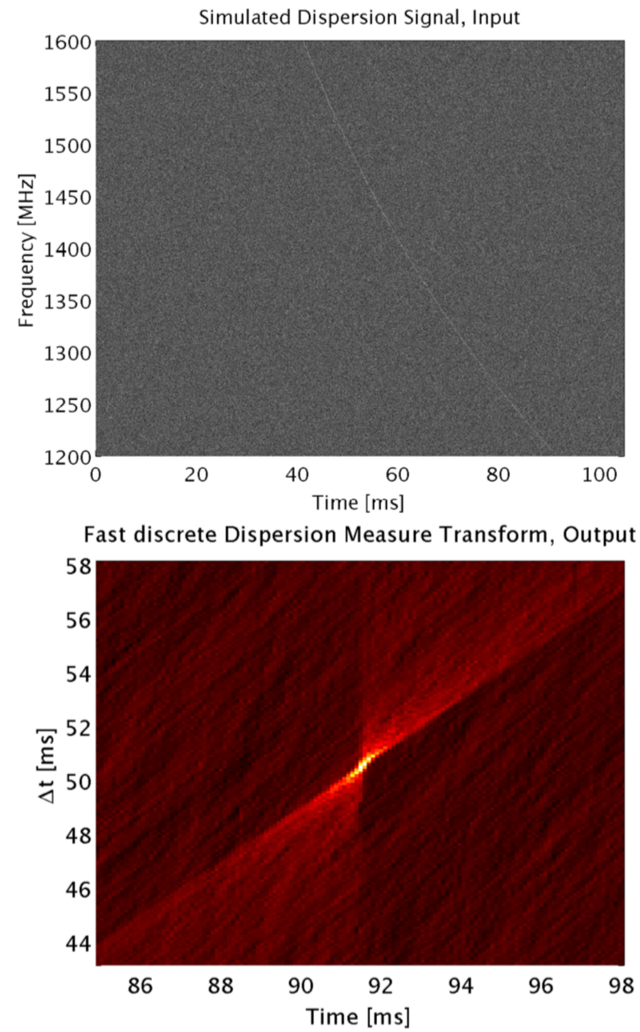


Figure 1 Block diagram of a 4 channel digital dispersion filter. Detected signals from a 4 channel receiver are input at the left; de-dispersed output signals are taken from the right. Rectangles represent summations, and circles represent unit delays. The indicated operations are performed from right to left.

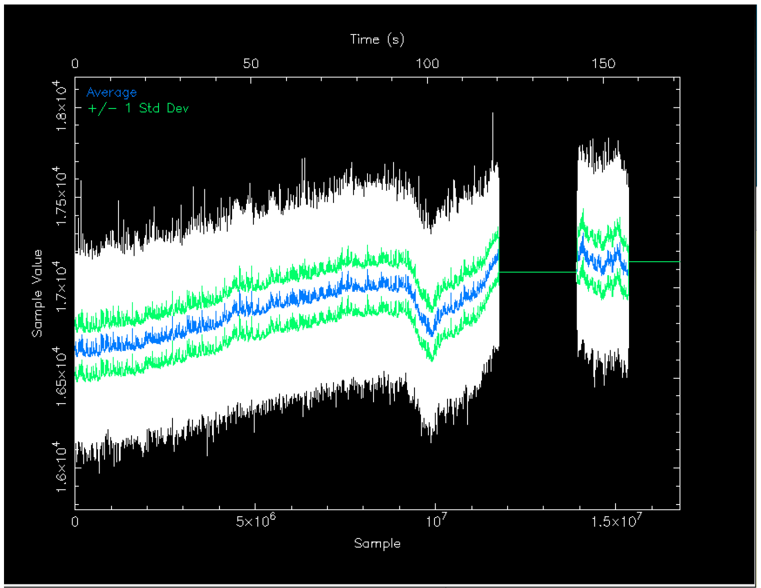


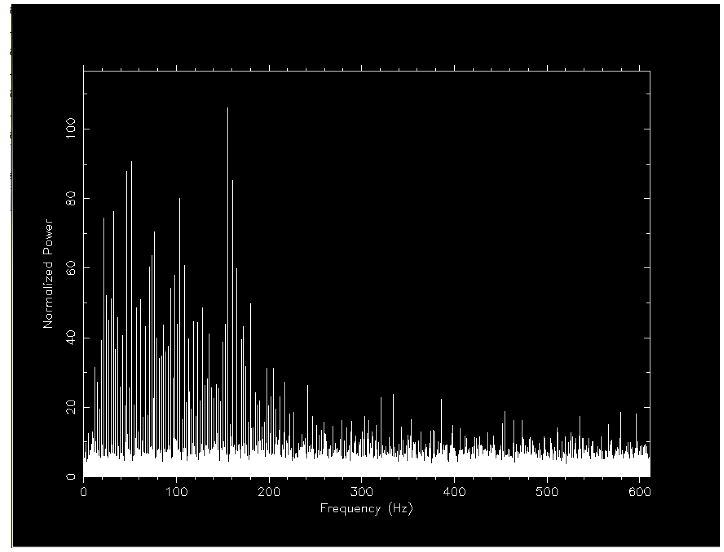
We even still use a version of the tree algorithm (bonsai) for CHIME Fast Radio Burst searching (image: K. Smith).

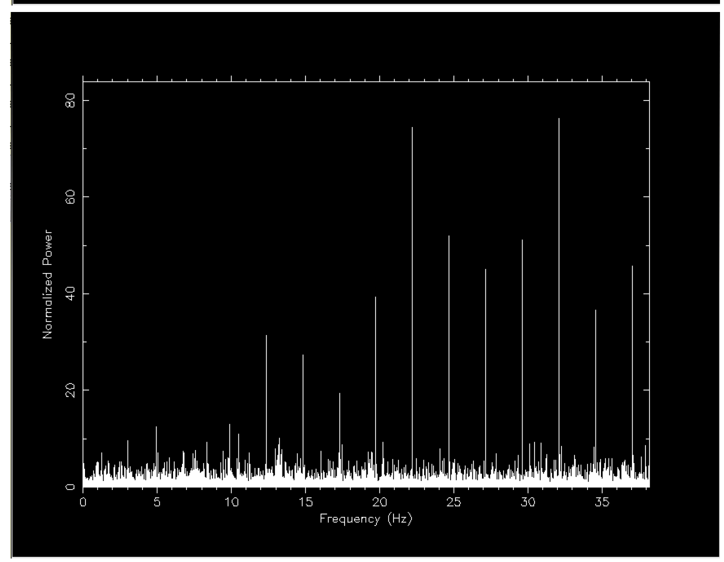


But there is competition now from the FDMT algorithm (Zackay & Ofek, ApJ **835**, 11 (2017)

Now we can go through the DM trial time series, and check each one for **periodicities**. Let's take an FFT of one time series at DM of 157 pc cm^{-3} (dereddening is also usually necessary):



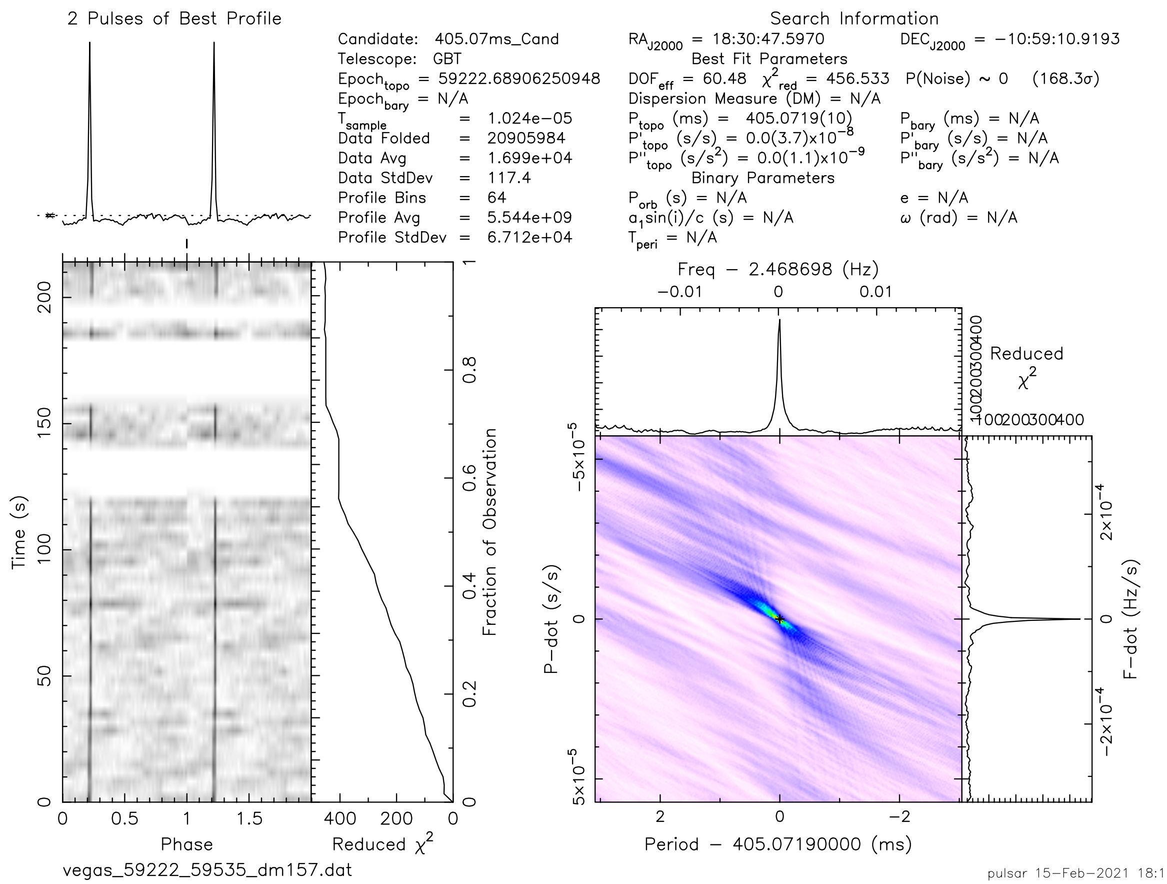




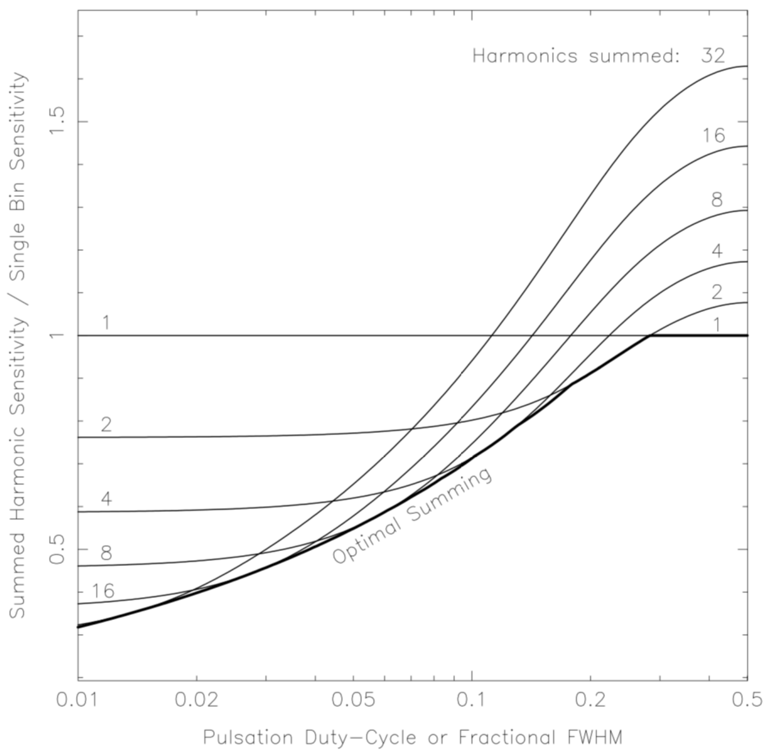
Lots of signals above the noise – and a spacing of about 2.5 Hz. These are harmonics of the pulsar's spin frequency.

With an interesting periodiocity identified at around 2.5 Hz, we can “fold” the dedispersed data on itself at that frequency and refine it to see what the actual pulse profile looks like:

The pulse is very narrow, as expected from the existence of so many harmonics.



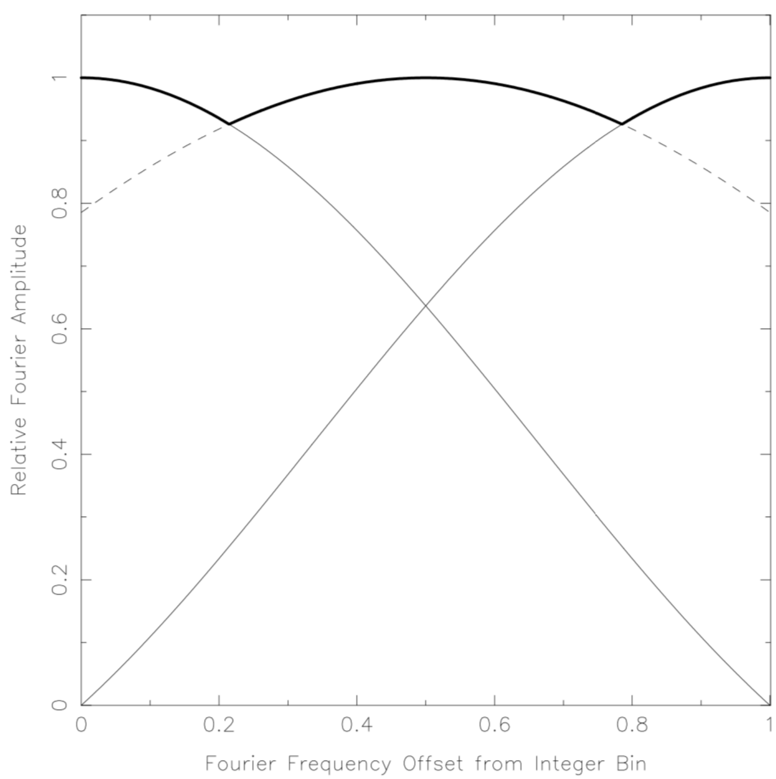
Most radio pulsars are much weaker than this and can't be picker out by eye like that one. Summing up the harmonics can help identify them: Stretch the spectrum by 2 and add to original, then repeat…



Ransom et al. 2002, AJ **124**, 1788.

How well this does depends on the duty cycle of the pulsar – wide profiles don't have many harmonics, so summing mostly means adding noise.

If the pulsar's frequency doesn't land in the centre of a frequency bin (width 1/(total time T), then sensitivity is reduced.

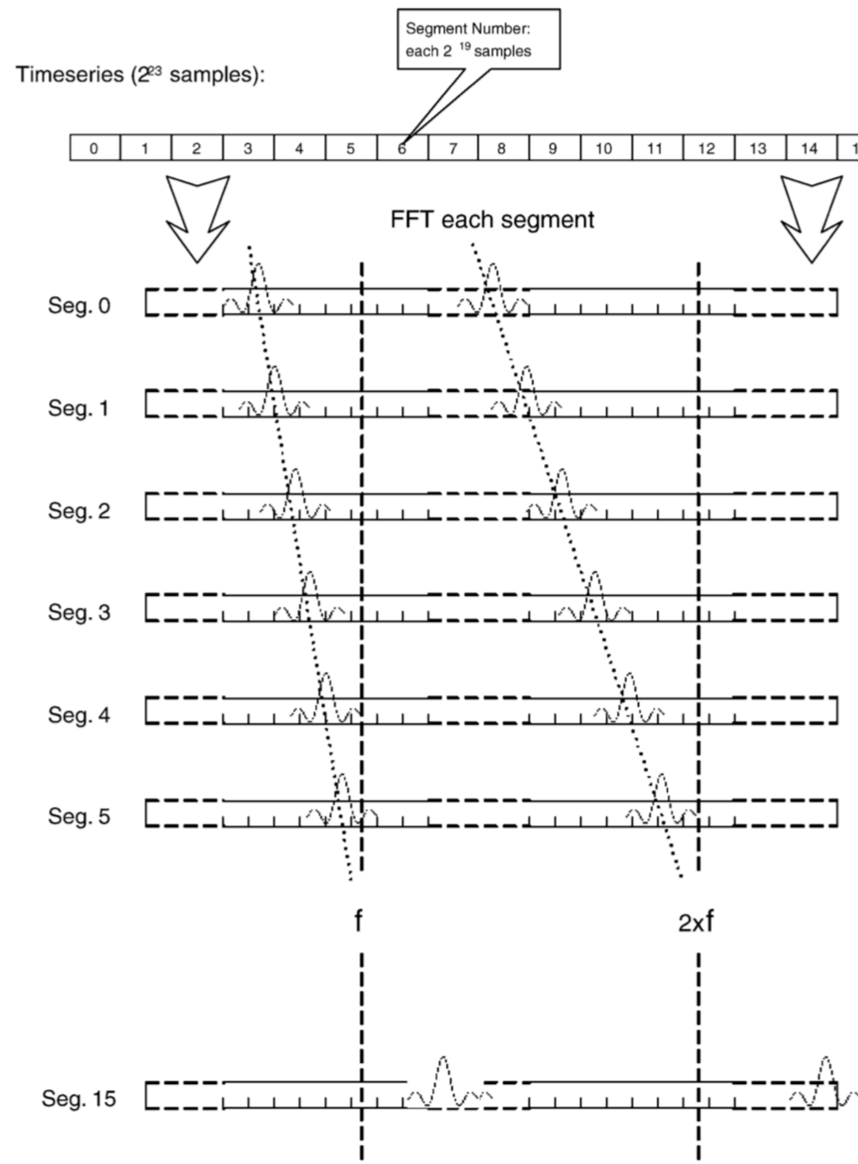


Ransom et al. 2002, AJ **124**, 1788.

Correction of this "scalloping" effect by "interbinning" – Fourier interpolation between neighbouring bins.

Some of the most exciting pulsars are in short-period binaries, so their observed frequencies aren't constant over an observation. So we need to search over **acceleration** space as well.

"Stack search" on long integrations (Faulkner et al. 2004, MNRAS **355**, 147) – computationally cheap but low-sensitivity as incoherent.



Coherent acceleration search methods are more sensitive. These include:

- Resampling the time series at many different trial accelerations before the periodicity search

- Looking for correlated signals in the frequency domain based on templates representing different accelerations (equivalent to resampling, but faster; Ransom et al. 2002, AJ **124**, 1788)

- Phase modulation searches looking for the full extent of the frequency changes for binary orbits << observation time (Ransom et al. 2003, ApJ **589**, 911)

- Dynamic power spectrum searches – similar to stack/slide.

A periodicity search can produce a lot of candidates. Is it worth looking at everything above signal-to-noise of 3? No! You are looking at the same data set in multiple different ways, increasing the number of trials. For a Fourier-amplitude search, the minimum interesting SNR is:

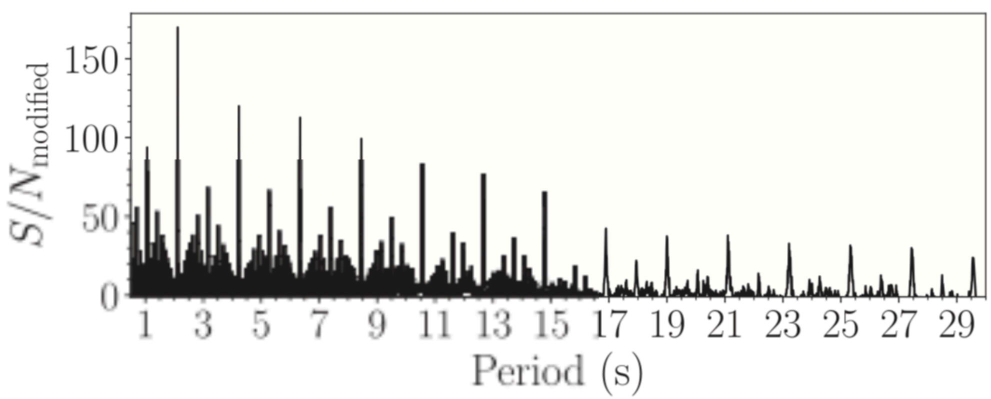
$$\mathrm{SNR}_{\mathrm{min}} = \frac{\sqrt{\ln(n_{\mathrm{trials}})} - \sqrt{\pi/4}}{1 - \pi/4} \simeq \frac{\sqrt{\ln(n_{\mathrm{trials}})} - 0.88}{0.47}$$

For most pulsar searches, this is about 8.

Still need human (and often now machine-learning) sifting.

The slowest pulsars are the most affected by red noise, leading to many false positives in the low-frequency candidates. One way to fight this is with the Fast Folding Algorithm (FFA; Staelin 1969, IEEEP, **57**, 724):

- Similar to FFT in avoiding duplicate summations (NlogN)
- Also good for finding faster, but weaker pulsars with lots of harmonics.
- Used in exoplanet transit searches as well!

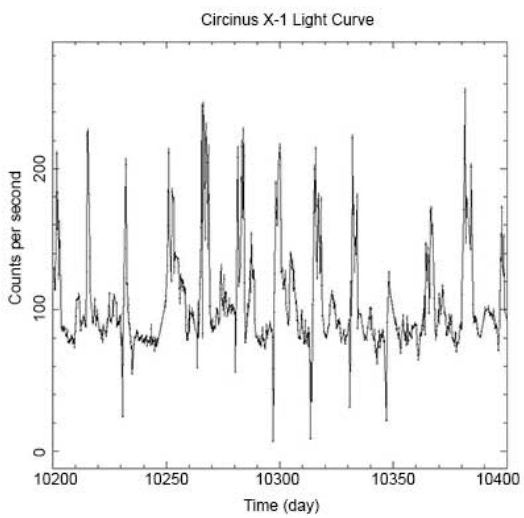


PSR 2004+3137 – Parent et al. 2017, ApJ **861**, 44

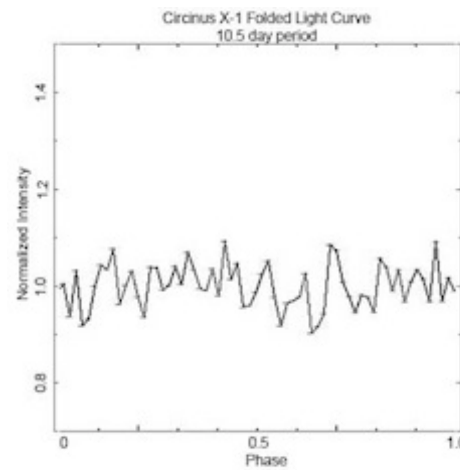
Radio data have high time and frequency resolution.

What if your data consist of events such as photon detections?

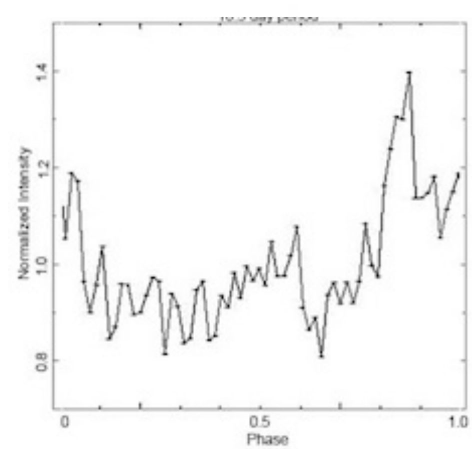
Epoch folding



10.5-day period



16.6-day period



https://imagine.gsfc.nasa.gov/science/toolbox/timing2.html

Epoch folding can be improved on, eg by making it Bayesian (Gregory & Loredo 1992, ApJ **398**, 146) and comparing the data to an unmodulated signal.

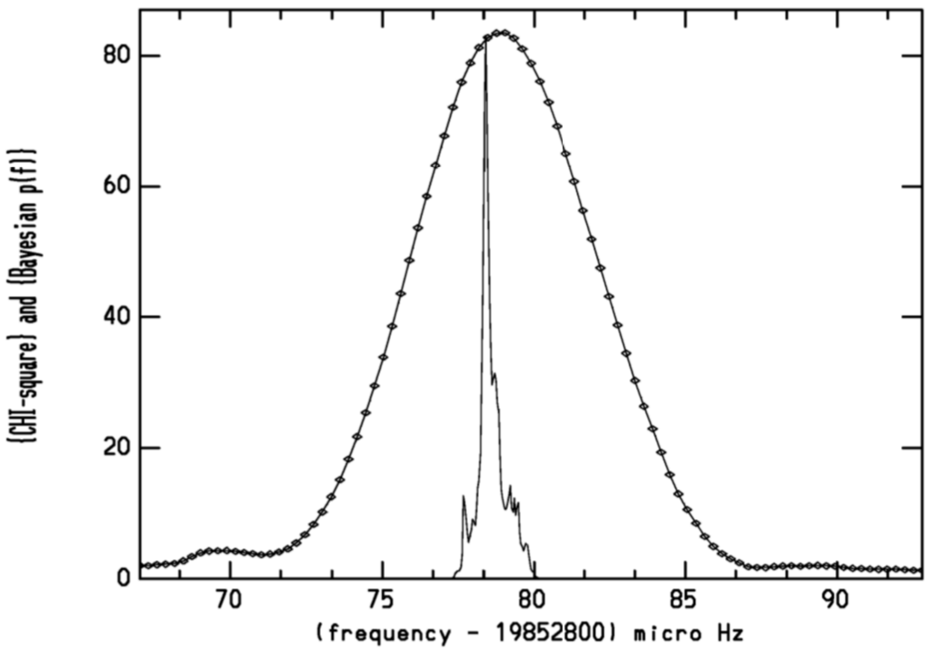


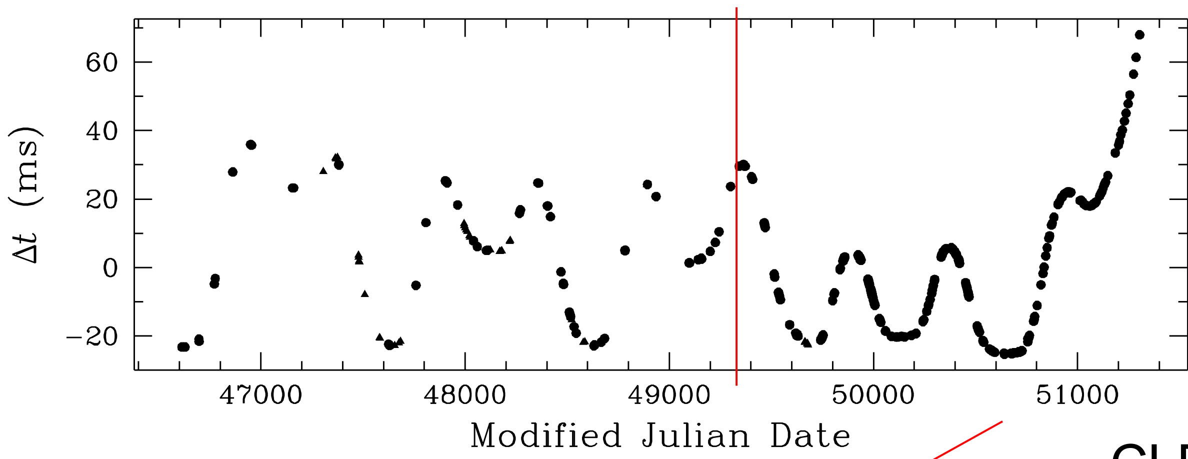
FIG. 4.—Close-up of the largest frequency peak, comparing the GL marginal probability density for f with the EF $\langle\chi^2\rangle_\phi$ statistic (*diamonds*). The $\langle\chi^2\rangle_\phi$ statistic vs. trial frequency results from EF analysis using $m = 5$ bins.

Gregory & Loredo 1996, ApJ **473**, 1059

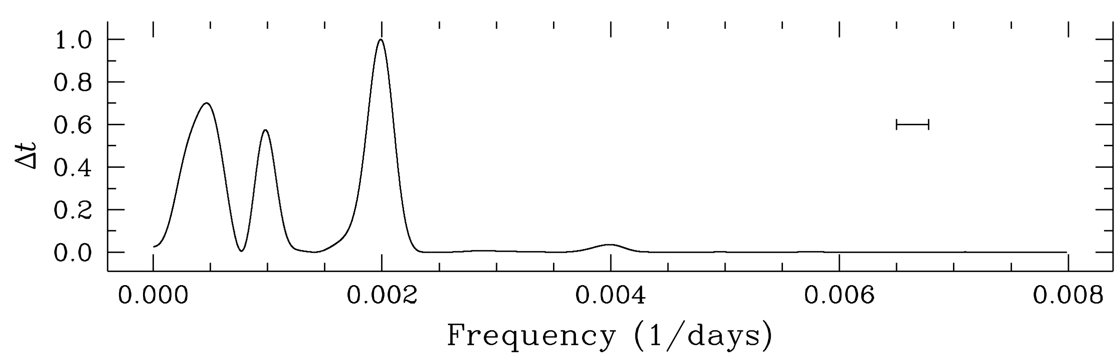
Comparison of GL method with Epoch Folding of ROSAT data on PSR B0540-69.

There are other algorithms, too: H-test, Z_m^2 test…

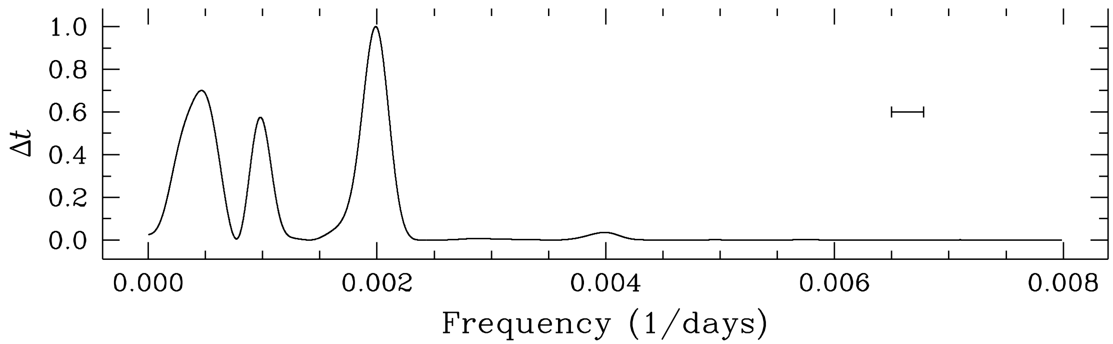
More ways of dealing with unevenly-sampled data…



CLEAN algorithm



Timing residuals for PSR B1828-11, Stairs et al 2000, Nature **406**, 484.



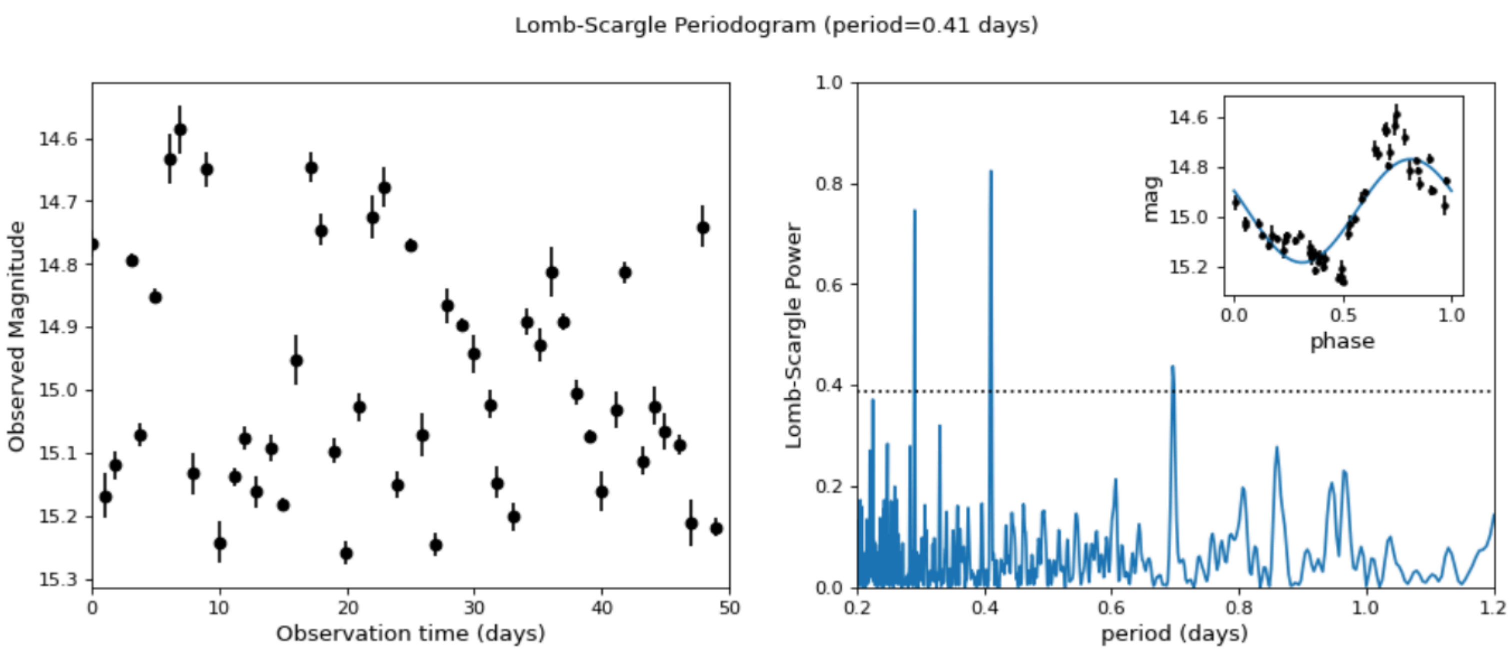
Stairs et al 2000, Nature **406**, 484.

This is a 1-D CLEAN algorithm, adapted from interferometry.

- Compute direct FT of data → “dirty spectrum”
- Compute FT of sampling times → spectral window
- Identify peak in dirty spectrum → “clean component”
- Subtract (fraction of) clean component convolved with spectral window
- Iterate until noise remains.
- Make clean spectrum from clean components and spectral window.

Roberts et al. 1987, AJ **93**, 968.

Another well-known algorithm for these cases is Lomb-Scargle, effectively fitting a sinusoid at multiple frequencies and taking the χ^2 of the fit for the periodogram. It allows an estimate of the false-alarm probability (0.01 in the example).

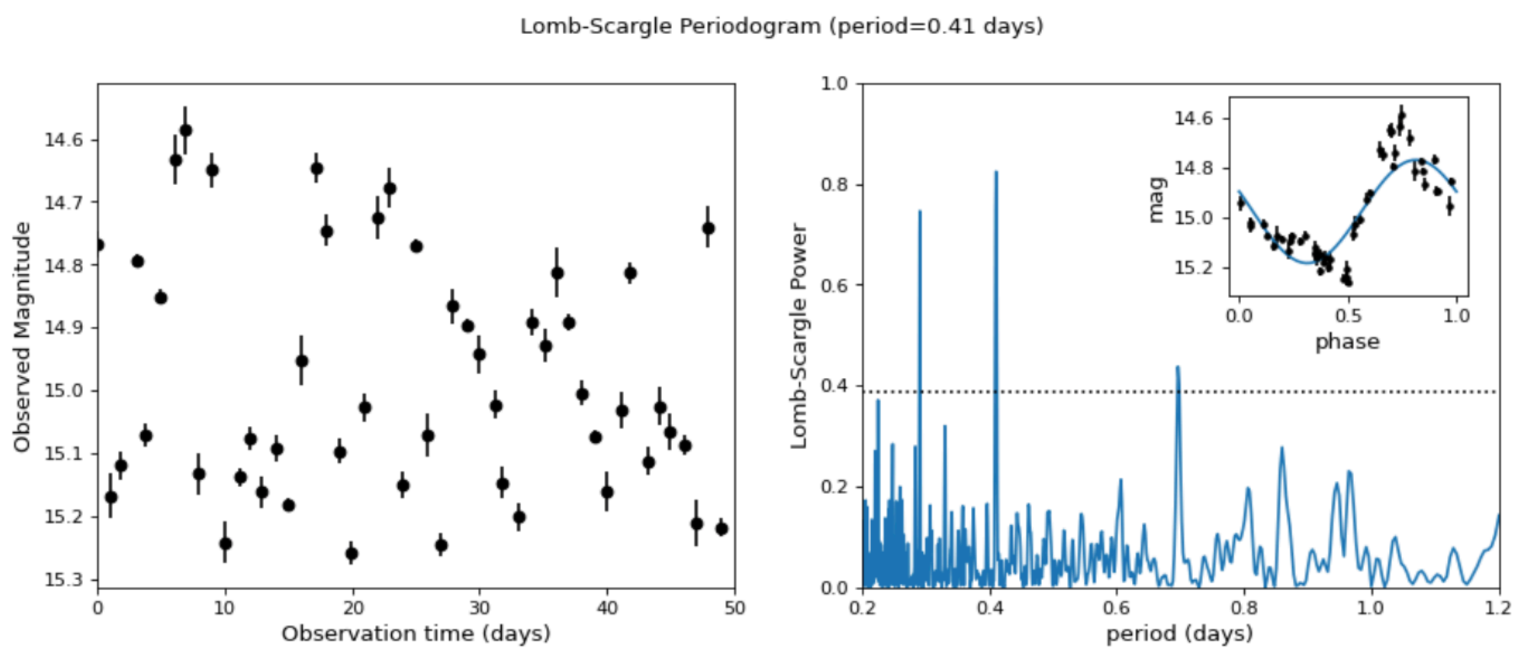


https://docs.astropy.org/en/stable/timeseries/lombscargle.html

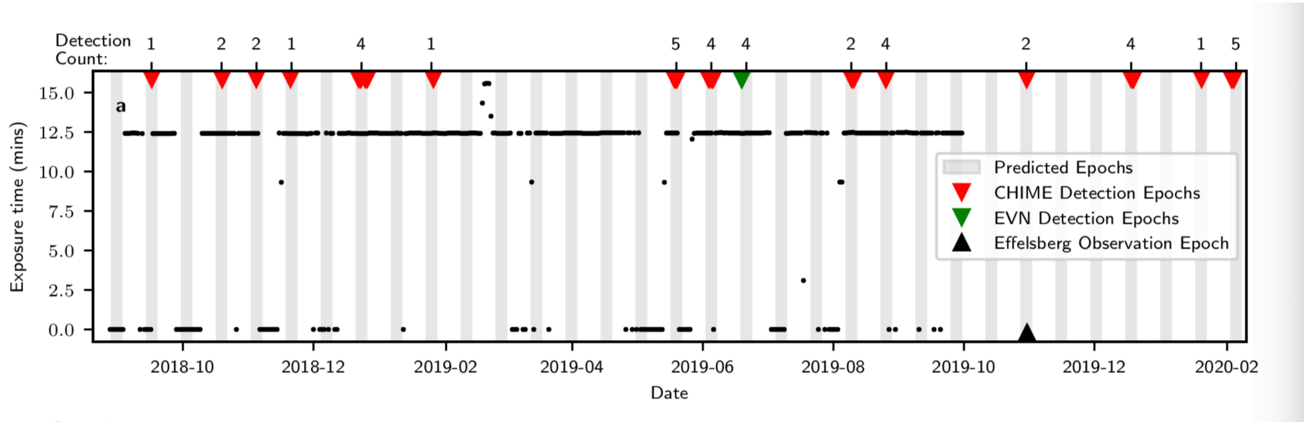
See also https://arxiv.org/pdf/1703.09824.pdf for a thorough review.

The true period in the data is 0.41 days (inset). What are those peaks at 0.29 days and 0.69 days? They are aliases. The observations happen about 1/day, so we can expect to see power at $f_{\rm obs} = f_{\rm true} \pm \frac{n}{\rm day}$ for integer *n*.

LS can be sensitive to frequencies above the Nyquist frequency – have to be careful!



https://docs.astropy.org/en/stable/timeseries/lombscargle.html



CHIME/FRB Collaboration, Nature **582**, 351 (2020)

This is the repeating FRB "R3" for which we found a periodicity last year. Bursts often come in clumps (red and green triangles).

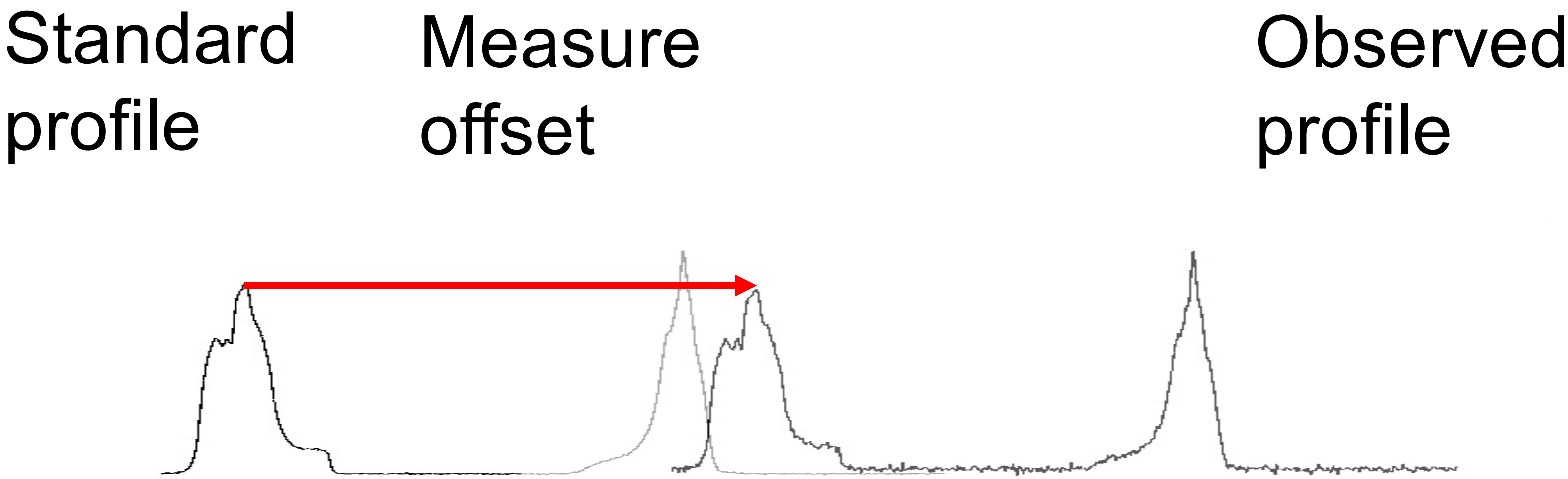
Effective folding, H-test, FT with incoherent harmonic summing all yield a period of 16.35 days. But CHIME is a transit telescope, so we have to allow for aliasing:

$$f_{\rm true} = \frac{N}{\rm sidereal\,day} \pm f_{\rm obs}$$

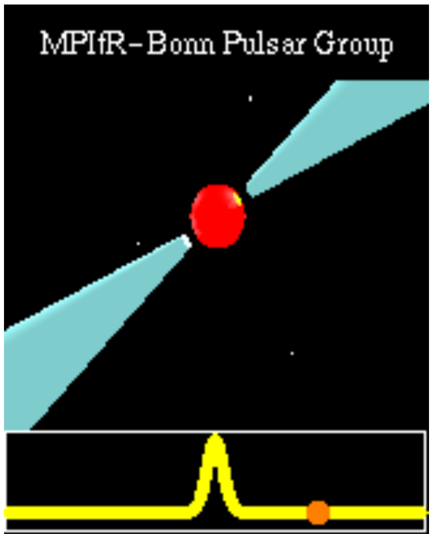
But we argue that N=0.

Coming back to pulsars, we want to take advantage of the reproducibility of the pulse profiles to determine precise Times of Arrival (TOAs) for high-precision timing.

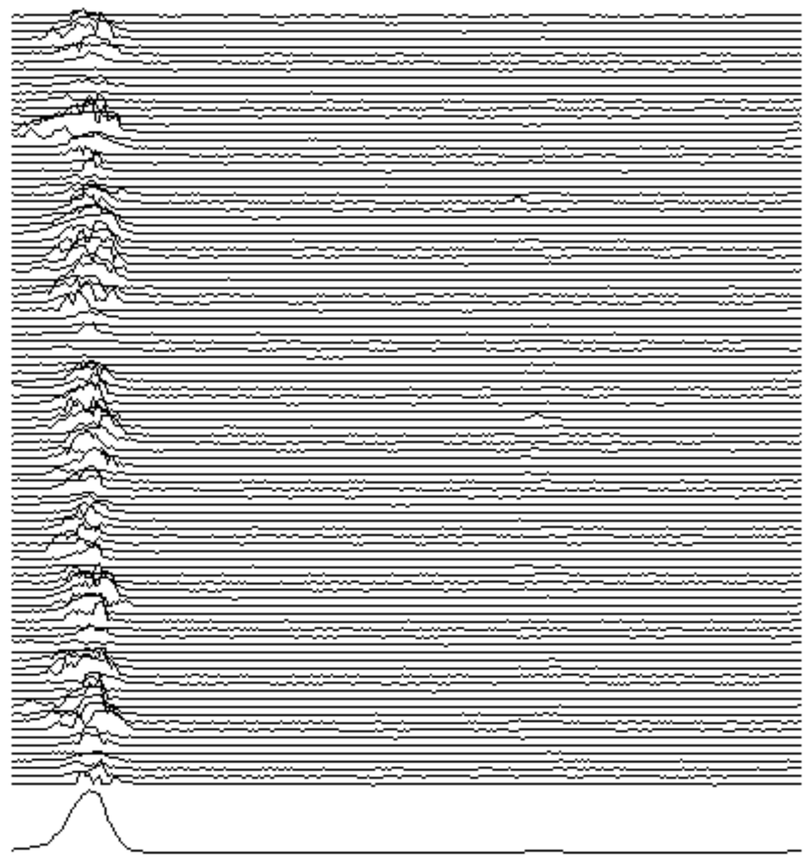
This is done by cross-correlation, with some tricks.



First we have to average over enough profiles to get a stable one – individual pulses vary a lot.

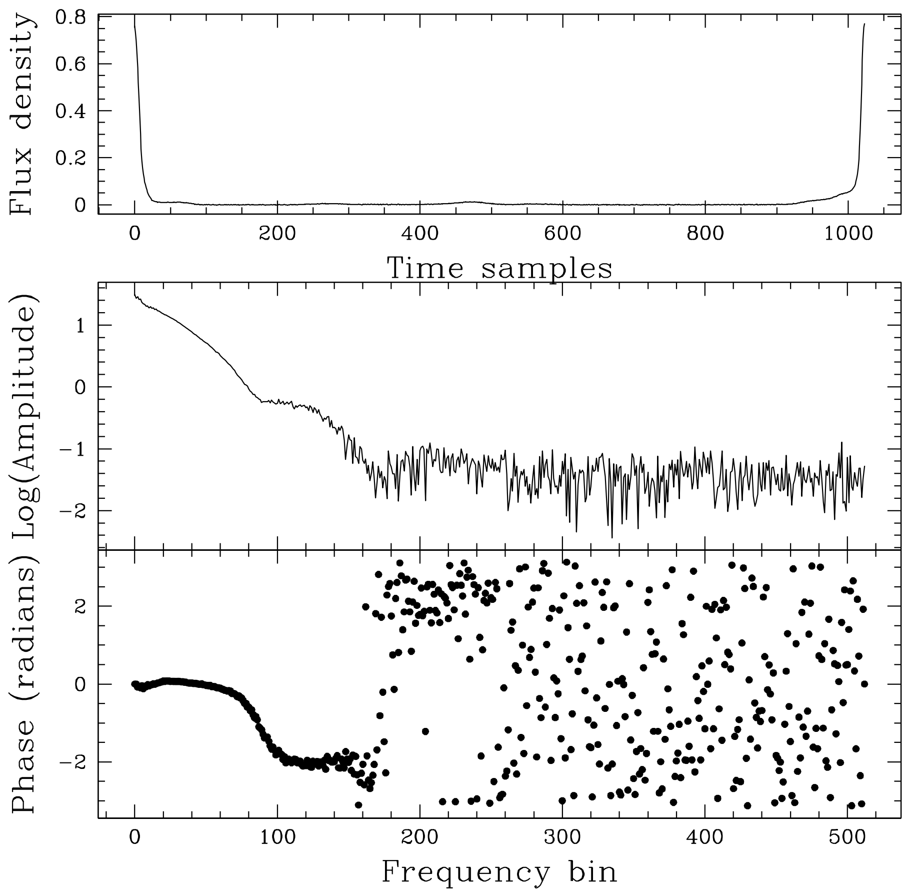


Lighthouse model



Add together several hundred pulses ➔ stable “integrated profile”

Most teams do the cross-correlation in the frequency domain, following Taylor 1992. The multiple harmonics give a better estimate of the phase shift.

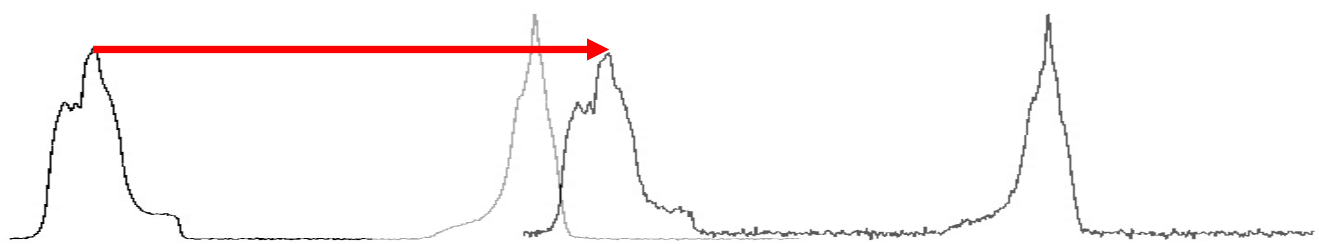


Standard profile and its Fourier representation for PSR B1534+12 at 430 MHz.

Standard profile

Measure offset

Observed profile



$$p(t) = bs(t+\tau) \leftrightarrow P_k e^{i\theta_k} = bS_k e^{i(\phi_k + k\tau)} + \text{noise} + \text{constant}$$

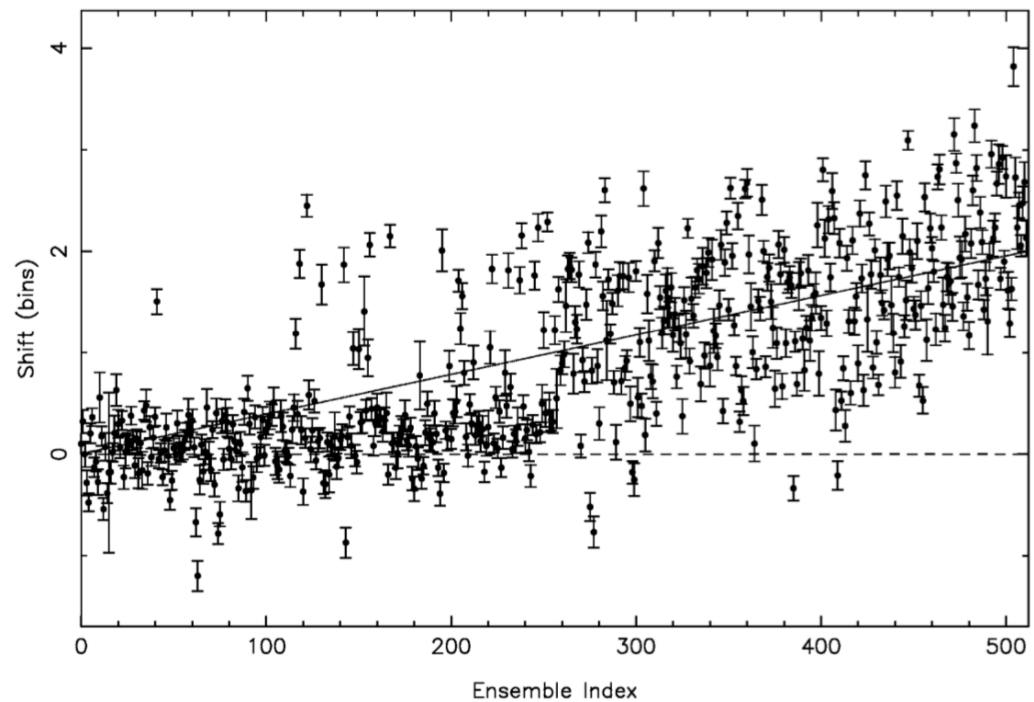
Minimize:

$$\chi^2(b,\tau) = \Sigma_{k=1}^{N/2} \left| \frac{P_k - bS_k e^{i(\phi_k - \theta_k + k\tau)}}{\sigma_k} \right|^2$$

We have to build the standard profile, or template, out of existing observations with the same observing parameters.

But there can be problems when the profiles used are too noisy: you can end up cross-correlating the noise!

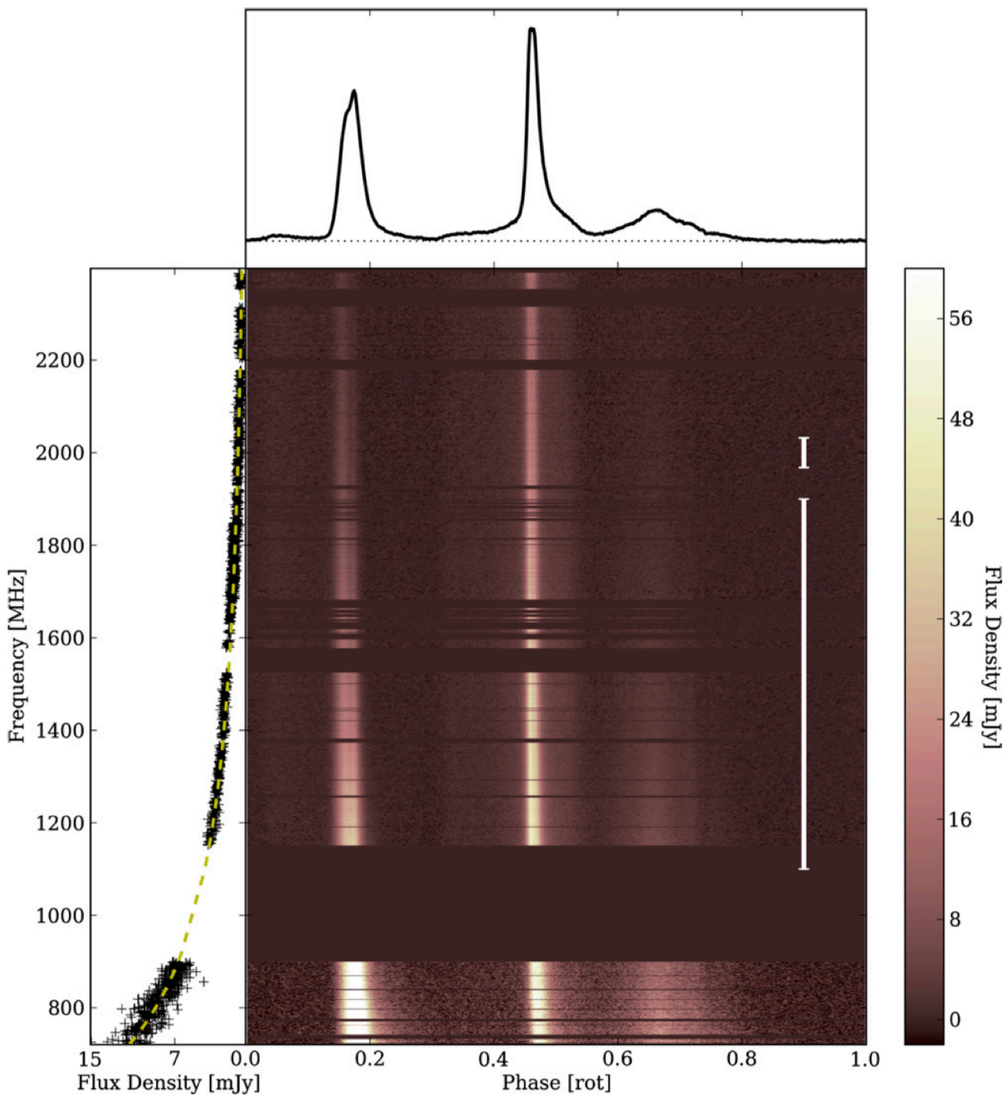
Hotan et al 2005, MNRAS **362**, 1267. 256 noisy profiles were used to make the standard, and the measured shift (in the frequency domain) was wrong for those.



So we’ll use high-SNR profiles (SNR > 25) to make the standard profile.

Another good idea: eliminate/reduce baseline noise by:

- Smoothing the standard profile eg using wavelets
- Zeroing out the baseline
- Making an analytic profile from a set of Gaussians



Another concern: frequency dependence of the profile, here seen for M28A.

Evolving gaussians were a first attempt at a 2-d template, but now a PCA-based template is giving better results.

Pennucci et al. 2014, ApJ, **790,** 93

Conclusions

The “best” technique for finding a periodicity depends strongly on the sampling of the data.

Watch out for aliasing!

Periodic signals can be useful once found, and the frequency domain is an excellent way to characterize and manipulate them.

New and improved algorithms always needed!

Please do the survey at http://bit.ly/TimeDomainII