Description of PowerFlux algorithms and implementation (draft)

LIGO-T050186-00-Z

Vladimir Dergachev

September 25, 2005

Contents

List of Figures

1 Introduction

PowerFlux is a program for estimating the flux of monochromatic gravitational waves from a particular source in the sky. It uses periodograms of fixed (usually 30 minutes) segments as input data. Because of this, PowerFlux is much less sensitive to variations in phase than coherent methods (for example ComputeFStatistic). The result is a tradeoff of sensitivity for drastic reduction of search space, allowing for full-sky full-bandwidth searches to be completed in reasonable time.

2 Common statistics

In this section we will describe common statistical algorithms employed by PowerFlux .

2.1 Weighted mean

The weighted mean of samples $\{s_i\}$ is computed using weights $\{w_i\}$ as follows:

$$\bar{s} = \frac{\sum_i w_i \cdot s_i}{\sum_i w_i}$$

when all $w_i = 1$ we obtain an ordinary mean value.

2.2 Optimal weighting

In principle, any weighting scheme produces correct answers as long as it is independent of the samples being weighted. However, some weighting schemes are better than others. If one uses variance of the average as a measure of optimality of the weighting scheme (the smaller the better), it is possible to determine optimal weights exactly.

Assume that each of the independent samples s_i has a variance σ_i^2. We introduce new variables u_i with the formula

$$u_i = \frac{w_i}{\sum_i w_i}$$

These variables satisfy the constraint $\sum_i u_i = 1$ and inequalities $u_i \geq 0$ - i.e. an $n-1$ dimensional simplex. Then

$$\bar{s} = \sum_i u_i s_i$$

and

$$V(\{u_i\}) = \mathrm{Var}\,(\bar{s}) = \sum_i u_i^2 \sigma_i^2$$

Assuming, for the moment, that u_i are independent the Jacobian of the map $V(\{u_i\})$ is $(2u_i\sigma_i^2)$. The minimum would be achieved either on the boundary of the simplex or in a point where the Jacobian is collinear with the Jacobian of the constraint (1...1).

In the latter case:

$$2u_i\sigma_i^2 = \lambda$$

and

$$u_i = \frac{\lambda}{2}\frac{1}{\sigma_i^2}$$

Thus, if we use $w_i = \frac{1}{\sigma_i^2}$ we would have an extremal value of the variance $\mathrm{Var}\,(\bar{s})$.

Is this a global minimum ? If we substitute these weights we obtain:

$$\mathrm{Var}\,(\bar{s}) = \frac{\sum_i \frac{1}{\sigma_i^2}}{\left(\sum_i \frac{1}{\sigma_i^2}\right)^2} = \frac{1}{\sum_i \frac{1}{\sigma_i^2}}$$

Restricting u_i to the boundary of the simplex is equivalent to removing one or more samples s_i from the computation. By considering a smaller simplex we will find that the global minimum is inside that smaller simplex - and thus has to use $\frac{1}{\sigma_i^2}$ weighting scheme albeit with some samples removed.

But then the formula for the variance will apply. Since removal of one or more samples removes one or more terms $\frac{1}{\sigma_i^2}$ from the denominator the result will be bigger.

Thus we obtain the following (intuitively true) result: as long as the variances are well known it is best to use all the data. The optimal weighting scheme is $w_i = \frac{1}{\sigma_i^2} = \frac{1}{\mathrm{Var}\,(s_i)}$.

Note: No assumptions have been made on a particular distribution of samples s_i - only that their variances exist and they represent independent measurements.

2.3 Median

A median of samples $\{s_i\}$ is a value such that equal numbers of s_i are strictly above and below the median. In case the total number of samples is even we use the average of two innermost samples as "canonical" value of the median.

2.4 Feldman-Cousins method

Feldman-Cousins method is a systematic way to provide simultaneous upper and lower estimates given a measurement of inherently positive quantity with a known noise distribution. When the measured value is small the lower bound estimate becomes zero and the scheme naturally transitions into an upper-limit mode.

The motivation and detailed description of Feldman-Cousins method can be read in [1]. Here we will provide a short description that would make it easier to understand the particular implementation used by PowerFlux .

Let x be the measured quantity, m is the estimated average level and v the estimated dispersion of x from the average level.

We define dx - signal strength parameter - as

$$dx = \frac{x - m}{v}$$

Then the upper and lower bounds are given by $f_{upper}(dx) * v$ and $f_{lower}(dx) * v$ where the functions f are computed using the 95% confidence level tables for Gaussian distribution given in [1].

It is important to address the question of determination of m and v. In Feldman-Cousins paper the distribution is assumed to be fixed and known in advance. Therefore, the renormalized dx values should have Gaussian distribution or one cannot rely purely on mathematical correctness of Feldman-Cousins approach.

In practice, PowerFlux assumes that a given stretch of 501 bins that are being analyzed has the same very slowly varying parameters m and v. Furthermore, PowerFlux assumes that the distribution is Gaussian with a few possible power excesses. Then m and v are determined by linear fit of the distribution of samples with 20% tails discarded from both ends.

The assumption of Gaussianity is obviously violated by steeply colored noise, however it can be argued that the result is an inflated noise estimate which produces correct, albeit inefficient, upper limits.

We must also note that determination of m and v from the same sample would change the results slightly in comparison with theoretical situation.

Ultimately, the effects are smaller or comparable to the effects arising from nonlinear interaction of different processing stages of PowerFlux . Since the upper limits produced are 95% confidence limits it seems fitting that a Monte-Carlo test of detection efficiency and correctness of entire PowerFlux will determine the correctness of the results.

If pathologically colored noise (for example, one that radically changes behaviour right before 20% quantile) shows to be a problem one should really investigate the affected frequency bands manually and determine why the detector would produce such an artifact.

2.5 Robust estimation of Gaussian distribution parameters

In order to derive Feldman-Cousins upper limits it necessary to determine the parameters of the underlying Gaussian distribution. It is important that such estimates be robust to influence of actual signals and detector artifacts.

The procedure we use is as follows:

1. The input data is 501 bins of weighted power averages.

2. The input is sorted and top and bottom 50 bins are discarded thus producing a sorted vector of 401 bins.

3. The mean μ is estimated as a mean of the remaining 401 bins.

4. The standard deviation σ is estimated as 5.22 times the coefficient of linear fit to the sorted vector:

$$\sigma = \frac{5.22}{401} \sum_{i=0}^{400} (s_i - \mu) \cdot \frac{i}{400}$$

This code is implemented in file `statistics.c`.

A simulation was implemented in R statistics environment (source file `S.R` located with source of this document) to test the performance of this code. It consisted of 10000 iterations each of which generated 501 numbers using standard normal distribution.

Parameter	Mean	Median	Standard deviation/$\sqrt{(}10000)$
μ	0.0002954174	0.0005878784	0.0004664218
σ	1.000346	0.9999546	0.0003987203

We would like to note that both mean and median of estimated parameters are closer than one standard deviation from each other. Thus this estimate is unbiased not only the conventional sense (mean equals true value), but also in the robust analog - median equals true value. This is important as median is preserved under transformation by a monotonic function.

3 Data flow

PowerFlux is organized as a monolithic engine. Because of memory and bandwidth constraints of existing hardware, the design is structured around working with a core dataset of periodograms and limiting derived data to a very small size in order to fit into the L2 caches of present CPUs.

Additionally, current interferometer data exhibits numerous artifacts (for example lines at 0.25 Hz increments and steeply colored noise) which makes it desirable to analyze as small a band as possible. Of course, the band cannot be too small, or it would not be sufficient for statistical analysis. The compromise employed in the latest version is to produce estimates for 501 bins of 1/1800 Hz each and use a slightly larger band in intermediate calculations (due to the need to apply Doppler shifts and, possibly, spindown compensation).

The schematic of data flow inside PowerFlux is presented on figure 1.

The algorithm proceeds as follows:

- `INIT` PowerFlux initializes various data and processes command line arguments.

- `PREP1` Sky grids (`patch_grid` and `fine_grid`) are computed and stored for future use. Their values are output for later analysis.

- `INPUT` SFT Powers are loaded. PowerFlux can perform a software injection of a single pulsar at this stage.

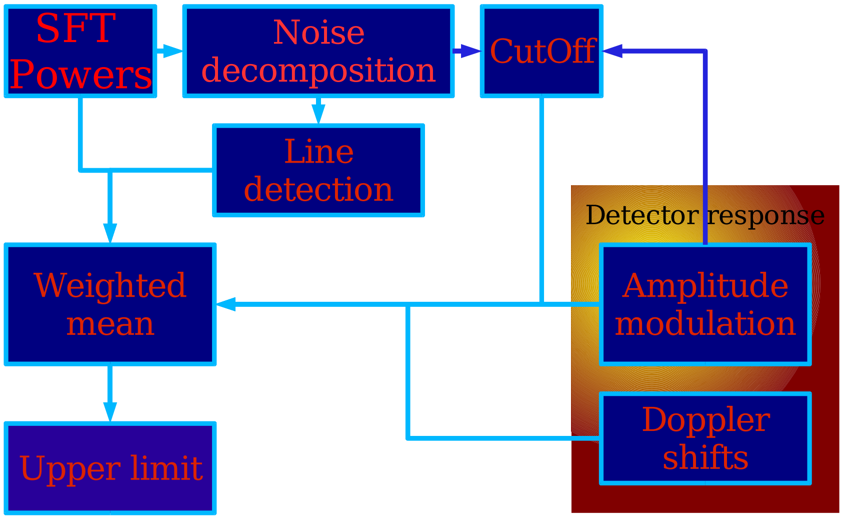

Figure 1: PowerFlux data flow

- `DIAG2` Diagnostic quantities are computed and output for each frequency bin: ordinary mean, standard deviation, Kolmogorov-Smirnov test against exponential distribution.

- `COMP3` Noise decomposition is performed. `TMedians` and `FMedians` are stored for future use.

- `DIAG4` Diagnostic quantities are computed and output: weighted mean, `CutOff` (without demodulation, that is without applying antenna pattern), weighted mean and regular mean computed using the `CutOff`

- `PREP5` Doppler shifts are computed for the entire run. Intermediate quantities for computation of amplitude modulated antenna response are computed and stored.

- `PREP6` `CutOff` values taking into account amplitude modulated antenna response are computed for each point in the patch grid and for each polarization. Their values are plotted and stored for diagnostics.

- `PREP7` Line detection is performed using weighted means for analysis.

- `MAIN LOOP` For each element of patch grid and each polarization we loop over all elements of fine grid inside it and accumulate weighted powers for each frequency bins independently, which is followed by immediate analysis using Feldman-Cousins approach to detection of excess power with the results stored for later processing and output. The same array of weighted powers is used for each polarization and each element of patch grid which improves memory and bandwidth efficiency.

- `OUTPUT` Sky maps and raw data are output for later analysis.

4 Input

PowerFlux input consists of a set of SFTs or periodograms and ephemeris files for computing detector response.

At the moment three formats for SFTs are supported: GEO (small binary header + binary data) and two custom formats for SFTs and periodograms which consist of an ASCII header followed by "binary" on a separate line and then binary data.

It is expected that the effects of windowing are normalized in such a way that a pure bin-centered sine wave of amplitude 1.0 would produce SFT power in corresponding frequency bin of 0.5. Of course, on top of this one applies the usual normalization required by corresponding format.

Thus given a bin-centered sine wave of amplitude 1.0 the power in corresponding bin should be

- Custom formats: $0.25 \cdot (\texttt{TimeBase})^2 \cdot (\texttt{Rate})^2$ - regardless of number of bins actually stored

- GEO: $(\texttt{Stored bins})^2$

Note that different windowing methods vary in response to non bin-centered signals. One should use `upper-limit-comp` and `lower-limit-comp` PowerFlux options to compensate fo that.

At the moment the SFTs or periodograms are expected have frequency binning of 1/1800 Hz.

5 Power-only software injection

PowerFlux can inject a simulated signal of a single pulsar into the input data. The algorithm modifies each SFT as follows:

- The frequency, as received by the detector, is computed from doppler shift and spindown parameter.
- The pulsar power $P_{\texttt{PULSAR}}$ as seen by the detector is computed using value of `fake-strain` parameter and amplitude modulation.
- The frequency is rounded to the nearest bin and the entire power times a factor of 0.7 is added assuming a random phase in relation to previous value:

  $$P_{\texttt{SFT}} \leftarrow P_{\texttt{SFT}} + P_{\texttt{PULSAR}} \cdot 0.7 + 2 \cdot \sqrt{P_{\texttt{PULSAR}} \cdot 0.7 \cdot P_{\texttt{SFT}}} \cdot \cos(\phi)$$

  Here ϕ is a random number uniformly distributed between 0 and π and independent for each SFT and $P_{\texttt{SFT}}$ is the power in SFT bin.

The factor 0.7 is used to account for decrease in the power in individual SFTs due to Hann windowing. The shoulder effects and variation in maximum bin power are not simulated.

6 Noise level estimation

At the current time the sensitivities of the interferometers are far from stable and can change considerably during a run. Additionally, it is useful to be able to run PowerFlux before complete identification of ”clean” science mode segments is performed, which may result in some exceptionally noisy segments making into the core data set.

Thus it is necessary to estimate the average noise level of different 30 minute periodograms in order to apply proper weighting and cutoffs to incoming data.

An additional complication is that the spectral shape of periodograms can be non-trivial, even on bands 0.25 Hz in size.

The algorithm employed is a variation of iterative proportional fitting algorithm.

The core data set is represented as a rectangular array. The band of required length is extracted from each peridiogram and forms rows of the array. The columns then correspond to a particular frequency.

Two vectors `TMedians` and `FMedians` (corresponding to columns and rows) are formed and initialized to 0.

The algorithm proceeds as follows:

1. Compute decimal logarithm of all entries of the array.

2. Compute median of each row in the array. Subtract median value from the row it was computed from and add it to the corresponding `TMedians` entry.

3. Compute median of each column in the array. Subtract median value from the corresponding column and add to the `FMedians` entry.

4. If all the medians computed during previous two steps are 0 or a negligibly small value, the algorithm is complete. Otherwise loop to step 2.

7 Line detection and mitigation

While ideally the noise spectrum of interferometer that does not receive a signal is Gaussian with slowly changing parameters, the reality is that artifacts abound. Line detection addresses the presence of excess power in isolated bins (lines).

In particular, during the S2 and S3 runs there were many lines in bins located at 0.25 Hz. Since it is highly unlikely that a naturally occurring phenomena will produce such signal, the presence of these lines - if not dealt with - will inflate upper bounds, as well as possibly prevent detection of true signals in the future.

7.1 Detection

Since there are so many lines and since they are irregular, it is desirable to have an automated way of detecting a line.

The algorithm proceeds as follows:

`INPUT`: array of mean powers, maximum number of lines to detect, number `nmost` - the largest rank we would still expect to exhibit ”average” behaviour (as opposed to bins with power excess) .

`LEVELS`: Compute median of mean powers, quantile `qlines` corresponding to the number of lines and quantile `qmost` corresponding to `nmost`

`PASS1`: mark bins with power value greater than `qmost` as `LINE_HIGH`, mark bins with power value greater than `qlines` as `LINE_CANDIDATE`, mark bins greater than 2 · `qmost` − `median` as `LINE_VERY_HIGH`

`PASS2`: Mark 5 or more consequentive bins marked `LINE_HIGH` as `LINE_CLUSTERED`. Also mark as `LINE_ISOLATED` bins that are marked `LINE_HIGH` and are at least two times as much above the median as adjacent bins.

`PASS3`: mark bins already marked with `LINE_CANDIDATE` and `LINE_VERY_HIGH` but not `LINE_CLUSTERED` as `LINE_YES`.

7.2 Mitigation

Due to application of Doppler shifts (and spindowns) the actual bin where the signal ends up varies during the run. The simplest line vetoing method is to simply ignore contribution of parts of the run where the signal falls into frequency bin contaminated by the line.

This is the approach currently implemented in PowerFlux. The drawback is that, for some parts of the sky, the combined Doppler shift from orbital and rotational motions of the Earth will stay constant for long periods of time, even during a 2 month run like S3. For these portions of the sky a particular signal frequency will yield too little accepted data, leading to increased effective noise.

However, the upper bound improves inversely proportionately to the square root of data samples, and thus even if only 20% of the samples remain, a reasonable estimate can be made.

7.3 Effects

The effect of vetoing single frequency bins is most pronounced at lower frequency, where the Doppler shifts are smaller and the signal does not span many bins. As an example, we will examine the effects of line exclusion on analysis of the 223.88-224.138 Hz band of S3 data produced by the L1 interferometer.

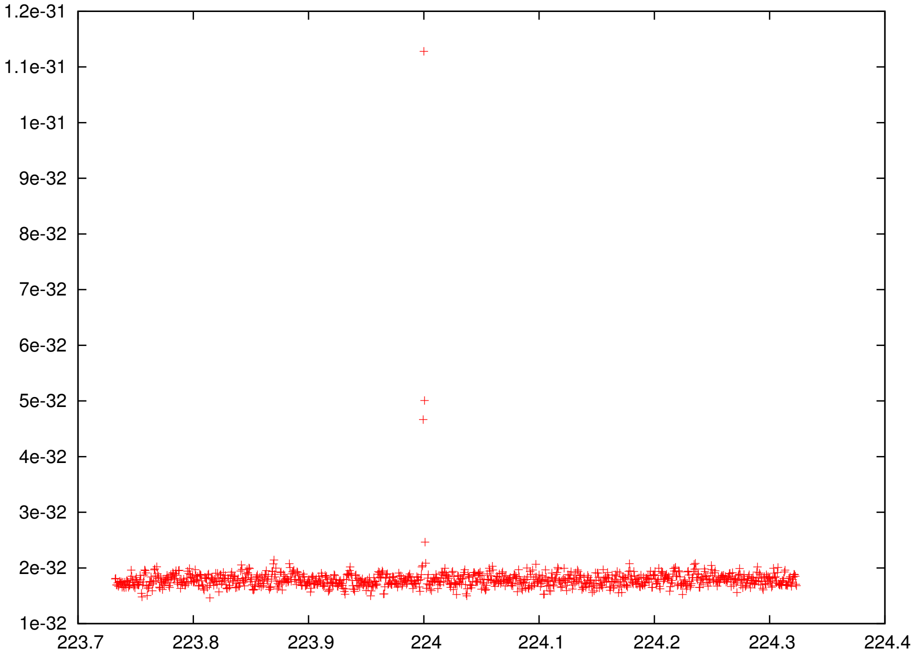

Figure 2: Background noise in 223.888-224.138 Hz band, L1, S3

The background noise level is shown in figure 2. The quantity plotted is power in arbitrary units. Note a high outlier at 224 hz as well as shoulders arising from Hann windowing.

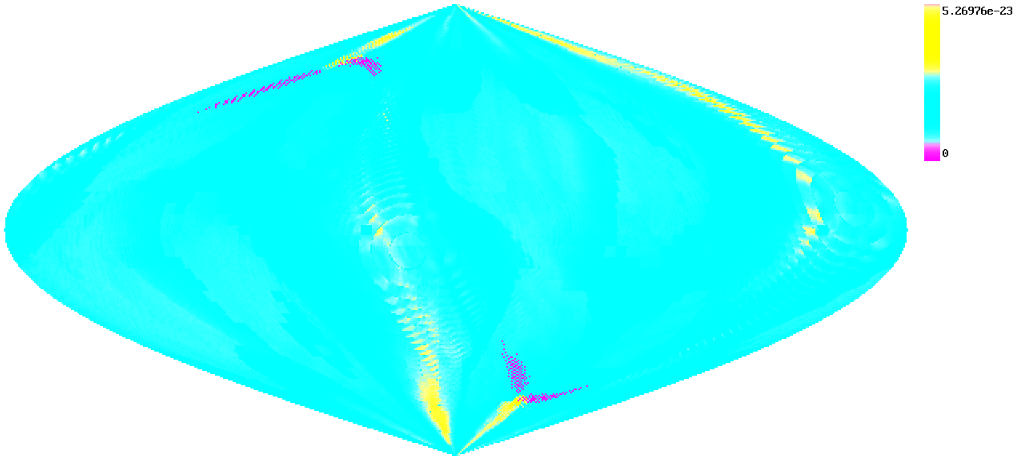

Figure 3: Effects of single bin line veto

Figure 3 shows the effect of excluding a single frequency bin at 224 Hz. The plotted quantity is a maximum over all detected bins in the band of a (highly preliminary) upper bound estimate on strain.

The magenta pixels mark sky positions where there was a signal that spent at least 80% of the effective observation time inside an excluded detection bin. It is important to note that the presence of even a single obscured detected bin would cause a particular location to be colored magenta. The yellow areas mark areas of large 95% CL upper limits, and are likely caused by the two shoulders which were not excluded.

Figure 4 shows a spectral plot of maximum of weighted power average over a declination band from -1.17 to -0.78 (in radians).

The maximum of mask ratio (ratio of weight of discarded bins to the total weight) is shown in figure 5.

The result of excluding four highest bins around 224 Hz is shown in figure 6. Note the greatly increased obscured area. Even though the yellow area is larger, the upper limit for the entire sky has improved 3 fold.

We see that there is a tradeoff - excluding the two shoulders would greatly increase obscured area, while using them to produce upper limits results in inflated estimate for a large portion of the sky-frequency space.

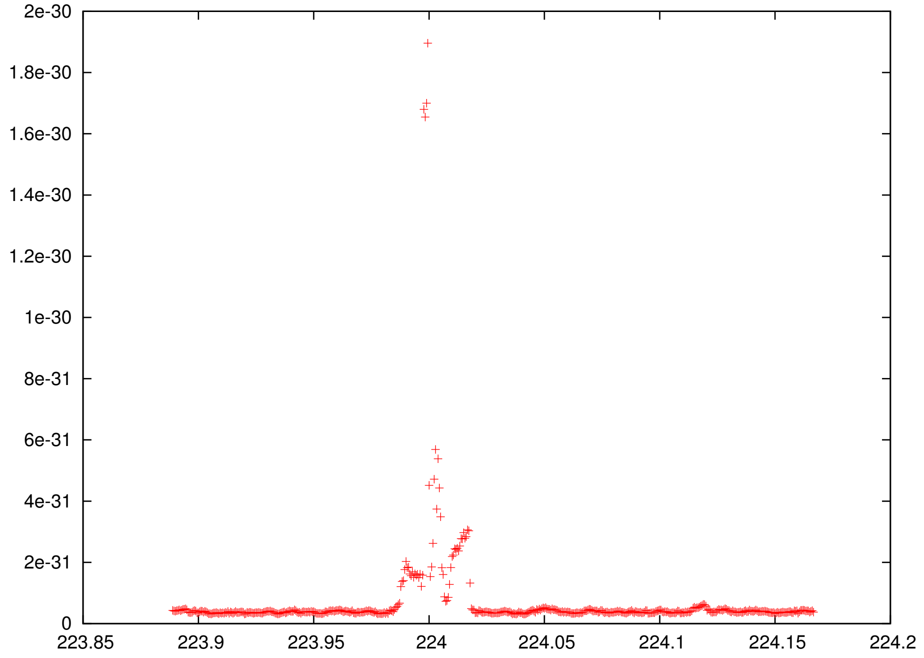

Figure 4: Maximum over declination band 1 of weighted power average

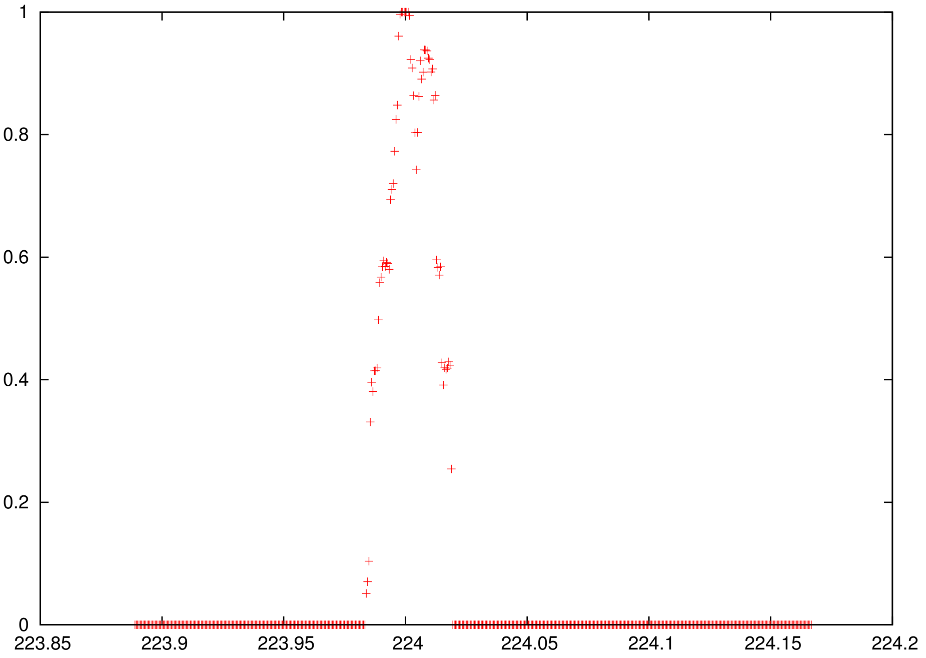

Figure 5: Maximum mask ratio over declination band 1

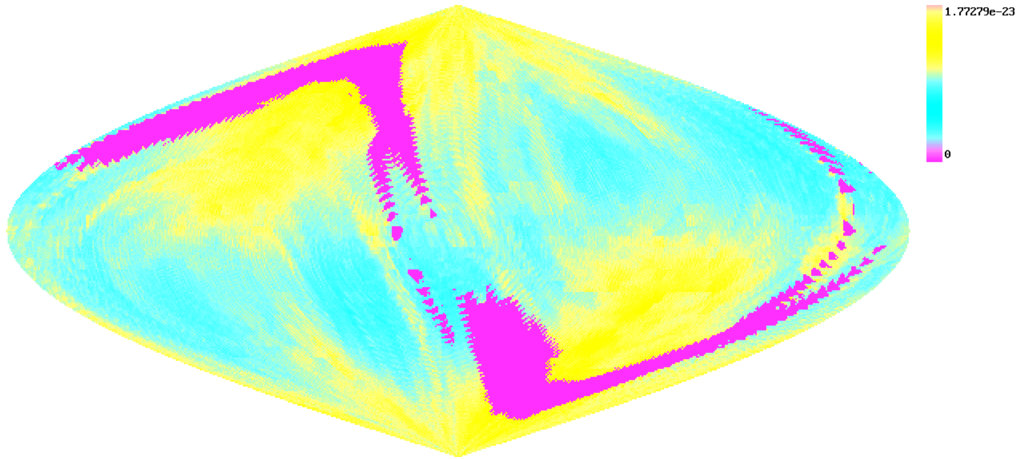

Figure 6: Effects of veto of four highest bins around 224 Hz

8 `CutOff`

Since even a perfect interferometer has different sensitivities to a particular direction and polarization during the run it is reasonable to discard SFTs that contribute the least (or pollute the most) information.

The algorithm currently used by PowerFlux is to veto all SFTs with estimated noise level larger than a sky position and polarization dependent `CutOff` value. The implementation takes into account the following considerations:

- The noise level is the noise level of AM demodulated signal
- It is not necessary to compute `CutOff` for each point in the fine grid as AM modulation changes slowly across the sky. Instead one `CutOff` value is computed per element of patch grid.
- weighted mean is better able to incorporate data with high noise level, the `CutOff` value is doubled when performing weighted averaging.

The `CutOff` is computed as follows: let σ_i be the estimated noise levels of the data, sorted from smallest to largest. Then

$$\texttt{NCutOff} := \mathrm{argmin}\ {}_k \frac{1}{k}\sqrt{\sum_{i=1}^{k} \sigma_i^2}$$

and

$$\texttt{CutOff} := \sigma_{\texttt{NCutOff}}$$

9 Modulation: AM response and Doppler shift

As the Earth travels in its orbit and rotates about its axis the interferometer changes position and velocity with respect to fixed direction in space. This produces amplitude and frequency modulation on the incoming signal which must be reversed in order to obtain good estimates. Fortunately, both effects can be considered slowly varying on the scale of single 1800 second SFT and thus can be corrected for, using only power data.

9.1 Doppler shift

Since the speed of interferometer is much smaller than the speed of light the frequency modulation is dominated by a component linear in the velocity of interferometer and direction of the sky. PowerFlux uses the LAL library to compute the coefficients of dependence on sky direction for GPS times of all segments used in the analysis. The result is stored in the array.

When a Doppler shift is needed PowerFlux takes a dot product of a particular direction in the sky (i.e. element of the fine grid) and the stored coefficients.

9.2 AM response

Gravitational waves are quadrupolar in nature and therefore the amplitude modulation (of SFT values) resulting from varying position of interferometer relative to the source is given by a polynomial of 4th degree in source coordinates (taken in Descartes (Cartesian), not spherical, system of coordinates).

Existing LAL functions perform adequately fast when one needs the response for 100 directions in the sky, but they are too slow to call for each point of fine grid separately. On the other hand, they have been well tested and are known to produce correct results.

The approach taken by PowerFlux is to call LAL functions for AM response for a small number of points in the sky and then make an exact linear fit to precomputed values of various monomials of degree 4 and smaller. Once this linear fit is obtained it is easy to compute AM response for arbitrary location in the sky using polynomial expression that we obtained.

The linear fit is performed for each SFT segment separately.

10 Main loop

In the main loop PowerFlux iterates over elements of fine grid, polarizations and frequency bins and computes weighted sum of powers for each combination.

Since the size of fine grid is quite large and we usually consider 501 frequency bins this task is split up into the outer and inner loops: the outer loop iterates over elements of the patch grid and the inner loop processes all points of fine grid corresponding to a particular element of the patch grid. The weighted means accumulated for a particular patch are then analyzed further and the diagnostics and upper and lower limits derived using Feldman-Cousins algorithms are stored in a separate array. The intermediate data is then discarded.

A number of aggregate quantities are computed and logged or stored for later output, in particular sky maps of maximums over frequency and maximums over particular areas in the sky.

11 Output

While the upper and lower limits obtained by PowerFlux are output after the main loop finishes, there is a large amount of diagnostic output produced as PowerFlux proceeds in its computation.

The files separate in three groups:

- Log files have extension `.log`. Two such files are produced:
  - `powerflux.log` contains line-wise records reporting input parameters, version of PowerFlux used to process data, various intermediate quantities as well as summary of results (in particular maximum values of the upper limits for several latitude bands).
  - `file.log` contains line-wise records detailing all raw data files produced during the run. A keyword - floats or doubles points out the format of the files.
- Data files have extension `.dat` and are flat arrays of numbers. The particular number format is specified in the `file.log` file. The endianness used is the endianness of the machine the computation was performed on.
- Graphics files are produced in `PNG` format. At the moment PowerFlux knows how to produce 2d-plots and sky maps.

11.1 Example output

11.1.1 Fake noise

Fake noise SFTs were created by Greg Mendel for comparison between StackSlide and PowerFlux codes. They contain uncorrelated gaussian noise in arbitrary units. There were 999 SFT segments covering time interval 742332616-744129016 (GPS seconds).

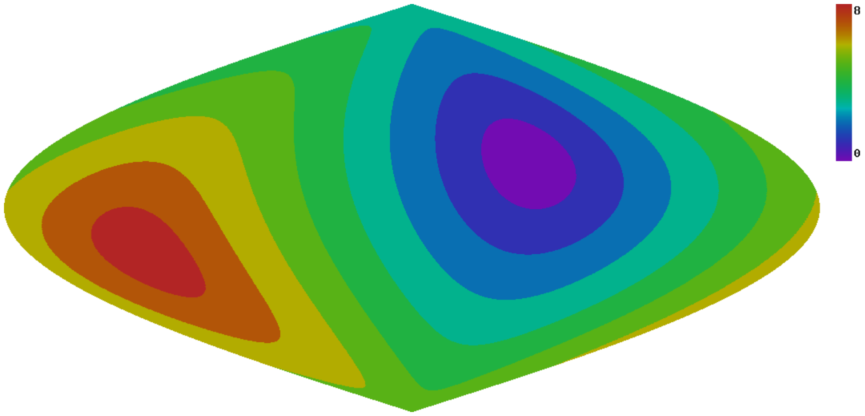

Figure 7: Sky bands for fake noise SFTs

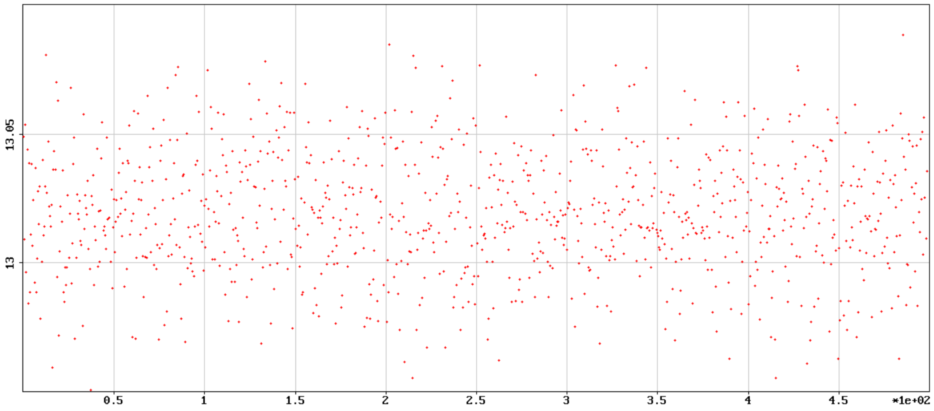

Figure 8: TMedians

The figures 8 and 9 contain plots of TMedian and FMedian values. No structure is visible as we would expect for random data.

On the other hand the plots of maximum of signal strength (figure 10) and maximum of upper limit on strain (figure 11) do show structure. The cause for this is non-uniform distribution of Doppler shifts (the time interval does not span an entire year) and non-uniform sensitivity of the detector to the different sky regions as shown on the total weight plot (figure 13).

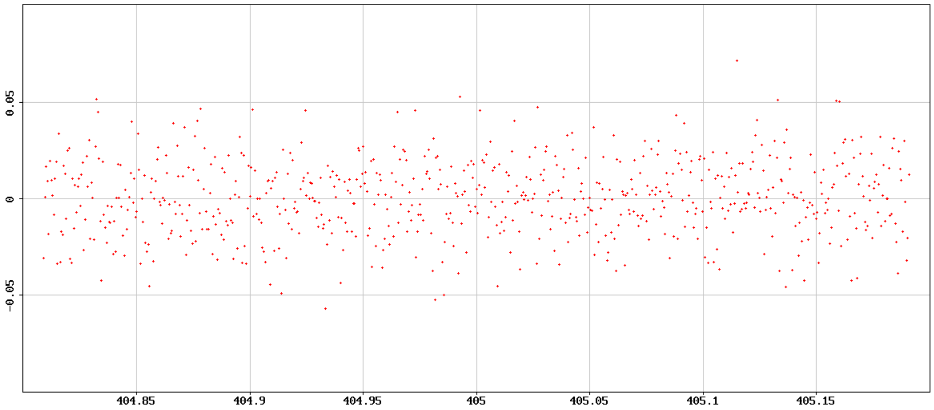

Figure 9: FMedians

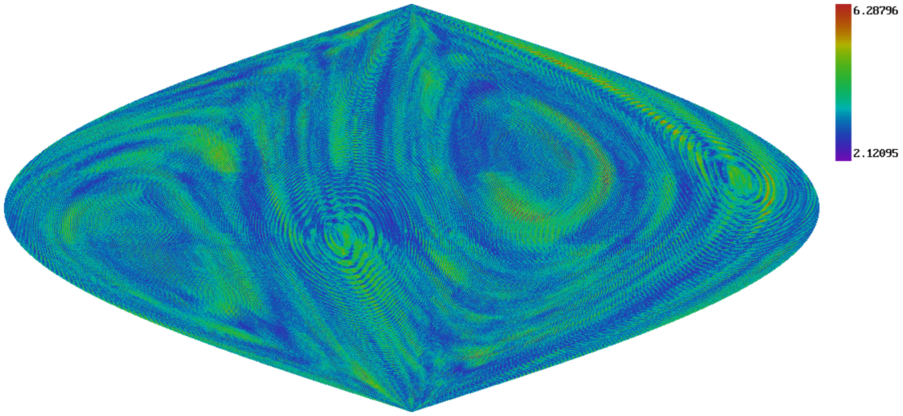

Figure 10: Maximum of signal strength for cross polarization

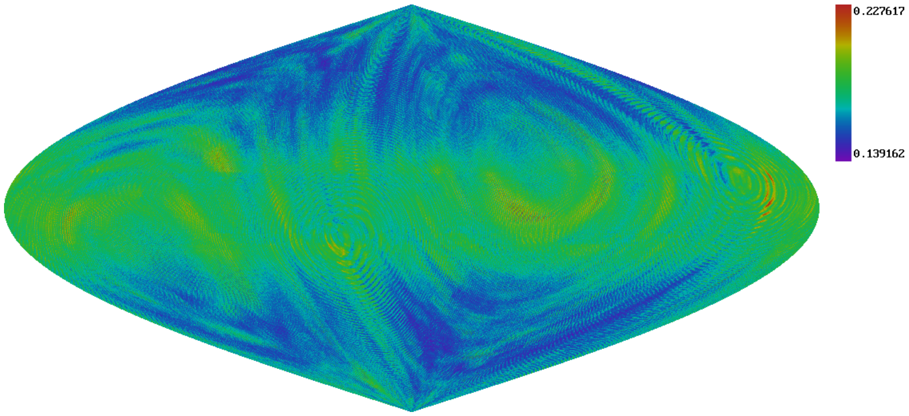

Figure 11: Maximum of upper strain limit for cross polarization

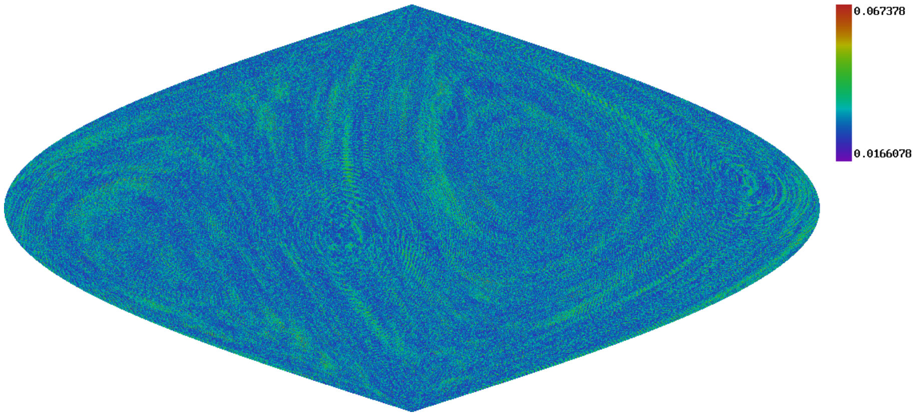

Figure 12: Kolmogorov-Smirnov test values for compliance to normal distribution

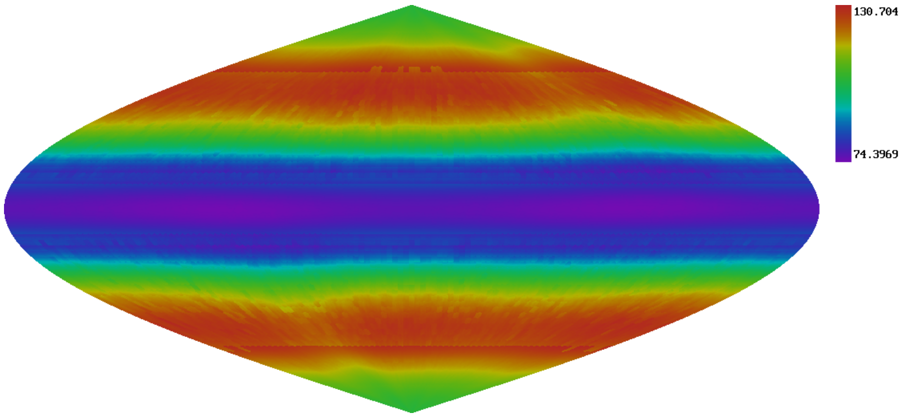

Figure 13: Total weight for cross polarization

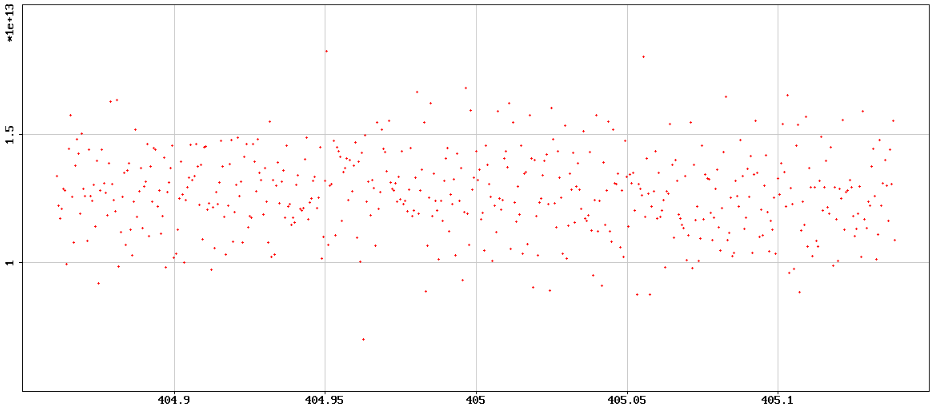

Figure 14: Example of weighted power average for a single sky position

11.1.2 Injected pulsar 2 (S3 H1)

The following plots show the results of running PowerFlux on the stretch of data when pulsar 2 was injected during S3 H1 run. Total of 2595 SFTs were used between GPS times 751704390 and 757697392 (these are times of start of first and last SFT segments).

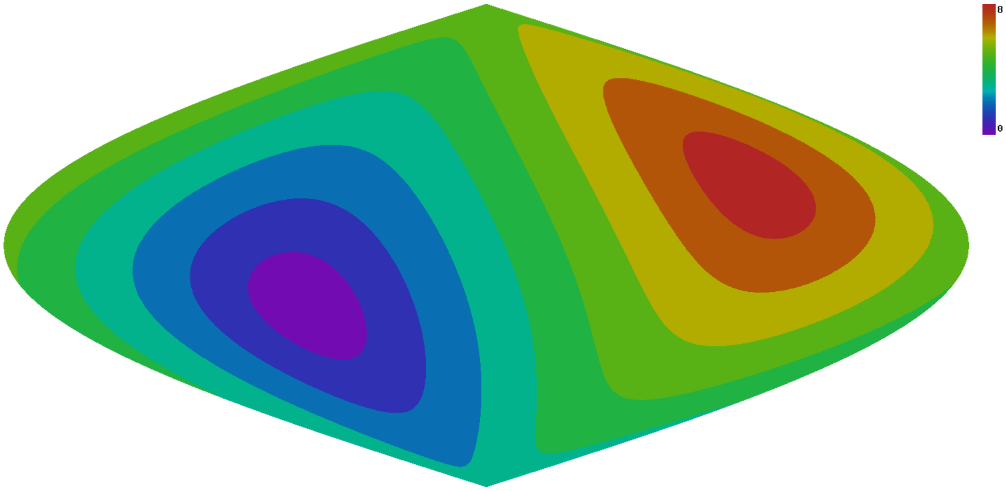

Figure 15: Sky bands for S3 H1

Note the gap in the TMedians plot (figure 16) - it is due to pulsar 2 signal being turned off during the run so the corresponding segments were excluded from the analysis.

The pulsar signal shows up clearly on the plots of maximum signal strength (figure 18) and strain upper limit (figure 19) as well as on an example plot of average power (figure 22) for a sky position chosen to be near the location of the incoming signal.

Note that the plot (figure 20) of Kolmogorov-Smirnov test for compliance with gaussian distrubtion in weighted power average (which also serves to check that the distribution parameters were estimated correctly) is similar to the Kolmogorov-Smirnov plot for fake noise (figure 12). This is because the Kolmogorov-Smirnov test checks for deviation between emperical distribution and ideal one and so deemphasizes tail behaviour.

The pattern in the total weight plot (figure 21 is remarkably different from the one for ideal noise (figure 13). This is due to variation in quality and quantity of the data for H1 during S3 run.

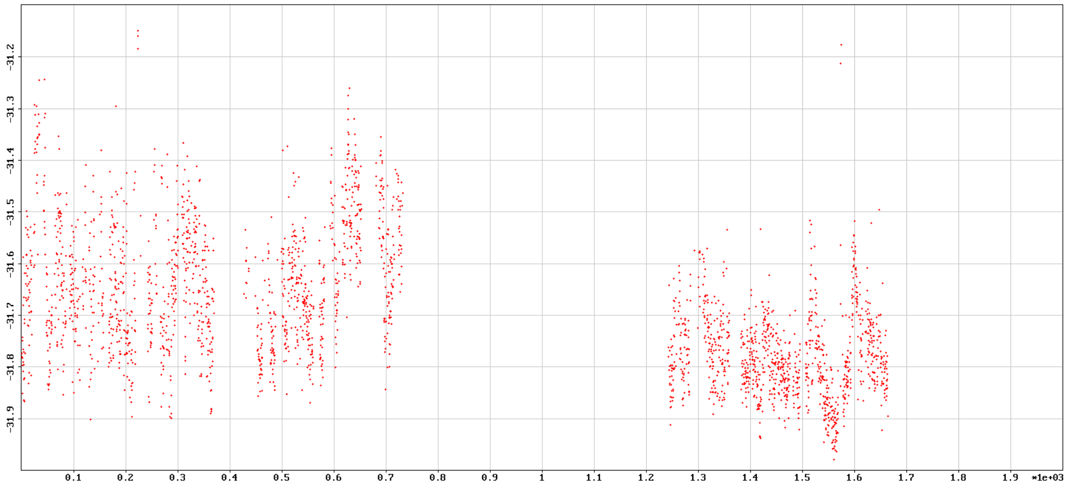

Figure 16: TMedians

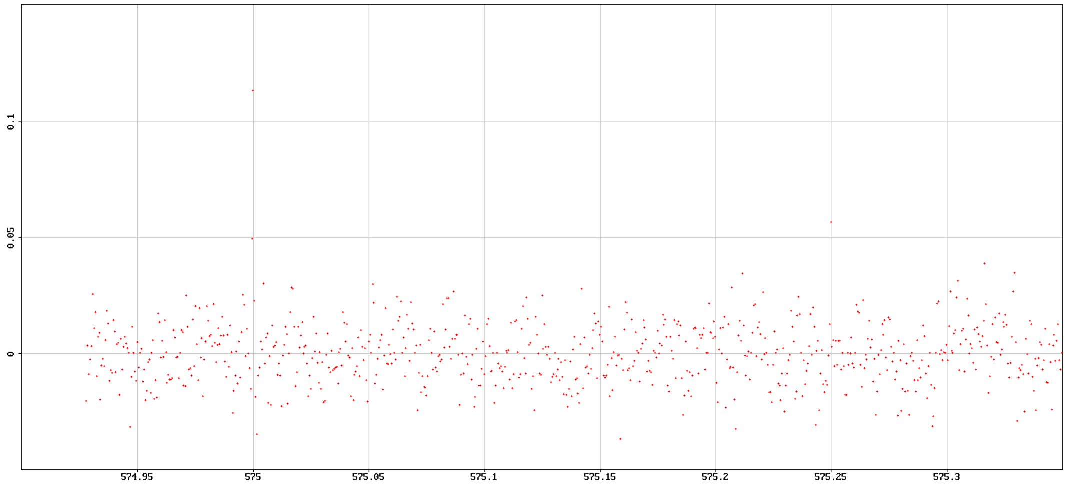

Figure 17: FMedians

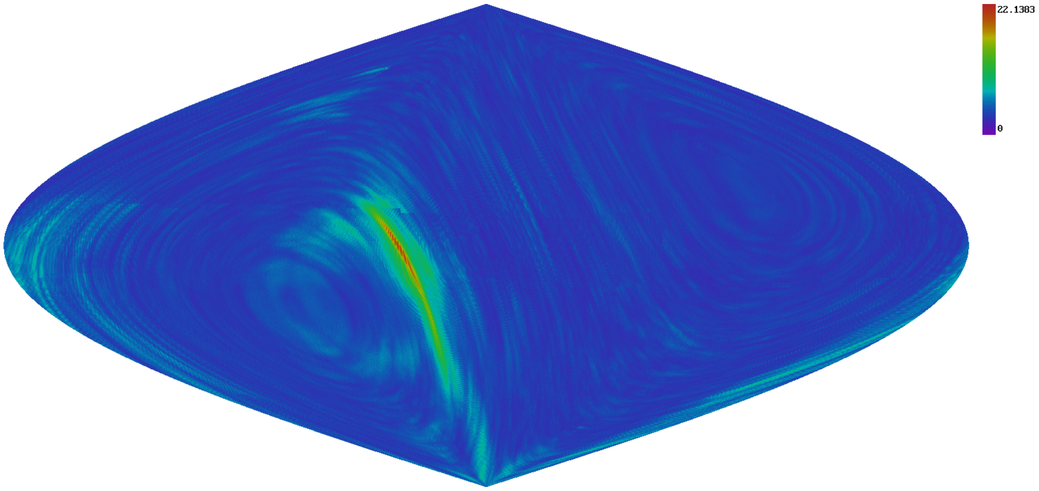

Figure 18: Maximum of signal strength for cross polarization

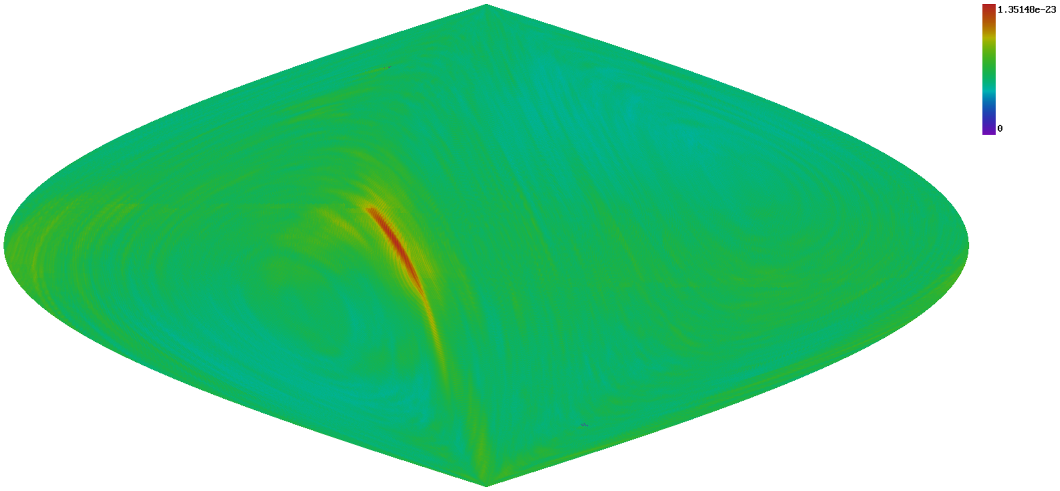

Figure 19: Maximum of upper strain limit for cross polarization

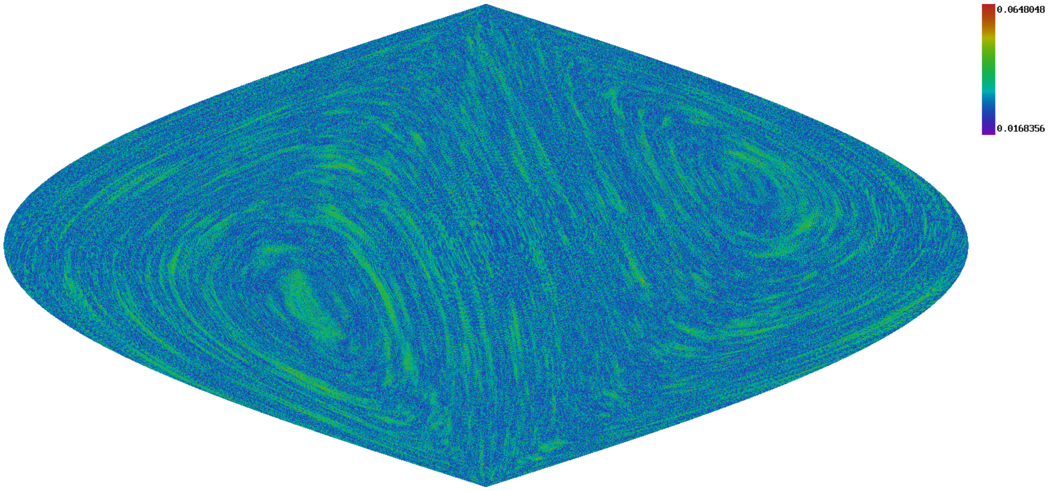

Figure 20: Kolmogorov-Smirnov test values for compliance to normal distribution

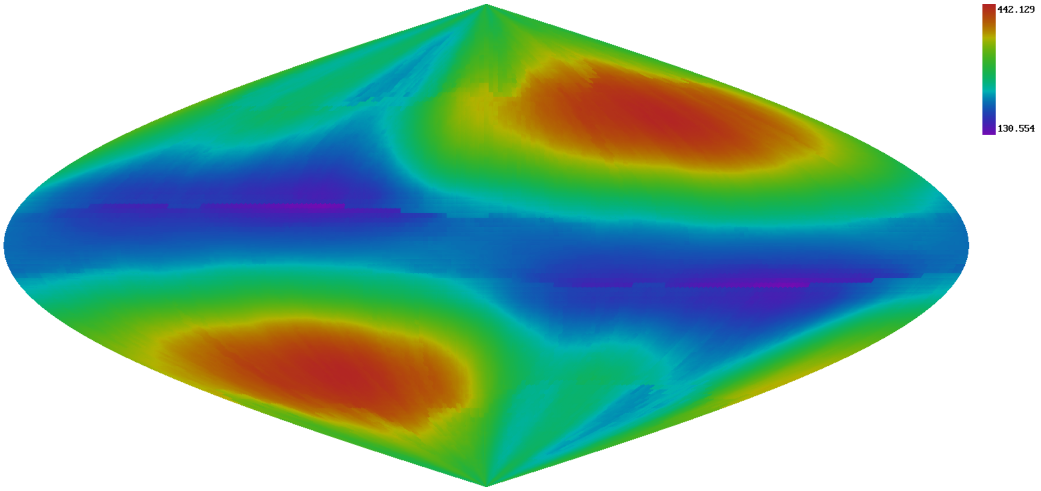

Figure 21: Total weight for cross polarization

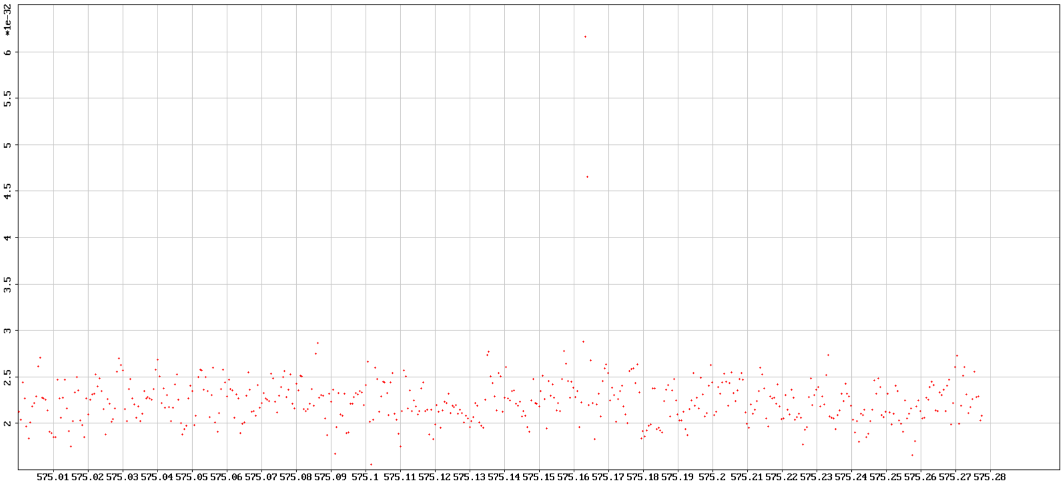

Figure 22: Example of weighted power average for a single sky position

Figure 23 shows average of TMedians according to the half hour the data was taken. We see that there is 30% variation in power, with worst occuring near 0 hours UTC time.

Figure 24 shows TMedians summed according to half hour the data was taken. We see that the variation exhibits the same pattern, but the variation is 150% - the time near 10 hours UTC but a lot more science mode data.

12 Sky bands

PowerFlux has the ability to produce summary results aggregated over a particular portion of the sky. Each sky bin is assigned a *band number* which is used to determine which portion of the sky the sky bin belongs too.

In principle, this marking can be arbitrary and it is easy to add code to draw different shapes. At the moment the only option implemented is the ability to mark several bands according to an angle to a particular vector.

This is useful for distinguishing upper limits with different influence of line removal. For this purpose PowerFlux can automatically compute *band axis* vector.

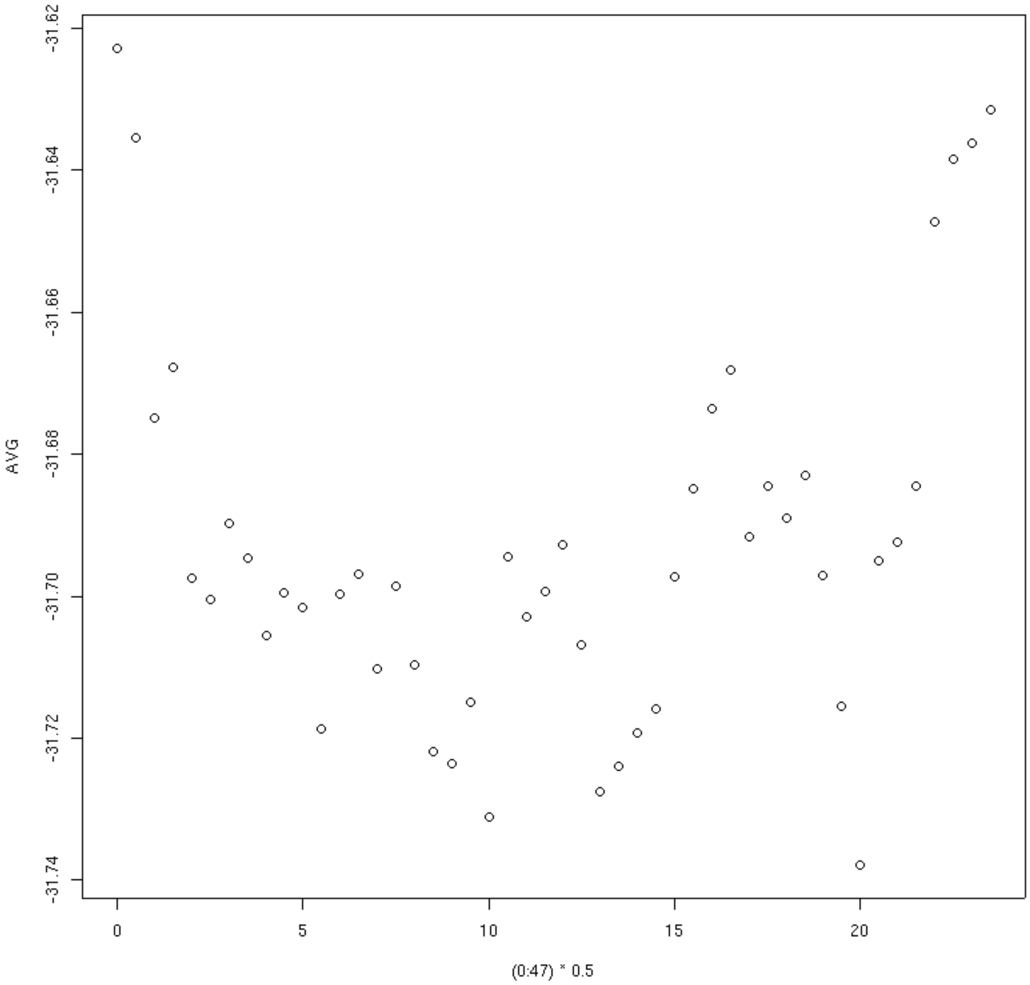

Figure 23: TMedians averaged according to half hour the data was obtained in (UTC time).

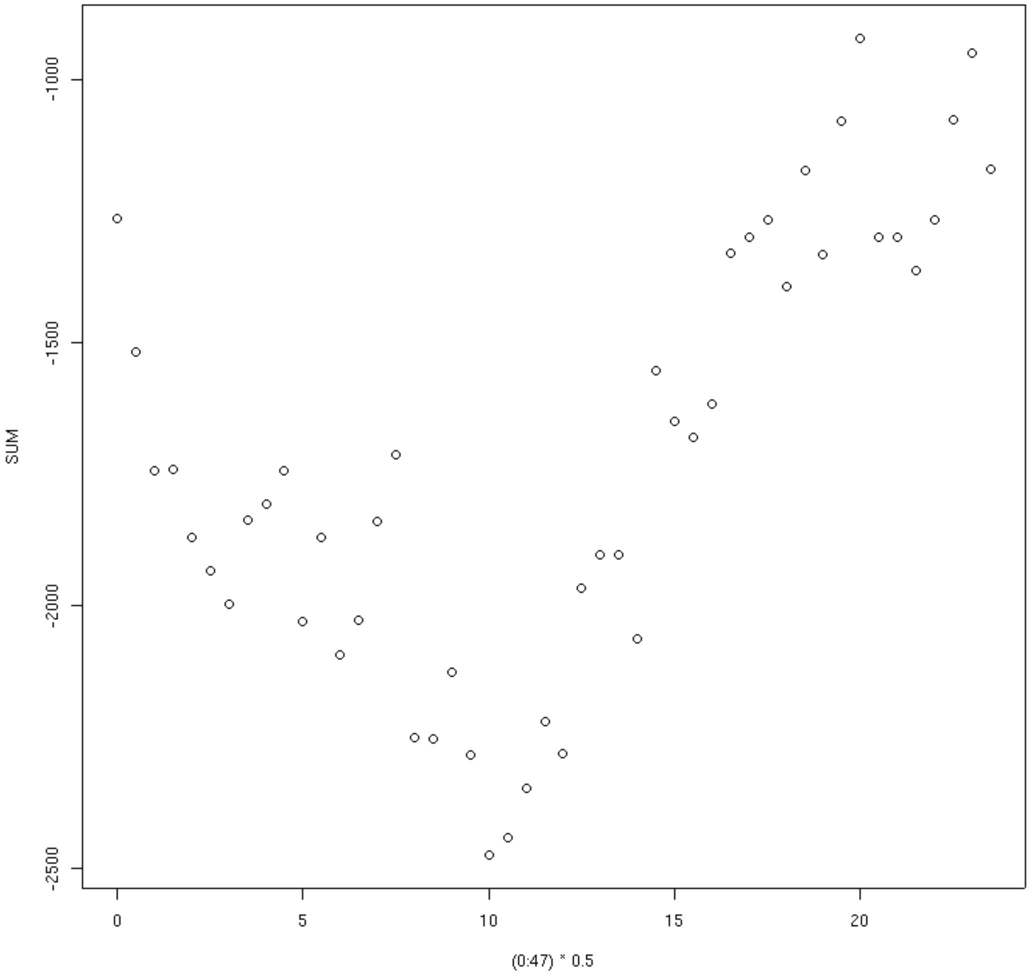

Figure 24: TMedians summed according to half hour the data was obtained in (UTC time).

12.1 Orienting bands

The following plots illustrate the influence of partitioning of sky into bands for setting upper limits. S3 L1 data was used.

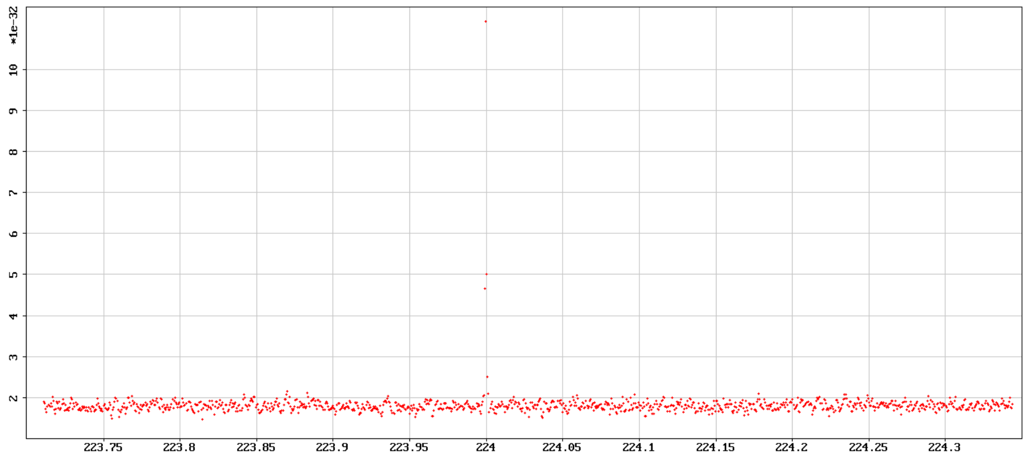

Figure 25: Background noise. Y axis is power in arbitrary units.

The figure 25 shows the background noise of the band 223.888889-224.138889 Hz. All 4 highest power bins were masked in this study.

The figure 26 shows the effect of masking - deep blue area consists of points on the sky for which at least one frequency bin is unusable as more than 80% of the data required to computed it was masked away.

Figure 27 shows partition of the sky into 9 bands in such a way that the middle band (band 4) contains the entire blue area consisting of masked points. This partition was calculated as follows:

First band axis vector is computed as a cross product of ecliptic pole and average detector velocity vector. The band axis vector thus represents the average direction towards the sun during the run.

The bands then are taken perpendicular to band axis vector, spaced equally in the angle to the band axis vector.

Figure 31 illustrates the same point for Monte carlo simulation. The x axis is the projection of the sky location onto the band axis vector. Note that large negative excesses are aggregated near small values of the projection.

Figures 28 and 29 show strain upper limit for cross polarization for band 0 (far from masked area) and band 4 (contains masked area).

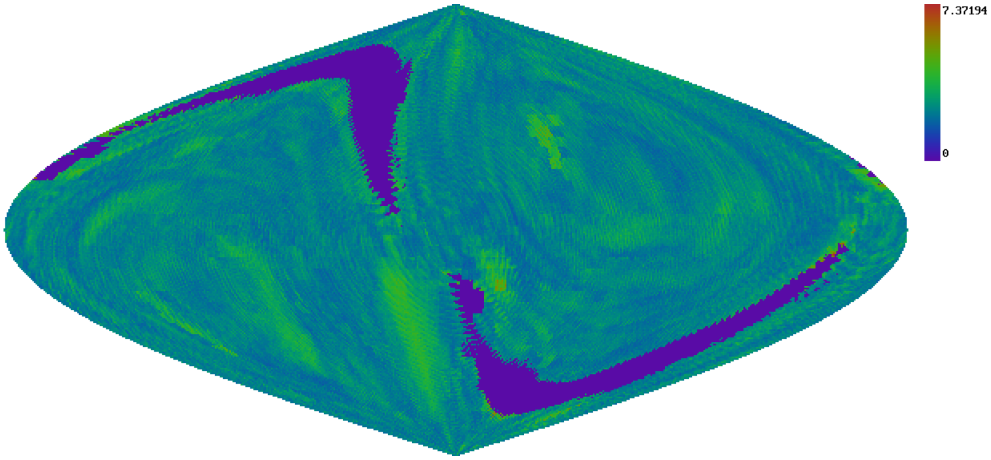

Figure 26: Signal detection strength. Deep blue area consists of masked points.

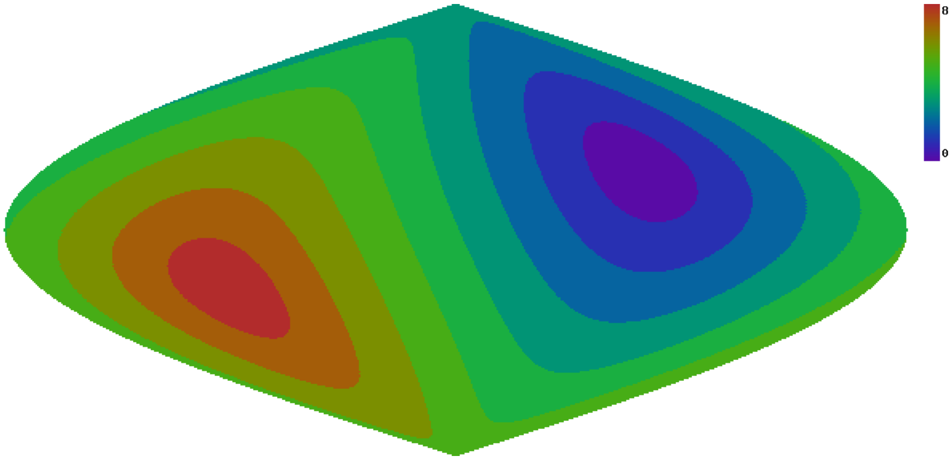

Figure 27: Partition into bands. Map is in equatorial coordinates

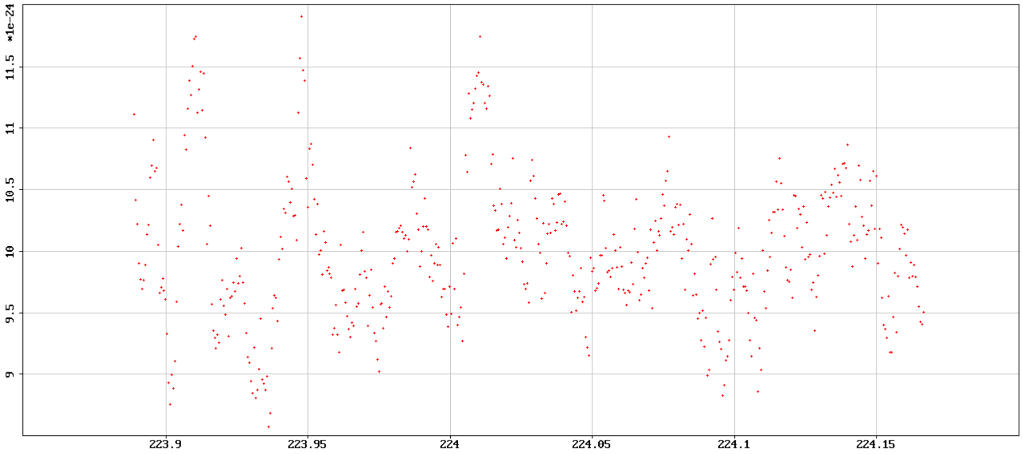

Figure 28: Cross upper strain for band 0 (maximum taken over sky locations in the band).

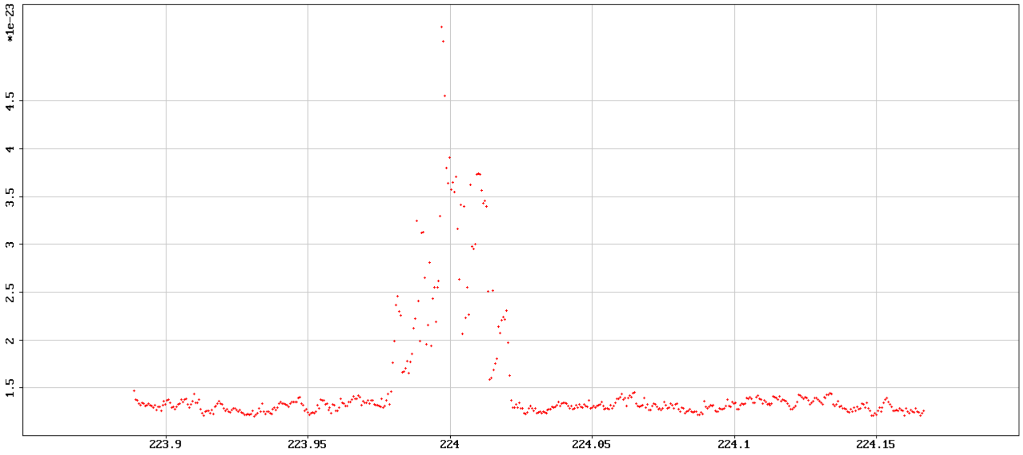

Figure 29: Cross upper strain for band 4 (maximum taken over sky locations in the band).

The influence of masked points on the upper limit of band 4 is highly visible. Moreover, all the bins in band 0 have upper limit less than 1.2e-23 while all bins in band 4 are above that value - even those seemingly not affected by masked points.

We can separate the sky into two regions in relation to masked area: "good" - bands 0,1,2 and 6,7,8 and "bad" - bands 3,4,5. Thus it appears that "good" region not only avoids the issue of masked area completely but also has better overall limits.

12.2 Monte Carlo simulation

The following plots show results of Monte Carlo simulation run performed with S3 L1 data. 8000 separate injections were made, each processed with a separate execution of PowerFlux program. 20 injections were made in each 0.25 Hz band between 200 Hz and 300 Hz. The signals were simulated using power only, with random phases injected for each SFT. Each signal was linearly polarized with random orientation.

In order to speed up the computation only the circular region around the location of injection was processed. The radius of this region is 0.1 radians. Thus the upper limits quoted are maximum for this region, not the entire sky.

Figure 30 shows strain upper limit excess (limit established minus injected value) versus frequency of injected signal.

As mentioned previously, figure 31 shows excess versus projection of the location of the injected signal to band axis.

Figure 32 shows dependence of strain on the number of masked points. The large negative excesses occur only when the number of masked points is large.

Plot 33 shows reconstructed frequency error versus decimal logarithm of injected strain.

Figure 34 shows reconstructed frequency error versus frequency of injected signal, zoomed in Y axis to show detection accuracy. The mean absolute deviation of the sample (after excluding points with small injected strains or small projection (absolute value less than 0.3) onto band axis) is 0.00081.

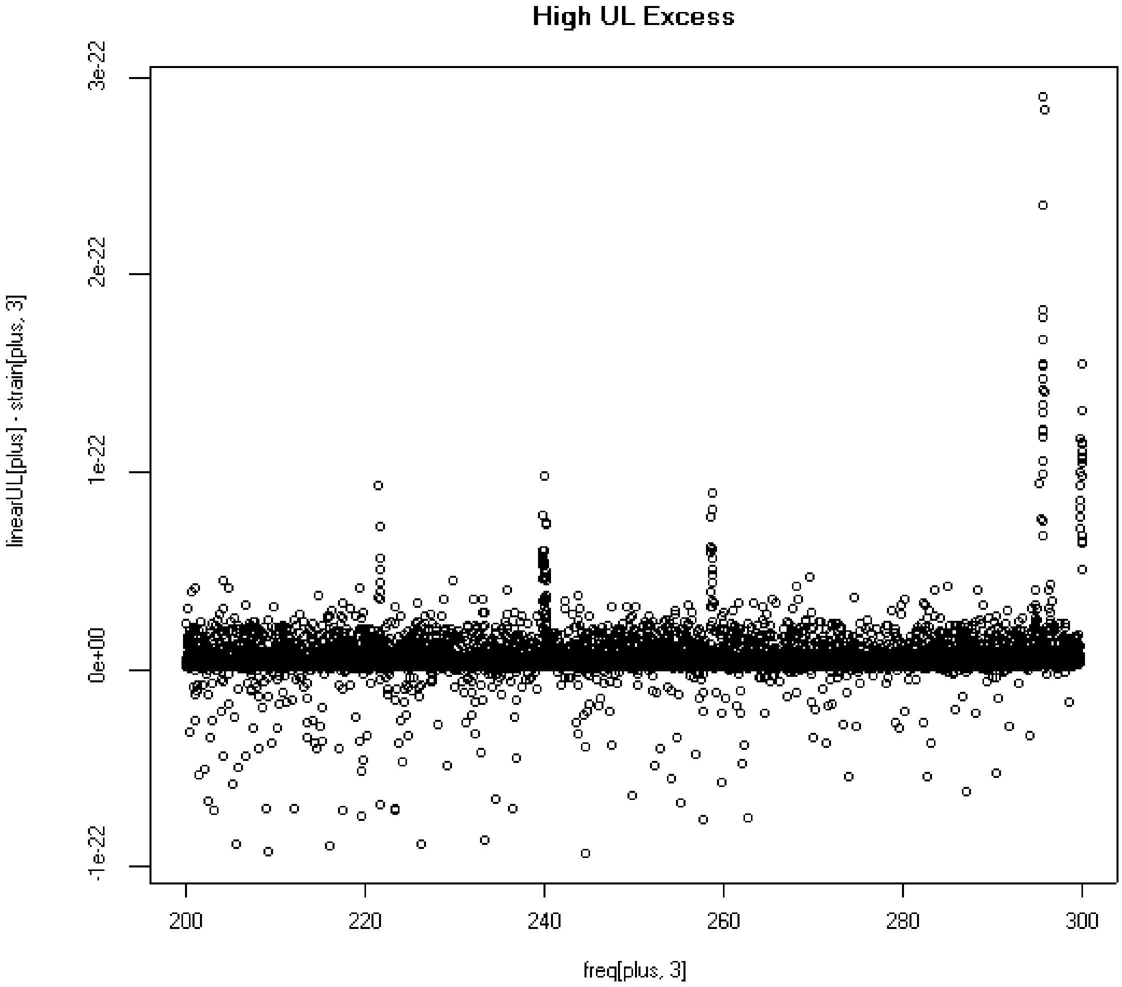

Figure 30: Upper limit power excess

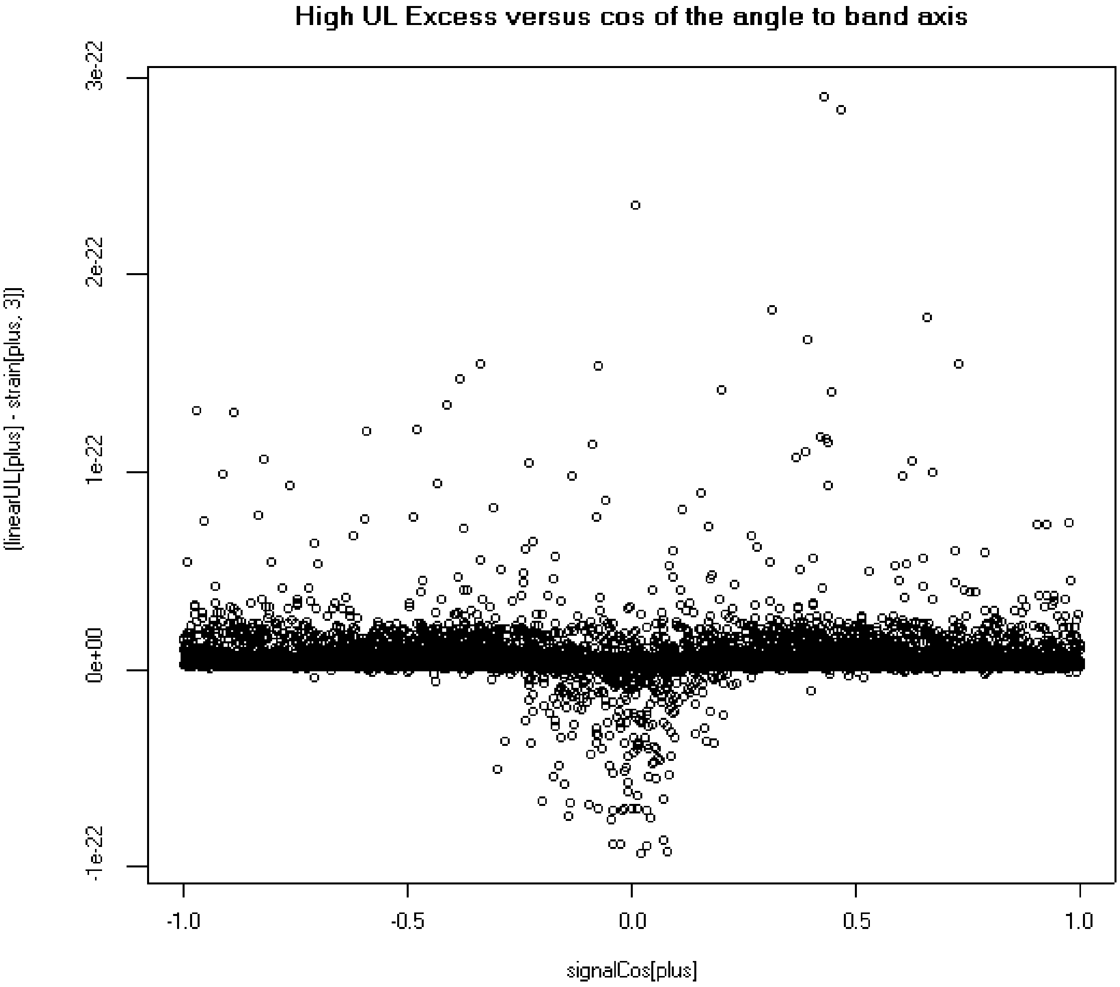

Figure 31: Upper limit power excess versus cos of the angle to band axis

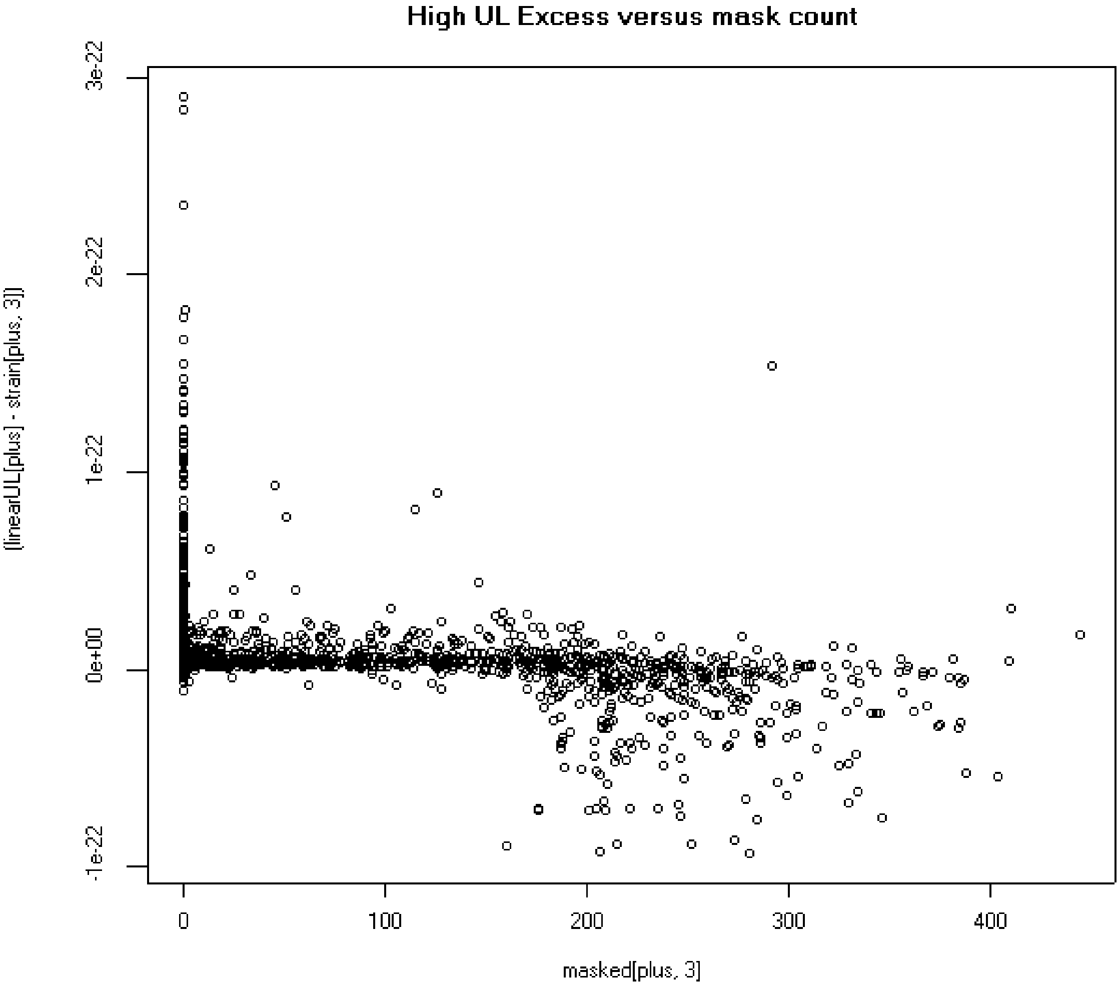

Figure 32: Upper limit power excess versus number of masked points

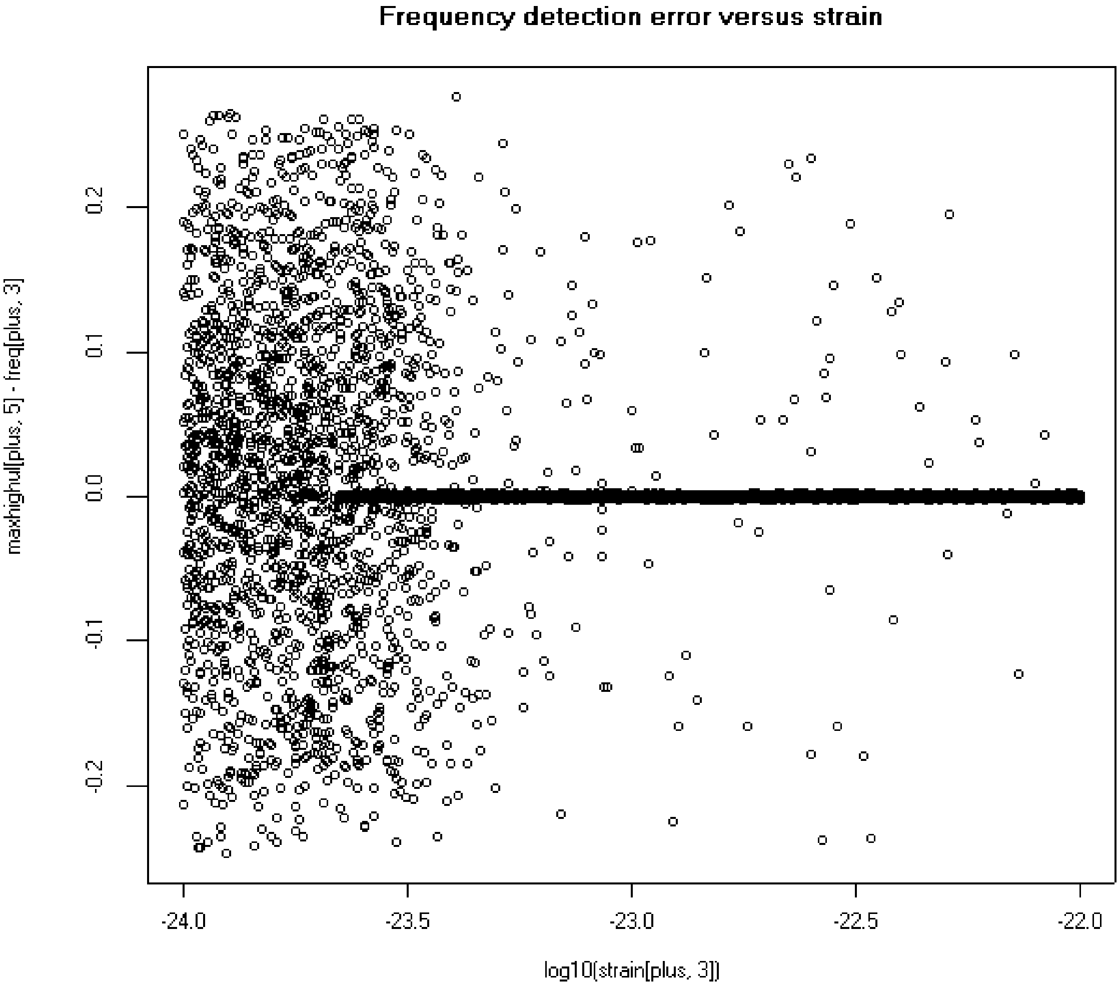

Figure 33: Frequency error versus strain

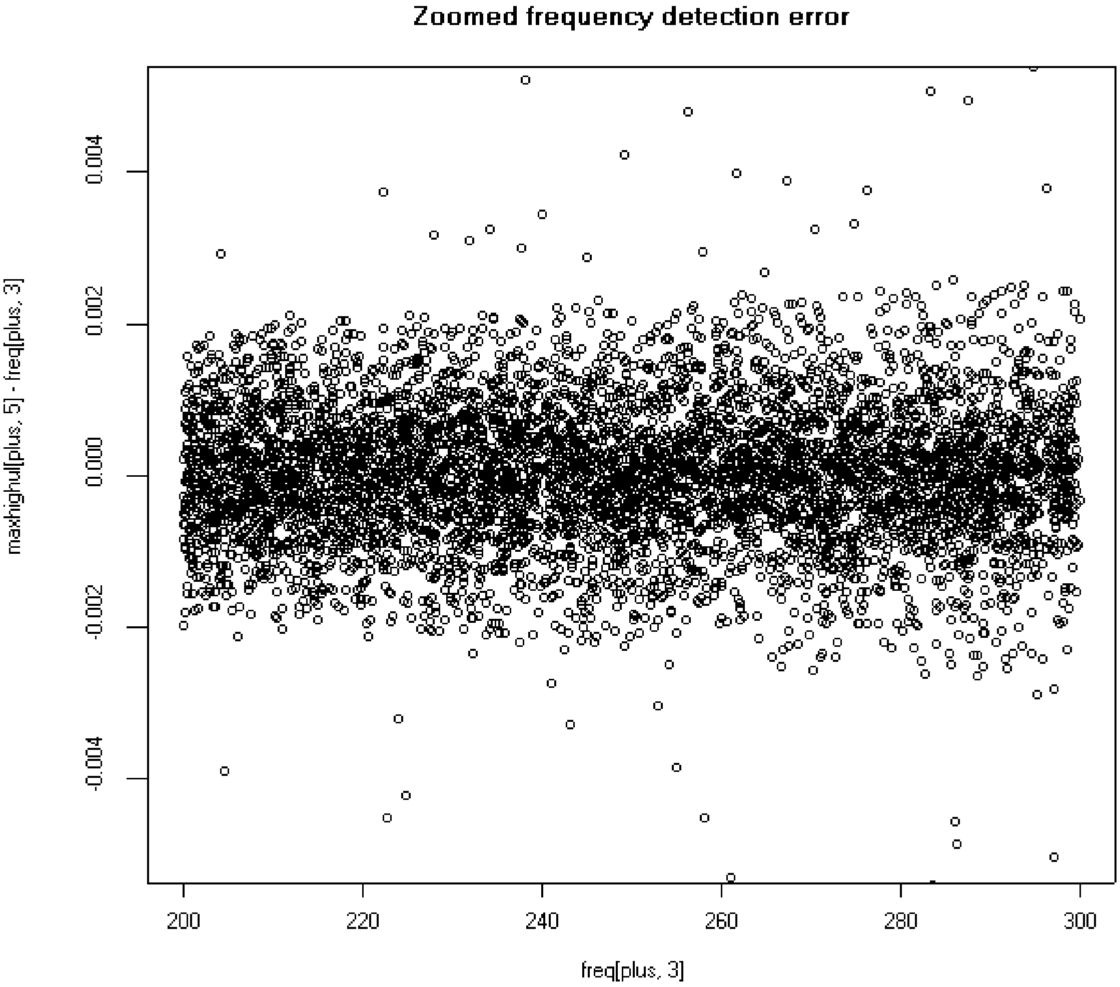

Figure 34: Zoomed frequency error versus injected frequency

12.3 Spindown dependent band assignment

The previous scheme provides a simple way to partition the sky into areas that have similar quality of results. However, it breaks down when spindown is taken into account - for example, signals with high spindown would spread their power among many frequency bins and every sky location should produce good results.

An alternative approach is to construct a quality function which values would correlated with the actual power spread of signal as received by interferometer.

The frequency as received by the detector differs from the original frequency of the gravitational wave by the following components:

- Doppler shift: $\frac{\vec{v}(t)\cdot\vec{r}f}{c}$ where $\vec{v}(t)$ is the velocity of the detector in frame of the gravitational wave, $\vec{r}$ is the unit vector in the direction of incoming wave and c is the speed of light constant.

- Spindown: $s(t-t_0)$ where t_0 is the time of the initial measurement and s is the spinup (or $\dot{f}$) parameter.

Thus, throughout the observation period the observed frequency varies as:

$$f(t) = \left(1 + \frac{\vec{v}(t)\cdot\vec{r}}{c}\right)(f + s(t-t_0))$$

And the bin number varies as:

$$N(t) = \left(1 + \frac{\vec{v}(t)\cdot\vec{r}}{c}\right)(f + s(t-t_0))w^{-1}$$

where w is the frequency resolution of SFT (currently 1/1800 Hz).

12.3.1 Approximation

In the following we will assume that rotational motion of Earth play negligible role and concern ourselves only with orbital motion.

This also allows us to assume that the power received by the detector is constant, or, more realistically, that the daily irradiation does not significantly vary.

Thus $P(t) = P_0$, $E(t) = E_0 + P_0(t-t_0)$.

We will also approximate the Earth orbit around the Sun with a circle and replace detector velocity with its average throughout the day.

Then the detector velocity vector can be represented as:

$$\vec{v}(t) = \vec{v}_0 \cos(\omega t) + \vec{v}_1 \sin(\omega t)$$

where $\vec{v}_0 \perp \vec{v}_1$ and $\omega = \frac{2\pi}{365 \text{ days}}$.

Let us also define $\vec{u}$ as:

$$\dot{\vec{v}}(t) = \vec{u} \times \vec{v}(t)$$

i.e. vector $\vec{u}$ points in the direction of the North ecliptic pole (TODO: check that not reverse) and has magnitude $|\vec{u}| = \omega$.

Therefore,

$$\dot{N}(t) = \left(1 + \frac{\vec{v}(t) \cdot \vec{r}}{c}\right) sw^{-1} + \frac{\dot{\vec{v}}(t) \cdot \vec{r}}{c}(f + s(t - t_0))w^{-1}$$

$$\dot{N}(t) = \left(1 + \frac{\vec{v}(t) \cdot \vec{r}}{c}\right) sw^{-1} + \frac{(\vec{u} \times \vec{v}(t)) \cdot \vec{r}}{c}(f + s(t - t_0))w^{-1}$$

$$\dot{N}(t) = sw^{-1} + \left(\frac{\vec{v}(t)}{c}s + \frac{\vec{u} \times \vec{v}(t)}{c}\left(f + s(t - t_0)\right)\right) \cdot \vec{r}w^{-1}$$

$$\dot{N}(t) = sw^{-1} + \left(\frac{\vec{v}(t)}{c}s + \frac{\vec{u} \times \vec{v}(t)}{c}f\right) \cdot \vec{r}w^{-1} + \frac{\vec{u} \times \vec{v}(t)}{c}s(t - t_0) \cdot \vec{r}w^{-1}$$

For the signal to be stationary $\dot{N}(t)$ must be 0. Neglecting the last term (which contains a product of s and $\vec{u}$) and first term in brackets (which contains the product of spindown and ratio of $\vec{v}$ to speed of light) we obtain the following equation:

$$\dot{N}(t) \approx sw^{-1} + \frac{\vec{u} \times \vec{v}(t)}{c}f \cdot \vec{r}w^{-1} = 0$$

12.3.2 Quality function

While we could use the value of $\dot{N}(t)$ itself as a quality function, it is more convenient to use

$$S := s + \frac{\vec{u} \times \vec{v}_{\text{avg}}}{c}f \cdot \vec{r}$$

in actual implementation. Here $\vec{v}_{\mathrm{avg}}$ is the average detector velocity during the run.

S has units of spindown and describes average cumulative signal drift.

Values of S that are high enough to guarantee accurate identification by PowerFlux can be expressed as

$$S_{large} = \frac{N_{bins}}{T_{obs} \cdot T_{coh}}$$

Where T_{obs} is the timebase of observation, T_{coh} is time interval used to produce SFTs and N_{bins} is the total drift in bin count.

Usually $N_{bins} = 5$ is a good value, so for a 1 months run with 1800 sec SFTs we have

$$S_{large} = 1.1e - 9\mathrm{Hz/s}$$

12.3.3 Results of Monte-Carlo test

Plots 35 and 36 show results of Monte-Carlo software injections into background data collected by H1 interferometer during S4 run.

The frequency bin footprint was computed as the highest bin injected into minus lowest one plus one.

The excess strain refers to the strain limit as established by PowerFlux minus the strain of the injected signal.

Red points on plot 36 correspond to injections with spindown less than $5e - 10$, while black points correspond to spindown less than $1e - 10$.

13 Running time

The performance of a program on a cluster can be significantly different from its performance on a standalone computer due to the need to share resources. Therefore we present a few graphs of PowerFlux running time on a Medusa cluster at UWM to give a feeling for the computation resources required to perform a full analysis.

Plot 37 shows running time in seconds versus band analysed. We see a general increase in computational requirements with frequency - this corresponds to the quadratic increase in number of points analysed. This is confirmed by plot 38 of the ratio of running time to number of points in the fine grid.

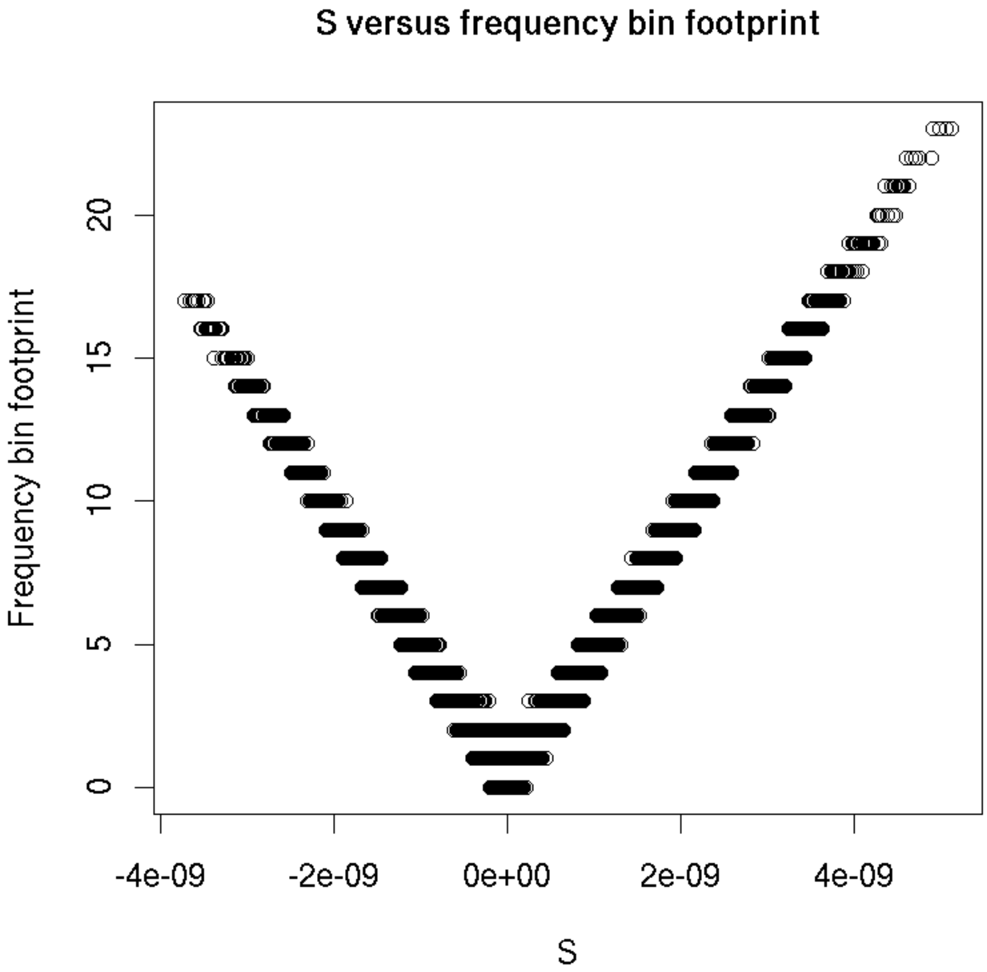

Figure 35: S versus frequency bin footprint

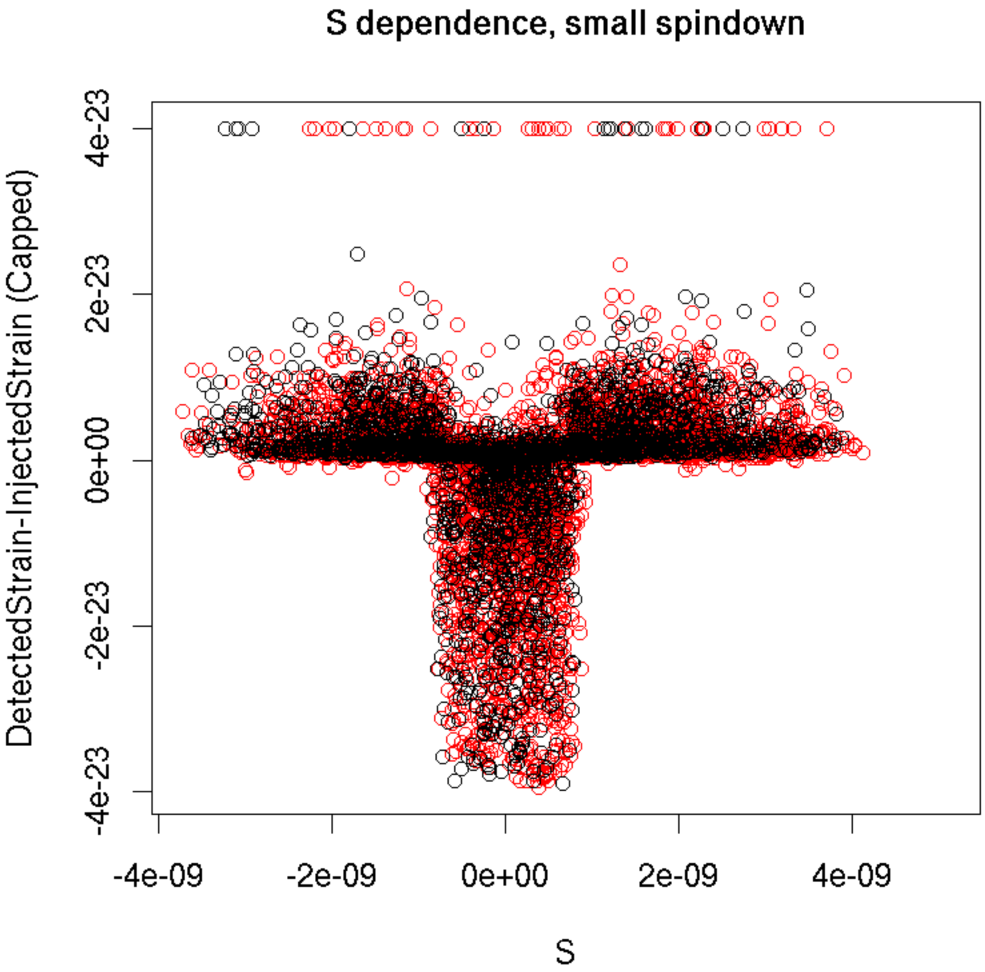

Figure 36: S versus excess strain

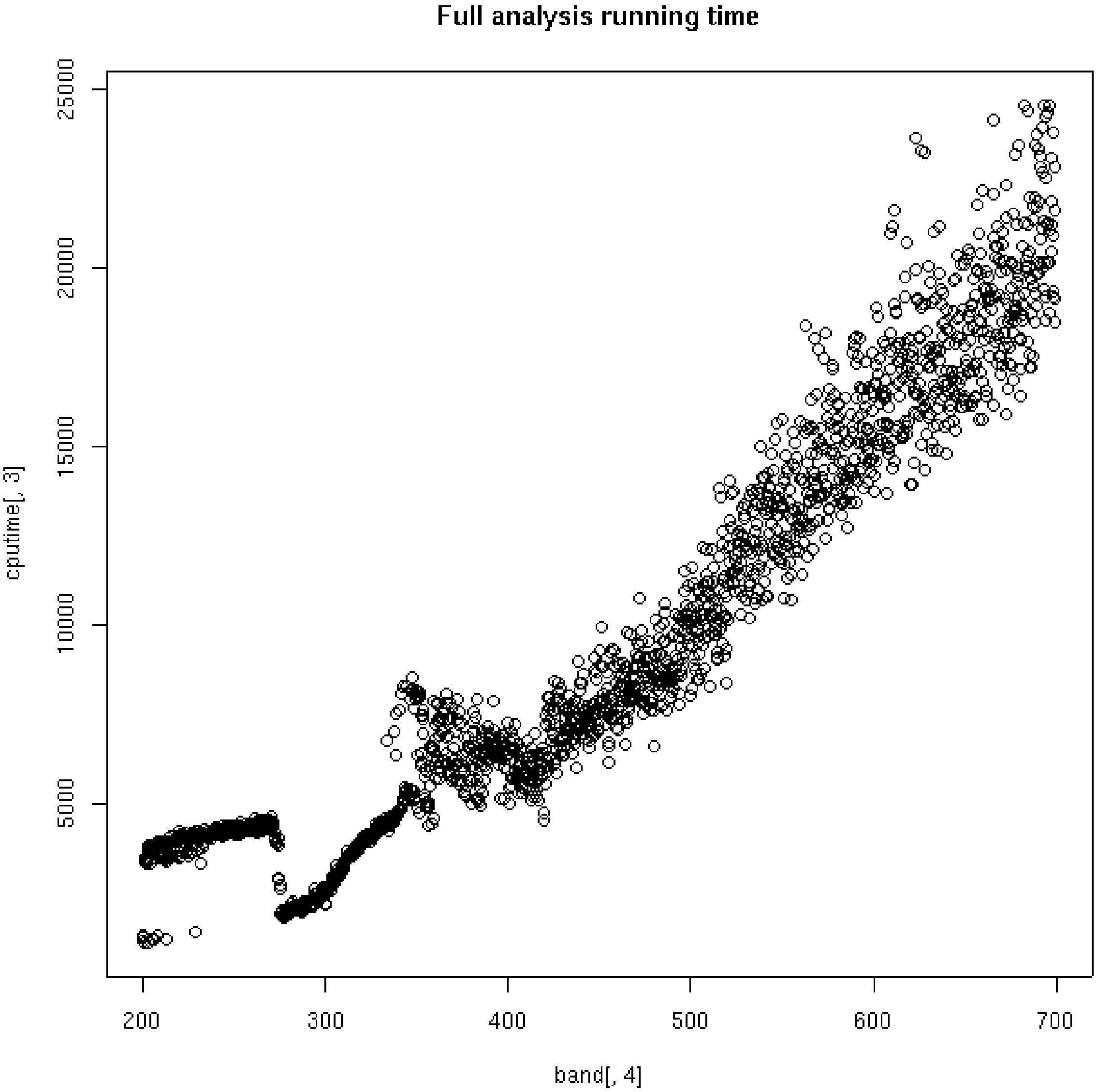

Figure 37: Full analysis running time

The total time (summed over all processes) is 222 days - which translates to less than a day of cluster compute time.

The odd behaviour for small frequencies is explained by the fact that all PowerFlux processes access data on the same RAID disk and that the processes are started from small frequencies to the largest. Apparently there is an inefficiency when many processes start to read data - even though much of it overlaps. Since the cluster has about 300 nodes the disk usage decreases once the first 300 processes started have read their data.

In fact, disk utilization problems persist throughout the entire run which can be felt as sluggish response of the disk array. Indeed, even though the jobs analysing high frequency bands take longer the increase is small compared to the total number of jobs accessing the disk (the number of processes accessing the disk decreases at most by a factor of 5).

This bottleneck might be worked around by some heuristic arbitration of disk array access times, or by moving the entire data set (110Gb for S3 H1 data) somewhere it can be accessed without delay.

Plot 39 shows running time for software injections analysis. We see much variation in actual time used, probably due to disk utilization and other jobs running on the cluster. Since the software injections restrict analysis to the portion of sky around injection point the speed of the analysis is mostly limited by disk access speeds.

14 Command line arguments

When started with `--help` option PowerFlux prints out a short summary of options:

```
powerflux 1.3.3

Purpose:
  Powerflux analysis program

Usage: powerflux [OPTIONS]...

  -h, --help                              Print help and exit
  -V, --version                           Print version and exit
  -c, --config=STRING                     configuration file (in gengetopt
                                            format) to pass parameters
```

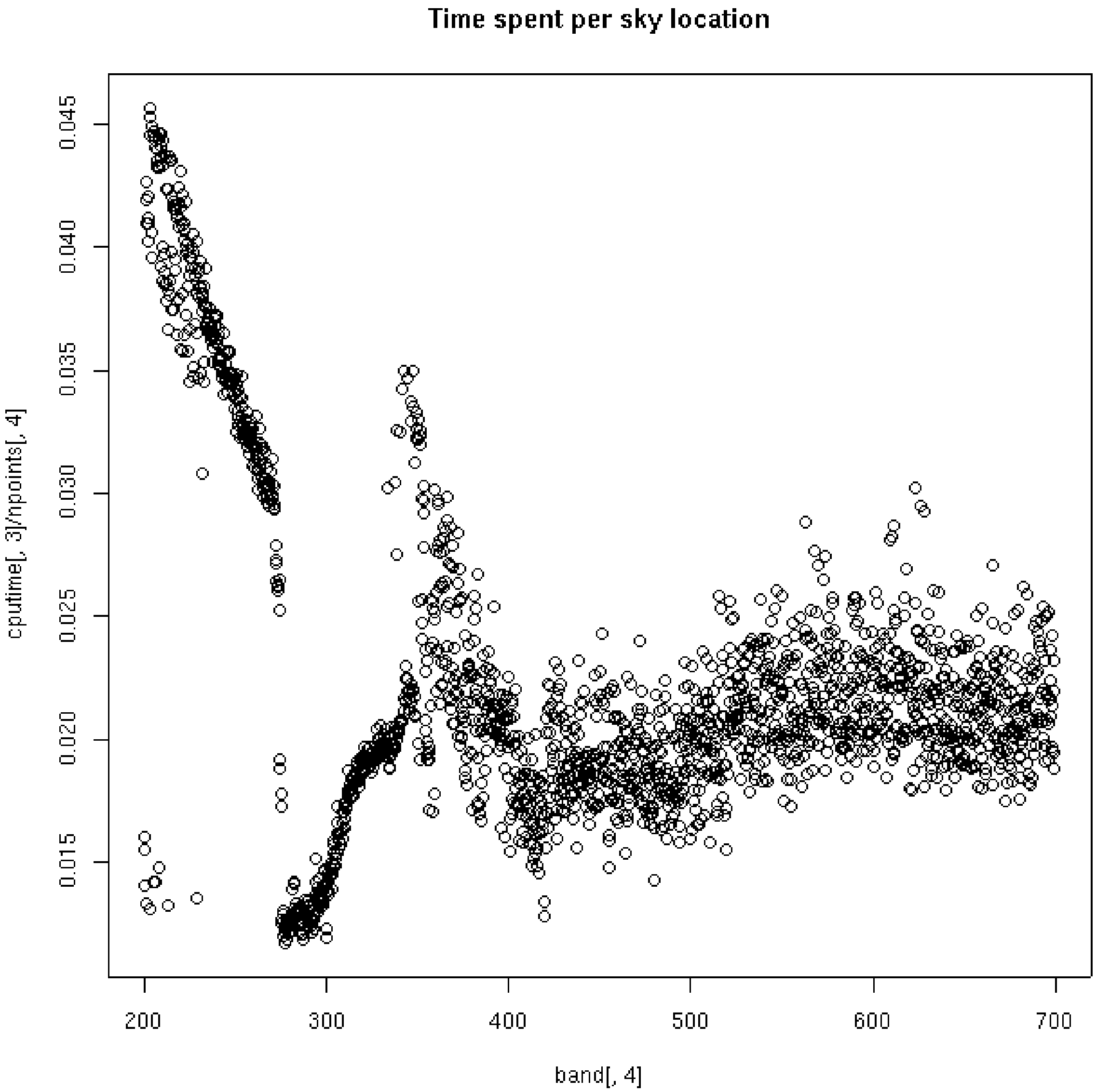

Figure 38: Time spent per sky location

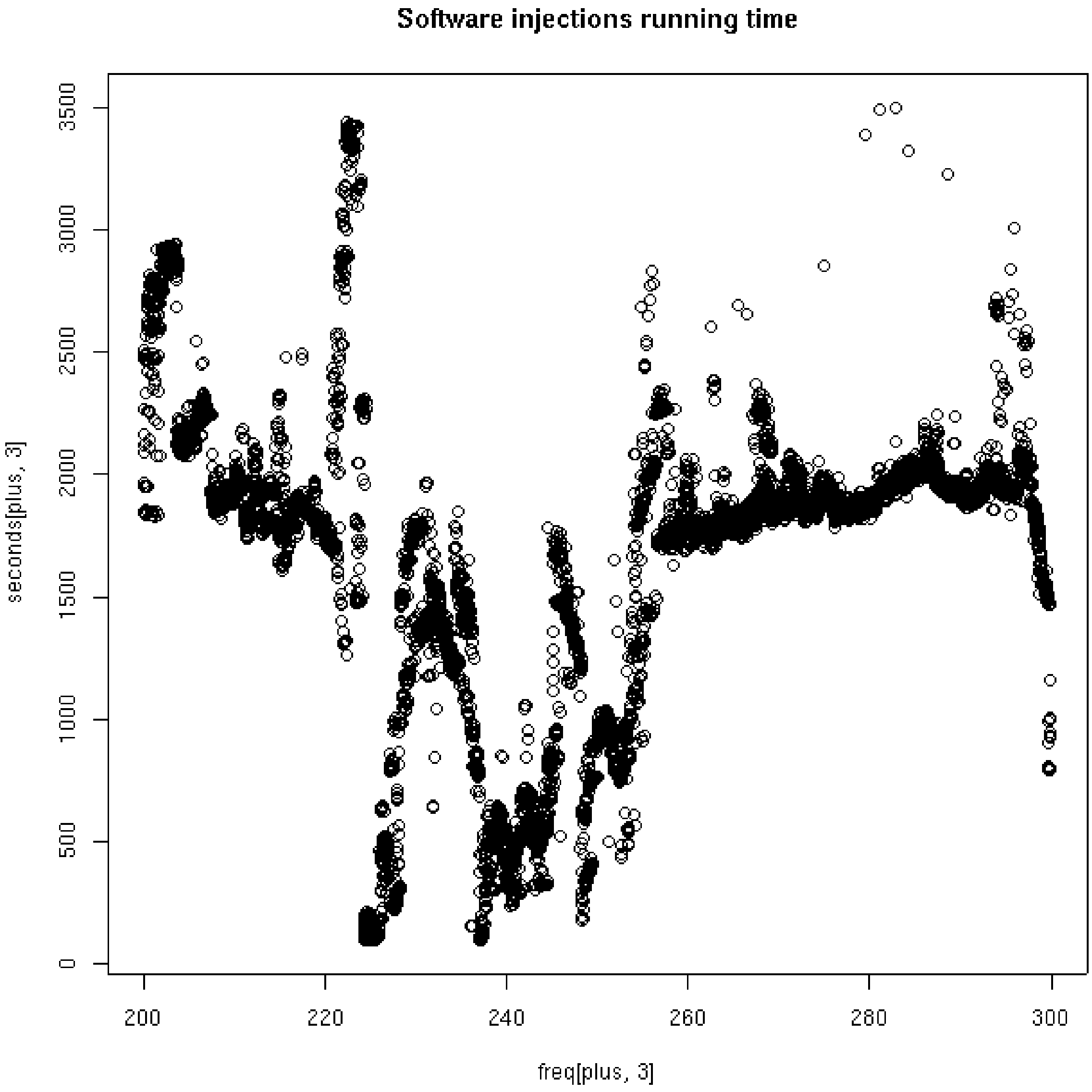

Figure 39: Software injections running time

```
    --label=STRING                   arbitrary string to be printed in the
                                       beginning of PowerFlux log file
    --sky-grid=STRING                sky grid type (arcsin,
                                       plain_rectangular, sin_theta)
                                       (default=‘sin_theta’)
    --skymap-orientation=STRING      orientation of produced skymaps:
                                       equatorial, ecliptic, band_axis
                                       (default=‘equatorial’)
    --skyband-method=STRING          method of assigning band numbers:
                                       angle, S  (default=‘S’)
    --nskybands=INT                  split sky in this many bands for
                                       logging maximum upper limits
                                       (default=‘5’)
    --large-S=DOUBLE                 value of S to consider good enough
    --band-axis=STRING               which band axis to use for splitting
                                       sky into bands (perpendicular to
                                       band axis) (possible values:
                                       equatorial, auto,
                                       explicit(float,float,float)
                                       (default=‘auto’)
    --band-axis-norm=DOUBLE          norm of band axis vector to use in S
                                       value calculation
    --fine-factor=INT                make fine grid this times finer
                                       (default=‘5’)
    --skymap-resolution=DOUBLE       specify skymap resolution explicitly
    --skymap-resolution-ratio=DOUBLE adjust default coarseness of the grid
                                       by this factor  (default=‘1.0’)
    --small-weight-ratio=DOUBLE      ratio that determines which weight is
                                       too small to include in max
                                       statistics  (default=‘0.2’)
-i, --input=STRING                   path to input files (power or SFT)
    --input-munch=STRING             how to derive SFT name from --input
                                       (highly arcane)  (default=‘powerflux135714951
    --input-format=STRING            format of input files (GEO, SFT,
                                       Power)  (default=‘GEO’)
    --segments-file=STRING           file with list of segments to process
                                       - this allows subsetting of full SFT
                                       set
    --veto-segments-file=STRING      file with list of segments *NOT* to
                                       process - this allows subsetting of
```

```
                                        full SFT set
-o, --output=STRING                   output directory
    --ephemeris-path=STRING           path to detresponse program from
                                        lalapps
    --earth-ephemeris=STRING          Earth ephemeris file, overrides
                                        ephemeris-path argument
    --sun-ephemeris=STRING            Sun ephemeris file, overrides
                                        ephemeris-path argument
-f, --first-bin=INT                   first frequency bin in the band to be
                                        analyzed
-n, --nbins=INT                       number of frequency bins to analyze
                                        (default=‘501’)
    --side-cut=INT                    number of bins to cut from each side
                                        due to corruption from doppler
                                        shifts
    --hist-bins=INT                   number of bins to use when producing
                                        histograms  (default=‘200’)
-d, --detector=STRING                 detector location (i.e. LHO or LLO),
                                        passed to detresponse
    --spindown-start-time=DOUBLE      specify spindown start time in GPS
                                        sec. Assumed to be the first SFT
                                        segment by default
    --spindown-start=DOUBLE           first spindown value to process
                                        (default=‘0.0’)
    --spindown-step=DOUBLE            step for processing multiple spindown
                                        values  (default=‘0.0’)
    --spindown-count=INT              how many separate spindown values to
                                        process  (default=‘1’)
    --orientation=DOUBLE              additional orientation phase,
                                        specifying 0.7853 will turn plus
                                        into cross  (default=‘0’)
    --nlinear-polarizations=INT       even number of linear polarizations to
                                        profile, distributed uniformly
                                        between 0 and PI/2  (default=‘4’)
    --no-demodulation=INT             do not perform demodulation stage,
                                        analyze background only  (default=
                                        ‘0’)
    --no-decomposition=INT            do not perform noise decomposition
                                        stage, output simple statistics only
                                         (default=‘0’)
```

```
--no-am-response=INT              force AM_response() function to return
                                    1.0 irrespective of the arguments
                                    (default=‘0’)
--subtract-background=INT         subtract rank 1 matrix in order to
                                    flatten noise spectrum  (default=
                                    ‘0’)
--three-bins=INT                  average 3 neighbouring bins to broaden
                                    Doppler curves  (default=‘0’)
--do-cutoff=INT                   neglect contribution from SFT with
                                    high effective noise level  (default=
                                    ‘1’)
--filter-lines=INT                perform detection of lines in
                                    background noise and veto
                                    corresponding frequency bins
                                    (default=‘1’)
--ks-test=INT                     perform Kolmogorov-Smirnov test for
                                    normality of averaged powers
                                    (default=‘1’)
--compute-betas=INT               compute beta coefficients as described
                                    in PowerFlux polarizations document
                                    (default=‘0’)
--upper-limit-comp=STRING         upper limit compensation factor - used
                                    to account for windowing in SFTs
                                    (possible values: Hann, flat,
                                    arbitrary number)  (default=‘Hann’)
--lower-limit-comp=STRING         lower limit compensation factor - used
                                    to account for windowing in SFTs
                                    (possible values: Hann, flat,
                                    arbitrary number)  (default=‘Hann’)
--write-dat=STRING                regular expression describing which
                                    *.dat files to write  (default=‘.*’)
--write-png=STRING                regular expression describing which
                                    *.png files to write  (default=‘.*’)
--dump-points=INT                 output averaged power bins for each
                                    point in the sky  (default=‘0’)
--focus-ra=DOUBLE                 focus computation on a circular area
                                    with center at this RA
--focus-dec=DOUBLE                focus computation on a circular area
                                    with center at this DEC
--focus-radius=DOUBLE             focus computation on a circular area
```

```
                                    with this radius
      --only-large-cos=DOUBLE     restrict computation to points on the
                                    sky with cos of angle to band axis
                                    larger than a given number

 Group: injection
      --fake-linear               Inject linearly polarized fake signal
      --fake-circular             Inject circularly polarized fake
                                    signal
      --fake-ra=DOUBLE            RA of fake signal to inject  (default=
                                    ‘3.14’)
      --fake-dec=DOUBLE           DEC of fake signal to inject  (default=
                                    ‘0.0’)
      --fake-orientation=DOUBLE   orientation of fake signal to inject
                                    (default=‘0.0’)
      --fake-spindown=DOUBLE      spindown of fake signal to inject
                                    (default=‘0.0’)
      --fake-strain=DOUBLE        amplitude of fake signal to inject
                                    (default=‘1e-23’)
      --fake-freq=DOUBLE          frequency of fake signal to inject
```

14.1 `config`

Specify the first part of SFT file name. The complete name is formed by appending an SFT number to the end.

Instead of specifying arguments on the command line it is possible to create a file with each line containing `option value` pair. The ”–” prefix in front of the option name must to be omitted.

In the following PowerFlux options would be referred by their name as used in the configuration file, for command line use prepend `"--"`.

14.2 Input/Output options

14.2.1 `input`

Specify the first part of SFT file name. The complete name is formed by appending an SFT number to the end.

14.2.2 `input-munch`

Specify how complete file name is formed. This is effectively the format argument to sprintf, which is also passed the string from `input` option and an integer SFT number. The default is `"%s%ld"`. To use 5 digit numbers (like `"00035"`) specify `"%s%05ld"`.

14.2.3 `input-format`

Specify the format of input SFT files.

- `Power` refers to ASCII header binary body power-only files produces by `make_sft_op`
- `SFT` refers to ASCII header binary body SFT files produced by `make_sft_op`
- `GEO` refers to binary header binary body GEO-style SFT files in common use in LSC.

14.2.4 `detector`

Specify detector location for the input files in use. This is either `LHO` or `LLO`

14.2.5 `segments-file`

Allows to restrict processing to only SFT files that are inside a list of segments described in the ASCII file specified with this option.

Each line specifies a single segment described the starting GPS time followed by ending GPS time. The rest of line is discarded (this makes possible to use standard segment list files).

14.2.6 `veto-segments-file`

Same as `segments-file` except this option specifies the list of times **not** to process. Useful for vetoing parts of data.

14.2.7 `ephemeris-path`

Path to files with ephemeris data (they can be found, for example, in `lalapps/src/detresponse/` directory).

14.2.8 `earth-ephemeris`

Specify Earth ephemeris file explicitly. This overrides `ephemeris-path` option.

14.2.9 `sun-ephemeris`

Specify Sun ephemeris file explicitly. This overrides `ephemeris-path` option.

14.2.10 `output`

Specify directory to put output of PowerFlux into. Everything that PowerFlux writes will be located in this directory.

14.3 Analysis parameters

14.3.1 `first-bin`

Specify first bin of 501 bin stretch to analyze. This is an integer in units of 1/1800 Hz. It specifies the *source* (i.e. decoded) frequency, not frequency as received by the detector.

14.3.2 `nbins`

Number of frequency bins to analyze. The default is 501. At the moment this value should not be changed - there are some hard-coded constants that rely on this number. In particular, the Feldman-Cousins method relies on constants produced by Monte-Carlo simulation on the assumption that `nbins`= 501.

14.3.3 `side-cut`

Due to the need to apply Doppler shift the actual number of bins read from SFT file is larger than `nbins` and varies with frequency and spindown. Normally PowerFlux will compute the number of extra bins to read automatically. This option allows an explicit override. Specifying `--side-cut=100` will cause PowerFlux to read all bins from `first-bin`−100 to `first-bin`+`nbins`+100.

14.3.4 `spindown-start`

Specify initial spindown value to process. It is a floating-point value in units of Hz/sec. Negative values correspond to frequency decreasing with time.

14.3.5 `spindown-count`

Specify the number of spindown values to process starting with value specified by `spindown-start` option.

14.3.6 `spindown-step`

Specify the increment between spindown values to process. Can be positive or negative.

14.3.7 `npolarizations`

Instruct PowerFlux to produce limits for specified number of linear polarizations. Plus and cross are always included.

14.3.8 `orientation`

Additional phase to add to polarizations. Specifying $\pi/4$ will produce plus polarization plots that coincide with cross plots produced with `orientation`= 0.

14.4 Analysis options

14.4.1 `no-demodulation`

Do not perform demodulation, stop after analyzing background. Since Feldman-Cousins is not performed `nbins` can be specified to an arbitrary number, although values exceeding 25 Hz require lots of computer memory (in excess of 2 Gb).

It is convenient to specify `--side-cut=0` to provide greater control over starting frequency.

14.4.2 `no-decomposition`

Only read in SFT files and output simple statistics. This is even faster than `no-demodulation` option. Same suggestions apply.

14.4.3 `no-am-response`

Assume that amplitude response is always 1.0 irrespective of time or sky position.

14.4.4 `skymap-resolution`

PowerFlux computes optimal resolution of skymaps automatically (It depends mostly on magnitude of Doppler shifts). This options allows to specify resolution explicitly skymaps. This option is very handy for comparing PowerFlux skymap output for different frequency bands.

14.4.5 `skymap-resolution-ratio`

PowerFlux computes optimal resolution of skymaps automatically (It depends mostly on magnitude of Doppler shifts). This options allows to force skymaps to have a finer or coarser resolution by a given factor.

14.4.6 `small-weight-ratio`

PowerFlux discards data in weighted sum that is assigned too small a weight. This option allows to specify it.

For a perfect instrument setting this ratio to 0 will produce the best result as the weighting scheme used will make good use even of data with very small weights.

In practice, discarding SFTs saves CPU cycles so it makes sense to skip those which provide marginal improvement.

Furthermore, the SFTs with small weight can often have radically different noise spectrum. The default value is prudent 0.2.

14.4.7 `three-bins`

Specifying `three-bins=1` causes PowerFlux to average every neighbouring three bins in its analysis. Because of this Doppler tracks are widened and a coarser skymaps may be used while still retaining full sky coverage.

The drawback is a factor of $\sqrt{2}$ loss in sensitivity.

14.4.8 `do-cutoff`

Enabled by default. Setting it to 0 will turn off Cutoff computation. This is similar to specifying `--small-weight-ratio=0.0`

14.4.9 `filter-lines`

Perform automatic detection of lines in background noise and veto corresponding frequency bins. Up to 5 frequency bins can be vetoed.

14.4.10 `subtract-background`

Subtract rank 1 matrix formed from `TMedians` and `FMedians` in order to improve noise performance for sky positions with large variation in Doppler shifts. Turning this option on will clobber signals from sky positions with small variation in Doppler shifts, therefore one should combine it with `only-large-cos=0.3`.

14.5 Data reporting options

14.5.1 `skymap-orientation`

The skymaps produced by PowerFlux can have different orientations to please the user. Possible choices are equatorial, ecliptic or "band-axis" - with band axis vector pointing to the North pole.

14.5.2 `nskybands`

Split sky in a given number of bands and report analysis results for each band individually.

14.5.3 `skyband-method`

Specify method used to partition sky into regions.

Possible values are `angle` and `S`. The latter method is useful on short timebases, while the former can used to partition the sky into bands along declination.

14.5.4 `band-axis`

By default PowerFlux computes optimal band axis automatically (this has to do with average detector acceleration during analyzed data set). However, it may be useful to specify it explicitly - for example for comparison of results between different interferometers.

Possible values are `equatorial`, `auto` and `explicit(%f,%f,%f)`.

14.5.5 `band-axis-norm`

Specify the norm of band axis vector explicitly. This is useful for comparison of results between different IFOs.

14.5.6 `large-S`

Specify values of S function considered to be good enough. All sky points with S value larger or equal to this value will be assigned to band 0.

14.5.7 `only-large-cos`

Restrict computation to only those areas of sky which projection to band axis has absolute value larger than a value specified to this option. If you want to do this (due to presence of line artifacts, for example) the recommended value is 0.3.

This cleans up the results reported for entire skymap and can significantly reduce computation time requirements.

14.5.8 `ks-test`

Perform and output results of Kolmogorov-Smirnov test for compliance of averaged weighted power with gaussian distribution with parameters employed later to establish Feldman-Cousins limits. This increases the computation time, but is a highly recommended cross check for analysis. High values of KS statistic indicate bands with pathological noise floor behaviour.

14.5.9 `compute-betas`

Compute and log values of beta constants (described in "Polarizations" document). These constant are useful for establishing limits on polarization different from one that was sampled. This is somewhat cpu-intensive so this option is turned off by default.

14.5.10 `upper-limit-comp`

A factor to multiply upper limits reported in strain units. One can specify a floating-point number or "Hann" for Hann windowed SFTs. This factor is used to account for the fact that non bin-centered (in frequency) signals would have smaller amplitude than bin-centered ones. For Hann windowed SFTs, using 1-bin mode the factor is 1/0.85.

14.5.11 `lower-limit-comp`

A factor to multiply lower limits reported in strain units. One can specify a floating-point number or "Hann" for Hann windowed SFTs. This factor is used to account for the fact that non bin-centered (in frequency) signals would have smaller amplitude than bin-centered ones. For Hann windowed SFTs, using 1-bin mode the factor is 1.

14.5.12 write-dat

By default PowerFlux writes a binary file with data for each plot it makes. You can use this option to specify a regular expression to filter what will actually be written.

This can significantly reduce storage requirements for PowerFlux output, as well as speed up computation.

14.5.13 write-png

By default PowerFlux creates a number of plots. You can use this option to specify a regular expression to filter what will actually be written.

This can significantly reduce storage requirements for PowerFlux output, as well as speed up computation.

14.6 Software injections

The following options provide interface to software injections done by PowerFlux itself (as opposed to using external programs). The injections are power-only, modeled with assumption of random phase of incoming signal to a particular frequency bin.

14.6.1 `fake-linear`

Perform injection of linearly polarized signal.

14.6.2 `fake-circular`

Perform injection of circularly polarized signal.

14.6.3 `fake-ra`

Specify right ascension of injected signal source in radians (values from 0 to 2π are acceptable).

14.6.4 `fake-dec`

Specify declination of injected signal source in radians (values from $-\pi/2$ to $\pi/2$ are acceptable).

14.6.5 `fake-orientation`

Specify polarization of injected signal (assumed to be linearly polarized). Valid values are between 0 and $\pi/4$.

14.6.6 `fake-spindown`

Specify spindown of injected signal in units of Hz/sec.

14.6.7 `fake-strain`

Specify strain of injected signal.

14.6.8 `fake-freq`

Specify frequency of injected signal.

15 Changes

01/20/05 First PowerFlux review

01/20/05 Add missing square root to the formula for determining `NCutOff`.

01/20/05 Add section describing sky bands and band axis vector.

01/25/05 Add section describing power-only software injection.

01/30/05 Add section describing computing resources utilization.

02/10/05 Add subsection with example plots of PowerFlux output.

02/10/05 Second PowerFlux review.

02/10/05 Add two plots describing daily variation in data quality for S3 H1.

03/13/05 Update –help output to latest version.

03/13/05 Add description of several more command line options.

03/13/05 Reword description about power-only software injections - add a separate item describing computation of incoming pulsar power.

04/21/05 Update PowerFlux options description to the most recent version.

05/05/05 Update PowerFlux options description to the most recent version. Add description of new options: `upper-limit-comp`, `lower-limit-comp` and `compute-betas`.

05/05/05 Add section describing format of input data.

05/05/05 Correct formula for normalization of GEO format.

05/06/05 Fix cut'n'paste error in description of `lower-limit-comp` option.

05/18/05 Add section describing robust estimation of Gaussian distribution parameters.

07/04/05 Update description of automatic line veto algorithm to correspond with latest PowerFlux code.

09/04/05 Update PowerFlux option description to correspond with the latest code.

09/04/05 Include description of S function used to assign bands.

References

[1] Gary J. Feldman, Robert D.Cousins, A Unified Approach to the Classical Statistical Analysis of Small Signals, arXiv:physics/9711021