LIGO T080061-00-Z

Adding Coherent Summing to PowerFlux

Junyi Zhang, Courtney Jarman, Keith Riles

(University of Michigan)

Introduction

In carrying out all-sky pulsar searches with PowerFlux, the strategy so far has been to search separately in H1 and L1 data for outliers and then to require coincidence in sky location, frequency and spindown for outliers to identify candidates for subsequent follow-up using multi-interferometer power summing to improve signal-to-noise ratio (SNR)[1]. When H1 and L1 sensitivities are equal, one thereby expects (ideally) to obtain an improvement in SNR by approximately a factor of √2, allowing false candidates to be rejected, in principle. In practice, with typical single-IFO thresholds at only 6.0-6.5, statistical fluctuations on both the single-IFO and the combined-IFO SNR's make it unsafe to require the nominal √2 improvement.

In this note we explore the potential gain in combined SNR from a coherent sum of SFT coefficients from pairs of coincident H1 and L1 30-minute SFT's. We first demonstrate analytically under highly idealized circumstances why one expects an additional √2 improvement, that is, a factor of 2 improvement over single-IFO SNR and then carry out more realistic simulations to demonstrate that the nominal improvement is nearly achievable after taking into the noise and antenna-pattern weighting algorithm used by PowerFlux.

This potential gain in combined SNR from the coherent summing method holds the promise of more clearly distinguishing coincident detector artifacts from true gravitational-wave signals, although it should be noted that the SFT coincidence requirement implies additional non-coincident H1 and L1 SFT's would continue to be added separately.

Basic Principles

To understand why coherent summing achieves an additional $\sqrt{2}$ factor of improvement over incoherent summing (equal-sensitivity IFO's), consider the following concrete case. Suppose during one 1800-second interval, the relevant Hanford SFT coefficient is composed of two parts, the signal and the noise. Let the signal be *a*, and the noise be x+yi, here x and y are N(0, $\sigma = b$) normal random variables. In the signal bin, the coefficient is a sum of signal and noise, while distant (noise) bins are taken to be distributed according to N(0, $\sigma = b$).

Single-SFT comparison:

Denote the Hanford site by 1 and Livingston by 2. For the Hanford spectrum, the signal bin coefficient is $a + x_1 + y_1 i$, the noisy bin coefficients are $\tilde{x}_1 + \tilde{y}_1 i$, and for Livingston, the signal bin coefficient is

$a + x_2 + y_2 i$ (we assume the phases are already matched, hence take a as a real number for simplicity), and noise bins coefficients are $\tilde{x}_2 + \tilde{y}_2 i$.

Now for the incoherent summing, the combined power in the signal bin is:

$$|a + x_1 + y_1 i|^2 + |a + x_2 + y_2 i|^2 = 2a^2 + x_1{}^2 + 2ax_1 + y_1{}^2 + x_2{}^2 + 2ax_2 + y_2{}^2$$

The average noise level in the noisy bins is:

$$< |\tilde{x}_1 + \tilde{y}_1 i|^2 + |\tilde{x}_2 + \tilde{y}_2 i|^2 > = < \tilde{x}_1^2 + \tilde{y}_1^2 + \tilde{x}_2^2 + \tilde{y}_2^2 > = 4b^2$$

The noise variance is:

$$Var(|\tilde{x}_1 + \tilde{y}_1 i|^2 + |\tilde{x}_2 + \tilde{y}_2 i|^2) = Var(\tilde{x}_1^2 + \tilde{y}_1^2 + \tilde{x}_2^2 + \tilde{y}_2^2) = 8b^4$$

Here we used the fact that $\tilde{x}_1, \tilde{y}_1, \tilde{x}_2, \tilde{y}_2$ are all independent random variables.

On the other hand, using coherent summing, we get:
The combined power in the signal bin:

$$\begin{aligned} &|a + x_1 + y_1 i + a + x_2 + y_2 i|^2 = (2a + x_1 + x_2)^2 + (y_1 + y_2)^2 \\ &= 4a^2 + x_1^2 + x_2^2 + y_1^2 + y_2^2 + 2y_1 y_1 + 2x_1 x_2 + 4ax_1 + 4ax_2 \end{aligned}$$

The average noise level in the noisy bins is:

$$< |\tilde{x}_1 + \tilde{y}_1 i + \tilde{x}_2 + \tilde{y}_2 i|^2 > = < \tilde{x}_1^2 + \tilde{y}_1^2 + \tilde{x}_2^2 + \tilde{y}_2^2 + 2\tilde{x}_1\tilde{x}_2 + 2\tilde{y}_1\tilde{y}_2 > = 4b^2$$

The noise variance is:

$$\begin{aligned} &Var(|\tilde{x}_1 + \tilde{y}_1 i + \tilde{x}_2 + \tilde{y}_2 i|^2) = Var((\tilde{x}_1 + \tilde{x}_2)^2 + (\tilde{y}_1 + \tilde{y}_2)^2) \\ &= Var((\sqrt{2} z_1 b)^2 + (\sqrt{2} z_2 b)^2) = 16b^4 \end{aligned}$$

Here z_1, z_2 are standard normal random variables.

Multi-SFT comparison:

The above relations hold for a single 1800-second period. We must aggregate N such periods to get the final power spectrum. For the incoherent summing case, after summing over N periods, we obtain the power in the signal bin:

$2Na^2 + N < x_1{}^2 + y_1{}^2 + x_2{}^2 + y_2{}^2 > = 2Na^2 + 4Nb^2$ (here the average is taken over the N segments, not over different bins in a single period, unlike in the above). The average noise level is $4Nb^2$, and the noise variance is $8Nb^4$.

The SNR is calculated as (signal bin power − noise power level) / (noise power standard deviation), i.e.

$$\text{SNR} = \frac{2Na^2 + 4Nb^2 - 4Nb^2}{\sqrt{8Nb^4}} = \frac{\sqrt{N} a^2}{\sqrt{2} b^2}$$

For coherent summing, we obtain the following power in the signal bin: $4Na^2 + N < x_1^2 + x_2^2 + y_1^2 + y_2^2 >= 4Na^2 + 4Nb^2$ (here again the average is taken over the N segments). The average noise level is $4Nb^2$, and the noise variance is $16Nb^4$

The SNR is $\frac{4Na^2 + 4Nb^2 - 4Nb^2}{\sqrt{16Nb^4}} = \frac{\sqrt{N}a^2}{b^2}$.

So the SNR for coherent summing is improved by a factor of $\sqrt{2}$, ie., 41%, with respect to incoherent summing. This is of course the ideal situation: perfect phase match, fixed signal frequency, identical signal strength for Livingston and Hanford, and simple algebraic average. In the next section we describe simulations where the ideal assumptions are relaxed.

Simulation Studies

In coherently summing strain SFT coefficients from Hanford and Livingston, one must take into account the time delay expected between the sites for a true gravitational wave, a time delay that is dependent upon the sky location of the source. The "SNR ridges" in the sky that arise from Doppler demodulation do not, in general, coincide with the ridges that arise from time delay corrections, providing (typically) improved localization of a source in addition to improved SNR. The time delay is accounted for via a relative phase correction between the Hanford and Livingston SFT coefficients prior to summing.

Several technical issues arise, however, in making the phase correction. First, the duration of an SFT is long enough that for higher frequencies, subject to larger Doppler modulations from the Earth's rotation, the slight mismatch in Hanford & Livingston detector-frame frequencies leads to a growing phase mismatch between the detectors. For example, for a 1000-Hz signal, the difference in Hanford and Livingston detector frame frequencies can be large as 0.4 mHz, leading to a change in phase mismatch as large as $3\pi/2$ in the worst case. The SFT windowing choice creates additional phase corrections that must be taken into account. As is done for PowerFlux, we choose to use Hann Windowing in the simulations and to apply phase corrections accordingly. As discussed below, we choose to use a "matched filter" approach to obtain coefficients based on the existing PowerFlux matched filter option used in single-IFO searches (as opposed to the default "one-bin" option).

To carry out these studies, a program called *simcoh* has been written to simulate production and analysis of one month of LIGO data (1440 30-minute SFT's). As is done by default in PowerFlux analysis, SFT's are Hann-windowed with 50% overlap, yielding 2879 SFT's in total. In the signal simulation *makefakedata* is used to generate the SFT's. Since makefakedata is hardwired to generate Tukey-windowed SFT's with a transition fraction of 0.05, we use a modified version (via the gdb debugger) to generate Hann-windowed SFT's instead. (An alternative approach would be to apply a digital filter to the Tukey-windowed SFT's.)

In detail, we run makefakedata four times under the gdb environment, twice to generate Hanford SFT,'s twice to generate Livingston SFT's. The commands are:

```
gdb makefakedata_v2 --command=gdbinit1
gdb makefakedata_v2 --command=gdbinit2
gdb makefakedata_v2 --command=gdbinit3
gdb makefakedata_v2 --command=gdbinit4
```

It takes about 6 minutes to generate all the SFTs needed (on the LIGO GC computer Marble SUN v40z-4cpu2.4 GHz each, 8 GB total memory) Then *simcoh* reads in the SFT's and performs an all-sky search for a GW signal. By default the sky locations are divided into a 100 × 100 grid, and a frequency search is done only for the 3 frequency bins nearest the true frequency.

Let's examine more closely the relation between inter-site time delay and phase mismatch due to differing detector-frame velocities. Suppose $f_1(t)$ and $f_2(t)$ are two exact sine waves with the same frequency f, and $f_2(t)$ is Δt earlier than $f_1(t)$, i.e. $f_2(t) = f_1(t+\Delta t)$. Then the Fourier coefficients would be $F_2 = F_1 \exp(i2\pi f \Delta t)$. On the other hand, if $f_1(t)$ and $f_2(t)$ are two exact sine waves with the same starting phase, but with slightly different frequency $\Delta f = f_2 - f_1$, then $F_2 = F_1 \exp(i\pi\Delta f T)$, where T is the length of the time series. This relation holds only if Δf is small enough, typically $\Delta f < 2/T$. In our case this condition is well satisfied even at a high frequency like 1000Hz (the maximum frequency difference between Hanford and Livingston for 1000Hz signal is about 1 bin in frequency domain, while the maximum frequency drift for a single site during 1800 seconds is about 0.1 bin). Now if we combine these two phase mismatches, i.e. if $f_1(t)$ and $f_2(t)$ are exact sine waves, with Δt time delay at the starting time, as well as a frequency difference Δf, their Fourier coefficients can be related by:

$$F_2 = F_1 \exp(i2\pi f \Delta t)\exp(i\pi\Delta f T) = F_1 \exp\left(i\frac{2\pi f \Delta t + (2\pi f \Delta t + 2\pi\Delta f T)}{2}\right) = F_1 \exp\left(i\frac{p_b + p_e}{2}\right)$$

here $p_b = 2\pi f \Delta t$ is the phase difference at the starting time, $p_e = 2\pi f \Delta t + 2\pi\Delta f T$ is the phase difference at the end time. We can approximate the $\frac{p_b + p_e}{2}$ by the phase difference at the mid point of the time segment p_m, or the average phase difference during the whole time segment p_a. Hence we get:

$$F_2 \approx F_1 \exp(ip_m) \approx F_1 \exp(ip_a)$$

Using this relation, we can simply calculate the time delay at the mid point of a time segment, Δt_m, and calculate $p_m = 2\pi f \Delta t_m$. This proves to be very good and efficient phase matching method, and *simcoh* use a LAL routine LALTimeDelay() to compute Δt_m .

After the above phase correction, the Hanford and Livingston signals may still have a 180 degree phase difference. This is because Hanford and Livingston are aligned in such a way that most of the time, the antenna pattern introduces a relative minus sign between Hanford and Livingston signals. However, there are occasional SFT's for which the amplitude responses from a particular source have the same sign. Hence *simcoh* samples the amplitude response function during the 1800 sec period to see if a 180 degree phase correction is needed. After taking care of this correction, the phase match is typically very good. For 1000Hz, the average phase mismatch is only about 0.1 radians (depends on the source location and simulation period).

As mentioned above, we use a matched filter in signal reconstruction, based on the seven bins nearest to the signal's expected frequency[2], exploiting the small spectral leakage from the Hann window. In the simulation, the matched filter is applied to the noise bins as well as to the signal bin, to be conservative, although this application is the most time-consuming step in the simulation.

To obtain clean SNR ratios that are robust against noise fluctuations, we carry out simulations with quite strong signal injections. In calculating the SNR a weighted average is used in accordance with the standard PowerFlux algorithm[2]. In detail, the formulae are the following:

For a particular assumed sky location and signal frequency we define the SNR in the usual way:

$$SNR = \frac{total\_power - mean\_of\_noise}{std\_of\_noise}$$

where

$$total\_power = \frac{\sum \frac{1}{2}(w * \max bin + w1 * \max bin1)}{\sum \frac{1}{2}(w + w1)}$$

$$mean\_of\_noise = \frac{\sum \frac{1}{2}(w * noise\_bar + w1 * noise\_bar1)}{\sum \frac{1}{2}(w + w1)}$$

$$std\_of\_noise = \sqrt{\frac{1}{\sum \frac{1}{2}(w + w1)}}$$

Where w, and w1 are the weights for different time segments (w for the non-overlapping segments and w1 for overlapping segments. Overlapping segments are 900 seconds behind the non-overlapping segments in time; otherwise they are quite the same). The summation is over 1440 segments (but for the first and last segments there are minor modifications).

$\max bin$ is the power in the bin where the maximum power for non-overlapping spectrum is supposed to be; while $\max bin1$ is the power in the bin where the maximum power for overlapping spectrum is supposed to be.

$$noise\_bar = mean(noise\_powerspectrum)$$

$$noise\_bar1 = mean(noise\_powerspectrum1)$$

(here the suffix “1” always mean “non-overlapping” segments)

$$w = \left(\frac{1}{std(noise\_powerspectrum)}\right)^2$$

$$w1 = \left(\frac{1}{std(noise\_powerspectrum1)}\right)^2$$

Note all the power spectrum are computed by adding Hanford power and Livingston power together, coherently or incoherently, depending on the option in simcoh.

For more information about simcoh usage, please refer to Appendix A.
The following is an illustration of the simcoh results. The makefakedata parameters used are:

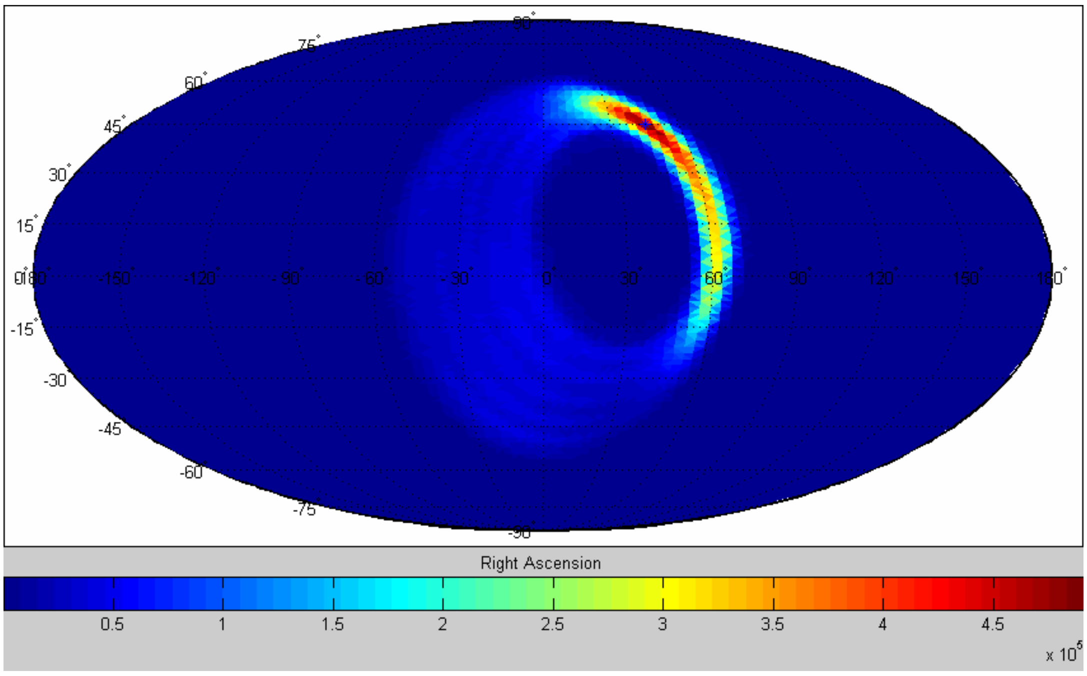
1800	Length of Time series in seconds
1440	total number of time series garneted
0.1	sigma (std of noise. When=0 only signal present)
0.24	Aplus
0.0	Across
0.8	psi
0.9	initial phase
100.4	signal frequency
0.7854	source Dec in radians (on the vertex of the grid)
0.7854	source RA in radians (on the vertex of the grid)
0	frequency spin down

And the searching grid is 100*100 over (-pi,pi),(-pi/2,pi/2). We did not use matched filter during this simulation. We typically chose very strong signals in order to evaluate accurately the improvement in SNR

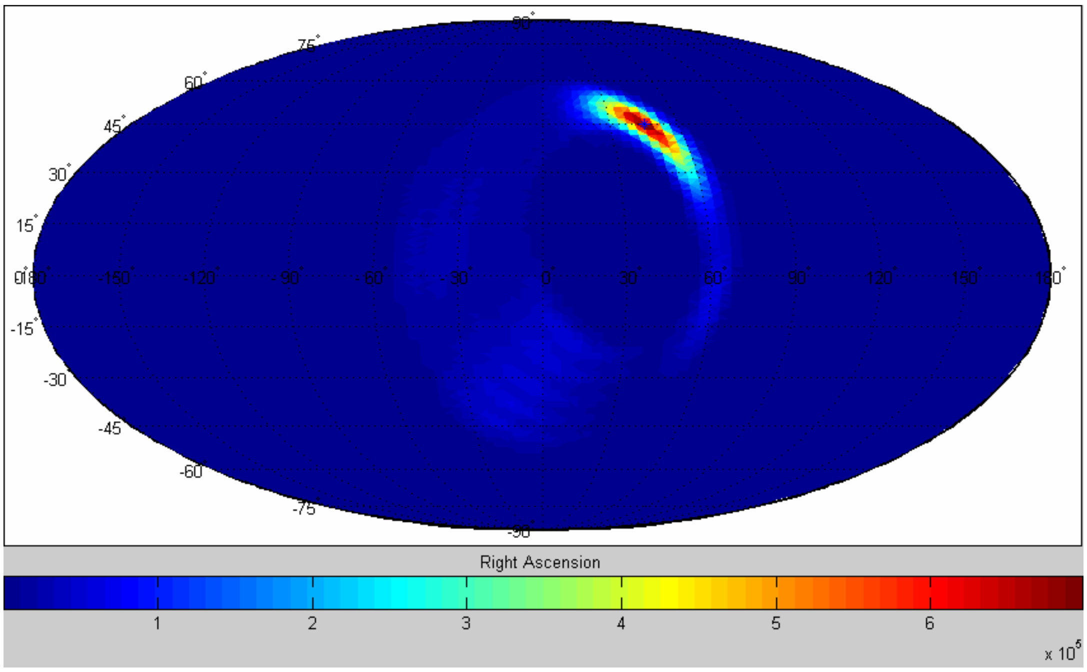
The first plot below is a SNR sky map computed without coherent summing. The second plot is computed using coherent summing. Both plots yield the true frequency of the source. The number in the color bar is the value of SNR. The true source location is indicated by “+”, the searched source location is indicated by “x”. As we can see, both coherent summing and coherent summing find the right source location. However, due to coherent summing, the SNR is increased by 733907/510841=1.44~sqrt(2). Also it is obvious that the bright area in the second plot is less than bright area in the first plot, confirming that coherent summing gives better localization than does incoherent summing

Additional simulation results are shown in the appendix B. We tested the program using signal sources at various special locations in the sky and found consistent results. Coherent summing yielded SNR improvements of about √2 w.r.t. incoherent summing.

Incoherent summing, with highest SNR=510841



Coherent summing, with highest SNR=733907



References:

1. B. Abbott et al, “All-Sky LIGO Search for Periodic Gravitational Waves in the Early S5 Data”, in preparation.
2. V. Dergachev, “Description of PowerFlux Algorithms and Implementation", LIGO technical document LIGO-T050186 (2005)

Appendix A

Technical Details about using simcoh.c

a). compiling simcoh

simcoh is a single-file program. Everything is contained in simcoh.c. Since simcoh.c used a lot of LAL routines, to compile simcoh.c, one must install LAL first and then specify the header files directory and corresponding library when compiling. Compilation is relatively easy. The following is an example:

```
gcc simcoh.c -o simcoh    -I/home/rarmen/opt/lscsoft/lal/include/(change to your own directory)
    -I/home/rarmen/opt/lscsoft/non-lsc/include/(change to your own directory) -lm -llal -llalsupport
```

b). before you can run simcoh, you must use makefakedata_v2 to generate SFT. There are about 20 parameters to be specified in makefakedata_v2, but the most important are Tsft_in_sec(the length of SFT, for the sake of simcoh, this must be set to 1800) , nTsft(the total number of SFT to be generated), first_SFT_frequency_in_Hz (the beginning of the frequency band we are interested in), SFT_freq_band_in_Hz(the width of the frequency band we are interested in), sigma(std of noise.When=0 only signal present),Aplus, Across(the signal amplitudes for two polarizations, note that Aplus/Across and sigma are not proportional, i.e. setting sigma=Aplus=1, Across=0 does not mean the signal and noise have equal amplitude), psi(polarization angle) ,f0(signal frequency), atitude_in_radians, longitude_in_radians(the sky location of the source). Also you have to specify the directory of sun and earth ephemeris files, directory for the resulting SFT files and their file names, and the GPS time that the signal is started in the simulation.

By the reason stated above, we should use a Hann window in SFT. As a result, we use gdb to alter makefakedata_v2 during runtime. You have to use commands like:

```
gdb makefakedata_v2 --command=gdbinit1
```

here *gdbinit1* is a script file for gdb, it should look something like:

```
    break 833
    commands
    silent
    set nbinw=18000
    cont
    end
    run -w -i In.data-test -I LHO -E /home/rarmen/opt/lscsoft/lal/share/lal -S 693326229
-n ./fdata/FourierH -G 693326229
    qui
    t
```

The file *In.data-test* stores the part of input parameters for makefakedata, it looks something like:

```
1800      %Tsft_in_sec
1440      %nTsft
1000.0    %first_SFT_frequency_in_Hz
10.0      %SFT_freq_band_in_Hz
0.1       %sigma_(std_of_noise.When=0_only_signal_present)
0.08      %Aplus
0.08      %Across
```

```
0.8       %psi
0.9       %phi0
1000.4    %f0
0.7854         %latitude_in_radians
0.7854          %longitude_in_radians
0          %max_spin-down_param_order
-0.e-9   %value of first spindown
0e-14   %value of second spindown
./testT8_1800    %name_of_time-stamps_file
```

Since simcoh need both LHO and LLO, both overlapping and non-overlapping SFT, we have to repeat that command four times, each using a different gdbinit file(which specify different SFT file names, detectors names and GPS starting time, but In.data file remain the same), as the example in the "Simulation Studies" part illustrated. It should be noted that the filename for the SFT files must be "FourierH"(LHO non-overlapping), "FourierH1"(LHO overlapping), "FourierL"(LLO non-overlapping) , "FourierL1"(LLO overlapping), or else simcoh cannot find the files. And the overlapping SFT GPS starting time must be set 900 seconds later than non-overlapping SFT

c) After all the data are generated; you can use simcoh to search over them for a signal. Of course you have to specify the parameters for simcoh, too. And some of them must be set according to corresponding makefakedata_v2 parameters. Here is a list of simcoh arguments:

arguments	meaning	default (if not specified)
-h	display argument list	
-t	use coherent summing	use incoherent summing
-mf0	do not use matched filter	use matched filter
-freq num	the signal frequency is set to num.	1000.4Hz
-ff num	set the first frequency in Hz to num, must be the same as in makefakedata_v2	1000Hz
-fb num	set the width of the frequency band to num, must be the same as in makefakedata_v2	10Hz
-psi num	polarization angle, must be the same as in makefakedata_v2	0.8
-fdir string	set the directory of the SFT files to string	../makefakedata/fdata/
-edir string	set the directory of the of earth and sun ephemeris data files to string	/home/rarmen/opt/lscsoft/lal/share/lal/
-s num	set the GPS start time to num (must be the same as non-overlapping SFT GPS start time)	693326229
-g num	The searching grid on SNR map is set to num*num	96
-iter num	set the total number of non-overlapping time segments	1440(one month)

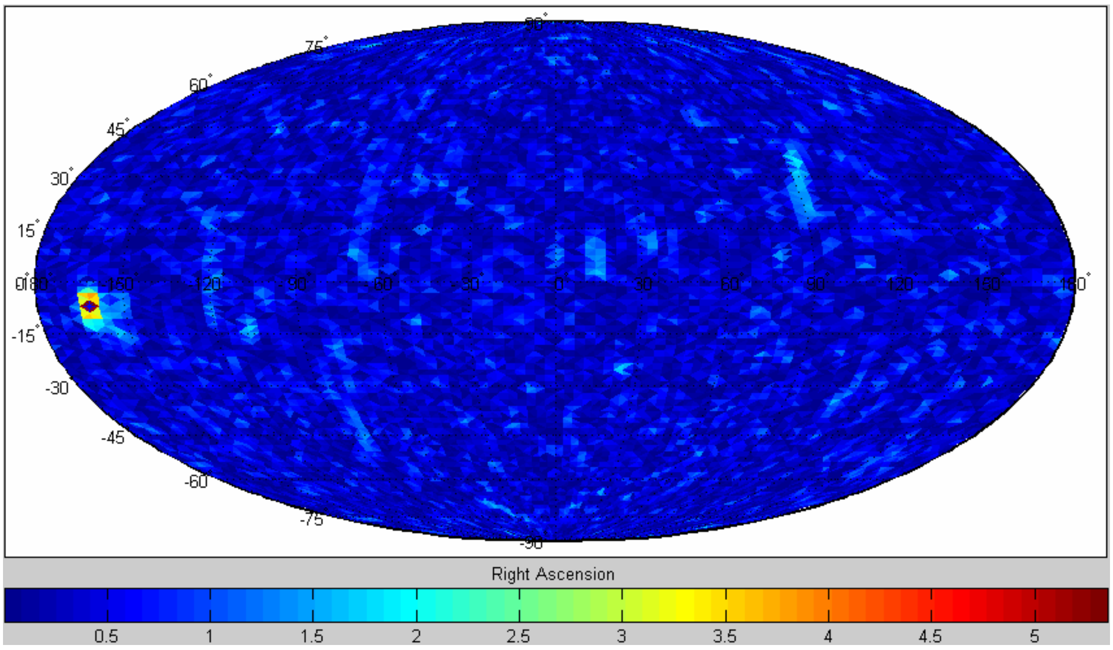
d) It must be noted that simcoh can only search over 3 frequency bins near the signal frequency. Also the segment length is always 1800s (half an hour). Besides, simcoh does not search over frequency spin down or polarization angle. All the tests done on simcoh are using linear polarized signal, with Across=0.

e) After simcoh finishes, it generates 3 .txt files. If the summing is incoherent, the filenames are 1.txt~3.txt. If the summing is coherent, the filenames are t1.txt~t3.txt.
These files should be processed by a matlab script named readfile.m.
Depending on the filenames, you can either use readfile("1.txt","2.txt","3.txt") or readfile("t1.txt","t2.txt","t3.txt"). This would again generate 3 files "f1.fig"~"f3.fig". Use matlab to open these files to see the SNR map and related numerical results.

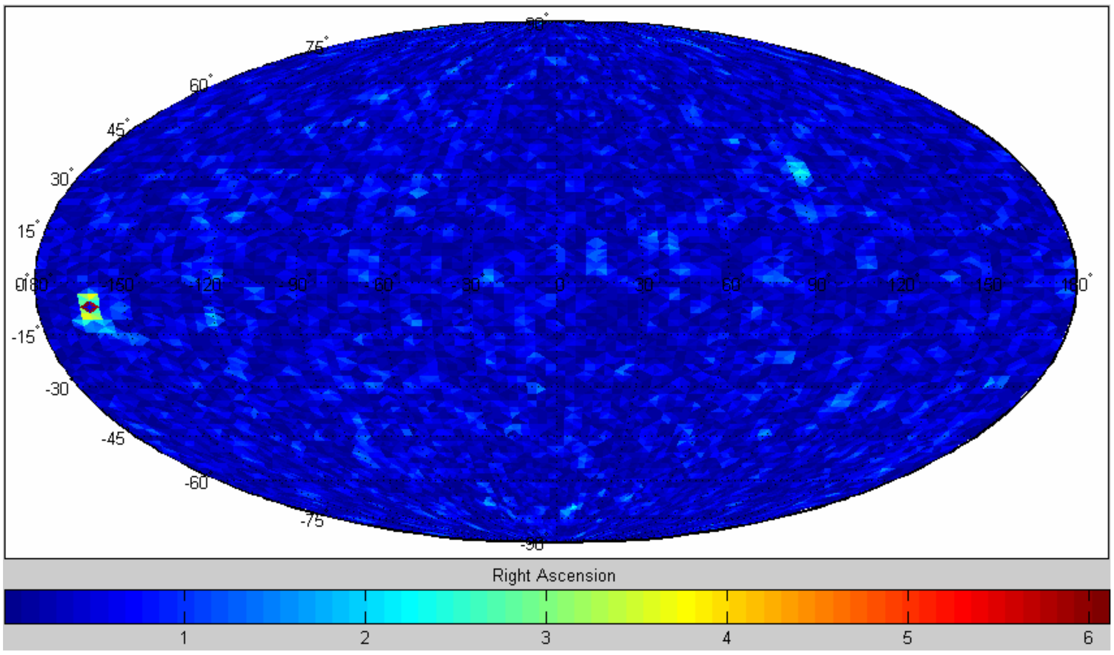
Appendix B:

First we want to see how coherent summing works for low SNR scenario. Since in PowerFlux the typical SNR is about 8, we would expect the coherent summing to boost SNR to about 11. So we set Aplus=0.00089, sigma=0.1, frequency=1000.4Hz, Ra= -2.82743, Dec= -0.125663, and we get:

Incoherent summing, with highest SNR=7.8605



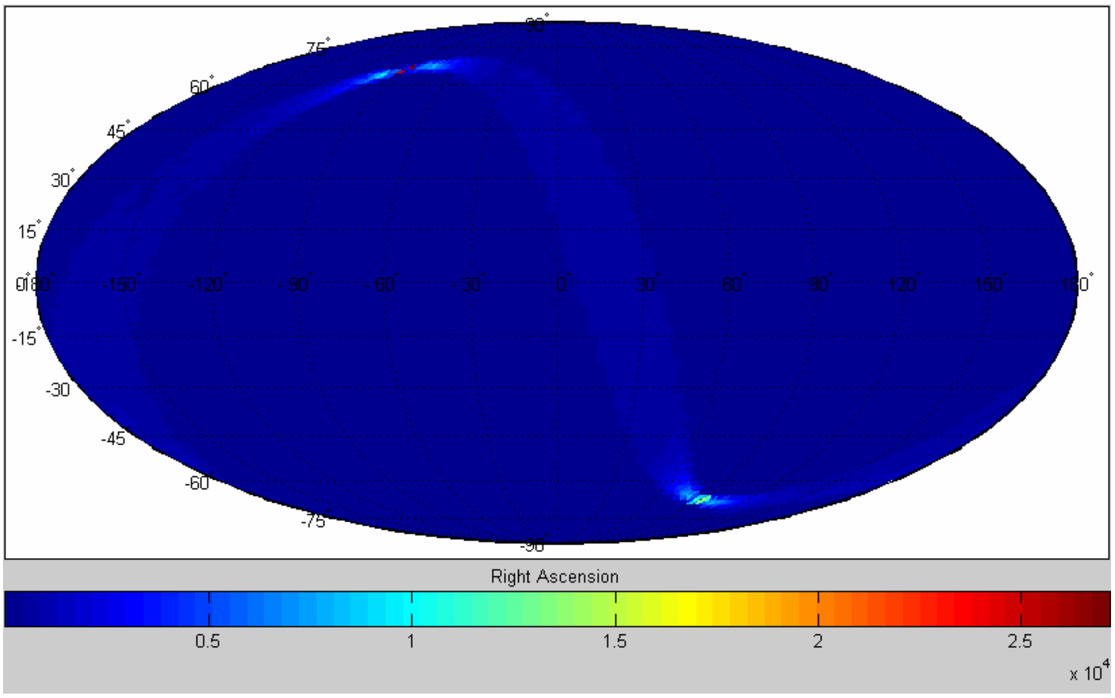
Coherent summing, with highest SNR=10.44



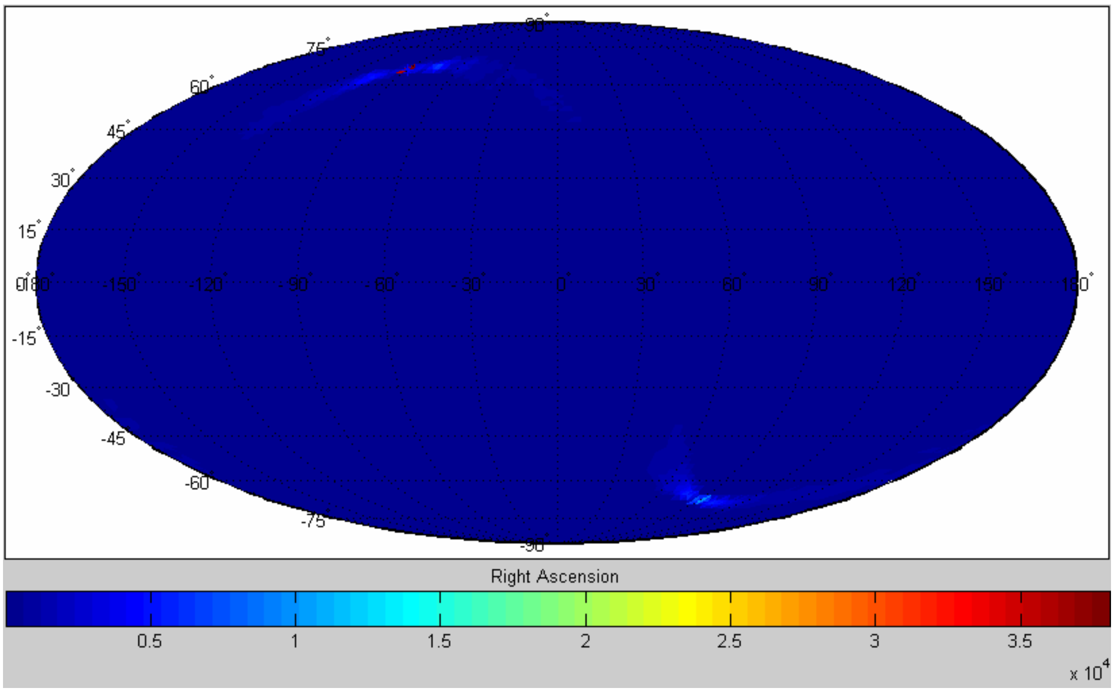
Next we want to show some high SNR plots. The following plots are for signal source located at special position in the sky map. In makefakedata the parameters are set to be: signal frequency=1000.4 Hz, Aplus=0.08,sigma=0.1:

1. source location at ecliptical north pole (Ra= -pi/2, Dec= 1.1454):

Incoherent summing, with highest SNR=73483,

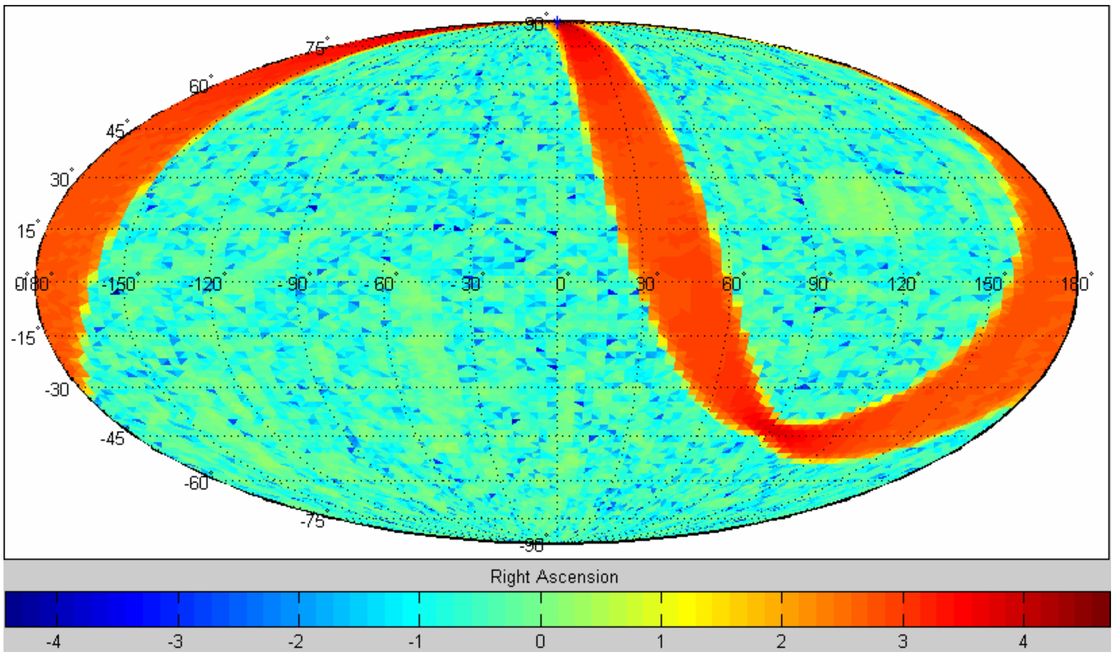


Coherent summing, with highest SNR=106178

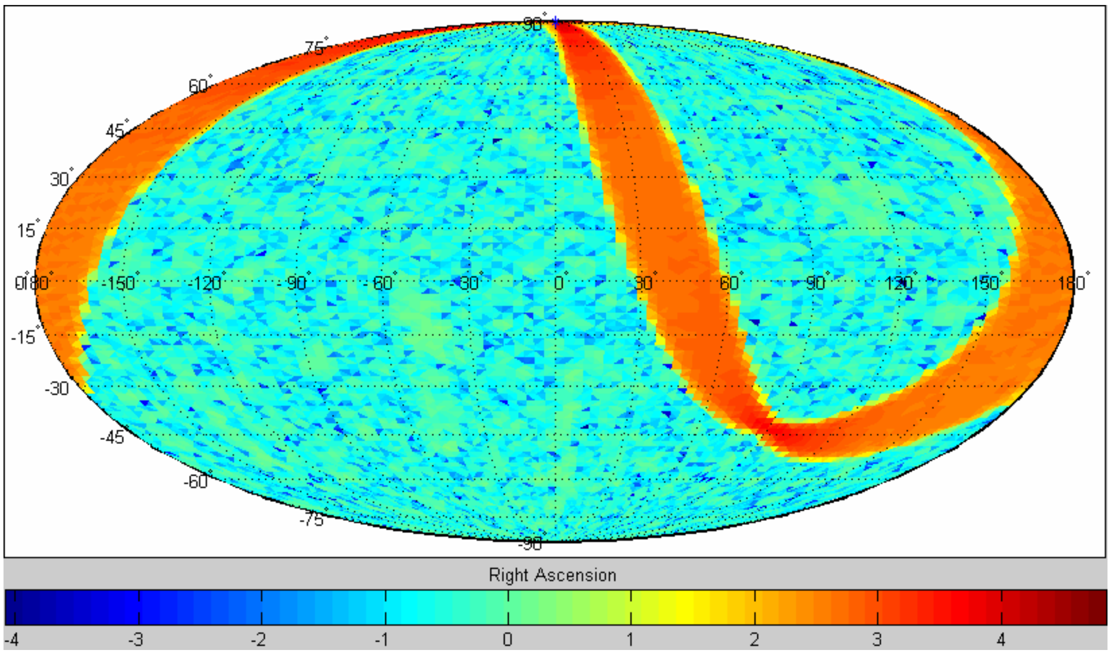


2. location at celestial north pole (In order to show a clear high-SNR ridge, the number at the color bar are set to be $\log_{10}$(SNR), instead of just SNR):

Incoherent summing, with highest SNR=64866.6,

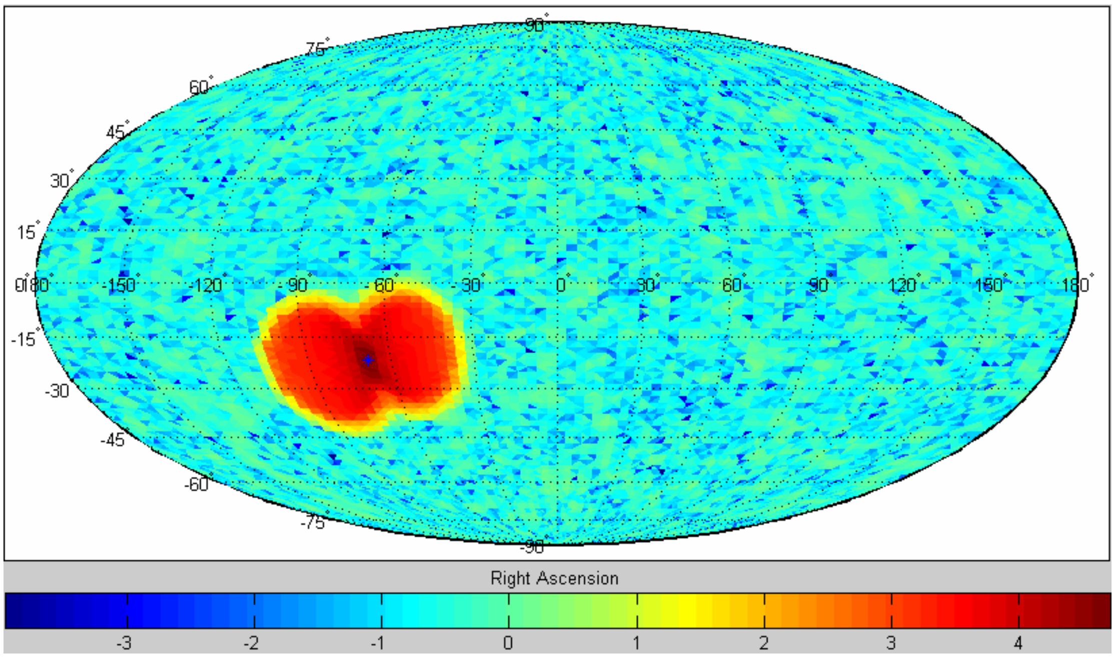


Coherent summing, with highest SNR=94142.8

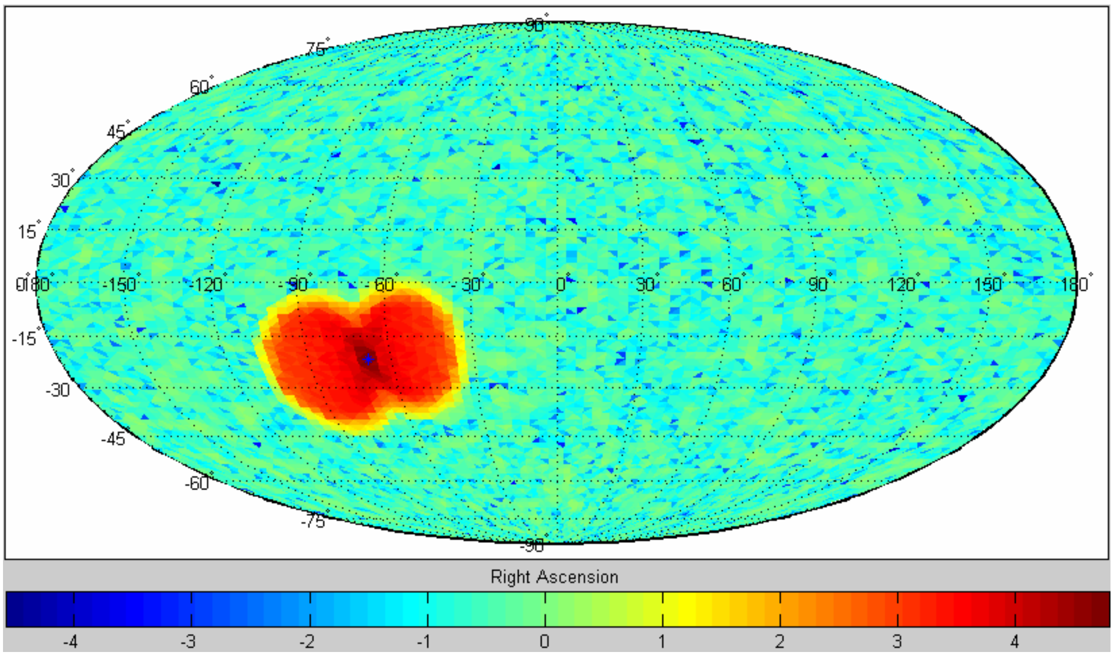


3. source location (Ra= -1.19380, Dec= -0.376991). The direction of the source is parallel with the average velocity of the detectors during the simulation (In order to show a clear high-SNR ridge, the number at the color bar are set to be $\log_{10}$(SNR), instead of just SNR):

Incoherent summing, with highest SNR=71240.5,



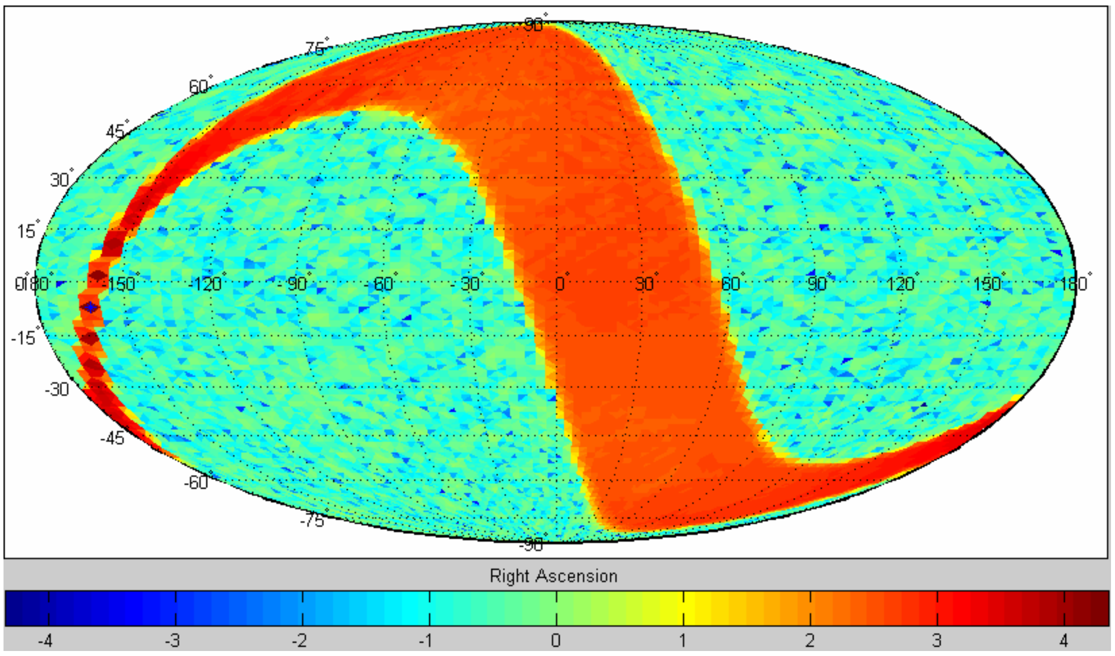
Coherent summing, with highest SNR=102659



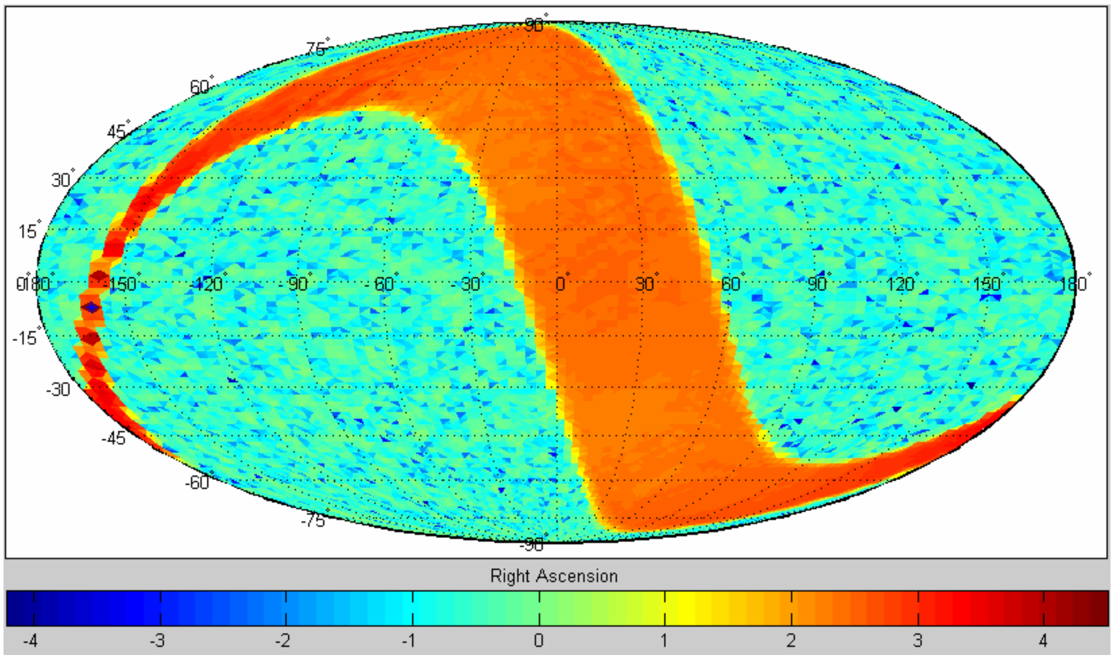
In these two plots, the high SNR ridge is basically a single blob, because the average velocity of the detector points toward the location of the signal source.

4. source location (Ra= -2.82743,Dec= -0.125663). The direction of the source is perpendicular to the average velocity of the detectors during the simulation (In order to show a clear high-SNR ridge, the number at the color bar are set to be $\log_{10}$(SNR), instead of just SNR):

Incoherent summing, with highest SNR=66105.7



Coherent summing, with highest SNR=94429.2



Here the high SNR ridge is a great circle, just as in the case where the signal is located in ecliptical north pole. In both cases, the direction of the source is perpendicular to the detector average velocity.