DEVELOPMENT OF AN INSPIRAL SEARCH PIPELINE USING THE NESTED SAMPLING TECHNIQUE

KIERSTEN RUISARD

Abstract. We introduce a new technique for gravitational wave signal detection and parameter estimation which we hope will provide a more sensitive algorithm to search for wave signals originating from compact binary coalesences (CBCs) associated with gamma-ray burst events (GRBs). In order to provide a better search algorithm, we focus on applying Bayesian inference to the search for signal around GRB events, using a nested sampling technique to estimate the evidence integral, a notoriously difficult to calculate quantity which often limits the applicability of Bayes Theorem. We design a pipeline tailored to the search for GRB-associated inspirals, and test its performance on both software injection recovery and off-source data. We also explore the usefulness of a Bayesian coherency test as a means of glitch rejection.

Date: August 9, 2010.

1. Introduction

The search for gravitational radiation, the rippling of space-time created by all masses accelerating through space, constantly endeavors towards the accuracy and range needed to make a confident detection. With recent and ongoing advances made to current interferometers and a new generation of advanced detectors in the process of being constructed, detection of a gravitational wave signal may be within our grasp. One of the most promising sources of gravitational wave signals is the inspiral phase which precedes the coalesence of compact binary systems. It is conceivable that the first signal detection might be found in the short, intense burst of radiation emanated during the final stages of inspiral, as the two objects merge.

As well as being a loud source of gravitational radiation, compact binary coalescences (CBCs) might also create gamma-ray bursts (GRBs), bright flashes of gamma-rays that last anywhere from seconds to minutes. These events are distributed isotropically over the sky, suggesting they originate from extragalactic sources. The brightness of these bursts infers that they are beamed along the source's spin axis rather than emitted in all directions, as this would require a seemingly unphysical release of energy. This beaming effect is characterized by the jet emission, or opening, angle. Studies of the afterglows of long-duration GRBs in the x-ray, optical and radio bands reveal that they originate from supernova events. However, similar surveys to determine the nature of short-duration GRBs ($T_{90} < 2s$) have yet to detect the same afterglow effect. As a result, the possible scenarios and opening angles for short GRBs are under debate. The coalescence of two neutron stars or a neutron star and black hole system provides a possible progenitor for these elusive events. ([1], [2]). When searching for gravitational waves emanating from GRBs, we can use the knowledge that the γ-rays are emitted along the spin axis of the system. The use of Bayesian inference gives us the tools to incorporate this background information into our search.

Although CBC events provide sources of high energy gravitational waves that we are hopeful to detect in the future, sensitivity of the current generation of detectors is limited by background noise. Transient events, known as glitches, mimic gravitational wave signals, requiring more rigorous methods for data analysis. For this reason, we investigate the use of a new technique for improved discrimination of signal above background events, in order to produce a more sensitive algorithm. The main component of this new search tool uses a nested sampling technique to estimate the evidence value, a quantity extremely useful when applying Bayesian statistics

The extent of the project studied under the NSF Gravitational Wave International REU Grant covered the following areas:

- Incorporating the knowledge that the inclination angle of a binary system must be lie within the range of the GRB opening angle into the nested sampling search algorithm.
- Examining the effect of prior knowledge of the inclination angle on parameter estimation when using the nested sampling code. (Section 2)

- Construction of a pipeline to incorporate the nested sampling technique (see [5] for more details) into a GRB search. (Section 3)
- Tackling the complications that arise with processing larger lengths of data efficiently and consistently. (Section 3.1)
- Testing code on injections to observe performance and efficiency (Section 3.2)
- Incorporating a scheme for glitch rejection into the pipeline, using a Bayesian coherency test introduced by [5]. (Section 4)

1.1. **Bayesian Statistics.** Bayesian inference, based on simple probability theory, is a useful tool for gravitational wave searches, as it allows a way to incorporate already known information into analysis. Bayes theorem can be stated as such,

$$p(H|\vec{d},I) = \frac{p(\vec{d}|H,I)p(H|I)}{p(\vec{d}|I)} \tag{1}$$

where $\vec{d}$ represents the observed data, H is a hypothesis, and I is relevant background information.

Here, $p(H|\vec{d},I)$ is the posterior probability, or the probability of hypothesis H given the data $\vec{d}$ and background information. $p(\vec{d}|H,I)$ is the likelihood function, read as the likelihood of observing this set of data $\vec{d}$ assuming the hypothesis H to be true. $p(H|I)$ is the prior probability, or probability that this hypothesis is true before analyzing $\vec{d}$. Finally, $p(\vec{d}|I)$ is the marginal likelihood over all hypotheses, also known as the evidence. This factor is the likelihood of the data, regardless of hypothesis.

The evidence is the sum of likelihoods over all hypotheses, where $p(\vec{d}|I) = \sum_i p(\vec{d}|H_i,I)$ as long as $\sum_i p(H|\vec{d},I) = 1$. That is, the set of models H_i must be representative of all the possible hypotheses. This evidence is often ignored as a normalizing factor, since there is rarely such an exhaustive set of hypotheses that constitutes all possible conclusions.

Given a set of data, we can calculate the probability density across the parameter space using the following restatement of Bayes Theorem:

$$p(\vec{\theta}|H,\vec{d},I) = \frac{p(\vec{d}|\vec{\theta},H,I)p(\vec{\theta}|H,I)}{p(\vec{d}|H,I)} \tag{2}$$

where $p(\vec{\theta}|H,\vec{d},I)$ is the probability of $\vec{\theta}$, a vector in parameter space. Likelihood values $p(\vec{d}|\theta,H,I)$ and prior probability function $p(\theta|H,I)$ are relatively easy to determine. In this case, the evidence $p(\vec{d}|H,I)$ is the likelihood marginalized over all values of θ. This evidence becomes important when distinguishing between competing models. It is also this quantity which is most difficult to calculate, and for which nested sampling provides a means of estimation.

Application of Equation 2 allows for parameter estimation to be performed, by gaining knowledge of the distribution of the posterior over each parameter in $\vec{\theta}$. Using the posterior probabilities from Equation 1, we can distinguish between possible hypotheses by measuring the odds ratio O:

$$O_{1,2} = \frac{p(H_1|\vec{d}, I)}{p(H_2|\vec{d}, I)} \tag{3}$$

$$O_{1,2} = \frac{p(H_1|I)p(\vec{d}|H_1, I)}{p(H_2|I)p(\vec{d}|H_2, I)} \tag{4}$$

$$O_{1,2} = \frac{p(H_1|I)}{p(H_2|I)} B_{1,2} \tag{5}$$

where $B_{1,2}$ is a ratio of likelihoods known as the Bayes factor. When drawing a comparison between two models, we assume the prior odds $p(H_1|I)$ and $p(H_2|I)$ to be known. We calculate the Bayes factor to provide a measurement of which model is favored by the data.

$$B_{1,2} = \frac{p(\vec{d}|H_1, I)}{p(\vec{d}|H_2, I)} \tag{6}$$

Calculation of the Bayes factor relies on being able to evaluate the following evidence integral:

$$Z = p(\vec{d}|H, I) = \int_{\vec{\theta}} p(\vec{\theta}|H, I)p(\vec{d}|H, \vec{\theta}, I)d\vec{\theta} \tag{7}$$

There are several difficulties with evaluating this integral. First, it can only be evaluated analytically in the simplest cases, certainly not when there are a large number of correlated parameters. Second, we use a post-newtonian model for the inspiral of a compact, non-spinning binary, which has 9 parameters. Integrating over a 9 dimensional parameter space can be computationally demanding, and most numerical techniques use too much CPU time for evaluation of this integral to be included in search pipelines. The nested sampling algorithm provides a novel approach to estimating the evidence integral by sampling from the likelihood function.

1.2. **Nested Sampling Algorithm.** The nested sampling algorithm is a tool developed by John Skilling ([3], [4]) to estimate the evidence integral. On the most basic level, the algorithm samples from the likelihood function and then sums the area under the likelihood as such:

$$evidence = Z = \int L d\chi \tag{8}$$

where χ represents the fraction of prior mass being evaluated. Before the samples are generated, the algorithm is given a function of prior probability for each parameter, $p(\theta|H, I)$. Then, a number of live points, N_{Live}, are sprinkled across the parameter space and the likelihood of each sample $p(\vec{d}|\theta, H, I)$ is calculated. The sample with lowest likelihood is then discarded and resampled at a higher likelihood. The steps of each live sample to

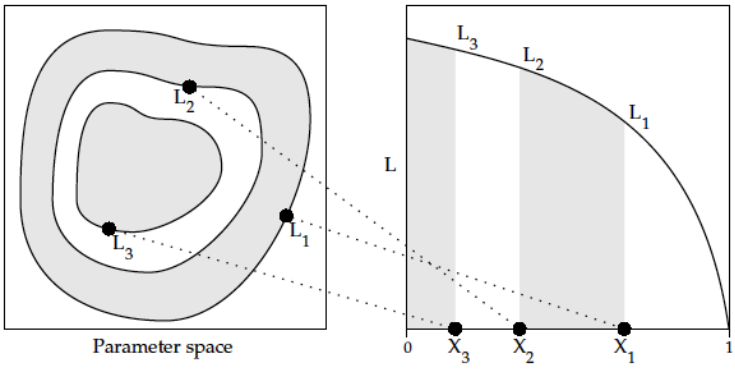


FIGURE 1. During the nested sampling process, live points are moved to increasingly higher likelihoods, as the theoretical contour in the parameter space shrinks. Figure borrowed from [3].

points of higher likelihood are made using Monte Carlo methods. Through this process of resampling, a set of live samples is created that is concentrated around the area of highest likelihood in the parameter space.

As seen in Figure 1, each sample can be seen as lying on a contour of equal likelihood. As the set of live samples are moved into higher likelihoods, the contour is shrinking towards the point of highest likelihood, and the prior mass contained within the contour is constantly decreasing. The term "nested sampling" refers to consistently sampling over a new domain which is located within the previous domain. As the likelihood increases, prior mass shrinks by a factor of $e^{1/n}$, where n is the iteration number. As the number of live samples N_{Live} is constant, the density of samples in each subsequent area of parameter space increases with each iteration.

Having sampled from the likelihood function across the parameter space with respect to the prior distribution, the evidence integral Equation 7 can be approximated by summing over all the samples, while taking into account the relative contribution of each iteration to the evidence. Therefore, the evidence for each iteration increases by adding the weighted likelihoods for each iteration:

$$Z_i = \sum_i L_i \omega_i \tag{9}$$

where the weight factor ω_i is the difference between the prior masses at subsequent contour levels,

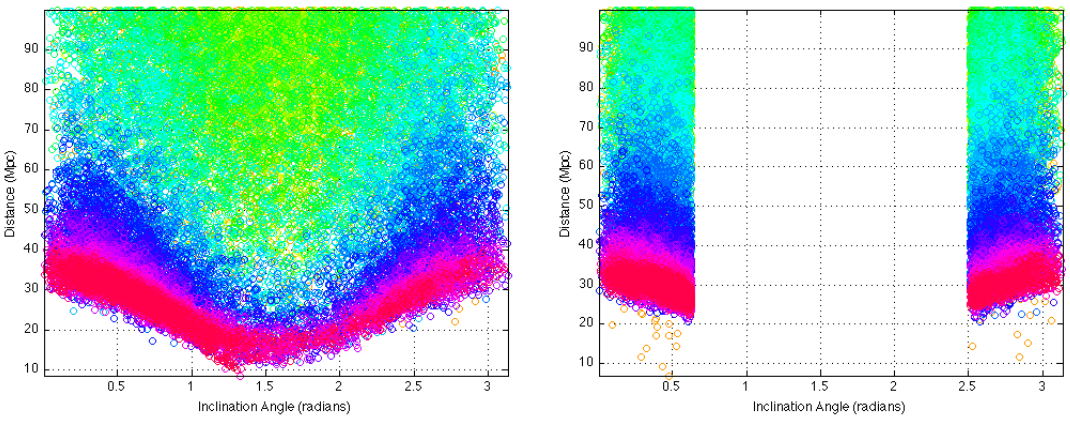
$$\omega_i = p(\theta_i|H, I)d\theta \tag{10}$$

Posterior samples can be created from the set of likelihood samples through the following relationship:

$$p_i = L_i \omega_i / Z \tag{11}$$

The algorithm developed by John Skilling [3] has been implemented into a nested sampling code specifically for gravitational wave searches by John Veitch [5]. It is this implementation of the code that is being utilized and tested in the following sections.

2. Effects of Prior Knowledge



(a) (b)

FIGURE 2. Plot of distance, in MegaParsecs, over inclination angle, in radians, for a set of nested samples collected for a software injection. The colors indicate likelihood values, with red being the highest and orange being the lowest. In plot (a) the inclination angle is sampled over all values from 0 to π. In plot (b), the inclination angle is limited between $\iota < 0.2\pi$ and $\iota > 0.8\pi$

The prior probability function, $p(\theta|H, I)$, allows one to incorporate prior knowledge into parameter estimation. Searching for inspiral signals around GRB triggers limits our search to binary systems whose angular momenta are oriented towards or away from us, placing the earth in the path of the emitted γ-ray jet. We know that the inclination angle, ι, must be less than or equal to the GRB opening angle in order for a GRB event to be observed.

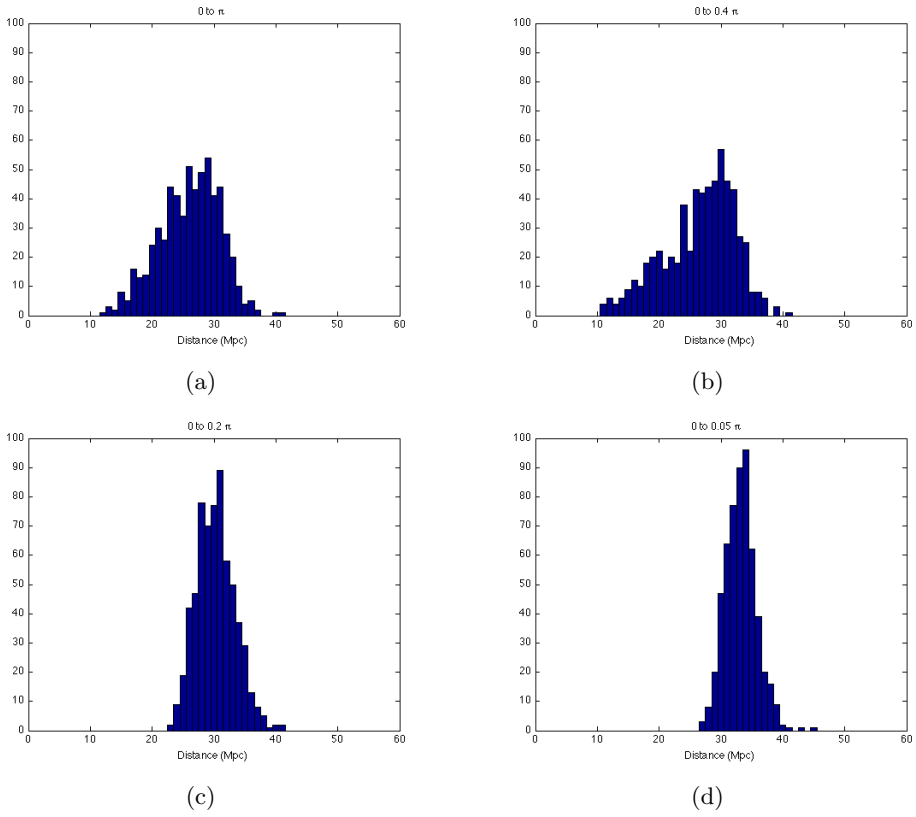
Constraining the inclination angle allows for more accurate estimation of source distance, as there is some degeneracy in the distance and inclination angle parameters. A binary oriented directly towards earth will have both a + and $\times$ polarization, resulting in a higher strength signal that will be detectable out to greater distances. A binary whose plane lies along our line of sight, or an edge-on view, will have only a $\times$ polarization, and the source must be closer in order to be seen. For a signal with given amplitude and unknown inclination angle, the distance will correlate with inclination angle, as seen in Figure 2.

To best tailor the nested sampling code for GRB-based searches, an option was added to constrain the prior probability function of the inclination angle to an arbitrary degree.

When the code samples in the parameter space, it will only draw from areas described by the prior.

An application of the nested sampling code's distance estimation, using the described limits, can be seen in Figure 3.

The range of jet-emission opening angles of short GRBs is not well-constrained, as these events are both rarer than their long-duration counterparts and the quickness of these events means that afterglows have not yet been detected [1]. Lacking a confident estimate, for all work described below, we limited the inclination angle to vary within the ranges of $(0, 20°)$ and $(160°, 180°)$



(a) (b) (c) (d)

Figure	Inclination Angle	Standard Deviation
(a)	0 to π	$\sigma = 5.3743$
(c)	0 to 0.4π	$\sigma = 5.7575$
(d)	0 to 0.2π	$\sigma = 3.0898$
(d)	0 to 0.05π	$\sigma = 2.4987$

FIGURE 3. Posterior PDF Distributions of source distance (Mpc), with inclination angle constrained as indicated, run on the same face-on software injection.

3. Inspiral Search Pipeline

Another aim of this project was to incorporate the nested sampling code into a fully automated pipeline, for easy application in GRB-triggered searches. A python script which creates a directed acyclic graph (DAG) defining a sequence of jobs, which constitute the pipeline, for submission to a computing cluster. Figure 4 illustrates this pipeline.

When using the nested sampling code in the GRB search, we examine a 6 second window around the GRB time, from 5s before to 1s after. The nested sampling code searches most effectively and efficiently when run over durations much shorter than 6 seconds. The ending time of the inspiral signal has a very high amplitude and short duration. We search for this end time by dividing the 6 seconds of data into segments of length 0.1s each. Also, division of the data into smaller segments allows us to take advantage of the ability of computing clusters to run jobs in parallel, reducing the overall time of the analysis. However, merely locating the end time does not characterize the entire signal, the duration of which may last up to 10s prior to endtime. To take this into account, it is necessary to recombine the data and be able to calculate a detection statistic and estimate parameters that reflect properties of the entire waveform. Application of Bayes theorem allows for coherent analysis of the data set as a whole.

3.1. **Combining Output.**

Finding the Global Bayes Factor. For a general case of 60 segments that we wish to combine for one result, we need to find the global Bayes Factor, which will be a measurement of the odds of a signal appearing at any time within the 6 second GRB window.

As seen in Equation 6, the Bayes factor for any one segment i is simply the ratio of likelihoods between the signal and noise models (respectively, m and n):

$$B_{m,n} = \frac{p(d_i|m_i, I)}{p(d_i|n_i, I)} \tag{12}$$

where the signal evidence $p(d_i|m_i, I)$ and noise evidence $p(d_i|n_i, I)$ are output by the nested sampling code. Therefore, Equation 12 is known for each of the 60 segments. From this knowledge we wish to calculate the global Bayes Factor.

We start by comparing the posterior probabilities of each model. We use the shorthand M_i to represent the posterior probability $p(m_i|d_i, I)$, the probability that there is a signal and Gaussian noise present in segment i, and N_i for posterior probability $p(n_i|d_i, I)$, or the probability that there is only Gaussian noise in segment i. Comparing the probabilities of signal and noise in all 60 bins, we start with the odds ratio:

$$O_{global} = \frac{M_1 \, or \, M_2 \, or ... \, M_{60}}{N_1 \, and \, N_2 \, and \, N_3 \, ... and \, N_{60}} \tag{13}$$

The above statement can be read as "The odds that there is a signal present somewhere in the data is equal to the probability that there is signal in one segment from a set of 60 over the probability that there is noise in all 60 segments." While intuitively sound, the

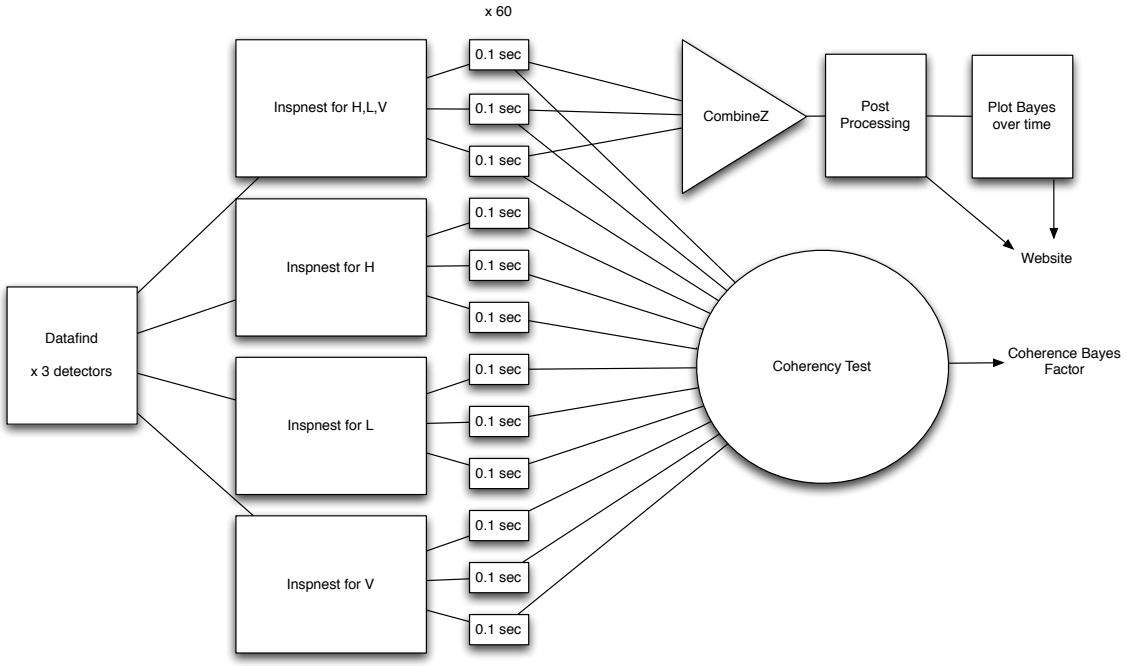


FIGURE 4. The structure of the GRB search nested sampling pipeline. Using an input of trigger time and a number of configuration options, the pipeline first finds the data, then runs the nested sampling code for a network of detectors (denoted here as three detectors H, L, and V), and for each detector individually. Additional scripts combine and process the data, as well as perform a coherency test.

above expression considers each segment independently. When looking at the probability that a signal is present in segment 1, one must simultaneously look at the probability that it is absent from all other segments. The expression becomes:

$$O_{global} = \frac{(M_1\, and\, N_2\, and\, N_3\, ...and\, N_{60})\, or\, (N_1\, and\, M_2\, and\, N_3\, ...and\, N_{60})\, or...}{N_1\, and\, N_2\, and\, N_3\, ...and\, N_{60}} \tag{14}$$

Here the probability of the signal being present in any segment is multiplied by the probabilities of the signal *not* being present in any other segment.

In probability theory, *and* represents multiplication of probabilities and *or* represents addition. Taking this into account, many elements cancel, leaving us with:

$$O_{global} = \frac{M_1}{N_1} + \frac{M_2}{N_2} + ...\frac{M_{60}}{N_{60}} = \sum_i \frac{M_i}{N_i} \tag{15}$$

For $i \in (0, 60)$ Remembering that M_i and N_i represent the posterior probabilities of signal and noise, respectively, and applying Bayes Theorem (Equation 1), we can make the statements:

$$M_i = p(m_i|d_i, I) = \frac{p(d_i|m_i, I)p(m_i|I)}{p(d_i|I)} \tag{16}$$

$$N_i = p(n_i|d_i, I) = \frac{p(d_i|n_i, I)p(n_i|I)}{p(d_i|I)} \tag{17}$$

$$O_{global} = \frac{1}{60}\sum_i \frac{p(m_i|I)p(d_i|m_i, I)}{p(n_i|I)p(d_i|n_i, I)} \tag{18}$$

The factor $\frac{1}{60}$ in Equation 18 accounts for the fact that the prior probability of a signal being present in the data, $p(m|I)$, should be the same no matter what length of time we examine, as we expect to see only one signal associated with any GRB. Summing over the prior means that it accumulates. Accumulation of the prior amounts to the statement that there is a certain rate of events and, when looking at longer lengths of time, the probability of finding a signal increases accordingly. When looking around a GRB event, we expect to see only one signal, and our prior probability does not change whether we look at one time segment around the event or 60 segments. Therefore, when looking at multiple segments associated with the same event, we must be sure to take the mean of the signal prior $p(m|I)$.

Therefore, we still need to divide the Bayes Factor by 60, making the global Bayes Factor equal to the mean of Bayes factors.

$$B_{global} = \frac{1}{60}\sum_i \frac{p(d_i|m_i, I)}{p(d_i|n_i, I)} = \langle B_i \rangle \tag{19}$$

Using this global Bayes Factor, we are able to make a definitive statement as to whether or not there is a detectable signal within the 6 second GRB triggered window. While technically Bayes Factors above 0 indicate favorable odds for existence of a signal, due to noise we set a non-zero detection threshold based on the loudness (that is, magnitude of Bayes factors) of the background.

Weighted Likelihoods. In addition to determining the total Bayes factor, the samples from each segment must be combined in order to perform parameter estimation for the entire data set. In order to combine the samples, the likelihood of each sample must be weighted appropriately for consistency across all 60 segments.

The nested sampling code outputs the likelihood for each sample, $p(d_i|\vec{\theta_i}, m_i, I)$ before being marginalized, where $\vec{\theta}$ is a vector containing specific value in the parameter space. As explained above, the likelihood of the signal being in any *one* segment is the likelihood of a signal being present in that segment and absent in all other segments. Therefore, the likelihood L that the signal appears in just one segment is:

$$L_i = p(d_i|\vec{\theta_i}, m_i, I) * \prod_{j\neq i} p(d_j|n_j, I) \tag{20}$$

In this case, the likelihoods of data points in a segment are weighted by multiplying them by the noise evidence for all other segments. This should make all samples comparable, necessary to create a set of global posterior samples.

Posterior Samples. In the case of an above-threshold Bayes factor, we wish to estimate the parameter values of the signal. In order to do this, we want to observe the distribution of the posterior probability function, specifically which values are occupied by the highest probabilities. Samples from the posterior can be produced from the output of the nested sampling code, which provides parameter values θ_i and the likelihood L_i for each sampled point for each of the 60 segments. Equation 11 provides the relationship through which posterior probabilities can be calculated from each sample, using the likelihood, evidence and weighting factor ω_i, which accounts for the fact that the nested sampling algorithm samples over a constantly decreasing fraction of prior mass.

However, as we simply wish to observe the distribution of posterior probability across the parameter space, direct calculation of the posterior is not necessary. After applying the above procedure for weighting the likelihoods of each sample, we wish to compile a set of samples representative of the posterior probability density function (PDF). We do this by rejection sampling from all 60 sets of nested samples to create a single set of posterior samples, keeping or discarding them proportional to their density.

In the end, we wish to have a density of samples proportional to the posterior probability function:

$$D_P(\vec{\theta}_i) \propto p(\vec{\theta}_i|d_i) \tag{21}$$

We are able to make a statement about D_N, the density of nested sampling points, which we must manipulate to find D_P.

The prior volume $\chi = \frac{Volume\ inside\ likelihood\ contour}{Volume\ of\ Prior}$ is 1 before the first iteration. As $x \to \infty$, the contour, indicating highest likelihood values, shrinks, and therefore the prior volume becomes more constrained.

Given a number N_{live} of points, ξ (the proportion of the prior with likelihood greater than a given L) shrinks with each iteration by a factor of $e^{1/N_{live}}$, where N_{live} is the number of live points. Although the area under the likelihood curve is shrinking, sampling continues at the same rate. Therefore, the density of samples after each iteration is proportional to $e^{x/N_{live}}$ and the prior probability.

$$D_N(\vec{\theta_i}) \propto p(\vec{\theta_i})e^{x/N_{live}} \tag{22}$$

In order to find a density proportional to likelihood, we must weight each sample according to its iteration number x. At this stage, samples are still kept separate in their own time bins, because the iteration number x is then simply the order of points.

Remembering Bayes Theorem, Equation 1, and knowing that the evidence $p(d_i|I)$ can be ignored as a normalizing constant, we set the probability density equal to the likelihood times the prior:

$$D_P(\theta_i) = p(d_i|\theta_i)p(\theta_i) \tag{23}$$

Substituting from Equation 22, we can say:

$$D_P(\theta_i) = \frac{D_N(\theta_i)}{e^{x/N_{live}}} * p(d_i|\theta_i) \tag{24}$$

Therefore, the density weight D_i for every sample θ_i can be expressed as

$$D_i = L_i * e^{-x/N_{live}} \tag{25}$$

To resample from the posterior PDF, the code generates a random number $u \in (0, 1)$ for each sample. If $u < \frac{D_i}{D_{max}}$, where D_{max} is the highest density weight, the sample is added to the posterior sample set. If not, it is discarded. This simple algorithm chooses points randomly, but favors those with higher likelihood, weighted by density so as not to *over favor* points from higher iterations.

3.2. **Results.**

Background Estimation. We ran the code on 20 off-source windows around open-box GRB090709B to estimate which value of the Bayes Factor should be considered as a detection threshold. In the case of this event, we define the detection threshold as the point above which only 5% of the distribution lies, which should give a 5% false alarm rate when searching for real signals. Based on only 20 off-source times, we estimate the detection threshold to be $B_{thresh} \approx 4$ for this particular GRB event. For the distribution of off-source Bayes factors for this particular event, see Figure 3.2 on Page 18. B_{thresh} can be adjusted according to desired false alarm probability.

Injection. The nested sampling code has been tested on a number of injections, both software and hardware. We used a 9-dimensional model in all investigations, with the parameters chirp mass, symmetric mass ratio η, right ascension, declination, time, distance, phase ϕ, polarization angle ψ, and inclination angle ι. For most injections, the sky location was fixed, so the code was searching over a 7 dimensional space. For the parameter estimation on one found injection, see Figure 6 on 19. The distribution of the Bayes factor over time for this particular injection can be seen in Figure 7 on Page 20.

Pipeline Efficiency. Runs done on 3000 injections with a variety of SNRs demonstrate the pipeline's efficiency at recovering signals. For the plot of this efficiency over SNR, see Figure 8 on Page 20. Thresholds were based on the single highest Bayes Factor when running over 3000 off-source times, prior to making injections. Choosing a threshold based the single loudest off-source time rather than using the 5% threshold gives a $\frac{1}{3000}$ false alarm probability, rather than $\frac{1}{20}$. Using this loudest background event as the distinguisher, the pipeline does not perform well at recovering sources until a coherency test, described in Section 4, is included. The coherency test drastically reduces the background, allowing us to detect more sources confidently.

Although Bayes factor is not linearly related to SNR, there is some correlation. For the distribution of the Bayes factors vs. SNR for all injections, recovered and non-recovered, see Figure 9 on Page 21.

4. Coherency Test

Using a network of detectors is beneficial to analysts in more than one way. Besides allowing for better parameter estimation, having multiple detectors allows for coherency tests. By cross-checking the data collected by separate detectors, it is possible to determine if an observation is consistent. A gravitational wave signal would appear in all detectors with physically consistent parameters. Instrumental glitches would perhaps resemble a detection, but would be unique for each detector.

Here we implement a Bayesian coherency test, introduced by [5], which relies on the Bayes Factor as a means to distinguish between coherent and incoherent models. The incoherent model, M_{inc}, is a model which has different values for a parameter x on each detector, $x_1 \neq x_2$. The coherent model, M_{coh}, represents the special case for which identical parameters are identified on each detector, $x = x_1 = x_2$.

$$M_{inc} = \frac{1}{2\pi\sigma_1\sigma_2} \exp\left(-\frac{1}{2}\left[\frac{(d_1 - x_1)^2}{\sigma_1^2} + \frac{(d_2 - x_2)^2}{\sigma_2^2}\right]\right) \tag{26}$$

$$M_{coh} = \frac{1}{2\pi\sigma_1\sigma_2} \exp\left(-\frac{1}{2}\left[\frac{(d_1 - x)^2}{\sigma_1^2} + \frac{(d_2 - x)^2}{\sigma_2^2}\right]\right) \tag{27}$$

In this case, each model is demonstrated for two detectors, noted as 1 and 2, although in practice this statement can be expanded to include an arbitrary number of detectors. x represents a single parameter. Again, although this is demonstrated for one parameter, x, Equation 27 can be expanded to include the 9-dimensional parameter model used in this paper.

To gain the evidence for each model, we integrate the likelihood function over the parameter space:

$$Z_{inc} = \int_{x_1}\int_{x_2} \exp\left(-\frac{1}{2}\left[\frac{(d_1 - x_1)^2}{\sigma_1^2} + \frac{(d_2 - x_2)^2}{\sigma_2^2}\right]\right) dx_2 dx_1 \tag{28}$$

$$Z_{coh} = \int_x \exp\left(-\frac{1}{2}\left[\frac{(d_1 - x)^2}{\sigma_1^2} + \frac{(d_2 - x)^2}{\sigma_2^2}\right]\right) dx = \int_x \exp\left(-\frac{1}{2}\sum_i \frac{(d_i - x)^2}{\sigma_i^2}\right) dx \tag{29}$$

for $i \in (1, 2)$ detectors.

For the incoherent signal, Equation 28, each detector can be integrated separately, resulting in the incoherent evidence being equal to the product of the evidence on each individual detector.

$$Z_{inc} = \int_{x_1} \exp\left(-\frac{1}{2}\left[\frac{(d_1 - x_1)^2}{\sigma_1^2}\right]\right) dx_1 \int_{x_2} \exp\left(-\frac{1}{2}\left[\frac{(d_2 - x_2)^2}{\sigma_2^2}\right]\right) dx_2 \tag{30}$$

$$Z_{inc} = \prod_i \int_{x_i} \exp\left(-\frac{1}{2}\left[\frac{(d_i - x_i)^2}{\sigma_i^2}\right]\right) dx_i$$

$$Z_{inc} = \prod_i Z_i \tag{31}$$

for $i \in (1, 2)$ detectors. For the LIGO Hanford, LIGO Livingston and Virgo detector network, we use $Z_{inc} = Z_H * Z_L * Z_V$.

Using these coherent and incoherent evidences, we can calculate a Bayes Factor to distinguish between the two. When comparing Z_{coh} to Z_{inc}, Z_{coh} should dominate if the signal has similar parameters $x_1 \approx x_2$. Z_{inc} should only dominate if $x_1 \neq x_2$, as the evidence for the coherent model will be comparatively low.

We find that there are a variety of tests that can be performed on our data, using the incoherent and coherent signal evidences, as well as the noise evidence. Remember that the Bayes Factor before coherency test is simply the coherent signal evidence over the noise evidence, as in Equation 6. We can expand on that comparison in several ways.

For the following, we calculate the odds of a signal appearing in any of n bins, using the following shorthand to represent posterior probabilities:

C= coherent signal model, $p(M|d, I)$

N= noise model, $p(N|d, I)$

I= incoherent signal model, $p(M_H|d_H, I) * p(M_L|d_L, I) * p(M_V|d_V, I)$

4.1. **Test 1: Coherency Mean.** The first test, called the Coherence-mean test, compares the probability of a coherent signal in one bin and incoherent signal in the other bins to the probability that there is incoherent signal in all bins.

$$O_{coh} = \frac{(C_1 \, and \, I_2 \, and \, I_3 \, and ... I_n) \, or \, (I_1 \, and \, C_2 \, and \, I_3 \, and ... I_n) \, or \, ...}{I_1 \, and \, I_2 \, and \, I_3 \, and ... I_n} \tag{32}$$

Understanding that *and* stands for the mathematical operator $*$ and *or* represents the operator $+$, we can cancel out most terms from the above statement, so that the odds ratio is simply a sum

$$O_{coh} = \sum_i \frac{C_i}{I_i} \tag{33}$$

Using the same process used to derive the global Bayes Factor above, Equation 19, we see that the coherent Bayes Factor is equal to the mean over all time segments:

$$B_{coh} = \langle \frac{Z_{coh}}{Z_{inc}} \rangle \tag{34}$$

4.2. **Test 2: Coherency-Noise Mean.** Another test that can be applied is the Coherent-Noise-mean. We compare the probability of a coherent signal in one bin and incoherent signal *or* noise in the other bins to the probability that there is incoherent signal *or* noise in all bins.

(35)
$$O_{coh} = \frac{(C_1\, and\, [I_2\, or\, N_2]\, ...and\, I_N\, or\, N_N])\, or\, ([I_1\, or\, N_1]\, and\, C_2\, and\, [I_3\, or\, N_3]\, ...and\, [I_N\, or\, N_N])\, or\, ...}{[I_1\, or\, N_1]\, and\, [I_2\, or\, N_2]\, and\, [I_3\, or\, N_3]\, ...and\, [I_N\, or\, N_N]}$$

Same as Equation 33, we cancel out most terms:

(36)
$$O_{coh} = \sum_i \frac{C_i}{I_i + N_i}$$

(37)
$$B_{coh} = \langle \frac{Z_{coh}}{Z_{inc} + Z_{noise}} \rangle$$

The coherency tests tend to bring down the detection threshold. Based on a run over 3000 off-source segments, the threshold Bayes Factor before coherency tests was Log $B_{thresh} = 2.5$. Using the coherency-mean, Log $B_{thresh} = 1.4$, which decreases even more using the coherency-noise-mean, to Log $B_{thresh} = 1.0$. See Figure 10.

The coherency test thus far proves effective at glitch rejection. The 3000 second window used for the 3000 injections (1 injection per second) included a very loud glitch at second 1626. As seen in Figure 12 on Page 23, this glitch appears before the coherency test is applied, but is strongly rejected by both tests.

As may be obvious from Figure 11 on Page 22, the coherency-mean and coherency-noise-mean tests provide very similar results, with the added noise evidence serving to further downweight the Bayes Factor at certain points, although without affecting the higher value found at the injection time. While we continue to put both tests to use, results suggest that a single test is sufficient for glitch rejection as both seem to perform equally well.

The use of a coherency test increases our confidence in picking out signals over background noise. Glitches are strongly rejected so that the loudest background events are noticeably smaller Bayes factors. Figure 8 clearly shows how coherency tests drastically improve efficiency when aiming for a low false alarm probability. By pinpointing and rejecting glitchy times, we stand a greater chance of making a true gravitational wave detection. The more rigorously we can target these glitches, the greater our range increases, allowing our analysis to be more confident at detecting sources at greater distances, as weaker signal will be more likely to stand out above a lower threshold Bayes factor.

5. Future Directions

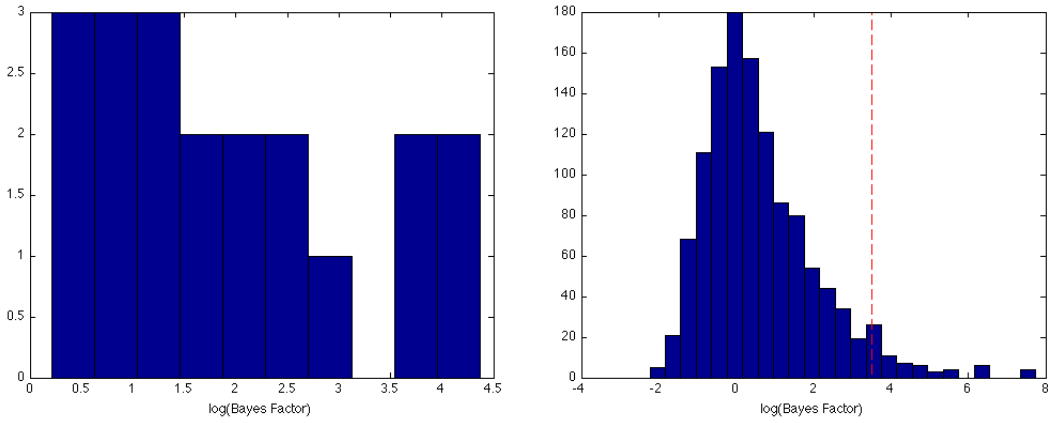
While the pipeline proves to be useful at recovering injections and matching their parameters, future direction includes more rigorous testing of the pipeline's consistency at making detections and excluding glitches, before it can reasonably be brought up for review. This includes comparing its performance with that of already existing pipelines. Also, we intend to analyze GRBs which have already or are in the process of being examined using other techniques, for further comparisons. Finally, alternative methods of glitch rejection will be contrasted with the Bayesian coherency test.

6. Concluding Remarks

Throughout the course of this summer research project, sponsored by the NSF Gravitational Wave REU Grant, significant progress was made on utilizing the nested sampling algorithm in a GRB-triggered inspiral search pipeline. Investigations made into the relationship between prior knowledge and parameter estimation shows that the tighter range we are able to isolate for short GRB opening angles, the more precisely distance can be measured. After taking into account the limitations of the nested sampling code, in amount of data which can be handled efficiently, we devised a method of combining multiple segments of data into one complete analysis in a way that is consistent with Bayesian inference. In this way, we can search across longer lengths of data without sacrificing large amounts of time to the analysis. Based on our investigations, the pipeline performs well at both finding injections and their parameters.

References

[1] D. Guetta and T. Piran (2008), [arxiv:astro-ph/0407429v2]
[2] S. Rosswog and E. Ramirez-Ruiz, (2003), [arXiv:astro-ph/0306172v1]
[3] John Skilling, "Nested Sampling for General Bayesian Computation" (2004)
[4] D. S. Sivia with J. Skilling, "Data Analysis: A Bayesian Tutorial" (Second edition, Oxford University Press 2006)
[5] J. Veitch and A. Vecchio, "Bayesian coherent analysis of in-spiral gravitational wave signals with a detector network" arxiv.org/abs/0911.3820v2
[6] J. Abadie *et al.* [LIGO Scientific Collaboration], (2010), [arXiv:1001.0165v2]



(a) Global Bayes Factors (b) All Bayes Factors

FIGURE 5. The distribution of Bayes Factors for 20 off source segments. Figure (a) shows the distribution of the 20 global Bayes Factors. Figure (b) shows the distribution of all 1200 raw Bayes Factors, calculated before the data was combined. Here the detection threshold is marked at $B_{thresh} = 3.90$. In Figure (a), the threshold is higher, between 4 and 4.5, because taking the mean favors high-valued outliers.

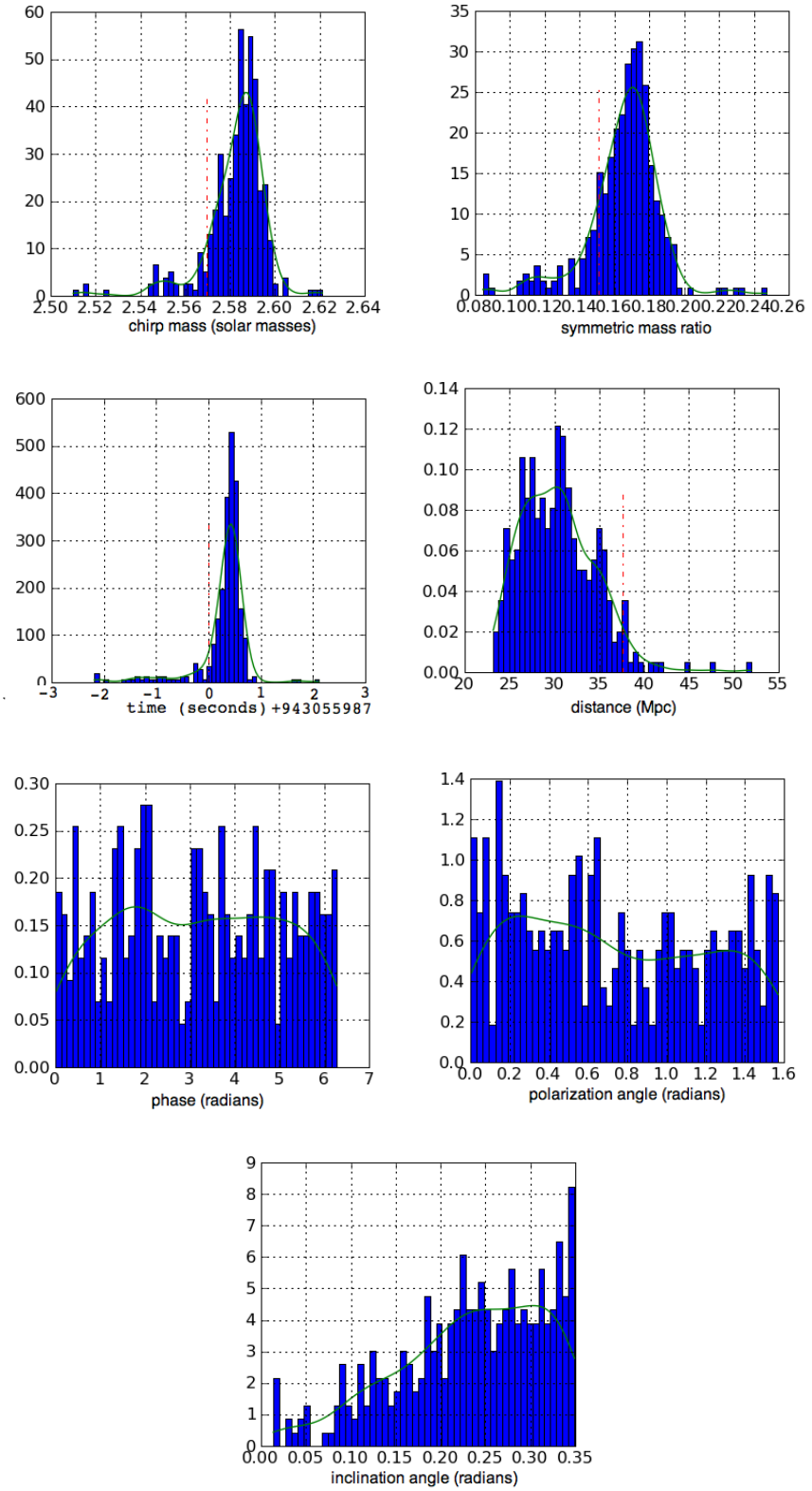


FIGURE 6. Posterior PDFs for face-on ($\iota = 0$) software injection, with SNR 7.42 and global Bayes Factor 15.90. RA and dec are not shown, as sky location was held fixed. y-axis is probability density.

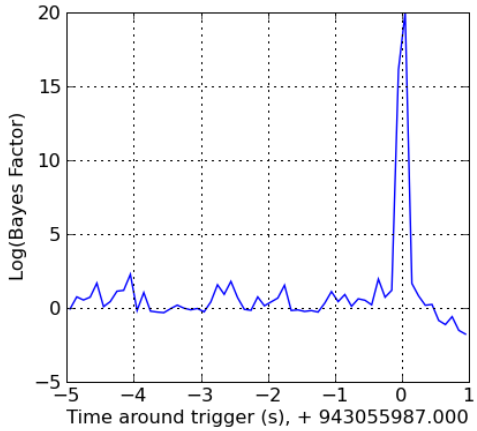


FIGURE 7. Log of the Bayes Factor over time for the software injection shown in Figure 6. There is a clear spike, high above the detection threshold, at t=0, where the injection is made. Global Bayes factor averages to be 15.90.

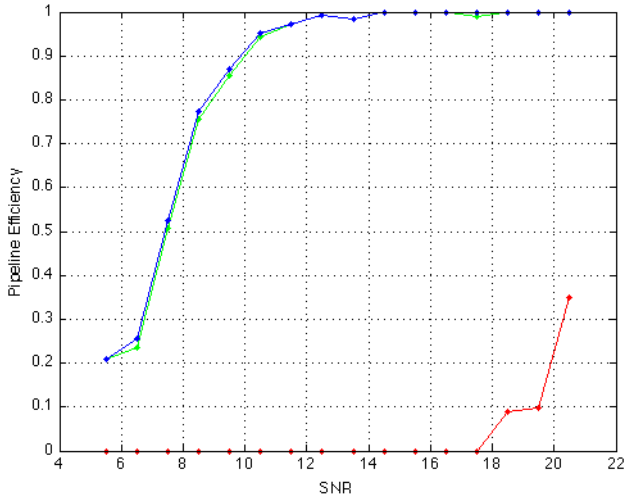


FIGURE 8. Detection efficiency plotted over SNR, for 3000 injections. Detections were calculated as percentage of injections recovered with Log(Bayes factor) greater than the threshold value, set as the loudest factor found over all 3000 times prior to making injections. The red line is the efficiency before a coherency test is performed, using threshold Bayes Factor $\mathrm{Log}B_{thresh} = 170.51$, the green line is using the coherency-mean test and $\mathrm{Log}B_{thresh} = 5.13$, and the blue line is with the coherency-noise mean and $\mathrm{Log}B_{thresh} = 4.56$.

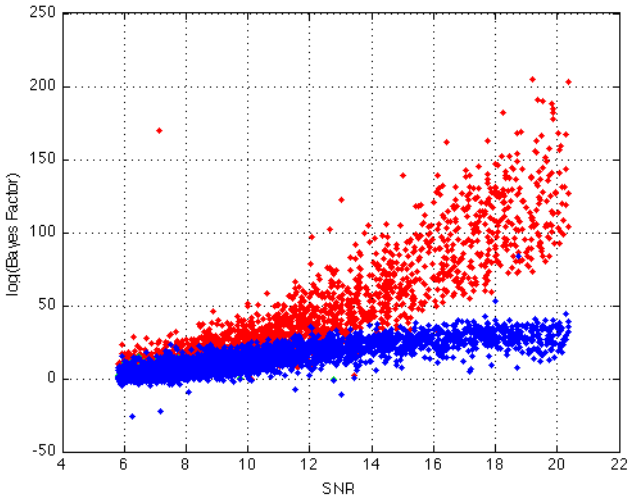


FIGURE 9. Log of the Bayes Factor plotted over SNR. Red points represent the Bayes Factors before any coherency test is applied. Blue points represent the Bayes Factors after the coherency-noise test. The coherency-mean test gives visibly the same output, and was excluded from this graph. There was one outlying point left out for better visualization, at (SNR, Log(B))= $(7.12, -852.5)$, which happens to coincide with the strongest glitch, at second 1626.

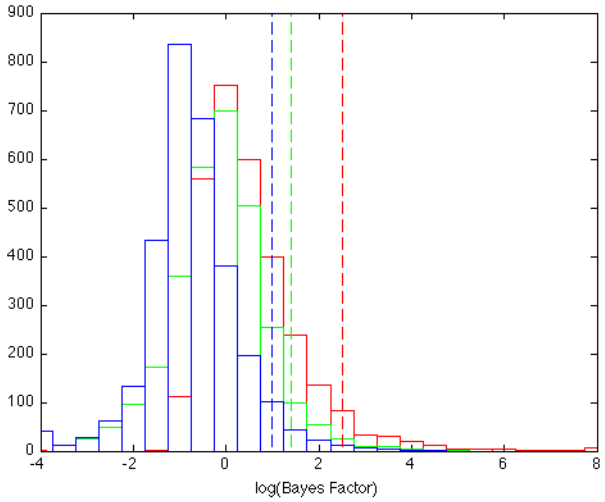


FIGURE 10. Background Estimation using Coherency Tests. Red bars are the bayes factors before a coherency test is performed, green bars are the bayes factors from the coherency-mean test, and the blue bars are from the coherency-noise mean.

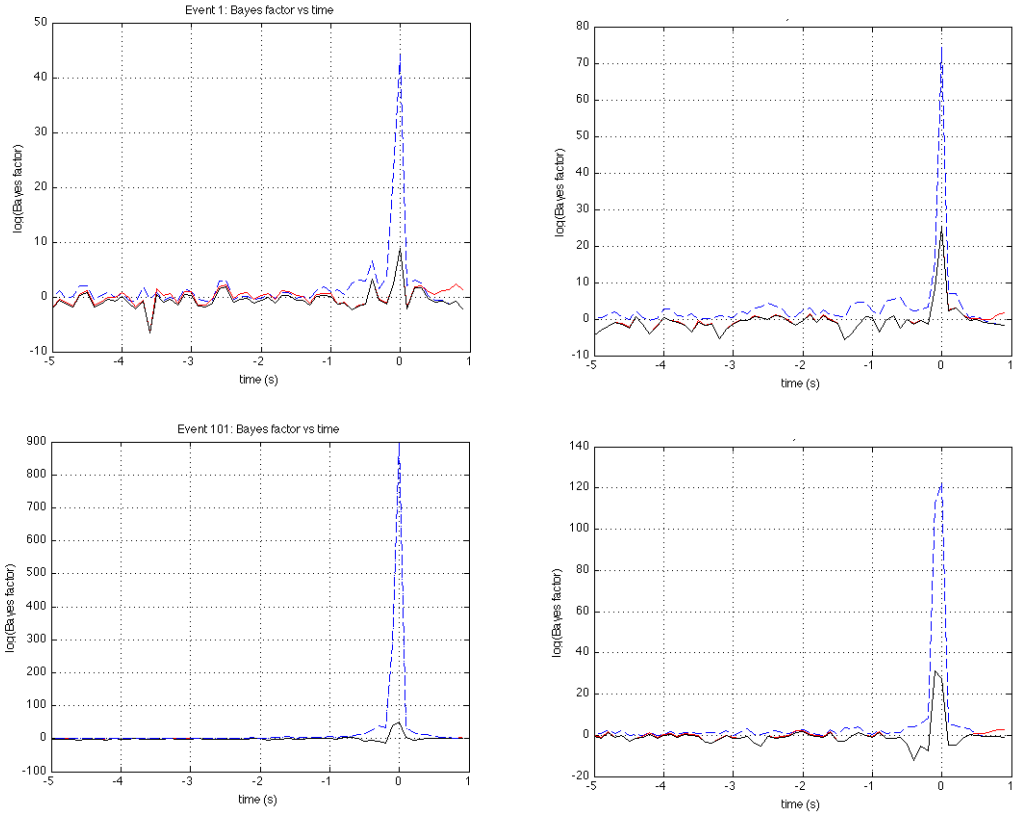


FIGURE 11. Log of the Bayes Factor plotted over time for four difference software injections. The blue dashed line is the Bayes Factor before coherency tests are applied. The red line is after the coherency mean has been applied, and the black line after the coherency-noise mean. The six second window is centered around the injection time at $t = 0$.

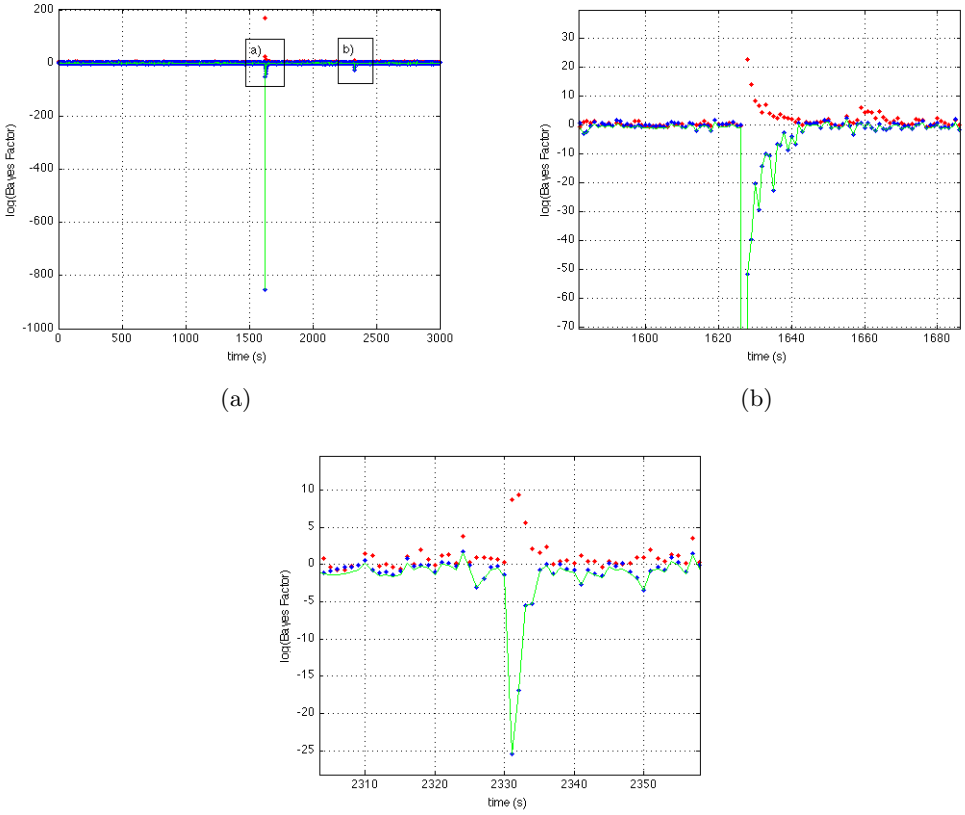


FIGURE 12. Bayes Factor plotted over 3000 off-source segments, of length 0.1 seconds each. The red dots mark the Bayes Factor before coherency test is applied, blue dots are the Bayes Factor from coherency-mean test, and the green line is from the coherency-noise mean. (a) shows the Bayes Factors for the entire 3000 second window. (b) shows a glitch rejection made around time 1630 and (c) shows a rejection around time 2330.