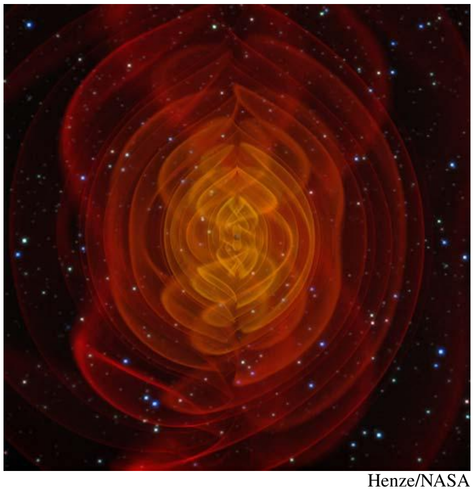
LIGO-T070181-00-Z
Stage de recherche, FIP M1

Michał Wąs

X-PIPELINE Search for Gravitational Waves Associated with GRBs
Recherche d'ondes gravitationnelles associées à des sursauts gamma à l'aide de X-PIPELINE

2007, February 15 – August 15

Advised by: Patrick Sutton, *California Institute of Technology & Cardiff University*
Alan Weinstein, *California Institute of Technology*



Henze/NASA

Abstract

Coherent data analysis techniques have recently been introduced to efforts to detect gravitational-wave (GW) bursts. One coherent data analysis package, X-PIPELINE, has been developed up to the point where it can search in real detector data for GWs associated with gamma-ray bursts (GRBs). In this report we use X-PIPELINE to test and compare two alternative methods for detecting GWs while coping with non-Gaussian background noise glitches. Both techniques show significantly improved sensitivity over the current LIGO-standard algorithm for searching for GWs associated with GRBs.

Résumé

Les algorithmes de recherche de sursauts d'ondes gravitationnelles ont récemment commencé à utiliser des méthodes cohérente d'analyse de données. Un des outils d'analyse de données, X-PIPELINE, à atteint la capacité de chercher dans des données réels des ondes gravitationnelles associées à des sursauts gamma. Ce rapport étudie et compare deux techniques de détection d'ondes gravitationnelles dans un bruit non Gaussien. Chacune de ces techniques a une sensibilité significativement supérieure, à celle de l'analyse standard utilisé à présent par LIGO, pour chercher des ondes gravitationnelles associées à des sursauts gamma.

Acknowledgments

I want to thank Alan Weinstein for this wonderful oportunity in participating in real scientific research. Great thanks for Patrick Sutton who guided me through this project, and was always generous with his time. I need also to thank Irene Baldon and Cindy Akutagawa for smoothing out all administartive difficulties, and the LIGO Scientific Collaboration and the Virgo Collaboration for the use of their data and computational facilities.

Disclaimer

The work presented in this report has not been reviewed by the LIGO Scientific Collaboration of the Virgo Collaboration, and therefore should not be taken to represent the scientific opinion of either organization. This report has been assigned LIGO Document Number LIGO-T070181-00-Z.

Contents

1 Introduction

Gravitational waves (GWs) are one of the early predictions of Einstein's theory of general relativity. Though predicted since 1916, and indirectly confirmed by observation of energy radiation [8], GWs (gravitational waves) have never been observed directly. However current interferometric gravitational waves detectors, like LIGO [16] and Virgo [17], are on the verge of being able to observe directly GWs emitted by astrophysical sources. There are four major families of GWs that could be detected with these instruments: bursts, continuous waves, inspirals and a stochastic GW background. Bursts are short signals (expected typically to last less than a second) that do not a have a good theoretical model. Continuous waves are long lived (days or months long) quasi-sinusoidal signals, inspirals are signals from coalescing compact binaries (neutron stars, black holes), and stochastic waves are a general background much like the cosmic microwave background.

In this work we will be concerned bwith analyzing data from detectors in order to detect bursts of GWs, especially in the case when a gamma ray burst can be associated with it. The main difficulty of this analysis is that burst signals are very similar to detector noise "glitches". A wide array of analysis methods called *pipelines* are at different stage of development in the burst group of the LIGO Scientific Collaboration and the Virgo collaboration. The topic of this internship is to complete missing parts of one of these pipelines, called X-PIPELINE, so that it can perform an analysis of currently available real data, and then to test how efficient the pipeline is in detecting GWs. This work has been done in close collaboration with and under the supervision of Patrick Sutton, the main developer of X-PIPELINE.

1.1 Gravitational Waves

Einstein's theory of general relativity describes gravity as a tensor field evolving under a set of non linear equations, this tensor field gives the definition of the space-time metric, the distance between things, through

$$\mathrm{d}s^2 = g_{\mu\nu}\mathrm{d}x^\mu\mathrm{d}x^\nu, \tag{1}$$

where x^0 is the "time" coordinate and $x^1,\ldots,x^3$ are the "space" coordinates. The full Einstein equations with the presence of sources can't be solved analytically in most cases, which makes computation of what the gravity field looks like for cataclysmic events like black hole mergers or supernova core collapse very difficult. However in the limit where the space time metric is nearly flat (Minkowski metric), Einstein's equations can be linearized, yielding wave propagation equations much like those for the electromagnetic field. More precisely, if we write the metric as

$$g_{\mu\nu} = \eta_{\mu\nu} + h_{\mu\nu}, \tag{2}$$

with

$$\eta_{\mu\nu} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{3}$$

then in the weak field limit, $|h_{\mu\nu}| \ll 1$, Einstein's equations in vacuum in the transverse-traceless gauge are [15]:

$$\partial^\sigma \partial_\sigma h_{\mu\nu} = 0, \tag{4}$$

with the gauge conditions,

$$\partial^\mu h_{\mu\nu} = 0, \tag{5a}$$

$$h^\mu{}_\mu = 0, \tag{5b}$$

$$h_{0\mu} = 0. \tag{5c}$$

We expect to observe planar waves, because the wavelengths to which ground-based detectors are sensitive ($\sim 1 \cdot 10^2 - 1 \cdot 10^4$ km) are much shorter than the distance to astrophysical sources. For a planar GW we can write

$$h_{\mu\nu} = \mathcal{R}e \left\{ H_{\mu\nu} e^{ik_\alpha x^\alpha} \right\}. \tag{6}$$

Einstein's equations become

$$k^\mu k_\mu = 0, \tag{7}$$

this equation means that GWs propagate at the speed of light. And the gauge conditions are

$$k^\mu H_{\mu\nu} = 0 \tag{8a}$$

$$H^\mu{}_\mu = 0, \tag{8b}$$

$$H_{0\mu} = 0. \tag{8c}$$

Only eight of the nine equations defining the gauge conditions are independent[1], and $h_{\mu\nu}$, being a symmetric 2-tensor, has ten degrees of freedom, so the space of

[1] equation (8c) implies equation (8a) for $\nu = 0$

solutions is two dimensional for a fixed k_μ. Those two dimensions are the two polarizations of a planar GW with a given direction of propagation (k^1, k^2, k^3) and frequency k^0. The two polarizations are commonly called the *plus polarization* and *cross polarization*. Let us choose the propagation along the z-axis, *ie* $k_\mu = (\omega 0 0 \omega)$, then solutions have the following form

$$H_{\mu\nu} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & H_+ & H_\times & 0 \\ 0 & H_\times & -H_+ & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{9}$$

Thus, while sources of GWs are often hard to model (and that is where interesting physics happens), the propagation of GWs from the sources to detectors is well understood.

1.2 Detection with an Interferometer

There are two main detection schemes for observing GWs : resonant bars and interferometers. Resonant bar detectors were originally conceived by Weber at the end of the 50's, and the idea of interferometers was introduced by several independent groups at the beginning of the 60's [10]. The current most sensitive designs use interferometers.

Interferometric GW detectors are Michelson interferometers with arms that are a few kilometers long or shorter, often with resonant Fabry-Perrot cavities as arms (or other techniques to increase the optical path), for more details see figure 1. Although the actual detectors are quite complicated, the principle of the measurement can be easily understood in the case of a simple Michelson interferometer.

A Michelson interferometer has two orthogonal arms of equal length. When a GW passes through the interferometer the metric is changed, and arms are no longer of equal length. This difference in length changes the light travel time in each arm, so that light beams that recombine at the beam splitter no longer have the same phase, and this changes the light power output of the interferometer.

More precisely for a metric $g_{\mu\nu}$ the light travel time is given by solving

$$\mathrm{d}s^2 = 0 = g_{\mu\nu}\mathrm{d}x^\mu \mathrm{d}x^\nu. \tag{10}$$

$g_{\mu\nu}$ can be assumed uniform in space-time because the GW wavelength is much larger than the detector arm. For the simple case where the plane defined by

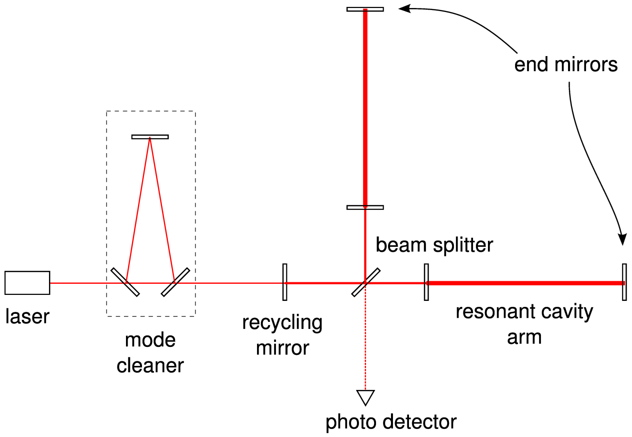


Figure 1: Simplified optical scheme of a LIGO interferometer. Light created by a laser goes through a mode cleaner that ensures a Gaussian light beam, then it is split in two by a 50/50 beam splitter. After bouncing on the order of a hundred times back and forth in each resonant cavity light is recombined by the beam splitter, so that most of the light goes back to the laser. The recycling mirror reuses this light to effectively strengthen input laser light. A passing GW changes the relative arm lengths, changing the intreference pattern at the beam splitter and causing light to pass to the photo detector.

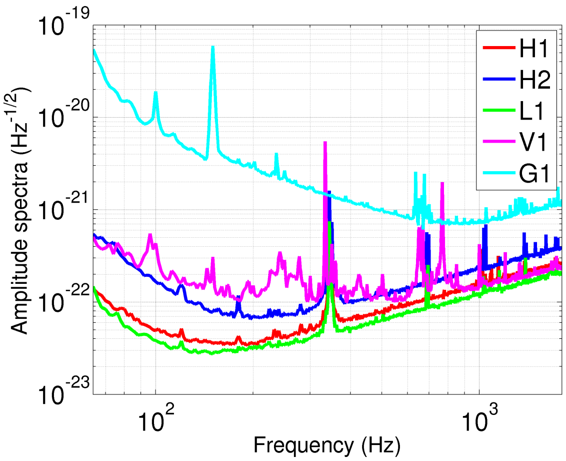


Figure 2: Amplitude noise spectra of major interferometers currently online, i.e., the amplitude of the Fourier transform of $n(t)$ (18). The different detectors are (the numbers are the arms length of the interferometer): H1 — LIGO 4 km at Hanford, H2 — LIGO 2 km at Hanford, L1 — LIGO 4 km at Livingston, V1 — Virgo 3 km near Pisa, G1 — GEO 600 m near Hannover.

the detector arms is orthogonal to the direction of propagation of the GW, we can define axes x and y along the interferometers arms, and the z axis along the direction of propagation. In this case the metric is

$$g_{\mu\nu} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1+H_+ & H_\times & 0 \\ 0 & H_\times & 1-H_+ & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{11}$$

We can rewrite (10) as (with the usual convention $c = 1$)

$$0 = -\mathrm{d}t^2 + (1+H_+)\mathrm{d}x^2 \quad \text{for the x arm,} \tag{12}$$

$$0 = -\mathrm{d}t^2 + (1-H_+)\mathrm{d}y^2 \quad \text{for the y arm.} \tag{13}$$

Integrating over the arm length L along the x-axis we get the light travel time along the x arm

$$\tau_x = 2\int_0^L \sqrt{1+H_+}\mathrm{d}x = 2L\sqrt{1+H_+} \simeq 2L(1+\frac{1}{2}H_+), \tag{14}$$

where the 2 accounts for the fact that the light has to go back and forth along the arm. Therefore the time travel difference is

$$\Delta\tau = \tau_x - \tau_y = 2LH_+, \tag{15}$$

which leads to a phase shift for light with wavelength λ

$$\Delta\varphi = 2LH_+\frac{2\pi}{\lambda}, \tag{16}$$

this is what is observed with the interferometer.

In the general case where the coordinates axes, the arm directions and the direction of propagation of the GW have no particular relationship, expression (16) gets complicated by a few rotation matrices, and the phase shift has the form

$$\Delta\varphi = \frac{4\pi L}{\lambda}F_+(\theta,\phi)H_+ + \frac{4\pi L}{\lambda}F_\times(\theta,\phi)H_\times. \tag{17}$$

Here θ, ϕ are the angles defining the direction of propagation of the GW, and F_+, $F_\times$ are the *detector responses* folding in the rotation matrices. It should be noted that a single interferometer is sensitive only to one GW polarization, because equation $0 = 2LF_+(\theta,\phi)H_+ + 2LF_\times(\theta,\phi)H_\times$ define a one dimensional hyperplane in the two dimensional space of polarizations $(H_+,H_\times)$.

This phase shift is recorded by a photo detector (the output light power scales as $P = P_0 \cos(\Delta\varphi)^2$), and after calibration the time series that gets recorded is

$$d(t) = F_+(\theta,\phi)h_+(t) + F_\times(\theta,\phi)h_\times + n(t), \tag{18}$$

with $n(t)$ the detector noise. This time series is dimensionless and is called *detector strain* or *"h of t"*. This seemingly simple scheme (which is very hard to implement in practice) leads to incredibly sensitive measures of distance (see Fig 2), $\Delta L \sim d(t)L \sim 10^{-18}\,\mathrm{m}$, a few orders of magnitude less than a proton radius! The main properties of this detector noise are : a colored, approximately gaussian noise, that is lowest at frequencies around $100-1000\,\mathrm{Hz}$; fine spectral lines, the main lines for LIGO detectors are near $350\,\mathrm{Hz}$, from the violin resonant mode of mirror suspension; and short non-Gaussian transients colloquially called "glitches". This noise changes from day to day due to environmental conditions (e.g. wind), technical problems, ... This complicated noise makes data analysis challenging and very interesting.

2 Methods

X-PIPELINE [12] is a complete data analysis pipeline for interferometric gravitational wave detection. It is able to read calibrated interferometer strain data, and output lists of candidate detection events and/or upper limits on unseen gravitational waves. Its goal is to detect gravitational wave *bursts*, short lived transients that do not have a good waveform model. This kind of transient can be produced by many cataclysmic astrophysical events: supernova core-collapse [3], black hole mergers [2], ... To accomplish this, strain data from a network of detectors are combined coherently taking advantage of the fact that the same gravitational waveform is seen in all detectors, with short time delays due to GW propagation at the speed of light. Coherent techniques for the detection of GW bursts are relatively new and under active development in the LIGO Scientific Collaboration. Here we will focus on a particular application of this package, the case where we look for GWs associated with a GRB (gamma ray burst) observed by an electromagnetic detector like the Swift satellite [5]. This electromagnetic *trigger* gives a source sky position and an event time.

2.1 Coherent detection statistic in Gaussian noise

Coherent detection methods (as opposed to incoherent methods) combine data from several detectors before processing to detect signals. In incoherent methods candidate events are constructed from each detector data strain independently, and one looks for events with similar duration, frequency band and amplitude that occur in all detectors simultaneously. Coherent methods combine detector strains directly to create a single list of candidate events for the whole network. So *a priori* a coherent method has the possibility to extract more information from the detector network than an incoherent method can.

Burst signals do not have a precisely known theoretical waveform, but various models predict signal with duration between a few milliseconds and a few seconds and with important contribution in the frequency range $100-1000\,\mathrm{Hz}$ where current detectors are most sensitive. Without a precise model, optimal filtering or template-based techniques cannot be used; however, the power of having a network of detectors (LIGO, Virgo, GEO) seeing the exact same waveform can be leveraged. We are going to assume in this section that the noise in each detector is Gaussian and independent. Equation 18 can be rewritten for the case of a network of N detectors as

$$\begin{bmatrix} d_1(t) \\ \vdots \\ d_N(t) \end{bmatrix} = \begin{bmatrix} F_1^+(\Omega)h_+(t-t_1(\Omega)) \\ \vdots \\ F_N^+(\Omega)h_+(t-t_N(\Omega)) \end{bmatrix} + \begin{bmatrix} F_1^\times h_\times(t-t_1(\Omega)) \\ \vdots \\ F_N^\times h_\times(t-t_N(\Omega)) \end{bmatrix} + \begin{bmatrix} n_1(t) \\ \vdots \\ n_N(t) \end{bmatrix}, \quad (19)$$

with Ω the source sky location (θ, ϕ), and $t_\alpha(\Omega)$ the time of arrival of the GW at detector α. Using the GRB sky position information the detector strains can be time shifted $d_\alpha(t+t_\alpha(\Omega)) \longrightarrow d'_\alpha(t)$, so that the previous equation reads

$$\begin{bmatrix} d'_1(t) \\ \vdots \\ d'_N(t) \end{bmatrix} = \begin{bmatrix} F_1^+(\Omega) \\ \vdots \\ F_N^+(\Omega) \end{bmatrix} h_+(t) + \begin{bmatrix} F_1^\times \\ \vdots \\ F_N^\times \end{bmatrix} h_\times(t) + \begin{bmatrix} n'_1(t) \\ \vdots \\ n'_N(t) \end{bmatrix}. \quad (20)$$

We know that GW signals are band limited[2], so looking at the Fourier transform of this equation helps in removing those frequencies that we are not interested in. And so that each frequency has the same statistical proprieties we whiten the data (divide by the detector amplitude spectra $\sigma_\alpha(f)$, *cf* Fig 2). This gives the following equation

$$\begin{bmatrix} \tilde{d'}^w_1(f) \\ \vdots \\ \tilde{d'}^w_N(f) \end{bmatrix} = \begin{bmatrix} F_1^+/\sigma_1(f) \\ \vdots \\ F_N^+/\sigma_N(f) \end{bmatrix} \tilde{h}_+(f) + \begin{bmatrix} F_1^\times/\sigma_1(f) \\ \vdots \\ F_N^\times/\sigma_N(f) \end{bmatrix} \tilde{h}_\times(f) + \begin{bmatrix} \tilde{n'}^w_1(f) \\ \vdots \\ \tilde{n'}^w_N(f) \end{bmatrix}, \quad (21)$$

or in a more compact form, dropping all sub/super-scripts

$$\mathbf{d} = \mathbf{F}^+ h_+ + \mathbf{F}^\times h_\times + \mathbf{n}. \quad (22)$$

In (22) we see that in each frequency bin, $\mathbf{d}$ is a N dimensional vector that is a sum of two parts: a noise vector for which each component has a standard normal distribution, and two detector response vectors whose directions do not depend on the incoming GW waveform (only the lengths do).

[2] Even a very broad-band GW signal is effectively badn-limited by the detector response; see Fig 2

[3] The network sensitivity to a linearly polarized waveform is the length of the vector $\mathbf{F}_+ \cos 2\psi + \mathbf{F}_\times \sin 2\psi$. In the 2-dimensional subspace spanned by detector responses this vector family can be written $\begin{bmatrix} a \\ b \end{bmatrix} \cos 2\psi + \begin{bmatrix} c \\ d \end{bmatrix} \sin 2\psi$, which is a parametric equation of an ellipse. So the sensitivity draws an ellipse on the plane.

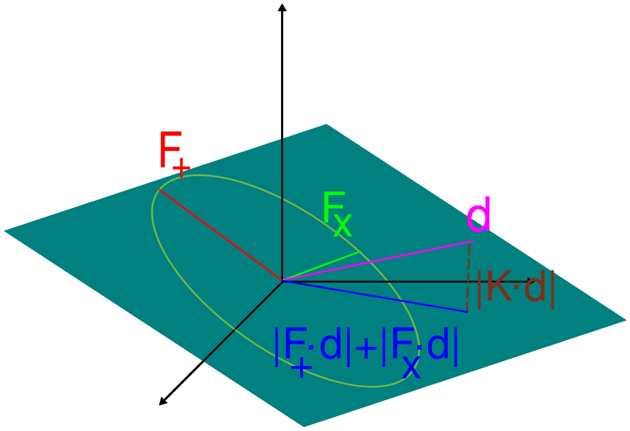


Figure 3: 3 dimensional space of detector strains (3 detector case). The green plane is the plane spanned by the detector response vectors. The yellow ellipse is the sensitivity to linearly polarized waves with different polarizations (see Fig 4). The red and green lines are the sensitivities to the plus and cross polarization in the DP frame. The magenta line is the vector of detector strains for one noise realization. The blue is the projection on the detector response plane. The burgundy dashed line is the projection on the null space.

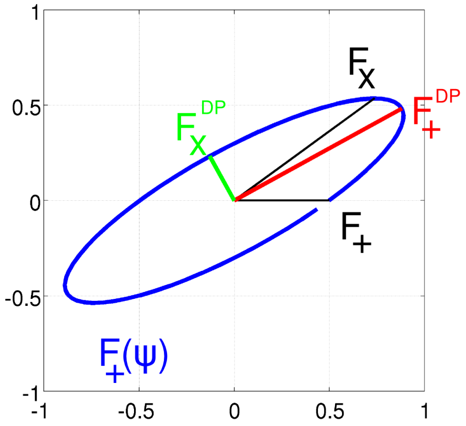


Figure 4: The 2-dimensional subspace of the detector strain space defined by $\mathbf{F}_+, \mathbf{F}_\times$. The blue ellipse gives the network sensitivity for linearly polarized GWs of polarization ψ^3. The black $\mathbf{F}_+, \mathbf{F}_\times$ are sensitivities for both polarization for a randomly defined plus polarization. The colored $\mathbf{F}_+, \mathbf{F}_\times$ are sensitivities for the plus polarization in the *dominant polarization frame*; they form the major and minor axes of the ellipse.

2.1.1 Dominant polarization frame

We have a N dimensional abstract space of detector strains defined by (22). One basis of this space is formed by the set of single detector strains (the basis in which all equations have been written so far); however, this is not the most convenient basis for writing detection statistics. The 2 dimensional subspace defined by $\mathbf{F}_+, \mathbf{F}_\times$ is a natural starting point for the construction of a better basis. If we look more precisely at this 2 dimensional space (see Fig 4), there is a direction in which the detector network is most sensitive and a direction in which the network is least sensitive. Choosing those two orthogonal directions and completing them to have a orthonormal basis is a very convenient basis choice. To further simplify things it is possible to define the $+$, $\times$ polarization (choose the xyz directions) so that $\mathbf{F}_+$ lies along the first basis vector, and $\mathbf{F}_\times$ along the second. This choice of polarization definition is called the *dominant polarization frame* [7]. In this basis the equation for detector strains is

$$\begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{bmatrix} = \begin{bmatrix} F^+_{DP} \\ 0 \\ \vdots \\ 0 \end{bmatrix} h_+ + \begin{bmatrix} 0 \\ F^\times_{DP} \\ 0 \\ \vdots \end{bmatrix} h_\times + \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_N \end{bmatrix}, \tag{23}$$

with m_i a different set of random variables with standard normal distributions, and the subscript DP denoting the dominant polarization frame. We have switched the subscripts from α to i to emphasize that we are writing $\mathbf{d}$ in a different basis.

2.1.2 Standard likelihood and hard constraint likelihood

Having expressed detector strain in a convenient basis we can use it to detect GW. When analyzing those strains we want to separate between to possible hypothesis: strains are a sum of GW signal with noise, or strains contain only detector noise. From the frequentist point of view the optimal choice for differentiating two hypotheses is to look at their *likelihood ratio*

$$L = \frac{P(\mathbf{d}|\text{signal in noise})}{P(\mathbf{d}|\text{noise})}. \tag{24}$$

It is the ratio of obtaining the detector strains that we see given each hypothesis. The optimality is given by Neyman-Pearson lemma [11]. This lemma states that for deciding between two hypotheses, comparing the likelihood to a threshold gives the minimal rate of false negatives (missed signals) for a fixed rate of false positives (noise tagged as signal), where the false positive rate is fixed by the choice of threshold.

We can use (23) and the assumption of Gaussian noise to compute this likelihood ratio. First the denominator is

$$P(\mathbf{d}|\text{noise}) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}} \exp(-\tfrac{1}{2}|d_i|^2) = \frac{1}{(2\pi)^{N/2}} \exp(-\tfrac{1}{2}\mathbf{d}^\dagger \mathbf{d}), \tag{25}$$

because we have already whitened the data. Getting the numerator is a little harder because we need to marginalize over all the possible waveforms $(h_+, h_\times)$. If we knew the exact waveform then

$$\begin{aligned} P(\mathbf{d}|h_+, h_\times, \text{signal in noise}) = {} & \frac{1}{\sqrt{2\pi}} \exp(-\tfrac{1}{2}|d_1 - F^+_{DP} h_+|^2) \\ & \frac{1}{\sqrt{2\pi}} \exp(-\tfrac{1}{2}|d_2 - F^\times_{DP} h_\times|^2) \prod_{i=3}^{N} \frac{1}{\sqrt{2\pi}} \exp(-\tfrac{1}{2}|d_i|^2). \end{aligned} \tag{26}$$

Without this knowledge we marginalize this expression assuming that all values of h_+, $h_\times$ are equally probable

$$P(\mathbf{d}|\text{signal in noise}) = \int\int \mathrm{d}h_+ \mathrm{d}h_\times P(\mathbf{d}|h_+, h_\times, \text{signal in noise}) \tag{27}$$

$$= \frac{1}{F^+_{DP}} \frac{1}{F^\times_{DP}} \prod_{i=3}^{N} \frac{1}{\sqrt{2\pi}} \exp(-\tfrac{1}{2}|d_i|^2). \tag{28}$$

Thus the likelihood ratio is

$$L(\mathbf{d}) = 2\pi \frac{1}{F^+_{DP}} \frac{1}{F^\times_{DP}} \exp \tfrac{1}{2}(|d_1|^2 + |d_2|^2). \tag{29}$$

We can then take the log of this likelihood ratio; this new quantity is often called *log-likelihood ratio*, and the main advantages are that it spans fewer orders of magnitude. But more important it is additive, $\log(L(x,y)) = \log(L(x)) + \log(L(y))$, where x, y are detector strains at two independent times/frequencies. The additivity is important for easy clustering, see Sec 2.5 . Dropping the terms that do not depend on detector strains we obtain

$$\mathrm{SL} \triangleq 2 \log L = |d_1|^2 + |d_2|^2 = |\hat{\mathbf{F}}^+_{DP} \cdot \mathbf{d}|^2 + |\hat{\mathbf{F}}^\times_{DP} \cdot \mathbf{d}|^2, \tag{30}$$

where $\hat{\mathbf{F}}^{+}_{DP}$, $\hat{\mathbf{F}}^{\times}_{DP}$ are unit vectors pointing in the directions of $\mathbf{F}^{+}_{DP}$, $\mathbf{F}^{\times}_{DP}$. The quantity SL is called the *standard likelihood*. It is the magnitude of the projection of the detector data $\mathbf{d}$ on the 2 dimensional subspace spanned by $\mathbf{F}^{+}$, $\mathbf{F}^{\times}$. If we look more precisely

$$\mathrm{SL} = |d_1|^2 + |d_2|^2 = |F^{+}_{DP} h_{+} + m_1|^2 + |F^{\times}_{DP} h_{\times} + m_2|^2. \tag{31}$$

In most network configurations $F^{\times}_{DP}/F^{+}_{DP}$ is small, while by design the noise contributions m_1 and m_2 have the same variance. So, the second term of the standard likelihood adds the same amount of noise but much less signal than the first. So Klimenko et al. proposed a much less noisy statistic

$$\mathrm{HC} = |d_1|^2, \tag{32}$$

and called it the *hard constraint likelihood*.

2.2 Coherent glitch veto

Real detectors do not have Gaussian noise. Detector noise has *glitches*, which are short transients of excess strain that look much like burst signals. To get rid of those false signals one option is to apply a coherent veto test [6].

The coherent veto test is based on the orthogonal complementary subspace to the one used for defining the standard likelihood; this subspace dimension is $N-2$ and is called the *null space*. The projection of $\mathbf{d}$ on this subspace contains only noise, and the presence or absence of GWs do not affect this projection in any way. But detector glitches will generally be present in this projection. The magnitude of this projection is called *null energy* (the square of the length of the burgundy line on figure 3). In the coherent veto test, the null energy needs to be small in order for a high detection statistic signal to be accepted as a GW and not as a glitch. Formally we can write

$$E_{null} = \sum_{i=3}^{N} |d_i|^2 = \sum_{\alpha}\sum_{\beta} K_{\alpha\beta} d^{*}_{\alpha} d_{\beta}, \tag{33}$$

where d_i is the strain in our new basis, d_α is the strain in the detector α (i.e., in the original basis), and $K_{\alpha\beta}$ is the appropriate projection operator.

We need to compare E_{null} to something if we want to say that it is small. For noise glitches, the strains for each detector are expected to be uncorrelated so that the expected value of the null energy does not contain cross-correlation terms,

$$E_{null} \simeq \sum_{\alpha} K_{\alpha\alpha} |d_\alpha|^2 = E_{inc}. \tag{34}$$

This autocorrelation part of the null energy is called *incoherent energy*. However for GW, strains in each detector are correlated, and by construction these correlations cancel themselves in the null energy, leaving only Gaussian noise, but they are not canceled in E_{inc}. The idea behind the veto test is to say that for $a \simeq b$, $|a-b|^2 \ll |a|^2+|b|^2$.

Thus for noise glitches we have $E_{null} \simeq E_{inc}$, and for GW we have $E_{null} < E_{inc}$, so the coherent veto test is to reject any event with

$$E_{inc}/E_{null} < C, \tag{35}$$

where C is a constant greater than 1 that needs to be tuned.

2.3 Glitch standard likelihood

Another option to cope with glitches is to include them in the derivation of the detection statistic. We want to distinguish between two hypothesis: signal in gaussian noise, and glitches in gaussian noise. There is a third possible hypothesis, gaussian noise alone, but it is between the first two hypotheses that all the difficult separation is. For that we use again the likelihood ratio (which is the optimal choice for choosing between two hypotheses) :

$$L = \frac{P(\mathbf{d}|\text{signal hypothesis})}{P(\mathbf{d}|\text{glitch hypothesis})} \tag{36}$$

We can assume that noise in each detector is independent, so

$$P(\mathbf{d}|\text{glitch hypothesis}) = \prod_\alpha P(d_\alpha|\text{glitch hypothesis}). \tag{37}$$

The main properties of glitches are that they have strains much larger that what we would expect from a Gaussian distribution, so a good model for Gaussian noise with glitches can be written as

$$\begin{aligned} P(x|\text{glitch hypothesis}) &= \frac{1-\varepsilon}{\sqrt{2\pi}}\exp(-\frac{x^2}{2}) + \varepsilon p(x), && (38)\\ &= \frac{1-\varepsilon}{\sqrt{2\pi}}\exp(-\frac{x^2}{2})\left(1+\frac{\varepsilon}{1-\varepsilon}\sqrt{2\pi}p(x)\exp\frac{x^2}{2}\right). && (39)\end{aligned}$$

Here $p(x)$ is a long tailed probability distribution, *ie* a probability distribution with $\int x^2 p(x) \gg 1$. Under those condition $\exp(\frac{x^2}{2})$ is growing much more quickly than

$p(x)$ is decaying. We can approximate $p(x) \simeq A$ by a constant, so that

$$P(x|\text{glitch hypothesis}) = \frac{1-\varepsilon}{\sqrt{2\pi}} \exp(-\frac{x^2}{2}) \left(1 + \underbrace{\frac{\varepsilon}{1-\varepsilon}\sqrt{2\pi}A}_{\triangleq R} \exp\frac{x^2}{2} \right). \quad (40)$$

The total propability distribution is

$$P(\mathbf{d}|\text{glitch hypothesis}) = \prod_\alpha \frac{1-\varepsilon_\alpha}{\sqrt{2\pi}} \exp(-\frac{|d_\alpha|^2}{2}) \left(1 + R_\alpha \exp\frac{|d_\alpha|^2}{2}\right). \quad (41)$$

The likelihood ratio can be re-written as

$$L(\mathbf{d}) = \underbrace{\frac{P(\mathbf{d}|\text{signal hypothesis})}{\prod_\alpha \frac{1}{\sqrt{2\pi}} \exp(-\frac{|d_\alpha|^2}{2})}}_{\text{standard likelihood term}} \frac{1}{\prod_\alpha \left(1 + R_\alpha \exp\frac{|d_\alpha|^2}{2}\right)} \frac{1}{\prod_\alpha (1-\varepsilon_\alpha)}. \quad (42)$$

We can then take the log of this likelihood :

$$\log L(\mathbf{d}) = \tfrac{1}{2}\mathrm{SL} - \sum_\alpha \log\left(1 + R_\alpha \exp\frac{|d_\alpha|^2}{2}\right) - \sum_\alpha \log(1-\varepsilon_\alpha). \quad (43)$$

We drop the last term since it does not depend on the data. Thus, the final detection statistic is

$$\mathrm{gSL} \triangleq \log L(\mathbf{d}) = \tfrac{1}{2}\mathrm{SL} - \sum_\alpha \log\left(1 + R_\alpha \exp\frac{|d_\alpha|^2}{2}\right), \quad (44)$$

which reduces to

$$\mathrm{gSL} \simeq \tfrac{1}{2}\mathrm{SL} - \log\left(1 + R \sum_\alpha \exp\frac{|d_\alpha|^2}{2}\right), \quad (45)$$

in the simplifying case of $R_\alpha = R \ll 1$.

The exponential is a rapidly growing function, so the second term in the glitch standard likelihood (see (45)) is approximately equal to $\max_\alpha |d_\alpha|^2$. It can be interpreted that the glitch standard likelihood removes from the network the detector for which the strain is the largest, and looks only at the rest of the detector network. The reason behind it is that a large strain in one detector may be simply due to a single detector glitch.

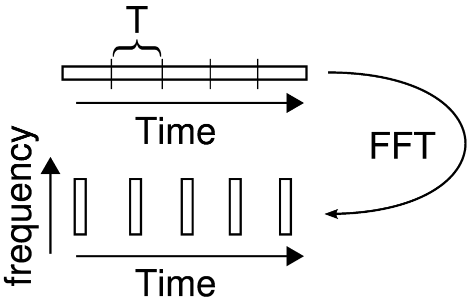


Figure 5: A time series transformed into a time-frequency map using FFTs of length T.

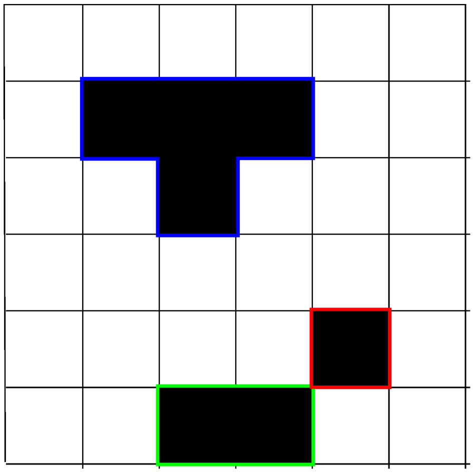


Figure 6: A time frequency map with black pixels clustered into 3 clusters.

2.4 Time-frequency map

Until now we have only considered likelihoods in one frequency bin, but detector data is a discrete time series. A common approach in data analysis, and the one used by X-PIPELINE is to make a *time-frequency map* or *spectrogram* of this time series, and compute the different statistics for each time-frequency bin (also called time frequency pixel). Making a time-frequency map is simple. First, cut the time series into chunks of length T, called the *analysis* or *integration time*. FFT (fast Fourier transform) each chunk separately (see Fig 5). Those chunks can be chosen to be consecutive, as in the figure, or overlapping so that a GW signal is not split between two time-frequency bins. This representation give us easy access to the time evolution of the spectral proprieties of the signal, and all statistics and other quantities that are functions of time and frequency.

2.5 Clustering

We have now a *TF map* (time-frequency map) of a detection statistic which *a priori* allows us to say for each pixel how likely it is to contain a GW signal. But this is not the question which we are interested in. What we would like to say is how likely it is that a stretch of several minutes of data around a given GRB contains a GW signal. This requires collecting statistics from the entire TF map into one quantity.

The solution to this problem used by X-PIPELINE is pixel clustering [13]. This approach consists in looking at a fixed percentage (e.g., 1%) of pixels with the highest value of the detection statistic (the 1% *loudest* or *brightest* pixels), and mark those pixels as *black pixels*. Then black pixels, that share a common side are grouped together into clusters (see Fig 6). Each cluster inherits the detection statistic from its constituent pixels by simply summing the values of the statistic in the pixels. This is where the additive proprieties of the log-likelihood ratio are important — the inherited detection statistic is exactly the detection statistic for the area defined by the cluster. Each cluster is a candidate detection event, and the loudest cluster on the TF map is the best candidate GW event for a given GRB.

This clustering technique relies on the assumption that GW signals are not completely random but have a continuous evolution in the time-frequency map. This is true for most burst signals which are roughly rectangular “blobs”, or for well modeled signals like inspirals or ringdowns which are continuous lines. In this method no particular shape in TF plane is assumed. Some of the other time-frequency methods [1] sum the detection statistic over a fixed set of rectangles of different size and aspect ratio, thus assuming that GW signals will fit well into these rectangles, which is not true for all known signals (e.g., inspirals).

One other important feature of the time-frequency maps is the analysis time T, which determines the aspect ratio of the pixels. A longer time gives pixels with poor time resolution but good frequency resolution, a shorter time gives pixels with good time resolution but poor frequency resolution. Depending on the signal duration, which span 2 orders of magnitude (few ms – few 100 ms), different analysis times may be optimal. Since each pixel has the same noise distribution, the optimal pixel size is the size for which the signal spans the smallest number of pixels, so that the statistic is the least polluted by noise.

Since the optimal analysis time for the incoming signal is not known, X-PIPELINE uses several analysis times, and applies a second layer of clustering between analysis times. For this second layer of clustering, clusters made from black pixels at two different analysis times that overlap in time and frequency are compared. The cluster that has the smallest detection likelihood is discarded, and only the louder one is kept as a candidate event.

2.6 Tuning and upper limits

In order to estimate how well a pipeline is working, we need a tool to quantify how well that pipeline separates background from noise. One way is to apply the pipeline to detector strains to which simulated (“injected”) GWs have been added. We then establish upper limits (UL), defined as the largest signal that the pipeline can’t distinguish from noise. But first we need to define what we mean by the “size” of a signal.

The quantity that is used in the LIGO burst group is

$$h_{rss} = \sqrt{\int_{-\infty}^{\infty} (h_{+}^{2}(t) + h_{x}^{2}(t))\mathrm{d}t}, \tag{46}$$

the root-sum-squared amplitude of the signal. The units of h_{rss} are $(\mathrm{Hz})^{-1/2}$, the same as for amplitude spectra. The advantages of this quantity are that it is simple to compute, and has an astrophysical meaning. It can be linked to the energy emitted as GW under the assumption of an isotropic radiation with [9]

$$E_{GW}^{iso} \simeq \frac{\pi^2 c^3}{G} D^2 f_0^2 h_{rss}^2, \tag{47}$$

where D is the distance to the source, and f_0 is the dominant frequency of the radiation. The drawbacks of this approach are that it does not involve detector sensitivity, so what is estimated is the combined sensitivity of the detectors and the data analysis pipeline. Thus the upper limits will depend on the family and frequency of waveforms used (since the detector sensitivity varies with frequency), and also on the sky position of the source.

2.6.1 Open Box

An “open box” analysis looks at data that may contain a GW signal. For a GRB, it is thought that any associated GW must occur within a few tens of seconds of the GRB trigger time [4]. We therefore analyze a few minutes of data around the GRB time, called the *on-source* data, for GWs.

The procedure for estimating upper limits in an open box analysis is:

1. Find the loudest event in the on source data (after applying the glitch veto of section 2.2, if used).

2. Inject a fixed signal at different times in the on source data (one at a time), and compute the detection statistic for each injection.

3. Compute what percentage of injections has a detection statistic greater than the loudest on source event.

4. Repeat the injections with the same waveform but with different h_{rss} amplitudes, and find the h_{rss} for which 90% (or 50%) of injections are louder than the loudest on source event.

This procedure give the 90% upper limit on the h_{rss} for a given waveform. It allows to say that with a 90% confidence level there has been no GW of the given waveform with an h_{rss} greater than some value. Using equation (47), this can be used to constraint the amount of energy emitted as GW by the GRB source.

2.6.2 Closed Box

A "closed box" analysis is very similar to an "open box" analysis, but it estimates the pipeline sensitivity using data that does not contain a GW, called *off-source* data. This permits tuning that is not biased.

The only difference in computing upper limits is in the first step. Rather than computing the loudest on source event, an off-source data stretch of a few hours is selected and cut into segments of the same length as the on source data. For each off source segment the loudest event is computed, and then the median of those events is computed as an estimate of what the loudest on-source event will be. (For a pessimistic estimate, the 90% percentile loudest off-source event is used as the estimate.) Then injections are compared to this estimated on source event rather than to the true on source loudest event.

In practice the off-source data and injections are analyzed first. This allows study of the properties of the noise. It also allows tuning of the analysis parameters (such as the threshold C in the coherent veto test, eqn (35)) to maximize the expected sensitivity of the analysis. After the tuning is fixed, the loudest event in the on-source data is determined. This "blind" tuning procedure gives unbiased ULs. Finally, if the loudest on-source event is much larger than expected from the off-source data, it may be considered as a true GW.

3 Results

The methods described in the previous section have been applied to 10 GRBs that happened during S5, the fifth science run of the LIGO network which started in 2005, and since May 2007 is coincident with the first the first science run (VSR1) of the Virgo detector.

The conventional name for GRBs are based on the date at which the GRB happened, for example GRB 070520B means that it is the second GRB of May 20, 2007. When there is only one GRB on a given day the final letter is not used.

3.1 Injections

Though the real astrophysical GW waveforms are not known, the best way to test a pipeline is to add some GW signal to the detectors noise and try to detect it. A standard choice for those injections are Gaussian-modulated sinusoids, or *sine-Gaussian*. These waveforms have no physical justification but they are very simple, and have similar proprieties to real signals, for example Lazarus [2] black hole merger waveforms (see Fig 9). Equations for circularly polarized sine-Gaussians (noted briefly CSG) are

$$h_+(t) = h_0 \sin(2\pi f_0 t) \exp\left(-\frac{(2\pi f_0 t)^2}{2Q^2}\right), \tag{48}$$

$$h_\times(t) = h_0 \cos(2\pi f_0 t) \exp\left(-\frac{(2\pi f_0 t)^2}{2Q^2}\right), \tag{49}$$

where f_0 is the central frequency, h_0 is an over all rescaling, and Q is a dimensional constant roughly equal to the number of cycles of oscillation greater than half the peak value. Linearly polarized sine-Gaussians (noted briefly SG) are the same except that the cross polarization is zero. In order to sample the parameter space of possible GW signals, the pipeline is tested using CSG and SG with 5 different frequencies and 3 different values of Q.

3.2 Incoherent versus null energy veto

We shall discuss here on an example of background events, from 24 hours background of data around the trigger time of GRB 070615 using the H1H2L1 network. If we look at a scatter plot (Fig 7) of this background we see that the

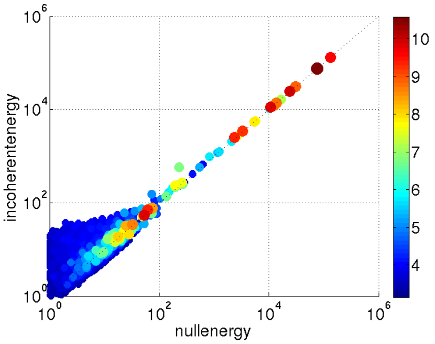


Figure 7: 24 hours of background events for GRB 070615. Each event is represented by a dot. The color of the dot is the log of the hard constraint likelihood, the x and y axis are respectively the null energy and the incoherent energy of the event. Loud (red and yellow) events are along the diagonal.

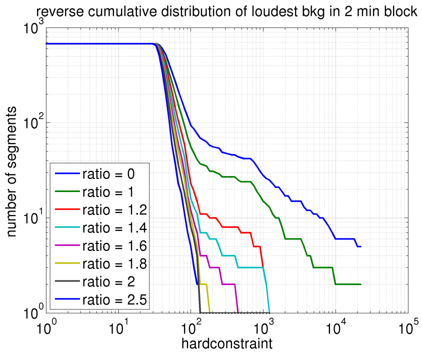


Figure 8: This figure show how many GRB 070615 off source segments have a loudest event louder than the value of the hard constraint on the x-axis. The different colors shows the effect of the veto test for different ratio thresholds, ratio = 0 means that the test is not applied. The test preferentially eliminates loud background events.

loudest background events fall on the diagonal, that means that loud background events are behaving as expected and have $E_{inc}/E_{null} \simeq 1$.

In order to have a better knowledge of how the coherent veto test is affecting the expected upper limits we can look at Figure 8. This shows the distribution of the loudest off source events, i.e., the distribution of the expected loudest on source event. We can see that setting the ratio threshold at 1.2 or above “kills” the worst glitches, but the effect of ratios between 1.8 and 2.5 are almost the same. A higher ratio threshold means that more background events are removed, but it cannot be too high, otherwise the veto test will reject real signals (injections), as can be seen on Figure 10. Thus ratios between 1.2 and 1.8 seem like a good choice.

To better pinpoint a good choice for the ratio threshold, an optimization with regard to several waveforms has been done. The waveforms used are CSG and SG at 100 Hz, 150 Hz, 250 Hz, 554 Hz and 1000 Hz, with Q equal to 4.5, 9 and 18. Depending on the waveform the optimal ratio (the ratio that gives the best expected UL) is between 1.3 and 2.3. Since this optimization does not give a unique preferred ratio, the ratio of 1.5 has been selected arbitrarily for all other analyses, as a good “middle of the road” choice.

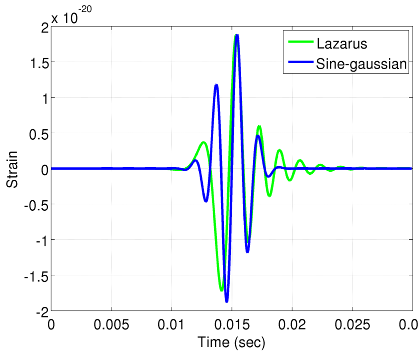


Figure 9: A Q=4.5 sine-gaussian at 554 Hz on top of a Lazarus simulation of the collision of two $15 M_\odot$ black holes. Only one polarization is plotted.

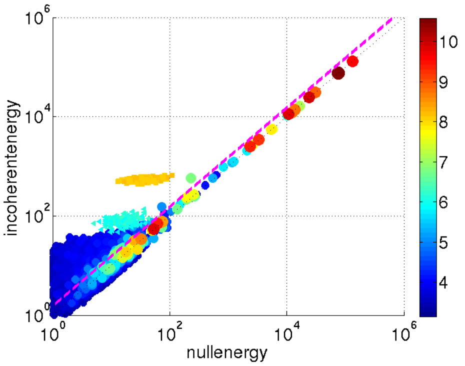


Figure 10: Figure as Figure 7 but with injections superposed. Injections are circular sine-Gaussians at 150 Hz with h_{rss} equal to $1.3 \cdot 10^{-21}\,\mathrm{Hz}^{-1/2}$ (yellow squares), $4.6 \cdot 10^{-22}\,\mathrm{Hz}^{-1/2}$ (cyan triangles). The magenta dashed line is the line of incoherent vs. null energy ratio equal to 1.5. Injections are above this line, and most loud background events are below.

3.3 Closed box upper limits

We shall presents results for closed box analyses of two representative GRBs : GRB 070615 and GRB 070520B. For the first GRB the full LIGO network was online (H1H2L1 network), and it is representative of the performance when detectors are working well. For the second GRB only H2 was off (H1L1V1G1 network), but the detectors were very glitchy at that time.

3.3.1 GRB 070615

In figure 11 we can see that as expected the 90% upper limits follow the detectors sensitivity. If we concentrate on one particular injection waveform, Q=9 circular sine-gaussian at 150 Hz, the detailed results for different detection schemes are in Table 1.

In the typical case (against the median), all detection schemes perform similarly. The hard constraint with veto performs the best and the glitch standard performs the worst, with a 20% difference between best and worst ULs. However if we look at the pessimistic case where there is a glitch during our detection

detection statistic	hrss 90% UL against median ($\mathrm{Hz}^{-1/2}$)	hrss 90% UL against 90 percentile ($\mathrm{Hz}^{-1/2}$)
hard constraint without veto	$2.3 \cdot 10^{-22}$	$5.6 \cdot 10^{-22}$
hard constraint with veto	$2.1 \cdot 10^{-22}$	$2.6 \cdot 10^{-22}$
glitch standard likelihood	$2.5 \cdot 10^{-22}$	$2.6 \cdot 10^{-22}$

Table 1: hrss upper limits on Q=9 CSG at 150 Hz for GRB070615.

window (against 90th percentile), the performance of the hard constraint drops radically. This is because it assume a Gaussian distributed noise (i.e., no glitches), and this assumption is wrong in the pessimistic case. However both approaches to cope with non-Gaussian noise work equally well, and the performance in the pessimistic case is not much worse than in the typical one.

For this network (H1H2L1), there are three detectors, thus the null space is one dimensional. H1 and H2 are two collocated and coaligned detectors, so $d_{\mathrm{H1}} - d_{\mathrm{H2}}$ is the null stream. This is going to be important when comparing the results with GRB 070520B.

3.3.2 GRB 070520B

A big difference between GRB 070520B and GRB 070615 is that during GRB 070520B the detectors were exceptionally glitchy. The easiest way to notice this is by looking at Figure 12. This shows that for GRB 070520B a large number ($\simeq$ 20%) of background segments contains an usually small number of events. Black pixels (see Sec 2.5) cover a fixed fraction (1%) of the TF map, so a small number of events (clusters) means that events contain lots of pixel, i.e., large clusters are created by strong detector glitches. Looking directly at the TF maps for a few of those segments confirms this diagnosis.

Table 2 shows the same UL results as Table 1 (for Q=9 circular sine-gaussian at 150 Hz), but for GRB 070520B.

For the typical case using the veto is harmful, but without the veto both statistics perform equally well. For the pessimistic case, again the hard constraint alone does not work well, and the glitch standard likelihood performance is not much worse than in the typical case. However the hard constraint with the veto yield exactly the same results as in the typical case.

The explanation for this strange behavior of the coherent veto is the incoherent

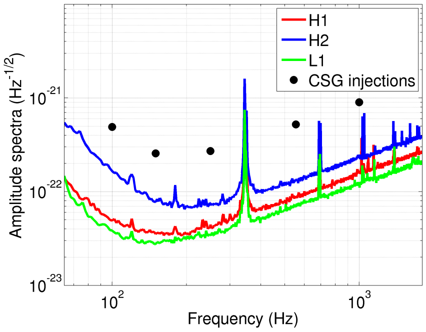


Figure 11: Amplitude spectra of detectors during GRB 070615, superposed with the values of the 90% upper limits on hrss at various frequencies, for detection with the hard constraint likelihood and without the coherent veto test. Upper limits are against the median loudest off source event.

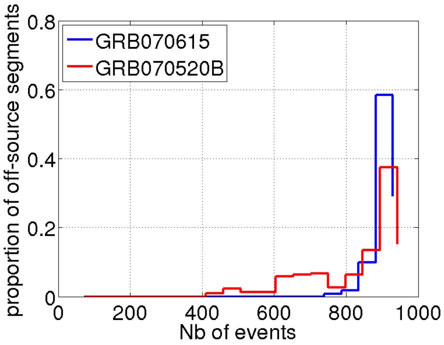


Figure 12: Normalized histogram of the number of events in each off source segment for GRB 070615 and GRB 070520B

detection statistic	hrss 90% UL against median (Hz$^{-1/2}$)	hrss 90% UL against 90 percentile (Hz$^{-1/2}$)
hard constraint without veto	$2.8 \cdot 10^{-22}$	$5.6 \cdot 10^{-22}$
hard constraint with veto	$3.7 \cdot 10^{-22}$	$3.7 \cdot 10^{-22}$
glitch standard likelihood	$2.7 \cdot 10^{-22}$	$2.9 \cdot 10^{-22}$

Table 2: hrss upper limits on Q=9 CSG at 150 Hz for GRB 070520B.

null stream sensitivity. In order for the test to work there has to be some GW signal to suppress from the incoherent energy, otherwise the null and incoherent energy will be approximately the same, and the injections will be rejected. For GRB 070520B the detector network is H1L1V1G1, so the dimension of the null space is 2. Virgo and GEO are much less sensitive than the LIGO 4 km detectors at the injection frequency of 150 Hz (one order of magnitude or more), so the null vector is composed of the Virgo and GEO strains and hardly contains any GW signal for injections near the 90% UL. The result in Table 2 shows that injections needs to have an h_{rss} of more than $3.7 \cdot 10^{-22}\,(\mathrm{Hz})^{-1/2}$ to pass the test, and this is what is limiting the sensitivity of the second detection scheme. By contrast, for GRB 070615 the sensitivity of the veto is determined by H2 not V1, and the H2 sensitivity is better by a factor of 2. Therefore, injections only need an h_{rss} greater than about $1.8x1022\,(\mathrm{Hz})^{-1/2}$ to pass the test, and thus the coherent veto does not limit the UL for GRB 070615.

3.4 Performance relative to other methods

The current LIGO-standard GRB analysis is a cross-correlation algorithm that has been applied to GRBs during S2, S3, S4 (short LIGO science runs in 2003-2005), and S5. A paper with the results of the S2-S4 searches is to be published shortly [14]. In addition to computing cross-correlations instead of likelihoods, this algorithm uses only two analysis times, and it does not use time-frequency maps, but rather computes the cross correlations across a fixed frequency range of 60-2000 Hz.

To compare how well X-PIPELINE works with regard to the current GRB analysis, both algorithms have been run on the same GRB: GRB 070201, for which only H1H2 were on. The results are summarized in the Table 3. These results are not the actual upper limits for the GRB[4], but estimate of the performance of each pipeline in the typical case, based on the closed box analysis (see Sec 2.6.2 for details).

T is the time length of the cross correlation, and is analogous to the analysis time used in X-PIPELINE. The upper limits obtained with X-PIPELINE are 50% or 35% lower than with cross-correlation, depending on whether the hard constraint or the glitch standard likelihood is used. Thus X-PIPELINE is a singnificantly

[4] Publication of actual ULs (or detections) from LIGO data requires an extensive and lengthy review process, and is not allowed in articles authored by a subset of the Collaboration, such as the present report.

circular SG frequency	cross-correlation T=25ms 90% UL $\times 10^{-22}\,\mathrm{Hz}^{-1/2}$	X-PIPELINE HC with veto 90% UL $\times 10^{-22}\,\mathrm{Hz}^{-1/2}$	X-PIPELINE glitch SL 90% UL $\times 10^{-22}\,\mathrm{Hz}^{-1/2}$
100 Hz	18.0	7.0	9.2
150 Hz	10.6	4.5	6.3
250 Hz	11.2	5.4	7.0
554 Hz	19.8	9.8	11.7
1000 Hz	34.6	18.9	19.9

Table 3: hrss upper limits on Q=9 CSG at 150 Hz for GRB 070201.

more sensitive pipeline than the currently used cross-correlation algorithm.

3.5 Conclusions

Coherent techniques for the detection of GW bursts are relatively new and under active development in the LIGO Scientific Collaboration. The objective of this project was to complete the development of the X-PIPELINE coherent analysis package, and test in by searching for GWs associated with gamma-ray bursts (GRBs).

Studies of GRBs using the LIGO, GEO, and Virgo detectors demonstrated the need for a method that copes with the non-Gaussian noise glitches that are common in the data. It has been shown that the coherent veto test is capable of rejecting detector glitches while keeping simulated GW signals added to the data; however, for this veto test to perform well the detectors used to make the null stream must have sensitivity similar to the best detectors in the network. For example, we have found that the H1H2 detector pair provides a null stream that does not limit the network sensitivity, while null streams using the less sensitive Virgo or GEO detectors can limit the network sensitivity.

An alternative approach of using the glitch standard likelihood (a single likelihood that includes both the detection statistic and a glitch veto) was also tested. We found that the glitch standard likelihood performs slightly worse than the hard constraint likelihood with coherent veto in the typical case; however, it can perform better when there is no good null stream available (e.g., when H1 or H2 is not available) and the detectors are glitchy. Finally, with either detection statistic, X-PIPELINE shows better sensitivity to GWs than the current LIGO-standard cross-correlation algorithm.

3.6 Work done

- Coding and testing of the two layers of clustering (including extensive testing of different clustering options for a large range of GW waveforms).
- Coding the post processing scripts that estimate upper limits and produce report web pages.
- Extensive work coding the python scripts that set up data analysis jobs on the Caltech computer cluster.
- The code development history is available at http://www.lsc-group.phys.uwm.edu/cgi-bin/cvs/viewcvs.cgi/matapps/src /searches/burst/coherent-network/?cvsroot=lscsoft#dirlist, it involves 9000 lines added and 6500 lines removed by user mwas
- Deriving, implementing, and testing an alternative GW detection statistic: the glitch standard likelihood
- Running the analysis on 10 S5 GRBs, including GRB 070201
- First steps in generalization of the GRB methods described to all sky searches using spherical harmonics decomposition.

A Howto run GRB searches with X-Pipeline

Working copy of a howto file, the most up to date version is available on CVS in
http://www.lsc-group.phys.uwm.edu/cgi-bin/cvs/viewcvs.cgi/matapps/src/searches/burst
/coherent-network/doc/code/grb.html?cvsroot=lscsoft

A.1 Getting X-Pipeline from CVS

To get X-Pipeline from CVS, do this (csh):

```
setenv CVSROOT ":pserver:anonymous@gravity.phys.uwm.edu:2402/usr/local/cvs/lscsoft"
cvs login
cvs checkout xpipeline
```

or this (bash):

```
CVSROOT=":pserver:anonymous@gravity.phys.uwm.edu:2402/usr/local/cvs/lscsoft"
export CVSROOT
cvs login
cvs checkout xpipeline
```

Hit enter when prompted for the password. This gets you the current snapshot of the Xpipeline subdirectories of the CVS archive.

A.2 Compiling

Compiling has been tested on MatLab release 2006a (32 and 64 bit) and r2007a (64 bit), but should for MatLab r14 and later. Compiling works on 32 bit and 64 bit machine, 64 bit executables run on the grid; 32 bit executables require a modification to *.sh scripts, those scripts load libraries according to processor and not compiler. For 32 bit code to run on the grid you need to hardwire variable $BIT to something else than x86_64 (e.g i686). Do not enable multithreading (available for release 2007a and later, is disabled by default)

1. Set your environment variables to point to the directories wheres X-Pipeline and MatLab are installed:

   ```
   export XPIPELINE_MATLAB_ROOT=/ldcg/matlab_r2006a/
   export XPIPELINE_ROOT=/archive/home/mwas/scratch_xpipeline/xpipeline/
   ```

   You can add these lines to your .bashrc or .bash_profile file so that the variables are set automatically when you log in. Otherwise you'll need to set them again when you log on to ldas-grid to launch jobs (see below).

2. compile frame functions. frgetvect.mex* that are in CVS might work for you but is safer to recompile them for your platform.

```
cd xpipeline
tar -xvzf v6r19b.tar.gz
cd v6r19b/mgr
./makegcc
cd ../matlab
./mymex
cp frgetvect.mex* ../..
cd ../..
```

3. open a matlab and add xpipeline to path

```
matlab -nodisplay
addpath([pwd '/'])
addpath([pwd '/post'])
addpath([pwd '/utilities'])
savepath ./pathdef.m
```

4. compile fastclusterprop, mergeLikelihoods, xdetection and xmakegrbinjectionfile

   - 32-bit compilation

   ```
   cd post
   mex fastclusterprop.cpp
   mcc -m mergeLikelihoods.m
   cd ..
   mcc -a post/fastclusterprop.mexglx -m xdetection.m
   mcc -m xmakegrbinjectionfile.m
   exit
   ```

   - 64-bit compilation

   ```
   cd post
   mex fastclusterprop.cpp
   mcc -m mergeLikelihoods.m
   cd ..
   mcc -a post/fastclusterprop.mexa64 -m xdetection.m
   mcc -m xmakegrbinjectionfile.m
   exit
   ```

A.3 Running an example analysis

5. cd to the grb directory and edit grb.ini file to suit your needs. The most important is to set in section [condor] log path to writeable files by you. Detailled instructions about grb.ini file in appendix B

```
cd grb
vi grb.ini
```

6. create dag files following these instructions http://www.ligo.caltech.edu/∼psutton/protected/xpipeline/grb_py.html

   ```
   ./grb.py -p grb.ini -g 854378604 -r 11.089 -d 42.308 -i H1 -i H2
   ```

7. ssh to ldas-grid or ldas-pcdev1 (condor jobs can't be submitted from ldas-pcdev2), cd into xpipeline/grb and submit jobs

   ```
   condor_submit_dag grb_*.dag
   ```

   You can check the status of jobs using:

   ```
   tail -f grb_on_source.dag.dagman.out
   ```

A.4 creating a html report page

To obtain injection results in a nice html format (for an analysis with output type 'clusters'):

1. copy/move results into the same directory (results are stored in output/*/*.mat)

2. copy your grb.ini file into this directory

3. copy pathdef.m file created during compilation into this directory

4. run grb/makeAutoFiles2.py with the same options as grb.py

   ```
   \$XPIPELINE_ROOT/grb/makeAutoFiles2.py -p grb.ini -g 854378604 -r 11.089 -d 42.308 -i H1 -i H2
   ```

5. open matlab **without display** and add xpipeline/post to path if you have not a pathdef.m file

   ```
   matlab -nodisplay
   addpath(path_to_xpipeline/)
   addpath(path_to_xpipeline/post)
   addpath(path_to_xpipeline/utilities)
   ```

6. run makeWebPage, for more details on makeWebPage see appendix C

   ```
   makeWebPage('fakegrb','auto.txt',1.5)
   ```

The output web page should look like this http://ldas-jobs.ligo.caltech.edu/ mwas/howtoTest/openbox/fakegrb.shtml

A.5 scripted html report page

creates the same report page with a single command

1. compile makeWebPage on a 32-bit machine

```
matlab -nodisplay
addpath(path_to_xpipeline/)
addpath(path_to_xpipeline/post)
addpath(path_to_xpipeline/utilities)
mcc -R -nodisplay -m makeWebPage.m
```

2. cd into the directory where grb.py has been run

3. create a text file named grbname.txt containing the grb name

   ```
   echo "fakegrb" > grbname.txt
   ```

4. run the script

   ```
   \$XPIPELINE_ROOT/grb/grbResults.sh openbox
   ```

5. your report web page is in the openbox subdirectory, its name is fakegrb.shtml

B grb.ini description

Important parts of grb.ini are :

- set log files to writable file names

B.1 Background

```
[background]
backgroundPeriod = 86400
; lagFile = lags.txt
; backgroundPositionsFile = backgroundPositions.txt
```

backgroundPeriod

B.2 analysis parameters

```
[parameters]
analysisTimes = 0.015625
blockTime = 128
likelihoodType = hardconstraint,nullenergy,incoherentenergy
minimumFrequency = 64
maximumFrequency = 1024
offsetFraction = 0.25
outputType = timefrequencymap
sampleFrequency = 4096
skyCoordinateSystem = radec
verboseFlag = 1
whiteningTime = 1
extractParamFileName = input/extractParameters.txt
```

outputType

- timefrequencymap - produce timefrequency maps of selected likelihoods for each analysis time clusters - produce list of clustered event, should be used with
- likelihoodType = hardconstraint,nullenergy,incoherentenergy
- followup - produce the loudest cluster within 0.5 sec of the middle of the time-frequency map for each sky positions. WARNING this is a quick and dirty hack. Parameters are hardwired in the code of xtimefrequency map. Output is stored in 2 dim array called likelihoodMap\likelihoodMapCell. first dim - sky position as in skyPositions, second dim - cluster proprieties as described in clusterTFmapNew
- timeseries - the outputs are time series (the time-frequency map summed over all frequencies) for each individual sky position.

catalogDirectory path to directory containing catalogs of GW waveforms (ZM, OB, ...), one such directory is on CVS xpipeline/waveforms

For more information check this link
http://www.ligo.caltech.edu/∼psutton/protected/xpipeline/xdetection_parameters.html

B.3 Post processing options

```
[extraction]
onSourceTimeWidth=0.1
onSourceTimeOffset=0.2
maxNanalysisTimes=0
clusteringMethod=connected
useClustering=1
useGeneralizedClustering=1
mergingThresh=10
sizeThresh=0
bpp=0.01
connectivityNumber=4
useIncNullEnergy=0
ratioIncNull=1.2
scatterPlot=1
eventProportion=0.01
clusterProportion=0.1
```

onSourceTimeWidth the loudest cluster in a time +/- onSourceTimeWidth around injection time is recorded as the injection event. onSourceTimeWidth is in seconds.

onSourceTimeOffset offset between the injection as recorded in the injection file and the window defined by onSourceTimeWidth (useful for inspirals, they are not centered on coalescence time). onSourceTimeOffset is in seconds

maxNanalysisTimes restrict post processing to the first maxNanalysisTimes analysis times. If set to 0, all analysis times are processed.

clusteringMethod available methods are: 'sumoverfrequency', 'maxoverfrequency','connected','fastconnected'. 'sumoverfrequency' is always computed during post processing whatever method is chosen. 'sumoverfrequency','maxoverfrequency' should not be used, they maybe slow.

useClustering 1 - use clusteringMethod, 0 - do not use any clusteringMethod ('sumoverfrequency' is still computed in a fast way)

useIncNullEnergy 1 - use incoherent vs null energy ratio as a consistency test. 0 - keep all events (the consistency test may always be applied at a later stage). WARNING this option might not work.

ratioIncNull event with use incoherent vs null energy ratio below ratioIncNull are rejected.

scatterPlot 1 is the only accepted value.

eventProportion keep only the loudest eventProportion of sumoverfrequency events

Options for connected and fastconnected clustering. Generelized clustering works only for connected clustering.

useGeneralizedClustering 1 - cluster within a block-city distance of up to 3 are merged together and saved as one bigger cluster. 0 - simple connected clustering

mergingThresh for generalized clustering, threshold on the significance of the first likelihood. Two nearby clusters are merged only if both of them have a significance above this threshold.

sizeThresh only clusters with size (number of pixels) greater or equal than sizeThresh are kept as event (and possibly merged). This option has been disabled.

bpp black pixel probability. bpp is the proportion of the TF map that will be set as black pixels.

connectivityNumber 4 - nearest neighboor connected clustering. 8 - next nearest neighbor connected clustering (pixels with touching corners are considered connected)

clusterProportion keep only the loudest clusterProportion of clusters

B.4 where and how to distribute results

This section is useful if you want to distribute not post processed results across multiple nodes, text files understood by extractMeasuredLikelihoods are produced for retrieval and post-processing. Or spread injections across multiple jobs.

```
[output]
nodePath = /data/node
nNodes = 4
numberOfFirstNode = 100

distributeOnSource = 0
jobNodeFileOnSource = distribonsource.txt

distributeOffSource = 0
onNodeOffSourcePath = psutton/fullsizetest/
jobNodeFileOffSource = distriboffsource.txt

distributeSimulation = 0
onNodeSimulationPath = psutton/fullsizetest/
jobNodeFileSimulationPrefix = distribsimulation
maxInjNum = 0
```

nodePath generic path to nodes, eg at CIT /data/node and the path to each node disk is /data/nodeX where X is a number between 1 and 330, if the path is something like /data/node003 rather than /data/node3 this script won't work.

nNodes number of nodes to use for results distribution

numberOfFirstNode number of the first node to use: nodes numberOfFirstNode to numberOfFirstNode+nNodes-1 will be used for postprocessing

distributeOnSource 1 - distribute results of on source jobs (in fact the results are stored locally in the output directory, because these results don’t take much space) 0 - save results in local output directory

jobNodeFileOnSource name of text file where distribution description is saved

distributeOffSource 1 - distribute results of off source jobs 0 - save results in local output directory

onNodeOffSourcePath path on node where the results are to be stored, the trailing / is important. eg psutton/fullsizetest/ save results in /data/nodeX/psutton/fullsizetest/

jobNodeFileOffSource name of text file where distribution description is saved

distributeSimulation 1 - distribute results of simulation jobs, 0 - save results in local output directory

onNodeSimulationPath path on node where the results are to be stored, the trailing / is important. eg psutton/fullsizetest/ save results in /data/nodeX/psutton/fullsizetest/

jobNodeFileSimulationPrefix prefix of text files describing distribution of simulation results, the waveform tag will be appended to this prefix

maxInjNum cut MDC simulation in jobs that contain no more than maxInjNum injections per job, results from those jobs can be merged using post/mergeLikelihoods.sh (run post/mergeLikelihoods.sh in output directory, results form output/*/*.mat are merged into output/*merged.mat). This option is useful to keep jobs shorter than 4 hours.

B.5 data frame files information

```
[input]
; ; -$\,$-$\,$-$\,$- Detector and channel info for simulated data from LIGO-Virgo project.
; detectorList = [’G1’,’H1’,’H2’,’L1’,’V1’]
; channelList = [’STRAIN’,’STRAIN’,’STRAIN’,’STRAIN’,’noise’]
; frameTypeList = [’SIM’,’SIM’,’SIM’,’SIM’,’SIM’]
; -$\,$-$\,$-$\,$- Detector and channel info for real data.
detectorList = G1,H1,H2,L1
channelList = DER_DATA_H,LSC-STRAIN,LSC-STRAIN,LSC-STRAIN
frameTypeList = RDS_C01_L3,H1_RDS_C02_LX,H2_RDS_C02_LX,L1_RDS_C02_LX
; ; -$\,$-$\,$-$\,$- Detector and channel info for real time-shifted data from the LIGO-Virgo
; detectorList = G1,H1,H2,L1,V1
; channelList = DER_DATA_H,LSC-STRAIN,LSC-STRAIN,LSC-STRAIN,h_16384HzNo50
; frameTypeList = G1_LIGOVIRGO_2B,H1_LIGOVIRGO_2B,H2_LIGOVIRGO_2B,L1_LIGOVIRGO_2B,V1_LIGOVIRGO
; frame cache:
; frameCacheFile = misc/framecache_mdc.txt
```

B.6 MDC injections frame files information

```
[mdc_insp1414]
; -$\,$-$\,$-$\,$- Info on MDC set: 1.4-1.4 inspirals.
;      ORDER MUST MATCH LIST OF KNOWN DETECTORS IN [input]!  Fill missing entries with ’None’.
channelList = None,STRAIN_INSP_1414,STRAIN_INSP_1414,None
frameTypeList = None,H1_GRB070201_INSP_1414,H2_GRB070201_INSP_1414,None
numberOfChannels = 165

[mdc_insp1410]
; -$\,$-$\,$-$\,$- Info on MDC set: 1.4-10.0 inspirals.
;      ORDER MUST MATCH LIST OF KNOWN DETECTORS IN [input]!  Fill missing entries with ’None’.
channelList = None,STRAIN_INSP_1410,STRAIN_INSP_1410,None
frameTypeList = None,H1_GRB070201_INSP_1410,H2_GRB070201_INSP_1410,None
numberOfChannels = 165
```

B.7 Log files

```
[condor]
universe = vanilla
; xsearch = /archive/home/psutton/coherent-network/xdetection.sh
dagman_log_on_source = /usr1/mwas/log/grb_jobs_on_source.logs
dagman_log_off_source = /usr1/mwas/log/grb_jobs_off_source.logs
dagman_log_simulations = /usr1/mwas/log/grb_jobs_simulations.logs
dagman_log_mdcs = /usr1/mwas/log/grb_jobs_simulations.logs
retryNumber = 0
```

retryNumber number of times condor retries to submit a job after a failure

B.8 LSCdataFind

```
[datafind]
; datafind  = /opt/lscsoft/glue/bin/LSCdataFind
; lal-cache =
server = ldas-cit.ligo.caltech.edu
; match = localhost
```

B.9 injections

```
[injection]
; population = uniform_sky
; strain-dist = loghpeak
; strain-scale-min = -0.2
; strain-scale-max = 3
injectionScales = 1
```

```
;injectionScales = 1,2,4,8
injectionInterval = 45.741592654
```

injectionScales comma separated list of injection scales that are to be done

injectionInterval for software injections only, time interval between two software injection, thus the number of software injection is approximately blockTime/injectionInterval.

Do not use software and MDC injections together, you need two separate grb.ini files to handle each injection type. Be sure that mdc_sets is put in comments if you use software injections.

B.10 software injections

```
[waveforms]
;laz = Lazarus!50~17.1968~1
dfm = DFM!10~A3B5G4,DFM!10~A4B5G4
```

Format :
waveform_tag = waveform_name!waveform_paramters,...

waveform_name,waveform_paramters have to be in the format understood by xmakewaveform; if there is more than one waveform in the comma separated list, for each injection one waveform is chosen randomly form the comma separated list.

For catalog waveforms like DFM or ZM. If your are not on CIT cluster change hardwired path in xmakewaveform to point to local xpipeline/waveforms.

B.11 MDC injections

```
[mdc]
;mdc_sets = mdc_insp1410,mdc_insp1414
```

mdc_sets comma separated list of mdc waveforms tag, for each mdc_tag a [mdc_tag] section must exist

in section [input] frameCacheFile has to be set, this file must say where frames with mdc injections **AND** frames with detector data are stored.

If you want to post-process MDC injections you need to create "injection files" of the form input/injection_mdc_tag.txt. Thoses files have to describe MDC injections in the following format: 7 columns white space sperated text file; gps injection time seconds, gps injection time nanoseconds, phi, theta, polarization angle, waveform name, waveform parameters. The last two column can have any value.

C makeWebPage.m

```
% [UL,BR,SNRlist,PN]=makeWebPage(grbName,autoFileName,ratioArray,analysisType,hwInjFile,mdcInjection)
%
% Create a report web page for one GRB
%
% webPageFileName - String. Name of ouput web page.
%
% autoFileName - name of file giving names of results files as produced by
% grb/makeAutoFiles2.py
%
% ratioArray - Vector. Vector of incoherent vs null energy ratio to try.
%
% analysisType - String. Optional. Possible values are
%
%       closedbox - the on source segment is not used. The loudest on
%       source event is _estimated_ as the median of the loudest events of
%       the backgroun off source segments (one loudest event per segment,
%       and then take the median)
%
%       snrulclosedbox - same as closedbox, however upper limits are not
%       done on the injection scale but on the total SNR desposited in the
%       network by the injections (see ulSNREsitmation for more details).
%       This analysis type is EXPERIMENTAL.
%       Plots displaying scatter plots with injections are wrong. In the
%       table at the bottom values are all wrong. The only meaningful
%       columns are the 50%, and 90% UL on the injections scale, they
%       contain the 50% and 90% UL on the total SNR deposited into the
%       network.
%
%       for any other string a standard open box is performed (UL against
%       the loudest on source event).
%
% hwInjFile - String. Optional. Name of a two column text file containing a
% list of start, stop
% times of time period to ignore (e.g. hardware injection periods). If
% the string is empty it is ignored.
%
% mdcInjection - Logical. Optional. Default 0, if set to 1 xmakewaveform is
% not used. Use for mdc injection not recongnized by xmakewaveform.
%
% WARNING: the first likelihood listed in likelihoodType (or in grb.ini)
% is used as the detection statistic
% WARNING: hardware injections removed ONLY if they happen in a time +-
% 100000 s GRB time, for longer off source period hardware injection are
% NOT removed
% WARNING: comments will appear when the web page file name has a shtml
% extension and viewed through a web server like Apache, local opening of
% this web page won't display comments.html
% WARNING: some display function might not work if there is more than 6
% detecors (search colorList in the code)
% WARNING: do not use with more than 1e6 injections, see KLUDGE in
% ulEstimation, ulTuning

% \$Id: makewebpage.html,v 1.1 2007/08/05 21:29:47 mwas Exp \$
```

- to add comments at the top of the results web page create a web page named comments.html
- figures are stored in 'figures' subdirectory, and their corresponding .fig files are in 'figfiles' subdirectory
- autoFiles created by makeAutoFiles2.py are: 'auto.txt', 'fastauto.txt'. Use 'auto.txt' if you want to tune the incoherent versus null energy ratio cut on the off-source segments. Use 'fastauto.txt' if you have decided on a fixed cut (1.5 seems to be a reasonable value) and want a faster analysis , do not put more than one ratio in ratioArray, otherwise your cut will be tuned on the on-source segment. For 'fastauto.txt' the off-source events are not analyzed.

C.1 Description of output report web page

This description is valied for analysis type openbox, for other analysis types it is not completely accurate.

C.1.1 Header

In the title of the web page (it should be in top bar of your borwser) is the path where the given report page has been detected. It is usefull for retrieving the .mat files associated with the web page.

The name of the grb is the one supplied to makeWebPage

C.1.2 detector proprieties section

- GRB trigger proprieties: GPS time and sky position
- antenna patterns for each detector that was on during the GRB
- list of the analysis time (fourrier integration time) used during the analysis
- amplitude spectra of the detectors represent the overall sensitivity GW of each detector, amplitude spectra divided by sqrt(F_+ˆ2 +F_xˆ2) gives the sensitivity coming from the GRB sky position

- plot of the "gaussianity", mean(x(f)^4)/(mean(x(f)^2))^2, the dashed line is the 3 sigma spread for a gaussian distributed noise. The equation for the spread is 2 +/- 3 * sqrt(20/N), where N is the number of sample that are used for making the mean.
- Contributions of each detector in the network to the F_+, F_x, and standard likelihood streams, those contributions should sum up to 1. The last plot F_x/F_+ show how the cross stream is sensitive with regard to the plus stream
- normalized contribution to the null stream, those contribution should sum up to the number of null stream (the dimension of the null space). Sensitivity to the incoherent null stream, it represent how much GW singal can be supressed from the incoherent energy when making the null energy, GW signal have to be 2 or 3 times above this curve to pass the incoherent versus null energy ratio test with ratio threshold equal to 1.5.

C.1.3 background section

- scatter plot of the events in the on source segment
- block time used for the on source segment
- table containg the 10 loudest on source events
- scatter plot of the events in the off source segments, when fastauto.txt is used on source events are used instead of off-source.

C.1.4 injections

each waveform has a seperated section, the section name give the arguments: waveform name and parameters used in xmakewaveform to create these injections. Possible clustering types are sumoverfrequency, connected. Currently the output is hardwired to connected, but can be easily changed by editing clusterTypeCell in makeWebPage. Connected displays the result for the method to which clusteringMethod is set (see grb.ini description for details)

- efficeny versus injection scale plot, the upper limit is based on a linear interpolation of this plot
- scatter plot showing on source background and injections with injection scale that is the nearest to the 90% UL
- same but with off source events
- cumulative histrogram of the detection statistic of the on and off source evetns, with and without the inc. vs null ratio test.

C.1.5 summary

Amplitude spectra and sensitivity in the incoherent null stream with 90% hrss UL superposed.

Table summarizing the results for each waveform, the column are

- waveform name and parameters, the link point to the section concering this injection

- central frequency of the injection
- 50% and 90% upper limits on hrss
- 50% and 90% upper limits on the injection scale
- the ratio used for these upper limits, it is optimized based on off-source background events. The optimization is done by computing for each off-source segment the 90% UL against the loudest event in this segment, and this for each ratio listed in ratioArray. The ratio for which the mean of those 90% UL is the lowest is selected as the best ratio. Outlier off source segment are skipped because they produce NaNs as UL, see bullet point below.
- percentenge of off-source segments that produced a NaN during the optimization. NaNs happen when the array of injection scales is not large enough and all injection scale have a detection probability below 90% or all are above 90%.
- amplitude SNR in each detector
- root-sum-squared of the SNR with noise spectra as in incoherent null stream. It is the SNR deposited into the incoherent energy.
- total SNR deposited into the network, the root-sum-squared of the amplitued SNR deposited in each detector.

D Script list and short descriptions

D.1 main subdirectory

xdetection.sh analyze h(t) data

xmakegrbinjectionfile.sh create injection file (list of simulated GW to add to on-source data) for one GRB (trigger time + sky position)

xmakeallskyinjectionfile.sh create injection file (list of simulated GW to add to on-source data) for a given trigger time, injections are uniformly distributed over the sky.

xpipeline.sh

D.2 grb directory

Unless specified, all those scripts should be run in the same directory for a given grb.

grb.py create a directory structure and dag files for running an analysis of one GRB (one trigger time + sky position)

grbAllSky.py modification of grb.py for running an analysis when only a trigger time is given. This script is not well tested

grbInjScales.sh estimate the best injection scales to use for a given GRB, and add this information to grb.ini. A grb.py script needs to be run before. It should also be run afterwards to apply the changes to grb.ini. It has only been tested for linear SG and circular SG.

grbRemove.py remove results that have been distributed (see section [output] of grb.ini) and are listed in the input text file

grbResults.sh post-process a GRB analysis to get upper limits and report web page

makeAutoFiles.py create files listing results after clustering, those files are needed by makeWebPage. Run this script if clustering is done separately from time frequency map creation (outputType time frequency map, and then makeExtract.py is applied)

makeAutoFiles2.py create files listing results after clustering, those files are needed by makeWebPage. Use this script for an analysis with outputType clusters.

makeExtract.py create dags to extract events from time frequency maps created by xdetection.sh with outputType timefrequencymap

multiGRB.py set up directories and dag files for analyzing a list of GRBs

D.3 post directory

extractMeasuredLikelihoods.sh create events list from time frequency maps

makeWebPage.sh create a report web page for one GRB

mergeLikelihoods.sh merge events list from off source analysis or simulations that have been separated into short jobs to keep below the 4 hour eviction time on CIT cluster.

mergeLikelihoodsWrapper.sh wrapper for running mergeLikelihoods.sh in condor jobs

References

[1] W. Anderson, P. Brady, J. Creighton, and É. Flanagan. Excess power statistic for detection of burst sources of gravitational radiation. *Phys. Rev. D*, 63:042003, 2001.

[2] M. Campanelli, B. Kelly, and Lousto C. The lazarus project. II. spacelike extraction with the quasi-kinnersley tetrad. *Phys. Rev. D*, 73:064005, 2006.

[3] H. Dimmelmeier, J. Font, and E. Müller. Relativistic simulations of rotational core collapse II. collapse dynamics and gravitational radiation. *A&A*, 393:523–542, 2002.

[4] B. Abbott et al. Search for gravitational waves associated with the gamma ray burst grb030329 using the ligo detectors. *Phys. Rev. D*, 72:042002, 2005.

[5] N. Gehrels et al. The *Swift* gamma-ray burst mission. *ApJ*, 611:1005, 2004.

[6] S. Chatterji et al. *Phys. Rev. D*, 74:082005, 2006.

[7] S. Klimenko et al. *Phys. Rev. D*, 72:122002, 2005.

[8] A. Hulse and J. Taylor. Discovery of a pulsar in a binary system. *Astrophys. J. Lett.*, 195, 1975.

[9] K. Riles. LIGO report no. T040055-01-Z. 2004.

[10] P. R. Saulson. Physics of gravitational wave detection: Resonant and interferometric detectors. Prepared for 26th SLAC Summer Institute on Particle Physics: Gravity – From the Hubble Length to the Planck Length (SSI 98), Stanford, California, 3-14 Aug 1998.

[11] L. Scharf. Topics in satistical signal processing. In *Les Houches, Session XLV, Signal Processing*, 1985.

[12] P. Sutton. http://www.ligo.caltech.edu/∼psutton/protected/xpipeline/xpipeline.html.

[13] J. Sylvestre. *Phys. Rev. D*, 66:102004, 2002.

[14] the LIGO Scientific Collaboration. Search for gravitational waves associated with 39 gamma-ray bursts using data from the second, third, and fourth LIGO runs. http://www.ligo.caltech.edu/docs/ScienceDocs/P/P060024-06.pdf.

[15] Robert M. Wald. *General Relativity*, chapter 4. University of Chicago Press, 1984.

[16] LIGO web page. http://www.ligo.caltech.edu/.

[17] Virgo web page. http://www.virgo.infn.it/.