SEARCHING FOR GRAVITATIONAL WAVES FROM COMPACT BINARY COALESCENCE USING LIGO AND VIRGO DATA

By

Collin D. Capano

B.S. Syracuse University, 2005

M.S. Syracuse University, 2008

DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY IN PHYSICS

Syracuse University

December 2011

SEARCHING FOR GRAVITATIONAL WAVES FROM COMPACT BINARY COALESCENCE USING LIGO AND VIRGO DATA

By

Collin D. Capano

B.S. Syracuse University, 2005

M.S. Syracuse University, 2008

Dissertation

Submitted in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy in Physics

Syracuse University

December 2011

Approved _____

Duncan A. Brown

Date _____

ABSTRACT

This thesis describes current efforts to search for gravitational waves from compact binary coalescences (CBCs) by the LIGO Scientific Collaboration (LSC) and the Virgo Collaboration. We briefly review the physics of gravitational-wave emission and detection, describing how gravitational waves are emitted from "inspiraling" compact stellar mass objects and how the LSC and Virgo try to detect them using interferometers. Next we review the data-analysis principles used to search for potential signals in the detectors' noise. These principles are employed by "`ihope`," which is the data-analysis pipeline used to search for CBCs. We describe each step in this pipeline and discuss how interferometer data is stored and examined. Next we present the results from a six-month long search which occurred in early 2007, during LIGO's fifth science run. This is followed by details of tuning studies carried out on LIGO's sixth-, and Virgo's second- and third-, science runs (S6, VSR2, and VSR3), which ran from July 2009 to October 2010. No gravitational waves were detected in these searches. A "blind injection" was performed during S6/VSR3 and detected by our pipeline, however. We detail studies into assigning a statistical significance to this injection. Next we use these studies to show that we can expect to detect gravitational waves with high significance using two detectors in the advanced detector era. Finally, we review some future developments for the CBC pipeline currently being undertaken.

Contents

List of Tables

List of Figures

Preface

The work presented in this thesis stems from my participation in the LIGO Scientific Collaboration and the Virgo Collaboration.

Chapter 6 presents results described in:

B. P. Abbott et al., "Search for Gravitational Waves from Low Mass Compact Binary Coalescence in 186 Days of LIGO's fifth Science Run," *Phys. Rev.,* **D80**, 047101 (2009)

The author was the primary analyst on one month of this data, and contributed to the results presented therein.

Chapter 7 presents some results described in:

J. Abadie et al., "Search for Gravitational Waves from Low Mass Compact Binary Coalesence in LIGO's Sixth Science Run and Virgo's Science Runs 2 and 3," *In preparation.*

As the results discussed in Chapter 7 are currently under review, they are potentially subject to change. Additionally, some of the plots presented in the chapter — namely those relating to data-quality investigations — will not be published and so have not been subject to the same rigorous review that published plots and results undergo. They therefore do not represent the scientific opinion of the LIGO Scientific Collaboration or the Virgo Collaboration.

Conventions

We use the Einstein summation convention in which repeated upper and lower indices are summed over. Greek indices indicate all four spacetime dimensions, Latin indicate spatial dimensions only. An over-arrow indicates a vector with spatial components only, e.g., $\vec{x} = (x^1, x^2, x^3)$.

Expectation values of a variable are indicated by $\mathbb{E}()$. Averages, unless otherwise noted, are indicated by an overbar.

The time-stamps of interferometer data are measured in Global Positioning System (GPS) seconds: seconds since 00:00.00 UTC January 6, 1980 as measured by an atomic clock.

Acknowledgments

Once this thesis exceeded 256 pages, my advisor challenged me to write 512. At first I did not think that would be possible. But as I sit here contemplating all the people that have my appreciation, I realize I could easily reach that number, and exceed it. Still, for the sake of brevity, and the members of my committee, I must limit my acknowledgements to a few individuals. I hope anyone I do not mention here will forgive me.

First, I would like to thank my advisor, Duncan Brown. Had he not asked me what I was planning to do three years ago for a research topic, I would not be here today. His mentorship, and his friendship, have made this all possible.

I would also like to thank the other members of my defense committee — Peter Saulson, Stefan Ballmer, Britton Plourde, Alan Middleton, and Scott Sampson — for taking time out of their busy schedules to read and critique my work.

I have learned much from all of the members of the LIGO Scientific Collaboration and the Virgo Collaboration. I would particularly like to thank Ian Harry, Dave McKechan, Thomas Dent, Duncan Macleod, Steve Fairhurst, Stuncan, Alan Weinstein, Drew Keppel, and Kipp Cannon for all their help, and for collaborating with me on a number of projects. A special thanks also goes to Andrew Lundgren for knowing, well, everything.

Closer to home, I would like to thank the members of the Syracuse University Gravitational Wave Group, particularly Peter Couvares, for all of the stimulating discussions over coffee and noodle bowls. A special thanks goes to Matt West and Larne Pekowsky for entertaining my hair-brained ideas. I could not have asked for better office mates. I would also like to thank Larne for answering all of my UNIX questions, and for the shared experience of writing our dissertations.

Outside of the gravity group, I would like to thank David Niedzwieki and Don

to
my Mom and my Dad

Chapter 1

Introduction

The direct detection of gravitational waves (GWs) promises to usher in a new era of astronomy. The GW spectrum represents an entirely new window on the universe, independent of, and complimentary to, electromagnetic (EM) radiation. Gravitational waves can be used to directly probe objects unobservable by EM telescopes; e.g., the properties of black holes, the equation of state of neutron stars, and the state of the universe prior to the emission of the cosmic microwave background. Joint GW and EM observations offer more possibilities, such as understanding the progenitors of short-hard gamma-ray bursts (GRBs) and measuring the expansion of the universe. The GW spectrum would also give us insight into the physics of strong field gravity and numerical solutions of the Einstein equations, as well as provide a test for alternative theories of gravity [1].

The U.S. Laser Interferometer Gravitational-wave Observatory (LIGO) and the French-Italian Virgo interferometer are seeking to make the first direct detections of gravitational waves [2]. To date, LIGO has completed six *Science runs*. The first five of these runs were known as *initial* LIGO. In LIGO's fifth science run (S5), which lasted from November 2005 to September 2007, the LIGO detectors reached their design sensitivity, as they were sensitive to gravitational waves with strain amplitudes of $\sim 10^{-21}$ in the $40 - 7000\,\mathrm{Hz}$ frequency band [2]. S6, also known as *enhanced* LIGO [3], lasted from July 2009 until October 2010. Hardware improvements were make to the detectors for S6; during this period the LIGO detectors met and exceeded the sensitivity of S5. Virgo has had three science runs. Virgo's first science run (VSR1) overlapped with S5, lasting from May 2007 until Octover 2007 [4]. During this period,

joint LIGO Virgo searches for GWs were carried out [5, 6, 7]. Virgo's second science run (VSR2) and Virgo's third science run (VSR3) ran from July 2009 until January 2010, and from August 2011 to October 2011, respectively. During these runs the Virgo detector operated with improved sensitivity over VSR1. LIGO, having just completed S6, is currently preparing for the Advanced LIGO era, which may begin as early as 2014 [8]. Virgo will also be upgraded on the same schedule as Advanced LIGO [4]. For the Advanced era, both the LIGO and Virgo detectors are projected to have a factor of ten improvement in sensitivity [8, 9], which will allow for GW detections from multiple astrophysical sources [8, 10].

During these Science runs, the LIGO Scientific Collaboration (LSC) and the Virgo Collaboration have carried out several searches for GWs from various astrophysical sources. These searches are broadly grouped into four categories [2]:

- *The CBC group*: These searches look for GWs from compact stellar mass binaries as they "inspiral" into each other and merge [11]. The waveform from these searches can be modelled and a matched-filter analysis used [12].

- *The Burst group*: This consists of un-modelled transient searches that look for GWs from systems that cannot easily be modelled, such as from core-collapse supernovae [13].

- *The Continuous-wave group*: This search looks for GWs from a "continuous" wave source, such as a pulsar [14].

- *The Stochastic group*: This consists of searches for a "stochastic background" of broadband GWs, that, for example, could have been emitted in the early universe [15].

In this thesis we focus on the search for GWs from CBCs. Within the CBC group several different searches are performed. Here, we focus on the *all-sky "low-mass" search*. This search looks for gravitational waves from coalescing binaries that have a total mass $M_{total} < 35\,M_\odot$ and a component mass $\geq 1 M_\odot$ using LIGO and Virgo data.[1] This search includes binary neutron stars (BNS), binary black holes (BBHs), and neutron-star black-hole binary (NSBH) systems.

---

[1]Throughout the rest of this thesis we will refer to this search as the *low-mass CBC search*.

Coalescing compact binaries with a total mass $< 35\,\mathrm{M}_\odot$ have a number of properties that make them promising candidates for detection by LIGO and Virgo [16, 17, 18, 19, 20, 21, 22, 6]. As the binaries' components spiral into each other, they emit gravitational waves that pass through the sensitive band of the LIGO and Virgo detectors. These waves can be well-modeled using the post-Newtonian approximation to General Relativity [23, 24, 25, 26, 27, 28, 29, 30]. Knowing the morphology of the waveform provides a number of advantages, as it allows us to use match filtering and signal-based vetoes [31, 32]. The detectors have stationary, Gaussian noise that is intrinsic to the detector, as well as non-Gaussian transients (*glitches*) which come from environmental and instrumental sources. Match filtering is the optimal method to find weak signals in the stationary noise since the signal-to-noise ratio (SNR) grows with the square root of the number of cycles [1]. For non-Gaussian glitches, signal-based vetoes provide a powerful way to separate noise from candidates [32, 33]. Additionally, by utilizing all the detectors in the LIGO-Virgo network, we can perform "coincidence tests" to ensure that triggers from different observatories are consistent [34].

In this thesis we detail how searches for these "low mass" CBCs are carried out and we give some recent results from this search. In Chapter 2 we review the theory behind gravitational waves, showing how they arise in General Relativity, and review the basics of the LIGO/Virgo interferometers. In Chapter 3 we review some of the principles of matched filtering, and how various statistical tests are used to search the data. Next, in Chapter 4, we detail how false alarm rates are calculated for GW triggers. In Chapter 5 we step through the pipeline used to search for CBCs, which makes use of the methods presented in Chapters 3 and 4. Chapter 6 presents results from the low-mass CBC search for GWs in 6 months of S5 data. Next, in Chapter 7, we give a detailed examination of the low-mass CBC analysis of S6, including tuning choices made and preliminary results from that search. In Chapter 8 we present an analysis of the S5 and S6 data that shows that we can detect GWs at expected rates in Advanced LIGO. Finally, in Chapter 9 we present some future directions for the low-mass CBC pipeline.

Chapter 2

Physics of Gravitational Waves and the LIGO/Virgo Interferometers

In this chapter we briefly review the theory of gravitational waves: how they arise in General Relativity (GR) and how they can be detected by an interferometer. We then review the physics behind the LIGO and Virgo interferometers, including their main sources of noise. Finally, we examine potential sources of GW waves that can be detected by the LIGO and Virgo interferometers.

2.1   Gravitational Waves in General Relativity

In this section we show how gravitational waves arise in General Relativity. We begin our discussion with Einstein's field equations, given by[1] [35]:

$$G_{\alpha\beta} = 8\pi T_{\alpha\beta} \tag{2.1}$$

where $T_{\alpha\beta}$ is the stress-energy tensor and $G_{\alpha\beta}$ is the Einstein curvature tensor, defined as:

$$G_{\alpha\beta} \equiv R_{\alpha\beta} - \frac{1}{2} R g_{\alpha\beta} \tag{2.2}$$

Here, $g_{\alpha\beta}$ is the metric on some manifold and R is the Ricci curvature scalar, which is obtained by contracting $R_{\alpha\beta}$, which is the Ricci tensor. The Ricci tensor is obtained

---

[1]We will work in "natural units" in which $G = c = 1$, where G is Newton's gravitational constant and c is the speed of light.

by contracting the Riemann curvature tensor, $R^{\alpha}{}_{\beta\mu\nu}$, on the first and third indices. Riemann is defined as:

$$R^{\alpha}{}_{\beta\mu\nu} \equiv \Gamma^{\alpha}{}_{\beta\nu,\mu} - \Gamma^{\alpha}{}_{\beta\mu,\nu} + \Gamma^{\alpha}{}_{\sigma\mu}\Gamma^{\sigma}{}_{\beta\nu} - \Gamma^{\alpha}{}_{\sigma\nu}\Gamma^{\sigma}{}_{\beta\mu} \qquad (2.3)$$

where the $\Gamma^{\alpha}{}_{\mu\nu}$ are the Christoffel symbols, given by:

$$\Gamma^{\alpha}{}_{\mu\nu} = \frac{1}{2}g^{\alpha\beta}(g_{\beta\mu,\nu} + g_{\beta\nu,\mu} - g_{\mu\nu,\beta}) \qquad (2.4)$$

The "," indicates the partial derivative of the tensor with respect to the trailing indices. The Christoffel symbols describe how the components of a basis vector change when transported on a curved manifold [36]. The Riemann tensor describes how a vector changes when parallel transported around a closed loop on the manifold. It thereby describes the curvature of the manifold, in 20 independent components (in a four-dimensional spacetime). The Ricci tensor reduces the number of components to 10, and the Einstein tensor to 6, leaving 4 components of the metric free [37].[2] Thus the Einstein field equations relate the curvature of spacetime to mass and energy, which gives rise to the "force" we call gravity.

In order to study GWs we will work in the *weak field limit*, for which:

$$g_{\alpha\beta} = \eta_{\alpha\beta} + h_{\alpha\beta} \qquad (2.5)$$

Here, $\eta_{\alpha\beta}$ is the Minkowski metric (we use the $(-, +, +, +)$ convention) and $h_{\alpha\beta}$ is a small perturbation on this metric, i.e., $|h_{\alpha\beta}| \ll 1$. With this assumption, the Riemann tensor simplifies to [36]:

$$R_{\alpha\beta\mu\nu} = \frac{1}{2}(h_{\alpha\nu,\beta\mu} + h_{\beta\mu,\alpha\nu} - h_{\alpha\mu,\beta\nu} - h_{\beta\nu,\alpha\mu}) \qquad (2.6)$$

If we define $\bar{h}^{\alpha\beta}$ to be the "trace reverse"[3] of $h_{\alpha\beta}$:

$$\bar{h}_{\alpha\beta} = h_{\alpha\beta} - \frac{1}{2}\eta_{\alpha\beta}h \qquad (2.7)$$

(here $h = h^{\sigma}{}_{\sigma}$ is the trace of h) then, to first order, we can simplify the Einstein curvature tensor to:

$$G_{\alpha\beta} = -\frac{1}{2}(\bar{h}_{\alpha\beta,\mu}{}^{\mu} + \eta_{\alpha\beta}\bar{h}_{\mu\nu}{}^{,\mu\nu} - \bar{h}_{\alpha\mu,\beta}{}^{\mu} - \bar{h}_{\beta\mu,\alpha}{}^{\mu} + \mathcal{O}(h^2_{\alpha\beta})) \qquad (2.8)$$

---

[2]These four free components allow us to choose any coordinate system, or "gauge", we wish without changing the underlying physics.

[3]Note: in this case, the over-bar does not indicate an average.

We can exploit gauge freedoms in the Einstein equations to simplify this more. If we chose the Lorenz gauge,

$$\bar{h}^{\mu\nu}{}_{,\nu} = 0 \tag{2.9}$$

then:

$$G_{\alpha\beta} = -\frac{1}{2}\bar{h}_{\alpha\beta,\mu}{}^{\mu} \tag{2.10}$$

The weak-field Einstein equations are thus:

$$\Box\bar{h}_{\alpha\beta} = -16\pi T_{\alpha\beta} \tag{2.11}$$

where $\Box$ is the four-space d'Almbertian operator. This is the wave equation. Analogous to the wave equation in electromagnetism, equation 2.11 has the solution:

$$\bar{h}_{\alpha\beta} = 4\int \frac{T_{\alpha\beta}(t-|\vec{x}-\vec{x}'|,\vec{x}')}{|\vec{x}-\vec{x}'|}\,\mathrm{d}x'^3 \tag{2.12}$$

Equation 2.12 relates the metric perturbation to its source. We will return to this equation shortly. First, to explore properties of GW propagation, we consider the field equations at points outside the source, i.e., in vacuum, for which $T_{\alpha\beta} = 0$. The wave equation becomes:

$$\Box\bar{h}_{\alpha\beta} = 0 \tag{2.13}$$

which has the solution:

$$\bar{h}_{\alpha\beta} = \Re A_{\alpha\beta}e^{ik_\mu x^\mu} \tag{2.14}$$

Each $\alpha\beta$ component of $\bar{h}$ is thus a sinusoidal wave that travels along the vector k^μ with amplitude $A_{\alpha\beta}$. It can be shown [36] that $k_\mu k^\mu = 0$, i.e., k^μ is a null vector. Identifying k^0 as the angular frequency of the wave, ω, this implies that $|\vec{k}|^2 = \omega^2$. Since, in general, $|\vec{k}| = \omega/v$ (where v is the phase velocity of the wave), v (as well as the group velocity) must be 1. The wave therefore travels at the speed of light. Additionally, in the Lorenz gauge:

$$A_{\alpha\beta}k^\beta = 0 \tag{2.15}$$

This means that $A_{\alpha\beta}$ must be orthogonal to $\vec{k}$; i.e., the wave is a transverse-plane wave.

We can exploit additional gauge freedoms to further simplify the wave equations 2.14. It can be shown [36] that we can impose two further conditions:

$$A_\alpha{}^\alpha = 0 \tag{2.16}$$

$$A_{\alpha\beta}u^\beta = 0 \tag{2.17}$$

while still satisfying the Lorenz gauge condition. The first makes $\bar{h}_{\alpha\beta}$ trace free; thus in this gauge, which we denote by a TT superscript,

$$\bar{h}^{\mathrm{TT}}_{\alpha\beta} = h^{\mathrm{TT}}_{\alpha\beta} \tag{2.18}$$

The second condition sets the α component of A orthogonal to some constant four-velocity of our choosing. If we orient our background reference frame such that the four-velocity lies along the time basis vector, $u^\beta = \delta^\beta{}_0$, and such that the spatial part of the wave-vector points along the z-axis (i.e., $k = (\omega, 0, 0, \omega)$) then this condition implies that $A_{\alpha 0}$ and $A_{\alpha z}$ are 0 for all α. (The second result follows from equation 2.15.) Thus in this gauge, the metric perturbation is both transverse and traceless, hence the TT label.

By going to the TT gauge we have greatly simplified the solution to the wave equation. Since the metric perturbation is transverse, the only non-zero elements are h^{TT}_{xx}, h^{TT}_{yy}, and h^{TT}_{xy} $(= h^{\mathrm{TT}}_{yx})$. Since it is traceless, $h^{\mathrm{TT}}_{xx} = -h^{\mathrm{TT}}_{yy}$. Thus we have reduced solutions 2.14 to two independent equations:

$$h^{\mathrm{TT}}_{xx} = \Re h_+ e^{ik_\mu x^\mu} = h_+ \cos(\omega(t-z) + \phi_0) \tag{2.19}$$

$$h^{\mathrm{TT}}_{xy} = \Re h_\times e^{ik_\mu x^\mu} = h_\times \cos(\omega(t-z) + \phi_0) \tag{2.20}$$

where h_+ $(= A_{xx} = -A_{yy})$ and $h_\times$ $(= A_{xy} = A_{yx})$ are the "plus" and "cross" polarizations of the gravitational wave. (ϕ_0 is some arbitrary initial phase.)

Since we have used all of our gauge freedoms to obtain these two solutions, the results are physically observable. We can see this by considering the proper distance between two freely-falling particles, one located at the origin ($x_1^i = (0,0,0)$) and the other at $x_2^i = (\delta x, 0, 0)$ when a gravitational wave passes by. In Cartesian coordinates the line element in the TT gauge is:

$$\mathrm{d}s^2 = -\,\mathrm{d}t^2 + \mathrm{d}z^2 + (1 + h_+ \cos(\omega[t-z] + \phi_0))\,\mathrm{d}x^2 \tag{2.21}$$

$$+ (1 - h_+ \cos(\omega[t-z] + \phi_0))\,\mathrm{d}y^2 + 2h_\times \cos(\omega[t-z] + \phi_0)\,\mathrm{d}x\,\mathrm{d}y \tag{2.22}$$

To get a sense of the effect of the passing wave, consider the case when the wave only has plus polarization. Additionally, let the initial phase of the gravitational wave be $\phi_0 = -\pi/2$. The proper distance between the two particles at arbitrary time t is:

$$\Delta l = \int \mathrm{d}s = \int_0^{\delta x} \sqrt{(1 + h_+ \cos(\omega t - \pi/2))}\, \mathrm{d}x \tag{2.23}$$

$$\approx \int_0^{\delta x} (1 + \frac{1}{2} h_+ \cos(\omega t - \pi/2))\, \mathrm{d}x \tag{2.24}$$

$$= \delta x(1 + \frac{1}{2} h_+ \cos(\omega t - \pi/2)) \tag{2.25}$$

In the second step we have used the fact that $|h_+| \ll 1$. At time $t = 0$, we get $\Delta l = \delta x$, as expected. At time $t = \pi/2\omega$, however, we get $\Delta l = \delta x(1 + h_+/2)$: the particle has moved from the origin by an additional $h_+/2$. If we perform the same analysis for another particle situated at $x_3^i = (0, \delta y, 0)$, we will get $\Delta l = \delta y(1 - h_+/2)$ at time $t = \pi/2\omega$. Likewise, if we consider particles situated at $x^i = (\delta x, \delta y, 0)$ and $x^i = (-\delta x, \delta y, 0)$ we will get similar oscillations when a cross-polarized wave passes through. Figure 1 shows what this would look like for a ring of particles. Thus, we can detect a gravitational wave by measuring how the proper distance changes between two freely-falling masses. How this is done is explained in section 2.2.

Return now to the source-dependent wave solution, equation 2.12. Since the TT gauge captures all of the physically measurable quantities of the gravitational wave outside the source, we can simplify the equation to:

$$h_{ij}^{\mathrm{TT}} = 4 \left[\int \frac{T_{ij}(t - |\vec{x} - \vec{x}'|, \vec{x}')}{|\vec{x} - \vec{x}'|}\, \mathrm{d}x'^3 \right]^{\mathrm{TT}} \tag{2.26}$$

In other words, we can calculate the measurable gravitational wave created by a source from the ij components of the stress-energy tensor alone. We then throw out the transverse parts and remove the trace. We will use this result in the next section to find a solution for the gravitational waveform from a compact binary system.

2.1.1 Gravitational Waves from a Compact Binary Inspiral

We now consider the gravitational radiation emitted from a system consisting of two compact spherical objects orbiting around each other. Let the mass of the objects be denoted by m_1 and m_2, and the total mass by M. Consider their motion when they are at a distance $a \gg 2M$. We will assume their motion is determined entirely
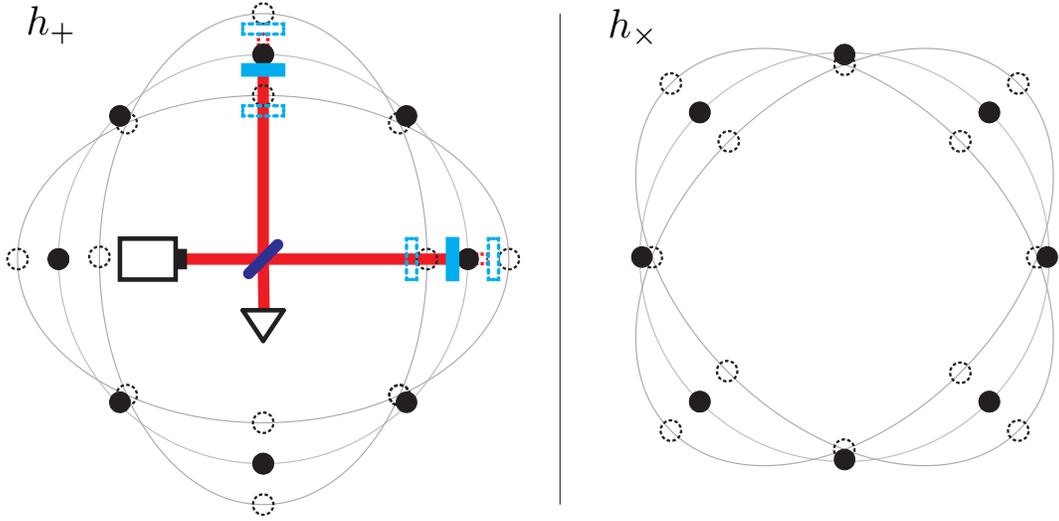
Figure 1 : A diagram showing the effect of a passing gravitational wave on a ring of particles. The left diagram shows the effect of a + polarized wave and the right shows the effect of × polarized wave. On the h_+ diagram we have attached a mirror to the $+x$ and $+y$ particles. These form the end-test masses of the interferometer shown. For a discussion of how the interferometer works, see section 2.2.

by their mutual gravitational attraction and that $m_1 \sim m_2 \sim M_\odot$. Although we will find that the orbit of the objects shrinks due to the energy loss from emitted GWs, with these assumptions, $\dot{a} \ll v$, where v is their relative orbital velocities. Thus we can assume quasi-stationary circular orbits. Additionally, since we have assumed $a \gg 2M$, $v \ll 1$; in this case, relativistic effects become negligible and we can use Newtonian mechanics and Kepler's third law — given by:

$$\Omega = \sqrt{\frac{M}{a^3}} \tag{2.27}$$

where Ω is the orbital angular velocity — to analyze the system.

Our assumptions also allow us to simplify equation 2.26. First, from energy and momentum conservation ($T^{\mu\nu}{}_{,\nu} = 0$) we have:

$$T^{00}{}_{,0} = -T^{0j}{}_{,j}$$
$$T^{j0}{}_{,0} = -T^{jk}{}_{,k}$$

Therefore:

$$2T^{jk} = T^{00}{}_{,00} x^j x^k - (T^{lm} x^j x^k)_{,ml} + 2(T^{lj} x^k + T^{lk} x^j)_{,l}$$

When we plug this into equation 2.26^4 we find that the last two terms are zero due to Stokes theorem. The T^{00} term gives the mass-energy of the source, ρ. We therefore have:

$$h_{ij}^{\text{TT}} = 2 \left[\partial_t^2 \int \frac{\rho(t - |\vec{x} - \vec{x}'|, \vec{x}')}{|\vec{x} - \vec{x}'|} \, \mathrm{d}^3 x' \right]^{\text{TT}} \tag{2.28}$$

Now, because we are far from the source, $|\vec{x} - \vec{x}'| \sim r$ (where r is the distance to the source), i.e., $|\vec{x}'|/|\vec{x}| \ll 1$. As is done in EM, we can therefore expand the denominator [37]:

$$\frac{1}{|\vec{x} - \vec{x}'|} = \frac{1}{r} + \frac{x^j x^{j'}}{r^3} + \frac{x^j x^k (3 x^{j'} x^{k'} - r'^2 \delta_{jk})}{2 r^5} + \dots \tag{2.29}$$

Keeping the first term yields:

$$\frac{2}{r} \partial_t^2 \int \rho(t - r) x'^j x'^k \, \mathrm{d}^3 x'$$

We recognize the integral as the second-time derivative of the second moment of the mass distribution, $\ddot{I}^{jk}$.[5] Since we eventually remove the trace, we can instead use the mass quadrupole moment, $\mathcal{I}_{jk}$, defined as:

$$\mathcal{I}_{jk} = \int \rho(\mathbf{x}) \left(x_j x_k - \frac{1}{3} \delta_{jk} \delta_{mn} x^m x^n \right) \mathrm{d}^3 x \tag{2.30}$$

Substituting this into equation 2.28 yields:

$$h_{jk}^{\text{TT}} = \frac{2}{r} \ddot{\mathcal{I}}_{jk}^{\text{TT}}(t - r) \tag{2.31}$$

This is an important result: we only get gravitational waves if we have a changing quadrupole (or higher-order) moment. Contrast this with EM, in which we only need a changing dipole moment to get electromagnetic waves.

Returning to our problem, to evaluate the amplitude of the GW emitted by the binary we must first evaluate $\mathcal{I}_{jk}$. Since we only have two masses, we can use the discrete form of the integral to directly write:

$$\mathcal{I}_{ij} = m_1 \left(x_{1i} x_{1j} - \delta_{ij} \frac{1}{3} r_1^2 \right) + m_2 \left(x_{2i} x_{2j} - \delta_{ij} \frac{1}{3} r_2^2 \right) \tag{2.32}$$

---

[4]We use the background metric $\eta_{\alpha\beta}$ to lower the ij indices on T.

[5]In general, we will use over-dots to indicate differentiation with respect to proper time, τ. However, because we are working in a Minkowski spacetime, $\tau \to t$. We can therefore use the dots here to indicate differentiation with respect to the coordinate time.

where $r_n^2 = x_n^2 + y_n^2$ is the radial position of the nth mass. Working in the center-of-mass frame of the binary, we have:

$$x_1 = r_1 \cos \Omega t$$

$$y_1 = r_1 \sin \Omega t$$

$$x_2 = -r_2 \cos \Omega t$$

$$y_2 = -r_2 \sin \Omega t$$

where

$$r_1 = a \frac{m_2}{m_1 + m_2} \tag{2.33}$$

$$r_2 = a \frac{m_1}{m_1 + m_2} \tag{2.34}$$

Substituting these values into equation 2.32 we have:

$$\begin{aligned}
\mathcal{I}_{xx} &= m_1 x_1^2 + m_2 x_2^2 - \frac{1}{3}(m_1 r_1^2 + m_2 r_2^2) \\
&= (m_1 r_1^2 + m_2 r_2^2)\left(\cos^2 \Omega t - \frac{1}{3}\right) \\
&= a^2 \left(\frac{m_1 m_2^2 + m_2 m_1^2}{m_1 + m_2}\right)\left(\cos^2 \Omega t - \frac{1}{3}\right) \\
&= \mu a^2 \left(\cos^2 \Omega t - \frac{1}{3}\right) \tag{2.35}
\end{aligned}$$

where:

$$\mu = \frac{m_1 m_2}{M} \tag{2.36}$$

is the reduced mass. By a similar calculation, we get:

$$\mathcal{I}_{yy} = \mu a^2 \left(\sin^2 \Omega t - \frac{1}{3}\right) \tag{2.37}$$

$$= -\mathcal{I}_{xx} \tag{2.38}$$

$$\mathcal{I}_{zz} = -\frac{1}{3}\mu a^2 \tag{2.39}$$

$$\mathcal{I}_{xy} = \mu a^2 \cos \Omega t \sin \Omega t \tag{2.40}$$

$$= \mathcal{I}_{yx} \tag{2.41}$$

The second time derivative of the quadrupole moment is thus:

$$\ddot{\mathcal{I}}_{xx} = -2\mu a^2 \Omega^2 \cos(2\Omega t) \tag{2.42a}$$

$$\ddot{\mathcal{I}}_{yy} = 2\mu a^2 \Omega^2 \cos(2\Omega t) \tag{2.42b}$$

$$\ddot{\mathcal{I}}_{xy} = -2\mu a^2 \Omega^2 \sin(2\Omega t) \tag{2.42c}$$

$$= \ddot{\mathcal{I}}_{yx} \tag{2.42d}$$

$$\ddot{\mathcal{I}}_{zz} = 0 \tag{2.42e}$$

In order to plug equations 2.42 into equation 2.31 we need the transverse part of the moments. We know that the gravitational waves will propagate outward from the binary. Therefore, if we project equations 2.42 into the spherical coordinates $\{r, \iota, \phi\}$ shown in figure 2, then $\mathcal{I}_{\iota\iota}$, $\mathcal{I}_{\phi\phi}$, and $\mathcal{I}_{\iota\phi}$ $(= \mathcal{I}_{\phi\iota})$ will form the parts of the tensor transverse to the propagation. Since the z components of the time derivative of the quadrupole are all zero, the transformation law is simplified to:

$$\ddot{\mathcal{I}}_{i'j'} = \Lambda_{i'}{}^i \Lambda_{j'}{}^j \ddot{\mathcal{I}}_{ij}$$
$$= \Lambda_{i'}{}^x \Lambda_{j'}{}^x \ddot{\mathcal{I}}_{xx} + \Lambda_{i'}{}^x \Lambda_{j'}{}^y \ddot{\mathcal{I}}_{xy} + \Lambda_{i'}{}^y \Lambda_{j'}{}^x \ddot{\mathcal{I}}_{yx} + \Lambda_{i'}{}^y \Lambda_{j'}{}^y \ddot{\mathcal{I}}_{yy}$$

We can further simplify the problem by setting the initial orbital phase, ϕ, to 0. Since we have assumed quasi-stationary circular orbits, we can later account for arbitrary initial phase by letting $\Omega t \rightarrow \Omega t + \phi$. Doing so simplifies the transformation matrix to:

$$\Lambda = \begin{pmatrix} \sin\iota & 0 & \cos\iota \\ \cos\iota & 0 & -\sin\iota \\ 0 & 1 & 0 \end{pmatrix}$$

Resulting in:

$$\ddot{\mathcal{I}}_{\iota\iota} = \ddot{\mathcal{I}}_{xx} \cos^2\iota$$
$$\ddot{\mathcal{I}}_{\phi\phi} = \ddot{\mathcal{I}}_{yy}$$
$$\ddot{\mathcal{I}}_{\iota\phi} = \ddot{\mathcal{I}}_{xy} \cos\iota$$
$$= \ddot{\mathcal{I}}_{\phi\iota}$$

To make the tensor traceless in this coordinate basis, we set:

$$\ddot{\mathcal{I}}_{\iota\iota}^{TT} = -\ddot{\mathcal{I}}_{\phi\phi}^{TT} = 1/2(\ddot{\mathcal{I}}_{\iota\iota} - \ddot{\mathcal{I}}_{\phi\phi})$$

Letting $\Omega t \to \Omega t - \phi$, we have:

$$\ddot{\mathcal{I}}_{\iota\iota}^{\mathrm{TT}} = \mu a^2 \Omega^2 \cos(2[\Omega t - \phi])(1 + \cos^2 \iota) \tag{2.43}$$

$$\ddot{\mathcal{I}}_{\iota\phi}^{\mathrm{TT}} = -2\mu a^2 \Omega^2 \sin(2[\Omega t - \phi]) \cos \iota \tag{2.44}$$

Substituting this into equation 2.31, and using Kepler's law to put a in terms of M and Ω ($a = (M/\Omega^2)^{1/3}$), we have:

$$h_{\iota\iota}^{\mathrm{TT}} = \frac{2}{r}\mu(M\Omega)^{2/3} \cos(2[\Omega(t - r) - \phi])(1 + \cos^2 \iota) = h_+ \tag{2.45a}$$

$$h_{\iota\phi}^{\mathrm{TT}} = -\frac{4}{r}\mu(M\Omega)^{2/3} \sin(2[\Omega(t - r) - \phi]) \cos \iota = h_\times \tag{2.45b}$$

Here we have identified the $\iota\iota$ and $\iota\phi$ parts of the GW field as the "plus" and "cross" polarizations by using the basis vectors [37]:

$$\mathbf{e}^+ = (\vec{e}_\iota \otimes \vec{e}_\iota - \vec{e}_\phi \otimes \vec{e}_\phi), \ \mathbf{e}^\times = (\vec{e}_\iota \otimes \vec{e}_\phi + \vec{e}_\phi \otimes \vec{e}_\iota) \tag{2.46}$$

In equations 2.45 we can see that the gravitational-wave frequency is *twice* the orbital frequency:

$$f_{\mathrm{GW}} = 2f_{\mathrm{orbit}} = \frac{\Omega}{\pi} \tag{2.47}$$

The inclination angle, ι, and the distance to the center of mass, r, are *extrinsic* parameters of the binary: they depends only on the orientation of the binary with respect to an outside observer. To separate the part of the waveform that depends on the *intrinsic* parameters of the binary — i.e., the physical parameters of the binary that are independent of the location of the observer[6] — we define the *sine chirp*, h_s, and the *cosine chirp*, h_c [31]:

$$h_c \equiv 2\mu(M\Omega^{2/3}) \cos(2[\Omega t - \phi_0]) \tag{2.48a}$$

$$h_s \equiv 2\mu(M\Omega^{2/3}) \sin(2[\Omega t - \phi_0]) \tag{2.48b}$$

such that:

$$h_+ = \frac{h_c}{r}(1 + \cos^2 \iota) \tag{2.49a}$$

$$h_\times = \frac{h_s}{r} \cos \iota \tag{2.49b}$$

(The reason for the name "chirp" will become clear below.) Here, ϕ_0 is the phase of the binary at some initial point in time.

---

[6]When we say "independent of the location of the observer" we actually mean "independent of the location of any observers that are in the same local Lorentz frame (LLF)."
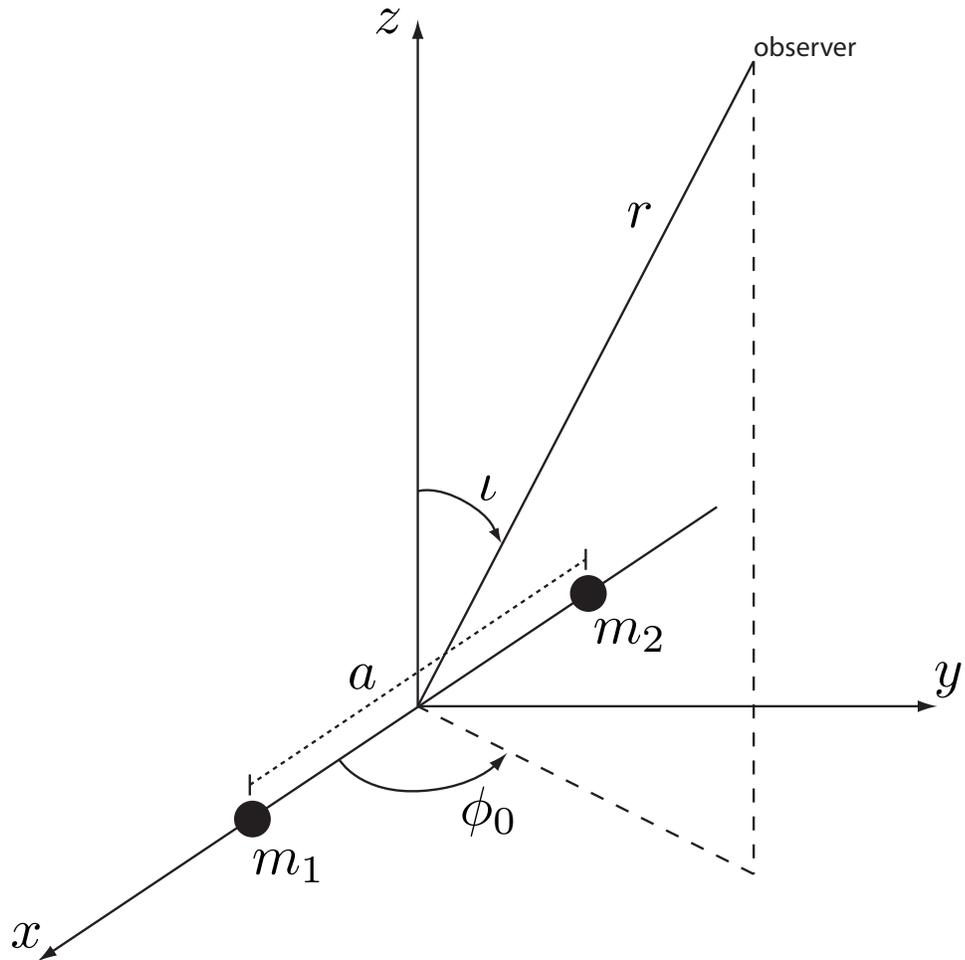
Figure 2 : The center-of-mass frame of a binary with separation distance a. The coordinates $\{r, \iota, \phi\}$ give, respectively, the distance to an observer, the inclination angle of the observer, and the phase of the binary. ϕ_0 is the initial phase. Figure originally published in [31].

2.1.2 Evolution of the Gravitational Waveform in Newtonian Physics

In Chapter 3 we will find that we use the sine and cosine waveforms to *match-filter* data from GW detectors. Match filtering offers the optimal way to find a CBC GW signal that is buried in noise. However, to perform the match filter, we need to generate a time or frequency series for the waveform. From equations 2.48 we see that this requires knowledge of how the orbital phase, Ωt, (or, via equation 2.47, the gravitational-wave phase) evolves with time (or frequency). We can find this by considering the energy lost by the binary due to gravitational radiation.

Using equation 2.31 it can be shown that the energy loss from GW emission is given by [37]:

$$\frac{\mathrm{d}E}{\mathrm{d}t} = -\frac{1}{5}\left\langle \dddot{\mathcal{I}}_{jk}\dddot{\mathcal{I}}^{jk}\right\rangle \tag{2.50}$$

where the brackets $<\cdot>$ denote an average over time. Using equations 2.42, we find:

$$\dddot{\mathcal{I}}_{xx} = 4\mu a\Omega^3 \sin(2\Omega t) = -\dddot{\mathcal{I}}_{yy}$$

$$\dddot{\mathcal{I}}_{xy} = -4\mu a^2\Omega^3 \cos(2\Omega t) = \dddot{\mathcal{I}}_{yx}$$

This gives:

$$\begin{aligned}
\frac{\mathrm{d}E}{\mathrm{d}t} &= -\frac{2}{5}\left\langle \dddot{\mathcal{I}}_{xx}^2 + \dddot{\mathcal{I}}_{xy}^2\right\rangle \\
&= -\frac{32}{5}\mu^2 a^4\Omega^6\left\langle \sin^2(2\Omega t) + \cos^2(2\Omega t)\right\rangle \\
&= -\frac{32}{5}\frac{\mu^2 M^3}{a^5}
\end{aligned}$$

In the last step we have again used Kepler's law to put Ω in terms of M and a. We can use this equation to find how the separation distance changes with time. Using the chain rule we have:

$$\frac{\mathrm{d}E}{\mathrm{d}t} = \frac{\partial E}{\partial a}\frac{\mathrm{d}a}{\mathrm{d}t} \tag{2.51}$$

From Newton's law of gravity and conservation of energy, the binary's energy is related to the separation distance by:

$$\frac{\partial E}{\partial a} = \frac{\mu M}{2a^2}$$

Substituting this into equation 2.51 yields:

$$\frac{1}{2}a^3\,\mathrm{d}a = -\frac{32}{5}\mu M^2\,\mathrm{d}t$$

Integrating this from some initial separation, a_0, at time t_0, to the separation, a, at some arbitrary time t gives:

$$a = a_0 \left(1 - \frac{t - t_0}{\tau_0} \right)^{1/4} \tag{2.52}$$

where:

$$\tau_0 = \frac{5}{256} \frac{a_0^4}{\mu M^2} \tag{2.53}$$

is the *time to coalescence*. That is, τ_0 is the amount of time it takes for the binary to go from the initial separation a_0 to $a = 0$. In equation 2.52 we can see that the orbital separation decreases with time due to the loss in energy. In other words, the two masses "inspiral" into each other.

We can use equations 2.52 and 2.53, and Kepler's third law to find how the orbital angular velocity changes with time. Re-arranging equation 2.27 to $a = (M/\Omega^2)^{1/3}$ and manipulating equation 2.53 to get a_0 in terms of τ_0, we have, from equation 2.52:

$$M^{4/3} \Omega^{-8/3} = \frac{256}{5} \mu M^2 (\tau_0 - t)$$

Here, we have set the initial time $t_0 = 0$. Rearranging, we have:

$$\Omega = \left[\frac{5}{256} \frac{1}{\eta M^{5/3} (\tau_0 - t)} \right]^{3/8} \tag{2.54}$$

where:

$$\eta \equiv \frac{\mu}{M} \tag{2.55}$$

is the *symmetric mass-ratio*. Using equation 2.47 to relate the gravitational-wave frequency to Ω, we find:

$$f_{\mathrm{GW}}(t) \propto (\tau_0 - t)^{-3/8} \tag{2.56}$$

Figure 3(a) shows what the gravitational-wave frequency looks like as a function of time. We see that the wave sweeps upward in frequency in a "chirp" pattern, and that it asymptotes at infinity as $a \to 0$.

We can also use equation 2.54 to find how the amplitude of the waveform changes over time. In equations 2.48 we see that the amplitude of the waveform is proportional to $M\Omega^{2/3}$. Thus:

$$|h_c| = |h_s| = 2 \left(\frac{5}{256} \frac{\mathcal{M}}{\tau_0 - t} \right)^{1/4} \tag{2.57}$$

where we have defined the *chirp mass*, $\mathcal{M}$, as:

$$\mathcal{M} \equiv \eta^{3/5} M \tag{2.58}$$

Thus the amplitude also increases with time, and it asymptotes at infinity as $a \to 0$. Figure 3(b) shows what the waveform evolution looks like for an equal mass binary ($\eta = 1/4$).

2.1.3 Orbital Dynamics in a Schwarzschild Spacetime

In the above analysis we assumed the relative velocities of the binary's component masses were small compared to the speed of light, and that the spacetime curvature was relatively flat. This allowed us to use Newtonian and Keplerian dynamics to analyze the binary. While this worked well when the masses were far apart from each other, it breaks down as the separation distance gets small. This was evident in the fact that the orbital angular velocity asymptotes at infinity as $a \to 0$. This means that the component masses' relative tangential velocity:

$$v = (M\Omega)^{1/3} \tag{2.59}$$

would also approach infinity. Clearly there is a separation distance for which our assumptions fail, and we must take into account relativistic effects.

To illustrate relativistic effects, we consider a test-particle with mass m orbiting a massive black hole with mass M, such that $m \ll M$. For simplicity we will assume that both the test mass and the black hole have negligible spin. Thus, we can describe the spacetime using the Schwarzschild metric [35]:

$$\mathrm{d}s^2 = -(1 - 2M/r)\,\mathrm{d}t^2 + \frac{\mathrm{d}r^2}{1 - 2M/r} + r^2\,\mathrm{d}\phi^2 + r\sin\iota\,\mathrm{d}\iota^2 \tag{2.60}$$

The coordinates are chosen such that $r = 0$ is at the center-of-mass of the system, which — since $M \gg m$ — is roughly at the "center" of the black hole. Without loss in generality we will assume the inclination angle is equal to $\pi/2$, so that the $\mathrm{d}\iota$ part of the line element is 0.[7] In this case the metric only has three elements:

$$g_{tt} = -(1 - 2M/r) \tag{2.61a}$$

$$g_{rr} = (1 - 2M/r)^{-1} \tag{2.61b}$$

$$g_{\phi\phi} = r^2 \tag{2.61c}$$

---

[7] For a treatment of this problem with $\iota \neq \pi/2$ see Box 25.4 of [35].
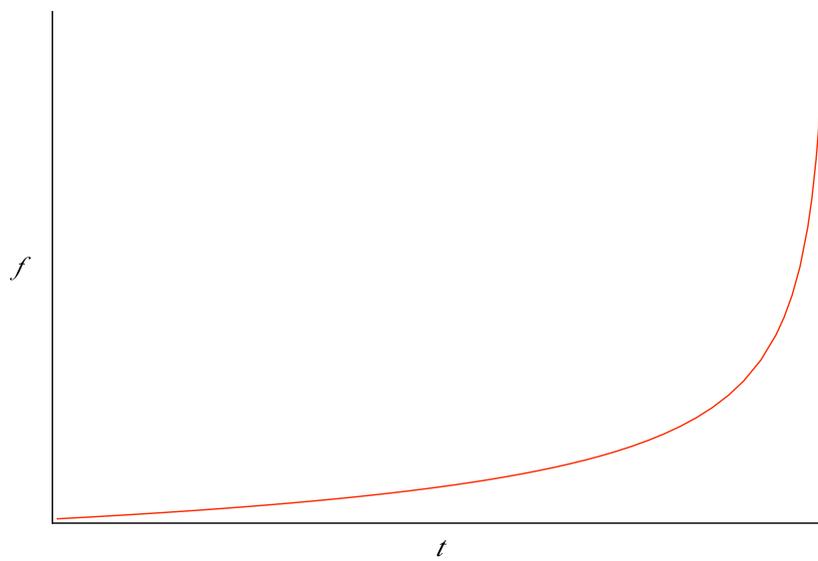
(a) $f(t)$
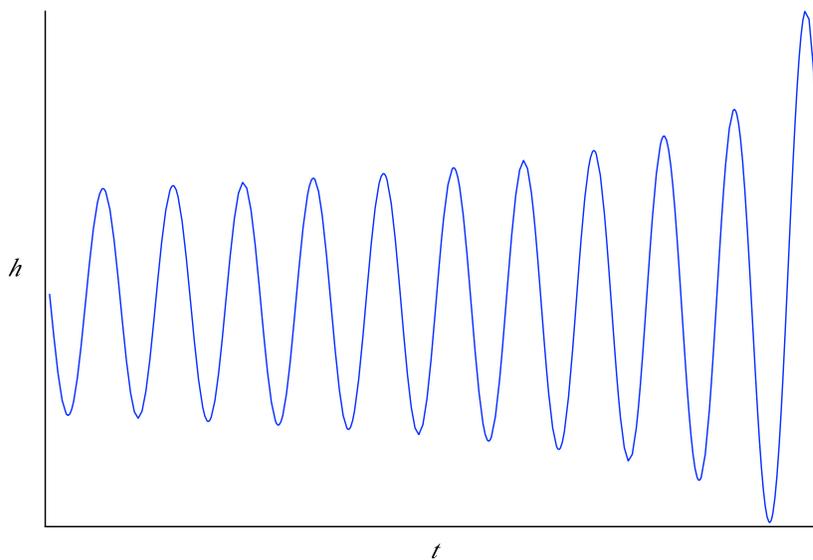
(b) $h(t)$

Figure 3 : The frequency evolution and waveform evolution of an equal mass binary using equations 2.56 and 2.57.

In the last section the equation of motion for the masses was given by Kepler's third law, which related the orbital angular velocity to the mass and separation distance. To analyze the test particle's orbital evolution we again need an equation of motion. However, since we are working in a curved spacetime it is not clear that Kepler's third law still holds. To find the relativistic equation of motion, we consider the particle's geodesic path as it orbits the black hole. Since no external forces act on the particle, the geodesic equation is [35, 36, 38]:

$$\frac{\mathrm{d}^2 x^\alpha}{\mathrm{d}\tau^2} + \Gamma^\alpha{}_{\mu\nu}\frac{\mathrm{d}x^\mu}{\mathrm{d}\tau}\frac{\mathrm{d}x^\nu}{\mathrm{d}\tau} = 0 \qquad (2.62)$$

where the Christoffel symbol is:

$$\begin{aligned}
\Gamma^\alpha{}_{\mu\nu} &= g^{\alpha\sigma}\Gamma_{\sigma\mu\nu} \\
&= \frac{1}{2}g^{\alpha\sigma}(g_{\sigma\mu,\nu} + g_{\sigma\nu,\mu} - g_{\mu\nu,\sigma}) \\
&= \frac{1}{2}g^{\alpha\alpha}(g_{\alpha\mu,\nu} + g_{\alpha\nu,\mu} - g_{\mu\nu,\alpha}) \qquad (2.63)
\end{aligned}$$

In the last step we have used the fact that there are no mixing terms in the metric.

Before analyzing these equations, we note a few properties of the metric which greatly simplify the problem. The metric has no t, ϕ, or (since we set the particle's inclination angle to $\pi/2$) ι dependence. Thus the t, ϕ, and ι components of the particle's conjugate four-momentum, p_α are conserved [35, 36, 38]. The particle's four-momentum is given by:

$$p^\alpha = mu^\alpha = m\frac{\mathrm{d}x^\alpha}{\mathrm{d}\tau} \qquad (2.64)$$

where u^α is the particle's four-velocity. The four-momentum is related to the conjugate momentum by: $p^\alpha = g^{\alpha\beta}p_\alpha$. Since p_α has no τ dependence, and since $g^{\alpha\beta}$ only depends on r, $\dot{p}^\alpha \sim \dot{r}$. (Dots over the coordinates indicate differentiation with respect to the proper time.) We shall assume, as we did in the previous section, that $\dot{r} \ll \dot{\phi}$, i.e., that the particle follows quasi-stationary orbits. In this case, $\dot{r} \approx 0$, as well as $\ddot{r}$. Thus:

$$\dot{p}^t = \ddot{t}/m = \dot{p}^\phi = \ddot{\phi}/m = \dot{p}^\iota = \ddot{\iota}/m = 0 \qquad (2.65)$$

Since we have fixed the inclination angle to $\pi/2$, $\ddot{\iota}$ is also zero. From equation 2.63, this means that $\Gamma^\iota{}_{\mu\nu}$ is zero, and so the ι equation of motion gives the trivial result

$0 = 0$. Thus we only have three equations to consider:

$$\Gamma^{\alpha}{}_{tt}\dot{t}^2 + 2\Gamma^{\alpha}{}_{t\phi}\dot{t}\dot{\phi} + \Gamma^{\alpha}{}_{\phi\phi}\dot{\phi}^2 = 0 \tag{2.66}$$

where $\alpha = r, t, \phi$. We further simplify these equations by noting:

$$\Gamma^{\alpha}{}_{tt} = \frac{1}{2}g^{\alpha\alpha}(2g_{\alpha t,t} - g_{tt,\alpha}) = -\frac{1}{2}g^{\alpha\alpha}g_{tt,\alpha}$$

$$\Gamma^{\alpha}{}_{\phi\phi} = \frac{1}{2}g^{\alpha\alpha}(2g_{\alpha\phi,\phi} - g_{\phi\phi,\alpha}) = -\frac{1}{2}g^{\alpha\alpha}g_{\phi\phi,\alpha}$$

$$\Gamma^{\alpha}{}_{t\phi} = \frac{1}{2}g^{\alpha\alpha}(g_{\alpha t,\phi} + g_{\alpha\phi,t} - g_{t\phi,\alpha}) = 0$$

Since $g_{tt,\alpha}$ and $g_{\phi\phi,\alpha}$ are only non-zero if $\alpha = r$, the only non-zero Christoffel symbols are $\Gamma^{r}{}_{tt}$ and $\Gamma^{r}{}_{\phi\phi}$. This means the only non-trivial equation of motion is the one for which $\alpha = r$. The non-zero Christoffel symbols are:

$$\Gamma^{r}{}_{tt} = -\frac{1}{2}g^{rr}g_{tt,r} = \frac{M}{r^2}(1 - 2M/r) \tag{2.67a}$$

$$\Gamma^{r}{}_{\phi\phi} = -\frac{1}{2}g^{rr}g_{\phi\phi,r} = -r(1 - 2M/r) \tag{2.67b}$$

Substituting these into the equation of motion yields:

$$\frac{M}{r^2}(1 - 2M/r)\dot{t}^2 - r(1 - 2M/r)\dot{\phi}^2 = 0$$

$$\Rightarrow \frac{\dot{\phi}^2}{\dot{t}^2} = \frac{M}{r^3}$$

Note that:

$$\frac{\dot{\phi}}{\dot{t}} = \frac{\mathrm{d}\phi/\mathrm{d}\tau}{\mathrm{d}t/\mathrm{d}\tau} = \frac{\mathrm{d}\phi}{\mathrm{d}t} = \Omega \tag{2.68}$$

where Ω is the angular velocity of the test particle as measured by an observer in the particle's LLF. Thus we have:

$$\Omega = \sqrt{\frac{M}{r^3}} \tag{2.69}$$

Evidently Kepler's third law is valid in the Schwarzschild metric for circular orbits.

As noted above, $\dot{\phi} = p^{\phi}/m$ and $\dot{t} = p^t/m$. We can therefore write:

$$\frac{p^{\phi}}{p^t} = \sqrt{\frac{M}{r^3}} \tag{2.70}$$

Now, since p_t and p_ϕ are conserved, we can set them equal to the constants $-E$ and L, respectively. We can identify these constants as the particle's energy and momentum as seen by an observer at $r = \infty$ [36]. Raising the indices, we have:

$$p^t = g^{tt}p_t = \frac{E}{1 - 2M/r} \tag{2.71}$$

$$p^\phi = g^{\phi\phi}p_\phi = \frac{L}{r^2} \tag{2.72}$$

Equation 2.70 is therefore:

$$\frac{(1 - 2M/r)L}{r^2 E} = \sqrt{\frac{M}{r^3}} \tag{2.73}$$

We can put L in terms of E by noting:

$$p^\alpha p^\beta g_{\alpha\beta} = p^\alpha p_\alpha = m^2 u^\alpha u_\alpha = -m^2$$

Since p^r and p^t are both zero,

$$p^\alpha p_\alpha = p^t p_t + p^\phi p_\phi = -\frac{E^2}{1 - 2M/r} + \frac{L^2}{r^2} = -m^2$$

Rearranging this gives:

$$L = r\sqrt{\frac{E^2}{1 - 2M/r} - m^2} \tag{2.74}$$

and using equation 2.73 we obtain:

$$E = m\frac{1 - 2M/r}{\sqrt{1 - 3M/r}} \tag{2.75}$$

We have arrived at a solution for the particle's energy as a function of its orbital distance. (Note that as $r \to \infty$, $E \to m$, meaning E becomes the rest-mass energy of the particle, as expected.)

Figure 4 shows a plot of E/m as a function of r/M. From this (or by taking the derivative of equation 2.75 and setting it equal to zero) we can see that E/m decreases monotonically from infinity, until it reaches a minimum at $r = 6M$. Keep in mind that we obtained this graph by considering quasi-stationary circular orbits. Thus the plot shows how much energy is needed at each r for the particle to maintain a circular orbit. This means that at radii $\geq 6M$ the particle can sustain stable circular orbits without needing any additional energy. Below $r = 6M$, however, energy must

be added to the particle in order for it to sustain its orbit.[8] Assuming the system evolves adiabatically (that is, no external energy is added), after the particle passes $r = 6M$ it will cease to inspiral and will plunge into the black hole. This part of the evolution is known as "merger."

The point $r = 6M$ is known as the inner-most stable circular orbit (ISCO). At this point our analysis, both in this section and the last, breaks down since we can no longer assume $\dot{r} \ll \dot{\phi}$. The phase and amplitude evolution of the gravitational wave emitted after this point must therefore be evaluated using more sophisticated techniques. Although we found this radius by considering a Schwarzschild metric, an ISCO will exist for equal-mass systems as well (it may not occur at $r = 6M$, however).

Many of the systems we will consider in this thesis (namely, binary neutron stars; see the next section) have masses low enough that the frequencies of their emitted GWs are of order a few kHz when they pass ISCO. Since our gravitational-wave detectors have low sensitivity at these frequencies (see section 2.2.2) we can simply terminate the matched filter at f_{isco}, which is the frequency of the gravitational wave when the source's component masses pass ISCO. This permits us to use the techniques outlined in the last section and the next to generate a GW waveform. Higher mass systems (binary black holes and binaries with a neutron start and a black hole) cross ISCO in the more sensitive parts of the detectors' bands. We still search for such systems with post-Newtonian (pN) templates, however, as we can still recover much of the signal. For still higher mass systems $M_{\mathrm{total}} > 25M_\odot$ we use more sophisticated techniques to generate the waveform. Searches for such systems, and how templates are generated for them, are beyond the scope of this thesis, however.

2.1.4 The Post-Newtonian Approximation

Let us recap the steps we took in section 2.1.2 to arrive at a solution for the gravitational waveform:

- First, we performed a multipolar expansion on the right hand side of equation 2.26 to get h_{ij}^{TT} in terms of increasing moments of the mass-energy density. This

---

[8]And below $r = 2M$ an infinite amount of energy would be needed for the particle to sustain a stable orbit. Thus $r = 2M$ is the event horizon of the black hole, also known as the Schwarzschild radius.
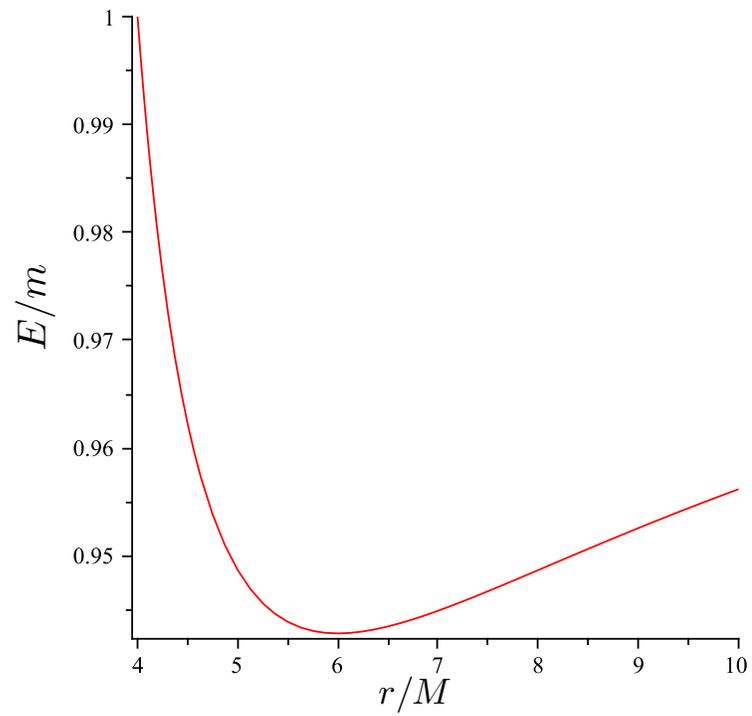
Figure 4 : A plot of the energy-per-unit mass of a test particle orbiting a Schwarzschild black hole as a function of its radial coordinate per the black hole's mass. The plot was created in Maple using equation 2.75.

eventually allowed us to solve the *energy-balance equation*:

$$\frac{\mathrm{d}E}{\mathrm{d}t} = -\mathcal{F} \tag{2.76}$$

Since we only kept the first (most-dominant) term of the expansion, we had $\mathcal{F} = \left\langle \dddot{\mathcal{I}}_{jk} \dddot{\mathcal{I}}^{jk} \right\rangle / 5$.

- Next, we found an expression for E in terms of the mass and the orbital separation. In section 2.1.2 we used Newton's law of gravity to find E. In section 2.1.3 we were a bit more sophisticated, and we obtained E by using the Schwarzschild metric.

- Using our expression for E in the energy balance equation, we were able to find how the phase and amplitude of the GW evolved with time.

If we keep higher order terms in the multipolar expression of h, then carried out the same steps, we would obtain a more accurate expression for the waveform. However, by performing the multipolar expansion after taking the transverse-traceless part of h, we have missed some terms. If we had performed the multipolar expansion and evaluated the integral *before* taking the transverse-traceless part, we would have found that the h_{0j} terms yield corrections to the gravitational wave from moments of the momentum density. In general, we would get [37]:

$$|h_+| \sim |h_\times| \sim \frac{\ddot{\mathcal{I}}_2}{r} \& \frac{\dddot{\mathcal{I}}_3}{r} \& \dots \& \frac{\ddot{\mathcal{S}}_2}{r} \& \frac{\dddot{\mathcal{S}}_3}{r} \& \dots \tag{2.77}$$

Here, & indicates addition with terms, and:

$$\mathcal{I}_l \sim ML^l, \quad \mathcal{S}_l \sim MvL^l \tag{2.78}$$

are the mass and momentum moments, respectively, of order l. L is the size of the source and v is the relative velocities of the objects. Since the number of time derivatives grows with each of these terms, we can see that each term grows as $\partial_t^{(2-l)} L \sim v^{(2-l)}$. Hence the effects from each term becomes more relevant as the relative velocity of the objects grows.[9]

---

[9]Since $\mathcal{S} \sim Mv$, its effect on h is one order lower from the mass-moment. In other words, $|\mathcal{S}_l| \sim |\mathcal{I}_{l-1}|$. For example, the mass quadrupole ($l = 2$) goes as v^0, where as the momentum quadrupole goes as v^1. Due to conservation of angular momentum, the $\mathcal{S}_1$ term, like the $\mathcal{I}_1$ term, has no effect on h. Thus the dominant term is the mass-quadrupole ($\mathcal{I}_2$), and so, as we only wanted leading order, we were correct neglecting the momentum moments in section 2.1.2.

To find an expression for E in section 2.1.2 we used Newtonian physics and the Einstein quadrupolar formula. If we consider relativistic effects due to spacetime curvature, as we did in section 2.1.3, we find that we can also express E in terms of a power series in v. For example, in the Schwarzschild case above, we can re-work equation 2.75 by noting that $v = r\Omega$. Using equation 2.69 we get:

$$v = \sqrt{\frac{M}{r}} \tag{2.79}$$

Thus we can re-write equation 2.75 as:

$$E = m\frac{1 - 2v^2}{\sqrt{1 - 3v^2}} \tag{2.80}$$

Since $v \ll 1$, we can write expand the denominator in a Maclaurin series, which results in:

$$\frac{E}{m} = 1 - \frac{1}{2}v^2 + \frac{3}{8}v^4 + \mathcal{O}(v^6) \tag{2.81}$$

To relate this to the phase of the gravitational wave, ϕ, we use the chain rule to write:

$$\frac{\mathrm{d}E}{\mathrm{d}t} = \frac{\mathrm{d}E}{\mathrm{d}v}\frac{\mathrm{d}v}{\mathrm{d}\phi}\frac{\mathrm{d}\phi}{\mathrm{d}t} \tag{2.82}$$

We can get $\mathrm{d}\phi/\mathrm{d}t$ by noting that:

$$\phi = \int 2\pi f_{\mathrm{GW}}\,\mathrm{d}t \tag{2.83}$$

thus:

$$\frac{\mathrm{d}\phi}{\mathrm{d}t} = 2\pi f_{\mathrm{GW}} \tag{2.84}$$

Also note that, from equation 2.69,

$$v = (\pi m f_{\mathrm{GW}})^{1/3} \tag{2.85}$$

We can therefore write:

$$\frac{\mathrm{d}\phi}{\mathrm{d}t} = \frac{2v^3}{m} \tag{2.86}$$

Substituting these results into the energy balance equation, rearranging terms, and integrating yields:

$$\phi(v) = \phi_0 + 2\int_v^{v_0} \frac{v^3(\mathrm{d}E/\mathrm{d}v)}{\mathcal{F}(v)}\,\mathrm{d}v \tag{2.87}$$

Here, ϕ_0 is the GW phase of the binary at the initial velocity v_0. (If we instead prefer ϕ as a function of f_{GW} we can use equation 2.85 to convert.) Thus, by expanding $\mathcal{F}$

and E to some desired power of v, and performing the integral, we can obtain higher order corrections to the gravitational wave. This is known as the *pN* approximation.

How the pN flux and energy is calculated to various orders is beyond the scope of this thesis. Here, we only note a few properties that are relevant to the low-mass CBC search that we will describe in later chapters:

- Each successive term in the expansion of $\mathcal{F}$ accounts for additional physics of the binary. For example, the "0pN" order ($\sim v^0$) results in GWs from a changing mass-quadrupole. We were able to compute the effects of this order above, in section 2.1.2. The "1pN" order ($\sim v^2$), which takes into account the mass-octopole and momentum-quadrupole, results in the perihelion shift of orbiting bodies. The next step is the "1.5pN" term[10] ($\sim v^3$), which results in frame dragging. Thus it takes into account the effects of spin/orbit[11] coupling. It also adds a correction for outgoing waves that scatter off of the binary's spacetime curvature, known as the "wave-tail". At 2pN we get effects from spin/spin coupling; etc.

- As described in the next chapter, in the low-mass CBC search we look for GWs by match filtering gravitational-wave detector data. When doing this, it is most important that we get the phasing of the waveform correct. Any minor errors in phase will quickly add up over the large number of cycles that the system spends in band, severely hampering our search. Thus we try to use the highest pN order to compute the phase evolution of the waveform. Currently, this is the 3.5pN term [39], which corresponds to seven orders beyond Newtonian.

- As compared to phase, the amplitude evolution is less critical to the search. Thus, for simplicity, we use the 0pN term, given in equation 2.57, to generate the amplitude evolution. These are known as "restricted" pN waveforms.

- For lower-mass systems in which spin is not important, we can further simplify the pN approximates by setting all spin degrees of freedom to zero. This is done in the current low-mass CBC search so that we can construct a "template bank" of waveforms; see section 3.2 in Chapter 3 for details.

---

[10]The numbering of the pN orders increase by 0.5 because they are based on powers of M/r.
[11]By "spin" we mean the intrinsic angular momentum of the component objects.

2.2 Detection of Gravitational Waves using Interferometers

Having established some properties of gravitational waves, we now turn to the question of detecting them. In section 2.1 we found that a passing GW will cause a particle at a distance δx from another particle to be displaced by an extra $\delta x h_+/2$ when the wave is at maximum amplitude. We can parametrize this as by defining the *strain*, h, as:

$$h \equiv \frac{\Delta L}{2L} \tag{2.88}$$

GWs are extremely weak: a binary system comprised of two objects each with masses $\sim M_\odot$ at a distance of $\sim 10\,\mathrm{Mpc}$ from Earth will produce a strain of $h \sim 10^{-21}$ [40]. In order to detect a passing GW we therefore need an extremely sensitive detector. A device that lends itself to such minute measurements is an *interferometer*.

Consider again figure 1. For simplicity, focus on the plus polarization. Now imagine we stick a mirror on the particles that lie on the $+x$ and $+y$ axes. Put a beam splitter at the origin, and shine a laser into the splitter, so that the laser beam gets split: half the light travels down the x-axis (or "arm") and the other done the y. After reflecting off the mirrors and travelling back down the arms the light will be recombined by the beam splitter. A straight forward analysis of the electric field in each arm [40] reveals that field of the combined light that is reflected back toward the laser is:

$$E_{\mathrm{refl}} = -ie^{i(2\pi f(t-L_x+L_y)}E_0 \sin(2\pi f(L_x - L_y)) \tag{2.89}$$

where E_0 is the amplitude of the field, f is the frequency of the light, and L_x and L_y are the lengths of the x- and y-arms as measured in the interferometer's rest frame, respectively.[12] The rest of the combined light exits down the $-y$ axis, in what we call the "output" port. The electric field of the output light is:

$$E_{\mathrm{out}} = -ie^{i(2\pi f(t-L_x+L_y)}E_0 \cos(2\pi f(L_x - L_y)) \tag{2.90}$$

Thus we can see that if the arms are of the same length (or, equivalently, if the light-travel time down each arm) is the same, the light will destructively interfere at the input port, resulting in zero reflected light, and constructively interfere at the output, resulting in maximum transmitted light. If the arms are slightly offset, however, the

---

[12]For now we will assume the entire interferometer sits in the same LLF.

strength of the light in the output will decrease. Thus by measuring the brightness of the light in the output port we can measure the relative phase difference of the light in each arm. This apparatus is known as a *Michelson* interferometer.

Now consider what happens to the light when a gravitational wave passes by. The line element is given by equation 2.21. For simplicity, consider only the effect on the light in the x-arm when the gravitational wave causes maximum displacement, i.e., when $\cos(\omega_{GW}t - \phi_{GW}) = 1$. Let us assume that the frequency of the gravitational wave is low enough that we can assume the strain is this constant value for the period of time that the light is in the arm. Since light travels along a null geodesic, ds^2 for light is zero. Thus we have:

$$dt^2 = (1 + h_+) dx^2 \tag{2.91}$$

Consider what happens to the light travel-time on the outgoing path. Integrating, we get:

$$\tau_{\text{out}} = \int_0^L \sqrt{1 + h_+} \, dx \approx L(1 + \frac{h_+}{2}) \tag{2.92}$$

where, as before, we assumed $|h_+| \ll 1$. On the return trip the light will incur the same dilation, giving a total round-trip travel time of:

$$\tau_{\text{rt,x}} = L(2 + h_+) \tag{2.93}$$

If we preform the same analysis on just the y-arm, we obtain $\tau_{\text{rt,y}} = L(2 - h_+)$. Taking the difference between the two yields:

$$\Delta\tau_{\text{rt}} = L(2 + h_+) - L(2 - h_+) = 2Lh_+ \tag{2.94}$$

We can relate this difference in travel time to a relative phase shift between the two beams:

$$\Delta\phi = 2\pi f \Delta\tau_{\text{rt}} = \frac{4\pi L}{\lambda} h_+ \tag{2.95}$$

where λ is the wavelength of the light. Since this phase change will result in a change in brightness in the output port, we can measure the amplitude of a passing GW by monitoring the output light.

2.2.1 Antenna Pattern

In the above analysis we only considered the effect of the plus polarization on the interferometer, with the "plus" being aligned along the arms of the device. Of course,

gravitational waves from real systems will not always have their plus polarizations aligned along the arms of interferometer. If the polarization was oriented such that it displaced both mirrors in the same manner, then the time dilation in each arm would be the same: $\tau_{\text{rt,x}} = \tau_{\text{rt,y}} \sim h_+$. Thus there would be no relative phase change between the two arms, and we could not measure the passing GW. The interferometer therefore has an *antenna pattern*.

We can relate the strain induced in the interferometer from any arbitrarily oriented system by [31]:

$$h(t) = F_+ h_+(t) + F_\times h_\times(t) \tag{2.96}$$

where:

$$F_+ = -\frac{1}{2}(1 + \cos^2\theta)\cos 2\varphi \cos 2\psi - \cos\theta \sin 2\varphi \sin 2\psi \tag{2.97}$$

$$F_\times = \frac{1}{2}(1 + \cos^2\theta)\cos 2\varphi \sin 2\psi - \cos\theta \sin 2\varphi \cos 2\psi \tag{2.98}$$

The angles θ, φ, and ψ are the Euler angles that relate the frame of the binary to the frame of the detector. Figure 5 shows what the antenna pattern looks like for a binary with $\iota = \psi = 0$. We see the "dead spots" — the point where the strain induced by the wave is 0 — are in the middle of the arms, in the plane of the interferometer. The best sensitivity occurs for systems that located directly above or below the interferometer (with $\iota = 0$); we therefore refer to such systems as being *optimally oriented*.

Returning to our general solutions for h_+ and $h_\times$ — equations 2.48 and 2.49 — we can account for the antenna pattern by writing [31]:

$$h(t) = \frac{A(t)}{\mathcal{D}} \cos(2\phi(t) - \theta) \tag{2.99}$$

where $\mathcal{D}$ is the *effective distance* of the binary from the detector, given by:

$$\mathcal{D} = \frac{r}{\sqrt{F_+^2(1 + \cos^2\iota)^2/4 + F_\times^2 \cos^2\iota}} \tag{2.100}$$

The phase angle, θ is given by:

$$\theta = \tan^{-1}\frac{F_\times 2\cos\iota}{F_+(1 + \cos^2\iota)} \tag{2.101}$$

$A(t)$ and $\phi(t)$ are calculated from the post-Newtonian approximation, discussed above. We will use these results in the next chapter.

The effective distance gives the distance to the binary as if it were optimally oriented. Note that a binary that is not optimally oriented is indistinguishable from a binary that is, but is further away. In order to break the degeneracy we need a network of detectors to triangulate the source. A brief overview of how this is done is presented in chapter 8.

2.2.2 The LIGO and Virgo Interferometers

The LSC operates three interferometers, or *IFOs*: two at the LIGO Hanford Observatory (LHO) in Hanford, Washington — which we will denote H1 and H2 — and one at the LIGO Livingston Observatory (LLO) in Livingston, Louisiana, which we denote L1. The Virgo Collaboration operates one interferometer, which we denote V1, in Cascina, Italy. These instruments run on the same basic principles as our toy interferometer described above. In practice, however, they are far more sophisticated, as they must account for a number of real-world problems. We note here a few of their basic properties that are relevant to the later chapters of this thesis.

In the above section we found that the phase shift induced in the arms of the interferometer was proportional to the amplitude of the gravitational-wave, and to the length of the arm. This would suggest that the longer we make the arms, the more sensitive our instrument will be. However, if the light remains in the arm for a full cycle of the gravitational wave, any time dilation gained on the crest of the wave will be cancelled by the trough. We therefore wish the light to remain in the arms for $1/2$ the period of the GW, as this will result in the maximum phase difference. If we are searching for waves with a frequency $\sim 100\,\mathrm{Hz}$, this sets an optimal length for the arms to be:

$$L = \frac{\lambda}{4} = \frac{c}{2f_{\mathrm{GW}}} = \frac{c}{200} \sim 1000\,\mathrm{km} \tag{2.102}$$

This is a hopelessly large distance for a ground-based interferometer. The LIGO and Virgo IFOs deal with this difficulty by using *Fabry-Perot cavities*.

Figure 6 shows a schematic of a Fabry-Perot interferometer. By placing a semi-transparent mirror in the near-end of each arm — referred to as the *input test masses* (ITMX and ITMY), to distinguish it from the *end test masses* (ETMX and ETMY) — a cavity is created. By keeping the length of the arms close to the resonance of the light, the light bounces around for 200 more times [31] before exiting the arms.

$$\sqrt{F_+^2(\theta, \phi, \psi = 0) + F_\times^2(\theta, \phi, \psi = 0)}$$
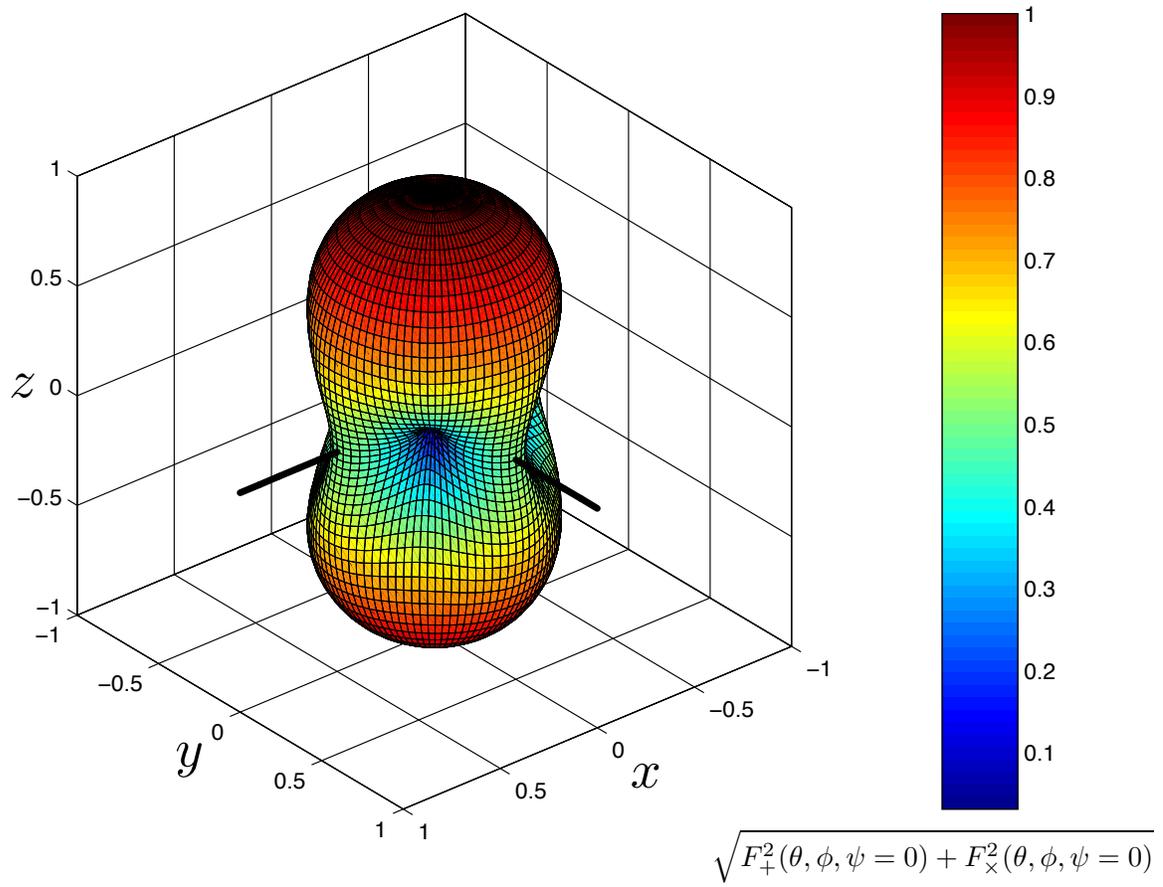
Figure 5 : The antenna pattern of an interferometer (IFO). The arms of the IFO are plotted for reference. Figure originally published in [31].

Thus the actual length of the arms can be $\mathcal{O}(1\,\mathrm{km})$ while the effective length will be $\mathcal{O}(10^3\,\mathrm{km})$. This results in the maximum detectable phase shift at $100\,\mathrm{Hz}$. Both the LIGO and Virgo interferometers are $\mathcal{O}(\mathrm{km})$: H1 and L1 are $4\,\mathrm{km}$ long, and V1 is $3\,\mathrm{km}$. In initial LIGO H2 was $2\,\mathrm{km}$, but this is planned to be expanded to $4\,\mathrm{km}$ for advanced LIGO.

For the Fabry-Perot cavity to work effectively, we need to keep the interferometer arms on resonance, or "locked." This is done by a feedback loop involving a series of servos. By monitoring motions of the mirrors through the output port (and other channels), the servos are actuated in order to keep the mirrors in lock. One of these channels monitors, and adjusts to, the differential arm length, or `DARM_ERR`. Since a passing gravitational wave would show up as oscillations in `DARM_ERR`, we can use this channel to obtain a measure of the strain. In order to generate the strain time-series, $s(t)$, calibration studies are carried out [41, 42, 43]. The calibrated time-series, known in LIGO as `LDAS_STRAIN`, is the data stream we monitor to search for GWs.
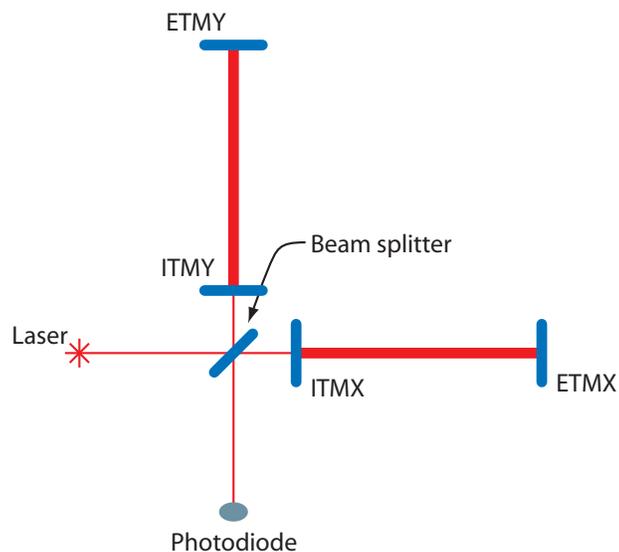
Figure 6 : Schematic of an interferometer with a Fabry-Perot cavity. Originally published in [31].

Since gravitational waves are so weak, sources of noise are a major concern. We briefly note a few here:

- *Seismic Noise*: As the name suggests, this is noise due to ground motion. This motion can couple into the interferometer through the test masses' suspension

system. To counteract it, a number of seismic isolation systems are put in place between the ground and the wires to which the test masses are attached. Seismic noise is primarily a problem at low-frequency. In initial and enhanced LIGO, the seismic "wall" was at $\sim 40\,\mathrm{Hz}$; in advanced LIGO it is projected to be at $\sim 10\,\mathrm{Hz}$. When analyzing LIGO and Virgo data, we use a lower cut-off frequency, f_0, to remove noise power below these frequencies.

- *Thermal Noise*: This is due to the thermal vibrations in the mirrors and the wires from which the mirrors hang. This noise primarily affects frequencies in the 40 to $200\,\mathrm{Hz}$ range.

- *Shot Noise*: This is the primary noise source at high ($\sim$kHz) frequency. If we view the light in the interferometer from a quantum perspective, we recognize that we are limited by Poisson counting statistics of the photons. The error in the mean number of counts in some given amount of time goes as $1/\sqrt{N}$. Thus, the error in our measurement of the phase of the light increases with higher frequencies. This suggests using a more powerful laser, which was done for enhanced LIGO, and is planned for advanced LIGO. However, more intense light will also cause higher radiation pressure on the mirrors — which is a noise source at lower frequencies — and so these considerations must be balanced. Additionally, more intense light will heat up the mirrors, which causes greater thermal noise.

These dominant noise sources, and other sources, add up to give the noise in the detector. We can visualize this by considering the amplitude spectral density, or "noise curve" of the detectors. Figure 7 shows what the noise curves looked like for initial LIGO and Virgo, and what they are expected to be for the advanced detector era.

All of these sources contribute to the *stationary* Gaussian noise of the detectors. In addition, there are a number of environmental and instrumental transient sources, or "glitches", for which we have no model. As discussed in the next few chapters, a major part of the CBC data-analysis pipeline is mitigating the effect of these transient sources.
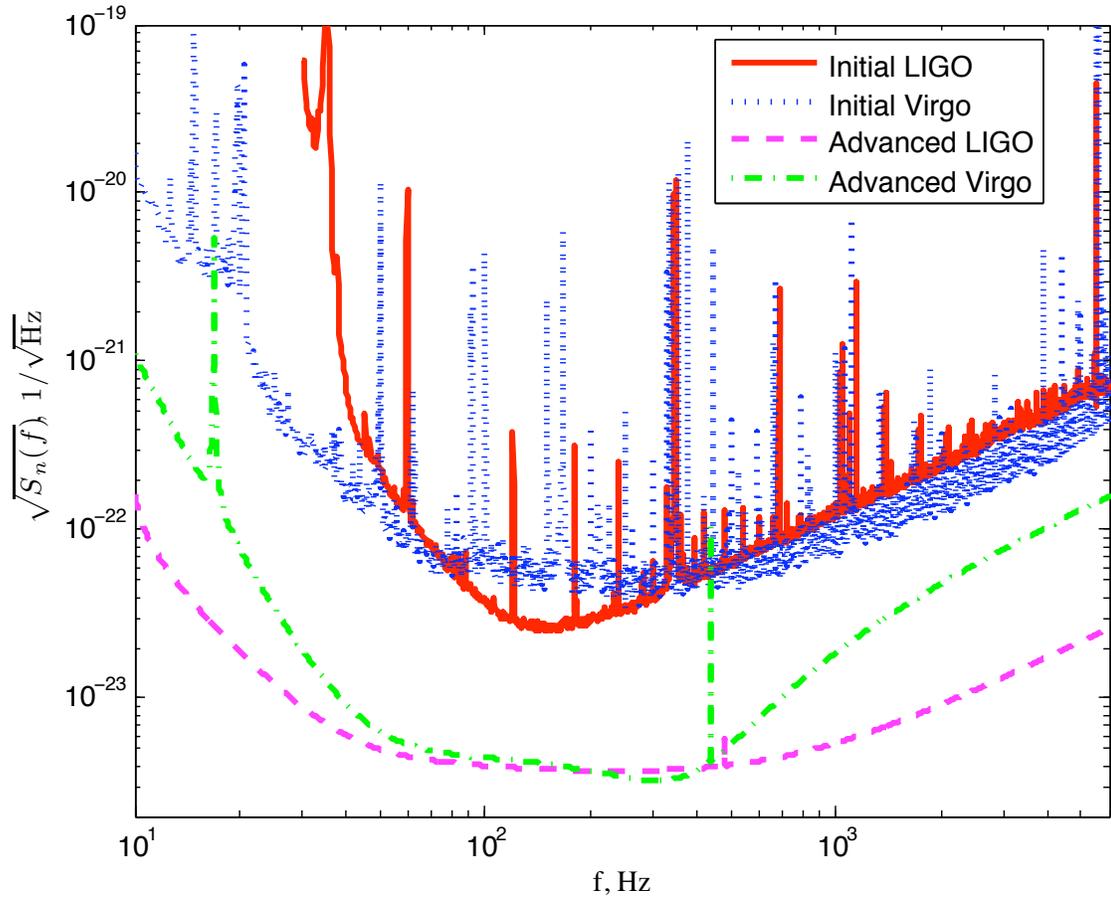
Figure 7 : Amplitude spectral densities of initial LIGO and Virgo detectors. Shown also are the expected noise curves for advanced LIGO and Virgo. Figure originally published in [10].

2.3 Astrophysical Sources of Gravitational Waves from Compact Binary Coalescence

Binaries with compact component masses on order of $\sim M_\odot$ which are orbiting each other at frequencies of $\sim 10^2$ Hz create GWs that can be measured by the LIGO and Virgo detectors. It is therefore possible to detects GWs from binary-neutron star systems, binary black hole systems and neutron-star black-hole binaries as they inspiral and coalesce.

Table 1 gives estimated rates of BNS, NSBH, and BBH coalescence in the universe. The component masses used in the estimates are $1.4/1.4M_\odot$, $1.4/10.0M_\odot$, and $10.0/10.0M_\odot$. The uncertainty in these estimates are large, varying by as much as two orders of magnitude. This is due to the small sample size of observed systems, and from uncertainties in the population synthesis models on which some rates are based. Rates of systems involving black holes are perhaps the most uncertain, since no black hole has (or can) be directly observed by existing telescopes.[13] We can be more confident about BNS rates, however, as these have been directly observed. Of note is the Hulse-Taylor pulsar (PSR B1913+16), which was shown to be losing energy at the rate expected from GW emission [44, 45]. The separation distance between the system's constituent neutron stars is such that they will not coalesce for another 300 million years, however [31].

Using the LIGO and Virgo noise power spectral density (PSD) we can estimate the expected rate of detections in a year of data at a given detection threshold (how this is done is detailed in the next Chapter). Table 2 shows the estimated rate in initial LIGO is and what it is expected to be for advanced LIGO. As we can see, the rates in initial LIGO are all less than one per year, even on the optimistic end. For advanced LIGO, however, we see that the expected detection rates are on order of ~ 10 per year. These rates are based on the projected sensitivity of the detectors and on our ability to detect a GW event at a signal-to-noise ratio of 8. (See the next chapter for a discussion of signal-to-noise ratio.) We expect advanced LIGO to have a range that is ~ 10 times that of initial LIGO. Thus we get factor of ~ 1000 increase in sensitive volume, which leads to the predicted detection rates.

---

[13]Being able to directly detect black holes, and measure properties of their spacetime curvature, is a major advantage of GW detectors.

If we do not detect a GW event (to some confidence level) we can use the time searched and the sensitivity of the instrument to set an upper-limit on the rate of CBCs in the universe. This is done using the *loudest-event statistic* method [46, 47, 48]. After performing a search, we calculate the relative probability that the loudest event was due to a GW as compared to the probability it was due to noise, or background. We then calculate the efficiency of the IFOs at detecting signals with the same value of the ranking statistic[14] as the loudest event. This is done by injecting fake signals, or "software injections" into the data then checking how well they are recovered by our data-analysis pipeline. Performing a Monte Carlo simulation over the target population gives us an estimate of the detector's efficiency. Since we know the effective distance of the injections, we can put the efficiency in terms of distance. From that, we get the sensitive volume of the universe the search was sensitive to.[15] Combining this with the observation time, we get a rate upper limit of:

$$\mathcal{R}_{90\%} = \frac{\beta}{VT} \tag{2.103}$$

where V is the sensitive volume and T is the observation time; the rate is computed to the 90% confidence level. The factor β depends on the likelihood that the loudest event was a signal. As the likelihood goes to zero (i.e., the probability is low that the data contains a GW), then $\beta \to 2.303$ at the 90% confidence level. As the likelihood goes to infinity (i.e., the probability is high that the data contains a GW) $\beta \to 3.9$ [46, 47, 48].

In Chapter 6 we present upper limits calculated from a six-month long search of S5 data using the loudest-event statistic method. For more details on the loudest-event statistic method see [46, 47, 48]. For more details on how the rates presented in Table 1 were calculated, see [10] and the papers cited therein.

---

[14]To rank triggers we use the inverse false alarm rate, or IFAR. See chapters 4, 5, and 6 for details.

[15]When computing the sensitive volume we must take into account the expected astrophysical distribution of sources. In initial S5 this was done by using a galaxy catalog. In S6, the range of the detectors is large enough that we can assume the sources are uniformly distributed across the sky, which allows us to simply use the range.

Source	R_{low} ($\text{Mpc}^{-3}\text{Myr}^{-1}$)	R_{best} ($\text{Mpc}^{-3}\text{Myr}^{-1}$)	R_{high} ($\text{Mpc}^{-3}\text{Myr}^{-1}$)
BNS	0.01	1	10
NSBH	6×10^{-4}	0.03	1
BBH	1×10^{-4}	0.005	0.3

Table 1 : Estimated rates of BNS, NSBH, and BBH coalescence in the universe. R_{best} indicates best estimate; "low" and "high" indicate pessimistic and optimistic rates, respectively. The component masses used in the estimates are $1.4/1.4\text{M}_\odot$ for BNS, $1.4/10.0\text{M}_\odot$ for NSBH, and $10.0/10.0\text{M}_\odot$ for BBH. For details on how these numbers were derived, see [10].

Era	Source	$\dot{N}_{\text{low}}$ (yr^{-1})	$\dot{N}_{\text{best}}$ (yr^{-1})	$\dot{N}_{\text{high}}$ (yr^{-1})
	BNS	2×10^{-4}	0.02	0.2
Initial	NSBH	7×10^{-5}	0.004	0.1
	BBH	2×10^{-4}	0.007	0.5
	BNS	0.4	40	400
Advanced	NSBH	0.2	10	300
	BBH	0.4	20	1000

Table 2 : Estimated detection rates in initial and advanced LIGO. The component masses used in the estimates are $1.4/1.4\text{M}_\odot$ for BNS, $1.4/10.0\text{M}_\odot$ for NSBH, and $10.0/10.0\text{M}_\odot$ for BBH. For details on how these numbers were derived, see [10].

Chapter 3

Obtaining Gravitational Wave Triggers from Interferometer Data

Using General Relativity we were able to write down an expression for the strain induced in an interferometer from a passing gravitational wave. In this chapter we provide a survey of how to detect gravitational-wave signals from CBCs in the interferometer data in the presence of noise. We begin by deriving the *matched filter* for a single template, which is the optimal filter if the detector noise is stationary and Gaussian. Next, we show how matched filtering can be performed for a bank of templates in order to recover the parameters of a signal. This is followed by a discussion of the χ^2 test, which can be used to suppress triggers from non-Gaussian transients that are present in real detector data. Finally, we describe how a coincidence test can be performed across multiple detectors to further decrease the number of false noise triggers produced by a search.

3.1 Detecting a Gravitational Wave Using a Matched Filter

Equation 2.99 gave the strain, $h(t)$, induced in an interferometer when a gravitational wave from an inspiraling compact binary with non-spinning component masses passes through. The strain was a function of chirp mass, $\mathcal{M}$, and symmetric-mass ratio, η. If we wish to search for a gravitational wave that came from a binary with given $\mathcal{M}$ and η, we can generate a waveform, or *template*, that characterizes it. We now address the question: if a gravitational wave that matches our template exists in the

IFO data, how do we find it?

Let the output of the detector be the time series $s(t)$, which is a measure of the displacement of the IFO's mirrors as a function of time. If a gravitational wave with waveform $h(t)$ exists in the data, then:

$$s(t) = h(t) + n(t) \tag{3.1}$$

where $n(t)$ is the strain induced by all other, non gravitational-wave, sources, or "noise," as a function of time. If no signal exists in the data, then $s(t) = n(t)$. We do not know *a priori* if h exists in the data; further, even if h does exist, we do not know *when* it occurs — that is, we do not know its coalescence time, τ_c.[1] Our goal, then, is to find a filter that takes $s(t)$ and $h(t - \tau)$ as input and returns a number, $\rho(\tau)$, that is proportional to the probability that h is in s with coalescence time τ, $P(h(t-\tau)|s(t))$. We additionally require that, if h is in s, this filter is at a maximum at the point that h occurs. If we have such a filter — known as the *optimal filter* — then we can determine that a signal exists in the data if $\rho(\tau)$ exceeds some pre-determined threshold ρ^*. Further, we can determine the coalescence time of h by simply incrementing τ and evaluating ρ at each increment, selecting points when ρ is at a maximum. (This is known as the method of maximum likelihood [31].)

We begin by assuming that $\tau = \tau_c = 0$ and looking for the filter that maximizes $P(h|s)$ (for now, we will drop the "(t)" for clarity). The problem of finding an optimal filter to extract signals from (Gaussian) noise is well studied, and has been applied to radar analysis for decades. Here, we follow the method outlined in [49, 50, 31] — all of which use methods detailed in Wanstein and Zubakov [51] — to derive the filter.

Using Bayes' theorem [52] we can write $P(h|s)$ as:

$$P(h|s) = \frac{P(s|h)P(h)}{P(s)} \tag{3.2}$$

---

[1]Note that the exact time when h occurs is somewhat arbitrary. We could choose any point in the evolution of the waveform and label that as the point that h occurs. For CBC templates, we chose to use the time of coalescence since it is easily identifiable: it is the point that the pN approximation goes to infinite frequency.

where:

$$P(s|h) \equiv \text{the probability of getting } s \text{ if } h \text{ exists in it}$$
$$P(h) \equiv \text{the probability of the signal, } h, \text{ occurring}$$
$$P(s) \equiv \text{the probability of getting } s$$

Since the signal either does or does not exist in the data, the probability of getting a particular instance of s is:

$$P(s) = P(s|0)P(0) + P(s|h)P(h) \tag{3.3}$$

where:

$$P(s|0) \equiv \text{the probability of getting } s \text{ if no signal exists}$$
$$P(0) \equiv \text{the probability of getting no signal}$$

Substituting this into 3.2 we have:

$$
\begin{aligned}
P(s|h) &= \frac{P(s|h)P(h)}{P(s|0)P(0) + P(s|h)P(h)} \\
&= \frac{P(s|h)}{P(s|0)\left(P(0)/P(h) + P(s|h)/P(s|0)\right)} \\
&= \frac{\Lambda}{P(0)/P(h) + \Lambda}
\end{aligned}
\tag{3.4}
$$

where

$$\Lambda \equiv \frac{P(s|h)}{P(s|0)} \tag{3.5}$$

is the *likelihood ratio*. $P(0)$ and $P(h)$ are known as *priors*: they represent our *a priori* belief that a signal does or does not exist, irrespective of the detector's ability to detect it. We do not need to concern ourselves with assigning values to them, however. Instead we note that $P(s|h)$ is a monotonically increasing function of Λ. Therefore, since we are only interested in a filter that maximizes $P(s|h)$ — and not the exact value of $P(s|h)$ — we can limit our focus to evaluating Λ, and threshold on the point that it reaches a maximum. We further note that the natural logarithm of Λ also increases monotonically with $P(s|h)$. Since we will be interested in evaluating the likelihood in the region that Λ is a maximum, it is common practice to instead evaluate the *log-likelihood*, $\ln \Lambda$, instead, as it varies less rapidly around the region of interest [52].

Before calculating the log-likelihood, we note that we do not know *a priori* the phase of the binary, θ. Therefore, we treat θ as a *nuisance* parameter, which we marginalize over [52]. Thus, we write the likelihood ratio as:

$$\Lambda = \int_0^{2\pi} p(\theta)\lambda(\theta)\,\mathrm{d}\theta$$

$$= \frac{1}{2\pi P(s|0)} \int_0^{2\pi} p(s|h(\theta))\,\mathrm{d}\theta \tag{3.6}$$

Here, $\lambda(\theta)$ is the likelihood ratio at a given θ, $p(\theta)$ is the prior probability of getting θ, and we have written $P(s|h)$ as the probability distribution function (PDF) $p(s|h(\theta))$; $P(s|0)$ remains outside of the integral as it has no θ dependence. Assuming any phase is equally likely, we set $p(\theta) = 1/2\pi$ and pulled it out of the integral, also.

To calculate the likelihood ratio, we need $p(s|h(\theta))$ and $P(s|0)$. We focus first on $P(s|0)$. If $s(t)$ has no signal in it then it is simply $n(t)$. $P(s|0)$ is therefore the probability of getting a particular realization of the noise. Let us assume that the noise is a stationary Gaussian process with zero mean. For our purposes it will be useful to characterize the noise by its PSD rather than its variance. The PSD is defined as the Fourier Transform of the autocorrelation of the noise [51]:

$$S_n(f) \equiv \int_{-\infty}^{\infty} R(\tau)e^{-i2\pi ft}\,\mathrm{d}t = \overline{\widetilde{n}^*(f)\widetilde{n}(f)} \tag{3.7}$$

where,

$$R(\tau) = \overline{n(t)n(t-\tau)} \tag{3.8}$$

is the autocorrelation function. Note that $R(0) = \overline{n(t)^2}$, which is the variance since $\overline{n(t)} = 0$. We restrict our analysis to positive frequencies; thus we will use the one-sided PSD $S_n(|f|) = S_n(f)/2$. It is shown in [49] that with these assumptions, the probability of getting a given realization of the noise, $n = s$, is:

$$P(s|0) = \alpha \exp[-\frac{1}{2}(s|s)] \tag{3.9}$$

where α is a normalization constant and the inner-product $(\cdot|\cdot)$ is defined as:

$$(a|b) \equiv \int_{-\infty}^{\infty} \frac{\widetilde{a}^*(f)\widetilde{b}(f) + \widetilde{a}(f)\widetilde{b}^*(f)}{S_n(|f|)}\,\mathrm{d}f \tag{3.10}$$

$$= 2\int_{-\infty}^{\infty} \frac{\widetilde{a}^*(f)\widetilde{b}(f)}{S_n(|f|)}\,\mathrm{d}f \tag{3.11}$$

In going from 3.10 to 3.11 we have used the fact that for real functions of time (which both h and s are), $\widetilde{g}^*(f) = \widetilde{g}(-f)$.

Using this result we can also find $p(s|h(\theta))$. If h is in the data, then $n(t) = s(t) - h(t, \theta)$. Thus,

$$
\begin{aligned}
p(s|h) &= \alpha \exp\{-\frac{1}{2}(s - h|s - h)\} \\
&= \alpha \exp\{-\frac{1}{2}[(s|s) - 2(h|s) + (h|h)]\} \\
&= P(s|0) \exp\{(h|s) - (h|h)/2\}
\end{aligned}
\tag{3.12}
$$

Substituting 3.9 and 3.12 into 3.5, the likelihood ratio becomes:

$$
\begin{aligned}
\Lambda &= \frac{1}{2\pi} \int_0^{2\pi} \exp\{(h|s) - \frac{(h|h)}{2}\} \, d\theta \tag{3.13} \\
&= \frac{1}{2\pi} e^{-(h|h)/2} \int_0^{2\pi} \exp\{(\, C(t)\cos(2\phi(t) - \theta) \mid s \,)\} \, d\theta \tag{3.14}
\end{aligned}
$$

In going from 3.13 to 3.14 we have used the general form of $h(t, \theta)$ given in equation 2.99, with $C(t) = A(t)/\mathcal{D}$. We have also pulled the $(h|h)$ term out of the integral as it is simply a number — it is the inner product of h with itself — and has no phase dependence. In order to evaluate the integral, we re-write the inner product of s with h as follows:

$$
\begin{aligned}
(h|s) &= (\, C(t)\cos(2\phi(t) - \theta) \mid s(t) \,) \\
&= (\, C(t)\cos(2\phi(t)) \mid s(t) \,)\cos(\theta) + (\, C(t)\sin(2\phi(t)) \mid s(t) \,)\sin(\theta) \\
&= x\cos(\theta) + y\sin(\theta) \\
&= |z|\cos(\Phi)\cos(\theta) + |z|\sin(\Phi)\sin(\theta) \\
&= |z|\cos(\Phi - \theta)
\end{aligned}
\tag{3.15}
$$

where:

$$
x = (\, C(t)\cos(2\phi(t)) \mid s(t) \,) \equiv (h_+|s) \tag{3.16}
$$

$$
y = (\, C(t)\sin(2\phi(t)) \mid s(t) \,) \equiv (h_\times|s) \tag{3.17}
$$

$$
|z| = \sqrt{x^2 + y^2} \tag{3.18}
$$

$$
\Phi = \tan^{-1}\left(\frac{y}{x}\right) \tag{3.19}
$$

Since the GW phase, $2\phi(t)$, is 90° out of phase in equations 3.16 and 3.17, we have identified x and y as the inner products of the plus and cross polarizations, respectively, with the data. Due to their orthogonality, note that:

$$(h_+|h_+) = (h_\times|h_\times) \qquad\qquad = (h|h) \qquad (3.20)$$

$$(h_+|h_\times) = (h_\times|h_+) \qquad\qquad = 0 \qquad (3.21)$$

Also note that we can let z be complex:

$$z = x + iy \qquad (3.22)$$

This will prove useful below.

Substituting equation 3.15 into the likelihood ratio, we have:

$$
\begin{aligned}
\Lambda &= \frac{1}{2\pi} e^{-(h|h)/2} \int_0^{2\pi} \exp\{|z|\cos(\Phi - \theta)\}\, d\theta \\
&= e^{-(h|h)/2} I_0(|z|) \\
&= e^{-(h|h)/2} \sum_{k=0}^{\infty} \frac{(|z|^2/4)^k}{(k!)^2} \qquad (3.23) \\
&\leq e^{-(h|h)/2} \left(\sum_{k=0}^{\infty} \frac{(|z|^2/4)^k}{k!} \right) \left(\sum_{m=0}^{\infty} \frac{1}{m!} \right) \qquad (3.24) \\
&\leq e^{-(h|h)/2} e^{|z|^2/4 - 1} \qquad (3.25)
\end{aligned}
$$

where $I_0(|z|)$ is the modified Bessel function of the first kind, which is equal to the sum in equation 3.23 [53]. To go from equation 3.23 to 3.24 we have used the fact that $(|z|^2/4)^k/k!$ and $1/k!$ are positive definite for all k. Thus, the likelihood ratio is maximized when it is equal to $\exp\{|z|^2/4 + 1 - (h|h)/2\}$, and so we can threshold when the log-likelihood is:

$$\ln\Lambda = \frac{1}{4}|z|^2 + 1 - \frac{1}{2}(h|h) \qquad (3.26)$$

The factor of 1/4 and the +1 term on the right-hand side of the equation only scale and offset $\ln\Lambda$ by constants. We can therefore threshold on:

$$4\ln\Lambda - 4 = |z|^2 - 2(h|h) \qquad (3.27)$$

instead, without any loss in ability to detect h. The $(h|h)$ term, however, causes the log-likelihood to be template dependent. Since we will be filtering multiple templates,

we prefer to use a quantity that is independent of the template used. Therefore, we normalize both sides by $(h|h)$. Dropping the remaining -2 offset, we are left with:

$$\frac{|z|^2}{(h|h)} \equiv \rho^2 \tag{3.28}$$

The quantity ρ^2 is our detection statistic.

In the absence of a signal, $(h|h)$ is the variance of the filter. To see this, first note that the mean of $(h|n)$ is zero:

$$
\begin{aligned}
\overline{(h|n)} &= \overline{2 \int_{-\infty}^{\infty} \frac{\widetilde{h}^*(f)\widetilde{n}(f)}{S_n(|f|)} \, \mathrm{d}f} \\
&= 2 \int_{-\infty}^{\infty} \frac{\widetilde{h}^*(f)\overline{\widetilde{n}(f)}}{S_n(|f|)} \, \mathrm{d}f \\
&= 0
\end{aligned}
\tag{3.29}
$$

since $\overline{n(t)} = \overline{\widetilde{n}(f)} = 0$. The mean of the square of $(h|n)$ is:

$$
\begin{aligned}
\overline{(h|n)^2} &= \overline{\left| 2 \int_{-\infty}^{\infty} \frac{\widetilde{h}^*(f)\widetilde{n}(f)}{S_n(|f|)} \, \mathrm{d}f \right|^2} \\
&= 4 \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{\widetilde{h}(f')\widetilde{h}^*(f)\overline{\widetilde{n}^*(f')\widetilde{n}(f)}}{S_n(|f'|)S_n(|f|)} \, \mathrm{d}f' \, \mathrm{d}f \\
&= 4 \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{\widetilde{h}(f')\widetilde{h}^*(f)S_n(|f'|)\delta(f' - f)}{2S_n(|f'|)S_n(|f|)} \, \mathrm{d}f' \, \mathrm{d}f \\
&= 2 \int_{-\infty}^{\infty} \frac{\widetilde{h}(f)\widetilde{h}(f)}{S_n(|f|)} \, \mathrm{d}f \\
&= (h|h)
\end{aligned}
\tag{3.30}
$$

Thus the variance is:

$$\sigma^2 = \overline{(h|n)^2} - \overline{(h|n)}^2 = (h|h) \tag{3.31}$$

and we can write:

$$\rho^2 = \frac{|z|^2}{\sigma^2} \tag{3.32}$$

We therefore identify ρ as the *signal-to-noise ratio (SNR)* of the template.

In order to evaluate ρ for arbitrary values of $\tau \neq \tau_c \neq 0$, we note that the Fourier

Transform of $h(t - \tau)$ is:

$$
\begin{aligned}
\widetilde{h}(f) &= \int_{-\infty}^{\infty} h(t - \tau) e^{-i2\pi ft} \, dt \\
&= e^{i2\pi f\tau} \int_{-\infty}^{\infty} h(t') e^{-i2\pi ft'} \, dt' \\
&= e^{i2\pi f\tau} \widetilde{h}(f')
\end{aligned}
\tag{3.33}
$$

$\widetilde{h}(f')$ is the Fourier Transform of $h(t)$ independent of the coalescence time. We can thereby account for the unknown coalescence time by filtering:

$$
e^{-i2\pi f\tau} \widetilde{h}(f)
$$

This introduces a time-dependence to ρ:

$$
\begin{aligned}
\rho(\tau)^2 &= \frac{|z(\tau)|^2}{\sigma^2} \\
&= \frac{1}{\sigma^2} \left((h_+(\tau)|s)^2 + (h_\times(\tau)|s)^2 \right) \\
&= \frac{1}{\sigma^2} \left\{ \left(\int_{-\infty}^{\infty} \frac{\widetilde{h}_+^*(f)\widetilde{s}(f)}{S_n(|f|)} e^{i2\pi f\tau} \, df \right)^2 + \left(\int_{-\infty}^{\infty} \frac{\widetilde{h}_\times^*(f)\widetilde{s}(f)}{S_n(|f|)} e^{i2\pi f\tau} \, df \right)^2 \right\}
\end{aligned}
\tag{3.34}
$$

To carry out the maximum likelihood operation, we create a SNR time series by incrementing τ and filtering using equation 3.34 at each step. We then select points where $\rho(t)$ is a maximum; if the SNR of a maximum point exceeds our pre-determined threshold we save it. Each of the saved events is a *trigger*.

Now that we have established the SNR as our detection statistic, let us examine a few important properties of it. In the absence of a signal ($s = n$) ρ is χ^2 distributed with two degrees of freedom; the squared mean is:

$$
\begin{aligned}
\overline{\rho^2} &= \frac{1}{\sigma^2} \left(\overline{(h_+|n)^2} + \overline{(h_\times|n)^2} \right) \\
&= \frac{((h_+|h_+) + (h_\times|h_\times))}{(h|h)} \\
&= 2
\end{aligned}
\tag{3.35}
$$

since $(h_+|h_+) = (h_\times|h_\times) = (h|h)$. If we had only filtered one of the phases, $\overline{\rho^2}$ of noise alone would be 1. While filtering both h_+ and $h_\times$ allowed us to account for the unknown phase, it has resulted in a doubling the noise floor. Our threshold for

keeping triggers must therefore be larger than $\rho^2 = 2$, as it is impossible to distinguish noise from triggers at or below this value.

As seen in equation 2.99, h is inversely proportional to the effective distance to the binary, $\mathcal{D}$. We can exploit this to relate $\mathcal{D}$ to the SNR. Assume the data only contains a signal (and no noise), i.e., $s = h'_+$, from a binary that has an effective distance of $\mathcal{D}'$ Mpc. For simplicity we also assume the signal is plus-polarized with coalescence time $\tau_c = 0$. The templates we filter with are generated at a canonical distance of 1 Mpc; therefore $h'_+ = \mathcal{D}'^{-1} h_+$. The SNR would be:

$$
\begin{aligned}
\rho^2 &= \frac{1}{\sigma^2} \left((h_+|h'_+)^2 + (h_\times|h'_+)^2 \right) \\
&= \frac{1}{\sigma^2} \left(\mathcal{D}'^{-2}(h_+|h_+)^2 + \mathcal{D}'^{-2}(h_\times|h_+)^2 \right) \\
&= \mathcal{D}'^{-2} \sigma^2
\end{aligned}
$$

Thus:

$$
\mathcal{D} = \frac{\sigma}{\rho} \tag{3.36}
$$

Since σ is inversely proportional to $\sqrt{S_n(|f|)}$, we see that it is a measure of the sensitivity of the detector. The noisier the detector is, the smaller σ is, and so for a given SNR, the range that we can detect will be smaller.

If the detector output is Gaussian, as we have assumed above, then we could relate SNR directly to false alarm rate, and we could simply set the SNR threshold based on some desired false alarm rate. Any trigger with a SNR above that threshold would be considered a gravitational wave. However, as we will see in section 3.3, the real detectors have high rates of non-Gaussian transients ("glitches") for which we have no model. We therefore have to measure the false alarm rate directly. How this is done is discussed in chapter 4.

Finally, we will find it useful to define the complex form of the SNR using the complex form of z given in equation 3.22:

$$
\begin{aligned}
\varrho &= \frac{z}{\sigma} \\
&= \frac{(h_+|s) + i(h_\times|s)}{(h|h)}
\end{aligned} \tag{3.37}
$$

$$
\rho = |\varrho| \tag{3.38}
$$

We will use this complex form in the χ^2 test, discussed below.

3.2 Filtering with Multiple Templates

In the above section we assumed that the signal we were looking for had the same parameters as the template we used to search for it. We now address how to search for signals across a range of chirp masses and symmetric-mass ratios — collectively known as *intrinsic* parameters — so that we can search for signals from many types of systems. Alternatively, if one signal exists in the data, we can think of this question as addressing how to estimate its parameters. The method presented here was first described in [54]; additional discussions and improvements can be found in [55, 56, 57, 58].

In order to find the intrinsic parameters we use a dicrete *bank* of templates. The templates are laid out in the bank by computing how quickly the inner product — or *overlap*, $\mathcal{O}$ — between a template with intrinsic parameters θ_μ and one with parameters $\theta_\mu + \Delta\theta_\mu$ falls off with increasing $\Delta\theta_\mu$. This is done by expanding the overlap around $\Delta\theta_\mu = 0$ [54, 55, 57]:

$$
\begin{aligned}
\mathcal{O} &\equiv (h(\theta_\mu)|h(\theta_\mu + \Delta\theta_\mu)) & (3.39) \\
&= 1 + \frac{1}{2} \left.\frac{\partial^2 \mathcal{O}}{\partial\Delta\theta^\mu \partial\Delta\theta^\nu}\right|_{\Delta\theta^\kappa=0} \Delta\theta^\mu \Delta\theta^\nu + \ldots \\
&\approx 1 - g_{\mu\nu}\Delta\theta^\mu \Delta\theta^\nu & (3.40)
\end{aligned}
$$

where

$$
g_{\mu\nu} = -\frac{1}{2} \left.\frac{\partial^2 \mathcal{O}}{\partial\Delta\theta^\mu \partial\Delta\theta^\nu}\right|_{\Delta\theta^\kappa=0} \tag{3.41}
$$

$g_{\mu\nu}$ is the metric on the parameter space around the point θ_μ; it gives the "distance" between two templates with slightly different parameters in terms of how much SNR will be lost from their mis-match. Using the metric we can determine how many templates to place in a region of parameter space for an acceptable loss in SNR due to the discretization. We quantify this by defining the *minimal match (MM)* [54], given by [58]:

$$
\min_{\theta^\mu} \max_i \; (h(\theta_i^\mu)|h'(\theta^\mu)) \geq MM \tag{3.42}
$$

where $i \in \{1, 2, \ldots N_{\text{templates}}\}$, $\mu \in \{1, 2, \ldots N_{\text{parameters}}\}$, and $h'(\theta^\mu)$ is the waveform of the signal in the data. In other words, the bank should be constructed such that there exists at least one template for which the match between it and any signal with

parameters θ^μ is $\geq MM$. We can use equation 3.36 to determine an acceptable MM. For example, if $MM = 0.97$, then the maximum loss in SNR for a signal that falls between two templates is 3%. The loss in effective range will be also be 3%, which translates to a loss in volume of $(\delta \mathcal{D})^3 = 9\%$. Choosing a larger MM may appear to increase the range. However, a larger MM increases the number of templates needed, and this must be weighed against computational cost and false alarm rate. The more templates we filter with, the higher the probability of getting accidental triggers from noise. This also decreases the sensitive volume as the SNR at which we could claim a confident detection would have to increase. A $MM = 0.97$ is the currently used limit in CBC searches.

If an analytic solution exists for the waveform in terms of the parameters θ_μ, then the metric can be evaluated directly to figure out the number of templates needed and where to place them. Laying the templates so that the overlap remains the same across the bank is best done in a coordinate system which is largely flat across the parameter space. Currently, the metric has been computed for $2\,\mathrm{pN}$ non-spinning templates, which are laid out on a hexagonal grid in τ_0 and τ_3 space. We derived τ_0 in Chapter 3; in terms of the total mass ($\mathrm{M_{total}}$), η and the lower cutoff frequency (f_0) it is:

$$\tau_0 = \frac{5}{256\pi f_0 \eta}(\pi \mathrm{M_{total}} f_0)^{-5/3} \tag{3.43}$$

and τ_3 is [57]:

$$\tau_3 = \frac{1}{8 f_0 \eta}(\pi \mathrm{M_{total}} f_0)^{-2/3} \tag{3.44}$$

We use τ_0 and τ_3 because the metric at $2\,\mathrm{pN}$ is almost flat in these coordinates [57]. Hexagonal griding — as opposed to square griding — allows the space to be covered efficiently using a minimal number of templates [58]. Once the templates are laid out we can convert to $\mathrm{M_{total}}$ and η by inverting equations 3.43 and 3.44 to obtain [57]:

$$\mathrm{M_{total}} = \frac{5}{32\pi^2 f_0}\frac{\tau_3}{\tau_0}, \quad \eta = \frac{1}{8\pi f_0 \tau_3}\left(\frac{32\pi \tau_0}{5\tau_3}\right)^{2/3} \tag{3.45}$$

See chapter 5 for a plot of a typical template bank used in S5 and S6.

Limiting the search to non-spinning templates reduces our efficiency of detecting GWs from binaries with spinning components. In parameter space a spinning binary will live in a region above the non-spinning plane. This means the SNR we obtain will be from the projection of the signal onto the non-spinning plane. The resulting loss in

SNR leads to a decrease in sensitivity to spinning signals. Astrophysical observations have indicated that spin should not be much of a concern for BNS systems [59, 60], as the component masses are too small to spin fast enough relative to the total angular momentum to have an affect on the GW in the LIGO and Virgo bands. However, spin does have a stronger effect on NSBH and BBH systems. In S5 we estimated the effect of the spin of these systems on our ability to detect; see Chapter 6 for details. We have done the same for S6, and these results will be presented in [61]. Investigations into how to expand the template bank into non-spinning regions without substantially increasing the false alarm rate are on-going.

3.3   The χ^2 Test

Up to this point we have assumed that the detector output in the absence of a signal is stationary Gaussian noise. Unfortunately, the real detectors have a number of non-Gaussian glitches due to various instrumental and environmental factors. Although these glitches do not have the same morphology as a gravitational wave from a CBC, they do cause the matched-filter to "ring off" high-SNR triggers. Figure 8 shows a histogram of trigger counts as a function of SNR (ρ) taken from H1 during two weeks of S6. For reference, a χ^2 distribution with two degrees of freedom — expected if the detector noise is Gaussian — is shown. Clearly, a large non-Gaussian tail is present in the data.

To better discriminate between potential signals and noise, we employ a χ^2 test. This test is described in detail in [32]. The basic idea of this test is relatively straightforward. Although a glitch can cause a trigger with the same (or larger) SNR as a signal, the manner in which the SNR is accumulated over time and frequency will most likely differ. For example, a delta function-like glitch will have a burst of power in a small window in the time domain, and the power will be smeared out across all frequencies. A CBC GW, on the other hand, will accumulate its power across the duration of the template, and it will do so in a "chirping" manner. Therefore, if we break the template into frequency bins of equal power, filter each of these sub-templates, and compare the SNR in each bin to the expected value, we can better discriminate glitch from signal.

Define a set of p matched filters such that the ith filter is carried out between the

frequencies f_i and f_{i+1}:

$$(h_i|s) = \int_{f_i}^{f_{i+1}} \frac{\widetilde{h}^*(f)s(f)}{S_n(|f|)}\, \mathrm{d}f, \quad i \in \{0 \ \ldots \ p-1\}, \ f \in [f_0, f_{\text{isco}}] \tag{3.46}$$

(Note that on the left-hand side of the equation we put the i index on h. This is because we could equally as well have broken the template up into each frequency bin, then filter each sub-template across the entire frequency range.) Each frequency bin is chosen so that the template has equal amounts of power in each bin. The amount of SNR accumulated in one bin is thus:

$$\rho_i = \sqrt{\frac{(h_{+i}|s)^2 + (h_{\times i}|s)^2}{(h|h)}}$$
$$= \frac{1}{p}\rho \tag{3.47}$$

where ρ is the total SNR. We can therefore check how well the measured SNR matches the expected SNR in a single bin by:

$$\Delta\rho_i = \rho_i - \frac{\rho}{p}$$

To check the match across the entire template, we define the χ^2 test as [32]:

$$\chi^2 = p \sum_{i=1}^{p} \left| \varrho_i - \frac{\varrho}{p} \right|^2 \tag{3.48}$$

Note that we have used the complex form of the SNR, defined in equation 3.37, and we took the modulus after the subtraction. This is done so we can directly compare each polarization of the phase between the measured and the expected SNRs.

Let us examine what values of χ^2 to expect for different situations. First, assume that the detector noise is Gaussian and that the data has a signal in it from a CBC at an effective distance $\mathcal{D}'$: $s(t) = n(t) + h'(t)$. For simplicity we will assume the signal is $+$ polarized. (Even if it was not, we could imagine doing a Gram-Schmidt orthogonalization of our templates to align one of the polarizations with that of the signal.) We will also assume, for now, that the signal matches one of our templates exactly. With these assumptions:

$$(h_+|h') = \mathcal{D}'^{-1}(h_+|h_+) = \sigma^2 \mathcal{D}'^{-1} \tag{3.49}$$

$$(h_{+i}|h') = \frac{1}{p\mathcal{D}'}(h_+|h_+) = \frac{\sigma^2}{p\mathcal{D}'} \tag{3.50}$$

$$(h_\times|h') = (h_{\times i}|h') = 0 \tag{3.51}$$

Here, h_{+i} and $h_{\times i}$ are the plus and cross polarization templates, respectively, in the ith bin. We wish to know what the mean χ^2 is in this case. In order to compute it, first note that:

$$\overline{\varrho_i^* \varrho} = \overline{\frac{z_i^* z}{\sigma^2}}$$

$$= \frac{\overline{(h_{+i}|n+h')(h_+|n+h')} + \overline{(h_{\times i}|n+h')(h_\times|n+h')}}{\sigma^2}$$

$$= \frac{\overline{(h_{+i}|n)(h_+|n)} + \overline{(h_{\times i}|n)(h_\times|n)} + (h_{+i}|h')(h_+|h')}{\sigma^2}$$

$$= \frac{2}{p} + \frac{1}{p}\xi'^2 \tag{3.52}$$

$$\overline{\varrho_i^* \varrho_i} = \overline{\frac{z_i^* z_i}{\sigma^2}}$$

$$= \frac{2}{p} + \frac{1}{p^2}\xi'^2 \tag{3.53}$$

$$\overline{\varrho^* \varrho} = \overline{\frac{z^* z}{\sigma^2}}$$

$$= 2 + \xi'^2 \tag{3.54}$$

where:

$$\xi'^2 = \frac{\sigma^2}{\mathcal{D}'^2} \tag{3.55}$$

The mean χ^2 is therefore:

$$\overline{\chi^2} = p\sum_{i=1}^{p} \overline{\left| \varrho_i - \frac{\varrho}{p} \right|^2}$$

$$= p\sum_{i=1}^{p} \left(\overline{\varrho_i^* \varrho_i} - \frac{2}{p}\overline{\varrho_i^* \varrho} + \frac{1}{p^2}\overline{\varrho^* \varrho} \right)$$

$$= p\sum_{i=1}^{p} \left(\frac{2}{p} + \frac{\xi'^2}{p^2} - \frac{4}{p^2} - 2\frac{\xi'^2}{p^2} + \frac{2}{p^2} + \frac{\xi'^2}{p^2} \right) \tag{3.56}$$

$$= 2p - 2 \tag{3.57}$$

Since n is Gaussian, $(h_+|n)$ and $(h_\times|n)$ will each be Gaussian also; we can therefore see that the χ^2 test in this case forms a classic χ^2 distribution (hence the name) with $2p - 2$ degrees of freedom.[2] We will find it useful to define the *reduced* χ^2:

$$\chi_r^2 \equiv \frac{\chi^2}{2p - 2} \tag{3.58}$$

---

[2]For a proof, see Appendix A of [32].

so that

$$\overline{\chi_r^2} = 1$$

in the case of Gaussian noise.

The fact that the mean χ^2 is independent of ξ' is significant. It means that for a well matched signal, χ_r^2 will be one regardless of the parameters or distance to the source that emitted the signal. It also means that if the detector contains only Gaussian noise, the χ^2 test has no effect. That is expected, since in Gaussian noise the matched filter is the best discriminator, and their is no need for anything but SNR to separate signal from noise. However, this result was dependent on the signal matching one of our templates exactly. In practice, that most likely will not be the case. For one, the dicreteness of the template bank will mean that a signal will most likely not match a template exactly. Additionally, since the templates are calculated using truncated post-Newtonian expansions, there will be some difference in the template and what Nature provides.

To investigate the effect of the mismatch, let us now assume that the signal does not match our best template perfectly. Without loss in generality, we again assume the signal has only plus polarization. In this case:

$$(h_+|\widetilde{h}') = \mathcal{D}'^{-1}(h_+|h_+) = \frac{\sigma^2}{\mathcal{D}'} \tag{3.59}$$

$$(h_{+\,i}|\widetilde{h}') = (1 - \epsilon_i)\mathcal{D}'^{-1}(h_+|h_+) = (1 - \epsilon_i)\frac{\sigma^2}{p\mathcal{D}'} \tag{3.60}$$

$$(h_\times|\widetilde{h}') = (h_{\times\,i}|\widetilde{h}') = 0 \tag{3.61}$$

where $\widetilde{h}'$ is the true signal (not to be confused with the Fourier Transform), and $(1 - \epsilon_i)$ is the mismatch between the signal and the template in the ith bin. We have not defined an overall mismatch between the template and the signal because that cannot be distinguished from the effective distance. In other words, any overall mismatch will simply make the signal appear to be farther away. Note that the ϵ_i are not necessarily equal. For example, if the template models the signal well during the inspiral phase, but diverges as the binary merges (as is the case with the pN

approximation), ϵ_i will grow with frequency. Equations 3.52 – 3.54 become:

$$\overline{\varrho_i^* \varrho} = \frac{2}{p} + \frac{(1 - \epsilon_i)}{p} \xi'^2 \tag{3.62}$$

$$\overline{\varrho_i^* \varrho_i} = \frac{2}{p} + \frac{(1 - \epsilon_i)^2}{p^2} \xi'^2 \tag{3.63}$$

$$\overline{\varrho^* \varrho} = 2 + \xi'^2 \tag{3.64}$$

$\overline{\chi^2}$ is therefore:

$$\overline{\chi^2} = 2p - 2 + \xi'^2 \sum_{i=1}^{p} \frac{\epsilon_i^2}{p} \tag{3.65}$$

From equation 3.36 we can see that ξ' is like a SNR: it is what the SNR of the signal would be if Gaussian effects are neglected. The $\sum_{i=1}^{p} \epsilon_i^2 / p$ term therefore couples the χ^2 to the SNR of the signal.

Although we arrived at this solution by considering the mismatch between a signal and a template, we could equally as well have considered $\widetilde{h}'$ to be a glitch. In that case, ϵ_i gives the match between the glitch and the template. Note that the $\epsilon_i \in (-\infty, \infty)$; i.e., they can be positive *or* negative, and they do not have to be less than one. An ϵ_i that is negative or greater than one means that the signal *over matches* the template in a given frequency bin. This is what we would expect if the signal was a glitch, for example: a few of the ϵ_i would be large, while all the others would be close to one. Thus a glitch would lead to a large χ^2.

The coupling between χ^2 and SNR means that both signals and glitches will have larger χ^2 with larger SNR. However, as we have set the minimal match to be > 0.97, $\epsilon_i < 0.3 \ \forall i$ due to the dicreteness of the bank. Monte carlo simulations of non-spinning signals have shown that mismatches due to differences in post-Newtonian orders are also relatively small. Thus, for (non-spinning) signals, χ^2 is weakly coupled to the SNR. Glitches, on the other hand, are expected to have large mismatches, resulting in a strong coupling between χ^2 and SNR. The overall effect is that signals and glitches get increasingly separated with increasing SNR. Indeed, this is borne out in the data. Figure 9(a) shows the H1 triggers shown in Figure 8 plotted in χ^2 versus ρ. The black crosses are "background" triggers — they are what survived slide coincidence (see Chapter 4 for details); the red crosses are "injections", i.e., simulated signals (see Chapter 5 for details). The separation at high SNR is clear; at low SNR it becomes difficult to distinguish the two.

Since spinning signals do not match the template bank as well as non-spinning, they tend to couple more strongly to their SNR. This can be seen in Figure 9(b), in which all the injections are spinning. Although many of them still separate from the background, the separation is not as clear. Note that the range of SNRs of the spinning signals is as large as the non-spinning; i.e., spinning signals, though they do not match the templates as well, can still ring off large SNRs. Yet due to the strong coupling between SNR and χ^2, they are "punished" by χ^2.

3.3.1 Applying χ^2

χ^2 is clearly a good discriminator between non-Gaussian noise and signal for known waveforms. We now turn to the question of how to apply this test. Three methods are used in CBC searches: a χ^2 threshold is applied [31], along with the r^2 *veto* [33, 62], and triggers' SNRs are re-weighted based on their χ^2 value via *New SNR* [61].

The most straight-forward use of χ^2 is to veto triggers that have a χ^2 value exceeding a given value. Due to the coupling between χ^2 and SNR for a mismatched signal, the threshold SNR dependent. Specifically, we threshold on [31]:

$$\chi^2 < \chi_*^2 \left(p + \rho^2 \delta^2\right) \tag{3.66}$$

where χ_*^2 and δ^2 are both tunable parameters. Since the noise distribution is unknown these parameters can only be determined using Monte Carlo techniques. Values are chosen conservatively so as to minimize the chance of vetoing GWs. The effect of the χ^2 threshold can be seen in 9.

We also make use of the r^2 *veto*, defined as [33, 62]:

$$r^2 = \frac{\chi^2}{p} \tag{3.67}$$

This is checked over a period of time: if r^2 is above a threshold value for a duration $\pm T$ of the trigger time, the trigger is vetoed. Both the threshold r_*^2 and the time checked, T, are tunable parameters that are determined from Monte Carlo studies [33]. The r^2 veto can target triggers that have a statistical downward fluctuation in χ^2 at the time of the veto. For this reason it can veto triggers that have lower χ^2 values then the threshold; this effect can also be seen in Figure 9.

Perhaps the most effective use of χ^2 has been the implementation of *effective* and *New* SNR. Instead of using χ^2 to veto triggers, we instead aim to re-weight their

SNRs based on their respective χ^2 values. Effective SNR is defined as [21]:

$$\rho_{\text{eff}} = \frac{\rho^2}{\sqrt{\chi_r^2 \left(1 + \frac{\rho^2}{m}\right)}} \tag{3.68}$$

where χ_r^2 is the reduced χ^2, defined above, and m is a tunable parameter which is typically set to 250. Effective SNR was used as the detection statistic for S5. However, it was found that some noise triggers were artificially promoted if they had statistical downward fluctuations in χ_r^2. For this reason, New SNR was developed for S6, defined as [61]:

$$\rho_{\text{n}}^2 = \begin{cases} \rho^2, & \chi_r^2 \leq 1, \\ \frac{\rho^2}{[(1+(\chi_r^2)^3)/2]^{1/6}}, & \chi_r^2 > 1, \end{cases} \tag{3.69}$$

Lines of constant ρ_{n} are shown on Figure 9. As can be seen, they are particularity effective at separating noise triggers from injections. Figure 10 shows a histogram of the same triggers shown in Figure 8, this time plotted against New SNR. Most of the non-Gaussian tail has been down-weighted, and moved below the SNR threshold (in this case, 5.5). For reference, a χ^2 distribution with two degrees of freedom is plotted. There is still some excess however; for this, we apply a coincidence test, described in the next section, and data-quality vetoes, discussed in chapter 5.

3.4   Coincidence Testing

To further reduce the number of non-GW triggers we employ a *coincidence test* across detectors [31, 63]. We expect gravitational waves to be the only correlated trigger source across detectors that are not co-located. Therefore, if we get a trigger in one detector, we can check the other detectors to see if a trigger exists. If none does, we can be confident the trigger was from a noise source, and so we can discard it.[3]

Since the detectors are not co-located, we must allow for a discrepancy in the time-of-arrival of signals that is on order of the light-travel time between the detectors. Additionally, since the template bank is constructed separately for each detector, the trigger is likely to be found with slightly different parameters in each detector. We

---

[3]Due to the antenna pattern of the detectors, and due to varying sensitivities, it is possible to have a GW in one detector and none in another. The chance of this happening decreases with the number of detectors in the network, however.

account for these differences by constructing an error, or *e-thinca*, ellipsoid around each trigger [34]. Recall equation 3.41, which related how quickly the overlap falls off around a point in parameter space to the metric on the signal manifold. We can use the metric to calculate the "volume", $\mathcal{E}(\theta^\mu)$, of parameter space within which a signal with parameters θ^μ could be expected to fall to some probability.[4] The surface of this volume is defined by [34]:

$$g_{\mu\nu}\,\mathrm{d}\theta^\mu\,\mathrm{d}\theta^\nu = \left(\frac{R}{\rho}\right)^2 \tag{3.70}$$

where R is determined by the target probability and ρ is the SNR of the signal. That the size of the volume depends inversely on SNR gels with expectation: the higher the SNR of the signal, the more confident we are in its recovered parameters; therefore, the smaller the size of the ambiguity ellipse.

By constructing error ellipses around the triggers in each detector we can look for coincidence. If the ellipsoid of a trigger in one IFO overlaps with the ellipsoid in the other, they are considered to be coincident. We parameterize the size of the ellipsoids to construct by an *e-thinca parameter*, which is related to the parameter R in the above equation.[5] However, the e-thinca parameter currently in use in CBC searches is not SNR dependent [34]. The size of the e-thinca parameter was chosen based on tuning studies. It was found [62] that an e-thinca parameter of 0.5 maximized the number of injections recovered while minimizing the number of noise coincidences, or false alarms. This value for e-thinca was used throughout S5 and S6.

Once coincidence has been drawn between one or more single-detector triggers, we can define a *coincident trigger* with *combined* parameters and statistics.[6] The combined values are calculated from the single-detector triggers. Parameters of the template from which the trigger came — e.g., chirp mass and total mass — are combined by taking the average of all the single-detector parameters in the coincident triggers. Combined statistics, such as *combined New SNR*, are calculated by taking

---

[4]There is one minor difference in the metric used to construct the e-thinca ellipsoid and that used to construct the template bank. The e-thinca ellipsoid adds a time dimension to the metric so that differences in the time-of-arrival of signals can be accounted for [34].

[5]How the overlap between two ellipsoids is determined and how exactly R relates to a desired probability is beyond the scope of this thesis. For more details, see [34].

[6]Note that here, and in the rest of this thesis, we will differentiate *coincident* triggers from *single-detector* triggers.

the quadrature sum of the single-detector statistics. For example, if a H1 single-detector trigger had a New SNR of 7.2 and a L1 single-detector trigger had a New SNR of 8.3, then — if the two triggers were coincident — their coincident trigger would have a combined New SNR of:

$$\rho_{\text{nc}} = \sqrt{\rho_{\text{n,H1}}^2 + \rho_{\text{n,L1}}^2} = \sqrt{(7.2)^2 + (8.3)^2} = 11.0$$

How other coincident parameters are calculated — such as the time of coincident trigger — are discussed in section 5.5.1 of Chapter 5.
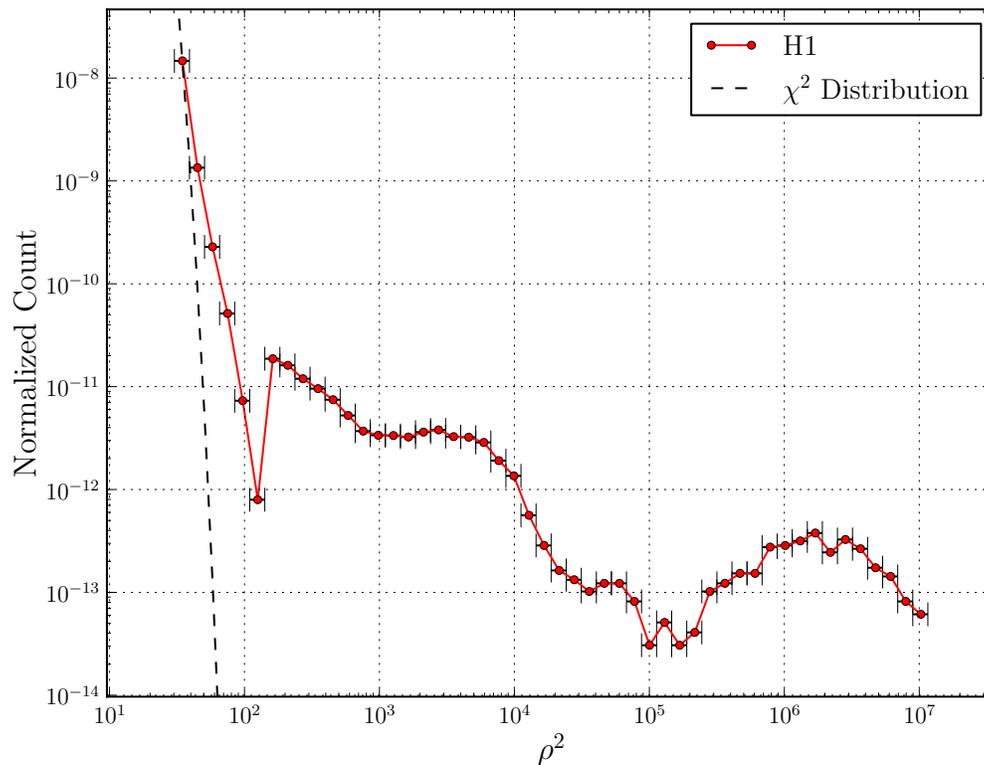
Figure 8 : Histogram of triggers as a function of SNR. Data is taken from two weeks of H1 data taken during S6. The mass range used for this plot is $M_{total} \in [2, 25)M_\odot$; see chapter 5 for details on the templates used. This plot was created using a single-stage pipeline, as discussed in 9, rather than the two-stage pipeline discussed in 5. This was done so as to remove complications from the intermediate coincidence stage. A χ^2 distribution with two degrees of freedom, expected if the detector output were Gaussian, is also plotted. Due to the SNR cutoff, it is difficult to determine the proper normalization. Here, we have normalized the data so that the bin with the largest count was made to lay on the χ^2 distribution. In this case, the data was multiplied by $\sim 10^{-14}$. The bin boundaries used are given by the x-error bars.
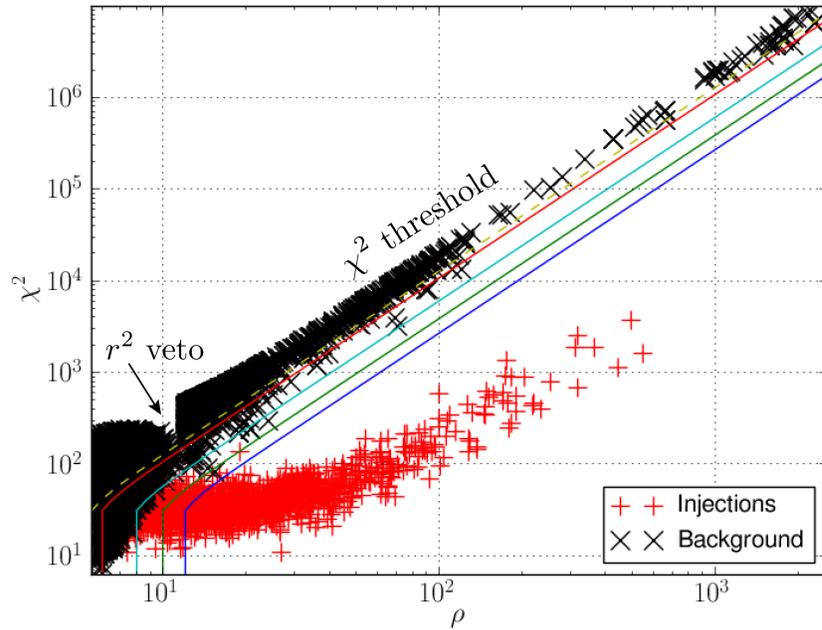
(a) Non-spinning injections
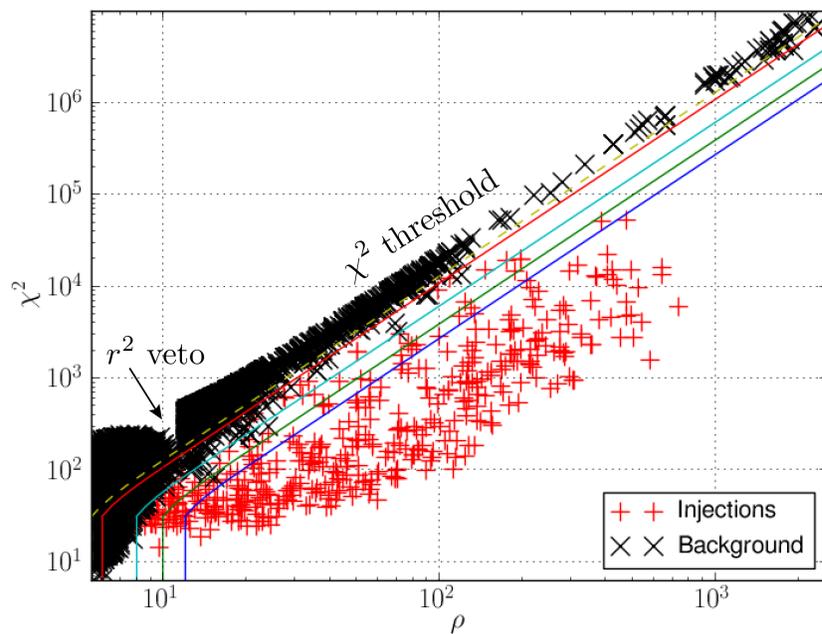
(b) Spinning injections

Figure 9 : χ^2 versus SNR of H1 triggers taken from two weeks of S6. Black crosses indicate "background" triggers, which are triggers that survived coincidence in time slides (see section 3.4 and Chapter 4 for details). Red crosses indicate "software injections" (see Chapter 5 for details). Colored lines indicate points of constant New SNR. The effect of the r^2 veto and χ^2 threshold are marked.
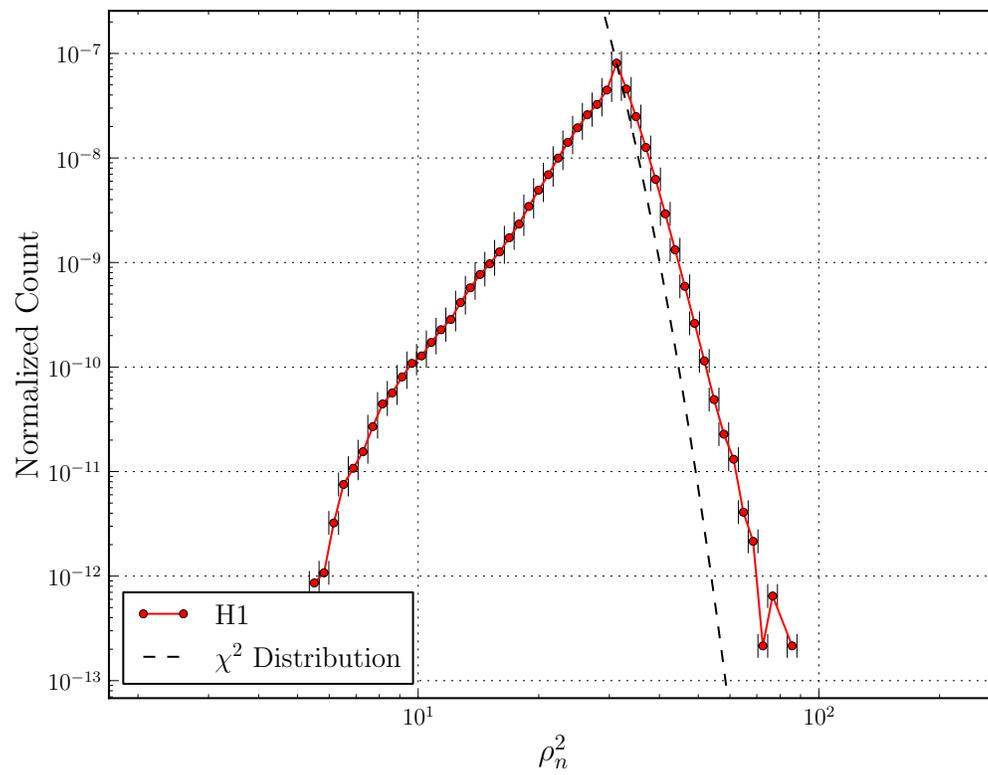
Figure 10 : Histogram of triggers as a function of New SNR. Data is the same as in 8. A χ^2 distribution with two degrees of freedom is plotted. As in Figure 10, the normalization is chosen that the bin with the largest count lies along the distribution. In this case, the data was multiplied by 10^{-13}.

Chapter 4

Ranking Triggers by False Alarm Rate

Once triggers have been obtained from the matched filter, re-weighted using the χ^2 statistic, and vetted via coincidence test, they need to be evaluated for their statistical significance. In the low-mass CBC search we do this by computing a FAR for each coincident trigger we obtain. This chapter details how that is done. We begin by reviewing properties of a Poisson distribution. Next, in section 4.2, we model the expected number of coincident triggers from a single template using the single-detector distribution of triggers. In section 4.3 we show how time sliding data between detectors allows us to estimate a background in order to measure a FAR. Section 4.4 describes how to extend the time-slide method across the template bank to compute a *combined FAR* for all triggers in a search. In section 4.5 we describe an alternate method to compute the combined FAR. Finally, in section 4.6 we summarize the time-slide method in an algorithm that can be used for computing FARs for a CBC search.

4.1 Poisson Process

Imagine that we have a detector with which we can measure triggers emitted from an arbitrary source. Let us assume that the triggers that the source emits occur at random intervals, and that the time at which a trigger is emitted is independent of the last event.

Now imagine that we perform an experiment of duration τ, during which we use our detector to monitor and count the cumulative number of events produced by the source as a function of time, $\Lambda(t)$. Let us repeat the experiment several times, taking the average $\Lambda(t)$ at set time intervals. If the source was *stationary*, we would find that $\overline{\Lambda}(t)$ is a linear function of time. In other words, the mean number of events a stationary source produces in several (hypothetical) iterations over the same period of time, τ, is the same as the mean number produced in several iterations over any other period of equal duration. If the source is *non-stationary*, the time dependence will be non-linear; i.e., two independent periods of time of equal duration would produce a different mean number of events.

To get a duration-independent quantity, we define the source's *rate density*, $\mathcal{R}(t)$, which is the derivative of $\Lambda(t)$ with respect to time:

$$\mathcal{R}(t) \equiv \frac{\mathrm{d}\Lambda}{\mathrm{d}t} \tag{4.1}$$

For stationary sources, $\mathcal{R}$ will be constant; for non-stationary sources, it will have a time dependence. The rate density is an *intrinsic* parameter of the source: it depends on the source's physical characteristics and is independent of the duration of time it is observed for. (Contrast this with Λ, which mixes the intrinsic parameters with the period of time that the source is observed, which is an *extrinsic* parameter). Whether or not a source produces a trigger in a given period of time is random; thus any measurement of the rate density is subject to uncertainty. We therefore define the "true" rate density as the value obtained if an infinite number of independent measurements with identical initial conditions were carried out over the same duration of time:

$$\widetilde{\mathcal{R}}(t) \equiv \lim_{N_m \to \infty} \frac{\sum_{k=1}^{N_m} \widehat{\mathcal{R}}(t)_k}{N_m} \tag{4.2}$$

In this analysis we will denote "true" rate densities by a tilde (e.g., $\widetilde{\mathcal{R}}$), measured values by a hat (e.g., $\widehat{\mathcal{R}}$), and average values by a bar (e.g., $\overline{\mathcal{R}}$).

If the source is stationary, then $\widetilde{\mathcal{R}}$ exists (at least theoretically; in practice, of course, we cannot perform an infinite number of measurements) and can be approximated by measuring $\mathcal{R}$ in a series of experiments. If we observe a source for a period of time t, then, from 4.1, the mean number of events produced during the experiment is:

$$\begin{aligned}
\Lambda &= \int_0^\Lambda d\Lambda' \\
&= \int_0^t \widetilde{\mathcal{R}}(t')\,dt' \qquad (4.3) \\
&\overset{=}{\underset{\widetilde{\mathcal{R}}(t')\to\widetilde{\mathcal{R}}}{}} \widetilde{\mathcal{R}}t \qquad (4.4)
\end{aligned}$$

Furthermore, if the source is stationary, then the probability distribution of getting k events from this system in the same period of time t is given by the well-known Poisson distribution [64]:

$$P(k|\Lambda) = \frac{\Lambda^k e^{-\Lambda}}{k!} \qquad (4.5)$$

From this, we find that $\widetilde{\mathcal{R}}t$ is also the expectation value of the number of events [64]:

$$\begin{aligned}
\mathbb{E}(N) &= \sum_{k=1}^\infty k P(k|\Lambda) \\
&= \sum_{k=1}^\infty k \frac{(\widetilde{\mathcal{R}}t)^k e^{-\widetilde{\mathcal{R}}t}}{k!} \\
&= \widetilde{\mathcal{R}}t e^{-\widetilde{\mathcal{R}}t} \sum_{k=1}^\infty \frac{(\widetilde{\mathcal{R}}t)^{k-1}}{(k-1)!} \\
&= \widetilde{\mathcal{R}}t e^{-\widetilde{\mathcal{R}}t} e^{\widetilde{\mathcal{R}}t} \\
&= \widetilde{\mathcal{R}}t \qquad (4.6)
\end{aligned}$$

Thus, in a single experiment, we can get a measurement of $\widetilde{\mathcal{R}}$ by simply dividing the number of events produced by the period. Averaging these measured values over N_t experiments yields:

$$\begin{aligned}
\overline{\mathcal{R}} &= \frac{\sum_{k=1}^{N_t} t_k \widehat{\mathcal{R}}_k}{\sum_{k=1}^{N_t} t_k} \\
&= \frac{\sum_{k=1}^{N_t} t_k \left(\widehat{N}_k / t_k \right)}{\sum_{k=1}^{N_t} t_k} \\
&= \frac{1}{T} \sum_{k=1}^{N_t} \widehat{N}_k \qquad (4.7)
\end{aligned}$$

where t_k and $\widehat{N}_k$ are the duration and number of triggers produced in the kth experiment, and $\text{T} = \sum_{k=1}^{N_t} t_k$. If the duration of each trial has roughly the same duration, τ, then:

$$\overline{\mathcal{R}} = \frac{\sum_{k=1}^{N_t} \widehat{N}_k}{\tau N_t} \tag{4.8}$$

$$= \frac{\overline{N}}{\tau} \tag{4.9}$$

Since $\overline{N}$ is the mean number of events produced by a Poisson process, the variance is simply $\overline{N}$ [64]. Thus, the error in $\overline{\mathcal{R}}$ is:

$$\delta\overline{\mathcal{R}} = \frac{1}{\tau}\sqrt{\frac{\overline{N}}{N_t}} \tag{4.10}$$

Once we have a measured value for $\mathcal{R}$ we can estimate the probability of getting k events from the source in any period t (assuming the source is stationary); it is:

$$P(k|\overline{\mathcal{R}}, t) = \frac{(\overline{\mathcal{R}}t)^k e^{-\overline{\mathcal{R}}t}}{k!} \tag{4.11}$$

with error:

$$\delta P = \left|\frac{\mathrm{d}P}{\mathrm{d}\overline{\mathcal{R}}}\right| \delta\overline{\mathcal{R}}$$

$$= \frac{t}{\tau}\sqrt{\frac{\overline{N}}{N_t}} e^{-\overline{N}t/\tau} \left|\frac{(\overline{N}t/\tau)^{k-1}}{(k-1)!} - \frac{(\overline{N}t/\tau)^k}{k!}\right| \tag{4.12}$$

In particular, we will be interested in the probability of getting one *or more* events. This is:

$$P(k \geq 1|\overline{\mathcal{R}}, t) = 1 - P(0|\overline{\mathcal{R}}, t)$$

$$= 1 - e^{-\overline{\mathcal{R}}t} \tag{4.13}$$

If the source is non-stationary, then an exact value of $\widetilde{\mathcal{R}}(t)$ does not exist. This is due to the fact that it is impossible to distinguish between fluctuations in the number of triggers produced by a random process from statistical variation and fluctuations

due to a changing rate density. In order to get a better measurement of $\widetilde{\mathcal{R}}$, we must observe it for a longer period of time. However, since we can only observe the source for a finite period of time, this results in a worse measurement of the time dependence of $\widetilde{\mathcal{R}}$, since any fluctuations that occur on a time scale smaller than the period of time we observe for will be indiscernible.

In this analysis, we will assume that the sources we observe are roughly stationary over the periods we observe them for. Although the interferometers do change over time, we leave it to other indicators, such as environmental and instrumental monitors, to inform us when these changes happen so that we can adjust our observation time accordingly.

4.2   Modelling the Expected Number of Coincident Triggers

Consider a coincident trigger produced by a network of GW detectors with *combined* New SNR $\rho^{\dagger}$.[1] We wish to know the significance of the event; that is, we want to know the probability that the event was created by a specific source. To determine that probability we have two choices: we can compare the event to the distribution of the desired source's triggers or we can compare the event to a distribution of triggers from all other, *background*, sources, which gives the *false alarm probability*. Since we do not know *a priori* the distribution of gravitational-wave triggers in the detectors (IFOs), in CBC searches we aim to compute false alarm probabilities. We define the false alarm probability, P_F, as being the probability of getting one or more triggers with a combined New SNR $\geq \rho^{\dagger}$ from a background distribution of triggers in the time searched, T_f (where "f" is used for *foreground*). From equation 4.13 this is:

$$P_F(k \geq 1 | \mathcal{F}(\rho^{\dagger}), T_f) = 1 - e^{-\mathcal{F}(\rho^{\dagger}) T_f} \tag{4.14}$$

$\mathcal{F}(\rho^{\dagger})$ is the false alarm rate of a trigger with combined New SNR $\rho^{\dagger}$. It is the rate density of all background coincident triggers; i.e., it is the rate density of all coincident triggers occurring from every source *except* gravitational waves. Since we do not

---

[1]Elsewhere in this thesis we denote combined New SNR by ρ_{nc}, reserving ρ for single-detector SNR. We will be making use of a number of subscripts in this chapter, however. For notational simplicity, we drop the subscript on combined New SNRs here, and denote them by ρ. We will not discuss SNR in this chapter, only New SNR.

have an analytic model for the interferometers' noise sources we must measure $\mathcal{F}(\rho^\dagger)$ directly. To see how this is done, we first consider the true rate density of coincident triggers from *all* sources in the IFOs (gravitational waves included), $\widetilde{\mathcal{R}}_{\text{all}}(\rho^\dagger)$. We then model the coincidence algorithm to derive an equation for this parameter. For simplicity, we only consider a single template; multiple templates are discussed in section 4.4.

Assuming stationary sources, in a single experiment of duration T_f, $\widetilde{\mathcal{R}}_{\text{all}}(\rho^\dagger)$ is given by:

$$\widetilde{\mathcal{R}}_{\text{all}}(\rho^\dagger) = \frac{\mathbb{E}\left(N_{\text{uncorr}}(\rho^{\dagger 2} \leq \sum_i \rho_i^2), T_f\right) + \mathbb{E}\left(N_{\text{corr}}(\rho^{\dagger 2} \leq \sum_i \rho_i^2), T_f\right)}{T_f} \quad (4.15)$$

Here, $\mathbb{E}\left(N_{\text{uncorr}}(\rho^{\dagger 2} \leq \sum_i \rho_i^2, T_f)\right)$ is the expected number of triggers from all *uncorrelated* sources that give a combined ρ^2 greater than or equal to $\rho^{\dagger 2}$ in time T_f. Uncorrelated means that a source that causes a trigger in one detector has no effect on the others. Conversely, $\mathbb{E}\left(N_{\text{corr}}(\rho^{\dagger 2} \leq \sum_i \rho_i^2, T_f)\right)$ is the expected number of *correlated* coincident triggers, which means they come from a source that causes triggers in all of the detectors.

If we have N_d detectors and N_s independent trigger sources in each detector, then the expected number of *uncorrelated* sources is given by:

$$\mathbb{E}\left(N_{\text{uncorr}}\left(\rho^{\dagger 2} \leq \sum_i \rho_i^2, T_f\right)\right)$$

$$= \int\limits_{\substack{\rho^{\dagger 2} \leq \sum_i \rho_i^2, \\ \rho_i \geq a \; \forall i}}^{\infty} \int\limits_{\substack{|t_i - t_l| \leq \mathfrak{T}_{i,l} \\ \forall l \neq i}}^{T_f} \prod_i^{N_d} \sum_j^{N_s} \mathbb{E}\left(n_{i,j}(\rho_i, t_i)\right) \, dt_i \, d\rho_i \quad (4.16)$$

$$= \int_{\mathcal{S}(\rho^\dagger)} \int_{\mathcal{T}(T_f)} \prod_i^{N_d} \sum_j^{N_s} \sum_{k=1}^{\infty} k \, P_{i,j}(k|\rho_i, t_i) \, dt_i \, d\rho_i \quad (4.17)$$

Here, the $n_{i,j}$ is the "number density" of triggers in the ith detector from the jth source per unit ρ_i per unit time. The $P_{i,j}(k|\rho_i, t_i)$ is the probability of getting k triggers from the jth source in the ith detector per unit ρ_i per unit time. (In going from equation 4.16 to equation 4.17 we have used the fact that the expected number of triggers produced by a source with PDF $P(k)$ is $\sum_{k=1}^{\infty} k \, P(k)$.) The regions of integration over ρ and t ($\mathcal{S}$ and $\mathcal{T}$, respectively) for a two-detector network are shown in Figure

11. The SNR integral is carried out such that the quadrature sum of the single-IFO New SNRs are $\geq \rho^\dagger$, with a lower-cutoff at a. This lower cut-off is the SNR cut that we impose in the matched-filter search; typically $a = 5.5$.[2] The time integral is carried out such that for each point in time in a given detector, t_i, we only integrate between $t_i \pm \mathfrak{T}_{i,l}$ in every other detector. $\mathfrak{T}_{i,l}$ is the duration of the coincidence window between the ith and the lth detector for the template we are considering. As discussed in chapter 3 it is an SNR-dependent quantity. However, since it is mostly dominated by the light-travel time between the ith and l^{th} IFO, here we approximate it to be a constant for each (i, l) pair of detectors.

$P_{i,j}(k|\rho_i, t_i)$ is a two-dimensional PDF: it has some distribution in time and some distribution in ρ. If we assume the sources' time and ρ dependence are independent of each other, then PDF is:

$$P_{i,j}(k|\rho_i, t_i) = P(k|\widetilde{r}_{i,j}, t_i) P_{i,j}(\rho_i) \tag{4.18}$$

In words: the probability of getting k triggers from the jth source in the ith detector with New SNR ρ_i in time t_i is the probability of getting k triggers from a source with rate-density $\widetilde{r}_{i,j}$ in time t_i times the probability of getting a trigger with New SNR ρ_i from the source. If we again assume that all of the sources are stationary Poisson processes in the time domain, then:

$$\sum_{k=1}^{\infty} k P(k|\widetilde{r}_{i,j}, t_i) \, \mathrm{d}t_i = \widetilde{r}_{i,j} \, \mathrm{d}t_i \tag{4.19}$$

With this assumption, the time integral can be carried out independent of the integral over ρ_i; it gives the volume bounded by a hyper-surface, which we designate $\mathcal{V}$:

$$\mathcal{V}(\mathrm{T_f}) \equiv \int\limits_{\substack{|t_i - t_l| \leq \mathfrak{T}_{i,l} \\ \forall l \neq i}}^{\mathrm{T_f}} \prod_i^{N_d} \mathrm{d}t_i \tag{4.20}$$

$\mathcal{V}$ has units of $[\text{time}]^{N_d}$. In the two detector case shown in Figure 11 $\mathcal{V}(\mathrm{T_f}) = \mathfrak{T}_{\mathrm{H,L}}(2\mathrm{T_f} - \mathfrak{T}_{\mathrm{H,L}})$. Plugging $\mathcal{V}$ into equation 4.16 we have:

---

[2]Although we do not apply a New SNR cut in the current search, for simplicity, we will assume that we do, and that it is the same as the SNR cut.

$$\mathbb{E}\left(N_{\text{uncorr}}\left(\rho^{\dagger 2} \le \sum_i \rho_i^2, \text{T}_\text{f}\right)\right) = \mathcal{V}(\text{T}_\text{f}) \int_{\mathcal{S}} \prod_i^{N_d} \sum_j^{N_s} \widetilde{r}_{i,j} P_{i,j}(\rho_i) \, \mathrm{d}\rho_i \quad (4.21)$$

To estimate the expected number of coincident triggers with combined New SNR $\ge \rho^\dagger$ from *correlated* sources, we assume that the sources create a trigger at the same time (modulo the light-travel time) in all detectors. Since we compute combined New SNR by taking the quadrature sum of the single-detector New SNRs, the expected number is the quadrature sum over the individual probabilities:

$$\mathbb{E}\left(N_{\text{corr}}\left(\rho^{\dagger 2} \le \sum_i \rho_i^2, \text{T}_\text{f}\right)\right)$$

$$= \sum_{j=1}^{N_s} \sqrt{\sum_{i=1}^{N_d} \left(\int_{\mathcal{S}(\rho^\dagger)} \int_0^{\text{T}_\text{f}} \mathbb{E}\left(n_{i,j}(\rho_i, t_i)\right) \, \mathrm{d}t_i \, \mathrm{d}\rho_i\right)^2} \quad (4.22)$$

$$= \text{T}_\text{f} \sum_{j=1}^{N_s} \sqrt{\sum_{i=1}^{N_d} \left(\widetilde{r}_{i,j} \int_{\mathcal{S}(\rho^\dagger)} P_{i,j}(\rho_i) \, \mathrm{d}\rho_i\right)^2} \quad (4.23)$$

In going from 4.22 to 4.23 we have used equation 4.18 and have assumed the correlated sources are stationary. Since all triggers will occur in each detector within the light-travel time between them the constraints are removed from the time integral, making it simply T_f. If we further assume that the correlated sources create the same single-detector New SNR in the all detectors, then we can simply the expression to:

$$\mathbb{E}\left(N_{\text{corr}}\left(\rho^{\dagger 2} \le \sum_i \rho_i^2, \text{T}_\text{f}\right)\right)$$

$$= \text{T}_\text{f} \sum_{j=1}^{N_s} \sqrt{N_d} \widetilde{r}_j \int_{\rho^\dagger/\sqrt{N_d}}^{\infty} P_j(\rho_i) \, \mathrm{d}\rho_i \quad (4.24)$$

Here, we have removed the i subscript since the rate parameter and New SNR distribution will be the same in all detectors.

Now let us assume that the only source that can create correlated triggers across detectors is a gravitational wave. In this case, the sum over the number of sources in equation 4.24 becomes a single term, with $j = $ GW. Assuming a GW creates perfectly correlated triggers in all detectors is a bit of an oversimplification: due to the antenna patterns of the detectors and their varying sensitivities, a gravitational

wave will only generate a trigger in all detectors with the same SNR for certain sky locations and orientations. However, since we are ultimately interested in computing background rates in this analysis, and not GW rates, we will assume that the detectors are roughly co-located with the same sensitivity. Under this assumption, we can use equation 4.24 to write:

$$\mathbb{E}\left(N_{\text{GW}}\left(\rho^{\dagger 2} \leq \sum_i \rho_i^2, T_f\right)\right) = \widetilde{r}_{\text{GW}}(\vec{\theta}) T_f \sqrt{N_d} \int_{\rho^{\dagger}/\sqrt{N_d}}^{\infty} P_{\text{GW}}(\rho)\, d\rho \qquad (4.25)$$

Note that $\widetilde{r}_j$ has been replaced by $\widetilde{r}_{GW}(\vec{\theta})$. This is the rate of CBCs in the universe; as it depends on the parameters of the source, $\vec{\theta}$, we have made the parameter dependence explicit, even though we are still only considering a single template.

The integral in equation 4.25 gives the sensitivity of the detectors to GW sources. To compute it, we need the dependence of the sensitivity as a function of ρ. This can be obtained as follows: we wish to know the number of sources the detector is sensitive to from here to some distance $D^{\dagger}$. If we assume that binary sources are distributed uniformly throughout the universe (which is a valid assumption for distances greater than $\sim 10\,\text{Mpc}$ and less than $\sim 1\,\text{Gpc}$ [10]), then this is:

$$(2.26)^{-3} \int_0^{D^{\dagger}} D^2\, dD \int d\Omega \qquad (4.26)$$

where $d\Omega$ is the solid angle. The factor $(2.26)^{-3}$ comes from approximating the volume enclosed by the detector's antenna pattern (which is peanut shaped; see Figure 5) by a sphere with a radius equal to the detector's "horizon distance". The "horizon distance" is the distance to a binary with optimal orientation and location (i.e., it is the length of the longest part of the peanut) [50]. As discussed in chapter 3, the sensitive distance is related to New SNR by:

$$D = \frac{\sigma_{\text{GW}}}{\rho} \qquad (4.27)$$

Substituting this into the above, and setting $\rho^{\dagger}/\sqrt{N_d} = \sigma_{\text{GW}}/D^{\dagger}$ we have:

$$(2.26)^{-3} \int_0^D D'^2\, dD' \int d\Omega = \frac{4\pi}{(2.26)^3} \int_{\rho^{\dagger}/\sqrt{N_d}}^{\infty} \frac{\sigma_{\text{GW}}^3\, d\rho}{\rho^4} \qquad (4.28)$$

Thus:

$$P_{\mathrm{GW}}(\rho) = \frac{4\pi\sigma_{\mathrm{GW}}^3}{(2.26)^3}\rho^{-4} \tag{4.29}$$

which gives:

$$\mathbb{E}\left(N_{\mathrm{GW}}\left(\rho^{\dagger 2} \le \sum_i \rho_i^2, \mathrm{T_f}\right)\right) = \frac{4\pi}{3}\frac{\sqrt{N_d}\,\sigma_{\mathrm{GW}}^3}{(2.26)^3(\rho^{\dagger}/\sqrt{N_d})^3}\widetilde{r}_{\mathrm{GW}}(\theta)\mathrm{T_f} \tag{4.30}$$

(Note that we only have one σ_{GW}^3 in the equation. This is because we assumed that all the detectors had the same sensitivity. If we had not made this assumption, the $\sqrt{N_d}\,\sigma_{\mathrm{GW}}^3$ term would instead be $\sqrt{\sum_i^{N_d}\sigma_{i,\mathrm{GW}}^6}$.) Adding this term to equation 4.21 and dividing by $\mathrm{T_f}$ gives the overall rate-density $\widetilde{\mathcal{R}}_{\mathrm{all}}(\rho^{\dagger})$.

As an example, consider two arbitrary detectors, H and L, that are roughly co-located and have similar sensitivities. Let us assume that each IFO has only one non-gravitational wave source, which we call "noise;" let the PDF of the noise in each detector be given by:

$$P_{i,j}(k|\rho_i, t_i) = \widetilde{r}_{i,\mathrm{noise}}\Phi(\rho_i^2, \sigma_{i,\mathrm{noise}}) \tag{4.31}$$

where $\sigma_{i,\mathrm{noise}}^2$ is the variance of the noise and Φ is some distribution in ρ_i^2. With these assumptions in mind, the rate density of coincident triggers from all sources would be:

$$\begin{aligned}
\widetilde{\mathcal{R}}_{\mathrm{all}}&\left(\rho^{\dagger} \le \sqrt{\rho_{\mathrm{H}}^2 + \rho_{\mathrm{L}}^2}\right) \\
&= \frac{\mathfrak{T}_{\mathrm{HL}}(2\mathrm{T_f} - \mathfrak{T}_{\mathrm{HL}})}{\mathrm{T_f}}\widetilde{r}_{\mathrm{H,noise}}\widetilde{r}_{\mathrm{L,noise}}\int_{\mathcal{S}}\Phi(\rho_{\mathrm{H}}^2, \sigma_{\mathrm{H,noise}})\Phi(\rho_{\mathrm{L}}^2, \sigma_{\mathrm{L,noise}})\,\mathrm{d}\rho_{\mathrm{H}}\,\mathrm{d}\rho_{\mathrm{L}} \\
&+ \frac{4}{3}\pi\frac{\sqrt{2}\sigma_{\mathrm{GW}}^3}{(2.26)^3(\rho^{\dagger}/\sqrt{2})^3}\widetilde{r}_{\mathrm{GW}}(\vec{\theta})
\end{aligned} \tag{4.32}$$

4.3   Computing Background Using Time Slides

The first term in equation 4.32 is the false alarm rate-density that we seek; thus, we could re-write the equation:

$$\widetilde{\mathcal{R}}_{\mathrm{all}}\left(\rho^{\dagger} \le \sqrt{\rho_{\mathrm{H}}^2 + \rho_{\mathrm{L}}^2}\right) = \widetilde{\mathcal{F}}(\rho^{\dagger}) + \widetilde{\mathcal{R}}_{\mathrm{GW}}\left(\rho^{\dagger} \le \sqrt{\rho_{\mathrm{H}}^2 + \rho_{\mathrm{L}}^2}\right) \tag{4.33}$$
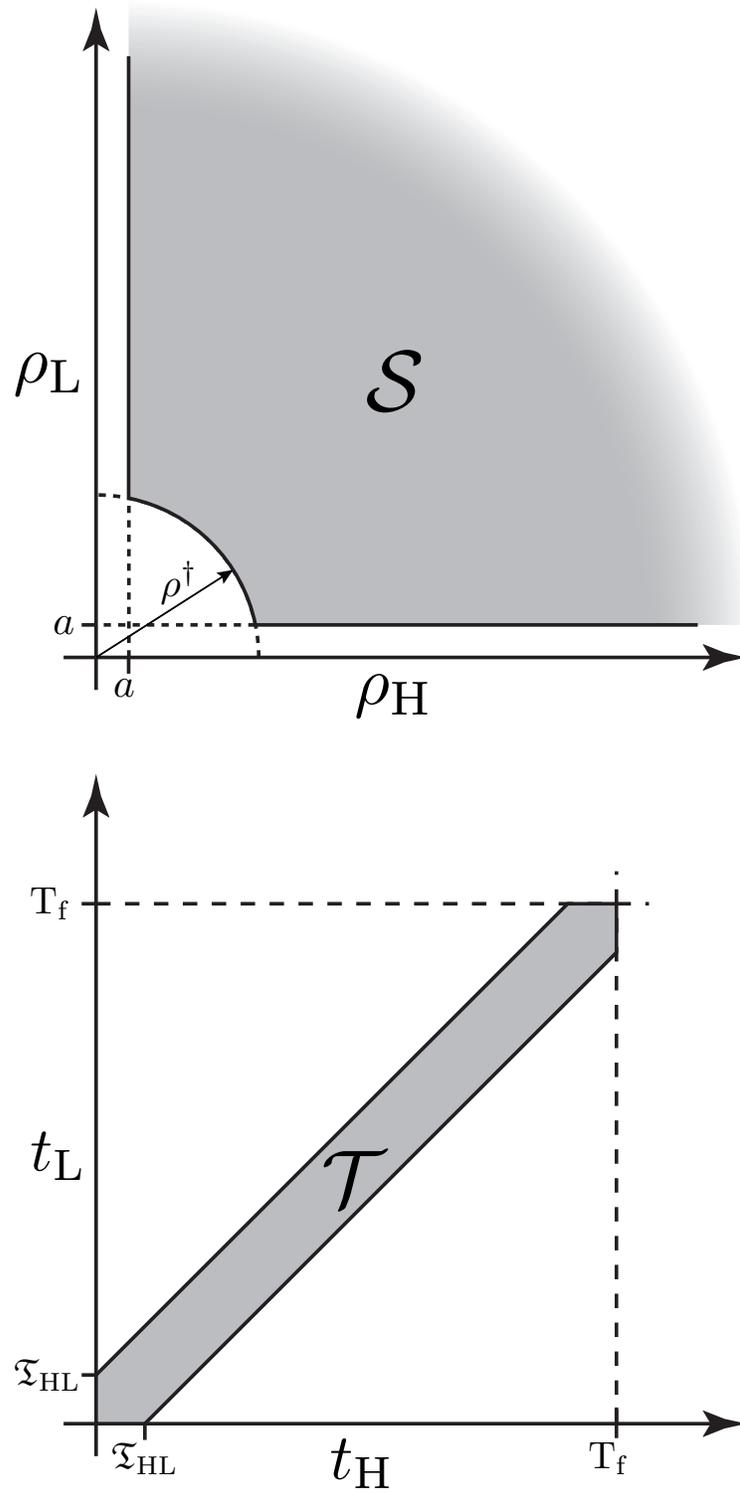
Figure 11 : The regions of integration in ρ, and time for a network of two detectors, H and L.

When we do an experiment and perform the coincidence test, we get a measure of $\widetilde{\mathcal{R}}_{\mathrm{all}}$. To get $\widetilde{\mathcal{F}}$ we must separate out the GW terms. Unfortunately, we cannot tell the difference between a coincident trigger that came from noise and one that came from a gravitational wave. (If we could, we would not have to do any of this.) Thus, we must find other ways to separate the two. One way to do so is to perform a *time slide*.

Consider what happens if we add a relative time offset to each detector's datastream that is larger than the coincidence window between them. Doing so causes the gravitational wave triggers to become uncorrelated across the detectors. The rate of coincident noise triggers, however, is unaffected, since they were uncorrelated anyway. Thus, if we perform the same coincidence test on this *slid* data as we did for the *zero-lag* data, the expected number of coincidences is (assuming each detector only has one noise source):

$$\mathbb{E}\left(N_{\mathrm{all}}\left(\rho^{\dagger 2} \leq \sum_{i=1}^{N_d}, \mathrm{T_f}, \vec{\mathcal{O}}\right)\right) = \mathcal{V}(\mathrm{T_f}) \int_{\mathcal{S}} \prod_{i=1}^{N_d} \left(\widetilde{r}_{i,\mathrm{noise}} P_{i,\mathrm{noise}}(\rho_i) + \widetilde{r}_{\mathrm{GW}} P_{i,\mathrm{GW}}(\rho_i)\right) \, \mathrm{d}\rho_i \tag{4.34}$$

Here we have made the dependence on the time offset explicit by defining an *offset-vector*:

$$\vec{\mathcal{O}} = \begin{bmatrix} 0 & \Delta t_2 & \dots & \Delta t_{N_d} \end{bmatrix} \tag{4.35}$$

(The offset for one of the detectors is always zero, since we are sliding the detectors with respect to each other.)

For example, in our two-detector case, the expected number of coincidences becomes:

$$\begin{aligned}
\mathbb{E} & \left(N_{\mathrm{all}}\left(\rho^{\dagger 2} \leq \rho_{\mathrm{H}}^2 + \rho_{\mathrm{L}}^2, \mathrm{T_f}, \vec{\mathcal{O}}\right)\right) \\
= & \ \mathcal{V}(\mathrm{T_f}) \int_{\mathcal{S}} \mathrm{d}\rho_{\mathrm{H}} \, \mathrm{d}\rho_{\mathrm{L}} \times \\
& \left(\widetilde{r}_{\mathrm{H,noise}} P_{\mathrm{H,noise}}(\rho_{\mathrm{H}}) + \widetilde{r}_{\mathrm{GW}} P_{\mathrm{H,GW}}(\rho_{\mathrm{H}})\right) \left(\widetilde{r}_{\mathrm{L,noise}} P_{\mathrm{L,noise}}(\rho_{\mathrm{L}}) + \widetilde{r}_{\mathrm{GW}} P_{\mathrm{L,GW}}(\rho_{\mathrm{L}})\right) \\
= & \ \mathfrak{T}_{\mathrm{HL}}(2\mathrm{T_f} - \mathfrak{T}_{\mathrm{HL}})(\widetilde{r}_{\mathrm{H,noise}} \widetilde{r}_{\mathrm{L,noise}} \, \mathcal{I}_{\mathrm{H,noise;L,noise}} + \widetilde{r}_{\mathrm{H,noise}} \widetilde{r}_{\mathrm{GW}} \, \mathcal{I}_{\mathrm{H,noise;L,GW}} \\
& + \ \widetilde{r}_{\mathrm{GW}} \widetilde{r}_{\mathrm{L,noise}} \, \mathcal{I}_{\mathrm{H,GW;L,noise}} + \widetilde{r}_{\mathrm{GW}}^2 \, \mathcal{I}_{\mathrm{H,GW;L,GW}})
\end{aligned} \tag{4.36}$$

where:

$$\mathcal{I}_{\text{H,noise;L,noise}} = \int_{\mathcal{S}} \Phi(\rho_{\text{H}}^2, \sigma_{\text{H,noise}}) \Phi(\rho_{\text{L}}^2, \sigma_{\text{L,noise}}) \, d\rho_{\text{H}} \, d\rho_{\text{L}} \tag{4.37}$$

$$\mathcal{I}_{\text{H,noise;L,GW}} = \frac{4\pi\sigma_{\text{GW}}^3}{(2.26)^3} \int_{\mathcal{S}} \rho_{\text{L}}^{-4} \Phi(\rho_{\text{H}}^2, \sigma_{\text{H,noise}}) \, d\rho_{\text{H}} \, d\rho_{\text{L}} \tag{4.38}$$

$$\mathcal{I}_{\text{H,GW;L,noise}} = \frac{4\pi\sigma_{\text{GW}}^3}{(2.26)^3} \int_{\mathcal{S}} \rho_{\text{H}}^{-4} \Phi(\rho_{\text{L}}^2, \sigma_{\text{L,noise}}) \, d\rho_{\text{H}} \, d\rho_{\text{L}} \tag{4.39}$$

$$\mathcal{I}_{\text{H,GW;L,GW}} = \left(\frac{4\pi\sigma_{\text{GW}}^3}{(2.26)^3}\right)^2 \int_{\mathcal{S}} \rho_{\text{H}}^{-4} \rho_{\text{L}}^{-4} \, d\rho_{\text{H}} \, d\rho_{\text{L}} \tag{4.40}$$

We can solve the GW/GW integral[3], getting:

$$\mathcal{I}_{\text{H,GW;L,GW}} =$$
$$\left(\frac{4\pi\sigma_{\text{GW}}^3}{(2.26)^3}\right)^2 \left(\frac{\rho^{\dagger 2} - 2a^2}{18\rho^{\dagger 6} a \sqrt{\rho^{\dagger 2} - a^2}} \left(1 + \frac{\rho^{\dagger 4}}{8a^2 \left(\rho^{\dagger 2} - a^2\right)}\right)\right.$$
$$\left. + \frac{1}{9a^3 \left(\rho^{\dagger 2} - a^2\right)^{3/2}}\right) \tag{4.41}$$

if $\rho^{\dagger} \geq \sqrt{2}a$;

$$\mathcal{I}_{\text{H,GW;L,GW}} = \left(\frac{4\pi\sigma_{\text{GW}}^3}{(2.26)^3}\right)^2 \frac{1}{9a^6} \tag{4.42}$$

otherwise. The other three integrals can be solved numerically, however; see Chapter 9 for details.)

---

[3]Note that if we did not impose the New SNR cut-off at a then the GW/GW integral would give infinity; i.e., we would expect an infinite number of events in the detector from gravitational waves! This is a result of the model we have chosen for GWs. If the limits of integration were $\rho = 0, \infty$, then, from equation 4.27, we would be considering every source from here to infinity. By assuming that the distribution of sources is uniform throughout the universe (which we did by setting $\tilde{r}_{\text{GW}}(\vec{\theta})$ to a constant), this means that if we look for an infinite distance, we will get an infinite number of sources — a GW version of Olbers' paradox [65]. Of course, the real universe is not infinitely old; beyond $\sim 1\,\text{Gpc}$ the number of CBCs begins to fall off [10]. Further, we have assumed that the GW/GW term is independent of the noise. In practice, however, we do clustering, which couples these terms. Even if we had an ideal detector with stationary Gaussian noise in SNR, below some SNR the noise triggers would effectively cluster away all GW events, even if the universe was infinitely old. We get around both of these difficulties by assuming that the SNR cutoff is high enough that these effects do not affect our model (which, with $a = 5.5$, is valid for both enhanced and Advanced LIGO.

By de-coupling the gravitational waves in each detector, the GW/GW term is now much smaller than in the zero-lag case. For example, we can calculate the expected number of GW/GW coincidences for a $1.4/1.4\,\mathrm{M_\odot}$ binary neutron star system in each case. From table 1, the rate of BNS coalescences in the universe is $\widetilde{r}_{\mathrm{GW}} = 1 \times 10^{-6}\,\mathrm{Mpc^{-3}\,yr^{-1}}$. During enhanced LIGO, the $4\,\mathrm{km}$ LIGO interferometers had a peak horizon distance to BNS systems of $\sim 40\,\mathrm{Mpc}$, which gives a sensitivity of $\sigma_{\mathrm{GW}} = \rho D_{\mathrm{horizon}} = 8 \times 40\,\mathrm{Mpc} = 320\,\mathrm{Mpc}$. Thus, if our two hypothetical detectors had the same sensitivity, then in one year we would expect:

$$
\begin{aligned}
\mathbb{E}\left(N_{\mathrm{GW}}\left(\rho^\dagger = 11.3, \mathrm{T_f} = 1\,\mathrm{yr}\right)\right) &= \frac{4}{3}\pi \frac{\sqrt{2}(320\,\mathrm{Mpc})^3}{(2.26)^3(11.3/\sqrt{2})^3}(1 \times 10^{-6}\,\mathrm{Mpc^{-3}\,yr^{-1}})(1\,\mathrm{yr}) \\
&= 0.03
\end{aligned}
$$

gravitational-wave events in zero-lag at a New SNR of 8 in each detector (which corresponds to a combined New SNR of $\rho^\dagger = \sqrt{2}\times 8 = 11.3$). In contrast, the expected number of coincident gravitational-waves in a one year-long time slide would be:

$$
\begin{aligned}
&\mathbb{E}\left(N_{\mathrm{GW}}\left(\rho^\dagger = 11.3, \mathrm{T_f} = 1\,\mathrm{yr}, \mathcal{O}\right)\right) \\
&= (3.2 \times 10^{-10}\,\mathrm{yr})(2\,\mathrm{yr} - 3.2 \times 10^{-10}\,\mathrm{yr})(1 \times 10^{-6}\,\mathrm{Mpc^{-3}\,yr^{-1}})^2 \\
&\quad \times\ \mathcal{I}_{\mathrm{H,GW;L,GW}}(\rho^\dagger = 11.3, a = 5.5) \\
&= 6 \times 10^{-13}
\end{aligned}
$$

Here, we have a coincidence window of $\mathfrak{T}_{\mathrm{HL}} = 10\,\mathrm{ms} \approx 3.2 \times 10^{-10}\,\mathrm{yr}$. This is about the light-travel time between Hanford and Livingston.

As can be seen in equation 4.36, doing the time slide has introduced mixing terms between GWs and noise. However, evaluating the integrals 4.37–4.39 requires knowledge of Φ, and a fit to the data in order to determine $\sigma_{i,\mathrm{noise}}$ and $\widetilde{r}_{i,\mathrm{noise}}$. Here we will assume the noise/GW terms are small relative to the noise/noise term, using the fact that number of expected gravitational waves in enhanced LIGO is less than one as our reasoning.

Since the noise/noise term is the dominate source in a time-slide, we have:

$$
\widetilde{\mathcal{R}}_{\mathrm{all}}\left(\rho^{\dagger 2} \leq \sum_{i=1}^{N_d} \rho_i^2, \mathcal{O}\right) \approx \widetilde{\mathcal{F}}(\rho^\dagger) \tag{4.43}
$$

Thus, we can measure the false alarm rate of a trigger with combined New SNR $\rho^\dagger$ by simply performing a time slide, counting the number of coincidences that have a combined New SNR $\geq \rho^\dagger$, and dividing by the slide duration. This method has the further advantage that we can perform multiple slides with the same data to improve our measurement of $\widetilde{\mathcal{F}}$. If we change the offset of each detector by enough to consider every slide to be independent — i.e., we slide the data enough so that no two events can be coincident in more than one slide — then, from equation 4.7, in N_t slides:

$$\overline{\mathcal{F}}(\rho^\dagger) = \frac{\sum_{k=1}^{N_t} \widehat{N}_k \left(\rho^{\dagger 2} \leq \sum_{i=1}^{N_d} \rho_i^2 \right)}{\mathrm{T_b}} \tag{4.44}$$

where $\widehat{N}_k \left(\rho^{\dagger 2} \leq \sum_{i=1}^{N_d} \rho_i^2 \right)$ is the measured number of coincidences with a combined New SNR $\geq \rho^\dagger$ in the kth slide, and $\mathrm{T_b}$ is the total *background time*. The background time is the sum of the length of each slide; the length of each slide is the intersection of every detector's foreground analysis time after they have been slid. Therefore:

$$\mathrm{T_b} = \sum_{k=1}^{N_t} \left| \left(\mathbb{T}^A + \mathcal{O}_k[A] \right) \bigcap \left(\mathbb{T}^B + \mathcal{O}_k[B] \right) \bigcap \ldots \bigcap \left(\mathbb{T}^{N_d} + \mathcal{O}_k[N_d] \right) \right| \tag{4.45}$$

(Here, $\mathbb{T}^i$ is the analysis segment of the ith detector; $|\mathbb{T}^i| = \mathrm{T_f}^i$.) With a finite data set, a single *linear* slide will not have the same duration as the foreground time since the overlap between detectors will decrease with increasing offsets. If the slide is done on a *ring*, so that data that is slid past the end of the segment is put at the beginning, than a slide will have the same duration as the zero-lag. However, if there are gaps in the data, then even sliding on a ring will result in different durations from slide-to-slide, since the gaps will align differently in each slide. We therefore define the *effective number of slides*, $\widetilde{N}_t$, as the ratio of the total background time to the foreground time:

$$\widetilde{N}_t \equiv \frac{\mathrm{T_b}}{\mathrm{T_f}} \tag{4.46}$$

so that equation 4.44 can be written:

$$\overline{\mathcal{F}}(\rho^\dagger) = \frac{\widehat{N}_{\mathrm{total}} \left(\rho^{\dagger 2} \leq \sum_i \rho_i^2 \right)}{\mathrm{T_f} \widetilde{N}_t} = \frac{\overline{N} \left(\rho^{\dagger 2} \leq \sum_i \rho_i^2 \right)}{\mathrm{T_f}} \tag{4.47}$$

Since the sum over the number of slides in the numerator of equation 4.44 is simply a sum of counts, here we have written it as $\widehat{N}_{\mathrm{total}} \left(\rho^{\dagger 2} \leq \sum_i \rho_i^2 \right)$, which is the total

number of triggers with combined New SNR $\geq \rho^\dagger$ (*false alarms*) in all the slides. Thus, $\overline{N}\left(\rho^{\dagger 2} \leq \sum_i \rho_i^2\right)$ is the average number of false alarms in a single experiment with a duration equal to T_f. Since this is the mean of a stationary Poisson process, from equation 4.10, the error in $\overline{\mathcal{F}}$ is:

$$
\begin{aligned}
\delta \overline{\mathcal{F}}(\rho^\dagger) &= \frac{1}{T_f} \sqrt{\frac{\overline{N}\left(\rho^{\dagger 2} \leq \sum_i \rho_i^2\right)}{\widetilde{N}_t}} \\
&= \frac{1}{T_f} \sqrt{\frac{\widehat{N}_{\text{total}}\left(\rho^{\dagger 2} \leq \sum_i \rho_i^2\right)}{\widetilde{N}_t^2}} \\
&= \frac{\sqrt{\widehat{N}_{\text{total}}\left(\rho^{\dagger 2} \leq \sum_i \rho_i^2\right)}}{T_b}
\end{aligned}
\tag{4.48}
$$

We have arrived at a way to calculate false alarm rates for a single template: perform the coincidence test in several times slides; compute the duration of each time slide; count the number of slide coincidences in all slides that have a combined New SNR greater-than-or-equal to a given foreground New SNR; divide by the total background time.

4.4 Computing FARs with Multiple Templates

To compute a false alarm rate across a bank of templates, we would replace the time integral in equation 4.17 with an integral over the template parameters, $\{\vartheta_m\}$, bounded by the e-thinca ellipsoid at ρ, $\mathcal{E}(\rho)$:

$$
\mathbb{E}\left(N_{\text{uncorr}}\left(\rho^{\dagger 2} \leq \sum_i \rho_i^2, T_f\right)\right) = \int_{\mathcal{S}(\rho^\dagger)} \int_{\mathcal{E}(\rho)} \prod_i^{N_d} \sum_j^{N_s} \prod_m^{n_p} \widetilde{r}_{i,j} P_{i,j}(\rho_i, \{\vartheta_m\}) \, d\vartheta_m \, dt_i \, d\rho_i
\tag{4.49}
$$

where n_p is the number of parameters used. For example, in a non-spinning search in which the templates are laid out in τ_0 and τ_3 space, this would be:

$$
\mathbb{E}\left(N_{\text{uncorr}}\left(\rho^{\dagger 2} \leq \sum_i \rho_i^2, T_f\right)\right) = \int_{\mathcal{S}(\rho^\dagger)} \int_{\mathcal{E}(\rho)} \prod_i^{N_d} \sum_j^{N_s} \widetilde{r}_{i,j} P_{i,j}(\rho_i, \tau_0, \tau_3) \, d\tau_0 \, d\tau_3 \, dt_i \, d\rho_i
\tag{4.50}
$$

Ideally, the probability of getting a trigger would be independent of the template parameters. In that case, the integral over $\mathcal{E}(\rho)$ could be carried out independent

of the source PDFs, as was done with the time integral in section 4.2. This would mean that the number of coincident triggers with a combined New SNR $\geq$ to any given value would only be dependent on ρ. This is ideal, since the false alarm rate is meant to be a measure of how likely a trigger is a gravitational wave, independent of parameters. It would also mean that we could simply count the number of slide coincidences to get the false alarm rate, as described above, without needing any correction for the parameters of the templates.

In practice, however, the false alarm rate of a template *is* dependent on its parameters. Figure 12 shows the dependence of the cumulative rate of coincident slide triggers on chirp mass ($\mathcal{M}$) when the rates are based on New SNR. These triggers were taken from 100 slides in six weeks (about $0.03\,\mathrm{yr}$ of analysis time) of enhanced LIGO data in which the $4\,\mathrm{km}$ Hanford (H1) and Livingston (L1) detectors were analyzed ("S6"; see Chapter 7 for details). As can be seen in Figure 12(a), the overall rate of triggers decreases with increasing chirp mass. This is due to the fact that the template bank is more sparsely populated at higher masses. Conversely, the variance of the rate of triggers roughly increases with increasing mass, leading to a higher rate of high New SNR triggers at high mass. This occurs because the inspiral phase of a higher-mass template spends less time in band compared to lower masses. As a result, χ^2 does a poorer job distinguishing high mass CBCs from glitches, which also tend to ring off the higher masses with larger SNR.

The joint effect of the lower overall rate and higher variance of high-mass templates can be seen in Figure 12(b), which plots the combined New SNR of the triggers versus chirp mass. (To go from 12(a) to 12(b), think of the color bar as the z-axis, and imagine rotating the axes so that $x \to y$, $y \to z$, and $z \to x$.) Due to the decrease in the overall rate, the cumulative rate drops off with increasing chirp mass at low combined New SNR (orange – yellow lines in the figure). However, due to the higher variance, the rate increases with increasing chirp mass (up to the middle of the chirp-mass space, anyhow) at large combined New SNR (the red line in the figure). This latter effect can be particularly problematic since we are most interested in triggers with higher New SNR. For example, consider what were to happen if a binary neutron star template with $\mathcal{M} = 2\,\mathrm{M}_\odot$ rang off with a New SNR of 10.5. If our template bank was restricted to binary neutron stars ($\mathcal{M} \approx 1\text{--}3\,\mathrm{M}_\odot$), the trigger would stick well above background, as can be seen in Figure 12(b). However, if we include the entire

bank in the false alarm calculation, then the event would be out-ranked by the two events between $\mathcal{M} = 7\text{–}9\,\mathrm{M}_\odot$, leading to a $\mathcal{F} \approx 50\,\mathrm{yr}^{-1}$. This high FAR would have more to do with our ranking statistic's inability to take into account the parameters of the template than with our degree of belief in how likely the trigger is a gravitational wave.

One way to fix this is to break up the chirp-mass space when computing FARs. If we bin the space such that the cumulative rate is roughly constant for a given New SNR across the bin, then we can use our naive approach of counting slide triggers to get $\mathcal{F}$. The $\mathcal{F}$ computed in each bin, however, is not the true FAR for the search, since we sliced up the parameter space in order to arrive at it. To understand why the slicing needs to be accounted for, consider the number of bins. Our initial choice of bins was arbitrary: we could choose to slice the parameter space up even more; in so doing, we would get a smaller $\mathcal{F}$. Yet this plummeting FAR would have little to do with the likelihood that the triggers are gravitational waves and everything to do with our choice of bins. Thus, to fix this, we must compute a *combined FAR*, $\mathcal{F}_\mathrm{c}$, across all of the bins.

In order to compute a combined FAR, we use the *uncombined* FAR — i.e., the FARs computed using the bins — as our ranking statistic. This means we must compute an uncombined FAR for the slide triggers. We do this in the same manner that we compute uncombined FARs for zero-lag triggers: we count the number of slide triggers that have a combined New SNR $\geq$ the ρ of a given trigger. By doing so we treat a given slide as if it were zero-lag. Thus, we do not include the slide in which we are computing uncombined FARs in the background estimate for that slide. Since we use one-less slide for measurement of the background triggers' FARs, the error in our measurement of the slide triggers FARs will be larger than that of zero-lag triggers. Assuming each slide has roughly the same duration then, from equation 4.48, the error is increased to:

$$\delta\overline{\mathcal{F}}_\mathrm{slide} \approx \frac{\sqrt{\widehat{N}_\mathrm{total}(1 - \widetilde{N}_t^{-1})}}{\mathrm{T_b}(1 - \widetilde{N}_t^{-1})} \tag{4.51}$$

For large N_t, however, the discrepancy between the zero-lag and time-slide $\delta\mathcal{F}$ is small.

Once we have uncombined FARs for both foreground and background triggers,

we compute the combined FARs by counting the total number of triggers with a $\mathcal{F}$ *less-than* the given trigger's FAR, $\mathcal{F}^\dagger$:

$$\overline{\mathcal{F}}_{\mathrm{c}}(\mathcal{F}^\dagger) = \frac{\widehat{N}_{\mathrm{total}}\left(\mathcal{F} < \mathcal{F}^\dagger\right)}{\mathrm{T_b}} \tag{4.52}$$

The reason we count the number of triggers that have a FAR *less-than* and not *less-than or equal-to* $\mathcal{F}^\dagger$ can be understood by considering what happens to a trigger that has a ρ larger than all the background in its bin. If a trigger is louder than all the background, it will have a measured FAR of 0. This does not mean we are absolutely certain the trigger is a gravitational wave. A FAR of 0 means that we have set an upper-limit on the true FAR to be less-than $1/\mathrm{T_b}$, within the error bars; i.e.:

$$\overline{\mathcal{F}}(\rho^\dagger) = 0 \Rightarrow \widetilde{\mathcal{F}}(\rho^\dagger) < \frac{1}{\mathrm{T_b}} \pm \frac{1}{\mathrm{T_b}} \tag{4.53}$$

Since we have only placed an upper limit on the uncombined FAR, we should also only be able to place a limit on the combined FAR; thus, if we got zero for $\mathcal{F}^\dagger$ we should also get zero for $\mathcal{F}_{\mathrm{c}}^\dagger$. (The combined FAR limit will be higher however; specifically, it will be $N_{\mathrm{bins}}/\mathrm{T_b}$, where N_{bins} is the number of bins.) Yet if we have, say, N_0 triggers in the background with $\mathcal{F} = 0$ and we were to count the number of triggers $\leq \mathcal{F}^\dagger$, we would get a combined FAR *equal* to $N_0/\mathrm{T_b}$, not less-than. Thus, we would have extracted a measured value for the combined FAR from a limit, without having had any new information about the trigger.

Breaking up the parameter space into bins and computing combined FARs from uncombined allows us to map the triggers onto a space where the parameter dependence has been removed. Note, however, that since a slide is not counted against itself, in each bin we will have at least one trigger with $\overline{\mathcal{F}} = 0$. This is the trigger that has the largest value of ρ; i.e., it is the loudest background trigger. (We could have more than one slide trigger in a bin with $\overline{\mathcal{F}} = 0$ if a single slide contained multiple triggers that were louder than the triggers in all other slides.) Thus, if we have N_{bins} bins, we will have at least N_{bins} background triggers with $\overline{\mathcal{F}} = 0$. This means that if a foreground trigger has $\overline{\mathcal{F}} = 1/\mathrm{T_b}$ in its bin, it will have $\overline{\mathcal{F}}_{\mathrm{c}} = N_{\mathrm{bins}}/\mathrm{T_b}$. In other words, even if a foreground trigger has a ρ that is far larger than all but one slide trigger, and that slide trigger is in the same bin, its combined FAR will increase by at least a factor of N_{bins}. If a trigger has a zero uncombined FAR (and therefore

a zero combined FAR), the smallest upper limit that can be placed on its true false alarm rate will also increase according to the number of bins, or *trials factors*. The number of bins should therefore be kept to a minimum.

Figure 13 shows the background distribution in uncombined FAR of the same set of triggers used in Figure 12. The bins used are clearly visible via the "steps" in Figure 13(b). These bins — which are $\mathcal{M}/\mathrm{M}_\odot \in \{[0.0, 3.48); [3.48, 7.4); [7.4, 15.8)\}$ — are the same that were used in the analysis of LIGO's fifth and sixth science runs (S5 and S6, respectively). As can be seen by the lines of constant high cumulative rate in Figure 13(b) (orange – yellow lines), the bins are not sufficient to make the rate completely independent of the mass. Indeed, this is impossible with any choice of bins, since there are simply not enough triggers at high mass to extend the high cumulative rate lines into that region. However, for *low* cumulative rates, the space has been sufficiently "flattened." Note, in particular, the red line, which shows a cumulative rate of $1/\mathrm{T_b}$ in each bin. In Figure 12(b), this line dropped off at low chirp mass, which would cause low-mass systems to get higher-than-expected false alarm rates, as in the case of the BNS system above. Now, low-mass systems with high ρ are ranked on par with the medium mass systems. Since these are the triggers we are most interested in, this choice of bins is sufficient for our purposes, and we have only incurred a trials factor of 3.

There is one last parameter that we need to be aware of when choosing the number of bins: the number of detectors contributing to the coincidence. Since combined New SNR is computed by taking the quadrature sum of the single-IFO New SNRs, coincidences that have a larger number of contributing detectors will have a larger combined ρ. For example, if we analyze three detectors — call them H, L, and V — then we can have four types of coincidences: HL, HV, LV, and HLV. A trigger with New SNR = 8 in each detector will have a combined New SNR = $\sqrt{2} \times 8$ if it is a HL, HV, or LV coincidence, and a combined New SNR = $\sqrt{3} \times 8$ if it is a triple coincidence. That may seem appropriate: after all, the more detectors that observe the event, the more likely we are to believe it to be a GW. However, due to the antenna pattern of the detectors, and the fact that they are not co-located nor have the same sensitivity, a GW will not be seen in all detectors if it came from a source in certain sky locations and orientations. If we were to compute FARs without first binning the triggers by the number of coincident triggers, or *coincidence type*, we

would subtly introduce a location and orientation bias into the FAR. To avoid this, we also bin triggers by their coincidence type.[4]

Across the course of an analysis period varying combinations of detectors will be available to analyze. (An IFO is not analyzable if it is not in *Science* mode; see Chapter 5 for details.) For example, in the three detector case given above, we can have four different *instrument times*: periods when HLV were on, and periods when HL, HV, or LV were on. Obviously, only certain coincident types can occur in a given period. Therefore, when computing a FAR for a trigger, we should not include analysis times in which it was impossible for the coincidence type of the trigger to occur. If we did, we would artificially decrease the false alarm rate of the trigger. As a result, each of the various instrument times are treated as independent experiments: both uncombined and combined FARs are computed within the instrument time, and FARs are never calculated across instrument times. This also means that coincidences of the same type cannot be combined across instrument times. For example, in the three detector case, we can get HL, HV, and LV coincident triggers in both HLV time and each of the respective double-coincident times. However, there is a slight difference between the doubles in triple time and the doubles in double time. The double-coincident triggers in HLV time are triggers that were coincident with one other detector *and not* coincident with the third detector. The doubles in double time, however, consist of triggers that would not have been coincident with the third detector *and* triggers that would have been coincident, had the third detector been on. This latter set is clearly a larger population. Thus, if coincidence types were grouped across instrument times (using the sum of the instrument times for the analysis time),

---

[4]Admittedly, simply binning the triggers by coincidence type then using the uncombined FARs to get the combined FAR is not the perfect solution to remove biases either. Doing so treats all the coincidence types with equal weight. However, if the detectors have varying sensitivities, it is more likely to get a gravitational wave from coincidences of the most sensitive detectors. A better approach would be to add some weight to each coincidence type that is proportional to the sensitivity of the constituent detectors before combining the FARs. The same argument applies to mass-bins: if the detectors are more sensitive to certain parameter ranges, we should weight accordingly. A likelihood-based weighting that attempted to remedy these concerns was used in the joint LIGO-Virgo search in the last few months of S5; see [6] for details. For the rest of S5 and S6, the detectors' sensitivities were roughly on-par with each other, so the simpler equal-weight method described here was used.

we would underestimate the FARs of the triggers from double-time and overestimate the FARs from triple-time.[5]

Summarizing, the number of bins, or trials factors, that we use depends on how strongly the ranking statistic couples to a parameter and on the number of coincident types possible in the instrument time being analyzed. In the low-mass CBC search we use three mass bins based on chirp mass. This means that we have a trials factor equal to 3 during double-coincident instrument times. During triple-coincident instrument times in which all coincident types are used, this factor increases to 12 (3 mass-bins times 4 possible coincidence types).

4.5 Alternate Method to Compute Combined FARs

There is an alternate, but equivalent, method to computing combined FARs that does not require computing uncombined FARs for all of the background triggers. We describe it here as it was used in the S5 search.

We begin by re-writing equation 4.52 as an explicit sum over the number of triggers in each bin:

$$\overline{\mathcal{F}}_{\mathrm{c}}(\mathcal{F}^{\dagger}) = \frac{1}{\mathrm{T}_{\mathrm{b}}} \sum_{p=1}^{N_{\mathrm{bins}}} \int_0^{\mathcal{F}^{\dagger}\mathrm{T}_{\mathrm{b}}} \widehat{n}_p(\mathcal{F}') \, \mathrm{d}\mathcal{F}' \tag{4.54}$$

Here, $\widehat{n}_p(\mathcal{F}')$ is the measured number of triggers in p^{th} bin with an uncombined FAR equal to $\mathcal{F}'$. We expect to get roughly one trigger in each bin with a given $\mathcal{F}$. For example, we expect to get one trigger in each bin with a FAR of 0, one with a FAR of $1/\mathrm{T}_{\mathrm{b}}$, etc. There can be some excursions from this rule; for instance, as described above, it is possible to get more than one trigger with a 0 FAR in a single bin if one slide has more than one trigger that is louder than all triggers is all of the slides. However, if all of the slides are of roughly equal duration, then the distribution of triggers in each slide will be about the same. Thus, $\widehat{n}_p(\mathcal{F}') \approx 1$ for all bins and all FARs, and equation 4.54 becomes:

---

[5]We can, however, group coincidence-types together across instrument times if we reanalyze the data with the other detector removed, or *vetoed*. For example, we can group HL triggers in HL time with HL triggers in HLV time if we re-analyzed the HLV data with V removed. This has been done occasionally in order to get a better estimate of double-coincident triggers with low or zero FARs.
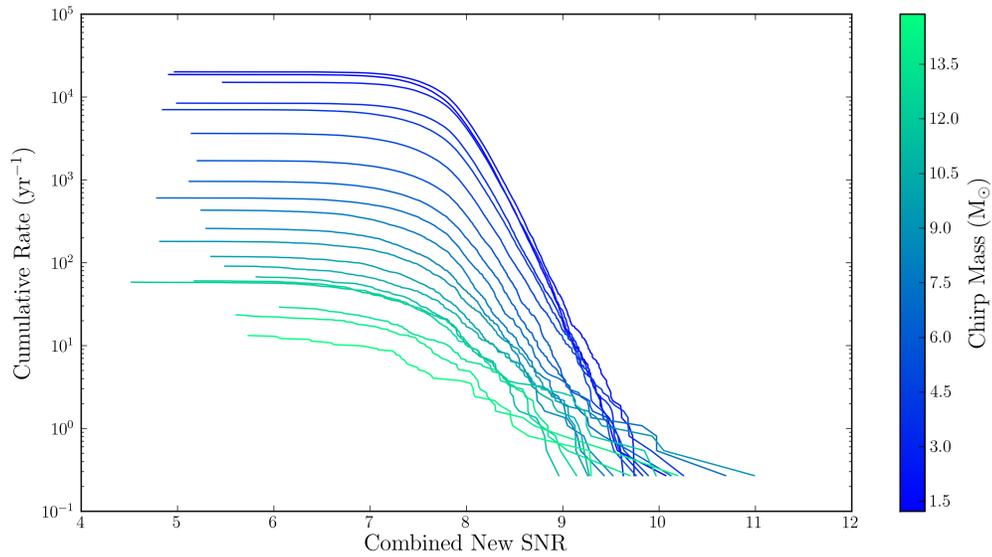
(a) Cumulative Rate v. Combined New SNR. Each line represents the cumulative rate in one chirp-mass bin as a function of Combined New SNR, the boundaries of which are shown in (b).
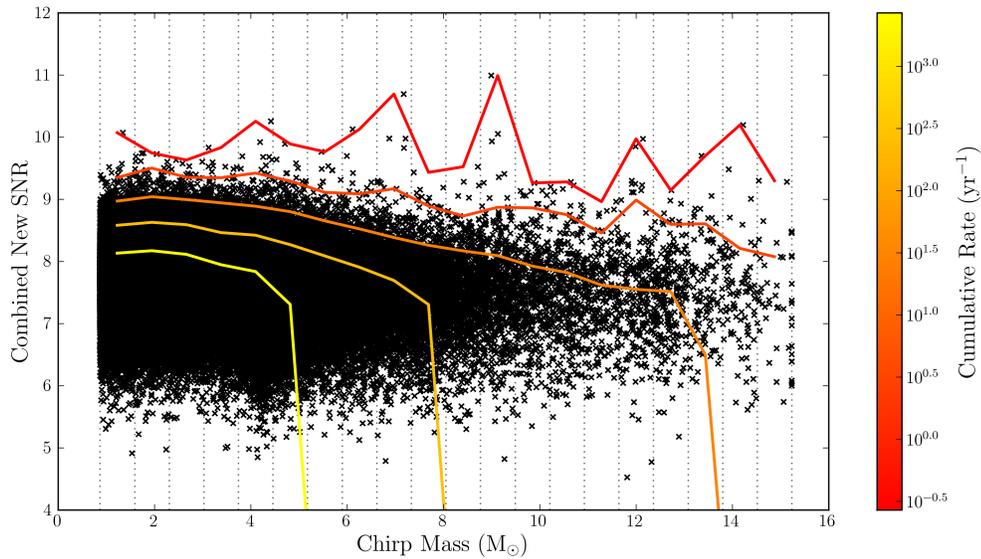
(b) New SNR v. Chirp Mass. Colored lines show lines of constant cumulative rates across the chirp-mass bins. Dashed lines show the bin boundaries used.

Figure 12 : The rates of slide coincidences in equally-spaced chirp-mass bins as a function of Combined New SNR. Black crosses are coincident triggers taken from 100 slides in six weeks of enhanced LIGO data. The cumulative rates (the y-axis in (a) and the color axis in (b)) are computed by counting the number of triggers with a Combined New SNR $\geq$ each trigger within each bin. Only the 4 km Hanford (H1) and Livingston (L1) triggers were analyzed during this time; as such, all coincidences are H1L1 triggers.
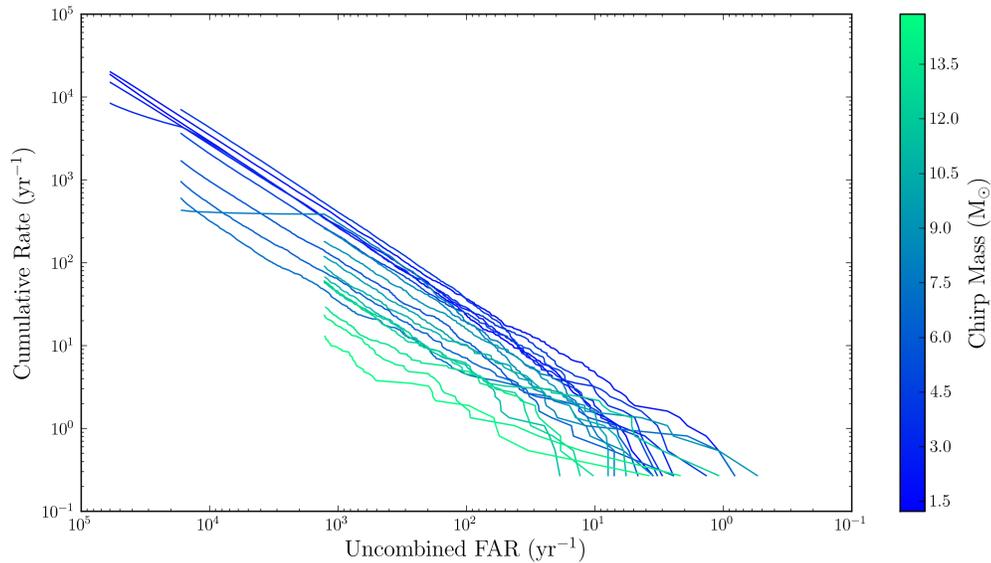
(a) Cumulative Rate v. Uncombined FAR. Each line represents the cumulative rate in one chirp-mass bin as a function of Uncombined FAR, the boundaries of which are shown in (b).
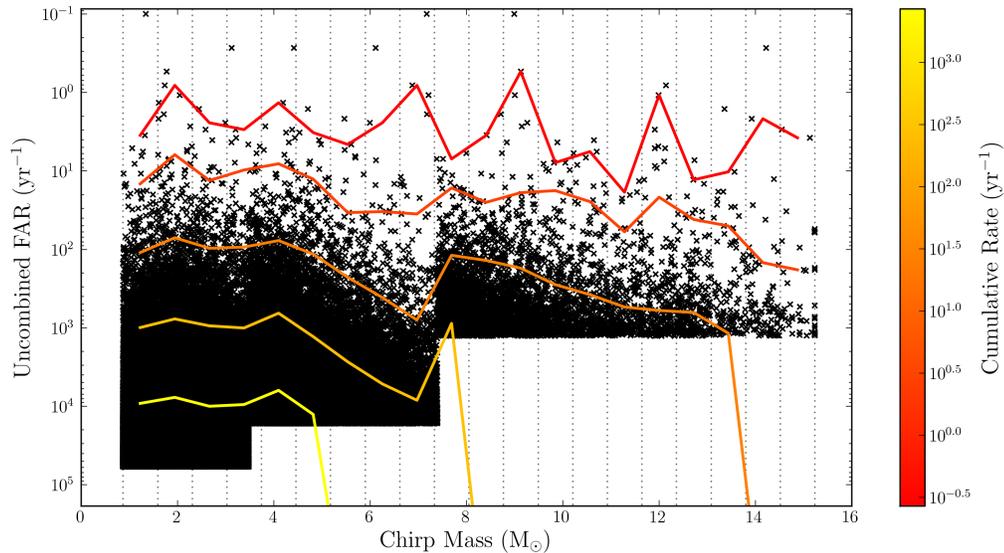
(b) Uncombined FAR v. Chirp Mass. Colored lines show lines of constant cumulative rates across the chirp-mass bins. Dashed lines show the bin boundaries used.

Figure 13 : The rates of slide coincidences in various chirp-mass bins as a function of Uncombined FAR. The data used is the same as that in Figure 12. The bin boundaries used to compute the Uncombined FAR are visible in the steps seen in (b). The Uncombined FAR were computed using `ligolw_cbc_cfar` using the same bins as used in the S5 and S6 searches (see Chapter 5 for details). The cumulative rates are computed by counting the number of triggers with an Uncombined FAR < each trigger within each bin. As a result the Uncombined FAR axes are inverted. Also note that the Uncombined FAR axes are plotted on a log scale.

$$\overline{\mathcal{F}_{c}}(\mathcal{F}^{\dagger}) \approx \frac{1}{T_b} \sum_{p=1}^{N_{bins}} \int_{0}^{\mathcal{F}^{\dagger}T_b} d\mathcal{F}'$$

$$= N_{bins}\mathcal{F}^{\dagger} \qquad (4.55)$$

Again, for this to be true, the number of triggers in each bin must be about the same. As can be seen in Figure 13(b) this is a valid assumption for low FARs (the top part of the graph). However, at high FARs (the low part of the graph), we run out of triggers in the higher mass bins. For example, consider what the combined FAR of a trigger with $\mathcal{F}^{\dagger} = 10^4 \, \mathrm{yr}^{-1}$ would be. We can imagine drawing a horizontal line on Figure 13 at $10^{-4} \, \mathrm{yr}^{-1}$ and adding up all the triggers that lie above that line. The contribution from the first two bins would be $2\mathcal{F}^{\dagger}$. However, in the third bin, the background has "run out" at this point, so that:

$$\widehat{n}_3(\mathcal{F}') = \begin{cases} 0 & \text{if } \mathcal{F}' > \max(\widehat{\mathcal{F}}_3) \\ 1 & \text{otherwise} \end{cases} \qquad (4.56)$$

$\max(\widehat{\mathcal{F}}_3)$ is the largest measurable FAR in the third bin; it is simply all the background triggers in the third bin divided by T_b. Thus equation 4.54 becomes:

$$\overline{\mathcal{F}_{c}}(\mathcal{F}^{\dagger} \approx 10^4 \, \mathrm{yr}^{-1}) \approx \frac{1}{T_b} \left(2 \int_{0}^{\mathcal{F}^{\dagger}T_b} d\mathcal{F}' + \int_{0}^{\max(\widehat{\mathcal{F}}_3)T_b} d\mathcal{F}' \right)$$

$$= 2\mathcal{F}^{\dagger} + \max(\widehat{N}_3)/T_b \qquad (4.57)$$

where $\max(\widehat{N}_3)$ is the number of background triggers in the third bin. Likewise, if we were to measure a combined FAR for an uncombined FAR at $\mathcal{F}^{\dagger} = 3 \times 10^4$, the second bin will have run out; giving:

$$\overline{\mathcal{F}_{c}}(\mathcal{F}^{\dagger} = 3 \times 10^4 \, \mathrm{yr}^{-1}) \approx \mathcal{F}^{\dagger} + \max(\widehat{N}_2)/T_b + \max(\widehat{N}_3)/T_b \qquad (4.58)$$

Generalizing, we have:

$$\overline{\mathcal{F}_{c}}(\mathcal{F}^{\dagger}) \approx N'_{bins}\mathcal{F}^{\dagger} + \frac{1}{T_b} \sum_{p=1}^{N''_{bins}} \max(\widehat{N}_p) \qquad (4.59)$$

where N'_{bins} is the number of active bins at $\mathcal{F}^{\dagger}$, N''_{bins} is the number of bins that have run out, and $\max(\widehat{N}_p)$ is the total number of triggers in the pth bin.

4.6 Algorithm for Computing False Alarm Rates

We have arrived at an algorithm to compute false alarm rates for coincident triggers produced in a CBC search. It is:

1. Perform several time slides by adding offsets to each detector and look for coincidences across the slid detectors. The offsets should be large enough to ensure that each slide is roughly independent (i.e., at least as large as twice the time-axis of the coincidence window; larger if a single event can cause multiple triggers across several seconds).

2. Bin the triggers by their coincidence type and time, and, if necessary, by template parameters.

3. Compute an uncombined FAR ($\overline{\mathcal{F}}$) for every zero-lag trigger in each bin by counting the number of slide, or background, triggers with a combined New SNR (ρ) greater-than-or-equal-to the combined New SNR of the given trigger.

4. Compute a final, combined FAR ($\overline{\mathcal{F}}_{\mathrm{c}}$) across all of the bins by either of the following methods:

   - Measure an uncombined FAR for every background trigger by counting the number of triggers in every other slide that have a combined New SNR that is greater-than-or-equal-to the combined New SNR of the given trigger. Next compute a combined FAR for the zero-lag triggers by counting the number of background triggers across all bins that have an uncombined FAR that is less-than the given trigger.

   - Use equation 4.59 to compute the combined FAR directly, without computing the uncombined FARs of the background triggers.

This method is valid as long as the number of GW/GW and GW/noise coincidences in the slides are small compared to the number of noise/noise coincidences, so that equation 4.43 is valid.

Chapter 5

The IHOPE Pipeline

In this chapter we describe in detail the `ihope` pipeline, which is the pipeline used in the all-sky search for gravitational waves from CBCs. This pipeline has been developed by a number of people over the past several years. The `inspiral` and `tmpltbank` programs (discussed in sections 5.3.4, 5.3.6, and 5.3.9) were developed during LIGO's first Science run, and refined for LIGO's second Science run [12, 63]. By S5, these programs had become a part of the Hierarchical Inspiral Pipeline Executable (`HIPE`) [21, 62], which is discussed in section 5.2.2 and 5.3. During S5 the pipeline was automated, and it became known as `ihope` [22]. Finally, during S6, Pipedown was developed and added to `ihope`. Pipedown is detailed in sections 5.2.3 and 5.5. The current version of `ihope`, discussed here, is currently being used to analyze data from S6 and VSR2 and VSR3 [61].

`Ihope` can by modeled by a directed acyclic graph (DAG). A DAG is a workflow in which the output of one program, or *node*, is the input of another node or nodes, such that the flow never loops back onto itself [63]. It is represented by a diagram in which the vertices are the programs and the edges show the interdependencies. `ihope` is a DAG of DAGs: its nodes are workflows that launch sub-workflows, which in-turn launch the programs that carry out the analysis. These DAGs are managed by the Condor High Throughput Computing system, which distributes the jobs across the computer cluster and manages the dependencies [66]. Figure 14 shows an overview of the `ihope` workflow. Data from each detector is retrieved, analyzed in parallel workflows, combined, and then output to a webpage. Each of these steps involves one or more DAGs.

In this chapter we discuss each of these steps in detail. Section 5.1 reviews the requirements of a CBC GW search pipeline; section 5.2 explains how the DAG is created at run-time; section 5.3 describes the in detail; 5.4 describes the tables used to store data; 5.5 describes Pipedown in detail and how the results are presented. Throughout this chapter we will make extensive use of the methods and concepts established in Chatpers 3 and 4. We refer the reader to those chpaters and the sources cited therein for more information.
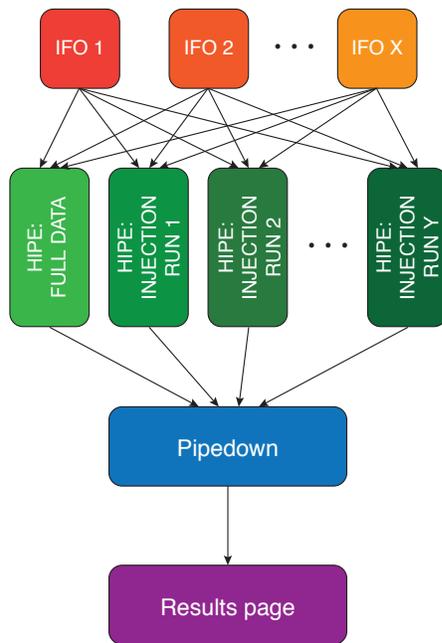
Figure 14 :   An overview of the `ihope` pipeline. The HIPE and Pipedown nodes are themselves workflows, and are detailed in sections 5.3 and 5.5, respectively.

5.1   Pipeline Requirements

Here we briefly review the key requirements of a pipeline used to search for GWs. Our goal is to search for GWs from coalescing binaries in a range of masses. Since gravitational waves couple very weakly to matter we must be able to detect strains of $\sim 10^{-21}$ in noise that has an RMS amplitude of $\mathcal{O}(10^{-22})$ at $100\,\mathrm{Hz}$. Our target sensitivity is to detect signals with a SNR of 8 in each detector.

When averaged over time the detectors' have a colored Gaussian noise distribution;

as shown in Chapter 3, the optimal tool to search for signals is therefore a matched filter [31]. Match filtering requires knowing the morphology of the waveform. Binaries with total masses (M_{total}) less than $25M_\odot$ emit most of their power in the LIGO and Virgo bands during their *inspiral* phases. This means that the waveform from these systems can be well modelled by the post-Newtonian approximation. Thus, to cover the desired range, we can fill a bank of templates using the methods described in section 3.2 of Chapter 3.

Environmental and instrumental factors can cause non-Gaussian transient noise (*glitches*) in the data. To deal with this our pipeline must be able to distinguish between triggers resulting from glitches and triggers resulting from gravitational waves. Since the morphologies of signals is known, and since the signals have a number of cycles in the frequency range (*band*) we are sensitive to, the χ^2 test discussed in section 3.3 is a powerful tool for discrimination [32]. Demanding that triggers be coincident in multiple detectors using the e-thinca ellipsoids discussed in section 3.4 will also filter out spurious triggers, since we do not expect environmental correlations across great distances [34]. After all filtering and tests have been applied, the statistical significance of a set of triggers has to be evaluated to determine the probability that a gravitational wave exists in the data. We can do this by calculating the triggers' *false alarm rates* (FARs) using the time-slide method discussed in Chapter 4. Since they only consist of noise triggers (assuming equation 4.43 holds), the coincident triggers resulting from time-slides are referred to as *background*, to distinguish them from coincident triggers in which no offset has been applied between detectors, or *foreground*.

Even though we apply χ^2 and coincidence tests, environmental factors can cause periods of elevated glitch rate in the detectors. If these periods are analyzed with periods of relatively clean data, they will pollute the background estimation, thereby decreasing statistical confidence in candidates. We therefore seek to remove such periods from the analysis. This is accomplished using *vetoes* [67, 68]. A veto is a half-open time-interval, or *segment* in which all the triggers are discarded. Vetoed time is not counted in the analyzed, or *live*, time when false alarm rates are calculated.

We define veto times by using *data quality (DQ) flags*. The detectors are equipped with environmental and instrumental monitors that are auxiliary to the GW channel. If these channels detect periods of heightened activity in the environment, such as

elevated seismic noise, an automated monitor will mark the period of time with a DQ flag [69]. Additional DQ flags can be added by hand; e.g., if a truck drives onto the site while GW data is being recorded, a person in the control room may add a flag for that period of time. The periods of time that these various flags are on are stored in a *segment database*, which is a central database that contains lists of segments detailing the times that the detectors were in various states [70]. If a DQ flag can be found to be strongly correlated with elevated glitch rates in the GW channel, and if it is found that the flag is *safe* — i.e., it will not be triggered by a GW wave[1] — then it is used as a veto.

Vetoes are categorized according to how well we can couple them to known environmental sources [67, 68]. Table 5.1 lists the various categories and their defining characteristics. These categories are applied cumulatively. We refer to category 1 vetoes as *CAT1*. Category 1 and 2 vetoes are referred to as *CAT2*; category 3 vetoes are referred to as *CAT3*. For CBC searches, we do not analyze anything prior to category 1; i.e., all matched filtering is carried out after category 1 vetoes are applied. CAT2 and CAT3 vetoes are applied when second stage coincidence is carried out (see section 5.3.10, below). We quote false alarm rates and base upper limits on data in which category 1-3 vetoes have been applied. We additionally check the data after category 1 and 2 vetoes have been applied for any loud triggers that may have been removed by category 3 vetoes. We do not use category 4 for the analysis. However, we do use category 4 vetoes in follow-up studies of loud candidates to provide insight into the cause of triggers. Hardware injections are left in the data after category 1 and 2, and are removed as a special veto prior to category 3 vetoes being applied.

The use of vetoes is key to our ability to detect gravitational waves. Without them, we could not detect signals at a SNR of 8 in each detector (see Chapter 8). To determine what DQ flags are most effective at removing triggers caused by environmental factors we perform *glitch studies*. This allows us to *tune* vetoes such that we maximize the probability of detecting a signal.

In addition to vetoes, each of the analysis methods described above have various parameters that can be tuned. This includes the number of χ^2 bins to use [31], the χ^2 threshold to apply [62], the duration of the r^2 veto [33], how finely to grid the parameter space and what coordinates to use [55, 56, 57, 58], what pN order to use

---

[1]This can be checked using hardware and software injections.

to generate templates [21, 22, 61], the size of the e-thinca parameter [34, 62], the number and placement of the chirp-mass bins for calculating FARs [62], and many other parameters not discussed here. Like vetoes, we wish to tune these parameters such that we maximize the probability of detecting a signal. However, we must do this in a *blind* fashion. That is, we must be able to tune parameters without knowledge of what is in the data, else the results will sway in favor of the analysts' bias.

To satisfy these conflicting requirements, we have designated a subset of data as *playground*. Playground is defined as data that occurs during the first 600 seconds of every 6370 seconds, starting from February 14, 2003 at 16:00:00 UTC (GPS time 729273613). It therefore consists of $\sim 10\%$ of the full data. The results of tuning studies are evaluated by considering the results of analyzing playground data. In addition to playground data, we permit ourselves to look at the most significant (*loudest*) coincidence triggers from slide data when ranked by FAR. If an auxiliary environmental or instrumental channel can be correlated with a loudest slide trigger, we can veto it. (For more on these *loudest slide studies* see chapter 7.) Only after all of the vetoes and parameters have been tuned do we look at, or *un-blind*, the full data results to see if a GW candidate exists.[2]

Finally, the pipeline must be able to evaluate its sensitivity and efficiency to sources in the universe. Doing so allows tuning studies to be carried out prior to doing the full analysis, and for the astrophysical rate of CBCs to be bounded after the analysis has completed. This can be done by performing *injections* of signals with known parameters into the detectors. Both *hardware* and *software* injections may be performed. Hardware injections involve actuating the mirrors to physically simulate a passing gravitational wave. This is the most robust test as it checks the ability of the detectors' response loop to measure GW strains and the ability of the pipeline to detect them. However, hardware injections prevent real GWs from being detected while they are occurring, limiting the number that can be preformed. Software injections involve adding a gravitational wave signal to the data stream on disk just prior to analyzing it. While this method does not test the hardware control systems, it has the advantage that it can be performed many times in parallel, without corrupting the original data. Thus, our pipeline must be able to perform software

---

[2]Full data results include data from playground time. Although playground is included in the final unblinded results, we exclude it for computing upper-limits [21, 22, 61].

Category	Description	Procedure
1	Data seriously compromised or missing.	Data never analyzed.
2	Instrumental problems with known coupling to $h(t)$.	Vetoed triggers discarded after second coincidence. Surviving triggers checked for candidates, but not used for upper limits.
3	Instrumental problems likely, casting doubt on triggers found during these times.	Vetoed triggers discarded after second coincidence. False alarm rates of surviving triggers are used in publications; upper limits are calculated using these vetoes.
4	Positive, but weak, correlations with false alarms. Large dead times.	Not used in the analysis, but used as a guide in detailed follow-ups of loud triggers.

Table 3 : The various veto categories used by the CBC group. Vetoes are applied cumulatively; statistical significance of candidates and upper limits are calculated after category 1, 2, and 3 vetoes are applied.

injections, and have a method for associating triggers with the injections that went into the data.

In summary, a pipeline used to search for gravitational waves from CBCs must:

- construct a bank of templates with which to filter [55, 56, 57, 58];

- identify *triggers* by filtering templates through the detector data [31];

- distinguish noise triggers from gravitational wave triggers [32];

- veto periods with elevated glitch rate due to environmental factors [67, 68]

- quantify statistical significance of triggers and rank them in a blind manner;

- evaluate the sensitivity and efficiency of the search to CBCs in the universe [21, 22, 61].

In the following sections we will see how `ihope` meets these requirements.

5.2   ihope at Runtime

The ihope pipeline is created by running lalapps_ihope. This sets up the workflow by doing the following at run time:

- set-up the directory structure to save all data to;

- copy all needed programs from their installed location to a local directory;

- retrieve analysis start and stop times;

- download a *veto-definer file* and find the start and stop times of all veto segments;

- run lalapps_inspiral_hipe multiple times;

- run lalapps_cbc_pipedown;

- create a cache file of the names and locations of all files that will be created;

- write a DAG that can be used to start and run the workflow.

These steps require the start and stop time (in GPS seconds) of the period to be analyzed — which we shall refer to as the *analysis period*[3] — as well as a *configuration file.* The configuration file is a text file containing all the information needed to setup and run the analysis. This includes: the names and locations of all the executables that will be run; variable arguments that these programs will need; the name and number of interferometers to analyze; the version of data files to retrieve and what channels to analyze; how many and what type of software injection runs to do; any other information needed by the DAGs to run. The configuration file provides a convenient way to manipulate the pipeline. Changing tuning parameters is largely accomplished by editing this file. Likewise, the difference between running a *low-mass* search ($2 < M_{\text{total}}/M_\odot < 25$) and a *high-mass* search ($25 < M_{\text{total}}/M_\odot < 100$) is determined entirely by the configuration file.

A directory named by the GPS start/stop times of the analysis period is created at runtime. All work is done in this directory. In it, a segments, executables,

---

[3]We typically analyze periods that are on order of a week to a month. For analysis periods used in S5 and S6 see chapters 6 and 7, respectively.

`datafind`, `full_data`, and `pipedown` directory are created, along with a directory for each injection run that will be carried out. With the exception of the `executables` and `segments` directory, each of the sub directories store a sub-DAG that will be run during the analysis (and will be explained below). The master DAG is saved in the GPS-times directory along with a master cache file of all the files that will be created. All programs that will be run are copied to the `executables` directory.

5.2.1 Science and Veto Segments Retrieval

During a Science run the LIGO and Virgo detectors can be in one of several different states at any given time, which we categorize into five different operating modes. We are only interested in analyzing times in which the detectors are in *Science* mode. This means they are "locked,"[4] no other experimental work is being done on them, and their state has been verified by a human. The interferometers can come out of Science mode many times across an analysis period; thus `ihope` must retrieve the start and stop times of Science segments that occurred in the analysis period. `Ihope` does this by running `ligolw_segment_query` [70] at run time. This program queries the segment database (described above) to retrieve the list of Science times during the desired analysis period. These results are saved to files in the `segments` directory. These files do not contain strain data; they only list the times that data can be retrieved. The results are later used to retrieve files containing strain data.

The Science segments, as well as most of the results of each step in the pipeline, are saved in *XML* files. An XML file stores data in a hierarchical structure of "elements," where each element may contain other elements [71]. In our case, the elements of the document are data tables; in turn, the data tables have columns, which are the elements of the table. The XML files used by `ihope` conform to the `LIGO_LW` data format, which specifies how tables and columns are named, and how they are stored to XML files. For more details on XML and the `LIGO_LW` format see [71].

As discussed in section 5.1, all of the DQ flags that may be used for vetoes are stored in the segment database [70]. What flags to use for vetoes, at what category, and for how long, are stored in a *veto-definer file*. This XML file contains a `veto_definer` table that lists each flag that should be used, what category the flag

---

[4]Recall from Chapter 2 that the interferometer light must be resonant in the Fabry-Perot cavities for the interferometer to operate. We refer to this as the interferometer being "in lock."

should be used at, the dates the flag is valid, and any padding (in seconds) to add to the flag. The table is veto-definer file is created by analysts who carry out studies to determine what DQ flags to use for a specific search. These studies draw on a number of different resources, including: data-monitoring tools that run at the observatories, an online log, or *e-log* that scientists use document events and studies carried out at the observatories,[5] as well as results `ihope` produces from playground data and injection studies. Entries are added to this table by hand after extensive data-quality investigations and safety checks.[6] Vetoes are fine tuned for specific searches and Science runs; each searches' set is saved in a different veto-definer file in a central repository. What veto definer file to use is specified in the `ihope` configuration file. At run-time, `ihope` downloads the desired file to the `segments` directory. It then runs `ligolw_segments_from_cats` [70] to query the segment database for flags specified in the veto-definer file. The vetoed segments for all of the instruments are added together and saved in XML files in the `segments` directory.

5.2.2 HIPE

Once the analyzable Science segments and the veto segments that will be applied are obtained, `ihope` runs `lalapps_inspiral_hipe`. This sets up the HIPE DAGs, which are the pipelines that carry out the search. Figure 15 shows a HIPE DAG for a single 2048s block of time. As can be seen in the diagram, HIPE is a *two-stage* pipeline. Data is matched-filtered [31], coincidence tests [34] are applied, then the process is repeated. The reason for using two-stages is the χ^2 test is computationally expensive [12]. To reduce the number of triggers for which we need to compute χ^2, an initial coincidence test is applied to single-detector triggers for which only SNR is calculated.[7] Recall from Chapter 3 that a coincidence test reduces the number

---

[5]See section 5.6.8 for more details on the e-log.

[6]For more details on how these studies were carried out for the low-mass CBC search in S6, see chapter 7.

[7]Using two stages complicates the pipeline, however, and makes it difficult to trace an event's progression through the various steps. It also hampers efforts to estimate FARs from single-detector triggers. For this reason, a single-stage pipeline is being developed which takes advantage of advances in computational power. See Chapter 9 for more details.

of triggers in the detectors since noise sources are uncorrelated between the sites.[8] After applying this *first coincidence* test, we calculate χ^2 for the surviving triggers and apply the χ^2 threshold and r^2 veto. We then perform a *second coincidence* test to check for coincidence among triggers that survived the χ^2 cuts [12].

HIPE can be run either with or without software injections. If injections are desired, lalapps_inspinj is run to create a list of injections to perform,[9] which are created and inserted into the data just prior to match filtering by lalapps_inspiral [63] when the DAG is run. At run-time ihope runs HIPE several times: once for each desired injection DAG[10] and once each for *zero-lag* and *time-slid* data. Zero-lag data is data for which the coincidence test is applied *without* adding a time offset to each detector. This is the data that could potentially contain coincident triggers from GWs. "Time-slid" data is data for which the coincidence test is applied after adding a time offset to each detector. Since the data is slid such that the offset is larger than the light-travel time between the detectors, time-slid contains "accidental" coincidences, i.e., we do not expect single-detector triggers from a single GW to be coincident with each other in time-slid data. This is the data we use for our *background* when calculating false alarm rates. (For details, see Chapter 4.) Each of these HIPE DAGs are distinguished from each other by a *user-tag*. Zero-lag data is labelled FULL_DATA, while time-slid data is labelled FULL_DATA_SLIDE. The user tag to use for each injection DAG is set in the configuration file by the analyst.

A description of what each program does when the HIPE DAGs are launched are discussed in section 5.3.

5.2.3 Pipedown

After all the instances of lalapps_inspiral_hipe have run, ihope runs lalapps_cbc_pipedown which sets up the Pipedown DAG. Pipedown takes the results of all the different HIPE DAGs, combines them into SQLite databases, computes and ranks triggers by FAR, and creates plots and tables of the results. Figure 16 details

---

[8]There are correlated noise sources between the two Hanford interferometers, H1 and H2, so this test is less effective for those two interferometers.

[9]See section 5.3.5 for details on how this is done.

[10]The number of injection runs to carry out and the types of injections to perform in those runs are specified in the configuration file. See section 5.3.5 for more details.
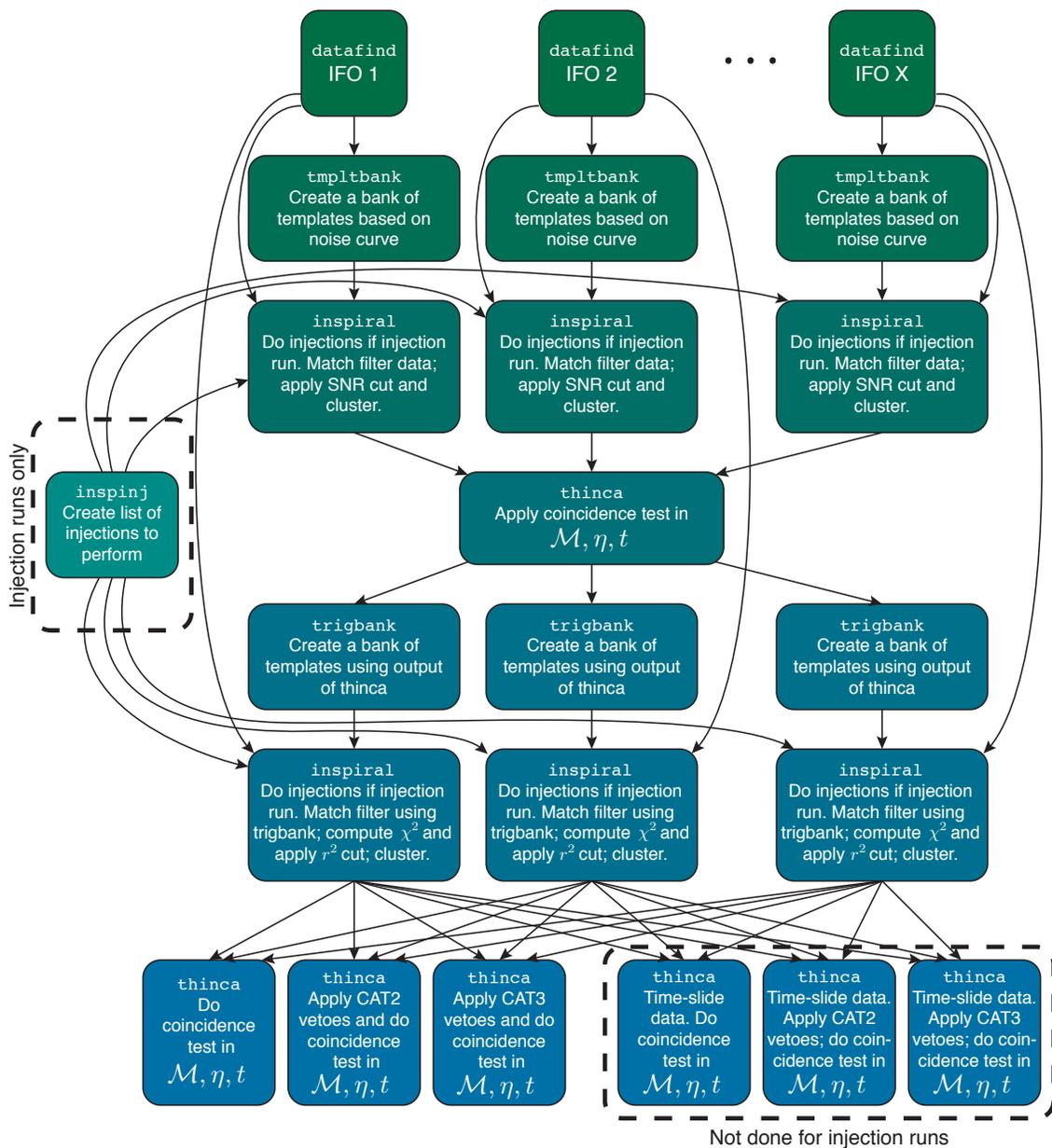
Figure 15 : The HIPEpipeline. This is run once for full-data and once for each injection run. For injection runs, lalapps_inspinj is run to generate a list of injections to create, and time slides are not done.

the steps Pipedown takes to carry out these goals. Shown are the steps taken for a single veto-category; this diagram is repeated for each veto-category (by Pipedown, not by `ihope`). In-depth details of Pipedown are discussed in Section 5.5.

5.2.4 DAX

After Pipedown has completed, `ihope` writes a *DAX* that can be used to launch the pipeline. A DAX is an abstract workflow in which elements such as file locations are variables. The DAX is turned into a DAG by the Pegasus Workflow Management Service [72]. The `HIPE` DAGs are then distributed across a compute cluster; after they finish, the Pipedown DAG is launched.

5.2.5 The Pipeline in Detail

Now that we have established an overview of `ihope`, we will step through it in detail, using a toy analysis period of 10 240s as an example. In this analysis we will use three interferometers: the 4-kilometer Hanford detector (H1), the 4-kilometer Livingston detector (L1), and the 3-kilometer Virgo detector (V1), and we will add one injection run, which we label `BNSINJ`. In section 5.3 we step through each of the programs that `HIPE` launches to carry out the analysis. Next, in section 5.5 we detail the programs launched by Pipedown in order to combine results from the `HIPE` DAGs.

The programs run in the `HIPE` DAG have been presented in prior publications; cf. [63, 12, 34, 62]. Here, we provide a review of how each of these programs are run in `ihope`. Pipedown is a new addition to the `ihope` pipeline, and is presented here for the first time. The routines the Pipedown programs carry out to compute false alarm rates are based on earlier programs described in [62], however.

5.3 HIPE in Detail

Figure 18 shows the segments that are available to analyze during our toy analysis period. The *selected segments* are Science segments minus CAT1 veto segments; they are what `ihope` passes to `HIPE` to analyze. Figure 19 shows all of the XML files that the `FULL_DATA HIPE` DAG creates in our toy analysis. We will use this figure and Figure 15 as guides as we step through each stage of `HIPE`.
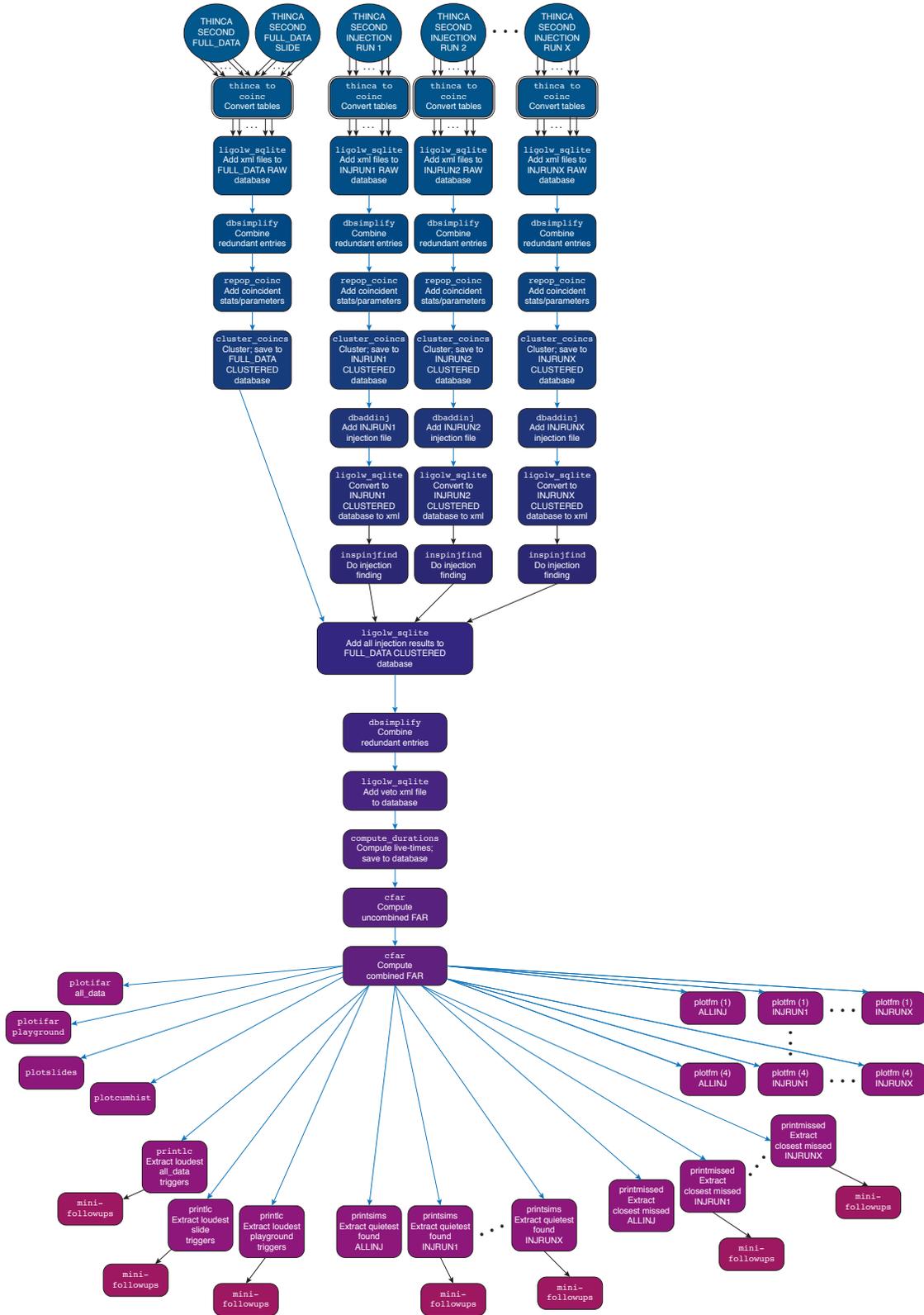
Figure 16 : The Pipedown pipeline for a single veto category. Each block represents a single node. Double bordered blocks represent multiple nodes. Circles represent batches of files. Black arrows represent XML files; blue arrows, SQLite databases. Each arrow represents a single file.

5.3.1   Data Find

As can be seen in Figure 15, the first step in `HIPE`is to run `ligo_data_find`. Data from all of the interferometers are stored in *frame files* in central locations at every computer cluster. Frame files can contain multiple channels recorded from the interferometers. For analysis purposes, we are only interested in files containing the strain data channel, which is called `LDAS-STRAIN` in the LIGO detectors and `h_16384Hz` in Virgo. `ligo_data_find` locates where the frame files covering the selected segments are in the file system on the cluster. It then creates a cache files listing the location of each these files in the `datafind` directory. These cache files are passed to `lalapps_tmpltbank` and `lalapps_inspiral`, which use them to locate and open the frame files for analysis.

5.3.2   From Continuous to Discrete Data

All of the equations we used Chapter 3 involving Fourier Transforms and the inner product have assumed continuous time- and frequency-domain data series. Likewise, the integrals over frequency space were from $-\infty$ to $+\infty$. In practice, of course, the data is discretely sampled and is neither continuous nor infinite in extent. Both the LIGO and Virgo strain data are sampled at $16384\,\mathrm{Hz}$. Since even the lowest mass templates (i.e., waveforms with the highest frequency components) used in current CBC searches terminate at frequencies $\sim 1.5\,\mathrm{kHz}$, this sampling rate is much higher than is needed for our purposes. To ease computational requirements the time series is therefore downsampled to $4096\,\mathrm{Hz}$ prior to analysis [63]. This sampling rate sets the Nyquist frequency, f_{Nyquist}, at $2048\,\mathrm{Hz}$. To prevent aliasing, a low-pass time-domain digital filter with a cutoff at f_{Nyquist} is implemented to pre-condition the data [63]. On the low-frequency end, seismic noise dominates the interferometers' power spectrum. We therefore also impose a high-pass digital filter in the time domain. The cutoff frequency of the high-pass filter, f_c, is set to be a several Hz lower than a low-frequency cutoff, f_0, that is determined by the characteristics of each detector's power spectrum. These values are set in the configuration file.[11] Both the low- and

---

[11]In both S5 and S6, f_0 was set to $40\,\mathrm{Hz}$ for the LIGO interferometers due to the rising seismic noise at that frequency (see Figure 7). For Virgo's first science run, f_0 was set to $60\,\mathrm{Hz}$ [6]; for VSR2 and 3 this was reduced to $50\,\mathrm{Hz}$ (see Chapter 7). For all interferometers, f_c was set to $30\,\mathrm{Hz}$ in S6.

high-pass filters will ring at the start and end of a time series, corrupting the data. Thus we must remove the first and last $t_{\rm pad}$ of data after applying the filters and prior to analyzing. The duration of $t_{\rm pad}$ is also set in the configuration file; for both S5 and S6 we used 8 s.

To Fourier Transform the data, we use the Fast-Fourier Transform (FFT) algorithm [12]. The FFT imposes two constraints on the data. First, the number of points in the data series must be a power of two. Second, the FFT associates the last point of the data series with the first point; i.e., it wraps the data around on a loop. This means that as a template is filtered toward the end of the time series, any points extending beyond the end of the series will be placed at the beginning, corrupting the data. Thus the first $t_{\rm chirp}$ points of the SNR time series are corrupted, where $t_{\rm chirp}$ is the *chirp length* of the template, and must be thrown out [31].

The chirp length is defined as the length of time it takes for the binary to go from f_0 to the frequency at which the binary passes ISCO, $f_{\rm isco}$. As discussed in Chapter 2, the pN approximation breaks down at $f_{\rm isco}$; therefore we must terminate all integrals at this point. Combining this with the limits of Nyquist and seismic noise, all frequency domain integrals are limited to the region $f \in [f_0, f_{\rm max})$ where $f_{\rm max} = \min(f_{\rm Nyquist}, \ f_{\rm isco})$ [31].

The PSD is estimated using a variation of Welch's method [31]. This involves breaking a data segment up into several bins of equal duration. Within each bin, the data is transformed to the frequency domain via the FFT. Thus the number of points in each bin must be a power of two, and the bins must be overlapping to account for corruption at the beginning and end of each segment. Since the data is discrete, this results in a discrete number of frequency bins. The *median* value within each frequency bin is then chosen across all the data segments to construct $S_n(|f|)$. We use the median — as opposed to the mean — to buffer the PSD from over-estimation in the presence of a signal or glitches [31].

The computation of $S_n(|f|)$ is also affected by the discontinuity between the first and last point of the data time-series. Additionally, recall from Chapter 3 that to match filter the data, we filter $\widetilde{s}(f)$ with:

$$\frac{\widetilde{h}^*(f)}{S_n(|f|)}$$

This is known as the *kernel* of the filter. Since $S_n(|f|)$ is a part of the kernel, the

impulse response of the kernel is such that the entire data series would be corrupted if a delta function existed in the data. To limit the corruption due to the wrap-around and the "ringing" of the kernel, we *truncate* the PSD [31]. This is done by estimating PSD using Welch's method, inverse transforming $\sqrt{S_n^{-1}(|f|)}$ to the time domain, zeroing out the first and last $t_{\text{invspectrunc}}$ seconds of the time series, then transforming back to the frequency domain. This limits the corruption due to the wraparound to the first and last $t_{\text{invspectrunc}}$ seconds of the time series. The truncation does cause some smoothing out of high Q features, such as power-line harmonics; however, since we search for relatively broad band signals this smoothing has little effect on the search. The value of $t_{\text{invspectrunc}}$ was set to 8 s for both S5 and S6.

For more details on how the PSD is estimated and the matched filter is implemented in CBCs searches, see [31] and [12].

5.3.3   Data Segmentation

`lalapps_inspiral` is the program that constructs the SNR time series for templates laid out in a bank by the program `lalapps_tmpltbank` [63]. The SNR time series is constructed according to equation 3.34 and the templates are laid out using the metric defined in equation 3.41. Since both of these equations involve the inner product defined in equation 3.11, both programs must compute the PSD, $S_n(|f|)$, as well as the Fourier Transform of the data, $\widetilde{s}(f)$.[12]

The wrap around of the FFT and the Welch median PSD estimation method put limitations on how much data can be analyzed at once; they also require data to be overlapped due to corruption at the start and end of segments. These considerations require us to carefully segment the data for analysis. The things we must consider are [31]:

- The low-pass and high-pass digital filters corrupt the first 8 s of an analysis segment, or *block*.

- In order to estimate the PSD using the Welch median method, we must break the block up into several segments; data in each segment is transformed via the

---

[12]We do not Fourier Transform the templates as this would be computationally expensive [31]. Instead, we use the *stationary-phase approximation*, which is an analytical method that allows us to express the templates in the frequency domain directly [73, 74]. For a derivation, see [31].

FFT independently.

- The FFT requires the number of points in each segment to be a power of two.

- The wrap around of the FFT causes the first t_{chirp} seconds of each segment to be corrupted due to the template, and the first and last $8\,\text{s}$ of the segment to be corrupted due to the inverse PSD.

- Each segment must therefore overlap so as not to lose data from the corrupted parts of the FFT.

In initial and enhanced LIGO the low-frequency cutoff f_0 was set to $40\,\text{Hz}$ (see Figure 7. In the low-mass CBC search, the longest template is therefore $\sim 33\,\text{s}$ in duration. The smallest power of two larger than 33 is 64; thus the first $64\,\text{s}$ of each segment must be overlapped by a previous segment. The end of each segment only needs to be overlapped by $8\,\text{s}$ to account for the PSD corruption. However, for book-keeping simplicity, the last $64\,\text{s}$ of each segment is also discarded. Thus, all the segments must overlap each other by $128\,\text{s}$. Since the segments must be a power of two, this means each segment must be $256\,\text{s}$ in duration [31].

In deciding upon the number of segments to use in the median PSD estimate we must consider a few factors. The more segments we have, the more accurate the PSD estimation will be. However, more segments also means that more time will be grouped together. If the PSD changes over this period, our estimate will be off. Furthermore, the number of segments used sets a minimum period of time that we must have continuous data. If a selected segment is shorter than that period, we cannot analyze it.

In initial and enhanced LIGO we have chosen to use 15 overlapping segments for each PSD estimation. Since each segment is $256\,\text{s}$ long, with overlaps of $128\,\text{s}$, this means that each block must be $2048\,\text{s}$ long. Add to this the needed $8\,\text{s}$ pad at the start and end of the block to account for the corruption of the filters, and we find that we need a continuous $2064\,\text{s}$ of data in order to do the analysis. If a selected segment is longer than $2064\,\text{s}$, then it will be broken up into several blocks. Each block is overlapped by $72\,\text{s}$ to account for the time thrown out at the beginning/end of the first/last segment in each block and the $8\,\text{s}$ pad. If the selected segment is not a multiple of 2048 (as it most likely will not be), then the last block in the segment is

overlapped more so as to analyze the entire period. The segments in this last block that overlap with non-corrupted time in the previous block are simply not match filtered, although they are used for the PSD estimation of the last block.

The effects of this data segmentation on our toy analysis can be seen in Figure 19. The top three lines show the selected segments. Each `TMPLTBANK` and `INSPIRAL` jobs corresponds to a single chunk in a single detector. As can be seen, every chunk overlaps by $72\,s$, except for the last one in each segment. Also note the first selected segment in V1. As seen in Figure 18, a CAT 1 veto broke the V1 Science segment into two selected segments. Since the first selected segment is only $\sim 900\,s$ long, it cannot be analyzed. This can be seen in the `TMPLTBANK` and `INSPIRAL` lines: there are no V1 jobs covering this period. This is part of the reason why CAT 1 vetoes are only used for seriously compromised data. Overuse of CAT 1 vetoes could lead to large amounts of unanalyzed time if they broke the data up into selected segments that were shorter than $2064\,s$.

5.3.4 Template Bank

The first step in the `HIPE` pipeline after `data_find` is to create a *template bank*.[13] As stated above, `lalapps_tmpltbank` is the program that constructs the bank. `HIPE` creates one `tmpltbank` job for each detector and for each chunk. Since the PSD is re-estimated for each chunk the metric on the parameter space (given in equation 3.41) will change from job-to-job, resulting in a different template bank for each analysis block and for each detector. However, the changes in the PSD are usually small, and so the bank stays roughly the same across an analysis period. In order to calculate the PSD `lalapps_tmpltbank` reads in the frame cache, loads the data, applies the low- and high-pass filters, down samples to $4096\,Hz$, then computes the PSD using the Welch median method [12].

Variables such as what pN order to use to lay out templates, what space to lay them out in, and what minimal match to use are command-line arguments to `lalapps_tmpltbank` and can therefore be set in the configuration file. As stated in Chapter 3, for S5 the bank was laid out in τ_0 and τ_3 space using 2.0 restricted pN

---

[13]Recall from section 3.2 that a template bank is a collection — or *bank* — of waveforms – or *templates* — that cover a parameter space. How they are laid out is determined by computing a metric on the parameter space [54, 55, 56, 57, 58].

templates to calculate the metric, such that the maximum loss in SNR due to the discreteness of the bank is $\sim 3\%$ [21, 22, 61]. By S6, it was possible to generate restricted 3.5pN templates. However, the bank metric and best coordinates to use to lay out templates is unknown at 3.5pN order. While the pN order of the templates should be the same as the pN order used to calculate the bank metric, it was found that using $3.5\,\mathrm{pN}$ waveform with a $2\mathrm{pN}$ metric did not significantly hurt the efficiency of the pipeline to recover software injections. As a result, restricted $3.5\,\mathrm{pN}$ templates were used in S6 and they were laid out using the $2\,\mathrm{pN}$ metric.

Figure 17 shows a typical template bank in both τ_0, τ_3 space and in m_1, m_2 space for the low-mass CBC search.[14] The parameters of each of these templates are stored in a `sngl_inspiral` table (the "sngl" stands for "single"; see section 5.4 for more details), which is saved in an XML file. Also stored in the `sngl_inspiral` table are the metric components around each template. Each `tmpltbank` job outputs one XML file, with naming convention:

$$\texttt{\{IFO\}-TMPLTBANK-\{GPS-START\}-\{DURATION\}.xml}$$

All of the `tmpltbank` files created in our toy $10\,240\,\mathrm{s}$ analysis can be seen in Figure 19.

5.3.5 Injections

Software injections are the CBC group's way to check that our pipeline is working.[15] Since the number we can perform is only limited by computational power, we can do a large number across a broad range of parameters and sky positions. Being able to perform such Monte Carlo simulations allows us to test the efficiency of our pipeline, tune parameters, and calculate rate limits to compare to astrophysical expected rates.

Injections are only performed in injection `HIPE` DAGs; these are later combined with non-injection results by Pipedown. For example, in our toy analysis, we have chosen to do one injection DAG, which we labeled `BNSINJ`. If we are performing an

---

[14]Note that the maximum total mass in these plots extends to $35\,\mathrm{M}_\odot$. This was reduced to $25\,\mathrm{M}_\odot$ for the last part of S6. See Chapter 7 for details.

[15]As mentioned in section 5.1 hardware injections exist too. These are also used to check that the pipeline is working. However, because we cannot perform the large number of hardware injections needed for Monte Carlo simulations — as we can with software injections — they are of limited use for tuning and efficiency studies.

(a) τ_3 vs. τ_0 (b) In component mass.
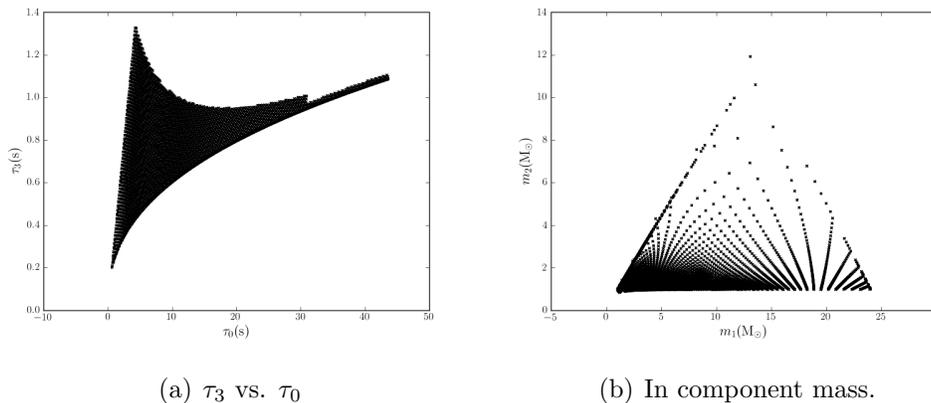
Figure 17 : A typical template bank in τ_0, τ_3 space and in m_1, m_2 space (where m_1 is the mass of the larger component mass). This plot generated from a H1 template bank file in the 10 240 toy analysis. Note that the templates are more evenly distributed in τ_0, τ_3 space. As discussed in section 5.3.4, this is because the parameter space is approximately flat in τ_0 and τ_3 [54, 55, 56, 57, 58].

injection run then, prior to launching first inspiral jobs, HIPE runs lalappps_inspinj. Inspinj generates a list of injections to perform based on the input arguments. It does not create the waveforms themselves; that is left to lalapps_inspiral (see below).

The configuration file determines how the injections are distributed in mass, mass-ratio, sky-location, orientation, and time. For a given injection DAG, a minimum and maximum component mass are specified, along with a maximum total mass. Additionally, what mass parameter to distribute the injections in must be specified. For S5 and S6 injections were chosen to be distributed uniformly in component mass. How to distribute the location of injections is also required. In S5 a galaxy catalogue was used for the location distribution [21, 22]; random galaxies were chosen for an injection to occur in. For S6, the range of the detectors was large enough to ignore inhomogeneities in the universe [10]; thus in S6 the location distribution was changed to be randomly distributed across the sky. The orientation of injections was distributed uniformly across inclination angles for both S5 and S6.

Injections are distributed randomly in time. However, to prevent too many injections from occurring in the same period, a time-step and time-interval argument are added. The time-step argument sets the average period of time between each injection, and the time-interval sets the interval around that time-step in which an

injection is randomly placed. For example, in S6 the time-step was set to $837\,\mathrm{s}$ and the time-interval was set to $300\,\mathrm{s}$. This means that if an injection occurs at time t_0, the next injection will be chosen to randomly occur in the interval $(t_0 + 837) \pm 300\,\mathrm{s}$.

We distribute injections in distance based on the range of the detectors. We are most interested in the region around the network's *horizon distance*,[16] we as this is where the efficiency quickly drops from ~ 1 to 0 (for a given false alarm rate). Since the detectors' sensitivity is mass dependent, we adjust the domain of distances in which to distribute injections according to the mass-range and type of injections being performed. Injections may be distributed uniformly in distance or in log distance. When we distribute uniformly in distance, we tend to over populate the outer regions of the range, since volume grows as r^3. If we distribute uniformly in log distance, we tend to over populate the inner regions of the range. For this reason, two injection sets are typically done for a single mass-range and injection type: one that is distributed linearly in distance and one distributed uniformly in log distance. Prior to analysis, the choice of injection ranges is based on a best-guess of what the range will be from PSD estimations at a single-detector SNR of 8. Since we cannot know the *exact* range of the network of detectors until after the analysis is complete, extra injection runs are done after the analysis. These runs are set up to better target the range around the horizon distance so as to have good statistics for upper-limit calculations.

The waveform generator used by `lalapps_inspiral` has the ability to create waveforms from several different template families, not just restricted non-spinning pN templates. Thus we check how well spinning waveforms are recovered using our non-spinning template bank. This was done for both S5 and S6 searches, as separate upper-limits were produced for spinning NSBH and BBH systems. Additionally, we can check how well various waveform families overlap with the pN approximation we use for the templates in the template bank. For example, in the S6 low-mass CBC search, NSBH injections were generated using the EOBNR psuedo-4pN model [75].

Each injection `HIPE` DAG will only draw from a single waveform family, with a give range of parameters, and with a given random-number-generator seed. To create a large number of injections, multiple injection runs are done; some of these

---

[16]Horizon distance is the furthest point the detectors can detect. From the antenna pattern shown in Figure 5, we see that this point corresponds to a binary situated directly above or below the detector, with an inclination angle of zero.

will have the same range of parameter, with only the random seed changing. When `lalapps_inspinj` runs in a given injection run, it saves the list of injections to perform to a `sim_inspiral` table that lists the times, parameters, and waveform family of the injections to create. (See section 5.4.2 for more details about the `sim_inspiral` table.) This table is saved to an XML file with naming convention:

HL-INJECTIONS_{SEED}_{USER-TAG}-{GPS-START}-{DURATION}.xml [17]

`SEED` is number given as a seed to the random-number generator in `inspinj` in the given DAG. `USER-TAG` is the user-tag given to each HIPE DAG by `ihope`. As discussed in section 5.2.2, the non-injection DAG will be named `FULL_DATA` whereas each injection run will have a unique name to distinguish it from the other injection runs. Unlike the `FULL_DATA` tag, the injections' `USER-TAG`s are set in the configuration file. For example, since we have done one injection run in our toy analysis, we have two unique user-tags: `FULL_DATA` and `BNSINJ`. One `INJECTIONS` file exists for each injection run; this file is loaded by `lalapps_inspiral` to insert the injections into the data.

5.3.6 First Inspiral

With the template bank and (for injection runs) list of injections generated, `HIPE` next runs `lalapps_inspiral` to match filter the data [63]. One inspiral job exists for each `tmpltbank` job; i.e., there is an inspiral job for each detector and for each analysis block. `Inspiral` loads the frame cache and the template bank corresponding to its block. Additionally, if `inspiral` is being run in an injection DAG, it will read in an `INJECTIONS` file.

`Inspiral` carries out the same data conditioning as `tmpltbank` (in fact, they call the same code): data is read in, low- and high-pass filtered, and re-sampled [63, 12]. If `inspiral` is given an injections file it will create the waveforms that overlap its analysis time prior to data conditioning. Waveforms are generated in the time-domain using the parameters stored in the `sim_inspiral` table and added directly to the data stream in memory. The data is then low- and high-pass filtered and re-sampled. Note

---

[17]The `HL` prefix on the `INJECTIONS` file is an anachronism from when searches were only done between the two LIGO sites. The `sim_inspiral` table actually contains injection information for all the detectors in the search.

that `tmpltbank` does not load injections. Instead, the same bank is used for both non-injection and injection runs. While this introduces a subtle difference from the real situation, the effect on the template bank is negligible, since the median estimator is used, and since the number of injections in a block is limited by the time-step and time-interval arguments given to `lalapps_inspinj`.

After the data is conditioned and the PSD estimated, `inspiral` reads in a template from the `TMPLTBANK` file, generates it (in the frequency domain), then match filters it with the data to create the (complex) SNR time series for the given template [63, 12]. This is repeated for each template in the bank file. For injection runs, an optional argument `enable-inj-filter-only` can be turned out that causes the `inspiral` to only filter segments containing an injection. This cuts down on computational time and has no effect on the analysis, since we are only interested in times around an injection (all other times are assumed to return the same result as the non-injection run).

As discussed in Chapter 3 we identify triggers by finding points where the SNR (ρ) is at a maximum. Due to noise, however, there will be multiple local maxima across the duration of a template. We must therefore employ time-domain clustering on the SNR so as to associate a single trigger with an event. This is done by using the *max over chirp length* algorithm [31]. Max over chirp length uses a sliding time window to select triggers: for every point in time, a point is only kept if there is no other point with a ρ greater than it within the chirp length of the template. Once a trigger is identified, the time at which it occurs is associated with the *coalescence*, or *end-time*, t_c, of the binary. If the SNR of the trigger exceeds our desired SNR threshold, it is kept. The SNR threshold is set in the configuration file; for both S5 and S6 it was set to 5.5.

Due to the high overlap between templates in the bank, a single event can create triggers across multiple templates. In order to associate a single trigger with a single event, we also need to cluster across the bank. `lalapps_inspiral` offers two options: time-window clustering, and *TrigScan* [76]. The time-window clustering is the simplest of the two: the time series is split up into windows with a set duration (determined in the configuration file). Within each window, the trigger with the largest SNR is kept while all others are discarded. This method has the advantage of being simple, and it guarantees that the trigger rate in a single detector will not

exceed one per the window duration. However, choosing the size of the window is difficult and somewhat arbitrary. Different templates have varying impulse responses and will "ring" for varying amounts of time depending on the strength of a signal or glitch. Thus the window is unlikely to cause only one trigger to be associated with one event. Additionally, a glitch that creates triggers at in the high-mass region of the template bank can cluster away a true signal candidate that created triggers at the low-mass region of the bank,[18] even though the glitch and the signal candidate look nothing like each other.

A more sophisticated approach is to use TrigScan clustering [76, 62]. Rather than simply use the time-dimension, TrigScan also pulls in information about the parameters of the templates to cluster. The method is similar to that of coincidence testing, discussed in section 3.4: for a given trigger, TrigScan constructs an e-thinca ellipsoid[19] with size ϵ_{ts} around the trigger using the bank metric computed by `tmpltbank`. (The value of ϵ_{ts} is determined from tuning studies and is set in the configuration file).) It then collects all triggers that fall within that ellipse. Ellipses are then constructed around each new trigger found, and more triggers are collected around them. This continues until no more triggers can be found within any ellipse. The trigger with the largest SNR amongst the collected triggers is kept and the others are discarded [76, 62]. By involving parameter information this method has the advantage that it can cluster on a good candidate on in one region of the template bank without being affected by a glitch in another region of the template bank. Also, since time is incorporated in the metric used to construct the e-thinca ellipse, the size of the cluster window in the time dimension will adjust for each template.[20] The disadvantage to this method is that it does not guarantee a maximum trigger rate. (This proved to be a problem for S6; see Chapter 7 for details.)

All surviving triggers are saved to the `sngl_inspiral` table with their template parameters, SNR, and *end time*. The "end time" is the point when the frequency in pN approximation goes to infinity [31]; it is how we identify when a trigger occurs. This table is stored in a XML file; the naming convention used is:

---

[18]Refer to figure 17.

[19]Recall from section 3.4 that an e-thinca ellipse defines the volume of parameter space around a template for which a signal with the template's parameters will fall to some probability [34].

[20]Basically, shorter-duration templates will have larger time windows since we cannot localize their parameters with as high confidence as we can longer-duration templates [76].

```
{IFO}-INSPIRAL_FIRST_{USER-TAG}-{GPS-START}-{DURATION}.xml
```

All of the `FULL_DATA INSPIRAL_FIRST` files that are created in our analysis are shown in Figure 19. There will be an equal number of `BNSINJ` files, the only difference in the names being the `USER-TAG`.

`lalapps_inspiral` is also used to calculate χ^2 values for triggers. However, since this is not used until the second stage of the pipeline, we withhold discussion of this until section 5.3.9.

5.3.7 First Coincidence

Now that we have lists of single-detector triggers we can perform the coincidence test across detectors. This is done by `lalapps_thinca` [62]. Each `thinca` job reads in multiple `inspiral` files. The number and duration of `thinca` jobs is determined by *coincidence times*. A coincident segment of time is determined by the number of detectors that are operational — i.e., in Science mode and not vetoed — during that segment. For example, if H1 and L1 are operating for the same period of time, we refer to this as H1L1-coincident time. One `thinca` job will be created for every contiguous coincidence time. This is illustrated in Figure 19. Only H1 and L1 were analyzed for the first $\sim 1700\,\text{s}$ of the analysis. Thus, one instance of `thinca`, or *job*, is run for this period. At GPS time 967230087 Virgo turns on, and for the next $2176\,\text{s}$ all three detectors are analyzed. This results in another `thinca` job being created for this H1L1V1-coincidence time. Afterward, L1 turns off, and so an H1V1 `thinca` job is created for the next $1755\,\text{s}$. This is again followed by a period of H1L1V1-coincident time.

Unlike `tmpltbank` and `inspiral` jobs, there is no minimum required duration of contiguous analysis time for `thinca`; in principle, a `thinca` job could be as short as one second. A maximum duration of $3600\,\text{s}$ is imposed to protect against memory errors. If a period of coincidence time is greater than $3600\,\text{s}$ and less than $7200\,\text{s}$, however, the time will be evenly split between two `thinca` jobs. This can be seen in the last H1L1V1-coincident segment in Figure 19. Note that no overlap is applied between `thinca` jobs.

`Thinca` performs the coincidence test using the method outlined in section 3.4: it constructs an e-thinca ellipsoid around each trigger with size ϵ_{thinca} using the metric

components computed by `tmpltbank` [34]. The size of ϵ_{thinca} is a tunable parameter and is set in the configuration file [62]. Single-detector triggers are considered coincident if their e-thinca ellipsoids overlap. A single trigger can take part in multiple coincidences if it overlaps with multiple triggers. A triple (or higher) coincidence can only occur if all three triggers in each detector overlap with each other. If one trigger overlaps with the other two, but those two do not overlap with each other, two double coincidences will be created.

Any single-detector triggers found to be in coincidence with a trigger with at least one other detector are saved to the `sngl_inspiral` table in the output XML file. One XML file is created for each `thinca` job. Note that this means a single `thinca` XML file will contain triggers from multiple detectors; how these are stored in the `sngl_inspiral` table is discussed in section 5.4. The naming convention for first coincidence files is:

$$\{\texttt{IFOS}\}\texttt{-THINCA\_FIRST\_}\{\texttt{USER-TAG}\}\texttt{-}\{\texttt{GPS-START}\}\texttt{-}\{\texttt{DURATION}\}\texttt{.xml}$$

`IFOS` is the coincidence time the `thinca` file covered; it is not necessarily the coincidence types stored in the file.[21] Any single-detector trigger that is not coincident with any other detector is discarded.

`Thinca` also has the ability to apply higher-category vetoes and do time slides. As this is not done until the second stage, we withhold discussion of this until section 5.3.10.

5.3.8 Trigbank

The completion of first coincidence concludes the first stage of the `HIPE` pipeline. To carry out the second stage of the pipeline, we must first gather all surviving triggers in preparation for the second run of `lalapps_inspiral`. This is done by `lalapps_trigbank` [63]. The number of `trigbank` jobs is determined by the number of single-detector analysis blocks and the number of coincidence times. Within each analysis block, a separate `trigbank` job is created for each coincidence time that exists in that block and for each detector. For example, in our toy analysis, the first H1

---

[21]For example, an H1L1V1-coincident time file will have prefix `H1L1V1`, yet can contain up to four coincidence types: H1L1, H1V1, L1V1, and H1L1V1.

analysis block overlaps H1L1-coincident time and H1L1V1-coincident time. There-fore, two `trigbank` jobs are created for H1 during this time, one for each coincidence time.

Each `trigbank` job loads all first `thinca` files of a given coincidence time that overlap its analysis block. All triggers from a single detector are picked out of the `thinca` files and redundancies are removed. We do this because the same template will have multiple entries in the `sngl_inspiral` table if it generated multiple single-detector triggers across the analysis block. As these files will eventually be loaded by `inspiral` to perform match filtering (see section 5.3.9, below), we only wish to keep one entry for the template, else `inspiral` will filter the same template multiple times. The results are then saved to a `sngl_inspiral` table in a `TRIGBANK` file; the naming convention is:

```
{IFO}-TRIGBANK_SECOND_{COINCIDENT-TIME}_{USER-TAG}
    -{GPS-START}-{DURATION}.xml
```

`COINCIDENT-TIME` is the coincident time from which the triggers in the file came. By adding this tag we ensure that a different file exists for each coincidence time. Figure 19 shows all of the `FULL_DATA TRIGBANK` files created in our toy analysis.

5.3.9 Second Inspiral

After the `TRIGBANK` files have been generated, `lalappps_inspiral` is run again [63]. As with first `inspiral`, the second instance of `inspiral` loads the strain data from the frame files, conditions the data, adds injections (for injection runs), and match filters the data. There are two important differences: first, χ^2 is now computed along with SNR, and a χ^2 threshold and r^2 veto are applied to triggers. Second, rather than load a `TMPLTBANK` file, a `TRIGBANK` file is used. Since the `TRIGBANK` file only contains the single-detector triggers that passed first coincidence, this reduces the number of triggers for which `inspiral` will have to calculate a χ^2 value.

χ^2 is computed for any trigger that exceeds the SNR threshold. (Note that triggers have been defined before this happens; i.e. the max-over chirp-length algorithm is still based solely on SNR.) As per the equations outlined in section 3.3 of Chapter 3, this is done by breaking the template up into bins of equal power, match filtering each bin individually, and comparing the SNR in each bin to the expected SNR in

a single bin [32, 12]. The number of bins used is set in the configuration file. For S5 and S6 16 bins were used, making the number of degrees of freedom equal to 30. This process is computationally expensive [12]; hence it is only used for triggers that survive first coincidence. Once the χ^2 value for a trigger is computed, the χ^2 threshold is applied. While the exact value of the χ^2 threshold is SNR dependent, the value of χ^2_* (see equation 3.66 in Chapter 3) is a tunable parameter [62] that is set in the configuration file. If a trigger has a χ^2 that exceeds the χ^2 threshold, it is discarded. `Inspiral` also computes an r^2 value for each trigger across a period of time that is also set in the configuration file (see equation 3.67 in Chapter 3). If the r^2 value exceeds a threshold r^2_* determined in the configuration file, the trigger is discarded. For values of χ^2_*, r^2_*, and the size of the r^2 window used for S5 see [33, 62]. The same values were used for S6. `Inspiral` does not compute effective SNR nor New SNR. Instead, the χ^2 value for surviving triggers and the number of degrees of freedom are saved to the `sngl_inspiral` table along with the triggers' other information (SNR, template parameters, end time). Later programs use this information to compute effective or New SNR as needed (cf. section 5.5.1).

Clustering across the bank (either by the hard-window method or TrigScan) is still based on SNR. However, because this occurs after the χ^2 and r^2 vetoes are applied, the trigger that survives bank clustering may not be the same as in first `inspiral`. For example, a glitch or signal may cause a trigger with a large SNR but a poor χ^2. If this template had the largest SNR across the bank, it would have been selected in the first stage. However, if the template's χ^2 or r^2 value exceeds the veto threshold, it will not survive to the bank clustering phase, and so another template will be selected.

As can be seen in Figure 19, one second `inspiral` job is run for every `trigbank` file. Since there are many more `trigbank` files than `tmpltbank` files, this may seem to defeat the purpose of limiting the number of triggers for which χ^2 is calculated. However, templates in a given `trigbank` file are only filtered through segments that overlap with the corresponding `thinca` file (the entire analysis block is used for to compute the PSD [63]). For example, in our toy analysis, there are two `trigbank` files — and therefore two second `inspiral` jobs — for the first H1 analysis block. This was because part of the way through the block, V1 turned on, and so we switched from H1L1-coincident time to H1L1V1-coincident time. The second `inspiral` job that reads in the H1L1 `trigbank` file in this analysis block will only match filter the

segments that overlap the H1L1-coincident time (which is the first 1672 seconds of the block). Likewise, the second second-`inspiral` job in this block, which is associated with the H1L1V1 `trigbank` file, will only match filter the segments that overlap H1L1V1-coincident time (which is the last 376 seconds). For this reason, the naming convention for second `inspiral`'s output XML files is:

```
{IFO}-INSPIRAL_SECOND_{COINCIDENT-TIME}_{USER-TAG}
    -{GPS-START}-{DURATION}.xml
```

By limiting the filtering in this manner we minimize the number of triggers for which we need to compute a χ^2 value [63].

5.3.10   Second Coincidence

Since many triggers will have failed the χ^2 and r^2 vetoes in second `inspiral` (and since slightly different templates may have taken their place) we must again perform the coincidence test [62]. This is carried out by `lalapps_thinca` with many of the same arguments as before. In fact, the CAT1, zero-lag coincidence job will perform exactly the same operations as its first stage counterpart; even the start/end times (which correspond to the coincidence times) will be the same. The only difference is the second stage coincidence will read in the corresponding `INSPIRAL_SECOND` files.

It is at this point that we perform time-slides and apply higher-category (i.e., > CAT1) vetoes. This results in several additional `thinca` jobs than were carried out at first stage. Specifically, for a non-injection run, there will one zero-lag and one slide `thinca` job for each category of veto analyzed.[22] We do not perform slides for injection runs, but we do apply higher vetoes.

Higher category vetoes are applied by feeding `lalapps_thinca` an ASCII file containing the list of veto segments to apply for a given detector (and by adding a `--do-veto` argument). These ASCII files are created by `ihope` at run-time from the veto XML files created by `ligolw_segments_from_cats`,[23] and are stored in the `segments` directory. A separate ASCII file is given to `thinca` for each detector. If

---

[22]Although there are, in principle, up to four categories of vetoes — see table 5.1 — not all of them have to be analyzed by `HIPE`. The specific categories that we wish to analyze are determined by the configuration file. For example, note that in Figure 19 there are no CAT4 `THINCA_SECOND` files.

[23]`Thinca` does not read the XML files directly because it was written several years before

a veto file for a detector is specified, `thinca` loads it, then removes all triggers from the given detector that have end times intersecting the veto segments. This is done prior to performing the coincidence test. As mentioned above, vetoes are applied cumulatively. Thus, the CAT2 veto file will contain the union of category 1 and 2 veto segments; CAT3, the union of 1, 2, and 3; etc. For this reason, `thinca` only needs to load one veto file per detector per job.

Slides are carried out by giving `thinca` a `--num-slides` argument, followed by arguments giving the relative offset to apply for each detector. `Thinca` uses these arguments to construct offset vectors for each slide. The offset values for each detector in a given slide are determined by the slide number and the relative offsets of the detectors. For example, if the H1 offset is 0 (set via the `--h1-slide` argument), the L1 offset is 5 (via `--l1-slide`), and V1 offset is 10 (`--v1-slide`), then for the third slide, the offset vector will be:

$$\vec{\mathcal{O}}_3 = [\Delta t_{\mathrm{H1}} \ \Delta t_{\mathrm{L1}} \ \Delta t_{\mathrm{V1}}] = [0\,\mathrm{s}\ 15\,\mathrm{s}\ 30\,\mathrm{s}] \tag{5.1}$$

The `num-slides` argument sets the total number of slides to perform. The number of slides will be twice the value given by this argument: one set of forward slides, and one set of backward slides. For example, if `num-slides` is 20, 40 total slides will be created: 20 slides with positive offsets and 20 slides with negative offsets. The number of slides to perform, and the offset for each detector is set in the configuration file. For both S5 and S6, the relative offset between the Hanford detectors and L1 was 5 s; the relative offset was 10 s between the H1 and V1 in S6. H1 and H2 were not slid with respect to each other (i.e., their relative offset was 0 for all slides) due to correlated noise between them (see Chapter 6 for details). In both S5 and S6, 100 slides were used (50 forward slides and 50 backward). These could be increased if we had a particularly loud coincident trigger; see Chapter 7 for more details.

Both the `num-slides` and offset arguments must be integers. This means that `thinca` cannot perform slides with offsets that are fractions of a second. Furthermore, due to the way `thinca` constructs the offset vectors, it is not possible to mix positive offsets with negative, nor is it possible to have two separate vectors in which two of the detectors have the same offset (unless the two detectors have the same relative

---

`ligolw_segments_from_cats`. Rather than try to adjust `thinca` to read the new XML files, we found it easier to simply convert the XML to ASCII format.

offset). For example, `thinca` cannot create the vectors $[\text{H1} = 0\,\text{s} \ \text{L1} = 5\,\text{s} \ \text{V1} = 10\,\text{s}]$ and $[\text{H1} = 0\,\text{s} \ \text{L1} = 5\,\text{s} \ \text{V1} = 20\,\text{s}]$. This limits the total number of slides `thinca` can perform for times in which there are more than two coincident detectors.

For each slide, `thinca` adds the offsets to the end times of the single-detector triggers then performs the coincidence test. This is done after vetoes are applied, so that higher-category vetoes are effectively slid around with the offset, too. Slides are performed on a *ring*: triggers that are slid past the end time of the `thinca` job are placed at the beginning of the time. This means that a `thinca` slide job does not have to load triggers from any other segment. It also means that triggers occurring in single-detector time cannot be slid into coincidence time. Thus we are safe discarding triggers that occur during single-detector time after the first stage coincidence test. Since there is no minimum required duration for a `thinca` file, this can mean that some slides will be redundant if a file is too short. Studies have shown that this is rare, however, and so it has little effect on the background analysis.

The naming conventions for second `thinca` output XML files are:

```
{COINCIDENT-TIME}-THINCA_SECOND_{COINCIDENT-TIME}_{USER-TAG}
    -{GPS-START}-{DURATION}.xml
```

for the CAT1 zero-lag files, and:

```
{COINCIDENT-TIME}
    -THINCA_SECOND_{COINCIDENT-TIME}_{USER-TAG}_CAT_{N}_VETO
    -{GPS-START}-{DURATION}.xml
```

for the Nth-higher veto category.[24] Slide files follow the same convention, except that THINCA_SECOND is replaced with THINCA_SLIDE_SECOND.

One important final note: by definition, when an detector is vetoed, it is no longer considered to be "on." This means that when we apply the higher-category vetoes, coincidence times will change. For instance, if a category-2 L1 veto comes on for t seconds during H1L1V1-coincident time, then at CAT2 (and all higher cumulative categories) that period of time is now H1V1-coincident time. The same rule applies to time slides. Second `thinca` files, however, are grouped by whatever the coincidence

---

[24]Note that the second COINCIDENT-TIME in the naming convention is unnecessary; it is simply a relic of the `trigbank` and second `inspiral` naming conventions and has no other meaning.

time was at CAT1, and this does not change after vetoes are applied (we seen this in Figure 19: all the `THINCA_SECOND` files have the same start and end times across all veto categories). This means that for higher veto categories, a `THINCA_SECOND` file will contain multiple coincidence times. Unfortunately, `thinca` does not store which triggers fall in which coincidence times, nor does it store the duration of each of the coincidence times in its output file. As discussed in Chapter 4, knowing the start and end of coincidence times is important information: uncombined FARs are computed for each coincidence type, and FARs are not combined across coincidence time. Thus later programs must re-apply the vetoes to sort out the triggers and times. This is done by `ligolw_thinca_to_coinc`, which is the first program to run in Pipedown (see section 5.5).
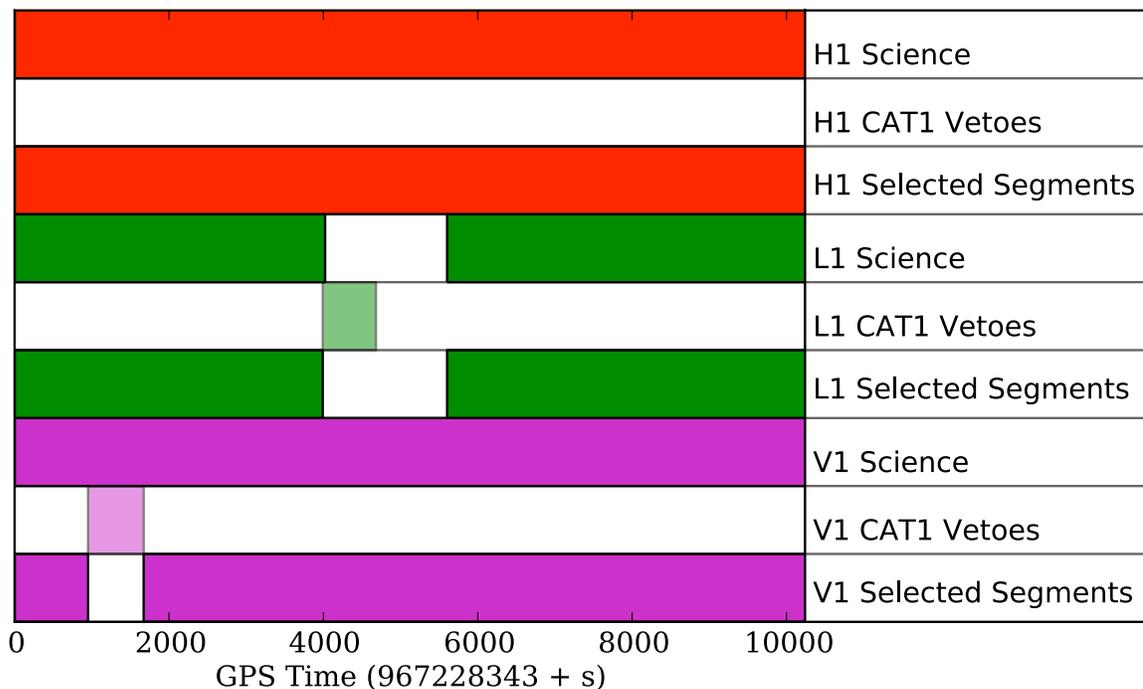
Figure 18 :   The Science, category 1 vetoes, and selected segments of H1, L1, and V1 between GPS times 967228343 and 967238583.

5.4   Data Storage

Before stepping through Pipedown, we pause to detail the table structure used to store triggers in files. Table structure is central to Pipedown: part of its purpose is
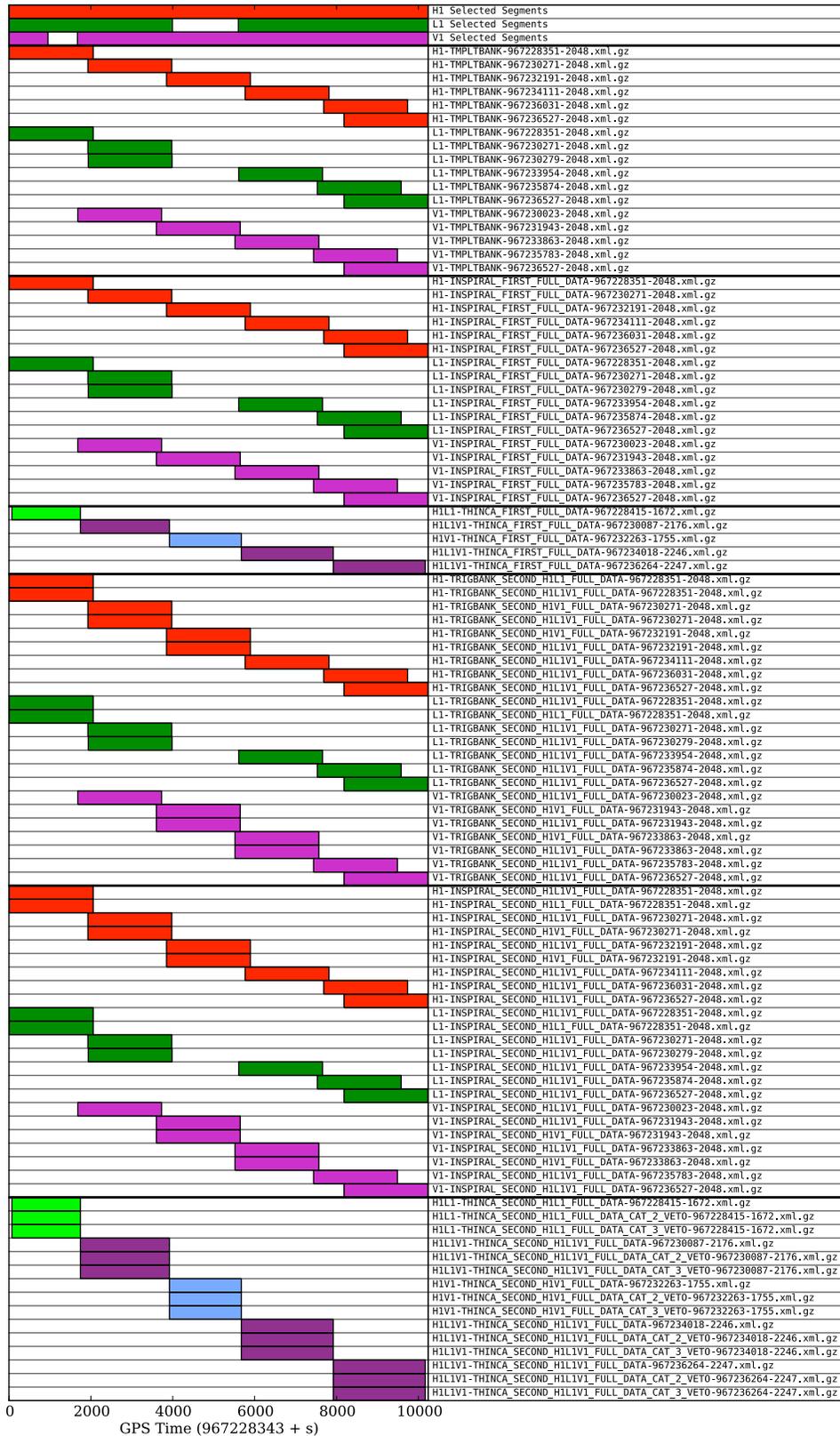
Figure 19 : The selected segments and all zero-lag FULL_DATA files created by HIPE for the segments shown in Figure 18.

to convert from older table structures to newer, more convenient ones, and all the programs Pipedown runs take advantage of these newer tables. We begin by reviewing the `sngl_inspiral` table, and how this was used to store data up to this point.

5.4.1 The `sngl_inspiral` Table

As discussed above, the `sngl_inspiral` table is the table used to store data by nearly every HIPE program: `lalapps_tmpltbank`, `lalapps_inspiral`, and `lalapps_thinca` all use it to store template and both single-detector- and coincident-trigger information. Consequently, the `sngl_inspiral` table has a large number of columns to store all needed data. Table 5.7 shows some of the relevant columns used for the CBC analysis and their purpose. There are several extra columns not discussed here.[25]

Each row in the `sngl_inspiral` table represents a single trigger or template in a single detector. Associated with each row is an `event_id` to uniquely identify each event. This provides a convenient way to retrieve information about a template or single-detector trigger. Coincident information is more difficult to store, however, since coincidence requires grouping rows together. Since the `sngl_inspiral` table was the only available way to store trigger data in a file when `lalapps_thinca` was written, a method was devised to encode coincidence in the `event_id`. The id was set to be 18 digits long. The first (starting from the left) 9 digits were the GPS time of the trigger. The next 4 digits gave the slide that the coincidence was from, where 0000 was zero-lag. The last 5 digits formed a counter that assigned a unique number to each coincidence. For example, the first coincident trigger in H1L1-coincident time in the first slide of our toy analysis has `event_id`:

$$\underbrace{967228415}_{\text{GPS time}} \quad \underbrace{0001}_{\text{slide number}} \quad \underbrace{00001}_{\text{event counter}}$$

Any rows that had `event_id`s with the same last 9 digits — i.e., if the slide number and event counter matched — are coincident triggers.

---

[25]To see all current columns of the `sngl_inspiral`, as well as any other table discussed in this section, see the `lsctables.py` code in the LALSuite repository. A URL to the repository is given in section 5.4.6.

5.4.2 The `sim_inspiral` and `process` Tables

Two other tables used by `HIPE`, as well as Pipedown, are worth mentioning: the `sim_inspiral` and `process` tables. The `sim_inspiral` table is similar to the `sngl_inspiral` table. Instead of storing triggers, it stores lists of injections. Some of the columns of the table are shown in Table 5.7. The table is indexed by the `simulation_id`. This id is simply a number: no information is stored in it, as it is with the `event_id`.

Note that both the `sim_inspiral` and `sngl_inspiral` table (as well as many other tables) have `process_id` columns. These are used to map their entries to the `process` table. This table — the columns of which are shown in table 5.7 — is used to store relevant information about the program that created the entry. Associated with the `process` table is the `process_params` table, which stores the arguments and values given to the program at run-time. For example, all the injections created by a single instance of `lalapps_inspinj` will have the same `process_id` in the `sim_inspiral` table. These will all point to a single entry in the `process` table and multiple entries (one for each argument given to `lalapps_inspinj`) in the `process_params` table. These tables are useful for troubleshooting problems and verifying the correct program versions were used when an analysis was run. They are also used to associate a set of injections with a given injection `USER-TAG`; see section 5.6.1 for details.

5.4.3 Coinc Tables

There were a few problems with using the `sngl_inspiral` table to store coincident information. First, the number of coincident events in a single file could not exceed 99 999, or else the event counter would spill over into the slide number. Likewise, no more than 9999 slides could be stored in a single file. Third, it consumed more memory than necessary. If a single-detector trigger formed coincidences in multiple slides (as is often the case), multiple entries would have to be created in the `sngl_inspiral` table for it. The values in all columns (of which there are many) for these entries would be exactly the same, except for the last 9 digits of the `event_id`. Finally, any coincident parameters, such as combined New SNR,[26] would have to be computed on

---

[26]Recall from section 3.4 of Chapter 3 that single-detector parameters and statistics are combined for coincident triggers. In the case of combined New SNR, we sum the single-detector New SNRs in

the fly. For coincident statistics that could not easily be computed on the fly, such as FARs, a `sngl_inspiral` column (meant to store single-detector information) had to be used, and the same value would be stored repeatedly for each detector in the coincidence.

To remedy this situation, coincident event tables (*coinc tables*) were developed. The coinc tables consist of the `coinc_inspiral`, `coinc_event`, `coinc_event_map`, and the `coinc_definer` table [77]. Additionally, there is the `time_slide` table, although this is considered to be a part of the *experiment* tables (see section 5.4.4). The columns in each of these tables and their purpose is shown in tables 5.7 – 5.7.

The `coinc_inspiral` table is the coincident analog to the `sngl_inspiral` table. Instead of storing single-detector information, the `coinc_inspiral` table stores *combined* statistics and parameters. As mentioned in section 3.4, how information is combined depends on the parameter or statistic. Combined New SNR is calculated by taking the quadrature sum of the constituent single-detector's New SNRs, and is stored in the `snr` column.[27] Intrinsic parameters of the templates, such as chirp mass (stored in the `mchirp` column) and total mass (stored in the `mass` column), are combined by taking the mean of the of single-detector parameters. Coincident end times are computed by taking the end time from the first detector when sorted alphabetically. For example, in an H1L1 coincidence, the coincident end time will be whatever the H1 end time is. In an L1V1 coincidence, the L1 end time will be used. Similar to the original intent of the `event_id`, a unique `coinc_event_id` is assigned to each coincidence in a file to provide a quick way to identify them. Additionally, the `coinc_event_id` is used to map entries in the `coinc_inspiral` table to entries in other tables. For every entry in the `coinc_inspiral` table there is one entry in the `coinc_event` table, as well as multiple entries in the `coinc_event_map` table with the same `coinc_event_id`.

The `coinc_event_map` table is used to map the coincident triggers in the

---

quadrature. For other parameters, see section 5.5.1.

[27]Admittedly, storing combined New SNR, not combined SNR, in a column named `snr` is a little misleading. When the column names were defined, the `snr` column was intended to be a catch-all for whatever statistic was used to rank triggers. This way, multiple searches that may not use the same ranking statistic could use the same column. In retrospect, however, doing it in this manner was probably not needed, since SQLite allows columns to be added at will (see section 5.4.7). At the very least, the column probably should have been called something along the lines of `ranking_stat`.

`coinc_inspiral` table to their constituent triggers in the `sngl_inspiral` table. As can be seen in Table 5.7, the `coinc_event_map` table has three columns: a `coinc_event_id`, an `event_id`, and a `table_name`. To map a single-detector trigger in the `sngl_inspiral` table to a coincident event, the single event's `event_id` is entered in the `coinc_event_map` table next to the `coinc_event_id` of the coincidence it is part of. Likewise, all other single-detector triggers that are a part of the coincidence will have entries in the `coinc_event_map` table with the same `coinc_event_id`. In this way, one coincident event is mapped to multiple single events. A single `event_id` does not have to be associated with one `coinc_event_id`. If a single event takes part in multiple coincidences it can also be added to the `coinc_event_map` table, with each entry being associated with a different `coinc_event_id`. By using an intermediate table we have allowed for many-to-many mappings between coincidences and single events. As a result, multiple entries do not have to be added to the `sngl_inspiral` table, and the `event_id`s no longer need to encode coincident information. `Event_id`s are therefore changed to simple counters used to index the `sngl_inspiral` table, so that each row has a unique id. This fixes the problem of counter-overflow; the number of events that can be stored in a single file is now only limited by disk or memory space.

Note that the table name from which a single event came — in this case, the `sngl_inspiral` table — is also stored in the `coinc_event_map` table. This is done so that the `coinc_event_map` table can be used to draw multiple types of maps, e.g., it is also used to draw mappings between coincident events and injections. The manner in which injection mappings are drawn is more complicated, however.

By changing the `event_id` to a simple counter, we have also freed it from storing time-slide information. Information describing which particular slide an event belongs to is instead stored in the `time_slide` table. As listed in table 5.7, the `time_slide` table has an `ifo`, an `offset`, and a `time_slide_id` column.[28] Each slide is given a unique `time_slide_id`; each slide id is repeated in the `time_slide` table for every detector that was analyzed, with each row giving the offset (in seconds) applied to that detector in that slide. For example, if H1, L1, and V1 are being analyzed, and one of the slides has the offset vector given in equation 5.1, then that slide will have three entries in the time slide table:

---

[28]Recall from Chapter 2 that "IFO" is an acronyms for "interferometer."

time_slide_id	ifo	offset
time_slide:time_slide_id:37	H1	0
time_slide:time_slide_id:37	L1	15
time_slide:time_slide_id:37	V1	30

where we have assigned the slide the arbitrary id `time_slide:time_slide_id:37`.[29] Storing time-slides in this manner makes it easier to find what offset was applied to each detector, and it allows for non-multiple slide vectors should a new version of `thinca` be written. Coincident triggers (and, via the `coinc_event_map` table, single-detector triggers) are mapped to time slides through two parallel tracks. One path is through the `coinc_event` table; the other is by way the `experiment_map` and `experiment_summary` tables, which are discussed in the next section.

The `coinc_event` table stores additional information about coincident triggers, or *events*, as well as map events to other tables. Table 5.7 shows the `coinc_event`'s columns. Every coincident event will have one entry in the `coinc_event` table via their `coinc_event_id`. Since the table also has a `time_slide_id` column, it maps the events to their respective time-slides (a many-to-one mapping). The table additionally has an `instruments` column which gives the coincident time that the coincident event occurred in, which is important for computing FARs. Coincident times are also stored in the `experiment` table, which is discussed in the next section. The `likelihood` column is used if a likelihood statistic is computed in a search. We do not do this in the low-mass CBC search, however, and so it is unused. That it exists in the table is due to the fact that the `coinc_event` table, along with the `coinc_event_map`, are search-independent tables. That is, they can be (and are) used by multiple search pipelines within the CBC group. Contrast this with the `coinc_inspiral` and `sngl_inspiral` tables, which are specific to CBC searches. An advantage of having search independent tables is that they provide an easy way to store data if a search is carried out that draws from multiple pipelines. Indeed, being able to combine results from multiple pipelines is an active field of research.

Note that we have referred to coincident triggers as coincident "events." The reason for this is a "coincidence" can occur between other things besides CBC single-detector triggers. For example, in order to know whether or not a coincident trigger

---

[29]All ids have the basic format `[originating_table]:[id_column_name]:[index_number]`. The index number is what makes each id unique within a given class.

resulted from an injection we must do *injection finding.* This involves checking the list of coincident-triggers that came out of the pipeline to see if any occurred around the time of an injection. If one is found, we draw a map between the injection (stored in the `sim_inspiral` table) and the coincident trigger (stored in the `coinc_inspiral`) table. This mapping is also a coincident "event." Thus, we see that an "event" is more abstract form of trigger.

To define what type of coincidence an event is, we use the `coinc_definer` table. This table is mapped to the `coinc_event` table via the `coinc_def_id` column. The `coinc_definer` table groups together various types of mappings; its columns are shown in table 5.7. A human-readable description of the type of mappings is given in the `description` column. For example, all coincident events that are mapped to `sngl_inspiral` triggers will be associated with a single `coinc_def_id`. The entry in the `description` column for this id will be `sngl_inspiral<-->sngl_inspiral coincidences`. Mappings between injections (which are stored in the `sim_inspiral` table) and coincident events will have a different `coinc_def_id`, and the `description` column will be `sim_inspiral<-->coinc_event coincidences`. The `coinc_definer` table is therefore useful to select coincidences formed by a specific method. For example, if two types of injection finding were being experimented with (e.g., a time-based method and an e-thinca ellipsoid-based method), the `coinc_definer` table would be needed to pick out which mappings occurred from which method. However, since the current low-mass CBC search only performs one type of coincidence test and one type of injection finding (see section 5.6.1 for details), the `coinc_definer` table is largely unused by Pipedown programs.

5.4.4 Experiment Tables

The coinc tables solved many of the problems and difficulties of saving coincident information to a file. However, neither the older table structure nor the coinc tables provided a way to store information about the experiments performed. In particular, there was no way to store the live time of a slide, nor the *data type* of a trigger, i.e., whether the trigger came from playground data, data with playground excluded, time-slides, or an injection run. Initially, there was also no way to store what coincidence time a trigger was in. This was amended by adding the `instruments` column to the `coinc_event` table. However, the `coinc_event` table only stores information if a

coincident event occurs. This means that if we perform an experiment, but nothing happens during it, we cannot store information about the experiment, such as what detectors were coincident during which times and for how long (i.e., live time). Even if no coincident triggers occur during an experiment, we still want to know information about it. This is particularly true for computing false alarm rates: we need to know all the time that was observed in every slide, regardless of whether or not a slide produced an event. The only way to keep track of experiment information, such as live times, was to store it in auxiliary ASCII files. This could be confusing, and it forced meta-data to be stored in file names.

For these reasons, the *experiment* tables were developed. The experiment tables consist of the `experiment`, `experiment_summary`, and `experiment_map` tables. Their columns are listed in tables 5.7 – 5.7. The `experiment` table lists the experiments that were performed as well as information about each one. The `gps_start_time` and `gps_end_time` columns give the period across which the experiment was performed. In `ihope`, these columns are set whatever the start and stop times were that were given to `ihope` to analyze. The `search_group` and `search` columns define what group performed the search (e.g., the CBC group, or the Burst group) and what type of search was performed (e.g., `lowmass`). Since different instrument times are mutually exclusive, and since we do not calculate FARs across them, every instrument time that was analyzed in the `search` by the `search_group` between the GPS start and end times is listed as an independent experiment in the `experiment` table. Thus the `experiment` table not only lists the experiments performed, it *defines* what an experiment *is*.

Within each experiment, we apply various vetoes, perform slides, and construct different *data types* (e.g., playground data is a distinct data type from slide data). Each of these actions results in a slightly different realization of an experiment. To define and list these realizations, we use the `experiment_summary` table. Table 5.7 shows the columns of the `experiment_summary` table. Each row has its own `time_slide_id` and `experiment_id` which maps it to the `experiment` and `time_slide` table; thus, each row gives information about every slide that was performed in each instrument time. Additionally, there is a `datatype` column to enumerate each of the data types constructed in every experiment. This can have one of five entries: `all_data`, `playground`, `exclude_play`, `slide`, and `simulation`. The `all_data`, `playground`,

`exclude_play`, and `simulation` are all zero-lag data types: `playground` is data consisting only of times occurring during playground seconds (defined in section 5.1, above), `exclude_play` is the converse, and `all_data` consists of both. The `slide` data type is any non-zero lag data. We do not define a `slide` playground because playground slides are rarely used.[30]

The `simulation` data type describes zero-lag data that has software injections in it. This allows us to easily separate injection runs from non-injection runs. In order to separate various injection runs we use the `sim_proc_id` column. The `process_id` of the `lalapps_inspinj` job that created a set of injections is stored in this column. All injections in a single injection run are created by a single `inspinj` job; thus all these injections will have the same `process_id` in the `process_id` column of the `sim_inspiral` table. By storing the same value in the `sim_proc_id` column of the `experiment_summary` table, we map the two tables together. Note that this mapping simply means that a given set of injections were performed during a realization of an experiment. It does not mean that all events that occurred during that realization were from injections. To pick out what events were due to injections and which are due to noise, we rely on mappings in the `coinc_event_map` table.

The final mapping in the `experiment_summary` table is the `veto_def_name` column. This column stores the name of a set of vetoes applied in the given realization of an experiment, e.g., `VETO_CAT3_CUMULATIVE` (the veto names are determined by `ligolw_segments_from_cats`). This column maps the row to the `segment_definer` table, which in turn maps to the veto segments, stored in the `segment` table. (See section 5.4.5 for more information on segment tables.) In this manner, the type of vetoes, and their segments, that were applied during an experiment can be retrieved without needing external files.

All of the `experiment_summary` columns we have reviewed so far serve to categorize realizations of experiments. The `duration` and `nevents` columns, on the other hand, store data about a realization. Perhaps the most important column is the `duration` column. This column stores the amount of live time that exists in during a coincident detector time during the experiment period (again, stored in the `experiment` table and mapped to via the `experiment_id`) for a given veto category

---

[30]Looking at all-data slides is not considered un-blinding the analysis.

in a given slide for a given data type. The live time is computed by a single program (`ligolw_cbc_compute_durations` — see section 5.6.2 below) for all experiment realizations, which stores the results in the `duration` column. This makes live time retrieval quick and easy for other programs. The `nevents` column stores the number of coincident events that occurs during an experiment realization.

The final experiment table is the `experiment_map` table. This table maps coincident events to realizations of experiments defined in the `experiment_summary` table. It has two columns: an `experiment_summ_id` and a `coinc_event_id`. Having a table devoted solely to mapping coincident events and experiments allows for many-to-many mappings. This is needed: an experiment will have many events in it, and a coincident event can occur in multiple experiment realizations. For example, all zero-lag events will belong to the `all_data` data type as well as either `playground` or `exclude_play`. The `experiment_map` allows all of these events to be efficiently stored in the same file, without needing to rely on file names to separate them. The effect of vetoes and which events survive which category — a question that comes up often — can also be ascertained by simply looking at what mapping exist between a coincident event and `experiment_summ_id`s of various veto categories.[31]

5.4.5 The Segment Tables

As they have already been mentioned a several times, and are used by several codes, we briefly describe here the segment tables. The segment tables consist of the `segment`, `segment_definer`, and `segment_summary` tables [70]. The columns of the `segment` and `segment_definer` tables are shown in tables 5.7 and 5.7, respectively. The `segment_summary` table is not shown nor described here, as it is not used by Pipedown programs.

The `segment_definer` table serves to identify a collection of segments by name. For example, all of the CAT3 veto segments for a given detector will be given a single `segment_def_id` and have the name `VETO_CAT3_CUMULATIVE`. The start and end times of each segment in the collection are listed in the `segment` table; the mapping between segments and their grouping in the `segment_definer` table occurs via the

---

[31]Currently, Pipedown stores different veto categories in separate files, and so checking veto categories via experiment tables is not currently taken advantage of. Combining vetoes into a single file is planned for the future, however.

segment_def_id. As mentioned above, entries in the experiment_summary table map to the veto segments applied via the name in the segment_definer table. The mapping is not done using the segment_def_id because each detector will have its own entry, and therefore its own segment_def_id, in the segment_definer table. Using these mappings, we can re-construct the periods of time vetoed in any realization or slide in an experiment.

For more information on the segment tables and how they are used by the segment database, see [70].

5.4.6 Other Tables

There are many other tables in use by searches in the CBC group as well as the rest of the LSC in addition to the ones presented here. These include meta-data tables such as the search_summary tables, as well as a large number of tables used by other searches and search groups. Since these tables have little, if any, impact on the low-mass CBC search, we do not describe them here. For a complete list of all the tables, and their columns, see the lsctables.py code in the LALSuite repository. This can be accessed online at: http://www.lsc-group.phys.uwm.edu/cgit/lalsuite/tree/glue/glue/ligolw/lsctables.py

5.4.7 File Formats

It will be useful to also make a note of the file formats used to store data. As described throughout section 5.3 all HIPE programs store data in LIGO_LW XML files. This is convenient because XML files can opened, viewed, and edited using any text editor; at the same time, they provide a standard format that can be interpreted by code [71].

The downside to XML files is that the entire file must be loaded by a program in order to parse its data. This puts a limit to how much data can be stored in a file, and has proven to be troublesome when computing statistics, such as FARs, that require large quantities of data. It also means that many files can only be handled by computers with large amounts of memory, which limits the number of nodes in a cluster a that a program can be executed on. Even then, periods of high glitch rate occasionally exhaust memory, thus requiring more aggressive clustering. Finally, it

can be difficult to quickly sort through large amounts of data to find a few triggers of interest, such as the ones with the smallest FAR. Doing so requires writing code that can filter through large numbers of files as well as access to library packages that know how to interpret `LIGO_LW` XML files.

For these reasons, Pipedown converts `LIGO_LW` XML files to SQLite databases [78], which are used by a number of its programs. SQLite has a number of advantages. Database files remain on disk: when a program reads from a database, it does not need to load the database into memory. Instead, it queries only the information it needs to perform a specific task. Once an operation is completed on that data, the results can be dumped back to the database and cleared from memory. Thus, file size is limited only by disk space, of which there is far more — and which is much cheaper — than memory. SQLite is also fast: it automatically invokes a number of optimization schemes which allows it to quickly read and write data, without needing special commands. Thus, when coding, we can concentrate more on what to do with data rather than how to get it. SQLite is also flexible. With `LIGO_LW` XML files, what columns exist in a table must be defined in `lsctables.py`. The only way extra columns can be added is to edit `lsctables.py` to add the column, re-install it, then re-create the XML. This can be frustrating when doing investigations into new statistics. SQLite, on the other hand, does not demand static table definitions. It can create new columns on the fly and add them to any table in the database. This is taken advantage of by a number of Pipedown programs, such as `ligolw_cbc_repop_coinc` and `ligolw_cbc_cfar`; see sections 5.5.4 and 5.6.3, respectively, for details. Finally, SQLite is well supported: it is developed and maintained by dedicated professional programmers, and is open-source software [78]. There also exists a Python interface for SQLite so that it can be used from within Python code, even though SQLite itself is written in C. Additionally, there exists a free SQLite command-line tool (installed on all the LIGO clusters) which can parse a database without needing support code to read data tables, as is the case with XML files. One can therefore quickly search data from the command line without needing to write new tools.

The downside to SQLite is database files are not human-readable. Either the command-line tool must be used to see data, or code must be written to print data out. Additionally, while the flexibility to create arbitrary columns on the fly is useful for investigations, once a statistic has been settled on, we wish to standardize the

column. Otherwise it could be difficult to know what is stored where when going back over results created by different people at different times. For these reasons, XML files are considered to be the "final" data products produced by the CBC pipeline. One of Pipedown's goals is to take all the data analyzed by `ihope` and disseminate it down to a few XML files containing a handful of "loudest" triggers that are of interest.

5.5 Pipedown in Detail

We now step through Pipedown in detail, using Figure 16 and our toy analysis as a guide. Pipedown is run after all `HIPE`runs have completed. It combines data from all these runs — including zero-lag, time-slides, and all injection runs — into a single database. FARs are calculated and a number of plots are generated using this database. Although the experiment table structure allows multiple veto categories to be stored in a single database, Pipedown currently keeps each veto category in a separate database. Thus, the pipeline shown in Figure 16 is repeated for each veto category.

5.5.1 `ligolw_thinca_to_coinc`

The first program run in Pipedown is `ligolw_thinca_to_coinc`. This program reads second `THINCA` files, and uses `sngl_inspiral event_id` to construct coinc and experiment tables, as well as convert the `event_id` to a counter. It also loads the veto-segments file corresponding to the given category that was produced by `ihope` at run time. The veto segments are used to determine what coincident-detector times triggers belong to. (As discussed in section 5.3.10, when higher category vetoes are applied, some coincidences will be moved to a different coincident-detector time than what is indicated in the file name. `Thinca` does not store this information, so it must be reconstructed.)

For non-injection runs, `thinca_to_coinc` combines the zero-lag and slide second `THINCA` files into a single XML file. Thus, for every two `FULL_DATA` second `THINCA` files, there is one `THINCA_TO_COINC` file. The output file has the same naming convention as the second `THINCA` files, except `THINCA_(SLIDE_)SECOND` is replaced with `THINCA_TO_COINC`. Although the veto-segments file is loaded by `ThincaToCoinc`, the

veto-segments stored in the veto file are not added to the output XML file.

In addition to constructing coinc and experiment tables, `ligolw_thinca_to_coinc` also computes some coincident parameters, storing them in the `coinc_inspiral` table. The coincident end time, combined chirp mass, combined total mass, and combined New SNR are computed for each coincident trigger.

5.5.2 `ligolw_sqlite`

The next step in the Pipedown pipeline is to convert XML files to SQLite databases. This is done by `ligolw_sqlite`. As can be seen in Figure 16, `ligolw_sqlite` turns all of the `thinca_to_coinc` XML files sharing a common `USER-TAG` into one database. (Blue arrows on the diagram indicate SQLite databases; black arrows indicate XML files.) The naming convention for the output database is:

```
{ANALYZED-IFOS}
    -{USER-TAG}_CAT_{N}_VETO_RAW_CBC_RESULTS
    -{GPS-START}-{DURATION}.sqlite
```

where `ANALYZED-IFOS` are all the IFOs that are in the database, `N` is veto category[32], `GPS-START` is the `ihope`start time, and `DURATION` is the `ihope`duration. For example, in our toy analysis, we would have two CAT3 databases:

```
H1L1V1-FULL_DATA_CAT_3_VETO_RAW_CBC_RESULTS-967228343-10240.sqlite
```

and

```
H1L1V1-BNSINJ_CAT_3_VETO_RAW_CBC_RESULTS-967228343-10240.sqlite
```

At this point the `FULL_DATA` database contains both zero-lag and slide data.

In addition to being able to add XML files to a database, `ligolw_sqlite` can also extract XML files from databases. As a result, it is used many times in Pipedown to go between SQLite databases and XML files, as can be seen in Figure 16.

---

[32]Note: in keeping with the `HIPE`naming conventions, for CAT1 databases, the `CAT_N_VETO_` part of the file name is excluded.

5.5.3 `ligolw_cbc_dbsimplify`

ID columns must be unique in a single file. For example, there can only be one coincident trigger in the `coinc_inspiral` table with
`coinc_event_id:coinc_event:coinc_event_id:0` in a given file. Since the IDs are simply counters, however, there is no way to ensure that they are unique across files. Thus, when `ligolw_sqlite` combines XML files into a single database, it has to increment all the table IDs to prevent "collisions." For example, if a coincident event in the database already has `coinc_event_idcoinc_event:coinc_event_id:0`, and we wish to add another `THINCA_TO_COINC` file that also has a coincident event with `coinc_event_id coinc_event:coinc_event_id:0`, this latter ID will be incremented to one plus whatever the largest ID is in the database. The same goes for all other IDs. This ensures that independent entries across files remain independent when combined into a single database or file.

However, we may not wish to keep all entries independent when combining files. For example, all the `THINCA_TO_COINC` files in a single `ihope` run will have the same entries in the `experiment` table. When `ligolw_sqlite` adds all of these files into a single database, it will keep all of the entries independent by incrementing the `experiment_id`s. In our toy analysis — in which we have five `FULL_DATA` `THINCA_TO_COINC`, with each one containing four `experiment` entries (one for each possible instrument time) — each `experiment` entry will be repeated five times in the `FULL_DATA` `THINCA_TO_COINC` database, leading to 20 total entries when we should only have 4. The same is true of any other table that stores information that spans multiple `THINCA_TO_COINC` files, including the `experiment_summary`, `time_slide`, and `coinc_definer` tables. The same is also true of the `veto_definer`, `sim_inspiral`, and segment tables. However, as these tables only exist in the veto file and the `INJECTIONS` file, we avoid having to deal with them by adding them to the database after all the `THINCA_TO_COINC` files have been added.

To remove these redundancies, `ligolw_cbc_dbsimplify` is run. This program loads a database created by `ligolw_sqlite` and removes any redundant entry from the `experiment`, `experiment_summary`, `time_slide`, and `coinc_definer` tables. It then updates the corresponding IDs in all other tables to point to the correct entry.

This is done in-place, i.e., the output database has the same name as the input database. Due to the redundancy problem, `ligolw_cbc_dbsimplify` should always be run after `ligolw_sqlite` is used to add files to a database.

5.5.4 `ligolw_cbc_repop_coinc`

After all the `RAW` databases have been created and simplified, Pipedown runs `ligolw_cbc_repop_coinc`. This program reads single-detector information from the `single_inspiral` table, computes a specified combined statistic, and saves it in the `coinc_inspiral` table. For example, if told to compute combined SNR, it will calculate the quadrature sum of the single-detector triggers' SNRs and save it to the `coinc_inspiral` table. If told to compute combined chirp mass, it will calculate the mean single-detector chirp mass and save it.

What column `repop_coinc` saves the combined parameter to is specified using the `output-column`. Repop_coinc takes advantage of SQLite's ability to create new columns in a table. The column does not need to already exist in the `coinc_inspiral` table, nor does it need to be apart of the "blessed" columns specified in the table definitions in `lsctables.py`. If the column does not exist, `repop_coinc` will simply create it, then store the data. If the column does exist, the data in it will be overwritten. This makes `repop_coinc` extremely useful for experimenting with new statistics: one can take a database created by Pipedown, and add to it any statistic they wish without needing to update and re-install the LALSuite code base. Since the experimental statistic can be added to the `coinc_inspiral` table, it can sit alongside currently used statistics (i.e., older data need not be overwritten), which makes it easy to compare the new statistic to older ones. However, columns that are not in the table definitions in `lsctables.py` will not be in XML files that are extracted from a database. Therefore, if a column is going to be used long-term, it should be added to list of "blessed" columns.

As `thinca_to_coinc` already populates most of the `coinc_inspiral` columns needed by the low-mass CBC search, `repop_coinc` is currently bypassed in the low-mass pipeline. However, as new statistics are invented or desired parameters are added, they will be computed by `repop_coinc` rather than `thinca_to_coinc`. This is partly because `thinca` and `thinca_to_coinc` are planned to be phased out in favor of a new coincidence program that uses the coinc and experiment table structure

natively. The primary purpose of such a program is simply to find coincidences between single-detector triggers; once coincidences have been drawn, it will be left to `repop_coinc` to compute and store all combined statistics.

5.6 `ligolw_cbc_cluster_coincs`

Although we apply some template bank-wide clustering in `lalapps_inspiral`, experience has taught us that a single event can still cause many triggers after `inspiral`'s clustering. Further, because it does not compute New SNR, bank clustering in `lalapps_inspiral` is based on SNR. We therefore try not to be too aggressive with clustering in `inspiral`, as it may not necessarily pick the best candidate. Instead, we do a final round of bank-wide clustering in Pipedown using `ligolw_cbc_cluster_coincs`.

`ClusterCoincs` clusters triggers based on combined statistics in the `coinc_inspiral` table. What statistic it uses is determined by the `ranking-stat` argument, set in the configuration file. In the low-mass CBC search we use combined New SNR.[33] The clustering is done using a sliding time window in a similar manner as max-over chirp length: if there is another trigger within $\pm t_{\mathrm{win}}$ of a given trigger's end time with a ranking-stat value greater-than it (or less-than it — see footnote 33), the trigger is deleted. The difference from max-over chirp-length is that the time window, t_{win}, used by `cluster_coincs` is a fixed value set in the configuration file. In the low-mass CBC search we set the window to 10 s for both S5 and S6.

Since all slides and zero-lag, as well as playground, all-data, and exclude-play, triggers exist in the same database, the `experiment_map` table is used to ensure the triggers are only clustered within their slides and data type. We do not wish to cluster triggers across data types: it is possible for a trigger to survive clustering in playground, but be clustered away in all-data. Thus, clustering is carried out by first deleting mappings between coincident events and `experiment_summ_id`s in

---

[33] In addition to `ranking-stat`, there is another argument, `rank-by`, which tells `cluster_coincs` whether to cluster on large values of the ranking-stat, or small values. Thus, the program can cluster using statistics in which larger is better, such as New SNR, or ones in which smaller is better, such as FAR.

the `experiment_map` table; only events with the same `experiment_summ_id` are compared to each other.[34] Events that have no mappings left to any `experiment_summ_id` are deleted entirely from the database.

In addition to binning events by data type, slide, and coincident-detector time, `cluster_coincs` can optionally bin by a coincident parameter and *coincidence type*[35] Two arguments can govern binning by a parameter range: `param-name`, which sets what column in the `coinc_inspiral` table to use, and `param-ranges`, which set the values to bin by. For coincidence type, `cluster_coincs` can be told to bin by the detectors that took part in the coincidence (using `group-by-ifos`), or the *number* of coincident detectors (using `group-by-multiplicity`). For example, if `group-by-ifos` is set, H1L1- and L1V1-coincident triggers would not be clustered together. If `group-by-multiplicity` is set, H1L1- and L1V1-coincident triggers would be clustered together, but H1L1V1-coincident triggers would not. If neither is set, all types are clustered together. These same arguments are used by several other programs to bin triggers, including `ligolw_cbc_cfar` and a number of the plotting programs.

In both S5 and S6, triggers were grouped by coincidence type when being clustered. In S6 triggers were additionally grouped into the same chirp-mass bins that are used to compute uncombined FARs (see section 5.6.3 for the bin boundaries). We chose to cluster in chirp-mass bins out of concern that higher trigger rates and larger New SNRs in one bin would cause triggers that would otherwise have lower uncombined FARs to be lost in other bins. This would negate some of the benefits of binning triggers when computing FARs. However, as a result of this choice, a loud event that created off triggers in all bins would have three candidate triggers: one in each bin. In this case, we resolved to use the trigger that had the lowest uncombined FAR. If a trigger was loud enough to have a FAR of 0 in multiple bins,[36] we would try to measure a FAR, either by doing more slides or by folding in data from other runs. If a FAR could still not be determined, we would use the one with the largest combined

---

[34]As a result, events must also be in the same coincident-detector time for them to be clustered together.

[35]By coincidence type, we mean by detectors that are coincident in a coincident trigger.

[36]Recall from Chapter 4 that a 0 FAR means that a coincident trigger is louder than all slide, or background, triggers. This means that the true FAR of the trigger is smaller than 1 over the background live time. Thus, 0 FARs represent upper limits on the false alarm rate of an event.

New SNR. These considerations became important in the investigation of the blind injection made during S6; see Chapter 7 for details.

As seen in Figure 16, `cluster_coincs` runs separately on each `RAW` database. It writes the results to a new database which has naming convention:

```
{ANALYZED-IFOS}-{USER-TAG}_CAT_{N}_VETO_CLUSTERED_CBC_RESULTS
    -{GPS-START}-{DURATION}.sqlite
```

In our toy analysis we thus have two clustered databases at CAT3:

```
H1L1V1-FULL_DATA_CAT_3_VETO_CLUSTERED_CBC_RESULTS-967228343-10240.sqlite
```

and

```
H1L1V1-BNSINJ_CAT_3_VETO_CLUSTERED_CBC_RESULTS-967228343-10240.sqlite
```

5.6.1   The Injection Branch

At this point in Pipedown, additional steps are carried out on the `CLUSTERED` injection databases that are not carried out on the `FULL_DATA` database. First, `ligolw_cbc_dbaddinj` is run on each injection database. This adds the `INJECTIONS` file that was created by `lalapps_inspinj` to the database with the corresponding `USER-TAG`. Included in the `INJECTIONS` file is the `sim_inspiral` table. The same code that `ligolw_sqlite` uses to add files to databases is used by `dbaddinj`. We do not need to run `dbsimplify` after it, however, because there are no tables in the `INJECTIONS` file that will lead to redundancies when added to the database. We do not use `ligolw_sqlite` to add the file because, in addition to adding the `INJECTIONS` file, `ligolw_cbc_dbaddinj` updates the `sim_proc_id` column of the `experiment_summary` table with the `process_id` of the instance of `insinj` that created the injections.

After `dbaddinj` is run, `ligolw_sqlite` is run to convert each injections database into a single XML file. The name of the XML files are the same as the `CLUSTERED` database, except the `.sqlite` file extension is replaced with `.xml`. This conversion must be done because the next program, `ligolw_inspinjfind`, can only read XML files.

`Inspinjfind` performs *injection finding*: it maps injections that were performed to triggers that were caused by the injection. To decide whether or not a trigger

was created by an injection `inspinjfind` uses a 1 s time window; i.e., any coincident trigger that occurs within ± 1 s of an injection is considered to have been caused by that injection. Note that this means that multiple events can be mapped to a single injection. Simply using a time window does increase the chance for random noise events to get mapped to an injection. We use such a large window for purposes of calculating upper limits. When calculating upper limits we are only interested in injections that are louder than the loudest event. Since (in principle) the only difference between the `FULL_DATA` results and the simulation runs are the injections, any event in the simulation runs that is louder than the loudest full-data event must be from an injection. Thus, we are safe simply using a time window for calculating upper limits. Admittedly, however, this method is not ideal when using injections to tune the pipeline. For that, an e-thinca-ellipsoid based approach might be better. Investigations into implementing such a method are planned.

`Inspinjfind` works on the XML file in place; i.e., the mappings are added directly to the file. The maps between the injections (stored in the `sim_inspiral` table) and coincident event are stored in the `coinc_event_map` table. How exactly the maps are done is more complicated than how single-detector events are mapped to their coincident counterparts; we do not discuss it here.

5.6.2 Preparing the Final Database

After `lalapps_inspinjfind` has run on each of the injection XML files, they are added to the `FULL_DATA CLUSTERED` database. This is done by `ligolw_sqlite` followed by `ligolw_cbc_dbsimplify`. Next the veto XML file (the same one given to `thinca_to_coinc`) is added to the database. Thus, the `FULL_DATA CLUSTERED` database contains all triggers from the zero-lag, slide, and injection runs, as well as all injections made (both "found" and "missed") in all runs, and the veto segments applied to the run. The `FULL_DATA CLUSTERED` database is therefore the final data product from the pipeline: it is from this that false alarm will be computed (and saved) and all plots will be generated. The other databases — i.e., all of the `RAW` and injection databases — are only intermediate data products.

Now that the veto segments are in the database, `ligolw_cbc_compute_durations` is run to compute the live time in each slide and zero-lag. To do the computation, `compute_durations` must know what times were analyzed (pre-vetoes). It gets this

from the `thinca` start and stop times, which is stored in one of the meta-data tables (specifically, the `search_summary` table, which is not described here). The `thinca` start and stop times are needed because these define the ring boundaries within which triggers and vetoes were slid around. For each entry in the `experiment_summary` table, the single-detector veto segments are retrieved (using the `veto_def_name` column to get the appropriate segments from the `segment` table) then slid around on the `thinca` rings according to the offset vector associated with the entry (retrieved from the `time_slide` using the `time_slide_id` column). Subtracting the duration of the vetoes from the total analysis time gives the live time of the entry. For the `playground` and `exclude_play` data types, the playground segments (computed from the GPS times) are additionally intersected with the analysis segments. The results are saved to the `duration` column of the `experiment_summary` table.

5.6.3   Computing False Alarm Rates

All of the pieces are now in place in the `FULL_DATA CLUSTERED` database to compute the FARs of the triggers. FARs are computed using the algorithm described in section 4.6: an "uncombined FAR" is computed in bins defined by chirp-mass and coincidence type, then "combined" across bins using the uncombined FAR. Both the uncombined and combined FARs are calculated using `ligolw_cbc_cfar`. `cFar` is run twice. On the first pass, the `group-by-ifos`, `param-name` and `param-ranges` arguments are given to `cFar`, which causes it to compute the uncombined FARs. As outlined in section 4.6, the FAR for each trigger is calculated by counting the number of background triggers in the same coincidence type and chirp-mass bin that have a new SNR greater-than-or-equal to it. Specifically, `cFar` uses the `experiment` and `experiment_summary` tables to grab all of the `slide` triggers in each coincident-detector time, and it uses the `ifos` and `mchirp` columns in the `coinc_inspiral` table to apply the bins. The background time is retrieved by summing the `slide exerpiment_summary` entries' live time. For `slide` triggers, triggers nor live time within that slide are used to compute their FARs. The uncombined FARs are saved to the `false_alarm_rate` column of the `coinc_inspiral` table.

Combined FARs are computed when `ligolw_cbc_cfar` is run again. This time, coincident triggers are not binned by chirp mass nor coincidence type, and the uncombined FARs computed on the first pass are used as the ranking statistic. This

result is saved to the `combined_far` column of the `coinc_inspiral` table.

With the completion of the second `cFar` job, the `FULL_DATA CLUSTERED` databases have obtained their final form. All other jobs launched by Pipedown either create plots, summarize the most significant events in short lists, or run follow-ups on loud triggers. The programs that carry these out are described in the next few sections.

5.6.4 IFAR Plots

Pipedown launches several plotting programs to summarize the results for analysts. Perhaps the most significant is `ligolw_cbc_plotifar`. `PlotIFAR` creates cumulative histograms of the triggers as a function of the *inverse* false alarm rate, or IFAR. Plots are created of both inverted uncombined FAR and combined FAR; figure 5.7 shows a sample of each. Both plots show the expected background distribution, which is simply 1/IFAR, as a dashed black line. The yellow regions show the expected background $\pm\sqrt{N}$ and $\pm 2\sqrt{N}$ where N is the number of counts. These regions are the expected variance of the background, which at high N corresponds to the 1 and 2σ significance bands in a Gaussian distribution.[37] The gray lines show the actual background distribution: each line represents a single slide.[38] The zero-lag distribution is plotted by the colored symbols. In the uncombined plot, the symbols are determined by the chirp-mass bin they represent and the color is determined by the coincident detector. The dashed colored lines show where the background for a particular detector combination (represented by the lines' colors) and chirp-mass bin (represented by the dash pattern) "runs out"; i.e., they are points of maximum (minimum) background FAR (IFAR). The uncombined plots are useful for understanding characteristics of the data. For example, we do not see any high mass H1L1V1-coincident triggers (purple squares) in the plot. This is expected, because we can see that the background for high mass H1L1V1-coincident triggers (the purple dashed line furthest to the right)

---

[37]The 1 and 2σ significance bands in a Gaussian distribution are the regions we expect $\pm 34\%$ and $\pm 47.5\%$ of the triggers in the distribution to fall.

[38]Note that the actual background appears to disagree with the expected variance (yellow regions) at high IFAR. This is due to the difference between Gaussian and Poisson distributions. At low N the Poisson distribution becomes asymmetric around the mean, so that the variance does not agree with the familiar 1 and 2σ bands of a Gaussian. If the yellow regions mapped out σ — i.e., if they mapped out where the regions were we expect to find a given percentage of the triggers — instead of variance then they would appear to agree with the gray lines.

runs out at an IFAR greater than the expected background at a cumulative number equal to one. Thus the probability of getting a H1L1V1-coincident high mass trigger in a single experiment realization is less than one. In the combined plot, all zero-lag triggers are plotted as blue triangles.

If a trigger is louder than all the background we set its FAR equal to zero. As discussed in Chapter 4, this means that we have placed an upper limit on the FAR equal to $1/T_b$ where T_b is the total background live time. On IFAR plots, zero-FAR triggers are therefore placed at the point corresponding to $x = T_b$ and an arrow pointing to the right is added to the trigger to indicate that the true IFAR is somewhere in the region to the right.[39] Such triggers are of utmost interest to us since they represent events that could potentially be a GW signal. The first thing we look for when looking at an IFAR plot is whether or not any trigger is louder than all the background.

Since the combined IFAR plot quickly conveys whether or not a trigger is louder than all the background — i.e., whether or not there is a GW candidate — as well as how significant triggers are and how consistent the zero-lag distribution is with the background, it is our primary detection plot. When we open boxes it is the first thing we look at. For example, in the sample combined IFAR plot shown in Figure 20(b), we see that the loudest zero-lag event has a IFAR of ~ 0.02, corresponding to a FAR of ~ 50 per year. There is therefore no GW candidate in this data. Additionally, since the zero-lag is consistent with background across the entire region, we can be fairly confident that we are treating zero-lag and slides the same, and that there is not something wrong in our analysis.

5.6.5 `PlotCumHist` and `PlotSlides`

In addition to IFAR plots, Pipedown also generates cumulative histograms as a function of combined New SNR using `ligolw_cbc_plotcumhist`. A sample plot is shown in Figure 5.7. To create the plot, bins of equal width in combined New SNR squared are constructed. Within each bin, the number of triggers that have a new SNR squared greater-than or equal-to the left-edge of the bin is plotted on the y-axis.

---

[39]Since the loudest background trigger will always have a zero-FAR, this point can be seen in the combined plot where the gray lines end; in the uncombined plot the point is marked by the solid black line.

Zero-lag results are indicated by the blue triangles; there is one triangle per-bin. For slide triggers, the cumulative number is computed by counting the number of triggers in each slide, then taking the mean of the counts. The black crosses show the mean background number in each bin, and they yellow regions indicate the standard deviation of the counts.

These cumulative histograms are used to check the SNR distribution of coincident triggers. We do not use cumulative histograms as our primary detection plot because they do not quantify the significance of triggers as IFAR plots do. We also cannot easily judge the significance of triggers across chirp-mass bins and coincidence types using these plots. If all of the categories are plotted together, as they are in Figure 5.7, then the effect of different rates and new SNR distributions, as discussed in Chapter 4, is not accounted for. It is possible to make `PlotCumHist` plot each category on separate plots; even so, this results in a large number of plots with no way to compare significance across plots.

Another program that creates plots useful for checking data is `ligolw_cbc_plotslides`. Sample plots are shown in Figures 5.7 and 5.7. These plots are used to check the distribution of live times and trigger rates across slides. Figure 5.7 shows the duration-per-slide for H1L1V1- and H1L1-coincident time from the same six weeks of data used for the `PlotCumHist` and `PlotIFAR` plots. Note that the H1L1V1-coincident time decreases with increasing slide number whereas H1L1 time increases. This is a common feature; it is due to the vetoes sliding around (the plots shown are after CAT2 and 3 vetoes have been applied). As described in section 5.3.10, when vetoes slide around their alignment across detectors changes. This causes times that were formerly in H1L1V1-coincident time to drop to double-coincident time (i.e., either H1L1-, H1V1-, or L1V1-coincident time), and double to drop to single-detector time. Although double-coincident times lose time to single-detector times, they gain time from triple. No double-coincident times can be promoted to triple, however, because the third detector is off for the entire double-coincident time segment. The net effect is that H1L1V1-coincident time only loses time while double-coincident times both lose and gain.

Figure 5.7 shows a sample rate-per-slide plot created bye `PlotSlides`. `PlotSlides` creates a separate plot for each coincidence type and all types combined (the color scheme for the zero-lag bar is the same as used by `PlotIFAR`). Shown is the rate of

H1L1V1-coincident triggers in H1L1V1-coincident time.

5.6.6 `PlotFM`

None of the plots discussed so far make use of the simulation data type (i.e., the data that contains injections). To make various plots showing found and missed injections Pipedown runs `ligolw_cbc_plotfm`. `PlotFM` is a versatile program that can create scatter plots of many different variables. In its default mode it creates plots of injected *decisive distance* versus chirp mass. Decisive distance is the second furthest effective distance out of the analyzed detectors;[40] it is so-named because at least two detectors are needed to create a coincident trigger. Even if a binary has an effective distance small enough to be seen in one detector, if the next largest effective distance in another detector is outside the that detector's range, no trigger will be created in that detector, and so we will have no coincident trigger. Thus, decisive distance is the main arbiter of whether or not we expect to see an injection. Figure 5.7 shows an example decisive distance versus chirp mass plot. "Found" injections (as determined by `ligolw_inspinjfind`) that had combined FARs equal to zero are plotted as blue stars. Injections found with non-zero combined FARs are plotted as colored circles, where the color corresponds to the FARs. Red crosses indicate "missed" injections. These plots are therefore useful to check how sensitive the detectors are, and whether or not this matches what is expected from the PSD. Nearby missed injections and close injections with poor FARs are followed up. These can indicate a problem with an analysis, or can point to data quality issues that may be dealt with by vetoing.

By making use of Python's `eval` function, `PlotFM` can set the x- and y-axes to be any function of any of the columns in the `sim_inspiral` table or `coinc_inspiral` table. What functions to plot are set on the command line, or, if running in Pipedown, in the configuration file. Thus, `PlotFM` can create any number of desired plots found-missed plots, which are useful for investigating the effects of changing tuning parameters and the efficacy of new statistics on injections. As an example, Figure 25 shows two additional plots currently created when Pipedown runs. The top plot is the fractional difference in recovered and injected chirp mass versus the difference and recovered and injected end times. This gives an indication of what "found" injections

---

[40]Recall from Chapter 2 that *effective* distance is the distance to a binary if it were optimally oriented.

are actually injections and which ones are probably due to noise triggers that happen to be within the time window `inspinjfind` uses to associate coincident triggers with injections. The bottom plot shows the difference in recovered and injected end times for each injection type. Not surprisingly, the best recovery is for BNS injections. This is because BNS waveforms have the most inspiral cycles in band, allowing for the most accurate timing reconstruction.

The second plot also shows that `PlotFM` can plot against strings as well as functions. Although the injection tag is not a column in the `sim_inspiral` table, it adds the column in memory using the `process_id`. Note that in this plot and in plot 25(a) there are no red crosses. Whenever a recovered parameter is plotted (which is any column in the `coinc_inspiral` table), no missed injections are plotted since recovered parameters are undefined for them.

`PlotFM` plots have been used for a number of studies in S6; see Chapter 7 for examples.

5.6.7 PlotROC

The program `lalapps_cbc_plotroc` creates *receiver operator characteristic (ROC)* plots. This program is *not* launched by Pipedown. However, it is often used to compare results from pipelines using different ranking statistics and vetoes. For this reason we describe it here.

`PlotROC` can read in multiple `FULL_DATA CLUSTERED` databases created by Pipedown. It then computes the "relative volume and time", VT, to which a search was sensitive to as a function of combined FAR. Recall from section 2.3 of Chapter 2 that rate-upper limits on CBCs go as $1/VT$, where T is the live time and V is the volume to which the search was sensitive at the false alarm rate of the loudest event. We wish a search to maximize VT since this will increase our chance of detecting one or more GW signals. Even if we do not detect a GW in a search, maximizing VT leads to a smaller rate-upper limit; as these limits approach the expected CBC rates, the results become more astrophysically relevant. However, since we must blind the analysis, (see section 5.1) we cannot know in advance what the FAR of the loudest event is going to be.

ROC plots give us a sense of how large VT will be in a search without forcing us to un-blind the analysis. This allows us to tune various parameters and vetoes prior

to un-blinding the results. Examples of ROC plots used in the S6 analysis can be seen in Figures 7.2.1 and 7.2.3. For each FAR plotted on the x-axis, `PlotROC` calculates VT by retrieving all injections that are labelled as "found" by `inspinjfind` with a FAR less-than or equal-to the given FAR. Since the injections are labelled as found, we assume that a GW signal that was emitted from a binary at the same physical distance would be detected by our pipeline with the same FAR as the injection. Thus by using the *distance*[41] of each injection we can calculate the volume of space to which the search was sensitive to at that FAR. To get a measure of the time, we sum the volumes calculated from every injection that has a FAR less-than the given FAR. Recall from section 5.3.5 that injections are uniformly distributed in time at semi-random intervals. The distribution is set up so that on average, we will get an injection every $800 \, \text{s}$. Thus the number of injections that are performed is proportional to the period of time searched [79]. Of course, by simply adding the volumes we end up with a somewhat arbitrary number in Mpc^3. However, since ROC plots are used to *compare* searches that used different tuning parameters, we are only interested in the *relative* value between the searches. Thus, all of the searches are normalized by dividing the largest sum of volumes from a single search at the largest false alarm rate, giving us a relative measure of the size of VT in each search. For this reason, the largest ROC curve always terminates at 1 (cf. Figure 7.2.3 in Chapter 7). Since large VT leads to higher detection rates, the search that has the highest (i.e., closest to the top of the plot) curve has (ostensibly) the better tuning parameters.

There are a few subtleties to ROC plots that must be considered, however. As discussed in section 5.6.1, it is possible to get noise triggers within the injection-finding time window of an injection. This is particularly true for injections that have larger effective distances, since it is more likely that any triggers they create will be clustered away by a noise trigger.[42] We can see that effect in figure 25, in which triggers with high FAR are randomly scattered across the recovered parameter plots. In other words, these injections have been labelled as "found" when they have not been found at all. This problem does not affect us when calculating rate-upper limits.

---

[41]By "distance," we mean the actual distance to the binary, regardless of antenna pattern; i.e., this is not the effective distance.

[42]Recall from Chapter 2 that SNR is inversely proportional to effective distance. Thus, the larger the effective distance, the smaller the SNR, which leads to a larger FAR.

As discussed in section 2.3 any coincident trigger in the simulation data type that is found with a false alarm rate smaller than the loudest trigger in the all-data data type must be from an injection, since the only difference between the simulation and all-data data types is injections are added to the former. In ROC plots, however, we are considering all possible FARs. Thus it is likely that at larger FAR (to the right of the plot), coincident triggers that have been labelled as "found" injections are most likely due to noise. Therefore, when considering ROC plots, we should focus on the left side of the plot, at lower FAR.

For these reasons, when performing tuning studies, we use both ROC and `PlotFM` plots. `PlotFM` is used to check for found and missed injections, and to see how well injections are recovered. ROC plots are used to give an approximate measure of how one search quantitatively compares to another. By balancing the strengths and weaknesses of the two types of plots we can establish what tuning parameters to use. To see examples of how this was done in S6, see Chapter 7.

5.6.8 `PrintLC` and `MiniFollowups`

While plotting programs are useful for quickly evaluating large amounts of data, exact information about specific triggers is hard to ascertain. To list more detailed information about events of interest we use `ligolw_cbc_printlc`.

`PrintLC` ranks triggers according to a given statistic — say, combined FAR — then prints all the information stored in the `coinc_inspiral`, as well as relevant information in the `experiment` and `experiment_summary` tables, for the top ten triggers. The output table can either be in HTML, `LIGO_LW` XML, or "wiki" format. ("Wiki" means the table is formatted for use in the CBC group's wiki page.) When run in Pipedown a separate loudest event list is created for each instrument time and saved to an XML file. The naming convention is:

```
{INSTRUMENT-TIME}
    -FULL_DATA_CAT_{N}_VETO_LOUDEST_{DATATYPE}_EVENTS_BY_{RANKING-STAT}_SUMMARY
    -{GPS-START}-{DURATION}.xml
```

`RANKING_STAT` is the ranking-stat used to rank the triggers (in our case, `COMBINED_FAR`). `DATATYPE` is one of `ALL_DATA`, `PLAYGROUND`, `EXCLUDE_PLAY`, or `SLIDE`; a separate file is created for each of these. These files only contain a single, customized `loudest_events` table that contains the columns `PrintLC` prints. In addition to this file, Pipedown also has `PrintLC` generate a `LIGO_LW` XML file containing all of the

data tables, pared down so that they only contain information relevant to the loudest events. These additional files have the same naming convention, sans the `_SUMMARY` part proceeding the `GPS-START` time.

`PrintLC` adds a column for the end time of the trigger in UTC, which doubles as a hyperlink to the *daily* `ihope` page for the day the trigger occurred on, and it replaces the `coinc_inspiral` table's `ifos` column with one that doubles as a hyperlink to the *e-log* page for each detector on that day. The daily `ihope` page is an automatically-generated HTML page that is created at the end of each day [80]. The page is created by running a shortened version of `ihope`; in this version, `lalapps_inspiral` is run once with χ^2 on using a smaller template bank. Additionally, no coincidence test is performed. The page is used as a DQ tool to get a sense of how the CBC search will respond to data on a given day, and to quickly assess the efficiency of automatically generated DQ flags. Figure 5.7 shows a screen shot of a sample page. For more details on daily `ihope`, see [80].

The e-log page is an online notebook used to keep track of events that occur at each of the observatories. Entries are added by operators, on-site instrumentalists, and visiting scientists. These entries provide valuable insight into the causes of DQ issues that would be difficult to ascertain from instrumentation alone. For example, if a truck drives on the site, someone will record it, or if a study of equipment is being done, explanations of the study and the results are added. A screen shot of a sample e-log page is shown in Figure 5.7.

`PrintLC` has the ability to add single-detector information to the `loudest_events` table in the summary file. We do this for slide triggers: since the `sngl_inspiral` table contains the un-slid end times, printing the single end times along with the coincident loudest slides, we can quickly see what events most hurt our search. This ability was one of the most useful applications of Pipedown: it allowed to us to loudest-slide information to fine tune vetoes during S6. For more details, see Chapter 7.

After `PrintLC` has run, Pipedown runs `minifollowups`. `MiniFollowups` takes for input the two XML files `PrintLC` generated for a given data type and the cache file generated by `ihope`. For every trigger in the `SUMMARY` file, `MiniFollowups` uses the information stored in the full XML file to track how the time evolved through the pipeline. Relevant data files (using the cache file) and a SNR-versus-time plot of the single-detector triggers within 10 s of the event time is made for each

stage (`INSPIRAL_FIRST`, `THINCA_FIRST`, `INSPIRAL_SECOND`, `THINCA_SECOND`) in the `HIPE` pipeline. These plots are saved to an HTML page.

`MiniFollowups` also launches *Omega scans* [81] of the GW channel in each coincident detector around the time of the loudest events. Omega scans generate time-frequency spectrograms of any of the detectors' data channels, including auxiliary environmental monitors. It is therefore an often-used tool for data quality investigations. However, for speed and simplicity, `MiniFollowups` only generates an Omega scan of the GW channel in each detector (`LDAS-STRAIN` in the LIGO detectors, and `h_16384Hz` in Virgo).

After the "minifollowup" page and Omega scans have finished, `MiniFollowups` adds links to the pages in the "mini followup" and "omega scan" columns of the `loudest_events` table in the `SUMMARY` XML file (these columns were created by `PrintLC`, but not populated). The table is then converted to HTML format and saved so that it can later be viewed in a web-browser. The naming convention of the file is the same as the `SUMMARY` XML file, with `.xml` replaced with `.html`.

Figures 5.7 and 5.7 show an example of what the loudest-events table looks like for the loudest `all_data` and `slide` events, respectively. Note that the single-detector information is added to the slide table. Figures 5.7 and 5.7 show the minifollowup page and Omega scan of the loudest `all_data` event; Figures 5.7 and 5.7 show the same for the loudest `slide` event. In this case, the loudest `all_data` event was a hardware injection. The chirp pattern can clearly be seen in the H1 scan (Figure 28(a)). Also note the sharp spike in triggers in the `MiniFollowups` plot. Contrast this to the `slide` event: the L1 Omega scan is clearly not a chirp, and the MiniFollowup plot shows an elevated trigger rate in L1 around the time of the event.

Note, however, that the lack of a noticeable chirp pattern in the Omega scan does not necessarily mean there is no signal there. Indeed, below a SNR of about 10, it becomes difficult to see a pattern. This fact is made evident by the L1 Omega scan of the hardware injection (Figure 28(b)): while the SNR in H1 is ≈ 24, the SNR in L1 is ≈ 9 (most likely due to the injection's sky-position and orientation). As a result, no chirp can be seen in the L1 scan. This highlights an important point: all of these follow-up tools and scans are not used to determine if an event is from a gravitational wave. The probability an event is a GW detection is based solely on the FAR. Rather, the follow-up tools are used to better understand what types of

glitches affect the performance of the pipeline. Glitches can thereby be classified; if an environmental or instrumental cause can be found, the detector can be fixed or vetoes can be created, thereby improving the search.

5.6.9 `PrintMissed` and `PrintSims`

To get more details about found and missed injections, Pipedown runs `ligolw_cbc_printmissed` and `ligolw_cbc_printsims`. These programs are similar to `PrintLC`: they create tables detailing information about specific events. Instead of printing information about `all_data` and `slide` events, however, these programs print information about injections.

   `PrintMissed` lists information about missed injections. These are injections for which no coincident event could be found within the one second window used by `ligolw_inspinjfind`. We are most interested in the closest missed injections, particularly if they are within the range that the PSD suggests we should be be able to find them. Thus, when run in Pipedown, `PrintMissed` prints the closest 10 missed injections by decisive distance. Since there are obviously no coincident detectors for missed injections, the decisive distance is determined by taking the second farthest effective distance out the detectors that were on (and not vetoed) during the injection. `PrintMissed` then returns all of the information stored in the `sim_inspiral` table for that injection, along with information about the experiment from the `experiment` and `experiment_summary` table. Like `PrintLC`, `PrintMissed` also gives hyperlinks to the daily `ihope` and e-log pages for the day that the missed injection occurred on.

   Figure 5.7 shows example tables produced by `PrintMissed` when run in Pipedown. For each injection type, two tables are generated: one that combines all of the injection runs together, and a separate table for each injection type. Figure 34(a) shows the table with all injections together, and Figure 34(b) shows the table for the `BNSLOGINJ` injection run. (In the analysis from which this table came, `BNSLOGINJ` were BNS injections distributed uniformly in log distance.) Notice that in this second table there are minifollowup links and links to Omega scans. Like `PrintLC`, the `PrintMissed` results are saved to a summary table in either HTML, XML, or wiki format. When run in Pipedown the individual injection run tables are saved to XML and passed to `MiniFollowups`. As with the `all_data` results, `MiniFollowups` creates a page summarizing how the injection fared through the pipeline and it launches Omega

scans of the time surrounding the injection. The minifollowup page for injections contains additional information not found in the `all_data` and `slide` pages. A table of injected parameters is added, along with a table summarizing whether or not the injection was found or missed (using a time window as the arbiter) at each stage in the pipeline. Figure 5.7 shows a screen shot of this latter table and the `INSPIRAL_FIRST` plot for a missed injection.

When used in conjunction with `PlotFM` plots, these `PrintMissed` tables allow us to spot and quickly follow up injections that we think should have been found. For example, the found/missed plot shown in Figure 5.7 is from the same analysis as the `PrintMissed` tables shown in Figure 5.7. Looking at the plot, the closest missed injection (a red cross with a chirp mass of $\sim 1.3\,\mathrm{M_\odot}$ at a decisive distance of about $30\,\mathrm{Mpc}$) is clearly in a range that it should have been found: other injections of the same mass and distance are all found with zero FARs. Looking at the table in Figure 34(a) we see that this is a `BNSLOGINJ` injection; going to the table in 34(b) we can access its minifollowup page and Omega scans. The table and `INSPIRAL_FIRST` plot shown in 5.7 are for this injection. In the `INSPIRAL_FIRST` plot there appears to be some sort of transient in L1 a couple seconds before the injection. This is confirmed by the L1 Omega scan, which is shown in Figure 36(b): a short-duration, broadband glitch is clearly visible $\sim 2\,\mathrm{s}$ prior to the end time of the injection.[43] Going back to the minifollowup table in Figure 35(a) we see that injection was found well in both H1 and L1 at `INSPIRAL_FIRST` and `THINCA_FIRST`, but no trigger was found in L1 at `INSPIRAL_SECOND`. We know that at `SECOND_INSPIRAL` χ^2 is calculated and the χ^2 threshold and r^2 veto is applied. We thus have a picture of why the injection was missed: the glitch two seconds prior most likely inflated the injection's χ^2 value, causing it to fail either the χ^2 test or the r^2 veto.

`PrintSims` is used to list information about "found" injections (as determined by `ligolw_inspinjfind`). It can print all of the columns that are in the `coinc_inspiral` table along with the matching injections' `sim_inspiral` information. We are most

---

[43]Note that the injection is not visible in the Omega scan. This is not because of it has small SNR. Since software injections are only added to the data in memory by `lalapps_inspiral`, they are not visible to Omega scans, which are generated from the frame files. The scans are centered on the end-time of the injection, however, and so we can get a sense of what was going on during the injection even if we cannot see the injection itself.

interested in the closest "quietest found" injections. Ideally, we would like to know more about the closest injections that have a FAR larger than the loudest `all_data` coincident trigger.[44] We cannot know this prior to un-blinding the analysis, however; since the `PrintSims` results are used to check the pipeline, we must choose a different FAR. Typically `ihope` is run on periods that are a few weeks long. With 100 slides, this means that any injection that is quieter than the loudest slide event will have a combined FAR of at least 1 per few years. Any event with that large of a FAR would not be considered as a candidate detection. Thus, we define "quietest found" as any injection that has a non-zero FAR. Pipedown has `PrintSims` sort these by decisive distance. Like `PrintMissed`, two tables are created for each injection type: one for all the injections together, and one for each injection type. An example of each of these tables is shown in Figure 5.7. The individual injection-type tables are passed through `minifollowups` and a minifollowup page similar to the missed injection's page is created, along with Omega scans. We can thereby follow-up quietest found injections in the same manner that we did the missed injections.

5.7 Tying it All Together: The `ihope` Page

The completion of all of the `minifollowup` jobs marks the end of Pipedown, and therefore the end of the `ihope` pipeline. By this point we have many files and plots containing a wealth of information sprawled across the `ihope` directory. To bring all of these files together into one easily-accessible place, we run `write_ihope_page`. This program copies the most useful plots and files to a directory that is accessible to all LSC and Virgo members through the internet. It then generates a web page that displays the plots and tables in click-able menus. Figure 5.7 shows a screen shot of one of these `ihope` pages. Shown is the IFAR plot with hardware injections left in the data (note the arrow on the loudest zero-lag triggers, indicating they have zero FARs) and the corresponding `PrintLC` table. Clicking on the links in the table will open up the other pages, such as the minifollowup page and the Omega scans. Note that this is only one subsection; the right menu bar provides links to many other sections summarizing playground data, all injection runs (which includes the `PlotFM`

---

[44]Recall again from section 2.3 that the FAR of the loudest all-data event is what determines the FARs of injections for which we set the sensitive volume.

plots discussed above, along with the `PrintMissed` and `PrintSims` tables), and other auxiliary information about the run.

The page shown in Figure 5.7 is a page for which the analysis has been un-blinded. When we do carry out a search for CBCs, a blinded `ihope` page is initially generated. The format of the page is the same, but only plots and tables containing playground data are shown; `all_data` information is off-limits (and the frog is not coming out of a box). Using this page, we check and follow-up the loudest slide triggers, as well as the missed and quietest found injections. If a new glitch class is found for which a cause can be determined, we create a veto for it (this would be a category 2 or higher veto). We then re-run the `THINCA_SECOND` stage and Pipedown, applying the new vetoes. The new blinded page is checked and, if everything looks good, an un-blinded page is generated. We then look at this page to see if we have found any gravitational waves.

Pipedown and the `ihope`page, in the current form described here, were not developed until after S6 had started. Prior to this, and for S5, scripts were run by hand to do the equivalent steps. However, the results were not as easily accessible, and so this method of checking loudest slide triggers to develop vetoes was not implemented until after Pipedown was completed, in the latter-half of S6. In Chapter 7 we detail some of the vetoes and tuning decisions that were made as a result of these loudest slide studies.

Column	Purpose
`ifo`	Name of detector the trigger/template belongs to, e.g., `H1`.
`end_time`	End time (in integer GPS seconds) of the trigger (not used by `tmpltbank`).
`end_time_ns`	The fractional seconds of the end time, in nanoseconds.
`template_duration`	The duration of the template (in seconds).
`eff_distance`	The effective distance to the binary (in Mpc). For templates, this is set to 1; for triggers, this is calculated using equation 3.36.
`mass1`	The component mass of one of the objects in the binary (in $M_\odot$).
`mass2`	The other component mass.
`mtotal`	The total mass of the binary (in $M_\odot$).
`mchirp`	The chirp mass, $\mathcal{M}$, of the binary (in $M_\odot$).
`eta`	The symmetric mass ratio, η, of the binary.
`tau0`	τ_0 (see equation 3.43)
`tau3`	τ_3 (see equation 3.44)
`snr`	The SNR, ρ, of the trigger.
`chisq`	The χ^2 value of the trigger.
`chisq_dof`	The number of χ^2 degrees of freedom.
`Gamma[0-9]`	The components of the bank metric at the trigger/template. (There are 9 of these).
`process_id`	Unique value used to map the trigger/template to the process that created it.
`event_id`	A unique value to identify the event. In `HIPE` this is used to draw coincidences. In Pipedown it is used as an index of the table.

Table 4 : Commonly used columns of the `sngl_inspiral` table. Not all columns are shown.

Column	Purpose
`waveform`	The waveform family used to generate the injection, e.g., `TaylorT4threePointFivePN`.
`geocent_end_time`	The end time of the injection as measured from the center of the Earth (in GPS seconds).
`geocent_end_time_ns`	The fractional seconds of the Geo-centric end time (in nanoseconds).
`[h,l,v]_end_time`	The end of time of the injection as measured at LHO, LLO, and the Virgo observatory, respectively. (There are three columns, one for each site).
`[h,l,v]_end_time_ns`	The fractional seconds of the LHO/LLO/Virgo end time (in nanoseconds).
`mass1`	The mass of one of the objects in the injection (in $M_\odot$).
`mass2`	The other component mass.
`mchirp`	The chirp mass, $\mathcal{M}$, of the injection (in $M_\odot$).
`eta`	The symmetric mass ratio, η, of the injection.
`distance`	The physical distance to the injection (in Mpc).
`longitude`	The longitudinal coordinate of the injection, as measured on Earth.
`latitude`	The latitude of the injection, as measured on Earth.
`inclination`	The inclination angle between the angular momentum vector of the injection and a line of site drawn from the center of the Earth to the center of the binary.
`spin1[x,y,z]`	The x, y, and z magnitudes of the spin of one of the injection's masses, as measured in the binary's coordinate system. (These are three separate columns.)
`spin2[x,y,z]`	The spin components of the other mass.
`f_lower`	The frequency at which the injection was started.
`f_final`	The GW frequency of the injection at coalescence.
`eff_dist_[h,l,v]`	The effective distance to the injection, as measured at LHO, LLO, and Virgo, respectively (in Mpc). (These are three separate columns.)
`process_id`	The `process_id` of the program that created the entry.
`simulation_id`	Unique id used to identify the entry.

Table 5 : Some of the columns of the `sim_inspiral` table.

Column	Purpose
program	The name of the program.
version	Currently used to store the git-hash tag of the program.
cvs_repository	Currently used to store the status of the git repository from which the program came.
cvs_entry_time	Currently used to store the GPS time that the program was last committed to git.
node	The cluster node that the program was executed on.
username	The UNIX user-name of the person that ran the program, or that launched the DAG that launched the program.
unix_procid	The process id assigned to the program by UNIX while it was running.
start_time	The GPS time that the program began running.
end_time	The GPS time when the program terminated.
ifos	The detectors analyzed by the program. Only used by HIPE programs.
process_id	A unique id to identify the entry.

Table 6 : Relevant columns of the process table. Not shown are the comment, jobid, and domain columns as they are rarely used.

Column	Purpose
program	The name of the program (also stored in the process table).
process_id	Maps the entries to a process in the process table.
param	An argument given to the program, e.g., num-slides might be here for a thinca entry.
type	The data type of the argument, e.g., float, or lstring.
value	The value given for the argument. In our num-slides example, this might be 50. If the argument requires no value (i.e., if it is just a flag), then this is empty.

Table 7 : Columns of the process_params table.

Column	Purpose
`ifos`	The detectors that took part in the coincidence. This is comma-separated, e.g. `H1,L1`.
`end_time`	The coincident end time (in integer GPS seconds).
`end_time_ns`	The fractional seconds of the coincident end time (in nanoseconds).
`mchirp`	The combined chirp mass, $\mathcal{M}_c$, of the coincidence.
`minimum_duration`	The duration of the shortest template in the coincidence (in seconds).
`snr`	The combined *new SNR*, ρ_{nc}, (not SNR) of the coincidence.
`false_alarm_rate`	The uncombined FAR of the coincidence (in yr^{-1}).
`combined_far`	The combined FAR of the coincidence (in yr^{-1}).
`coinc_event_id`	A unique value to identify the coincidence.

Table 8 : The columns of the `coinc_inspiral` table and their purpose.

Column	Purpose
`coinc_event_id`	A unique value to identify coincidences and to map them to single-detector events.
`event_id`	The id of one of the coincident event's constituents. This may not necessarily point to a column named "event_id"; e.g., to map an injection to a coincident events, the injection's `simulation_id` will be stored here.
`table_name`	The name of the table that the `event_id` is found in.

Table 9 : The columns of the `coinc_event_map` table and their purpose.

Column	Purpose
coinc_event_id	Used to map entries to coincident events.
time_slide_id	Used to map entries to a slide (see time_slide table).
coinc_def_id	Used to map entries to a coinc-definer entry (see coinc_definer table).
instruments	Instruments that were on at the time of the coincidence. Also stored in the experiment table.
likelihood	Likelihood statistic of the coincidence. This is not used for low-mass CBC searches.
process_id	Unique value used to map coincidences to the process created them.

Table 10 : The columns of the coinc_event table and their purpose.

Column	Purpose
search	Name of search that produced a class of coincidences, e.g., inspiral.
description	Human-readable entry describing the type of co-incidence, e.g., sim_inspiral<-->coinc_event coincidences.
search_coinc_type	Integer to identify the type of coincidence.
coinc_def_id	Unique value used to index each row in the table.

Table 11 : The columns of the coinc_definer table and their purpose.

Column	Purpose
instrument	One of the analyzed detectors.
offset	The offset (in seconds) applied to the detector.
time_slide_id	Used to identify a slide; each slide has a unique id.
process_id	Unique value used to map the slides to the process that created them.

Table 12 : The columns of the time_slide table and their purpose.

Column	Purpose
experiment_id	Unique id to index the row.
instruments	The instruments that were on for the experiment, e.g., H1,L1,V1. If a coincident event happens during the experiment, then this is the instrument-time of the event.
search_group	The search group that carried out the experiment, e.g., cbc.
search	The type of search done, e.g., lowmass.
gps_start_time	The GPS start time of the experiment.
gps_end_time	The GPS end time of the experiment.
comments	Add a comment to the experiment. (Not typically used.)
lars_id	The LARS id of the experiment. This is a number meant to index all searches.

Table 13 : The columns of the experiment table and their purpose.

Column	Purpose
experiment_summ_id	Unique id to index the row.
experiment_id	Maps the row to an experiment in the experiment table.
time_slide_id	Maps the row to a slide in the time_slide table.
veto_def_name	The name of the vetoes applied, e.g., VETO_CAT3_CUMULATIVE. This can be used to map the row to the collection of veto segments to the segment_definer table.
datatype	The data type of the row. This can either be all_data, playground, exclude_play, slide, or simulation.
sim_proc_id	If the data type is simulation, this maps the row to an injection set in the sim_inspiral table via the set's process_id.
duration	The live time of the slide.
nevents	The number of coincident events that occurred in the slide.

Table 14 : The columns of the experiment_summary table and their purpose.

Column	Purpose
experiment_summ_id	Maps the row to a particular instantiation of an experiment in the experiment_summary table.
coinc_event_id	Maps the row to a coincident event.

Table 15 : The columns of the experiment_summary table and their purpose.

Column	Purpose
process_id	The process id of the process that created the group of segments.
segment_def_id	Unique id to identify the entry.
ifos	The detector(s) that the segments are from.
name	A name to identify the segments, e.g., VETO_CAT3_CUMULATIVE.
version	The version of the flag in the segment database that was used to generate the segments.
comment	A user-added comment about the segments. Not typically used.

Table 16 : The columns of the segment_definer table.

Column	Purpose
process_id	The process id of the process that created the segment.
segment_id	Unique id to identify the segment.
start_time	Start time of the segment, in GPS seconds.
start_time_ns	Fractional seconds of the start time, in nanoseconds.
end_time	End time of the segment, in GPS seconds.
end_time_ns	Fractional seconds of the end time, in nanoseconds.
segment_def_id	Maps the segment to an entry in the segment_definer table.

Table 17 : The columns of the segment table. Note: the nanosecond columns are not used in the segment database, and so are currently not used by ligolw_segment_query, ligolw_segments_from_cats, nor any Pipedown programs. This may change in the future.
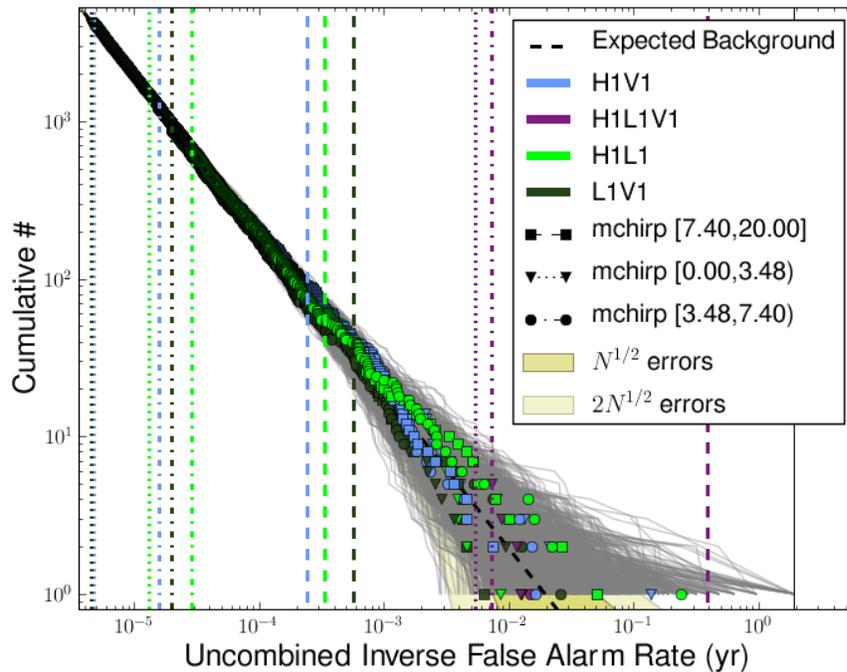
(a) Uncombined IFAR plot
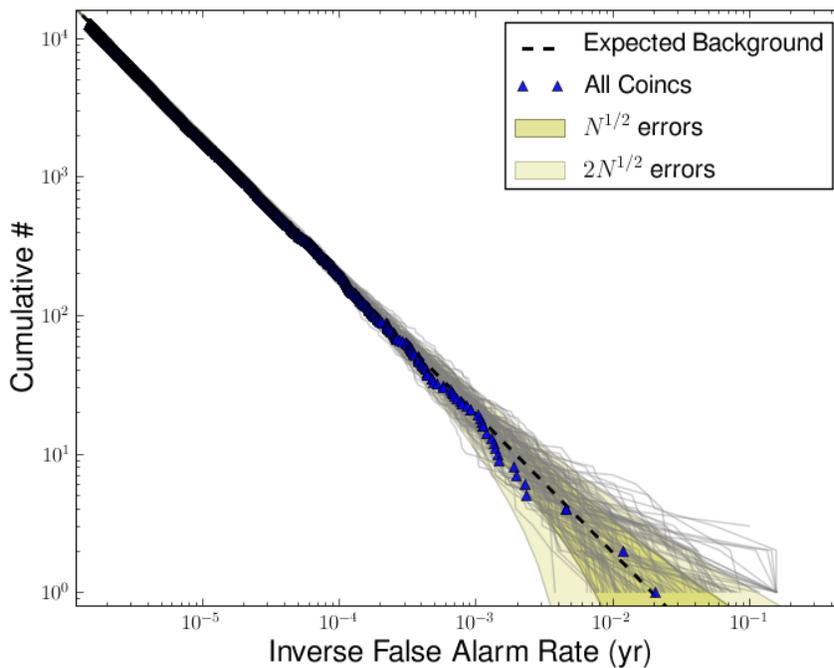
(b) Combined IFAR plot

Figure 20 : Sample IFAR plots created by `ligolw_cbc_plotifar`. Both plots taken from ~ 6 weeks of S6 data, between GPS times 931035296 and 935798487. In both plots, the black dashed-line indicates the expected background distribution, yellow-shaded regions are the expected background $\pm\sqrt{N}$ and $\pm\sqrt{2N}$, and gray lines are slide distributions when treated as zero-lag. The blue triangles in the bottom plot indicate the zero-lag distribution of combined IFARs. The top plot shows the uncombined IFAR distribution, with each symbol representing a separate chirp-mass bin and each color a coincident-detector combination. Colored dashed lines indicate maximum (minimum) background FAR (IFAR). See section 5.6.4 for more details.
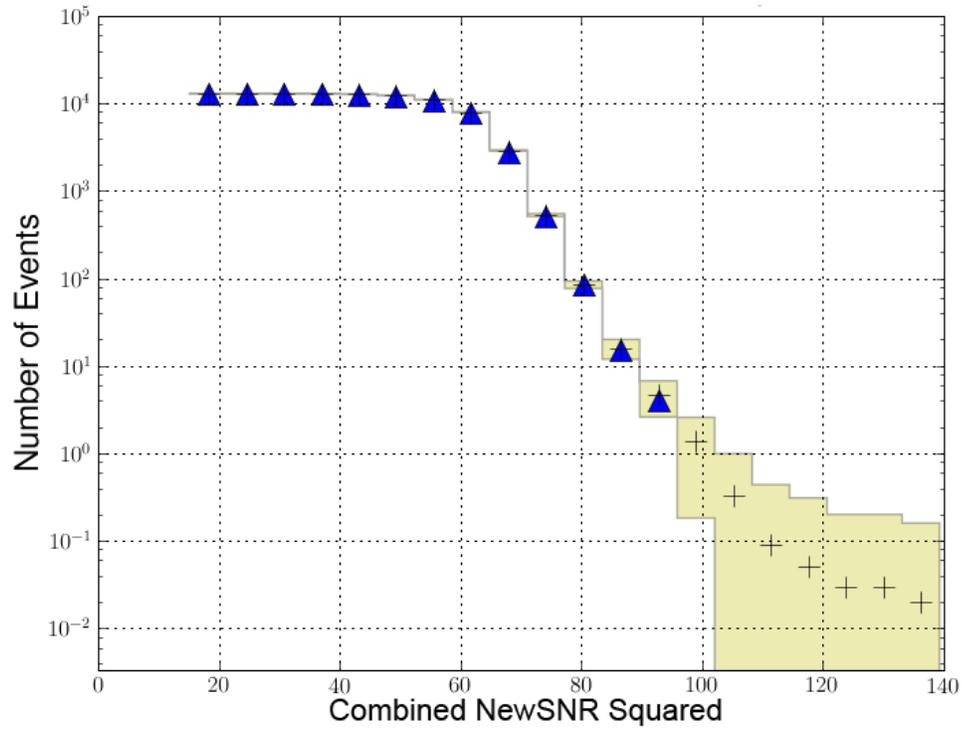
Figure 21 : Sample cumulative histogram created by `ligolw_cbc_plotcumhist`. Data taken from ~ 6 weeks of S6 data, between GPS times 931035296 and 935798487. The data is binned and the cumulative number of triggers with New SNR greater-than or equal-to a given bin are plotted on they y-axis. Zero-lag triggers are indicated by blue triangles. The mean slide counts are indicated by the black crosses and the shaded regions show the standard deviation across the slides. See section 5.6.5 for more details.
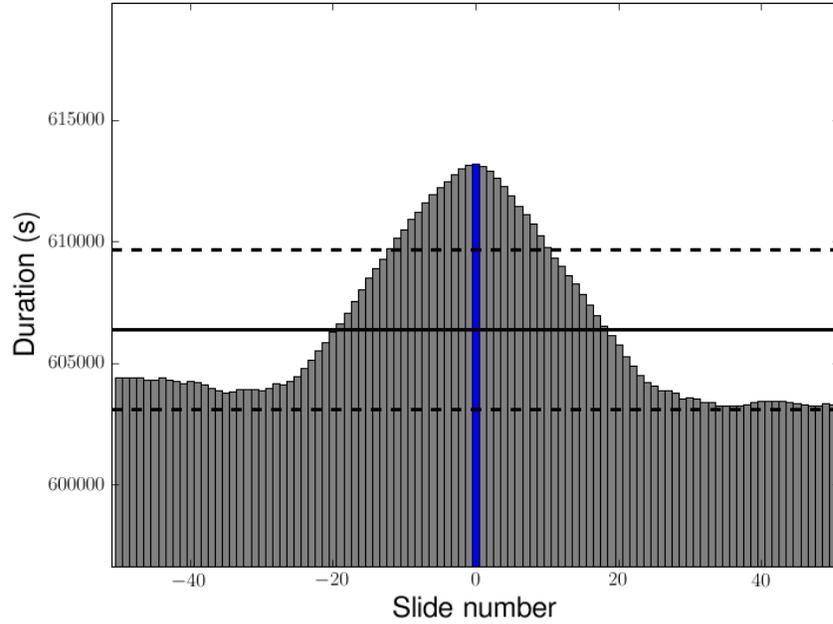
(a) Sample duration-per-slide plot for H1L1V1 time.
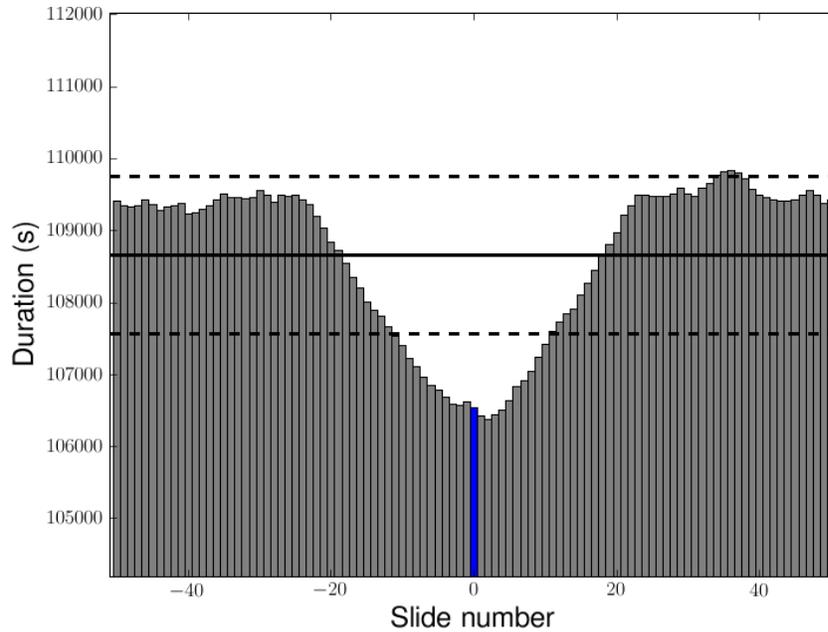
(b) Sample duration-per-slide plot for H1L1 time.

Figure 22 : Sample duration-per-slide plots created by `ligolw_cbc_plotslides`. Both plots taken from ~ 6 weeks of S6 data, between GPS times 931035296 and 935798487. The top plots shows the duration-per-slide for H1L1V1-coincident time; the bottom, H1L1-coincident time. These plots are after CAT3 (cumulative) vetoes have been applied. Gray bars show the duration in each slide; the blue bar shows the zero-lag duration. The black solid line is the mean duration and the dashed lines show the standard deviation. See section 5.6.5 for more details.
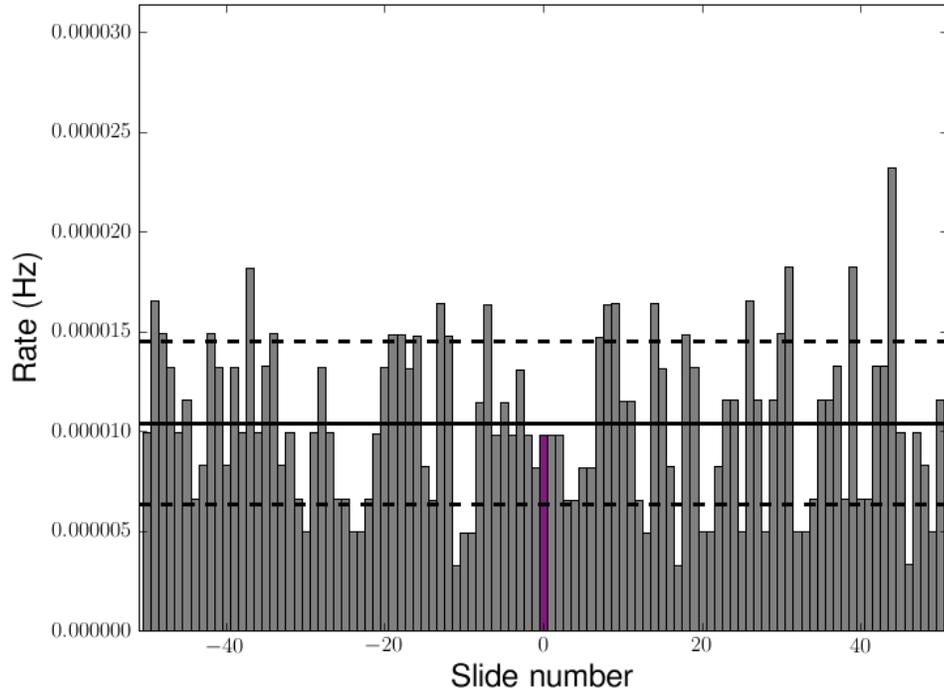
Figure 23 : Sample trigger-rate-per-slide plot created by `ligolw_cbc_plotslides`. The data is from the same time as Figure 5.7. Shown is the rate of H1L1V1-coincident triggers in H1L1V1-coincident time in each slide. Gray bars indicate the rate in each slide; the purple bar shows the rate in zero-lag. (The color of the zero-lag bar is based on the coincident detectors.) The black solid line shows the mean rate and the dashed lines the standard deviation. See section 5.6.5 for more details.
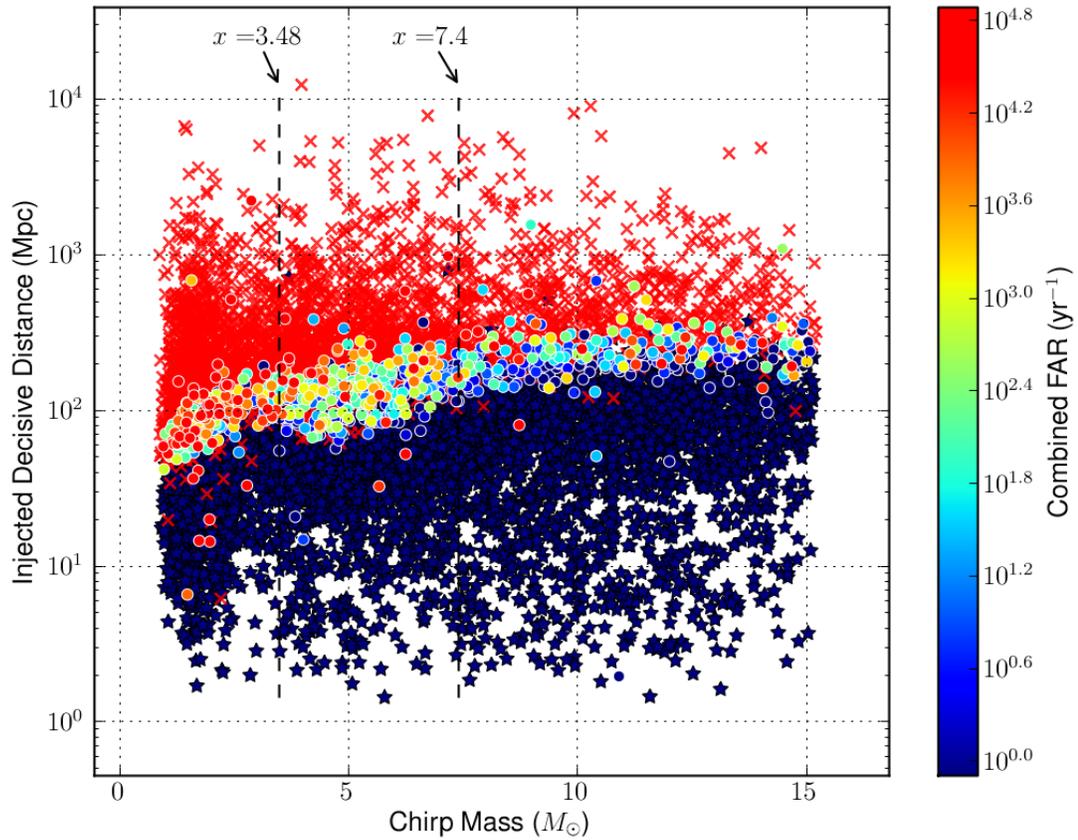
Figure 24 : Sample decisive distance versus chirp mass plot for H1L1 time created by `ligolw_cbc_plotfm`. The data is taken from ~ 6 weeks of S6 data, between GPS times 961545543 and 965174487. The blue stars are injections found with a combined FAR equal to zero. The circles are injections found with non-zero combined FAR; they are colored according to their combined FAR. Missed injections are indicated by red crosses. The black dashed lines show the chirp mass boundaries used. See section 5.6.6 for more details.
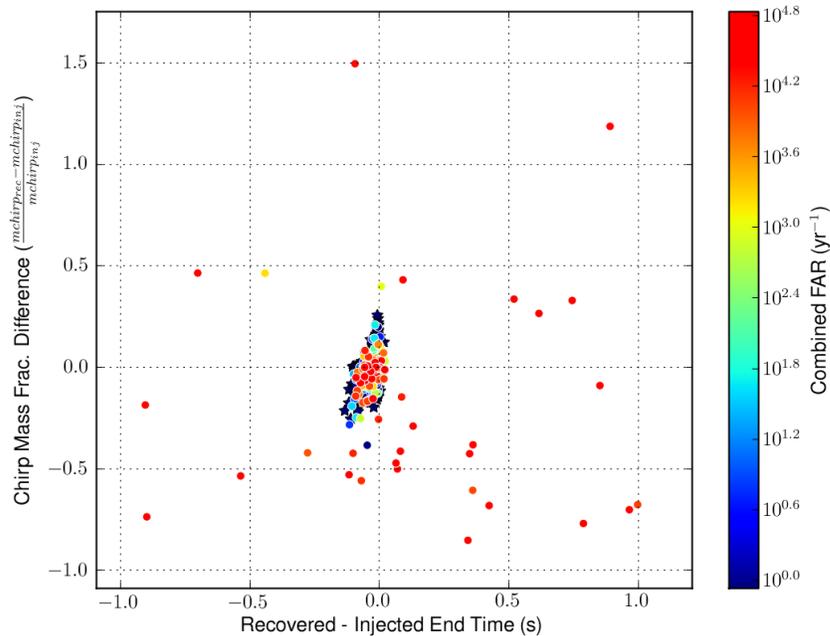
(a) Chirp mass fractional difference versus end-time accuracy.
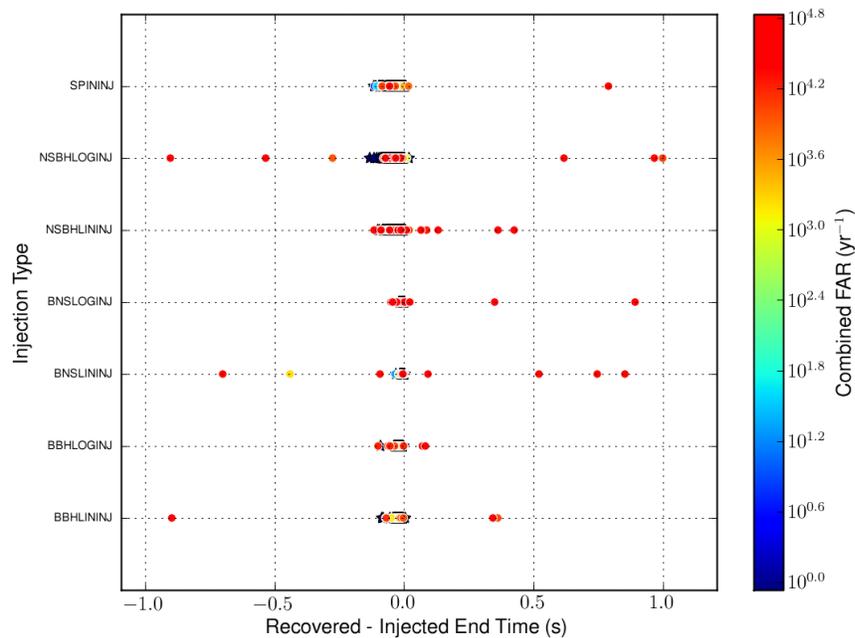
(b) Injection type versus end-time accuracy.

Figure 25 :   More examples of `PlotFM` plots. The data used and color-coding is the same as in Figure 5.7. Here, we have plotted the fractional difference in recovered and injected chirp-mass (top) and the injection type as a function of the difference in injected and recovered end times. Plots such as these help to establish what "found" injections are actually due to noise triggers occurring within the injection-finding time-window, t_{injfind}, of an injection. (See section 5.6.1.) For example, the orange-red dots (representing "found" injections with high combined FAR) that are scattered across the fractional chirp-mass and end-time recovery are most likely due to noise triggers occurring with $\pm t_{\text{injfind}}$ of an injection and not due to the injection itself. Contrast these to the blue stars — injections "found" with 0 combined FAR — which all have good recovered parameters. It is therefore highly likely that these triggers came from injections.

rank (using combined far)	combined far	fap	fap 1yr	snr	end time	end time utc (click for daily ihope)	ifos (click for elog)	mass	mchirp	mini followup	omega scan	all data duration (days)
1	0.0	0.0	0.0	14.36	962784032.29	Sat 10 Jul 2010 08:00:17	H1,L1	22.35	3.37	link	H1 L1	
1	0.0	0.0	0.0	25.07	962784032.37	Sat 10 Jul 2010 08:00:17	H1,L1	15.33	4.01	link	H1 L1	
1	0.0	0.0	0.0	22.29	962841819.07	Sun 11 Jul 2010 00:03:24	H1,L1	3.35	1.46	link	H1 L1	
1	0.0	0.0	0.0	11.95	963216044.39	Thu 15 Jul 2010 08:00:29	H1,L1	17.56	7.62	link	H1 L1	
1	0.0	0.0	0.0	16.42	963360297.47	Sat 17 Jul 2010 00:04:42	H1,L1	3.48	1.48	link	H1 L1	3.96
1	0.0	0.0	0.0	23.28	963446849.94	Sun 18 Jul 2010 00:07:14	H1,L1	3.62	1.56	link	H1 L1	
7	2.78	0.0297	0.94	21.45	963216044.36	Thu 15 Jul 2010 08:00:29	H1,L1	24.14	6.11	link	H1 L1	
8	5.56	0.0585	1.0	16.52	962899495.34	Sun 11 Jul 2010 16:04:40	H1,L1	13.7	5.86	link	H1 L1	
9	214.95	0.9	1.0	8.83	963394154.58	Sat 17 Jul 2010 09:28:59	H1,L1	13.25	3.58	link	H1 L1	
10	301.12	0.96	1.0	8.76	963270289.48	Thu 15 Jul 2010 23:04:34	H1,L1	17.57	3.66	link	H1 L1	

Figure 26 : An example `all_data` loudest-events list created by `PrintLC` when run in Pipedown. This list shows the top ten loudest (i.e., most significant) events, as ranked by combined FAR, in a search. The "end time utc" column provides a link to the daily `ihope` page [80], the "ifos" column provides a link to the e-log page, the "mini followup" column provides a link to the minifollowup page, and the "omega scan" column provides a link to the Omega scans of the event. These links are used to investigate what caused each event. See section 5.6.8 for more details.
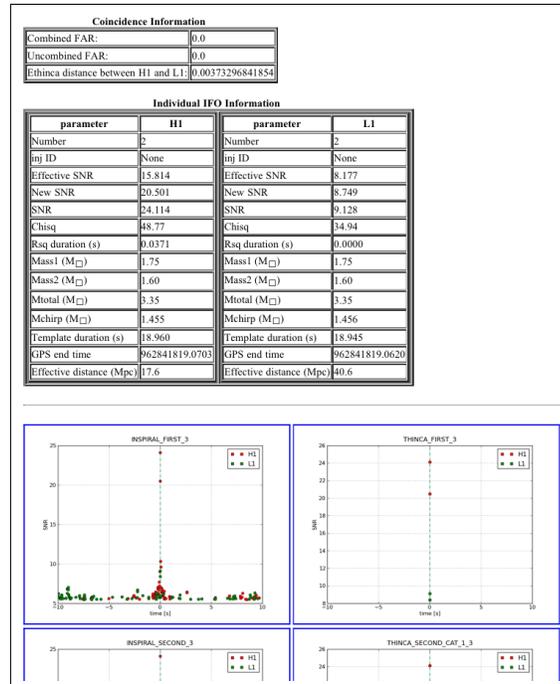
(a) A screen shot of the minifollowup page.
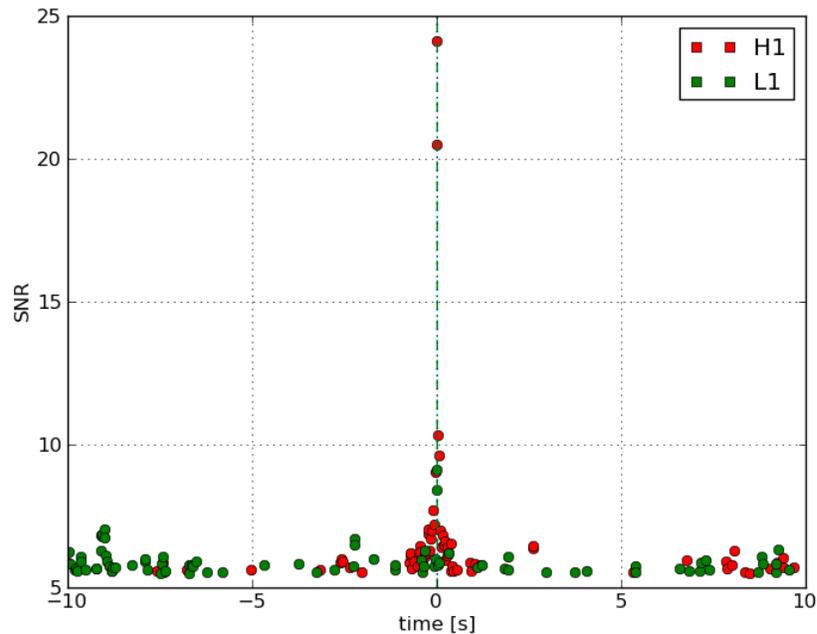
(b) A close-up of the `INSPIRAL_FIRST` minifollowup plot.

Figure 27 : The minifollowup page for the third event, a hardware injection, in the loudest events list shown in Figure 5.7. The top figure shows a screen shot of the page. On this page, there is a table giving more details about the single-detector parameters, as well as plots of SNR versus time within $\pm 10\,$s of the event are shown for each stage in the `HIPE` pipeline (first inspiral — second coincidence). The bottom figure shows `INSPIRAL_FIRST` SNR verus time plot from the page. We see that there is a spike in SNR at the time of the injection in both detectors surrounded by relatively low-level noise. This is what we would expect to see for a GW signal.
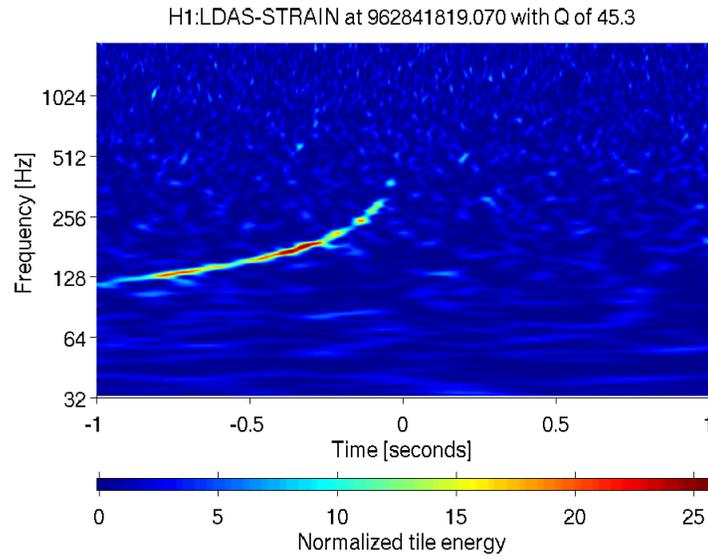
(a) The H1 Omega scan.
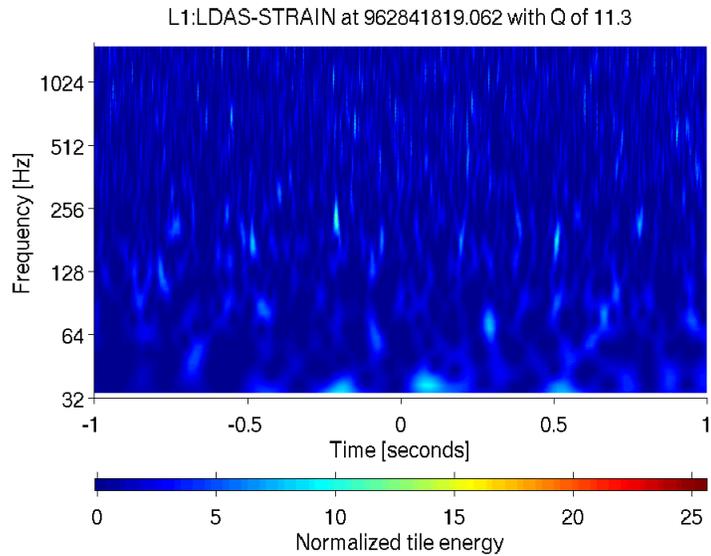
(b) The L1 Omega scan.

Figure 28 : An example of an Omega scan. These scans are of the hardware injection shown in Figure 5.7. The "normalized tile energy" is roughly equivalent to $SNR^2/2$ in a time-frequency tile. For Gaussian noise in the absence of signal, the normalized tile energy rarely exceeds 8. In the H1 scan (top) a chirp pattern is clearly visible (compare this to Figure 3(a)). However, in the L1 scan (bottom) nothing can be seen, since the injection had a SNR $<\sim 10$ in L1. This is why we do not simply look at Omega scans to determine if an event is a GW signal. Scans can provide clues as to the cause of an event. In this case, due to the clear chirp and the lack of noticeable noise in L1 (cf. Figure 5.7), we can be confident this is a GW signal. See section 5.6.8 for more details.

rank (using combined far)	combined far	fap	fap 1yr	snr	mass	mchirp	sngl ifo (click for elog)	sngl end time	sngl end time utc (click for daily ihope)	mini followup	omega scan	slide duration (days)
1	0.0	0.0	0.0	11.05	25.0	10.39	H1	962107614.84	Fri 02 Jul 2010 12:06:39	link	H1 L1	12.77
							L1	962107809.82	Fri 02 Jul 2010 12:09:54			
1	0.0	0.0	0.0	10.09	4.65	1.86	H1	964025786.55	Sat 24 Jul 2010 16:56:11	link	H1 L1	12.77
							L1	964023611.55	Sat 24 Jul 2010 16:19:56			
1	0.0	0.0	0.0	9.73	10.12	3.55	H1	962271729.08	Sun 04 Jul 2010 09:41:54	link	H1 L1	12.85
							L1	962271754.08	Sun 04 Jul 2010 09:42:19			
4	0.87	0.0298	0.58	9.87	4.48	1.64	H1	962222722.94	Sat 03 Jul 2010 20:05:07	link	H1 L1	12.77
							L1	962222882.93	Sat 03 Jul 2010 20:07:47			
5	1.15	0.0395	0.68	9.44	16.25	4.02	H1	964565889.85	Fri 30 Jul 2010 22:57:54	link	H1 L1	12.77
							L1	964566074.88	Fri 30 Jul 2010 23:00:59			
5	1.15	0.0395	0.68	9.48	19.97	3.57	H1	964909555.37	Tue 03 Aug 2010 22:25:40	link	H1 L1	12.77
							L1	964909740.34	Tue 03 Aug 2010 22:28:45			
7	1.73	0.0588	0.82	9.07	20.77	7.84	H1	964214725.75	Mon 26 Jul 2010 21:25:10	link	H1 L1	12.79

Figure 29 : An example `slide` loudest-events list created by `PrintLC` when run in Pipedown. This is similar to the `all-data` loudest-events list shown in Figure 5.7, except that the un-slid single-detector end times are shown. This allows us to easily follow-up loud slide events which — since they determine the FAR of potential signals — are the events that have the greatest impact on the pipeline to detect. We can use this information to provide clues for new vetoes. See Chapter 7 for how these *loudest-slide studies* were carried out in S6.
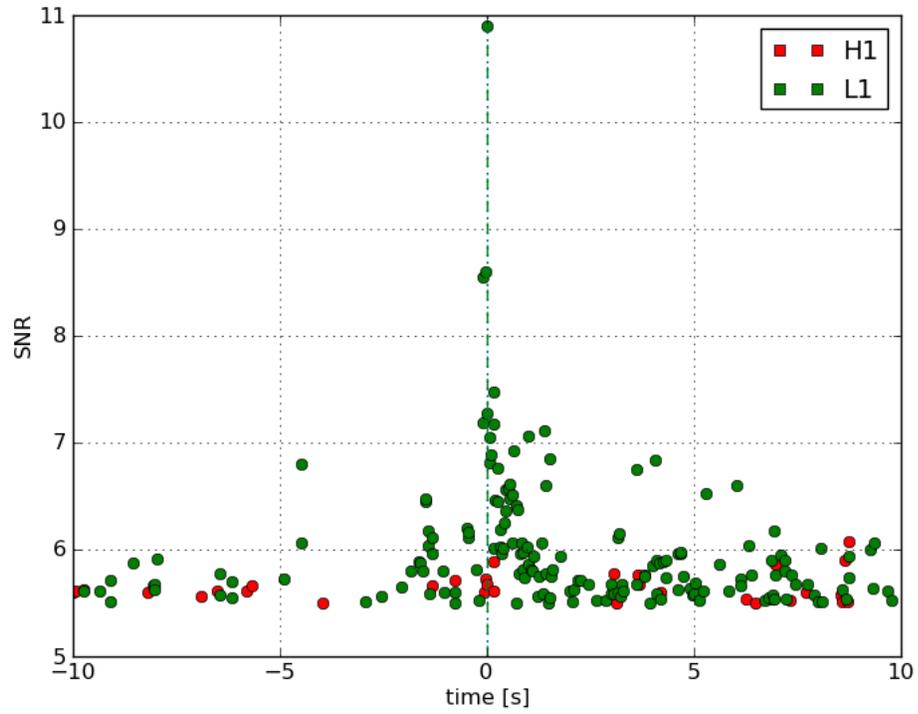
Figure 30 : The INSPIRAL_FIRST minifollowup plot for the first loudest slide event (the one on Friday July 2 2010) shown in Figure 5.7. Contrast this to the minifollowup plot shown in Figure 27(b), which is of a hardware injection. In this case, the SNR of triggers is elevated in L1 around the time of the event, suggesting that an environmental or instrumental source has caused heightened noise in L1 at that time.
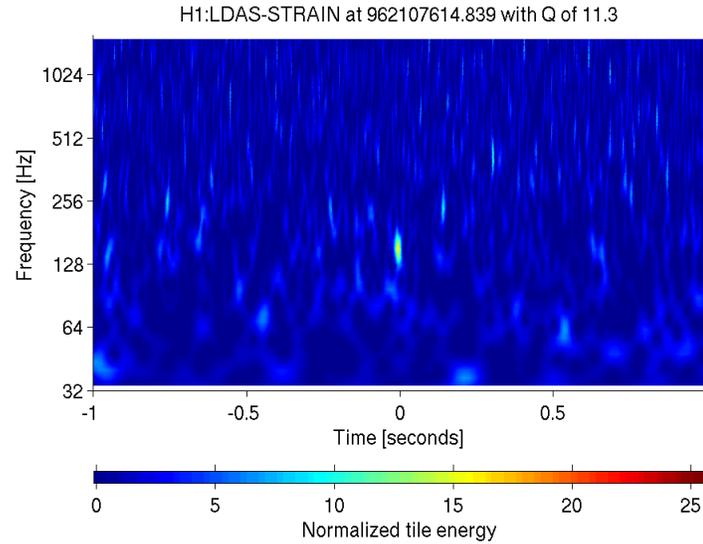
(a) The H1 Omega scan.
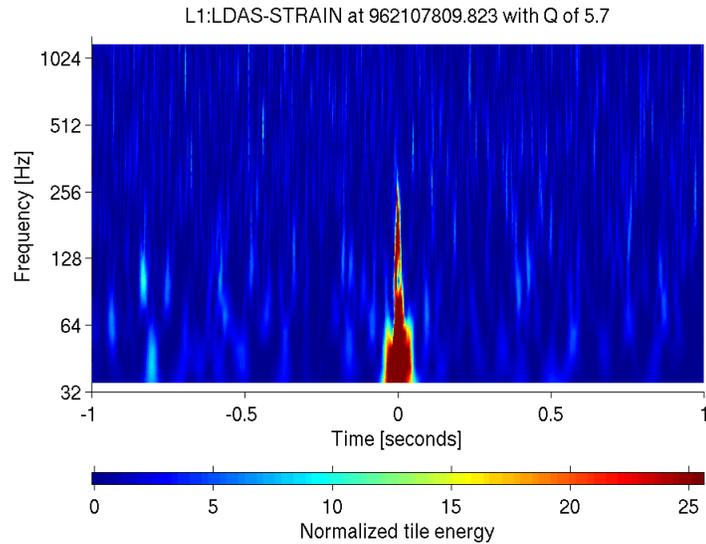
(b) The L1 Omega scan.

Figure 31 : The Omega scans of the slide event shown in Figure 5.7. Compare these scans to those of the hardware injection in Figure 5.7. The H1 scan is relatively clean, and so it could contain a GW signal. The L1 scan, however, clearly shows a low-frequency glitch; this looks nothing like a chirp we would expect to see from a GW. Based on this, we can search the e-log and use other tools (some of which are discussed in Chapter 7) to understand what caused this glitch in L1. If an environmental cause can be found, we can veto this period of time in L1, which would remove this slide coincidence from the background estimation.
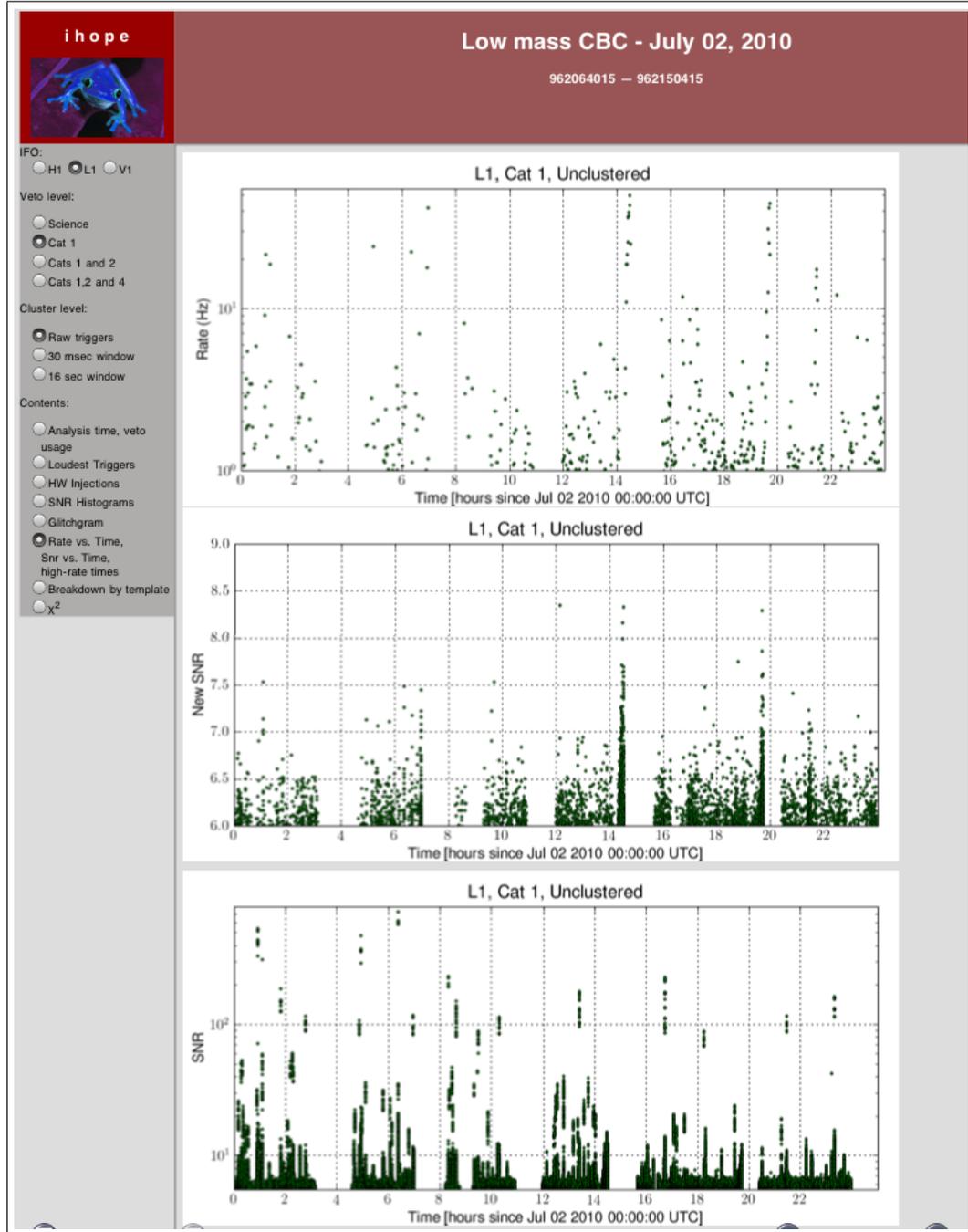
Figure 32 : A screen shot of a daily `ihope` page. This page is from the same day as the loudest slide event shown in Figures 5.7 and 5.7. It can be accessed from the loudest-events table by clicking on the `sngl end time utc` column for the trigger. Shown is the rate, New SNR and SNR of triggers as a function of time for L1 after CAT1 vetoes have been applied. As evident from the side bar, there are several other plots and lists available for all the detectors at various vetoes and clustering windows. For more details on daily `ihope` see [80].

Figure 33 : A screen shot of a L1 e-log. The day shown is from the same day as the loudest slide event shown in Figures 5.7 and 5.7. It can be accessed from the loudest-events table by clicking on the "L1" in the `sngl ifo` column for the trigger. For a further discussion of the e-log page see section 5.6.8.

rank	decisive distance	end time	end time utc (click for daily ihope)	elogs	mchirp	mass1	mass2	eff dist h	eff dist l	eff dist v	sim tag
1	31.06	962109028.31	Fri 02 Jul 2010 12:30:13	H1,L1	1.49	1.66	1.77	28.19	31.06	29.78	BNSLOGINJ
2	32.97	964206193.78	Mon 26 Jul 2010 19:02:58	H1,L1	1.12	1.16	1.43	18.93	32.97	31.67	BNSLOGINJ
3	33.88	963280364.98	Fri 16 Jul 2010 01:52:29	H1,L1	2.56	1.54	6.14	24.65	33.88	70.52	NSBHLININJ
4	34.84	963457773.22	Sun 18 Jul 2010 03:09:18	H1,L1	1.9	1.81	2.65	34.84	23.87	39.12	BNSLOGINJ
5	39.71	961546487.61	Sat 26 Jun 2010 00:14:32	H1,L1	4.28	3.08	8.23	39.71	35.02	17.03	NSBHLININJ
6	43.67	963836899.56	Thu 22 Jul 2010 12:28:04	H1,L1	1.18	1.15	1.6	34.93	43.67	75.05	BNSLOGINJ
7	44.11	965142142.9	Fri 06 Aug 2010 15:02:07	H1,L1	2.18	2.44	2.58	31.46	44.11	30.32	BNSLOGINJ
8	44.56	963459394.58	Sun 18 Jul 2010 03:36:19	H1,L1	1.09	1.21	1.29	37.17	44.56	84.45	BNSLOGINJ
9	45.9	962746123.31	Fri 09 Jul 2010 21:28:28	H1,L1	1.37	1.45	1.7	45.9	34.87	64.59	BNSLININJ
10	46.97	963460414.96	Sun 18 Jul 2010 03:53:19	H1,L1	1.51	1.35	2.25	46.97	40.24	25.87	BNSLOGINJ

(a) All injections.

rank	decisive distance	end time	end time utc (click for daily ihope)	elogs	mini followup	omega scan
1	31.06	962109028.31	Fri 02 Jul 2010 12:30:13	H1,L1	link	H1 L1
2	32.97	964206193.78	Mon 26 Jul 2010 19:02:58	H1,L1	link	H1 L1
3	34.84	963457773.22	Sun 18 Jul 2010 03:09:18	H1,L1	link	H1 L1
4	43.67	963836899.56	Thu 22 Jul 2010 12:28:04	H1,L1	link	H1 L1
5	44.11	965142142.9	Fri 06 Aug 2010 15:02:07	H1,L1	link	H1 L1
6	44.56	963459394.58	Sun 18 Jul 2010 03:36:19	H1,L1	link	H1 L1
7	46.97	963460414.96	Sun 18 Jul 2010 03:53:19	H1,L1	link	H1 L1
8	49.4	962027807.21	Thu 01 Jul 2010 13:56:32	H1,L1	link	H1 L1
9	49.81	964459682.88	Thu 29 Jul 2010 17:27:47	H1,L1	link	H1 L1
10	50.22	964093061.91	Sun 25 Jul 2010 11:37:26	H1,L1	link	H1 L1

(b) The BNSLOGINJ table.

Figure 34 : Example tables produced by PrintMissed when run in Pipedown. The top table shows the 10 closest missed injections out of all of the injection runs. The bottom table shows the 10 closest missed injections in the BNSLOGINJ run. Note that in the second table, minifollowup and Omega scan links have been added. See section 5.6.9 for more information.

Parameters of the loudest (by SNR) recovered single ifo triggers at each stage of the pipeline

step	F/M	SNR	Mchirp	eta	eff_dist	chisq	eff_snr	new_snr	end_time	ethinca	Veto ON/OFF
INSPIRAL_FIRST	FOUND in H1 L1	H1 : 14.805 / L1 : 10.285	H1 : 1.493 / L1 : 1.493	H1 : 0.249 / L1 : 0.249	H1 : 30.240 / L1 : 30.155	H1 : 0.000 / L1 : 0.000	H1 : 0.000 / L1 : 0.000	H1 : 14.805 / L1 : 10.285	H1 : 962109028.314 / L1 : 962109028.305	H1 and L1: 0.001	H1: No info / L1: No info
THINCA_FIRST	FOUND in H1 L1	H1 : 14.805 / L1 : 10.285	H1 : 1.493 / L1 : 1.493	H1 : 0.249 / L1 : 0.249	H1 : 30.240 / L1 : 30.155	H1 : 0.000 / L1 : 0.000	H1 : 0.000 / L1 : 0.000	H1 : 14.805 / L1 : 10.285	H1 : 962109028.314 / L1 : 962109028.305	H1 and L1: 0.001	H1: No info / L1: No info
INSPIRAL_SECOND	FOUND in H1	H1 : 14.805	H1 : 1.493	H1 : 0.249	H1 : 30.240	H1 : 33.245	H1 : 12.328	H1 : 14.401	H1 : 962109028.314		H1: No info
THINCA_SECOND_CAT_1	MISSED										
THINCA_SECOND_CAT_2	MISSED										
THINCA_SECOND_CAT_3	MISSED										
THINCA_SECOND_CAT_4	MISSED										

(a) A screen shot of the found/missed table produced by `MiniFollowups`.
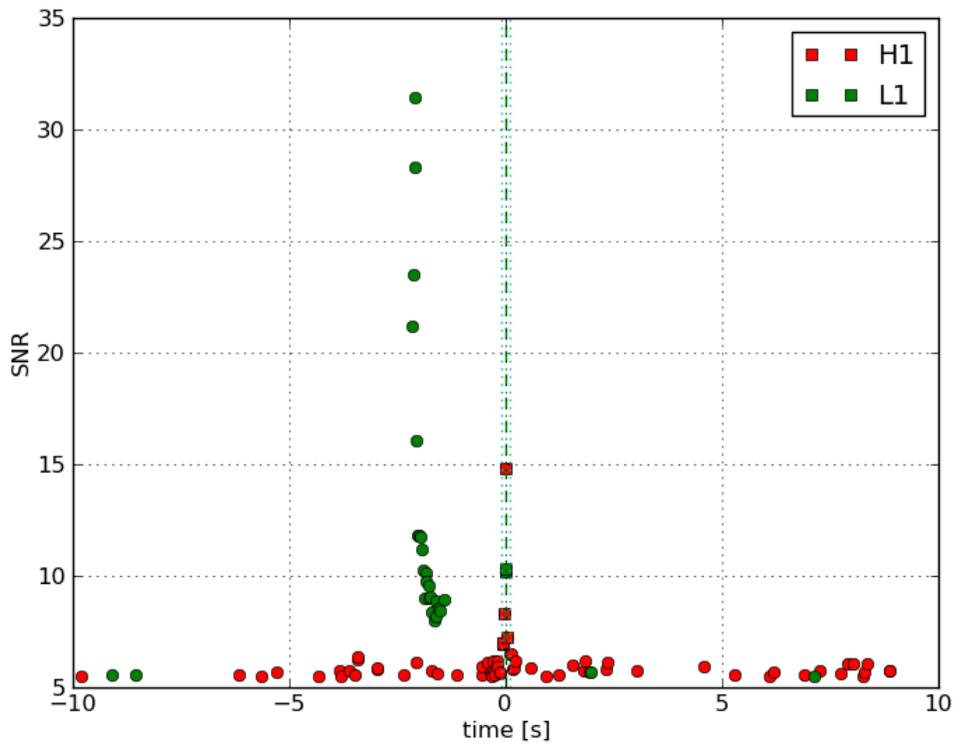
(b) The `INSPIRAL_FIRST` plot from the minifollowup page for this missed injection.

Figure 35 : An example of the output produced by `MiniFollowups` when run on a missed injection. The top figure is a screen shot of the found/missed table that traces whether or not the injection was found, and with what parameters, at each stage in the pipeline. The bottom figure shows the `INSPIRAL_FIRST` plot created for the event. As with the MiniFollowup plots shown in Figure 27(b), this plot shows the SNR versus time of single-detector triggers within ±10 s of the injection. Here, a large spike in SNR is evident in L1 just prior to the injection. This follow-up is for the closest missed injection shown in the tables shown in Figure 5.7.
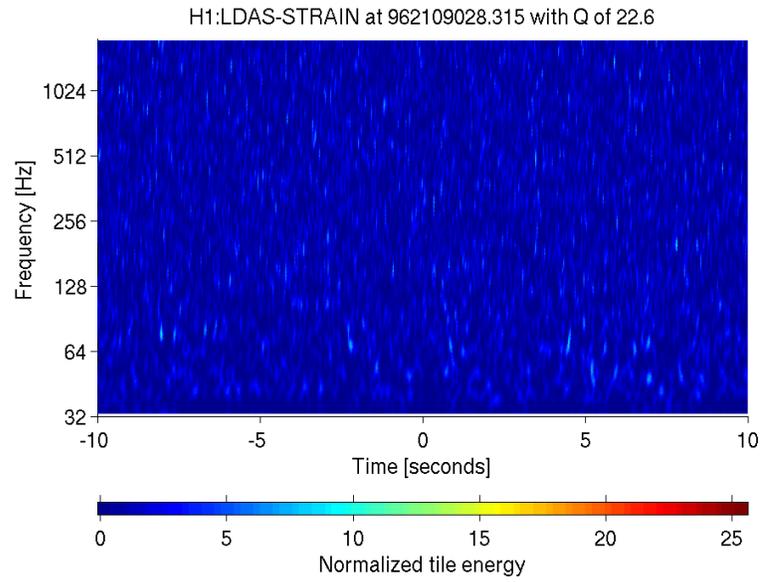
(a) The H1 Omega scan.
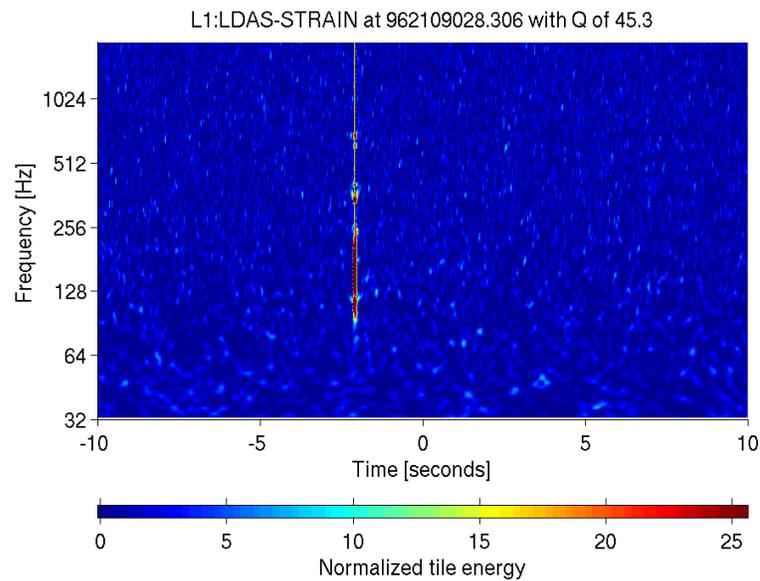
(b) The L1 Omega scan.

Figure 36 : The Omega scans of the missed `BNSLOGINJ` injection detailed in Figure 5.7. The H1 scan is clean, but the L1 scan shows a short-duration broadband glitch $\sim 2\,\mathrm{s}$ before the injection's end time (which is placed at $0\,\mathrm{s}$ on the spectrograms). The presence of this glitch is most likely reason why the injection was missed. Note that the H1 (and possibly the L1) SNR of the injection is large enough to be seen in Omega scan. Nothing is seen because Omega scans do not currently show software injections.

injected decisive distance	injected end time	injected end time utc (click for daily ihope)	elogs	injected mass1	injected mass2	sim tag	recovered combined far	recovered snr
2.98	962379542.09	Mon 05 Jul 2010 15:38:47	H1,L1	1.37	13.0	NSBHLININJ	47063.68	7.64
5.68	962034493.04	Thu 01 Jul 2010 15:47:58	H1,L1	10.37	9.5	SPININJ	0.86	9.48
10.78	964461612.2	Thu 29 Jul 2010 17:59:57	H1,L1	10.5	1.54	SPININJ	7022.47	8.34
11.19	962366920.02	Mon 05 Jul 2010 12:08:25	H1,L1	3.91	15.33	SPININJ	48.83	9.0
17.67	964732781.99	Sun 01 Aug 2010 21:19:26	H1,L1	1.45	1.55	BNSLOGINJ	40442.97	7.76
25.72	964219485.77	Mon 26 Jul 2010 22:44:30	H1,L1	2.28	2.67	BNSLOGINJ	67954.64	6.33
27.48	961560771.64	Sat 26 Jun 2010 04:12:36	H1,L1	1.1	11.45	SPININJ	0.86	9.88
31.61	964479231.58	Thu 29 Jul 2010 22:53:36	H1,L1	1.16	7.02	SPININJ	685.31	8.81
33.27	961545821.3	Sat 26 Jun 2010 00:03:26	H1,L1	2.79	20.09	SPININJ	68100.27	5.98
42.94	964097246.98	Sun 25 Jul 2010 12:47:11	H1,L1	2.59	14.74	NSBHLININJ	67127.69	6.75

(a) All injections.

injected decisive distance	recovered combined far	recovered snr	recovered ifos	injected end time	injected end time utc (click for daily ihope)	elogs	mini followup	injected mass1	injected mass2	injected mchirp	omega scan
17.67	40442.97	7.76	H1,L1	964732781.99	Sun 01 Aug 2010 21:19:26	H1,L1	link	1.45	1.55	1.3	H1 L1
25.72	67954.64	6.33	H1,L1	964219485.77	Mon 26 Jul 2010 22:44:30	H1,L1	link	2.28	2.67	2.15	H1 L1
43.79	0.86	9.92	H1,L1	964824030.28	Mon 02 Aug 2010 22:40:15	H1,L1	link	1.41	1.33	1.19	H1 L1
47.63	0.86	9.92	H1,L1	964953890.87	Wed 04 Aug 2010 10:44:35	H1,L1	link	1.96	1.43	1.45	H1 L1
48.76	5.14	9.71	H1,L1	963643623.93	Tue 20 Jul 2010 06:46:48	H1,L1	link	1.08	1.52	1.11	H1 L1
49.14	5.14	9.74	H1,L1	964041347.47	Sat 24 Jul 2010 21:15:32	H1,L1	link	1.31	2.23	1.48	H1 L1

(b) The BNSLOGINJ table.

Figure 37 : An example of the tables produced by PrintSims when run in Pipedown. The top table shows a portion of the quietest found table for all the injection runs. The bottom table shows a portion of BNSLOGINJ table. Note that in the latter table, MiniFollowup and Omega scan links have been added. See section 5.6.9 for more details.
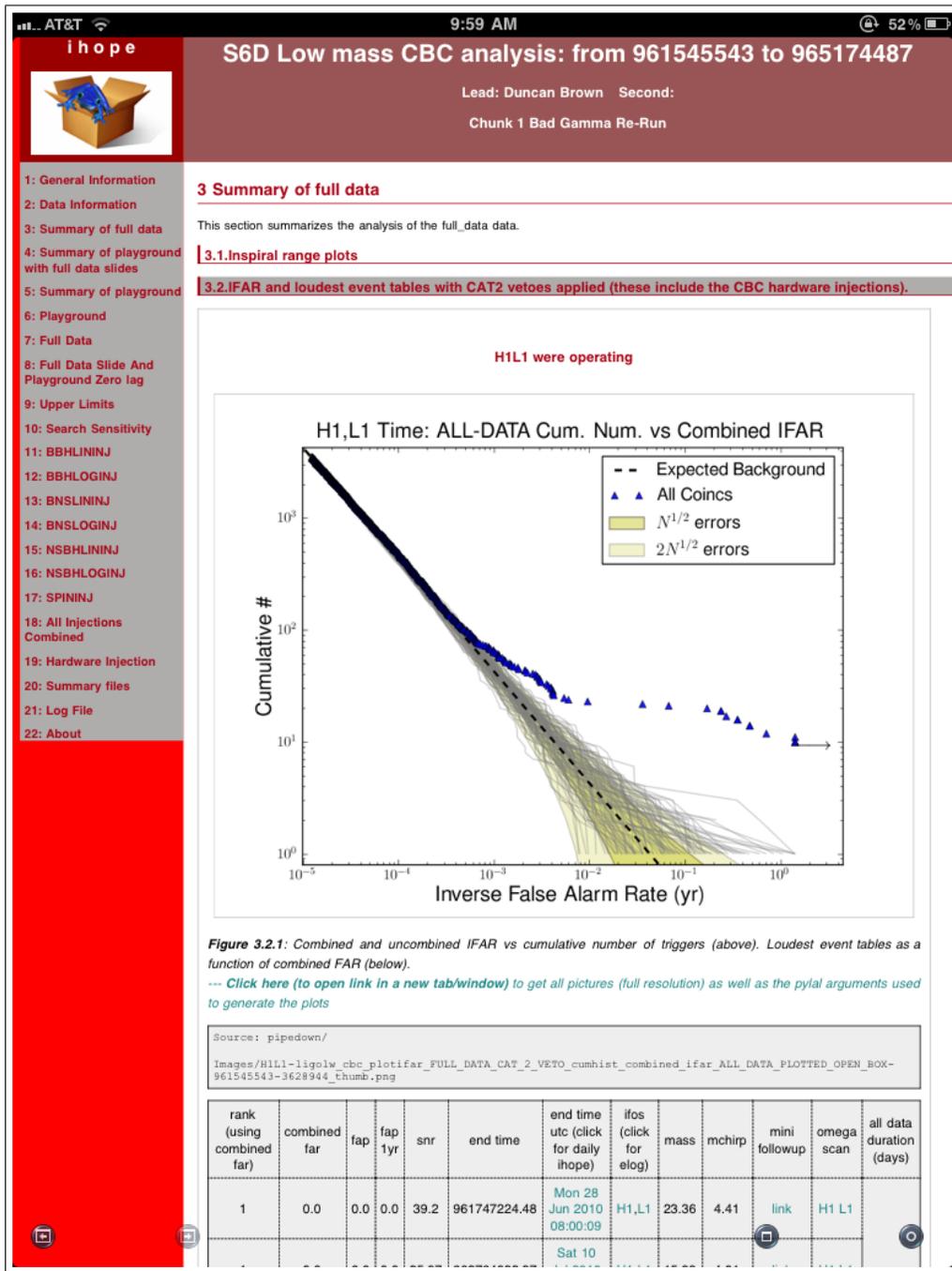
Figure 38 : A screen shot of an `ihope` page. This is the final page that is created after `ihope` has completed. It summarizes all of the data and plots generated by `ihope` and provides links to auxiliary pages to further follow-up triggers. For more details, see section 5.7.

Chapter 6

S5 Results

In November 2005 the three first-generation detectors of the LIGO reached design sensitivity and began a two-year period of observations (known as the fifth science run, or S5) which concluded in October 2007 [82]. As discussed in Chapter 1, one of the most promising sources of gravitational waves for LIGO is a CBC; the inspiral and merger of BNS, BBH, or a NSBH systems [16, 17, 18, 19, 20, 21]. These systems spiral together as they emit energy in the form of gravitational waves, finally merging to form a single object, which then settles down to equilibrium. Ground-based gravitational-wave detectors are most sensitive to waves with frequencies between ~ 40 and 1000 Hz, corresponding to the late stages of inspiral and merger. In this chapter we report the results of a search for gravitational waves from binaries with total mass between 2 and 35 $M_\odot$ and a minimum component mass of 1 $M_\odot$ (the "low-mass" CBC search) in LIGO observations between November 14, 2006 and May 18, 2007 (the "S5 12–18 month" search). The results of a search for these systems in data taken from November 4, 2005 to November 14, 2006 — known as the "S5 first year search" — were reported in [21]. From May–October 2007, the Virgo gravitational-wave detector operated in coincidence with the LIGO detectors [83] and the LIGO data from that period were analyzed together with the Virgo data (known as the "S5-LV search"). The joint analysis required significant modifications to our analysis pipeline. Therefore, results of that search are not reported here; see [6] for more details. In contrast, the results presented here were obtained with substantially the same analysis pipeline used in [21], and described in Chatper 5.

No gravitational-wave signals were observed during this search and so we report

upper limits on CBC rates using the upper limits of [21] as prior rate distributions.

6.1   The Data Analysis Pipeline

The data-analysis pipeline used in this search is fundamentally the same as that of [21] and discussed in Chapter 5. Here, we only describe the major components and highlight differences to the previous, "S5 first year" search, referring to [20, 21] for details. The most substantial change in this analysis was a modification to the way in which the significance of candidate events are compared to instrumental noise background. In previous searches, the noise background was computed *using the entire observation period* by introducing an artificial time shift between data recorded at the two LIGO observatories. In this search, the observation period was split into six four-week segments and one 18 day segment (referred to as "months") and the instrumental background was measured *independently in each month*, as the detector behavior varied over the course of the S5 run. Candidate triggers were therefore compared to a background that better reflected the instrumental behavior at the time of the candidate. Each month was searched independently for gravitational-wave candidates. In the absence of a detection, the results from the months were combined (together with the results from [21]) to set an upper limit on the CBC rate.

We searched for gravitational-wave signals when at least two of the LIGO detectors were operational. This comprised a total of 0.28 yr when all three detectors (the 4 and 2 km Hanford detectors, denoted H1 and H2, respectively, and the 4 km Livingston detector, denoted L1) were operational (H1H2L1 coincident data), 0.10 yr of H1H2 coincident data, 0.02 yr of H1L1 coincident data, and 0.01 yr of H2L1 coincident data. Noise correlations between the co-located H1 and H2 detectors cause our method of estimating the instrumental background using time-shifted data to fail, and so we did not search data when only the H1H2 detectors were operating. Approximately 10% of the data was designated *playground* and used for tuning our search pipeline.[1]

Post-Newtonian (pN) theory provides accurate models of the inspiral waveform predicted by General Relativity up to ISCO [23, 24, 25, 26, 27, 28, 29, 30]. The frequency of the waveform from low mass binaries targeted in this search sweeps

---

[1]See section 5.1 for a further discussion of "playground" data.

across the sensitive band of the LIGO detectors. Therefore, we searched for signals from our target sources by match filtering the data with pN templates terminated at ISCO. This method is suboptimal if a true signal differs from our template family due to unforeseen physical effects. Matter effects in BNS and NSBH are not included in our templates, but are expected to be important only at higher frequencies [84, 85]. We constructed template banks [58] of restricted second order pN waveforms in the frequency domain [86, 87, 24] such that no more than 3% of the SNR was lost due to the discreteness of the bank [55]. A "trigger" was generated if the matched-filter SNR of the strain data filtered against the template exceeded a threshold of 5.5 [12]. We demanded that triggers are coincident in time of arrival and mass [34] in at least two of the three LIGO detectors. When all three detector were operating we could obtain (in principle) four possible types of coincidence: H1H2L1 triple coincident triggers and three different double coincident types: H1H2, H1L1 and H2L1. We discarded H1H2 double coincident triggers, due to the problems estimating the background for these triggers, and we discarded H2L1 triggers when the H1 detector was operating nominally (since the 4 km H1 detector was more sensitive than the 2 km H2 detector).

Coincident triggers were subjected to consistency checks using signal-based vetoes [88, 32, 33]. Times of poor detector data quality were flagged using environmental and auxiliary data; triggers from these times were also vetoed [21]. We constructed three categories of data-quality vetoes depending on the severity of the instrumental artifact being flagged ("CAT1", "CAT2", and "CAT3" vetoes; see section 5.1 for a more details). In our primary search, and in our upper limit computation, we vetoed coincident triggers that fall in times from either category 1, 2, or 3. We also considered detection candidates in data with only category 2 vetoes applied, in case a loud signal was present that may otherwise have been vetoed. Surviving triggers were clustered in time and ranked by the effective SNR statistic, which was computed from the trigger's matched-filter SNR and the value of the χ^2 signal-based veto for that trigger [20]; see equation 3.68.[2] After discarding playground data and times in both veto categories, a total of 0.21 yr of triple coincident data (H1H2L1) ,0.02 yr of H1L1 coincident data, and 0.01 yr of H2L1 coincident data remained. In the absence of a detection, these data were used to compute upper limits on CBC rates.

---

[2]We used effective SNR as opposed to New SNR because New SNR had not been developed at the time of the search.
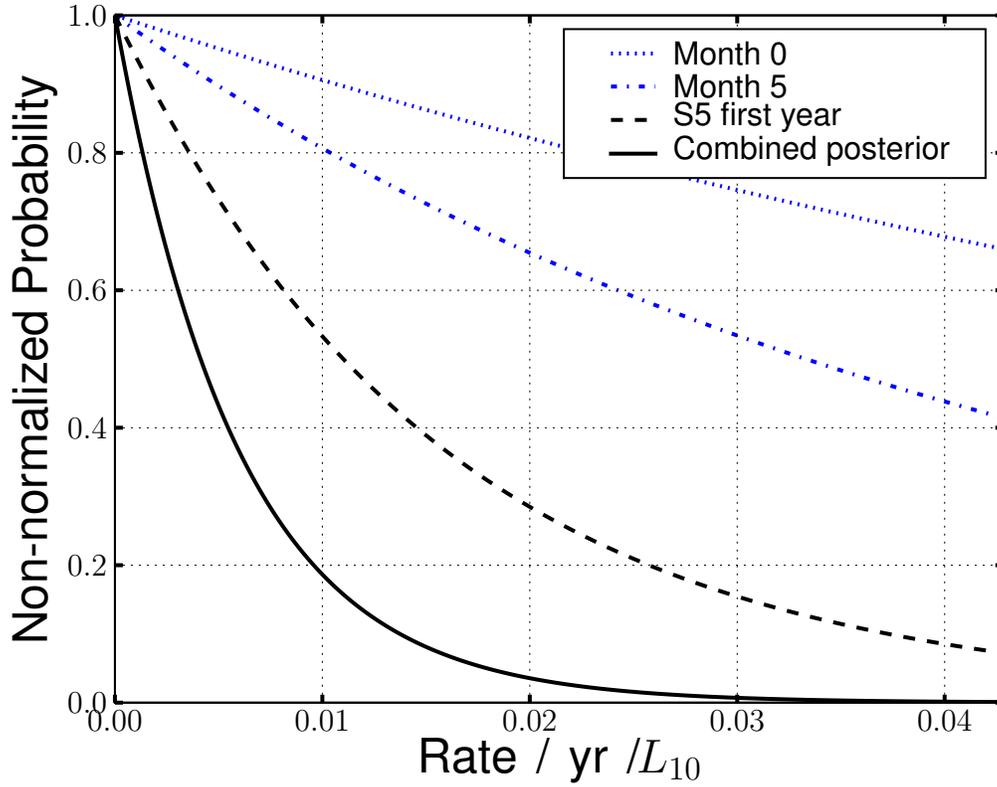
Figure 39 : The posterior distribution for the rate of BNS coalescences. The dashed black curve shows the rate computed in [21]. The solid black curve shows the result of this search using the previous analysis as a prior. The figure also shows the rate distributions for two of the individual months computed using a uniform prior. The improvement from month 0 to month 5 is due to increasing detector sensitivity during this search.

The rate of instrumental noise artifacts was measured by time-shifting data from the Livingston and Hanford observatories (H1 and H2 data are kept fixed with respect to each other). The data was offset by more than the light-travel time between observatories, thus triggers which survived the pipeline were due to noise alone. We performed 100 such time-shifts to obtain a good estimate of the noise background in our search. CBC signals of higher-mass contain fewer gravitational-wave cycles in the sensitive band of our detectors: our signal-based vetoes were not as powerful. High-mass templates are therefore more sensitive to non-stationary noise transients and hence our FAR for these system is larger. In order to account for this mass-dependent behavior we computed the background for three different mass regions and compared foreground and background within each of these ranges. Specifically, in each region we counted the number of background triggers with effective SNR greater-than or equal-to a given foreground trigger; dividing this number by the amount of background time analyzed gave us the FAR for that trigger. This allowed us to define a single detection statistic for every trigger in each of the mass categories. The FAR could then be directly compared to obtain a ranking of the significance of the triggers, regardless of their mass [21].

6.2   Search results

The seven months of data were analyzed separately using the procedure described above. No gravitational-wave candidates were observed with a FAR significantly above those expected from the noise background. The loudest trigger in this search was a triple coincident trigger with a FAR of 6 per year. This is consistent with the expected background, since we searched 0.21 yr of data. The second and third loudest triggers had FAR values of 10 and 11 per year, respectively. Although we did not have any detection candidates, we exercised our follow-up procedures by examining any triggers with a FAR of less than 50 per year. This exercise prepares us for future detections and often identifies areas where our search pipeline can be improved to exclude noise transients.

In the absence of detection candidates, we used our observations to set an upper limit on CBC rates. We followed the procedure described in [46, 47, 48] and used the results reported in [21] as prior information on the rates. We present five different

classes of upper limits. The first three limits are placed on binaries of neutron stars and/or black holes assuming canonical mass distributions for BNS [$m_1 = m_2 = (1.35 \pm 0.04)$ M$_\odot$], BBH [$m_1 = m_2 = (5 \pm 1)$ M$_\odot$], and NSBH [$m_1 = (5 \pm 1)$ M$_\odot$, $m_2 = (1.35 \pm 0.04)$ M$_\odot$] systems. We also present upper limits as a function of the total mass of the binary and, for NSBH binaries, as a function of the black hole mass. We combined the results from each of the seven months, along with the prior results from the first year analysis, in a Bayesian manner, using the same procedure as described in [21].

We first calculated upper limits on BNS, BBH and NSBH systems assuming the objects have no spin; these results are summarized in Tables 18 and 19. The rate of binary coalescences in a galaxy is expected to be proportional to the blue light luminosity of the galaxy [89]. Therefore, we placed limits on the rate per L$_{10}$ per year, where L$_{10}$ is 10^{10} times the blue solar luminosity (the Milky Way contains ~ 1.7L$_{10}$ [90]). To calculate the search sensitivity, the analysis was repeated numerous times adding simulated signals with a range of masses, distance and other astrophysical parameters to the data. Table 19 shows the sensitivity of the LIGO detectors to coalescing binaries quoted in terms of the horizon distance, i.e., the distance at which an optimally oriented and located binary would produce an SNR of 8.

There are a number of uncertainties which affected the upper limit calculation, including Monte Carlo statistics, detector calibration, distances and luminosities of galaxies listed in the galaxy catalog [89], and differences between the pN templates used to evaluate efficiency of the search and the actual waveforms. The effect of these errors on the cumulative luminosity are summarized for the BNS search in Table 18. We marginalized over all of the uncertainties [46] to obtain a posterior distribution on the rate of binary coalescences.

In Fig. 39, we show the derived distribution of the rate of BNS coalescences. The distribution is peaked at zero rate because there were no detection candidates. We include the distribution for all searches previous to this one (which was our prior). In addition, we present the result that would be obtained from each month, were it analyzed independently of the others and of the previous searches. This provides an illustration of the amount that each month contributes to the final upper

Coincidence time	H1H2L1	H1L1	H2L1
Observation time (yr)	0.21	0.02	0.01
Cumulative luminosity (L_{10})	490	410	110
Calibration error	23%	23%	26%
Monte Carlo error	3%	7%	10%
Waveform error	31%	32%	31%
Galaxy distance error	16%	16%	3%
Galaxy magnitude error	19%	19%	17%

Table 18 : Detailed results from the BNS search. The observation time is the time used in the upper limit analysis. The cumulative luminosity is the luminosity to which the search was sensitive above the loudest event for each coincidence time. The errors in this table are listed as one-sigma logarithmic error bars (expressed as percentages) in luminosity associated with each source error.

limit result, and demonstrates the improvement in sensitivity of the detectors during the search. The upper limit is finally obtained by integrating the distribution from zero to $\mathcal{R}_{90\%}$ so that 90% of the probability is contained in the interval. The results obtained in this way were: $\mathcal{R}_{90\%,\mathrm{BNS}} = 1.4 \times 10^{-2}\,\mathrm{yr}^{-1}\mathrm{L}_{10}^{-1}$, $\mathcal{R}_{90\%,\mathrm{BBH}} = 7.3 \times 10^{-4}\,\mathrm{yr}^{-1}\mathrm{L}_{10}^{-1}$, and $\mathcal{R}_{90\%,\mathrm{NSBH}} = 3.6 \times 10^{-3}\,\mathrm{yr}^{-1}\mathrm{L}_{10}^{-1}$.

Additionally, we calculated the upper limit for BBH systems as a function of the total mass of the binary, assuming a uniform distribution of the component masses. For NSBH systems, we constructed an upper limit as a function of the black hole mass, assuming a fixed neutron star mass of $m_{\mathrm{NS}} = 1.35\mathrm{M}_{\odot}$. These upper limits are shown in Fig 40.

Finally, we present upper limits on coalescence rates where the spin of the components of the binary is taken into account. Astrophysical observations of neutron stars indicate that their spins will not be large enough to have a significant effect on the BNS waveform observed in the LIGO band [59, 60]. Theoretical considerations limit the magnitude of the spin, S, of a black hole to lie within the range $0 \leq S \leq Gm^2/c$. However, the astrophysical distribution of black hole spins, and spin orientations, is not well constrained. Therefore, we provide a sample upper limit for spinning systems using a spin magnitude and orientation distributed uniformly within the allowed values. This gives upper limits on the rate of BBH and NSBH

Component masses ($M_\odot$)	1.35/1.35	5.0/5.0	5.0/1.35
D_{horizon} (Mpc)	~ 30	~ 100	~ 60
Cumulative lminosity (L_{10})	490	11000	2100
Nonspinning upper limit $\left(\mathrm{yr}^{-1}L_{10}^{-1}\right)$	1.4×10^{-2}	7.3×10^{-4}	3.6×10^{-3}
Spinning upper limit $\left(\mathrm{yr}^{-1}L_{10}^{-1}\right)$	...	9.0×10^{-4}	4.4×10^{-3}

Table 19 : Overview of results from BNS, BBH and NSBH searches. D_{horizon} is the horizon distance averaged over the time of the search. The cumulative luminosity is the luminosity to which the search was sensitive above the loudest event for times when all three LIGO detectors were operational. The first set of upper limits are those obtained for binaries with non-spinning components. The second set of upper limits are produced using black holes with a spin uniformly distributed between zero and the maximal value of Gm^2/c.

systems of: $\mathcal{R}_{90\%,\mathrm{BBH}} = 9.0 \times 10^{-4}\ \mathrm{yr}^{-1}\mathrm{L}_{10}^{-1}$ and $\mathcal{R}_{90\%,\mathrm{NSBH}} = 4.4 \times 10^{-3}\ \mathrm{yr}^{-1}\mathrm{L}_{10}^{-1}$. These rates are about 20% larger than the non-spinning rates.

Discussion We searched for gravitational waves from CBCs with total mass between 2 and 35 $M_\odot$ in LIGO observations between November 14, 2006 and May 18, 2007. No detection candidates with significance above that expected due to background were found in the search. By combining this search with our previous results, we set a new upper limit on the CBC rate in the local universe which is approximately a factor of 3 lower than that reported in [21]. This improvement was significant, even though we searched only two thirds as much data as in [21]. It was due, in part, to improvements in detector sensitivity during the second year of S5 which increased the horizon distance. Moreover, the shorter analysis time and improved stationarity of the data led to many of the months having a less-significant loudest event than in the previous search. Both of these effects increased the luminosity to which the search was sensitive, thereby improving the upper limit.

Astrophysical estimates for CBC rates depend on a number of assumptions and unknown model parameters, and are still uncertain at present. In the simplest models,

the coalescence rates should be proportional to the stellar birth rate in nearby spiral galaxies, which can be estimated from their blue-light luminosity [89]. The optimistic, upper end of the plausible rate range for BNS is 5×10^{-4} $\mathrm{yr}^{-1}\mathrm{L}_{10}^{-1}$ [91, 92] and 6×10^{-5} $\mathrm{yr}^{-1}\mathrm{L}_{10}^{-1}$ for BBH and NSBH [93, 94]. The upper limits reported here are ~ 1–2 orders of magnitude above the optimistic expected rates. The most confident BNS rate predictions are based on extrapolations from observed binary pulsars in our Galaxy; these yield realistic BNS rates of 5×10^{-5} $\mathrm{yr}^{-1}\mathrm{L}_{10}^{-1}$ [91, 92]. Rate estimates for BBH and NSBH are less well constrained, but realistic estimates are 2×10^{-6} $\mathrm{yr}^{-1}\mathrm{L}_{10}^{-1}$ for NSBH [93] and 4×10^{-7} $\mathrm{yr}^{-1}\mathrm{L}_{10}^{-1}$ for BBH [94]. Thus, the expected rates are ~ 2–3 orders of magnitude lower than the limits presented in this chapter. The Advanced LIGO and Virgo detectors, currently under construction, will increase our horizon distance by an order of magnitude or more, allowing us to measure the rate of CBCs in the Universe.
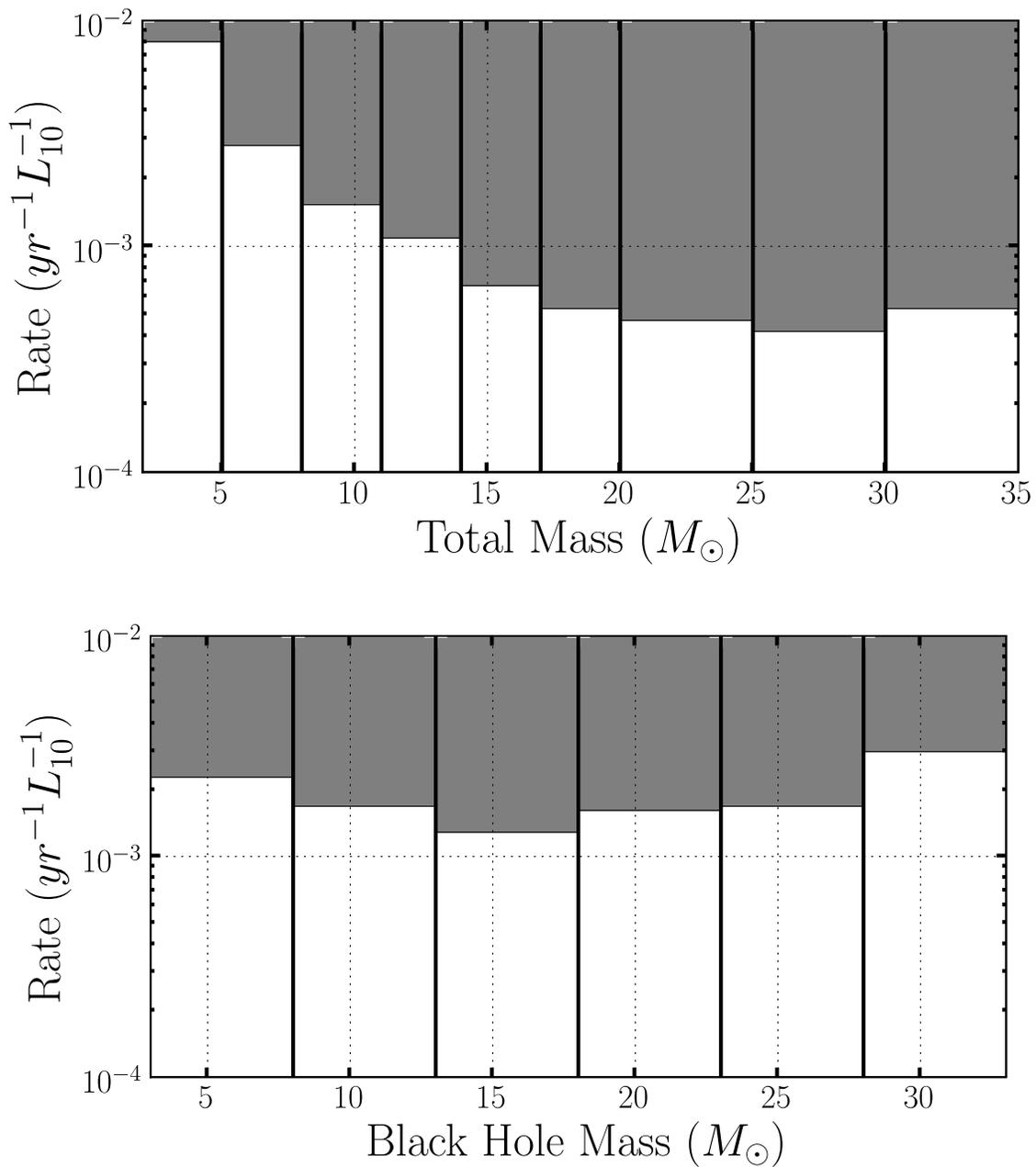
Figure 40 : The marginalized 90% rate upper limits as a function of mass. The upper plot shows limits for BBH systems as a function of the total mass of the system. The lower plot shows limits for NSBH systems as a function of the black hole mass, assuming a fixed neutron star mass of $1.35 M_\odot$. Here the upper limits were calculated using only H1H2L1 data since the relatively small amount of H1L1 and H2L1 data made it difficult to evaluate the cumulative luminosity in the individual mass bins.

Chapter 7

S6 Tuning and Results

The sixth LIGO Science (S6) run began on 7 July 2009 and ended on 20 October 2010. This observing run overlapped with two separate Virgo Science runs: Virgo's second science run (VSR2), which ran from 7 July 2009 to 11 January 2010, and Virgo's third science run (VSR3), which ran from 11 August 2010 to 20 October 2010. A number of improvements were made in both detector hardware and CBC analysis software between S5 and S6. The software improvements have already been described in prior chapters: New SNR was developed to replace effective SNR, and Pipedown was implemented.

In this chapter we describe the S6 and VSR2/3 analysis. Section 7.1 describes the hardware improvements made to the detectors between S5 and S6. In section 7.2 we describe each of the four epochs the analysis was broken into. Section 7.3 details some of the major data quality (DQ) issues that arised during S6/VSR2/3 and how they were dealt with, along with tuning decisions made. In this section we give an example of a veto developed from the loudest-slide studies that were implemented in the second half of S6. Finally, in section 7.5 we give the results of the search. No gravitational waves from CBCs were detected. We describe a *blind injection* that was made, and found, during S6/VSR3. Upper limits on the rate of CBCs will be presented in a forthcoming LSC and Virgo publication [61].

7.1 Hardware Improvements

The two 4 km LIGO interferometers, H1 and L1, were used for S6. The 2 km Hanford detector, H2 was not operational during this run. Several hardware changes were made to the LIGO detectors so that prototypes of Advanced LIGO technology could be installed and tested. This included the installation of a more powerful, 35 W laser, and the implementation of a DC readout system that included a new Output Mode Cleaner on an Advanced LIGO seismic isolation table [3]. In addition, the hydraulic seismic isolation system was improved by fine-tuning its feed-forward path. Known as "HEPI feed-forward," this improvement was implemented in January of 2010; it is described in more detail in section 7.2.2, below.

Several hardware enhancements were also made to the Virgo detector in the period between VSR1 and VSR2. A more powerful laser was installed, along with a thermal compensation system and scattered light was better mitigated. During early 2010, monolithic suspension was installed, which involved replacing Virgo's test masses with new mirrors hung from fused-silica fibers. Following this upgrade Virgo began VSR3.

The average sensitivity of the detectors to binary coalescence signals in each epoch is shown in Figures 7.2 – 7.2. These figures show the distance at which an optimally oriented and located binary would produce a SNR of 8 in a given detector. The figures show how the detectors were improved over the course of the run, and they eventually surpassed the best S5 ranges.

7.2 S6 Epochs

S6 and VSR2/3 were broken into four epochs: *S6A*, which ran from 7 July 2009 to 1 September 2009; *S6B*, 24 September 2009 to 11 January 2010; *S6C*, 6 February 2010 to 25 June 2010; *S6D*, 26 June 2010 to 20 October 2010. Table 20 lists the analyzed time (live time) and duty cycle in each epoch after CAT3 vetoes[1] have been applied. Across all of S6 we analyzed 0.48 years of data, giving a duty cycle of 0.41.

Figure 41 shows a plot of the BNS inspiral range (with each component mass = $1.4 \, M_\odot$) across all of S6 and the span of each epoch. The start and end times of the epochs were based on a combination of instrumental and analysis factors. S6A

---

[1]See section 5.1 for definition of vetoes and veto categories.

ended at a pre-planned commissioning break to try to improve the detectors after learning lessons from the first two months of running. S6B ran from the end of the commissioning break until the end of VSR2. At this point, Virgo was taken off line for eight months in order to install the monolithic suspension. S6C therefore consisted only of coincident time between Hanford and Livingston. Another commissioning break was taken at the end of S6B, hence the gap between the end of S6B and the start of S6C. S6D was to begin when Virgo came back online in August of 2010. During S6C we noticed that non-Gaussian noise transients (*glitches*) often created triggers with SNRs above threshold when match filtered with templates that had total masses $> 25 \, \mathrm{M_\odot}$. Thus we decided to lower the mass-range of our template bank[2] from $2 \leq \mathrm{M_{total}}/\mathrm{M_\odot} \leq 35$ to $2 \leq \mathrm{M_{total}}/\mathrm{M_\odot} \leq 25$. We wanted to do this as soon as possible, and so the somewhat arbitrary date of 26 June 2010 was chosen as the break between S6C and S6D. Aside from Virgo coming back online, there were no major instrumental adjustments between S6C and D. There were, however, new vetoes implemented for S6D based on CBC results in S6C. These new vetoes, as well as more details about the decision to decrease the range of the template bank, are discussed in section 7.3. In the next few sections we give more details about each of the epochs.

Epoch	Live Time by Instrument Time (days)				Total Live Time (days)	Duty Cycle
	H1L1	H1V1	L1V1	H1L1V1		
S6A	1.2	10.9	10.0	7.1	29.2	0.53
S6B	7.1	20.6	9.4	10.6	47.7	0.44
S6C	39.3	–	–	–	39.3	0.28
S6D	22.6	7.7	8.3	19.7	58.3	0.50
Total	70.2	39.1	27.7	37.4	**174.5**	**0.41**

Table 20 : The analyzed time (*live time*) in each epoch, and the total for S6/VSR2/3. All times are calculated after CAT3 vetoes have been applied.

---

[2]Recall from section 3.2 that a template bank is the collection of waveforms we use to match filter the data.
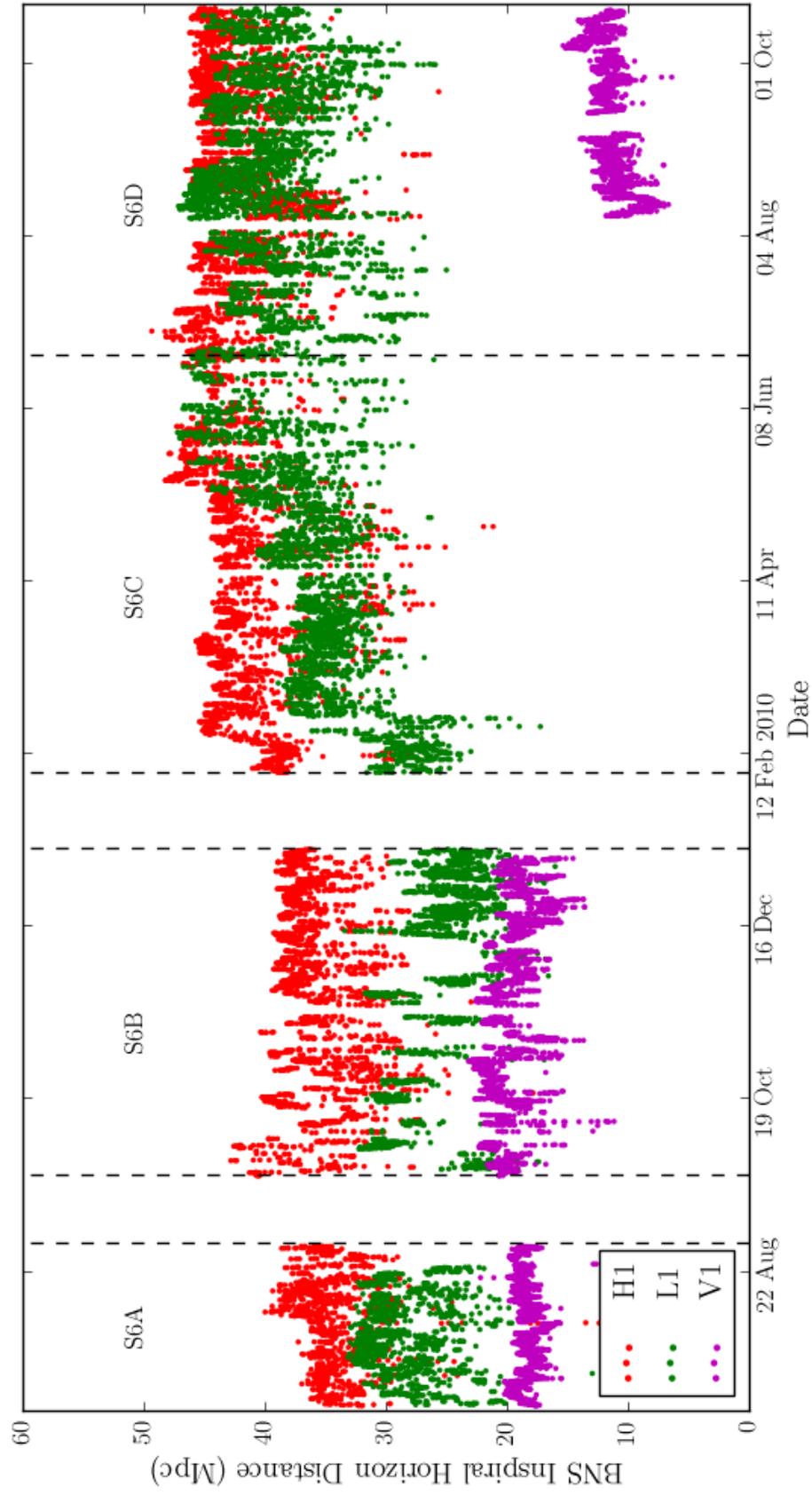
Figure 41 : The inspiral range for a $1.4/1.4\,M_\odot$ BNS system at SNR 8 in each IFO across S6. The range is computed by `lalapps_tmpltbank` using equation 3.36. Each dot represents the range in a 2048 s–long analysis chunk.

7.2.1 S6A

At the start of the S6 Science run, the LIGO detectors had lower sensitivity and a higher rate of non-Gaussian transient noise (glitches) than in later epochs. This can be seen in Figure 41. In fact, the S6A LIGO ranges were somewhat lower than they had been during their peak sensitivity in S5. Figure 7.2 shows the average inspiral range versus binary total mass during S6A as compared to best ranges of S5. We can see that Virgo, however, showed much improvement over VSR1.

In the joint LIGO and Virgo search that occurred during VSR1 and the last five months of S5 (the "S5-LV" search), false alarm rates (FARs) were based on a likelihood statistic that took into account the relative sensitivities of various interferometer combinations. The sensitivities were used to apply a weighting factor to each coincidence type so that more sensitive detector combinations were promoted, allowing false alarm rates to be computed once across all combinations (as opposed to using equal-weighted bins to compute combined FARs from uncombined, as described in Chapter 4) [6]. This was implemented for two main reasons: first, with both H2 and V1 active, four interferometers had to be analyzed, which led to a large number of coincidence types and coincident-detector times to consider. Second, Virgo's sensitivity was much lower than the LIGO detectors in VSR1, and, due to high amounts of low-frequency noise, its low-frequency cutoff had to be set to $60\,\mathrm{Hz}$. Thus its template bank was truncated to have a maximum chirp mass of $\sim 2.6\,\mathrm{M_\odot}$ [6]. The probability that various coincidence types detected a GW was therefore far from equal, and so re-weighting of triggers' SNRs needed to be applied. As can be seen in Figure 7.2, however, the range of V1 was substantially better in VSR2 — better than H2 — and we were able to lower its low-frequency cutoff so that it could cover the same mass range as LIGO. Essentially, V1 in S6 had taken the place of H2 in S5. Further, since V1 was not co-located with the other detectors, we could slide all the instruments against each other, which allowed us to analyze all instrument times. (Recall from the last chapter that H1 and H2 could not be slid against each other because they were co-located, and so H1H2-coincident time could not be analyzed.) For these reasons, and the fact that the S5-LV likelihood method was still being developed when S6A started, we decided to analyze S6 in the same manner as in the S5 search described in
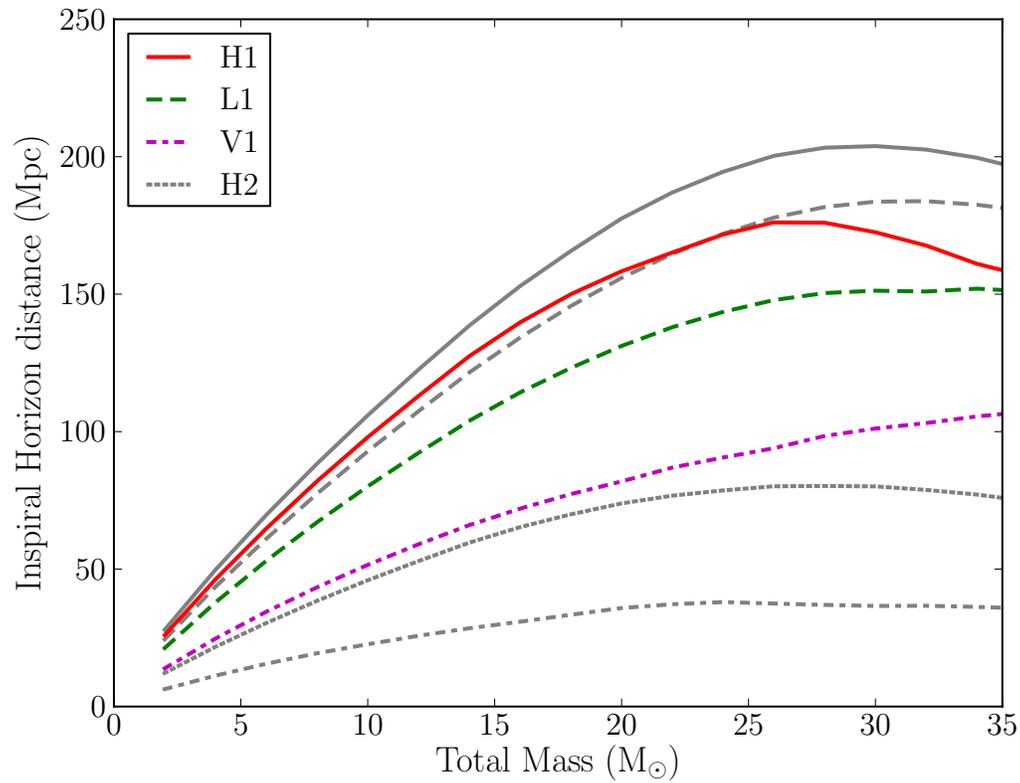
Figure 42 : Average inspiral range of S6A. Ranges were computed by `lalapps_tmpltbank` using equation 3.36 with $\rho = 8$, then averaged over all analysis chunks in the epoch. S6 ranges are in color; best S5 ranges are in gray. Although H2 was not used in S6, it is shown for comparison to V1.
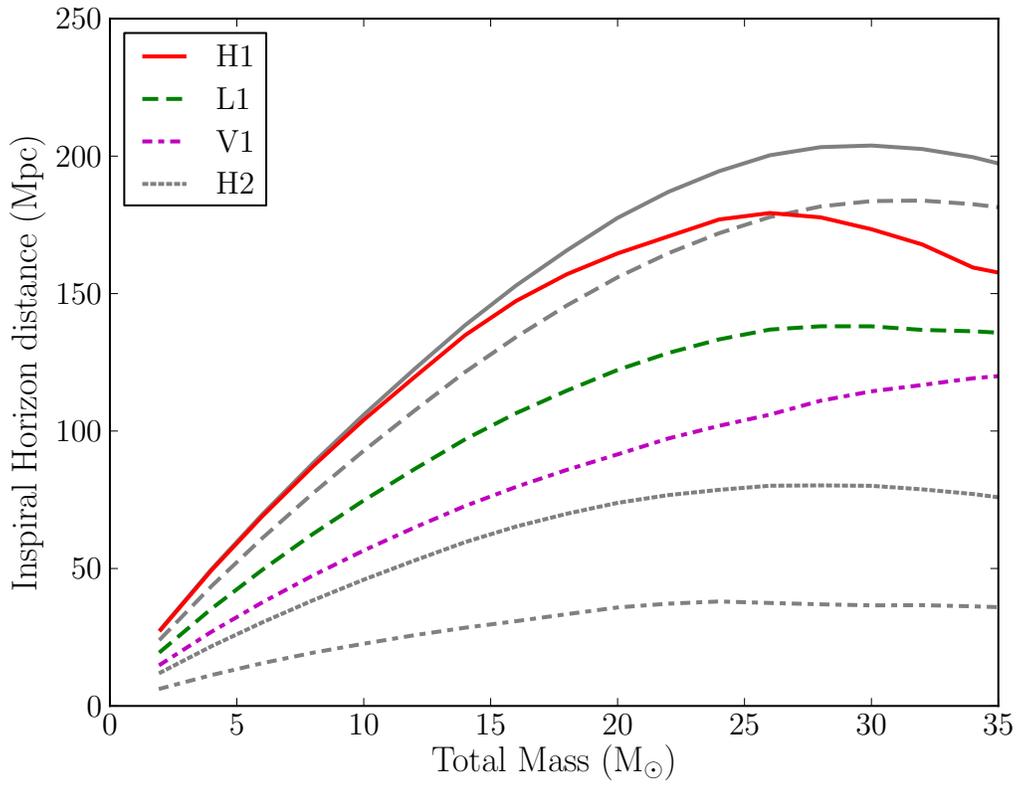
Figure 43 : Average inspiral range of S6B. Ranges were computed using the same method as in Figure 7.2.
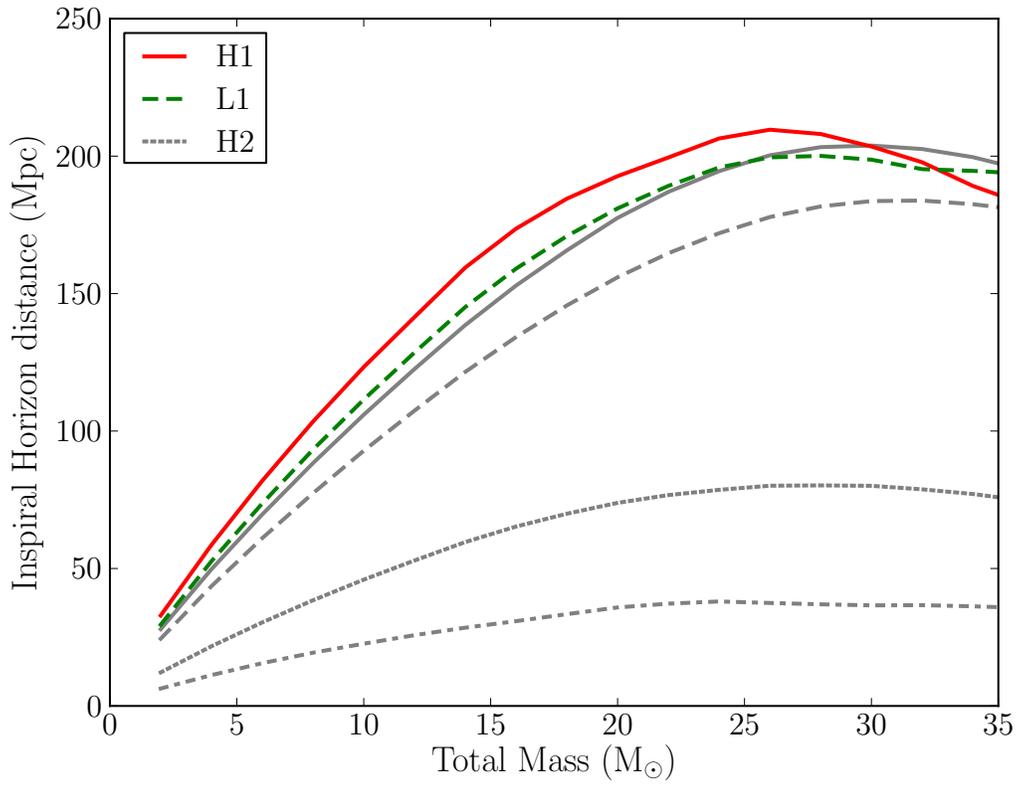
Figure 44 : Average inspiral range of S6C. V1 is not shown as it was down for commissioning during this period. Ranges were computed using the same method as in Figure 7.2.
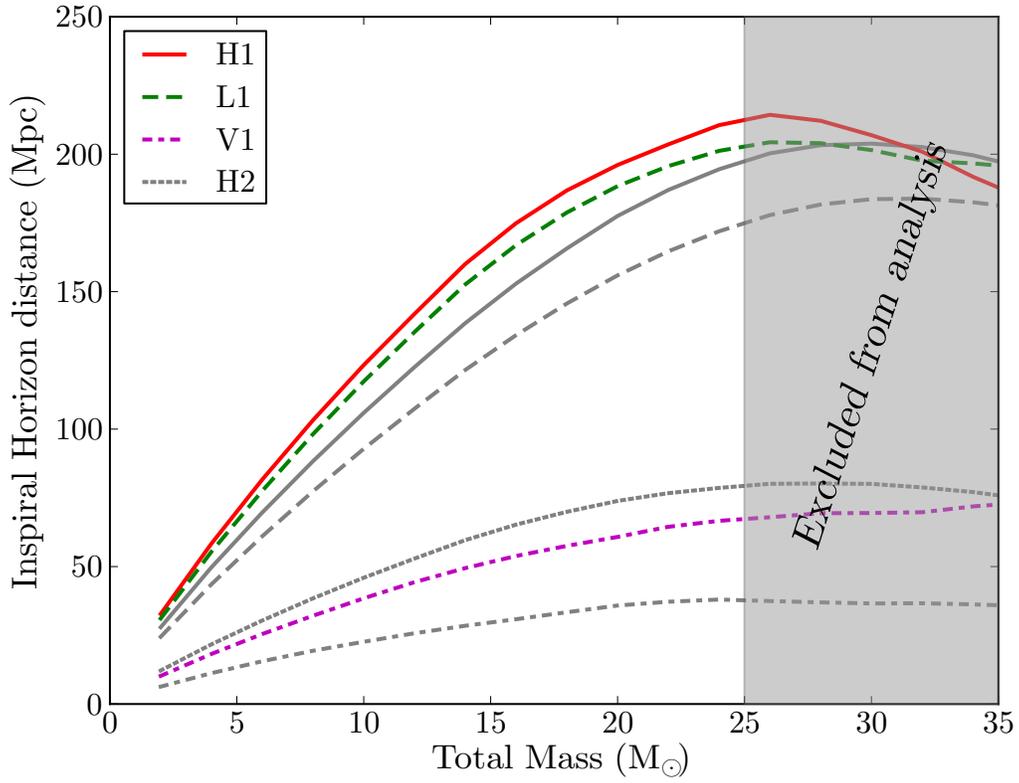
Figure 45 : Average inspiral range of S6D. Shaded region indicates the mass range that was excluded in the low mass search for this period. Ranges were computed using the same method as in Figure 7.2.

Chapter 6 (the "12-18 month search"[3]), using combined FAR as our ranking statistic, and with all the coincidence types being given equal weight. We also used many of the same tuning parameters as the S5 12-18 month search: the SNR cut, χ^2 and r^2 veto thresholds, chirp-mass bins, and size of the e-thinca parameter[4] all remained the same.

There were a few minor adjustments, however. As mentioned in Chapter 3, we switched to using 3.5 restricted pN templates, although the template bank metric was still calculated using the 2pN approximation. New SNR was implemented as our ranking statistic for computing uncombined FARs after it was noticed — in the 12-18 month search and the S6A CBC high-mass search[5] — that effective SNR tended to over-weight triggers with statistically low χ^2 values. Pipedown replaced older, more cumbersome, scripts to do post-`HIPE` processing. As discussed in Chapter 5, we decided to do coincidence clustering within each chirp-mass bin as opposed to across all bins, as done in the 12-18 month search. We also switched algorithms for computing combined FARs from the method discussed in section 4.5 of Chapter 4 to using slide triggers' uncombined FARs, discussed in section 4.4. This change had little effect on the analysis, as they are equivalent. In the 12-18 month search, `ihope` was run on month-long blocks of data as opposed to the year-long block used in the S5 first-year search. As discussed in Chapter 6, the duration of `ihope` analysis periods were decreased to better reflect changing detector behavior. We continued this trend of decreasing the analysis periods in S6: in S6A we decided to run `ihope` in week-long periods, with a different analyst being in charge of each analysis period. Thus, S6A was (initially) broken into 8 week-long runs.

Initially we also planned to use TrigScan clustering[6] in `lalapps_inspiral`[7] to cluster triggers across templates. However, we were surprised to find that trigger rates were much higher than in S5. TrigScan clustering was unable to keep the rate low-enough for many `inspiral` jobs to finish (recall from Chapter 5 that a disadvantage of TrigScan is that it cannot guarantee a maximum trigger rate). Many

---

[3]The name 12-18 month comes from the fact that that search covered months 12 to 18 of S5.

[4]See chapters 3 and 4 for definitions of these parameters.

[5]Recall from Chapter 1 that the CBC high-mass search covers the mass ranges $25 \leq \mathrm{M_{total}}/\mathrm{M_\odot} \leq 100$

[6]See section 5.3.6 in Chapter 5 for a description of TrigScan clustering.

[7]Recall from Chapter 5 that `lalapps_inspiral` is the program we use to perform match filtering.

jobs took several days to complete, or would simply run out of memory. As a result, only two weeks out of the eight were able to finish with TrigScan. Figure 7.2.1 shows a comparison of the average trigger rate-per-template between one of the weeks that finished ("S6aWk3") and one of the months from the S5-LV analysis ("lvMonth8") at each stage of the pipeline. The rate was clearly higher for both H1 and L1 at all stages in the pipeline. In particular, the rate at second inspiral (2 on the x-axis) was not much lower than first inspiral (0 on the x-axis), implying that χ^2 was being calculated for a large number of triggers.[8] As this comparison had to use one of the weeks that finished, the other weeks that did not finish most likely had an even higher rate as compared to S5.

To mitigate the high rates we decided to switch from TrigScan to time-window clustering, discussed in section 5.3.6. What size window to use was an open question and so we used the two weeks that completed to investigate various clustering windows. We aimed to find the smallest window that allowed the search to continue, and that resulted in the most efficient recovery of vetoes. Three windows were considered: $10\,\mathrm{ms}$, $30\,\mathrm{ms}$, and $100\,\mathrm{ms}$. The $10\,\mathrm{ms}$ window was found to be too short: the trigger rates were still too high for many runs to complete. This left the $30\,\mathrm{ms}$ and $100\,\mathrm{ms}$ windows. Figure 7.2.1 shows a ROC plot[9] using $30\,\mathrm{ms}$ and $100\,\mathrm{ms}$ clustering. Note that in this plot we also considered raising the low-frequency cutoff to $65\,\mathrm{Hz}$. As can be seen, the $30\,\mathrm{ms}$ window with the standard $40\,\mathrm{Hz}$ low-frequency cutoff provided the best results. We therefore decided to use the $30\,\mathrm{ms}$ clustering for all of S6. Why TrigScan clustering did not work as well is an open question. It may have simply been due to the high trigger rate. Another possibility is the switch from 2pN to 3.5pN templates somehow affected its ability to properly cluster [80]. We plan to re-address this before Advanced LIGO.

7.2.2 S6B

S6B began after the commissioning break in September of 2009. While the sensitivity in H1 improved slightly, the sensitivity in L1 was lower than in S6A. Further, due to inclement weather, L1 struggled to stay in Science mode for most of the period, and so the duty cycle was much lower. As stated above, S6B ran from September to

---

[8]Refer to Chapter 5 for a description of each of these steps in the `ihope` pipeline.

[9]Refer to section 5.6.7 for a description of ROC plots.
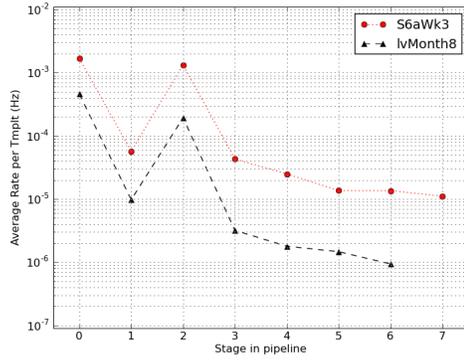
(a) H1
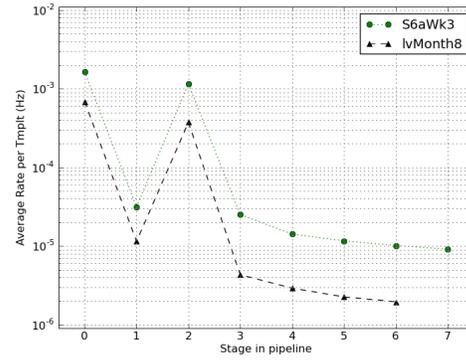
(b) L1
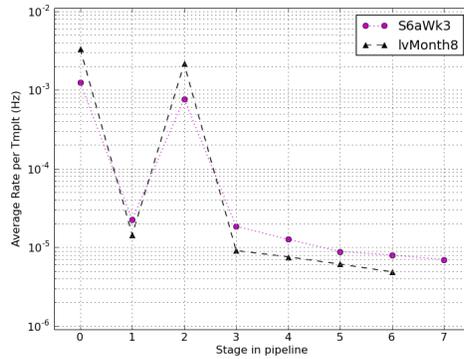
(c) V1

Figure 46 : The average trigger rate per template in week 3 of S6A as compared to one month from the S5-LV analysis. Each point represents a different stage in the pipeline: $0 \rightarrow$ INSPIRAL FIRST, $1 \rightarrow$ THINCA FIRST, $2 \rightarrow$ INSPIRAL SECOND, and $3 \rightarrow$ THINCA SECOND. (See Chapter 5 for a description of each of these stages.) Stages 4–7 represent the higher-category vetoes being applied at THINCA SECOND; there is an extra stage in S6A because hardware injections were removed by an extra veto-category (between steps 5 and 6), whereas in the LV search hardware injections were removed as a part of the CAT2 vetoes (at step 4).
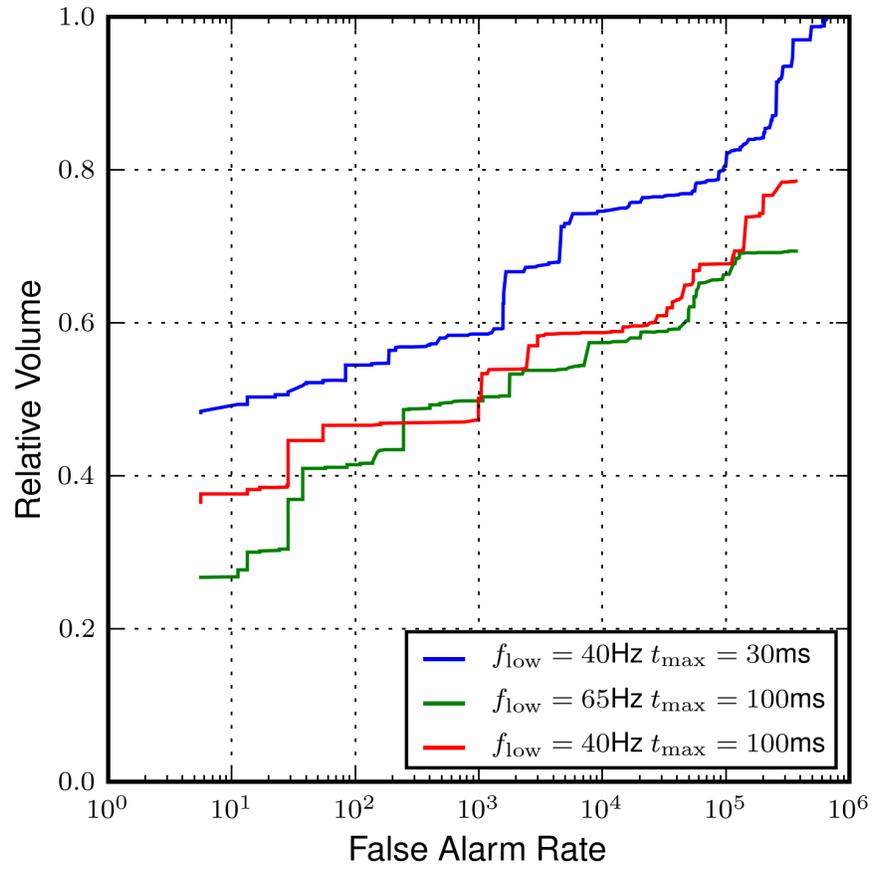
Figure 47 : ROC plot from week 3 of S6A using various clustering windows and low-frequency cutoffs.

January. Fall brings a number of storms to the Gulf of Mexico, resulting in increased *microseismic* (0.1–0.35 Hz) noise. The effects of this noise can be seen in Figure 41: note that L1 is off for large periods of time, and that it has lower sensitivity when it is in Science mode.

To mitigate the seismic problem at Livingston Hydraulic External Pre-Isolation (HEPI) feed-forward was developed during S6B. Many of the interferometer's optics sit on tables housed in vacuum chambers. These chambers are situated on hydraulic actuators, which are used to actively dampen seismic noise. In the feed-forward process, the output from seismometers on the floor were fed into the actuators to cancel out the seismic motion [95]. The digital filters used in the feed-forward were implemented and tuned across S6B. This led to L1 becoming more stable, allowing for longer Science mode periods. The effect can be seen in Figure 41: a few days prior to 16 December L1 begins to stay in Science mode for longer periods of time. The increased stability eventually also allowed the laser power to be increased to $14\,\mathrm{W}$[10] which in turn led to better range.[11]

Due to the sensitivity of the LIGO detectors and L1's low duty cycle, we decided to analyze S6B in three analysis periods. The first lasted from the start of S6B until 11 November 2009, at which point the detectors were re-calibrated. The second and third periods ran from 11 November to 11 December, and 11 December to 11 January, respectively. We chose 11 December to end the second period so that each period was approximately the same duration.

7.2.3 S6C

After V1 went offline in January of 2010, another commissioning break was taken to improve the sensitivity of the LIGO detectors. Thus the S6C analysis began approximately two weeks after the end of S6B, on 6 February. As can be seen in Figures 41 and 7.2, the commissioning throughout S6A and B paid dividends during S6C and D: it was during these periods that the LIGO detectors reached their peak sensitivity, surpassing that of S5.

---

[10]The laser power was set to $14\,\mathrm{W}$ at both H1 and L1 during the evening and on weekends — when there was little seismic noise from human activity — throughout S6C and D. During working hours, the laser was typically set to $10\,\mathrm{W}$.

[11]Recall from Chapter 2, increasing laser power decreases shot noise.

It was during S6C that Pipedown was completed.[12] This allowed us to perform studies to improve the sensitivity of the search that used the loudest coincident-triggers from time-slides (*loudest-slide studies*). New data-quality tools were also finished, including daily `ihope` [80] and a DQ wiki page. This page, which was generated each week, brought together information from online analyses and veto tools, including daily `ihope`, an online excess-power search (Omega), "hveto" (for *Hierarchical Veto*, this is a tool that hierarchically ranks various auxiliary channels as potential vetoes by the significance of the correlation between the vetoes and triggers from the GW channel [96]), and UPV (for *Used Percentage Veto*, this is a tool that looks for correlations between auxiliary channels and the GW channel; if a channel is strongly coupled for a set of triggers, it is used as a veto [97]). We therefore implemented the following system to analyze the data:

- Each week a different analyst would be assigned to complete the DQ study. This involved examining the DQ page for that week and checking the e-log for potential issues that would affect the CBC analysis. The analyst would record their findings on the DQ page (as this page was a wiki, it could be edited after it was generated) and then present them on a weekly teleconference devoted to the S6 analysis.

- A "lead" analyst would run `ihope` on two weeks' worth of data, with a "second" providing support (e.g., if the lead could not make a teleconference or had difficulty getting the analysis to complete, the second would provide help). We required that the lead analyst be one of the two people who performed the DQ study for one of the weeks that the `ihope` run covered. Thus the analyst would be intimately familiar with potential issues in his or her run.

- The lead or second would generate a blinded `ihope` page[13] after `ihope` completed (typically a few days). We would look over the page on the S6 tele-conference, paying careful attention to loudest-slide events and nearby missed

---

[12]Pipedown was used throughout S6A and B. However, for both of those periods, it could only produce tables of loudest zero-lag events and IFAR plots. An early form of the `ihope` page and loudest-events table was used for S6A. For S6B injection finding, `PrintMissed`, and `PrintSims` tables were added. By S6C, loudest-slide tables were added along with `PlotFM` plots.

[13]Recall from section 5.7 that a "blinded" page is one that only presents results from playground-analysis time and injections. An "unblinded" page is one in which all data from all times is presented.

injections. We were particularly interested in slide events that had combined New SNRs greater than 11.3 as this would prevent detecting a GW with a SNR of 8 in each detector.[14] The DQ studies would be reviewed to see if any flags or events recorded in the e-logs could explain the loudest slides. If a flag was found that passed safety checks — i.e., it did not veto injections — it would be added to the collection of vetoes used by the search as either a category 2 or category 3 veto. UPV vetoes were also added at this point, as category 3 vetoes. The lead or second would then re-run the second coincidence stage and Pipedown with these new vetoes (this typically took a few hours). The new blinded `ihope` page would be checked again to make sure no new vetoes could be derived and that there were no glaring problems. An unblinded page would then be generated and presented on the full CBC teleconference to see if there were any GW candidates.

- This process would be repeated on successive weeks, with each new analysis using the updated vetoes from the previous two weeks. The analysis was thereby fine-tuned as S6C continued.

This method of using loudest slide triggers resulted in a number of vetoes being generated specifically for the low-mass CBC search. Many of these vetoes were only used once, as they were based on a specific event that occurred at one of the detectors, such as a computer malfunctioning on a particular day. However, the study did result in several long-term vetoes being implemented. Section 7.3.1 gives an example of one of these vetoes. Routinely un-blinding the analysis also decreased the latency between when data was taken and when results were obtained. By the end of S6C and throughout S6D we were un-blinding results and checking for GW signals approximately two weeks after the data had been taken.

We decided to decrease the mass-range of our template bank to $2 \leq M_{\text{total}}/M_\odot \leq 25\,M_\odot$ at the end of S6C, based on results obtained during that epoch. This decision to change the upper-mass limit of the bank was influenced by two observations: first,

---

[14]Recall from section 3.4 that combined New SNR is calculated from the quadrature sum of the single-detector triggers. If a loudest-coincident trigger had a combined New SNR greater than 11.3, all zero-lag triggers would have a false alarm rate of at least 1 per few years. We cannot claim statistical confidence that a zero-lag trigger was from a GW signal with a FAR that large.

we found that the templates in the 25–35 M$_\odot$ range frequently created triggers with relatively large New SNR when a glitch existed in the data. Nearly half of the top 10 loudest events came from this part of the template bank in each of the two week periods, despite it covering less than a third of the total mass space. The loudest-slide events in the medium and high chirp-mass bins[15] were also largely dominated by this mass range. The other factor leading to the decision was that the high-mass CBC search, which covers the mass range $25 \leq \mathrm{M_{total}/M_\odot} \leq 100$, overlapped the low-mass search in this region. This overlap was unneeded since the template placing algorithm in `lalapps_tmpltbank` would adequately cover the $\mathrm{M_{total}} = 25\,\mathrm{M_\odot}$ line within each search. We therefore had two questions to answer: first, which search was more sensitive to astrophysical GW signals in the 25–35 M$_\odot$ region, the low-mass or the high-mass? Second, if the high-mass search was more sensitive in this range, and we decreased the upper limit of the low-mass search template bank, what would the effect on the low-mass search be?

To answer the first question, we ran the low-mass and high-mass search on the same 3 weeks of data (weeks 11-14 of S6C) and then compared how well they recovered injections in the overlap region ($\mathrm{M_{total}} \in [25, 35]\,\mathrm{M_\odot}$). Figure 48(a) shows the found/missed plot[16] as a function of total mass in the overlap region for the low-mass search, and Figure 48(b) shows the same for the high-mass search. The high-mass search appears to be more sensitive: more injections are recovered at farther distances, and a few injections that were missed in the low-mass search are now found. That the high-mass search is more sensitive in the overlap region is further supported by Figure 7.2.3, which shows the ROC plot for injections in the overlap region.[17]

---

[15]Recall from Chapter 4 that we bin coincident triggers by chirp mass in order to calculate "uncombined" FARs. The bins used are $\mathcal{M}/\mathrm{M_\odot} \in \{[0.0, 3.48); [3.48, 7.4); [7.4, 15.8)\}$, which we refer to as the "low", "medium", and "high" chirp-mass bins, respectively.

[16]Recall that a "found/missed" plot compares properties of injections that are recovered at the end of the pipeline to those that are missed. Refer to section 5.6.6 for further discussion of these plots.

[17]It is not surprising that the high-mass search is more sensitive in this mass region. The high-mass search uses templates that include the "merger" part of a GW signal, whereas the low-mass templates only include the "inspiral" part of the signal. (Recall from Chapter 2 that merger occurs when the component masses pass the ISCO radius, at which point they plunge into each other.) Since ISCO grows with mass, and since GW-frequency goes as the inverse of the separation distance (cf. equations 2.53 and 2.56), high mass systems merge at lower gravitational-wave frequencies.

The high-mass search was clearly more sensitive in the overlap region, suggesting we should decrease the upper limit of the mass-range covered by the template bank of the low-mass search. To see the effect this might have on the low-mass search, a test `ihope` analysis with a template bank covering the range $2 \leq M_{\text{total}}/M_{\odot} \leq 25$ was carried out over the first two weeks of S6C and compared to the original run, which had the $2 \leq M_{\text{total}}/M_{\odot} \leq 35$ template bank. Figure 7.2.3 shows the found/missed plot as a function of chirp mass for the two runs (spinning injections are excluded in these plots). There are a few subtle differences. Several injections on the edge of the range are found with lower FARs with the lower-bank. This is expected since the volume of parameter-space we are searching is smaller, effectively giving a smaller trials factor.

Only one injection appears to have a lower false alarm rate in the analysis with a lower-mass bank: the one with $\mathcal{M} \approx 3.8 M_{\odot}$ and decisive distance $\approx 18\,\text{Mpc}$. In the analysis with $M_{\text{total}} \in [2, 35]M_{\odot}$ the injection was "found," albeit with a large combined FAR (signified by the orange dot in Figure 50(a)). In the analysis with $M_{\text{total}} \in [2, 25]M_{\odot}$ the injection was missed (signified by the red cross). This injection never should have considered "found" in the $M_{\text{total}} \in [2, 35]M_{\odot}$ analysis, however: a follow-up investigation found that a glitch occurred during the injection. This glitch created a trigger when match filtered with a high-mass template. Since `ligolw_inspinjfind` only uses a time-window to match injections with triggers, it labelled the injection as "found" even though the recovered parameters were not close to the injected. In the $M_{\text{total}} \in [2, 25]M_{\odot}$ analysis, no templates match filtered with the glitch created triggers and so the injection was considered missed, as it should have been. Thus, this injections was another indication that the $2 \leq M_{\text{total}}/M_{\odot} \leq 25$ template bank was better suited for the low-mass search.

Another metric of interest was the effect of using the $2 \leq M_{\text{total}}/M_{\odot} \leq 25$ template bank on the loudest-slide events. Table 7.2.3 compares the three loudest-slide triggers from the original run and from the new run. There was little difference in the low and medium chirp-mass bins, but the high-mass bin shows clear improvement. The loudest event in the high chirp-mass bin with the $2 \leq M_{\text{total}}/M_{\odot} \leq 35$ bank had a combined New SNR of 11.78 whereas with the smaller bank it is 9.64. This means

---

This means that higher-mass binaries will have fewer inspiral cycles and will merge in the frequency range to which the LIGO and Virgo detectors are most sensitive.

that an event in the high chirp-mass bin with SNR 8 in each IFO (the SNR on which we base projected detection capabilities) would now stick out above the background.

Using a template bank with $2 \leq M_{total}/M_\odot \leq 25$ offered clear advantages and little or no disadvantages as compared to the bank with $M_{total} \leq 35M_\odot$. For this reason we decided that we should lower the upper limit of the template bank to $M_{total} \leq 35M_\odot$ for all subsequent analysis. This decision was made as the analysis of the two weeks ending on 25 June 2009 was being completed. We therefore decided to stop S6C on that date, and begin "S6D" with the analysis starting on 26 June using the $2 \leq M_{total}/M_\odot \leq 25$ template bank. It was suggested that we re-analyze all of the S6C weeks with this new bank. However, given the relatively minor differences in the found/missed plots, we decided against it. This decision was further bolstered by Figure 7.2.3, which shows a ROC plot comparing the analysis with $M_{total} \leq 25M_\odot$ to the analysis with $M_{total} \leq 35M_\odot$. The gain in relative volume was not considered large enough to warrant a re-analysis.

7.2.4 S6D

The method of analyzing data on two week periods that was pioneered during S6C was continued throughout S6D. There were a few differences between S6C and D: in addition to reducing the template bank, described above, we also implemented two new automated vetoes: SeisVeto and a "SNR > 250" flag. These are discussed in more detail in sections 7.3.1 and 7.3.2, respectively. The other major difference from S6C was that VSR3 began during S6D.

Virgo came back online on 11 August 2009 to begin VSR3. As mentioned above, in the break between VSR2 and VSR3, monolithic suspension was installed in an effort to increase Virgo's range. Unfortunately, a mirror with an incorrect radius of curvature was installed in the process. As can be seen in Figure 41, this resulted in V1 having lower sensitivity than it did in VSR2. The rate of non-Gaussian transient noise (glitches) was also higher in V1 during VSR3.

Given Virgo's reduced sensitivity, we were concerned about leaving H1V1- and L1V1-coincident triggers that occurred during H1L1V1-coincident time in the data.
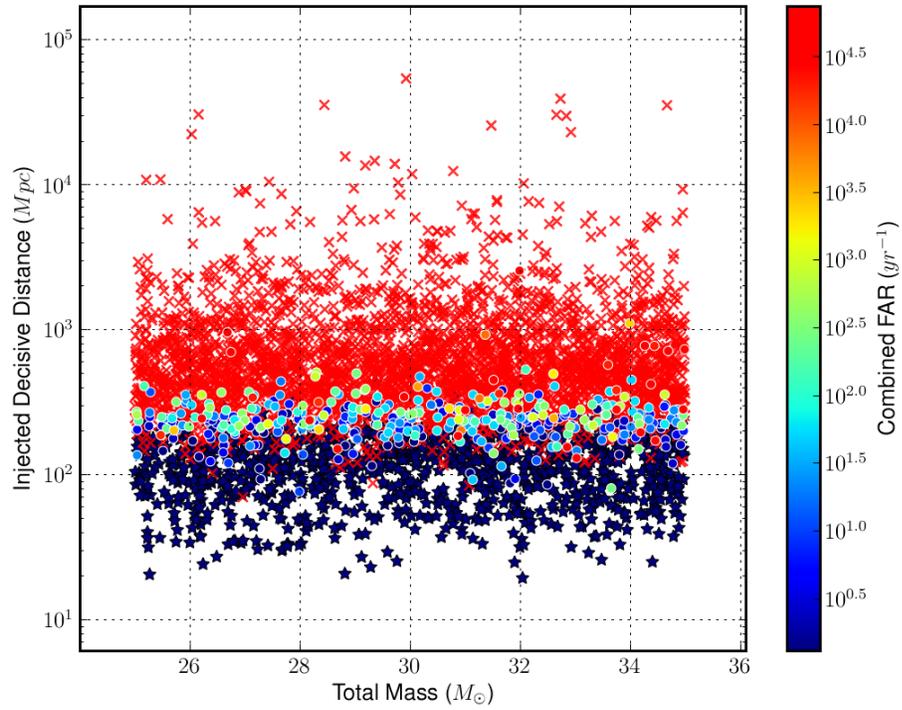
(a) Results using the low-mass search.
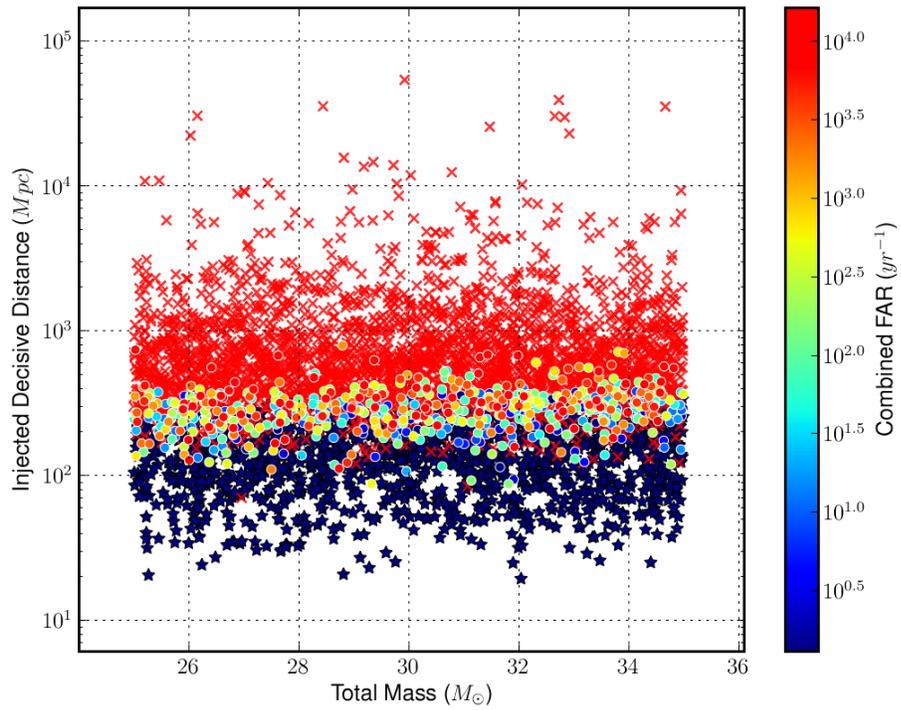
(b) Results using the high-mass search.

Figure 48 : Found/missed plots as a function of total mass in the overlap region ($25 \leq \mathrm{M_{total}/M_\odot} \leq 35$) between the low-mass and high-mass search. Plot generated by `ligolw_cbc_plotfm` by running on three weeks of S6C data.
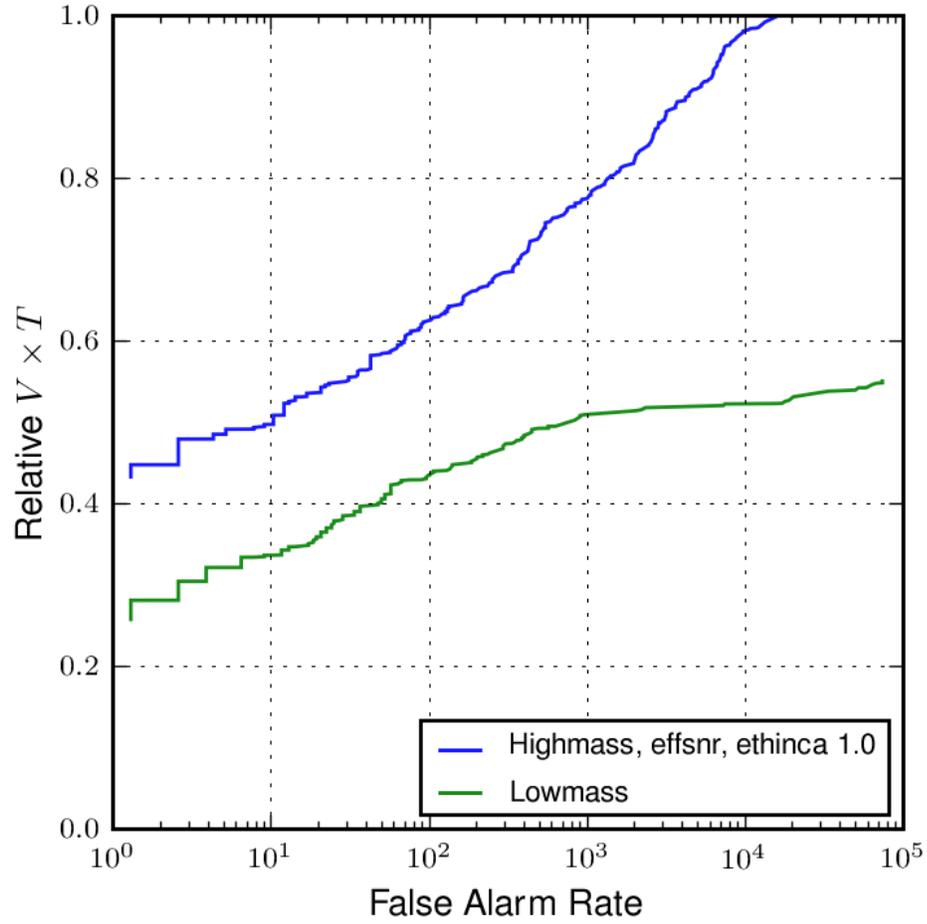
Figure 49 : ROC plot comparing the sensitive volume in the overlap region between the low-mass and high-mass CBC searches. All injection used for this plot have $M_{total}/M_{\odot} \in [25.0, 35.0]$. We see that the high-mass search has a larger relative volume at all FARs, indicating it is more sensitive to GW signals in this mass-range. For more details on ROC plots, see section 5.6.7.
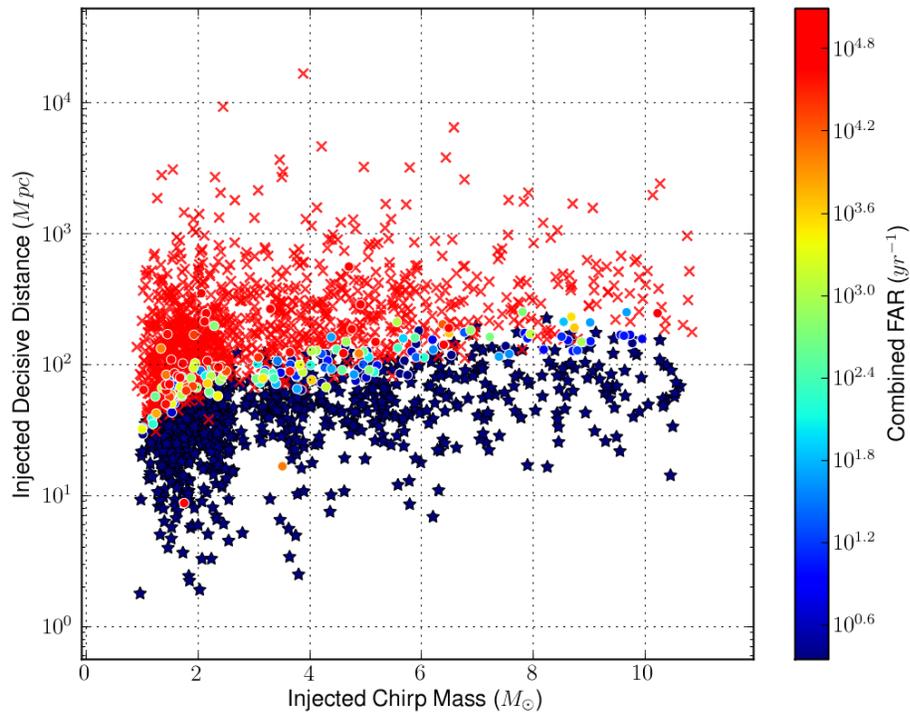
(a) Results using the full bank.
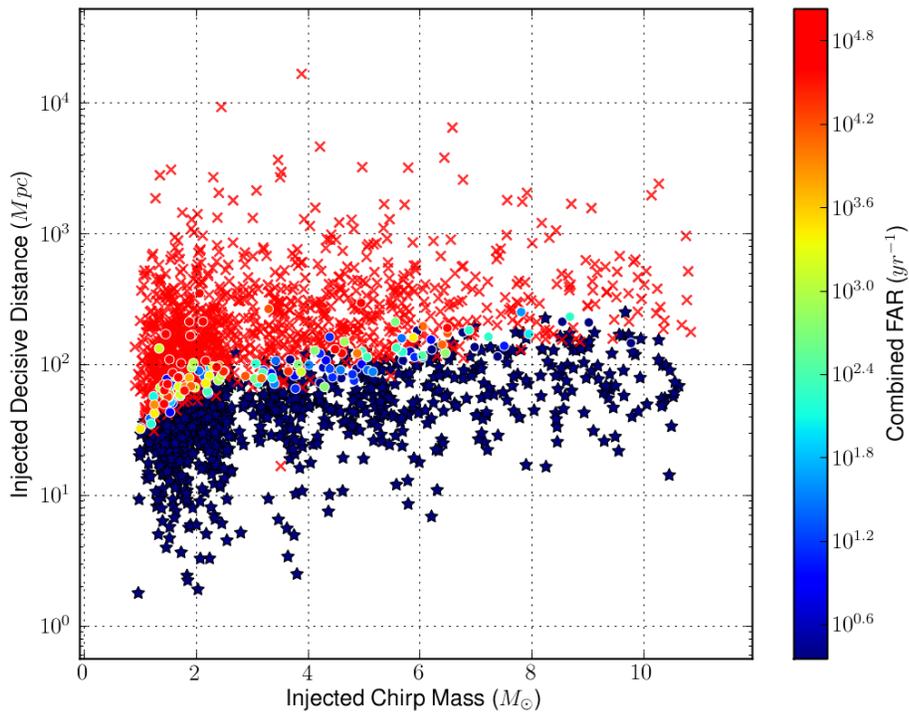
(b) Results using the reduced bank.

Figure 50 : Found/missed plots as a function of total chirp mass. Both plots were generated by `ligolw_cbc_plotfm` using the first two weeks of S6C data. The top plot shows the results from using a template bank spanning $2 \leq M_{\text{total}}/M_{\odot} \leq 35$ ("full" bank) and the lower plot shows the results from using a $2 \leq M_{\text{total}}/M_{\odot} \leq 25$ ("reduced" bank). Only non-spinning injections with total mass $\leq 25\,M_{\odot}$ were used in each plot.
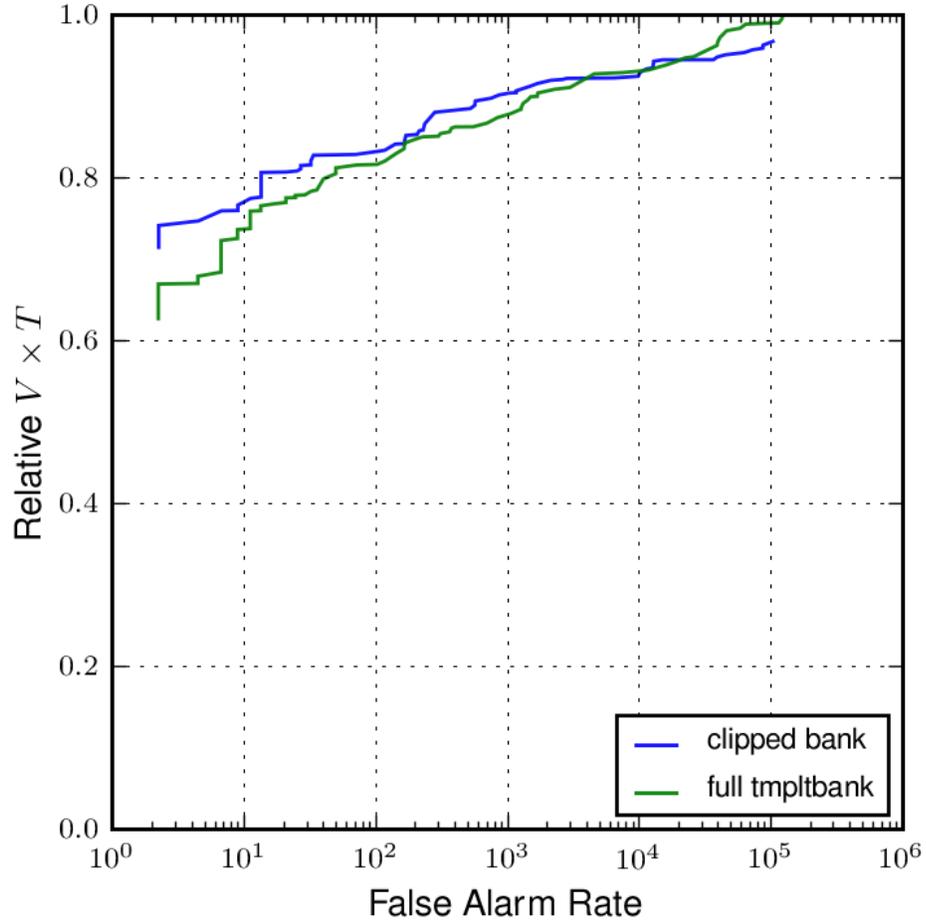
Figure 51 : ROC plot comparing the sensitive volume between the analysis using a template bank that spanned $2 \leq M_{\text{total}}/M_\odot \leq 25$ (here, labelled as the "clipped bank") and the analysis using a template bank that spanned $2 \leq M_{\text{total}}/M_\odot \leq 35$. Only injections distributed uniformly in linear distance were used in this plot; log-distributed injections showed similar results.

As discussed in Chapter 4, if we use all the possible coincidence types[18] in H1L1V1-coincident time, we have a trials factor of 12 (4 coincidence types × 3 chirp-mass bins). This means that the combined FAR of all (loud) triggers will be 12 times larger than the uncombined FAR. In doing so we have treated all the coincidence types with equal weight. However, with V1's reduced sensitivity, it was far less probable that an H1V1- or L1V1-coincident trigger was a caused by a GW than H1L1- or H1L1V1-coincident triggers. That the combined FARs of H1L1- and H1L1V1-coincident triggers would gain a factor of 12 due to the much weaker H1V1- and L1V1-coincident triggers would clearly be an over-estimation of the false alarm rate. The simplest fix was to simply remove H1V1- and L1V1-coincident triggers in H1L1V1-coincident time. This would decrease the trials factor in H1L1V1-coincident time to 6. Before doing this, however, we investigated what the effect of removing these triggers would be. Namely, we wanted to know if any injections would be missed, and we wanted to weigh this against the decrease in FARs for H1L1- and H1L1V1-coincident triggers.

In order to perform the study, two weeks of S6D/VSR3 data were used (from 3 September to 18 September 2010). Both `ligolw_cbc_cluster_coincs` and `ligolw_cbc_cfar` have the ability to remove specific coincidence types from a database; thus we simply re-ran `cFAR` on the Pipedown database generated by the standard run to re-compute the combined FARs with the H1V1- and L1V1-coincident trigger in H1L1V1-coincident time excluded.

Figure 52(a) shows the found/missed plot in H1L1V1-coincident time for the "original" analysis (with V1 doubles included); Figure 52(b) shows the same, but with the V1 doubles excluded. Some injections do appear to be missed when the H1V1- and L1V1-coincident triggers ("V1 doubles") are excluded. To get a better handle on how many, and whether or not they are really "found" injections (as opposed to being a glitch within the injection-finding window), the plots in Figure 7.2.4 were created. The top plot shows found/missed injections in H1L1V1-coincident time as a function of coincidence type in the original analysis (with V1 doubles included). There are clearly some injections found as H1V1- and L1V1-coincident triggers, some of which

---

[18]Recall that a "coincidence type" refers to the detectors contributed to a coincident trigger. For example, if a trigger in H1 is coincident with a trigger in V1, then the coincidence type of the trigger is H1V1. In H1L1V1-coincident time there are four possible coincidence types: H1L1-, H1V1-, L1V1-, and H1L1V1-coincident triggers.

are louder than all time-slide (or "background") coincidences. To get a sense of how many of these triggers are actually due to injections (as opposed to being from a glitch that occurred near the time of the injection), the bottom plot shows the accuracy of the recovered chirp mass: triggers with a recovered chirp-mass fractional accuracy ~ 0.0 are most due to injections. When more detailed information was printed for some of these injections, it was found that their effective distance in V1 was lower than that in H1 or L1, meaning that, despite Virgo's poor over-all sensitivity, there were still small areas in the sky to which V1 was more sensitive due to the relative antenna patterns of the detectors.

Still, the relative number of injections found as H1V1- and L1V1-coincident triggers as compared to the number found as H1L1- or H1L1V1-coincident triggers is clearly small. To weigh the gain in detecting these extra few injections against the cost of doubling the trials factor, the ROC plot in Figure 7.2.4 was created. This plot compares the relative sensitive volume of leaving H1V1- and L1V1-coincident triggers in H1L1V1-coincident time (blue curve) to excluding them (green curve). While the two curves are roughly the same at high FARs, removing the V1 doubles provides more sensitive volume at lower FARs, which is the region of most interest since we would never claim a detection at the higher FARs.[19] Note in particular that at lower FAR the green curve has approximately half the FAR as the blue curve at the same relative volume, as expected from the reduction in trials factor. Based on these results and the small number of injections that would be missed, we therefore decided to exclude H1V1- and L1V1-coincident triggers in H1L1V1-coincident time from the analysis for S6D.[20]

7.3 DQ Issues

In the above sections we have described some of the data quality issues that arose during S6 that resulted in tuning changes. These were the elevated trigger rate,

---

[19]Also, recall from section 5.6.7 that ROC plots can be affected by injections being mis-labelled as found if a glitch occurs near the time of the injection. This occurs most often at high FARs.

[20]Note that we still kept all triggers in H1V1- and L1V1-coincident time. Since these times are mutually exclusive from H1L1V1- and H1L1-coincident time, leaving them in the analysis had no effect on the FARs of H1L1- and H1L1V1-coincident triggers. Thus we only stood to gain by considering these times.
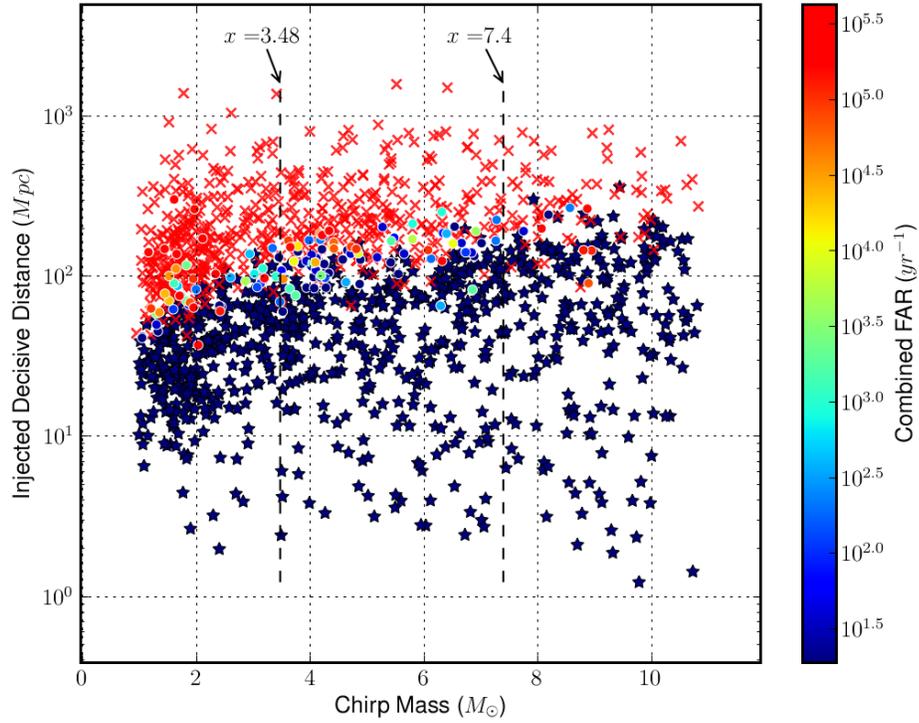
(a) H1V1 and L1V1 doubles included.
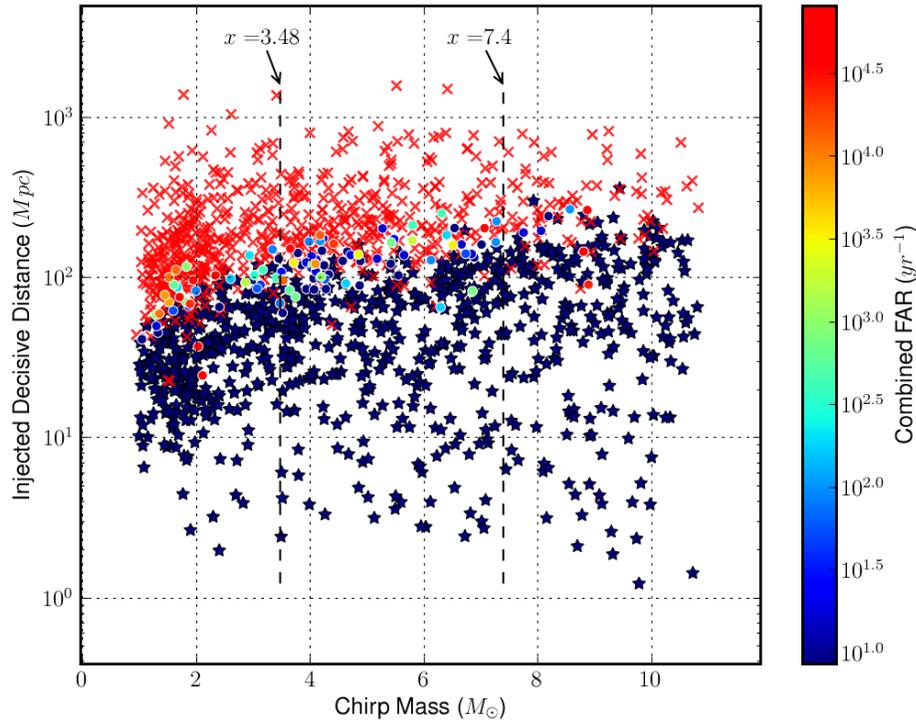
(b) H1V1 and L1V1 doubles excluded.

Figure 52 : Found/missed plots as a function of chirp mass in H1L1V1-coincident time. The top plot includes H1V1- and L1V1-coincident triggers, the bottom plot was created with H1V1 and L1V1 coincidences excluded. Both plots created by `ligolw_cbc_plotfm` using two weeks of S6D/VSR3 data.
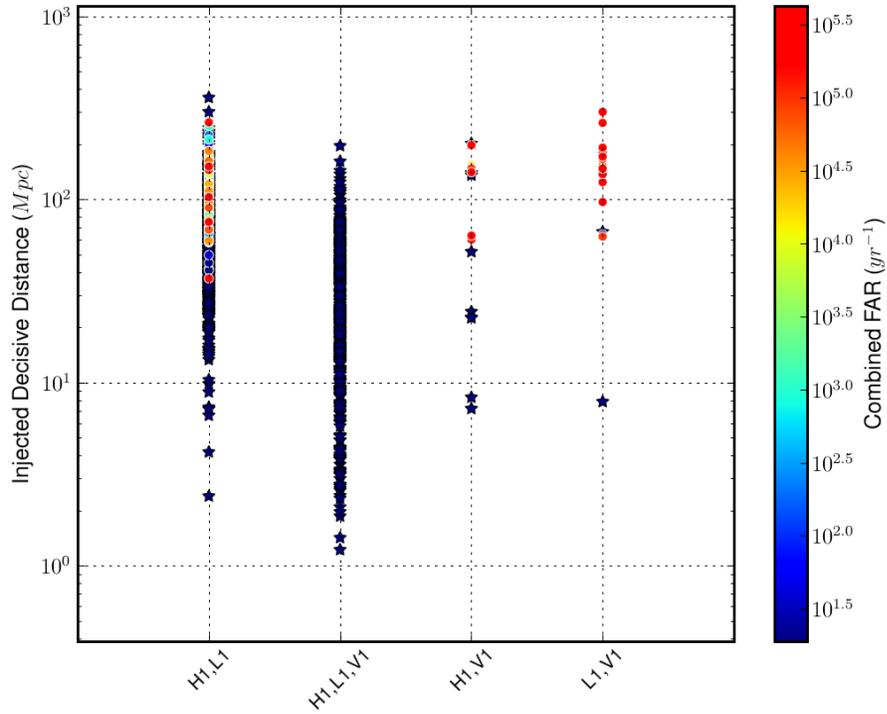
(a) Decisive distance v. coincidence type.
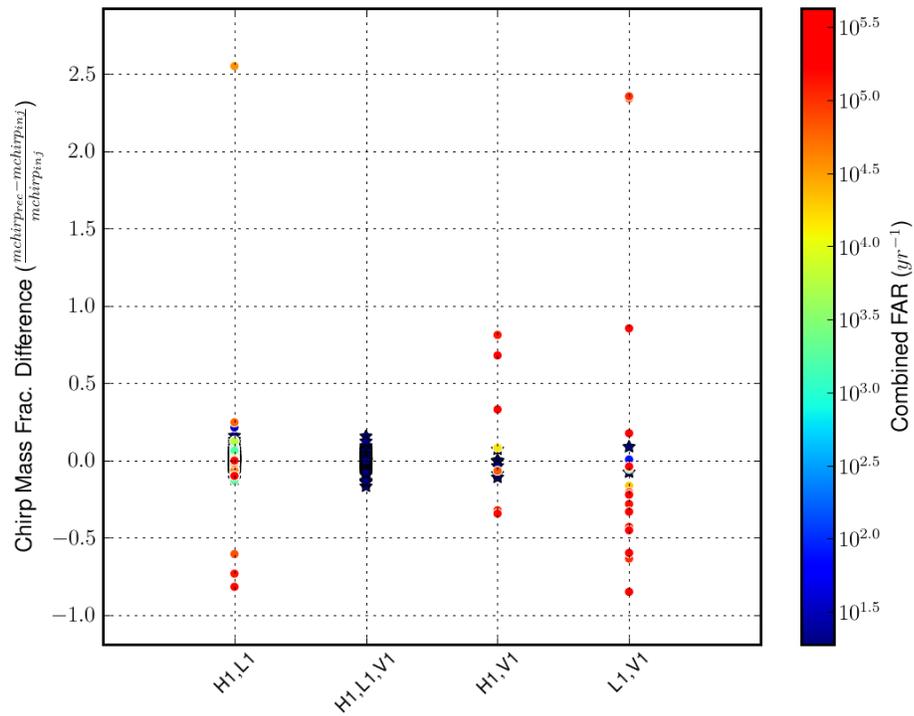
(b) Recovered chirp mass accuracy v. coincidence type.

Figure 53 : Found injections as a function of the coincidence type that they were found with. Both plots were created by `ligolw_cbc_plotfm` using the same data as used in Figure 7.2.4.
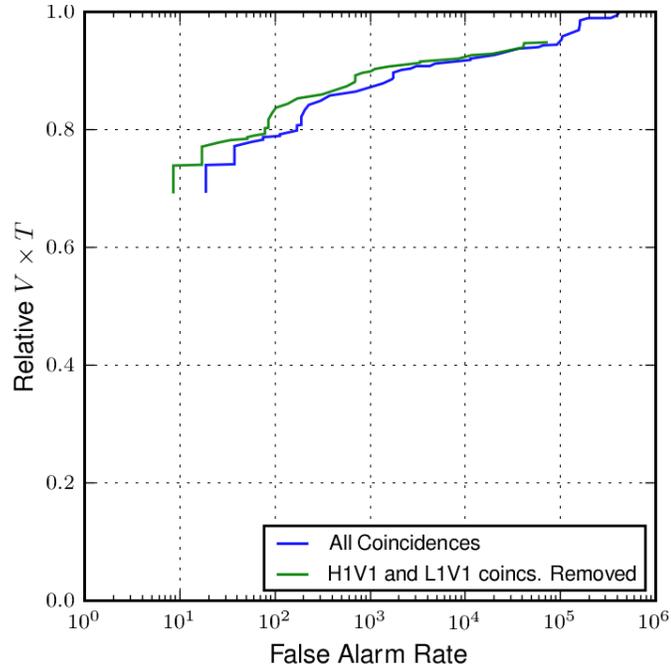
(a) Injections distributed uniformly in linear distance.
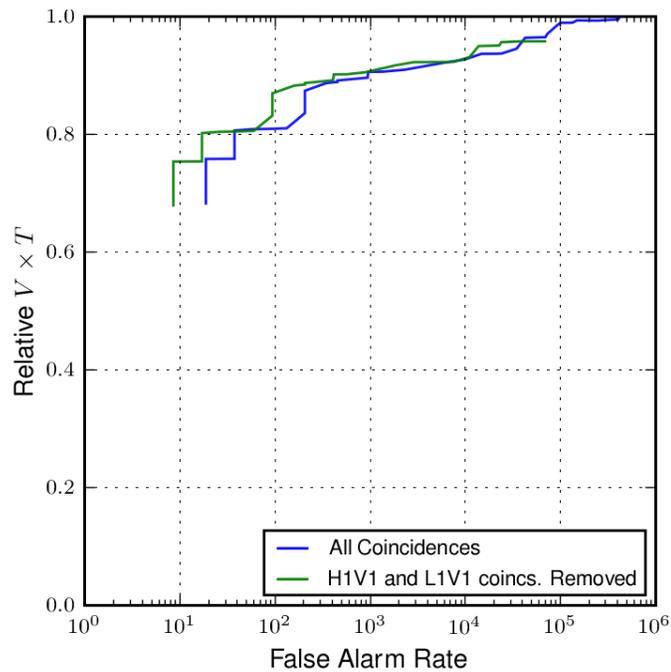
(b) Injections distributed uniformly in log distance.

Figure 54 : ROC curves comparing relative sensitive distance in H1L1V1-coincident time when H1V1- and L1V1-coincident triggers are included (blue curve) to when they are excluded (green curve). The top plot was created using injections distributed uniformly in linear distance; in the bottom injections were distributed uniformly in log distance. Both plots created by `lalapps_cbc_plotroc` using the same data as used in Figure 7.2.4.

the switch to a smaller template bank, and the decision to drop H1V1- and L1V1-coincident triggers in H1L1V1-coincident time during S6D/VSR3. Here we discuss two of DQ issues that we found during S6 and how we dealt with them. The two DQ topics listed here are by no means exhaustive. They are simply meant to highlight some results from detailed loudest-slide and other DQ studies.

7.3.1 The H1 `LVEA_SEISZ` Veto: An Example Veto using Loudest Slides

As described in section 7.2.3, during S6C and D we used the loudest-slide triggers from the low-mass CBC search to identify potential new vetoes. Many of the resulting vetoes were only used once based on observations noted in the e-log of some event at the detector sites. However, the loudest-slide (i.e., background coincident triggers computed from time-slides that had the smallest FARs) results did result in a few vetoes that were used extensively. Here we present an example of one of those flags, the H1 `LVEA_SEISZ` veto.

When we looked at the initial blinded `ihope` page for weeks 13 and 14 of S6C (1 May – 14 May 2009), we found that the loudest-slide trigger in the high chirp-mass bin ($\mathcal{M} \geq 7.4 \mathrm{M}_\odot$) had a relatively large combined New SNR of 13.4. The `ihope` analysis of the next two weeks (15 May – 28 May 2009) also revealed a relatively loud slide trigger, this time in the medium chirp-mass bin ($3.48 \leq \mathcal{M}/\mathrm{M}_\odot < 7.4$) with a combined new SNR of 11.9. Table 22 lists some details about these triggers. There were no entries in either the Hanford or Livingston e-logs to suggest a cause for these events. However, as can be seen in Figure 7.3.1, the Omega scans of the H1 GW channel showed similar signatures. When a full-Omega scan was run of every channel at Hanford, we found considerable seismic noise occurring in the `LVEA_SEISZ` channel during both events. Figure 7.3.1 shows the scan from this channel during these events. The `LVEA_SEISZ` channel monitors vertical ground motion in the *LVEA*, which is the building that houses all of the interferometer's central optics, such as the beam splitter, intermediate test masses, and dark port, as well as the laser.[21]

We therefore developed a flag that monitored the RMS value of the `LVEA_SEISZ` channel in the 3–10 Hz band [98]. When the channel exceeded 1000 counts (counts are an arbitrary unit used for LIGO channels; they are a measure of the amplitude

---

[21]The x and y seismic channels also picked up motion. We used the z channel as it was found to be the most efficient at targeting resulting glitches in the GW channel.

of the channel), the flag would go off. Efficiency studies found the flag to be good at targeting noise transients caused by the seismic noise, with little "deadtime" (this is how much time the flag is on for). It was therefore added as a category-3 veto. Table 23 shows the loudest-slide events in the high and low chirp-mass bin in weeks 13 and 14 and weeks 15 and 16, respectively, after the application of the veto. The tail of the background distribution has dropped substantially, and the loudest event is now below a combined New SNR of 11.3. This means that if a GW signal created a coincident trigger with a New SNR of 8 in each detector, it would be louder than all background coincident triggers.

Initially, we only intended to use the flag for weeks 13-16 of S6C. Studies on the later weeks found that the flag became even more efficient, however, and so the flag was entered into online monitors as:

$$H1:DMT-BRMS\_SEISMIC\_LVEA\_Z\_3\_10\_HZ\_THRESH\_1E3;$$

it was used as a veto until the end of S6C. As we had already opened boxes on weeks 1-12, we did not apply it to those weeks. Further studies showed it would not have been as effective during those weeks, anyhow. Why this is, and what the cause was for the increased seismic noise in later weeks of S6C, is unknown [95].

In S6D the `LVEA_SEISZ` veto was superseded by SeisVeto. SeisVeto improved on `LVEA_SEISZ` by used multiple seismic channels. The output from these channels were loaded into the hveto algorithm along with daily `ihope` triggers to find the strongest correlated channels each day. This automation gave even better efficiency with very little deadtime [99]. For more details on SeisVeto, see [100].

7.3.2 The "Spike" Glitch

One of the major DQ issues of S6 was the "Spike" Glitch. Beginning in August 2009, loud, short-duration ($\sim 1\,\mathrm{ms}$) glitches began to appear in the Livingston data. These glitches were characterized by a sharp downward "spike" in the GW channel followed by "ringing" (due to the response filters in the interferometer); Figure 7.3.2 shows a close-up of one. The glitch would vary in strength; some were so loud that they could be seen in the raw (unfiltered) time-series of the GW channel. Figure 58(a) shows the time series of an example loud spike and Figure 58(b) shows the corresponding Omega scan. The high-amplitude noise surrounding the glitch is due to the impulse

Chirp Mass Bin	Run	$\mathcal{M}(M_\odot)$	$M_{\text{total}}(M_\odot)$	Combined New SNR
Low	Full Bank	2.34	6.26	9.85
	Reduced Bank	2.68	10.37	9.81
Medium	Full Bank	5.36	18.88	10.63
	Reduced Bank	5.24	19.91	10.29
High	Full Bank	12.87	29.61	11.78
	Reduced Bank	9.86	23.11	9.64

Table 21 : Comparison of loudest slide events between low-mass CBC runs, one using the "full" template bank ($2 \leq M_{\text{total}}/M_\odot \leq 35$) and one using the "reduced" bank ($2 \leq M_{\text{total}}/M_\odot \leq 25$). Results are taken from the first two weeks of S6C data. The low, medium, and high chirp-mass bins are defined as $\mathcal{M} \in [0.0, 3.48)$, $[3.48, 7.4)$, and $[7.4, \max(\mathcal{M})]$, respectively, where $\max(\mathcal{M})$ is the largest possible chirp mass with each bank.

Analysis Period	Chirp-mass Bin	$\mathcal{M}(M_\odot)$	$M_{\text{total}}(M_\odot)$	Combined New SNR	IFO	Single-IFO End Time (GPS seconds)
1 – 14 May	High	12.7	33.3	13.4	H1	957127982.06
					L1	957128122.02
15 – 28 May	Medium	4.0	18.8	11.9	H1	958413266.29
					L1	958413311.36

Table 22 : The loudest-slide events in the high chirp-mass bin ($\mathcal{M} \geq 7.4 M_\odot$) of weeks 13 and 14, and in the medium chirp-mass bin ($3.48 \leq \mathcal{M}/M_\odot < 7.4$) of weeks 15 and 16, respectively, of S6C prior to application of the LVEA_SEISZ veto. (Being the loudest slide events in their respective bins, both of these events had 0 combined FAR.) Both of these events were found to be caused by seismic noise at Hanford.
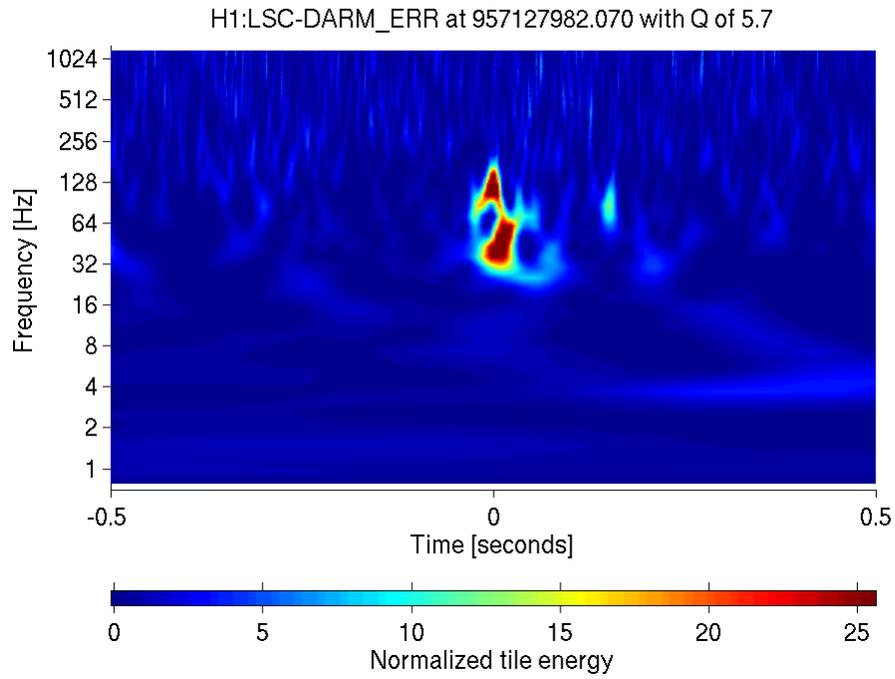
(a) H1 GW channel at 957127982.06 (the loudest-slide event in weeks 13 and 14).
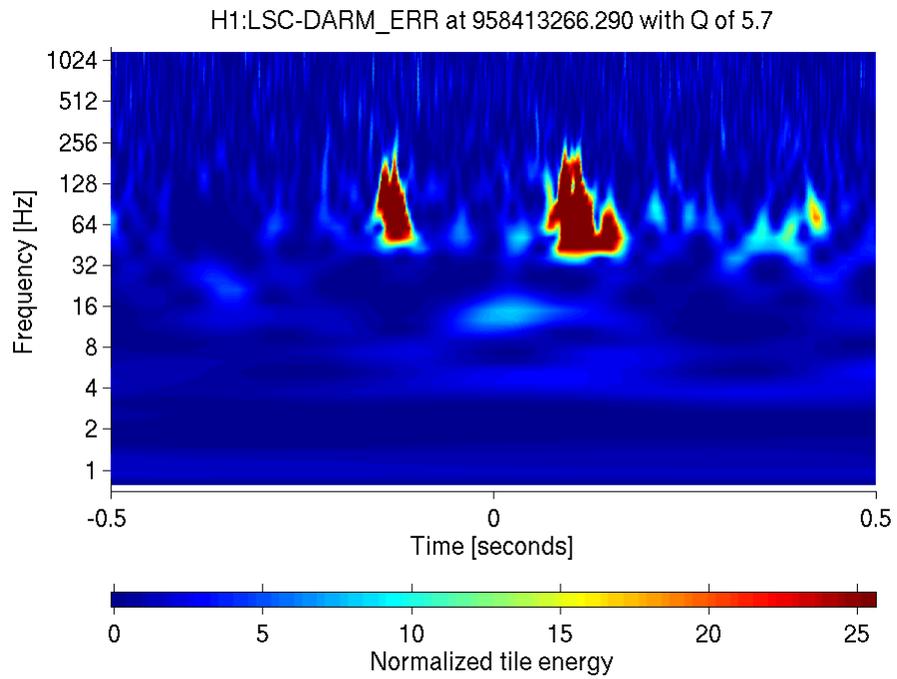
(b) H1 at 958413266.29 (the loudest-slide event in weeks 15 and 16 of S6C).

Figure 55 : Omega scans of the gravitational-wave channel (DARM_ERR) during the loudest-slide events shown in Table 22.
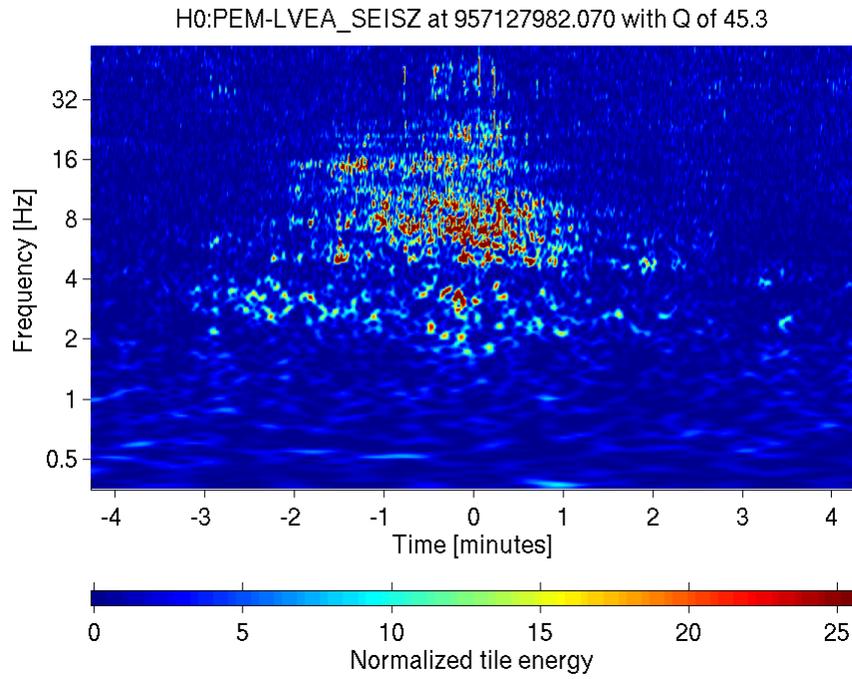
(a) H1 LVEA_SEISZ at 957127982.06 (the loudest-slide event in weeks 13 and 14).
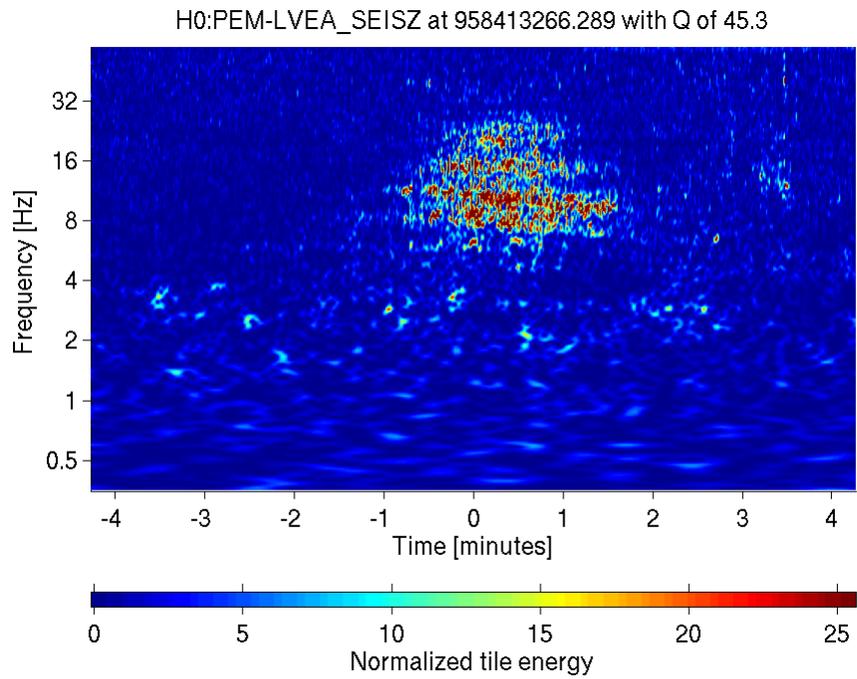
(b) H1 LVEA_SEISZ at 958413266.29 (the loudest-slide event in weeks 15 and 16 of S6C).

Figure 56 : Omega scans of the LVEA_SEISZ channel during the loudest-slide events shown in Table 22. Note that the time scale is plotted in minutes here whereas in Figure 7.3.1 it is in seconds.

response of Omega's whitening filters and the way they average power over time to highlight transients in the power spectrum.

The spike glitch also adversely affected the CBC match filter. Figure 7.3.2 shows a typical response by `lalapps_inspiral` when a spike glitch passes through; in this case the glitch was the one shown in Figure 7.3.2. This plot was created with the $2 \leq M_{\text{total}}/M_{\odot} \leq 35$ template-bank, without any clustering. Each trigger is colored by its τ_0 value, which essentially gives the duration of the template. The large spike in SNR that begins at the time of the glitch ($0\,\text{s}$) is due to the impulse response of the templates. Although the SNR of these triggers is quite large (in this case, the loudest trigger had a ρ 8300), they did not typically show up in our loudest events nor our loudest slides. This is because they would have very poor χ^2 values, resulting in low new SNRs. Of greater concern was the "shoulders" surrounding the spike and the tail proceeding the glitch. This tail could last several seconds after the glitch and its triggers could have lower χ^2 values. We therefore wished to find the cause of the glitch, so that the detector could be fixed, or, if that was not possible, so we could veto it.
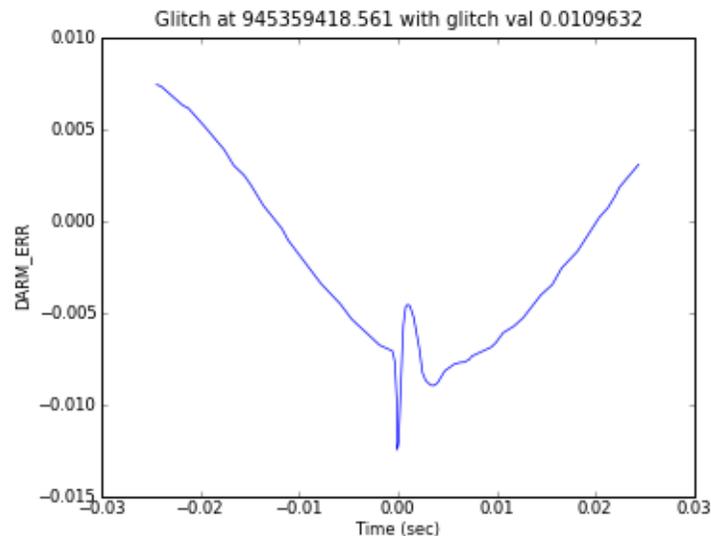
Figure 57 : A close-up of a spike glitch. Shown is the GW channel (DARM_ERR) raw time series.

Unfortunately, despite a number of intensive data-quality studies and feedback from instrumentalists, we were never able to determine the cause of the spike glitch. Not only did this prevent us from fixing the instrument so the glitch would go away,

but it also made it difficult to veto. As stated above, we typically only veto triggers if they can be strongly correlated with an auxiliary channel. This is to avoid accidentally vetoing a gravitational wave. The spike glitch was clearly not a GW, but we had no other way to veto it than to use the GW channel. We therefore decided to use a DQ flag based on daily `ihope` triggers. Named `H1:DCH-CBC_SNR_GT_0250`, this flag would go off whenever triggers exceeded a SNR threshold of 250. The flag was used as a CAT3 veto in S6D, and a "padding" of ± 8 s was added to it. The padding was added to remove the "shoulder" triggers created by the spike. 8 s was chosen based on the length of the inverse spectrum truncation.

There was some concern that vetoing triggers using the output of daily `ihope` would cause us to miss a nearby event. However, any real GW event that had a SNR above 250 would stick above the background at CAT2. Since we check results after CAT2 vetoes have been applied, and the SNR > 250 flag was applied at CAT3, this veto would not prevent us from detecting such an event.

7.4   Results

As described in section 7.2, we initially analyzed the data in several small analysis periods — weekly runs in S6a; three month-long chunks in S6B; bi-weekly runs in S6C and D — and opened boxes on each of these analyses. We did this so we could obtain results quickly, and so we could pro-actively adapt the pipeline to DQ issues, as detailed above. However, we did this with the knowledge that we would have to re-run the pipeline over these periods. Re-runs were required in order to take advantage of final calibration studies, as well as fix minor issues in published veto segments.[22] For speed and simplicity these re-runs are being carried out in larger $\sim$ 6-week long chunks. S6A is being analyzed with one run; S6B is divided into two chunks, one prior to 11 November 2009, and the other from 11 November to 11 January 2010. S6C and D are each being re-analyzed in three chunks. Aside from the grouping of analysis periods together, no other changes have been made to the runs. For example, although new vetoes were created throughout S6C and D, these

---

[22]Upon review, a CAT1 veto-flag was found to stay on (and off) for slightly longer periods than it should have. The effect on the analysis is small, amounting to a change in live time of less-than a day across all of S6. For the sake of rigor, however, we have fixed the flag for re-runs.
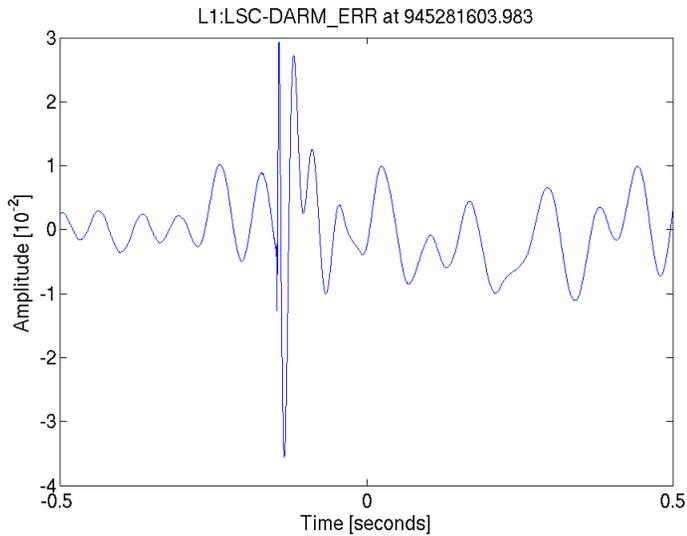
(a) Raw time series.
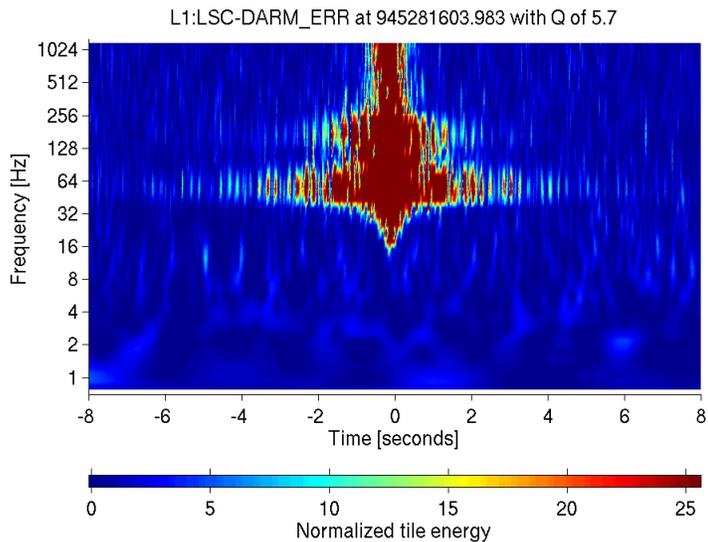
(b) Omega scan.

Figure 58 : An example spike glitch. This glitch occurred at GPS time 945281603.98 (19 December 2009 18:13:08 UTC). The top plot shows the raw time series from the GW channel; no filtering has been applied to this plot. The bottom plot shows the Omega scan of this glitch. Omega uses a whitening filter to create the bottom plot, hence the high-amplitude noise lasts several seconds.

Analysis Period	Chirp-mass Bin	$\mathcal{M}(M_\odot)$	$M_{total}(M_\odot)$	Combined New SNR	IFO	Single-IFO End Time (GPS seconds)
1 – 14 May	High	9.1	25.4	10.1	H1	957858489.74
					L1	957858414.75
15 – 28 May	Medium	4.4	20.9	10.3	H1	958306864.45
					L1	958306784.5

Table 23 : The loudest-slide events in the high chirp-mass bin ($\mathcal{M} \geq 7.4M_\odot$) in weeks 13 and 14, and in the medium chirp-mass bin ($3.48 \leq \mathcal{M}/M_\odot < 7.4$) in weeks 15 and 16, respectively, of S6C after the application of the LVEA_SEISZ veto. (Being the loudest slide events in their respective bins, both of these events had 0 combined FAR.)
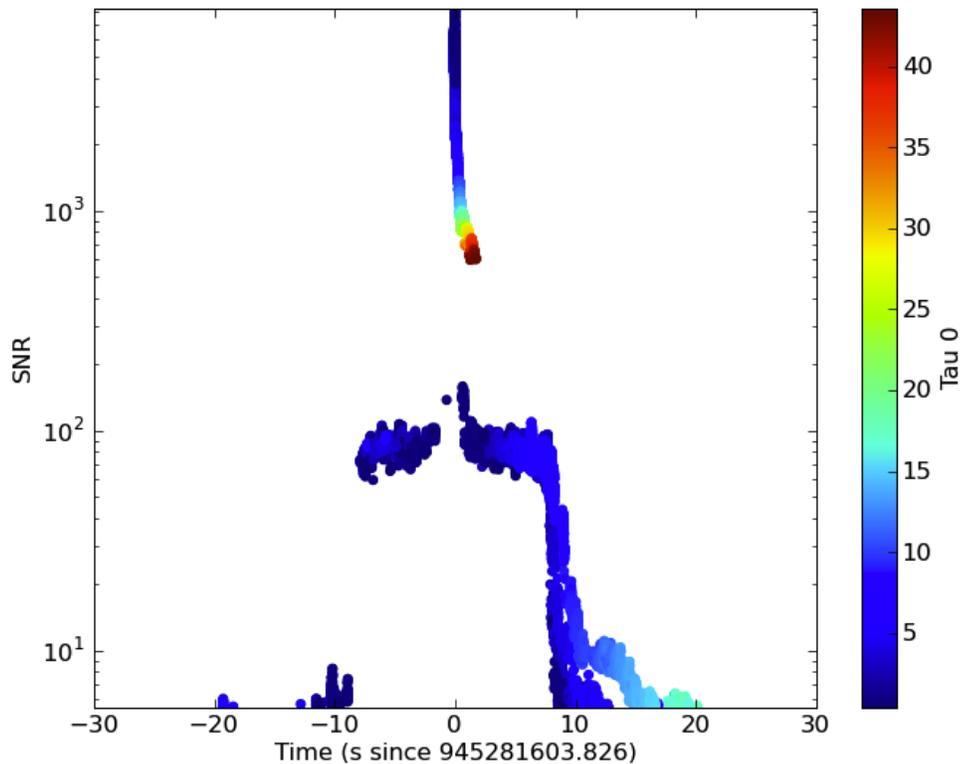
Figure 59 : The unclustered triggers created by the spike glitch shown in 7.3.2 when passed through lalapps_inspiral.

have not been retroactively applied to earlier periods.

As the changes between the original runs and the final re-runs are minor, we do not expect our result to change. The result of the S6/VSR2/3 analysis is: No gravitational-wave candidates were detected. Currently, the most significant event is an H1L1-coincident trigger in H1L1V1-coincident time with a combined FAR of 2.2 yr^{-1}. The second and third most significant triggers had combined FARs of 5.6 yr^{-1} and 9.4 yr^{-1}, respectively. All of these triggers are consistent with background: having analyzed ~ 0.5 yr of data, we would expect the loudest event to have a FAR of $\sim 2 \text{ yr}^{-1}$. Although no detection candidates have been found, we will perform detailed follow-ups on the loudest triggers in each epoch once the re-runs have completed. These follow-ups are carried out to aid in future DQ investigations.

As a part of the re-runs we are performing extra injection runs. This is so we can improve our statistics for placing upper-limits. Since these runs are still on-going, and since the final upper limits will need to be reviewed once they are produced, we do not present upper limits here.

7.5   The Blind Injection

A hardware injection was injected into the LIGO and Virgo detectors during S6D/VSR3 without the search groups' knowledge.[23] The purpose of this *blind-injection challenge* was to test the groups' abilities to detect signals and to exercise the LIGO and Virgo Collaborations' procedures in the event that a real gravitational-wave candidate is detected. The *blind injection* was found by the CBC group with a FAR low enough to be considered a detection candidate. We treated the event as if it were real: detailed studies were done to establish the significance and parameters of the event, and to vet the detectors of any possible environmental or instrumental causes. Only after all procedures were reviewed and a paper draft written was the event unveiled as an injection. Although the event was not real, we present here details of the analysis of the injection, including methods used to calculate its FAR.

---

[23]The search groups did know that zero or more blind injections could happen over the course of the Science run, however.

7.5.1 Observation and FAR Estimation

The blind injection occurred on 16 September 2010 at 06:42:23 UTC (GPS time 968654558.0) and was identified by multiple searches. It was initially observed, within a few hours, by a low-latency Burst search. Since Burst searches use un-modelled waveforms, it did not give the event a FAR small enough to be considered a detection. However, it was significant enough to stand out as a potential CBC candidate, and so the collaboration was alerted to its presence in the data.

Although we knew about the potential candidate prior to opening the box, we carried out the analysis of the two weeks containing the event as we did every other period: a blinded `ihope` page was generated, loudest-slides were examined, vetoes were updated, and a re-run was carried out with the new vetoes to prepare of the un-blinding of the results. The injection was loud enough in H1 that it was a part of the loudest H1L1-slide event in the medium chirp-mass bin during H1L1V1-coincident time. In this event the candidate in H1 was coincident with a random noise event in L1. The slide event had a combined New SNR of 11.56 and the chirp was visible in the H1 Omega scan. (See Table 25 for more details.) As with any loud slide-event we checked the DQ studies for the week and the e-log to see if we could find an auxiliary channel to veto the times in either of the detectors.[24] No cause could be found in either detector, and so the slide event stayed in the data.

When the box was opened the injection was found as *two* events: one in the low chirp-mass bin, and one in the medium chirp-mass bin. Both triggers were found as H1L1 coincidences in H1L1V1 time and had combined FARs of zero; i.e., both events were louder than all the background in their respective bins. Figure 60 shows the unblinded IFAR plot for H1L1V1-coincident time from the analysis and Table 24 shows the parameters of the two events that were found. That the injection created two triggers was a result of our earlier decision to cluster coincident triggers within each chirp-mass bin. Since both triggers had zero combined FARs it was not immediately clear which trigger to keep. Further complicating the matter was that the triggers landed exactly on the bin boundary: the low-mass trigger had a chirp

---

[24]Since the blind injection is a hardware injection, it is created by actuating one of the mirrors. There is a channel that monitors the input to this mirror, and so it is possible to check whether or not an event is a blind injection without having to wait for the "envelope" to be opened. We forbid ourselves from checking this channel, however, as it would make the entire challenge pointless.

mass of $3.47\,\mathrm{M_\odot}$ and the high mass trigger had a chirp mass of $3.48\,\mathrm{M_\odot}$; the boundary was $\mathcal{M} = 3.48\,\mathrm{M_\odot}$. Since we needed to estimate a combined FAR for the candidate, we decided to find an estimate in each bin. Whichever trigger had the lower-combined FAR we would keep.

In order to estimate a FAR for the candidate event we initially combined data from adjacent analysis periods. This was standard procedure; it was performed on several different occasions in prior instances when a trigger was found with zero FAR in a single week of analysis. However, the candidate was the loudest H1L1-coincident trigger in all of S6. Thus, even if we used all the data in S6, we could still only estimate the *uncombined* FAR to be < 1 in 23 years. This was not small enough to claim a detection. It was also not clear that using data from all of S6 was the correct approach, since the data quality had changed substantially from S6A to when the event occurred, in S6D. Additionally, computing a combined FAR using data from across epochs would have been difficult, since in S6A and B the trials factor in H1L1V1 time was 12, in S6D it was 6, and in S6C there was only H1L1 time.

To get a better estimate of the FAR, two methods were pursued. One was to perform an extrapolation using data from the 100 slides. Figure 62 shows a histogram of all of the H1L1 triggers in S6C and D as a function of combined new SNR (ρ_{nc}) in the low and medium chirp-mass bins. Zero-lag triggers are represented as blue dots; slide triggers are black. The blind injection is the blue dot all the way to the right, at $\rho_{\mathrm{nc}} \approx 12$. The structure of the distribution is largely due to χ^2 re-weighting and the SNR cutoff: non-Gaussian triggers have been down-weighted, removing any tail in the distribution, and the peak at $\rho_{nc} \approx 8$ is due to the SNR cut. Indeed, any trigger below a combined new SNR of 7.7 ($= \sqrt{2} \times 5.5$) must be a trigger that was down-weighted, since we apply the SNR cut at 5.5. For combined new SNRs > 8.5 the distribution appeared to be Gaussian, particularly in the low chirp-mass bin. We therefore performed a least-squares fit to the background distribution using a Gaussian of the form:

$$A exp\left[-\frac{x^2}{2\pi\sigma^2}\right]$$

where A and σ^2 were the fit parameters. We used the slide data in the range $\rho_{\mathrm{nc}} \in [8.5, \max(\rho_{\mathrm{nc}})]$ to perform the fit. The green-dashed line shows the result of the fit.

Next, we created a cumulative histogram plot in each bin, shown in Figure 63,

with the y-axis normalized by the zero-lag live time. Using the fitted parameters from the non-cumulative histograms, we plotted the complimentary error function:

$$y(\rho_{nc}) = A \mathrm{erfc}\left(\frac{\rho_{nc}}{\sqrt{2\sigma^2}}\right)$$

This is shown as the green-dashed line, from $\rho_{nc} = 8.5$ to the combined New SNR of the blind injection. Thus, the point on the y-axis where the green-dashed line ends gives the uncombined FAR of the blind injection. In the low chirp-mass bin this yielded a FAR ≈ 1 in 4×10^6 years, and in the medium bin we obtained a FAR ≈ 1 in 2×10^6 years.

On the cumulative histogram plots we additionally plotted the probability density function that the zero-lag events came from the background distribution as a color map. This map was computed using the Poisson distribution (equation 4.5 in Chapter 4). At each point along the x-axis, λ was determined using the cumulative rate of the closest slide data point greater-than that point. For points past the loudest-slide point (indicated by the vertical black line), the extrapolation is used. Note that this was done prior to dividing by the zero-lag live time, so that the cumulative-rates were unit-less; i.e.:

$$\lambda = N_{\mathrm{cum,slide}} \frac{T_f}{T_b}$$

where $N_{\mathrm{cum,slide}}$ is the number of slide triggers with $\rho_{nc} \geq$ the combined new SNR at the given point on the x-axis, T_f is the zero-lag live time, and T_b is the total slide live time. The probability density at each point along the y-axis at that x-value is then determined by substituting the (unit-less) y-value as k in equation 4.5. The entire plot was then divided by T_f to get it into units of yr^{-1}. The color map ends at $y = 1/T_f$ because the Poisson distribution is not defined for $k < 1$. (Technically, it is not defined for any non-integer k. To make the plot cleaner we interpolated between integer points using the Gamma Function.) The black dashed-lines indicate $N\sigma$ points, as determined by the Gaussian distribution. For example, 2σ corresponds to a probability density of 0.045, and so the second black dashed-line maps out points where the distribution is equal to 0.045. We do this because a false alarm probability equal to 5σ is generally considered the "gold" standard for a new detection in the particle physics community.

While the extrapolation appears to fit the low chirp-mass bin well, it clearly deviates from the slide distribution in medium chirp-mass bin. This is largely due

to the loudest slide event, which happens to be the event shown in the first row of table 25: it resulted from the blind injection in H1 being coincident with a random noise event in L1. If we remove $\pm 8\,s$ around the end time of the injection from each detector when performing the slides, then any slide event that could be caused by the blind injection goes away. Figure 64 shows the cumulative histogram for the low and medium chirp-mass bins when this is done. The low chirp-mass bin is largely unaffected, but in the medium chirp-mass bin the fit matches better. The extrapolated FARs were largely unaffected when doing this; both low and medium chirp-mass bins still gave ≈ 1 in $4 \times 10^6\,$yr and ≈ 1 in $2 \times 10^6\,$yr, respectively.

We did not feel confident using these extrapolated FARs. The validity of extrapolating over six orders of magnitude was hard to determine. Additionally, we had no expected model for the background. Even if new SNR removed all non-Gaussian transients above an SNR of 5.5 we had no model for the effects of coincidence testing and the various stages of clustering. We chose to fit a Gaussian largely because that is what the slide distribution looked like. However, even when the blind injection was excluded from the background estimation, the extrapolation still appeared to deviate slightly from the slide triggers in the medium chirp-mass bin. This suggested that a Gaussian was not the correct distribution to use in the medium bin.

The 100 slides that we typically do is a small percentage of the total number of possible slides. Thus, another way to estimate the FAR was to do more time slides. Since we use offsets that are multiples of $5\,s$ between H1 and L1 in the standard search, we decided to do the maximum number of $5\,s$ slides that was possible in S6D. We limited the estimation to S6D to try to ensure that the background rate was roughly constant across the period (recall from Chapter 4 that assuming a stationary source is central to the FAR analysis). Also, rather than perform slides on rings, we used linear slides.[25] This way we could use single-detector times in the background estimation. In this manner we were able to perform 4×10^6 slides, resulting in

---

[25]Recall from section 5.3.10 that `lalapps_thinca` slides single-detector triggers on *rings* prior to doing the coincidence test. Triggers that are slid past the end time of a particular file are placed at the beginning. Since `thinca` jobs are set-up so that they only span coincident detector times (after CAT1 vetoes have been applied), any times for which a single detector is operating ("single-detector time") will not be used for the coincidence test. Contrast this to *linear slides*, in which triggers are slid without regard to when a particular `thinca` file starts or ends. Thus, single-detector times can be slid into coincident-detector time.

$\sim 1 \times 10^6$ *effective slides.* (Recall from chapter 4 that the effective slide number is obtained by dividing the total slide live time by the zero-lag live time.) This resulted in a total background live time of 2.0×10^5 years. To remove any effect on the background estimation from the first coincidence testing, we created a list of single-detector triggers with χ^2 values using a prototype single-stage pipeline. This pipeline was the same as the `HIPE` pipeline described in Chapter 5, except that χ^2 was calculated at first Inspiral and no second stage was carried out.

Two independent scripts were used to carry out the slides to check the results. Due to the computational cost of doing so many slides, only single-detector triggers that had new SNRs large enough to cause a coincidence as loud or louder than the injection's combined new SNR were considered by both scripts. Additionally, since the injection was found as an H1L1-coincident event, we did not use V1 in the analysis. Thus we effectively combined H1L1V1-coincident time and H1L1-coincident time into one, and pulled out all H1L1 coincidences from H1L1V1-coincident triggers. Both scripts agreed with each other. The result was that there were 4 coincidences with combined new SNRs $\geq$ the injection's ρ_{nc} in the low chirp-mass bin, giving an uncombined FAR of 1 in $50,000$ years. In the medium chirp-mass bin there were 5 louder coincidences, giving an uncombined FAR of 1 in $40,000$ years. This was with the candidate event left in the background. If the candidate was removed, no coincidences were louder than it, putting a limit on the uncombined FAR of < 1 in $200,000$ years. One of the two scripts was able to additionally compute the FAR using 1 second slides. When this was done, one event was found to be louder than the injection even when it were removed in the low chirp-mass bin. In the middle chirp-mass bin, however, there were still no events louder than the injection. Based on this result, along with the fact that the combined new SNR was larger in the middle chirp-mass bin, we decided to keep the trigger in the middle bin, i.e., the one with a chirp mass of $3.48 \mathrm{M}_\odot$, and discard the other.

Figure 65 summarizes these results, which is a cumulative histogram in the middle chirp-mass bin. This plot was created in the same manner as the cumulative plots shown in Figures 63 and (minus the extrapolation). Here, the zero-lag triggers in S6D (S6C is not included) are the blue triangles. The background from the standard 100-slide analysis are represented by the dots. The crosses represent the loudest triggers from the 4 million 5 second slides. The black dots and crosses are from

the background estimations in which the injection was left in the background. The gray dots and crosses are from the background estimation if the injection is excluded. Rather than use a color map to show the probability density, we plotted shaded contours to represent the 1–$5\,\sigma$ regions of the PDF. The contours were calculated using the background with the injection left in. Had the injection been a real event, this plot would have been our main figure of merit in the resulting detection paper.

The question of whether or not to leave the injection in the background was hotly debated prior to opening the envelope. If the event were a real gravitational wave, we would remove it from the data. Otherwise, leaving it in would break the assumption, made at the end of Chapter 4, that the total rate density calculated from time-slides is the same as the FAR. In this case the noise/GW terms in equation 4.36 at low FAR would be on par with the noise/noise term. Thus we could not make the approximation in equation 4.43. Yet if the event were from noise, removing it would underestimate its FAR, as well as the FAR of all other events. Ultimately we decided to present both backgrounds on the plot (and in the detection paper), but, to be conservative, quote the false alarm rate with the event left in the background. (This is why the background points with the injection left in are black, whereas the event-free background points are light gray, and why the probability contours are based on the former estimate.) Thus the quoted uncombined false alarm rate for the event was 1 in $40{,}000$ years.

To get a combined FAR for the injection we multiplied the uncombined FAR by a trials factor of 6. This gave a final false alarm rate for the event of ~ 1 in 7000 years. Since we folded all H1L1-coincident triggers in together, this was a slight overestimate. Had we included V1, some of those triggers would have been H1L1V1 coincidences, and so would have been excluded from the bin. Determining what triggers would have formed triple-coincidences would have been impossible because we folded H1L1-coincident time together with H1L1V1-coincident time. While we could have determined what triggers would have been coincident with V1 in H1L1V1-coincident time, we could not know this in H1L1-coincident time. Since the number of H1L1V1 coincidences are typically small compared to the number of H1L1-coincident triggers, however, this overestimate would have been minor.

Even with the injection left in the background, the event was still close to a 5σ event, as can be seen in 65. Thus we went a head with writing a detection paper,

and a number of intensive data-quality investigations were carried out on the period of time. The event passed all tests. The LIGO observatories were in quiet night-time operation with near maximum astrophysical range; the Virgo detector had quiet environmental conditions, except for elevated micro-seismic levels. The lack of an observed signal in Virgo was consistent with a GW. V1's sensitivity was a factor of four lower than the LIGO detectors at the time. Since the largest single-detector SNR, recorded in H1, for the event was ~ 15 (note the H1 SNR in the loudest slides in table 25), it very likely had a SNR below our threshold of 5.5 in V1.

7.5.2 Conclusions

The blind injection challenge in S6 was considered a success. The event was identified, a false alarm rate was calculated, and it was vetted by DQ investigations. For the first time we wrote a paper detailing a highly-probable GW detection. Had the event been real this paper would have been published. Even though it was not real, the experience gained will prove valuable for the first real detection, which is expected to happen in advanced LIGO. In addition to highlighting the strengths of our pipeline, it also unearthed some weaknesses. That a GW event can hurt its own FAR if left in the background estimation was not fully appreciated prior to the exercise. How exactly to deal with this scenario, particularly if there are multiple GWs in the data, is still an open question. It will be an area of active research between now and advanced LIGO.

After the event was revealed to be a blind injection, a veto was created for it and it was removed from the data. As stated above, with the injection removed, there were no gravitational-wave candidates in S6/VSR2/3. A paper presenting updated upper-limits as well as more details about the blind injection is in progress.
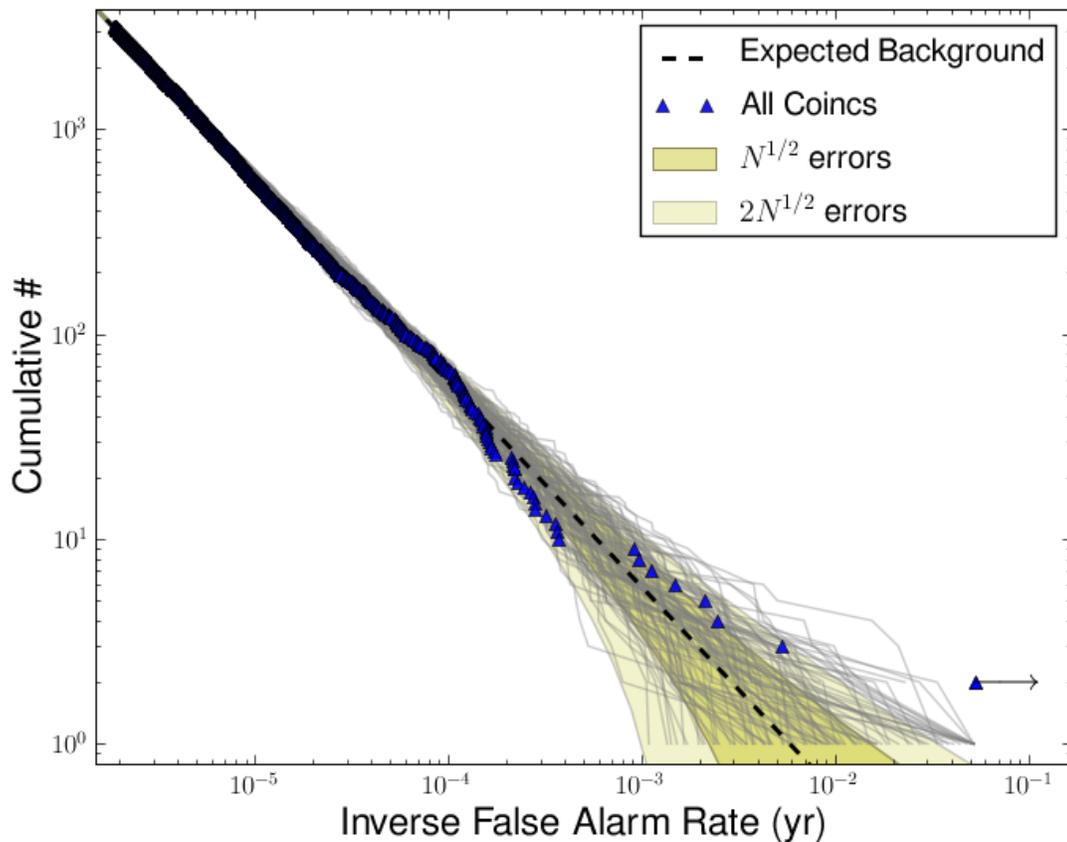
Figure 60 : IFAR plot from the two-week-long `ihope` run containing the blind injection. The two coincident triggers that the injection created can be seen as the triangle with a cumulative number of 2 at the lower-right of the plot. Note the arrow pointing to the right. This indicates that the two triggers were louder than all slide-coincident triggers in the two-week long analysis period that the injection occurred in.

Chirp-mass Bin	Combined New SNR	$\mathcal{M}$ ($M_\odot$)	M_{total} ($M_\odot$)	IFO	Single-IFO New SNR	Single-IFO SNR	Effective Distance (Mpc)
Low	12.06	3.47	24.01	H1	10.29	12.14	54.6
				L1	6.29	8.25	83.6
Medium	12.48	3.48	23.17	H1	10.29	12.14	54.6
				L1	7.06	8.854	81.2

Table 24 : The two events resulting from the blind injection that survived clustering. Both events occurred at GPS time 968654557.87, and had a zero combined FAR.

Analysis	Combined New SNR	$\mathcal{M}$ $(M_\odot)$	M_{total} $(M_\odot)$	IFO	Single-IFO New SNR	Single-IFO SNR	Single-IFO End Time $(968654558+s)$
100 slides	11.56	4.62	13.41	H1	10.33	15.34	0
				L1	5.19	5.64	-230
4×10^6 5 s slides	12.67	4.40	16.53	H1	10.33	15.34	0
				L1	7.33	8.48	-5548535

Table 25 : The loudest slide events in the medium chirp-mass bin involving the blind injection in the standard, 100 slide analysis, and in the 4×10^6 5 s slide analysis. In both cases the injection in H1 is coincident with noise in L1.


(a) H1

(b) L1

(c) V1
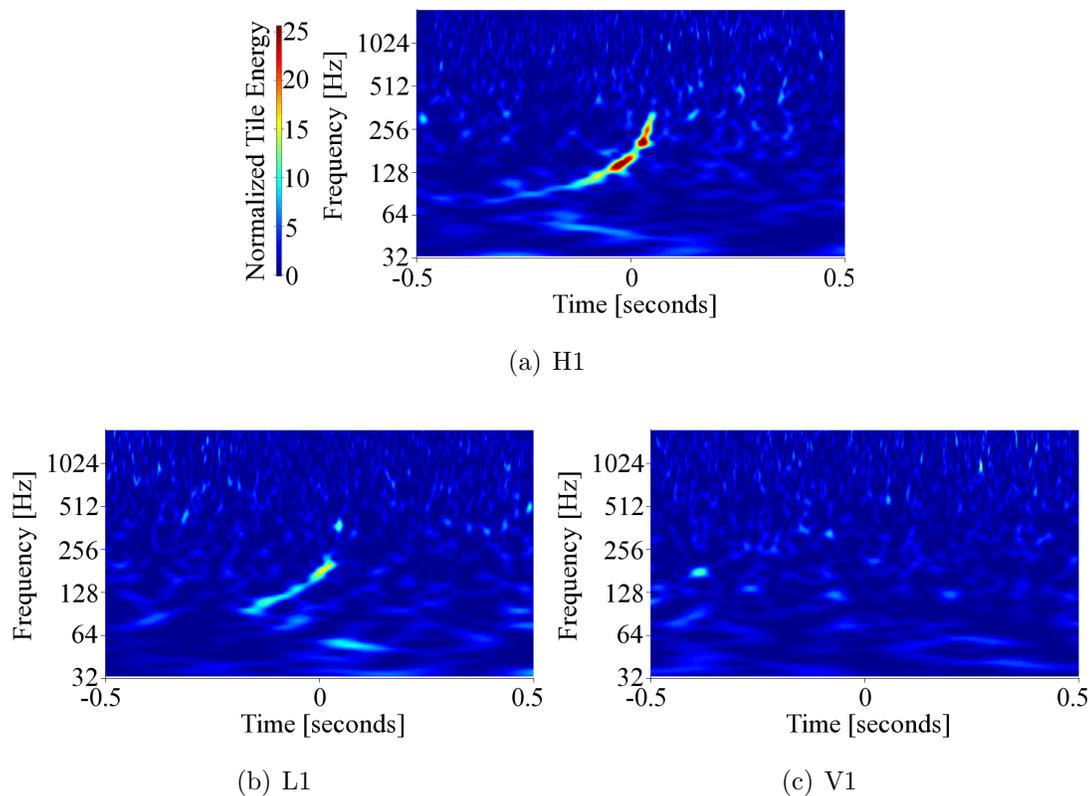
Figure 61 : Omega scans of H1, L1, and V1 at the time of the blind injection. The chirp is clearly visible in H1 and L1.
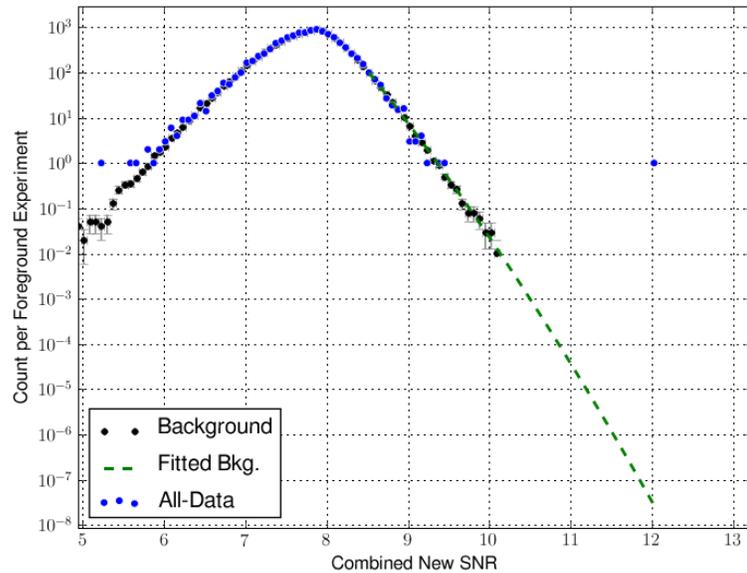
(a) The low chirp-mass bin.
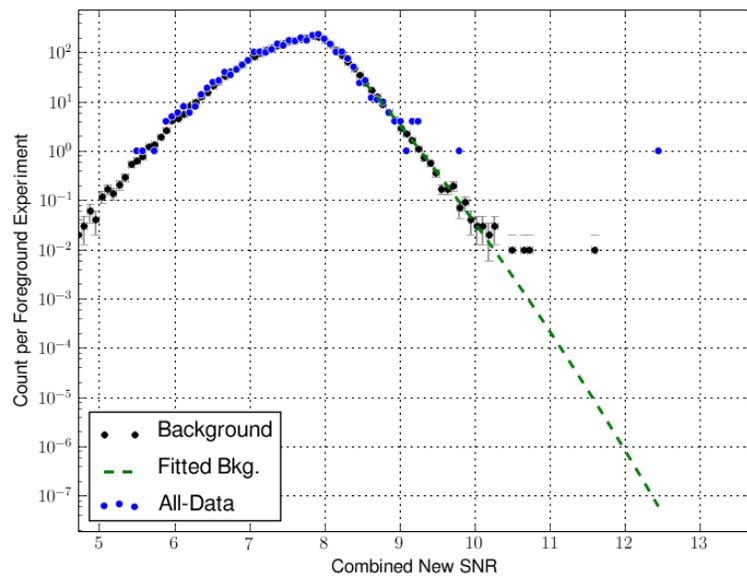
(b) The medium chirp-mass bin.

Figure 62 : Histograms of all of the H1L1-coincident triggers in S6C and D as a function of combined New SNR (ρ_{nc}) in the low and medium chirp-mass bins. Zero-lag triggers are represented as blue dots; slide triggers are black. The y axes are normalized by T_{exp}/T_f, where T_{exp} is the duration of the experiment type for a given set of points — zero-lag or slide — and T_f is the zero-lag (foreground) duration. Thus, zero-lag trigger counts are divided by 1, whereas slide-trigger counts are divided by $T_b/T_f \approx 100$, where T_b is the sum of live times in all slides (the background live time). The x-error bars on the slide triggers indicate the bin widths used in the histogram. The y-error bars were calculated by dividing the standard deviation in each bin by the square root of the effective number of slides ($= T_f/T_b \approx 100$). The green-dashed line is a Gaussian fit to the slide data using points from combined new SNR 8.5 and above. The blind injection is the blue dot all the way to the right, at $\rho_{nc} \approx 12$.
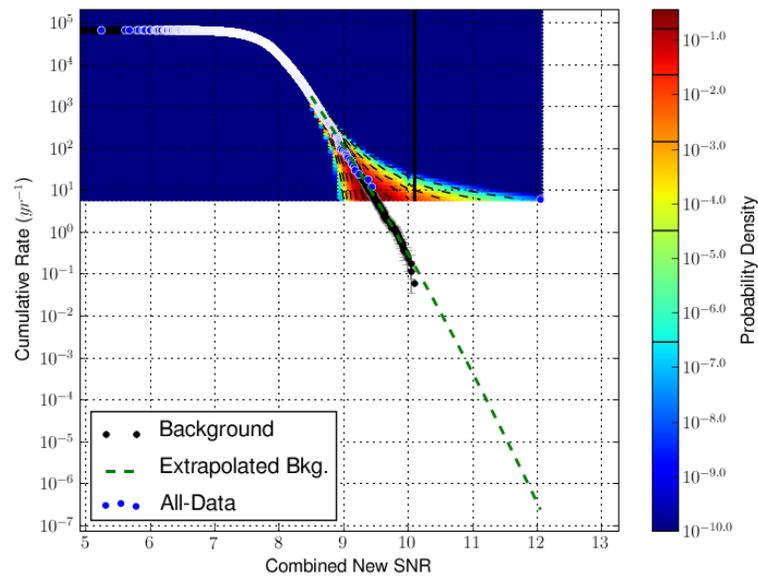
(a) The low chirp-mass bin.
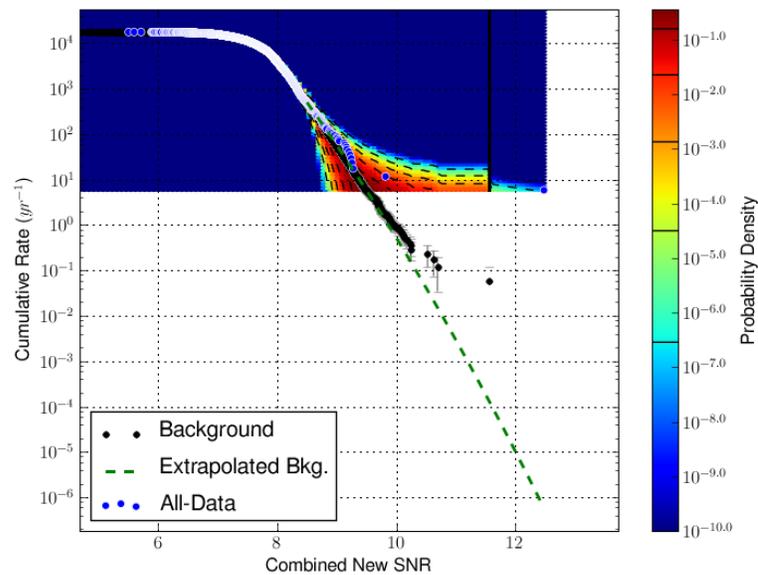
(b) The medium chirp-mass bin.

Figure 63 : Cumulative histogram of all H1L1-coincident triggers in S6C and D as a function of combined new SNR in the low and medium chirp-mass bins. Zero-lag triggers are represented by blue dots; slide triggers are black. In order to make the probability-density color map, the y axes were initially normalized by $T_{\exp}/T_{\mathrm{f}}$, as in Figure 62. All points were then divided by T_{f} to put the y axis in units of inverse years. The error bars on the slide triggers are calculated using equation 4.48 in Chapter 4. See the text for details on how the probability density was calculated. The green-dashed line shows the complimentary error function using the fit results from Figures 63.
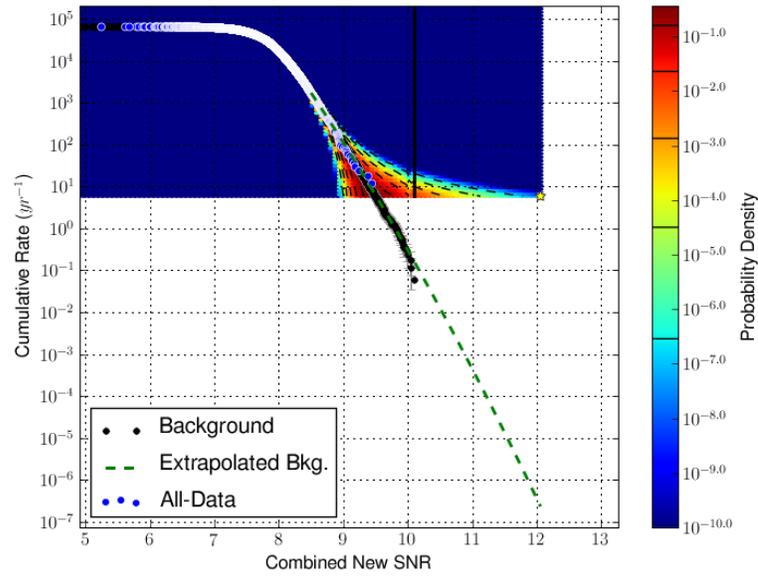
(a) The low chirp-mass bin.
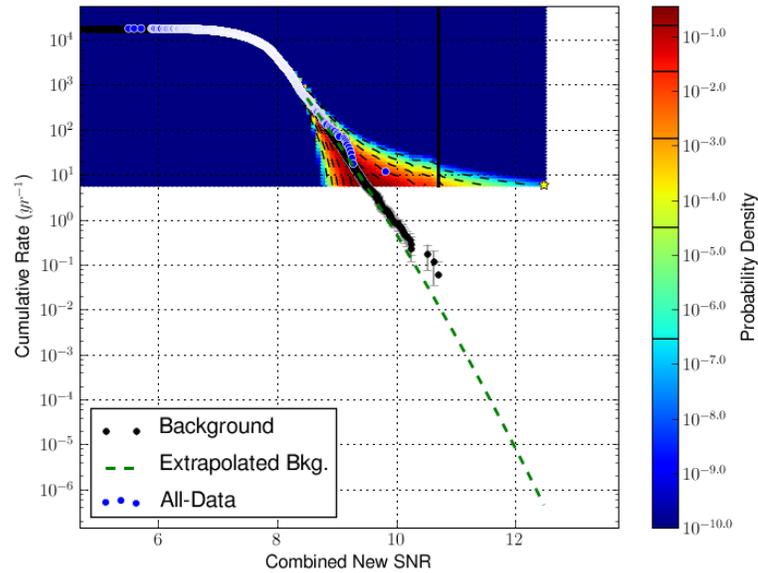
(b) The medium chirp-mass bin.

Figure 64 : Cumulative histograms with the blind injection excluded from the time-slides. These plots are generated in the same manner as in Figure 63. Here, the yellow star represents the blind injection. A Gaussian was fitted to non-cumulative slide distributions with the blind injection removed. The fit values were then used to create the extrapolations, shown as the green-dashed line.
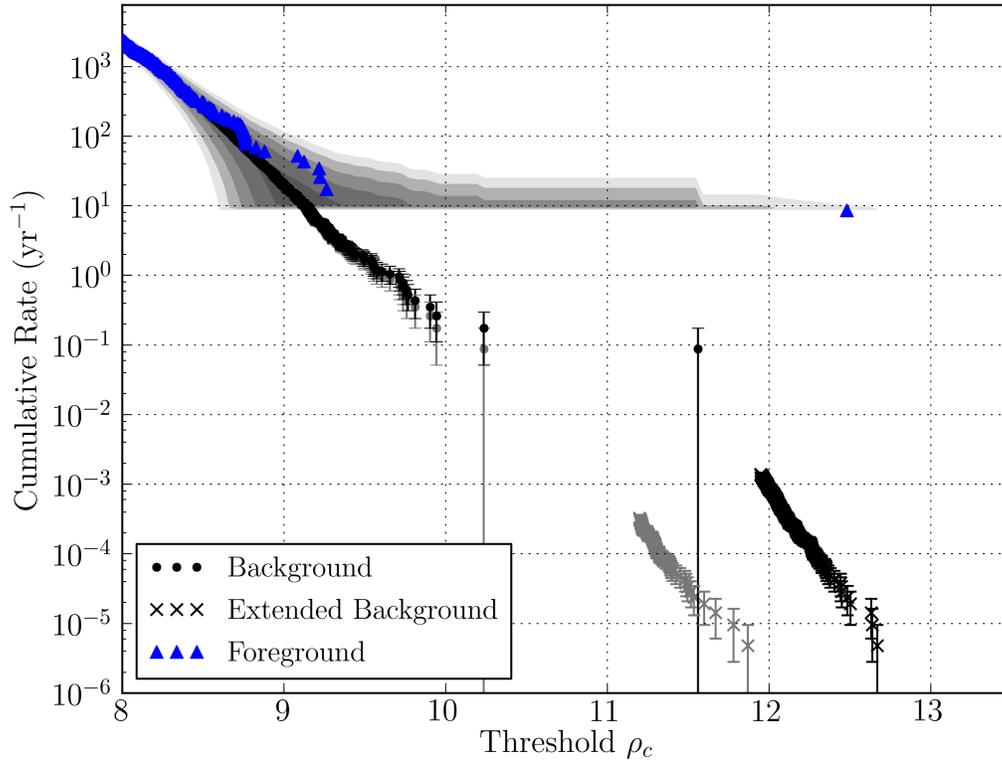
Figure 65 : The cumulative rate of H1L1-coincident triggers with chirp mass $3.48 \leq \mathcal{M}/\mathrm{M}_\odot < 7.40$ in S6D as a function of combined New SNR (here, labeled "Threshold ρ_c"). This plot was generated in a manner similar to the Figures 63 and 64, except that no extrapolation was done. We have also replaced the probability density color map by the gray shaded contours, which show the $1-5\sigma$ (dark to light) regions of the PDF. The blue triangles show coincident events. Black dots show the background estimated from 100 time shifts and black crosses show the extended background estimation from all possible 5-second shifts on this data. The gray dots and crosses show the corresponding background estimates when a ± 8 seconds of data around the time of the blind injection are excluded. Had the blind injection been real, this plot would have been one of the main figures of merit used in the detection paper.

Chapter 8

Can We Detect with Two Detectors? A Study for LIGO South

The LSC currently plans to have three LIGO detectors when Advanced LIGO comes online: one interfermeter at Livingston, L1, and two at Hanford: H1 and H2. H2 is planned to have 4 km arms, instead of the 2 km length it had in initial LIGO. However, in the past year the possibility opened up for a LIGO interferometer to be built in Australia. In this case, LIGO would still have three interferometers; rather than have two at Hanford, the optics for H2 would be shipped to the Australian site to be set up there.

8.1 Advantages of LIGO South

A number of studies were carried out to evaluate the pros and cons of having a H1L1S1 (S1 for the Australian detector) network as opposed to a H1H2L1 network. Most of these studies found that the H1L1S1 network had a number of advantages over the H1H2L1 network. Perhaps the biggest benefit is sources can be better localized if the third detector is in the southern hemisphere. Since the interferometers are not directional antennas, a network is needed to localize a source. Essentially, by measuring the difference in time of arrival between the various detectors in the network, we can

triangulate the source. The accuracy of localizing a source is given by [101, 102]:

$$p(\mathbf{r}|\mathbf{R}) \propto p(\mathbf{r}) \exp\left[-\frac{1}{2}(\mathbf{r} - \mathbf{R})^T \mathbf{M}(\mathbf{r} - \mathbf{R})\right]$$

where $\mathbf{R}$ is the actual location, $\mathbf{r}$ is the reconstructed location, $p(\mathbf{r})$ is the prior distribution on $\mathbf{r}$, and $\mathbf{M}$ is given by:

$$\mathbf{M} = \left[\sum \sigma_i^{-2}\right]^{-1} \sum_{i<j} \frac{\mathbf{D}_{ij}\mathbf{D}_{ij}^T}{\sigma_i^2 \sigma_j^2}$$

Here, σ_i is the timing accuracy in the ith detector and $\mathbf{D}_{ij}$ is the distance between the ith and j^{th} detectors. Thus, the larger the distance between any two detectors, the more strongly peaked the probability distribution will be around the actual location.

It was found [102] that in the H1L1S1 network, an optimally located[1] coalescing binary at SNR 7 could be localized to an area of 8.5 square degrees with 90% probability. In the H1H2L1 network the best localization was 12.5 square degrees. If Virgo is added to the network, then with H1L1S1V1 the localization can be as small as 2.8 square degrees with 90% probability, whereas it is 10.0 square degrees with H1H2L1V1.

Another advantage of an Australian detector, as opposed to two detectors at Hanford, is that a relative time-offset can be added between all of the detectors to perform background estimation. Recall from Chapter 6 that in S5 we could not analyze H1H2-coincident time due to correlated noise. Further, if there was some environmental factor that caused one of the Hanford detectors to come out of Science mode, it would most likely also cause the other Hanford detector to come out of Science mode. If we instead have three non-co-located detectors, then we increase the probability that at least two detectors will be operational at any given time, thereby increasing our total observation time.

8.2   Can We Detect at SNR 8 with Two Detectors?

There were a number of clear advantages to moving the second Hanford detector to Australia. However, one concern about doing this was the effect it would have on the

---

[1]In this case, "optimally located" means that the binary is overhead the plane formed by the network of detectors. The worst localization occurs when the source is in the plane of the detectors. The numbers presented are therefore best-case scenarios.

Advanced LIGO projected chronology. If the optics for the second Hanford detector were shipped to Australia, it would likely take an additional two or more years to bring the detector online. This means that we would only have two detectors for the first few years of Advanced LIGO.[2] Our projected detection rates are based on being able to detect gravitational waves from CBCs at an SNR of 8 in each detector. We know that non-transient Gaussian noise, or glitches, can affect our ability to detect, but this can be mitigated by performing coincidence tests across detectors [34]. If we only have two detectors, can we confidently detect a GW at SNR 8?

To answer this question we looked at the rates of H1L1 coincidences in S5. We used S5 as opposed to S6 because S6 the S5 analysis was complete and triggers were available at the time of this study. All of the months from the 12–18 month analysis were used, along with four months from the S5-LV analysis. All H1L1 coincidences were used in double-, triple- and (in the case of the LV months) quadruple- coincident times. For H1H2L1, H1L1V1, and H1H2L1V1 coincidences, the H1L1 part of the coincidence was used to re-calculate the combined New SNR of the trigger.

Figure 8.3 shows the result of the analysis for the low chirp-mass bin ($\mathcal{M} <$ 3.48). Plotted is the cumulative rate of coincident zero-lag ("Foreground") and slide ("Background") triggers as a function of combined New SNR. The slide triggers were generated from the standard 100-slide `ihope` analysis. The y-axis was generated in the same manner as in the cumulative-rate plots in Figures 63 – 65 of Chapter 7. Indeed, the plots presented here were a predecessor to the cumulative-rate plots shown in Chapter 7. The dashed-red line shows the combined new SNR of a signal with SNR 8 in each detector (= 11.3). Also plotted, as a yellow star, is the combined new SNR of a blind CBC injection that was performed during S5.[3]

It is evident from the plot that an event with a new SNR of 8 in each detector would stand out above the background. In this case, we would place an upper-limit on the FAR as being < 1 in 50 years. We can arrive at the same conclusion from looking

---

[2]There is also the Virgo detector, but as Virgo is maintained and funded by a different collaboration, they may not have the same schedule as LIGO. Thus, in this study we wanted to focus on what could be accomplished using only the LIGO detectors.

[3]Note: this is *not* the blind injection that was performed during S6. At the time of this study, the S6 blind injection had not happened yet. The S5 injection, which occurred during S5-LV analysis, was missed because it was performed when an earthquake was passing Livingston, resulting in L1 being CAT3 vetoed during the time. For more details on this blind injection, see [6].

at Figure 64(a) of Chapter 7. (Recall that these plots were also created using only H1L1-coincident triggers.) There we can use the extrapolation to estimate what the FAR would be for an signal with combined new SNR of 11.3. In the low chirp-mass bin we would have a FAR $\approx$ 1 in 10^4 years.

8.2.1 Deeper Study

There are a few important features to this analysis. First, note that we have shown results after CAT3 vetoes have been applied. Figure 67(a) shows the same plot as in Figure 8.3 after CAT2 vetoes have been applied. In this plot we can see that an event at SNR 8 in each detector *would not* stand out above the background; it would only have a FAR of $\sim$ 1 in 3 years. This highlights the importance of DQ investigations. Without good data quality, we cannot detect at SNR 8 using two detectors.

Second, note that we have plotted results as a function of combined *New* SNR. Figure 67(b) shows the results after CAT3 as a function of combined SNR. The large tail due to non-Gaussian transients has buried the SNR 8/8 line in background; such an event would have a FAR of $\sim$ 30 per year if combined SNR was our ranking statistic. Clearly, without new SNR we could not detect at SNR 8 (or 40, for that matter). However, when talking about real signals, we have interchanged New SNR and SNR: we quote rates based on a *SNR* of 8 in each detector, yet we have placed the red-dashed line at a *New SNR* of 8 in each detector. The reason we can do this is New SNR reduces to SNR for (well-modelled) GWs. Figure 8.3 shows a plot of combined new SNR versus combined SNR for (non-spinning) injections. The vertical dashed line shows the point at which combined $\rho = 11.3$. At this point, we can see that most of the injections have a combined New SNR > 11. Note that some of the lower points are due to glitches being mistakenly mapped to injections; cf. Figure 8.3. Thus to detect with two detectors at the quoted rates, we need New SNR, which in-turn requires that we have templates that match the GW signal well.

Finally, note that we have so far only shown plots for the low chirp-mass bin. The background does not fall off as sharply in the higher chirp-mass bins. This can be seen by comparing Figure 64(a) to Figure 64(b): the medium chirp-mass bin has a higher tail, leading to larger FARs at the same New SNR. However, as can be seen in the extended background with the blind injection removed (the gray crosses) in Figure 65, we would still get a FAR of 1 in 3500 years in the medium chirp-mass bin at a

combined New SNR of 11.3. Though higher than the low chirp-mass bin, this FAR is still low enough to claim evidence for a detection. Further, we are most concerned with the low chirp-mass bin, as this contains BNS systems. As discussed in section 2.3 the rate of BNS coalescence in the universe is the best determined, as it is based on astronomical observations of BNS systems [10]. We are therefore most confident in quoted expected detection rates for BNS coalescence; it is for these systems that we wish to ensure that we can detect at SNR 8.

8.3   Conclusions

This study showed that building LIGO South would not hamper our abilities to detect gravitational waves at the quoted rates in the near term. Thus the decision of whether or not to build LIGO South can be based solely on practical aspects, such as whether Australia can commit the funds needed to build the detector. Regardless of whether or not LIGO South is built, this study proved useful, as it gave us confidence that we can detect GWs at expected rates in Advanced LIGO. Admittedly, we cannot be certain that the character of the noise will be the same in Advanced LIGO. However, by comparing the S5 cumulative rate plot in this chapter to the corresponding S6 plots in Chapter 7, we see that we obtained largely the same (better, even) background distribution in S6, despite a number of hardware differences between the runs. Thus, by making using of New SNR, and with equally successful DQ studies in Advanced LIGO, we can expect to detect at a SNR of 8 in each detector.
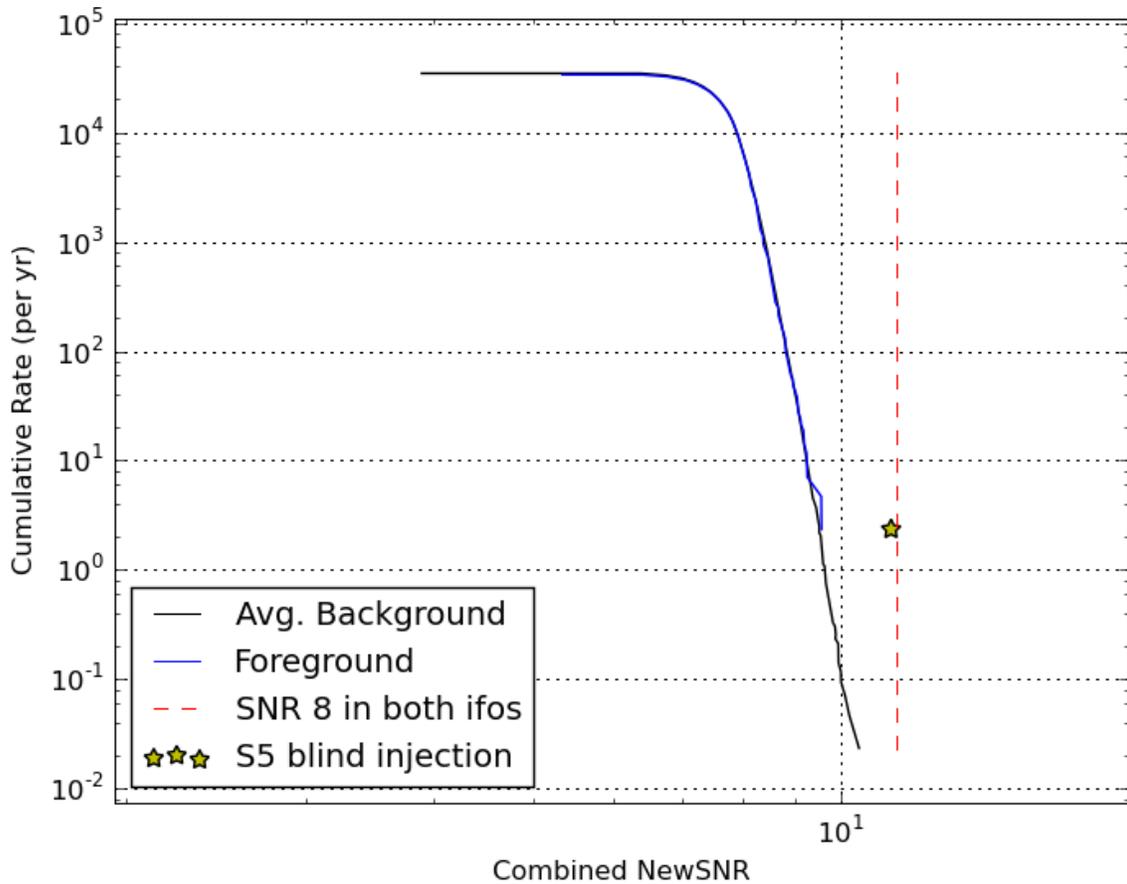
Figure 66 : Cumulative rate of H1L1 zero-lag ("Foreground") and slide ("Background") coincidences in S5 after CAT3 (cumulative) vetoes have been applied. Only triggers in the low chirp-mass bin ($\mathcal{M} < 3.48\,\mathrm{M}_\odot$) are shown. Data used is from all of the S5 12-18 months, plus four months from the S5-LV analysis. The black line indicates the background; the blue line indicates the foreground. As with the cumulative rate plots in Chapter 7, the foreground rates are computed by dividing the cumulative count of foreground triggers by the total zero-lag live time (T_{f}). The background rates are computed by dividing the cumulative count of background triggers by total slide live time (T_{b}), which is equivalent to dividing by the effective number of slides ($= T_b/T_f$), then dividing by the zero-lag live time. Hence the black line is labelled as the "Average Background." The red-dashed line indicates the combined new SNR for an event with single-detector new SNR of 8 in each detector. The yellow star shows the combined new SNR of a blind injection that was made during the S5-LV analysis.
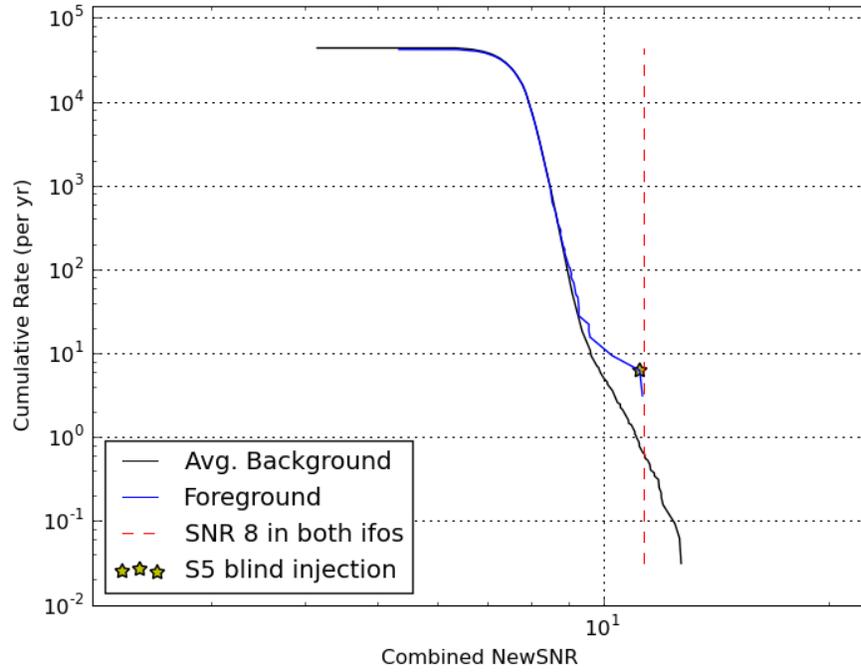
(a) After CAT2 vetoes applied.
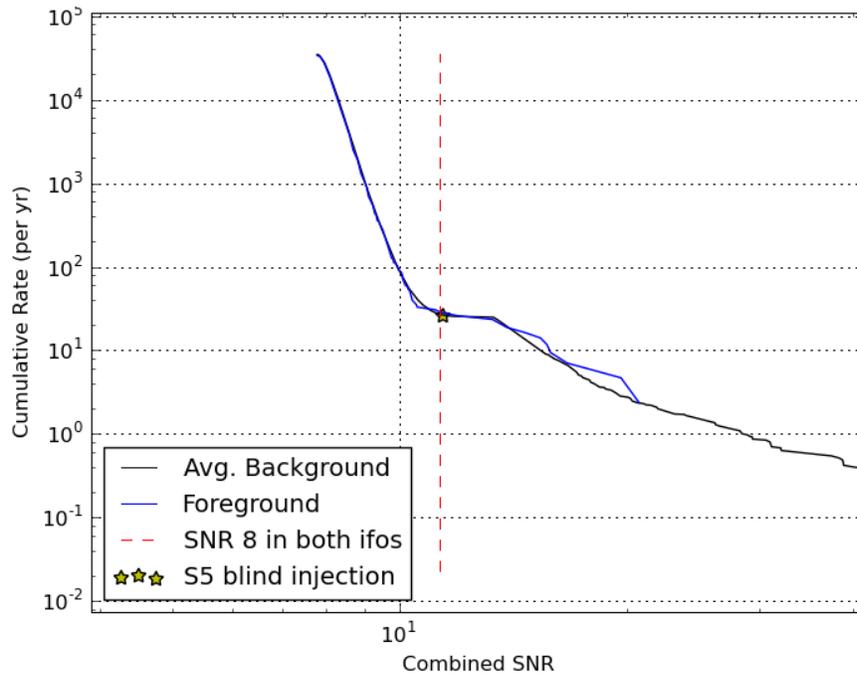
(b) Cumulative rate vs. combined SNR, after CAT3 vetoes applied.

Figure 67 : Cumulative rate plot for the same period of time shown in Figure 8.3, after CAT2 vetoes have been applied (top) and as a function of combined SNR (bottom). In the bottom plot, CAT3 vetoes have been applied.
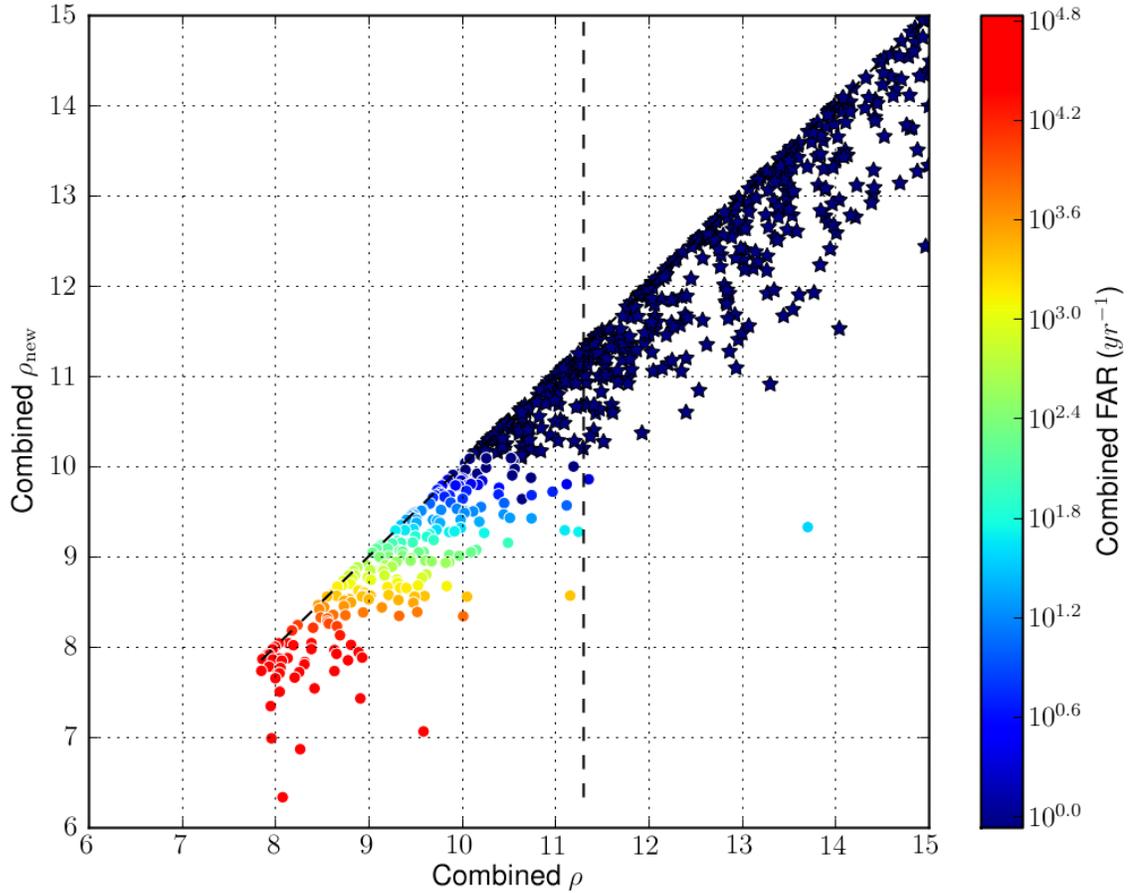
Figure 68 : Combined New SNR versus combined SNR for non-spinning injections. All injections were found as H1L1 coincidences. Results taken from 6 weeks of S6D data. The diagonal black-dashed line indicates $y = x$; the vertical dashed line shows the point where combined SNR = 11.3. This plot was created by `ligolw_cbc_plotfm`. As with all PlotFM plots, the blue stars indicate injections found with 0 FAR in their analysis period and colored circles indicate injections with non-zero FAR.
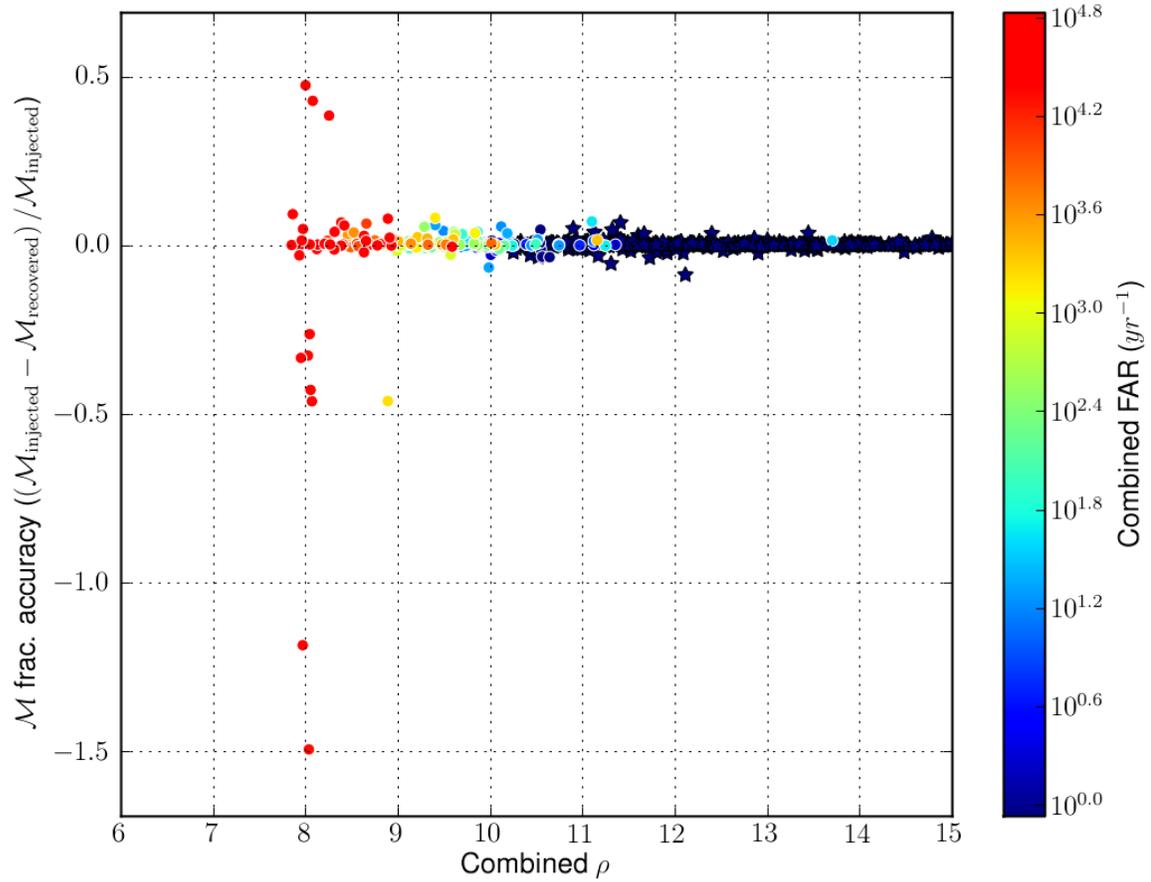
Figure 69 : Recovered chirp mass fractional accuracy of the injections shown in Figure 8.3 as a function of combined SNR. Triggers close to zero were most likely caused by injections. Triggers further from the zero line are most likely glitches that happened to be within $\pm 1\,\mathrm{s}$ of the injection.

Chapter 9

Future Developments and Conclusions

We have stepped through, in detail, the low-mass CBC pipeline, and we have described how this was used to obtain search results from CBCs. We now conclude by briefly presenting some future developements for the pipeline.

9.1   Gating

In chapter 7 we saw the effects a loud transient — the spike glitch — had on the matched filter. We dealt with the spike glitch by adding a category 3 veto around the time of the glitch. Due to the ringing of the filter we had to add a $\pm 8\,\mathrm{s}$ padding to the glitch. However, as can be seen in figure 7.3.2, triggers can be last for several tens of seconds after the glitch; this duration increases with the size of the glitch.

Another possibility is to remove the glitch *prior* to match filtering. We can do this by applying a Tukey window to the data stream to ramp the output down to 0 at the time of the glitch. We call this "gating." Figure 70 shows a time series in which a spike-glitch occurred with a gate applied. In 70(b) the window that is applied can be seen. We must ramp the data down (in this case, by using the first half cycle of a cosine) so as not to introduce ringing at the point the window turns on, and we must do the same after the glitch to ramp the data back up. In this manner we can cleanly remove the glitch from the data: compare the Omega scans in Figure 71 with the glitch in the data to the scan with the glitch gated out.

Zeroing out the data around the glitch does mean that we will slightly underestimate the PSD. However, if the gated data is only a few seconds out of an entire analysis block, the effect should be small. It may also be possible to correct for the zeroed out data by taking the average power across the segment, then add a correction factor. Even if we add a correction factor to the PSD, we should probably limit the amount of time that is gated in an analysis block. The upper limit on how much time can gated in a block requires more study.

Knowing what to gate also needs study. In the case of the spike glitch, we could not find the cause, as we could not correlate it with any auxiliary environmental or instrumental channels. We got around this difficulty by applying the SNR > 250 flag at CAT3, which allowed us to check the results at CAT2 to ensure we did not accidentally veto a loud GW trigger. Since gating involves removing the glitch before filtering, we would not have that luxury, unless we were to match filter the data multiple times. Another possibility is to establish a SNR for which un-modelled "Burst" searches do as-good-as CBC searches at finding GW signals. We would then remove any trigger that exceeded that threshold, leaving it to the Burst, or a mixed Burst-CBC search to find loud triggers. Finding that threshold, and determining the safety of such a scheme, is yet to be done.

9.2   Single Stage Pipeline and Updates to Pipedown

Using a two stage-pipeline, as presented in chapter 5, makes it difficult to follow-up triggers, and to do things such as perform more slides, as discussed in chapter 7. `HIPE` was developed to have two stages in order to save on the computational cost of calculating χ^2. This was done several years ago. Since then, advancements in computing speed have made it possible to use a *single-stage* pipeline. In this, χ^2 is calculated at first inspiral, and so no second stage is needed. Having such a pipeline would make it far simpler to follow-up triggers and to know how they evolve through the pipeline.

As mentioned in 7 a prototype single-stage pipeline already exists. Some updates need to be made to Pipedown to be able to use this pipeline, however, as some of the programs rely on conventions of the current `HIPE` pipeline. Of note: in the singe-stage pipeline, `lalapps_thinca` is replaced by an updated version, `ligolw_thinca`, which
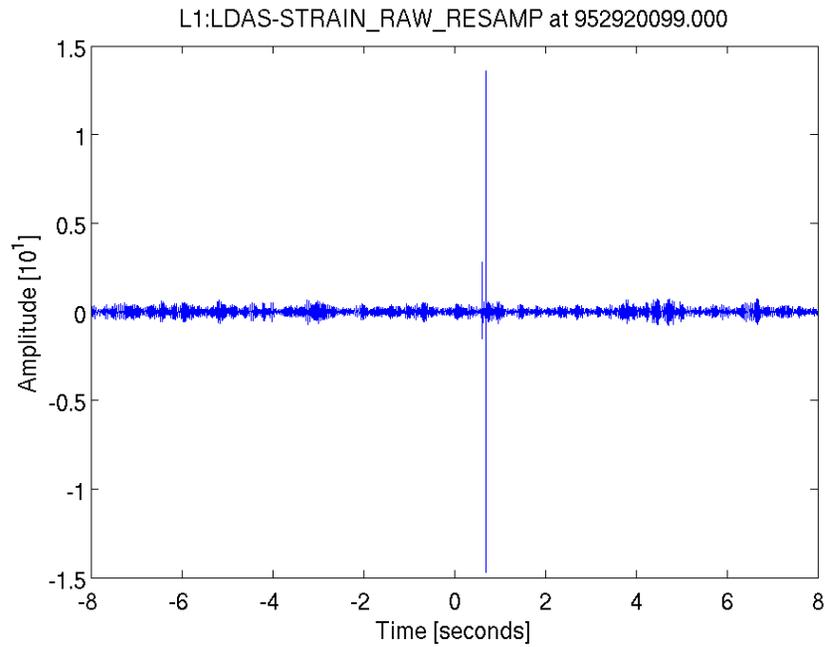
(a) Raw time series.
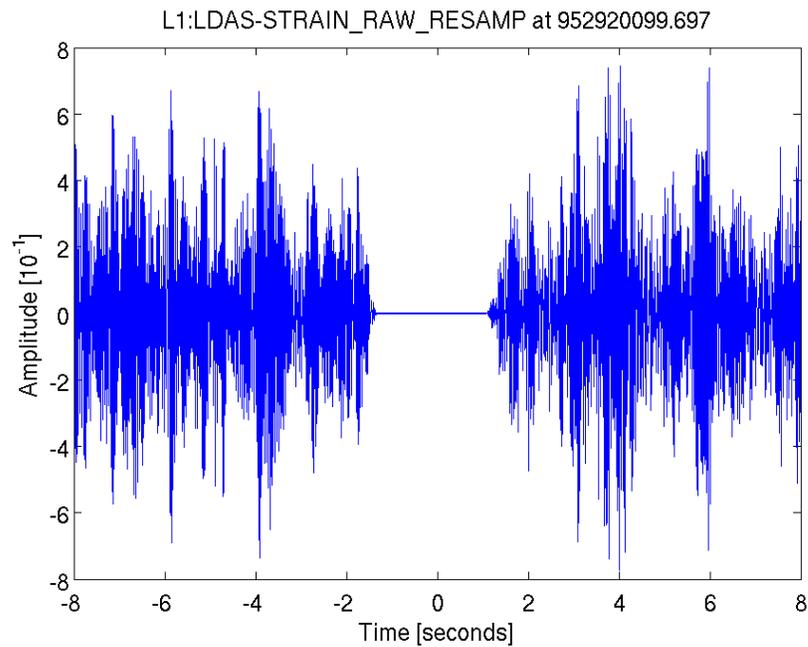
(b) Gated time series.

Figure 70 : Raw and gated time series of a L1 spike glitch that occurred during S6. In the raw time series (top) we can see that this was actually two glitches in quick succession. For this reason, the gate was chosen to be $\pm 1.5s$ around the larger glitch. The effect of the ramping down of the Tukey window can be seen in the gated time-series.
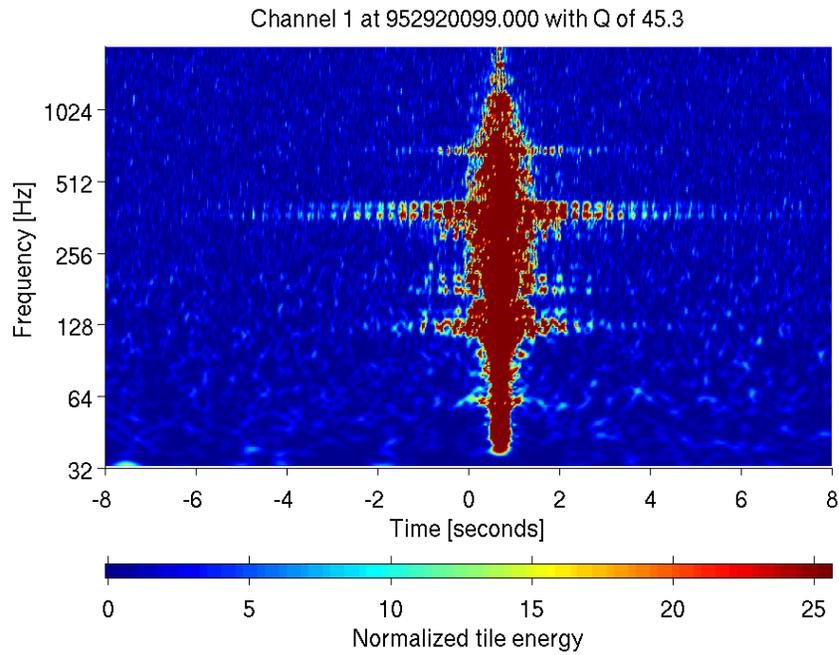
(a) Scan with glitch in.
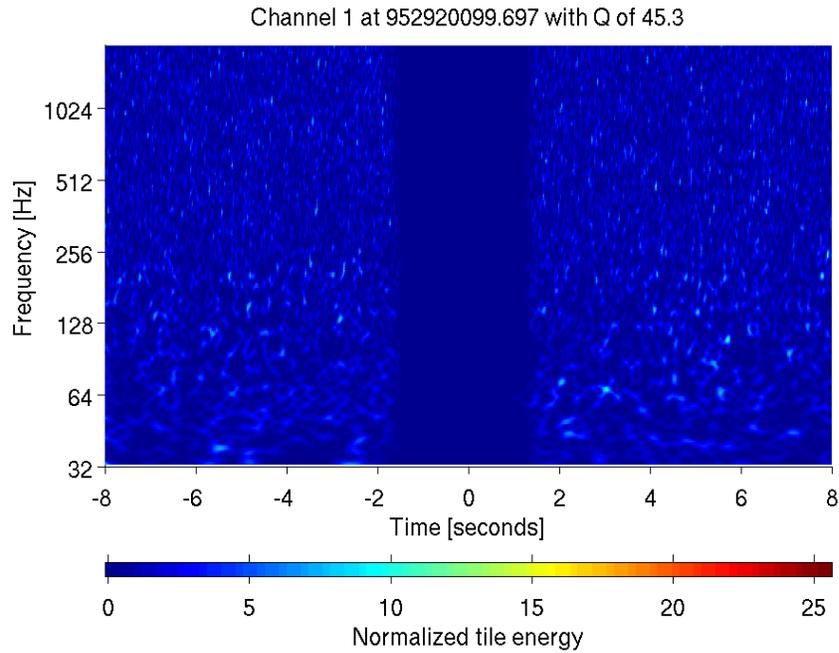
(b) Scan with gate.

Figure 71 : Omega scan of the spike glitch shown in figure 70. The top plot shows the scan with the glitch in the data; the bottom plot shows the scan with the glitch removed. Omega applies a whitening filter when these scans are generated. The "ringing" of the filter is evident in the top plot. With the glitch gated, we see that there is no ringing. When used in a CBC search, this implies that we would have neither the central tower of triggers, nor the "shoulder" and "tail" seen in Figure 7.3.2.

natively makes use of the Coinc and Experiment tables discussed in chapter 5. It also performs linear slides as opposed to slides on a ring. This removes the need for `ligolw_thinca_to_coinc` in Pipedown.

A few other improvements are also planned for Pipedown. As mentioned in Chapter 5, multiple veto categories can be stored in a single database. Pipedown currently does not take advantage of this. As part of the switch to a single-stage pipeline, we plan to add a program to Pipedown to apply vetoes in the database, so as not to have to run `ligolw_thinca` multiple times. Also, a number of programs in Pipedown cannot read data from multiple databases. This makes it difficult to combine results from multiple `ihope` analysis periods. Reading from multiple databases will require a slight update to the `experiment` table. Namely, we plan to replace the `gps_start` and `gps_stop` columns with a `segment_def_id` column. This would point to a list of segments in the `segment` and `segment_definer` tables that give the analyzed time covered by the database.

Pipedown can also be greatly simplified. Currently, it has to create intermediate databases that separately store injection data from each of the runs. These `RAW` databases lack much information, such as the vetoes applied and the live time. This is done so as to limit the size of the `LIGO_LW` XML files that need to be extracted for `lalapps_inspinjfind`. Since clustering is done on these intermediate databases it makes it difficult to test new statistics. If one wants to cluster with the new statistic, they must re-run clustering on all of the individual `RAW` databases, carry out injection-finding, create a new final database, and re-compute the live time. If `lalapps_inspinjfind` could read databases instead, all of the injection runs could be added to the `FULL_DATA RAW` database at the outset, greatly simplifying the pipeline and the ability to test new statistics. Work on this is planned. Along with this update to the injection-finding program, we plan to implement more sophisticated injection- finding techniques, such as using e-thinca windows. This will decrease the chance that a glitch that occurs near the time of an injection will be mapped to the injection, increasing confidence in ROC plots.

9.3 Conclusion

The data-analysis methods, pipeline, and search results presented in this thesis are the result of years of work by a number of people throughout the LSC and the Virgo Collaboration. Although no gravitational waves have been detected as of yet, injection studies, comparisons of our rate-upper limits to astrophysical predictions, and the result of the S6 blind injection challenge have made us confident that we will be able to detect when Advanced LIGO and Virgo come online. The work to build Advanced LIGO is currently underway. When it is finished it will present new challenges to our data-analysis pipeline, some of which are outlined here. In the coming years we will continue to improve and refine our analysis techniques to meet those challenges, so that, when a CBC gravitational-wave signal happens, we will detect it.

Bibliography

[1] B. S. Sathyaprakash and Bernard F. Schutz. Physics, astrophysics and cosmology with gravitational waves. *Living Rev. Rel.*, 12(2), 2009.

[2] B. Abbott et al. LIGO: The Laser Interferometer Gravitational-Wave Observatory. *Rept. Prog. Phys.*, 72:076901, 2009.

[3] R. Adhikari, P. Fritschel, and S. Waldman. Enhanced ligo. Technical Report LIGO-T060156-v1, LIGO Project, 2006.

[4] F Acernese et al. Virgo status. *Classical and Quantum Gravity*, 25(18):184001, 2008.

[5] J. Abadie et al. All-sky search for gravitational-wave bursts in the first joint LIGO-GEO-Virgo run. *Phys. Rev. D*, 81:102001, 2010.

[6] J. Abadie et al. Search for compact binary coalescences in ligo and virgo data from s5 and vsr1. *Phys. Rev. D*, 82:102001, 2010.

[7] LIGO Scientific Collaboration and Virgo Collaboration. Search for gravitational-wave bursts associated with gamma-ray bursts using data from ligo science run 5 and virgo science run 1. *Astrophys. J.*, 715:1438–1452, 2010.

[8] Gregory M Harry and the LIGO Scientific Collaboration. Advanced ligo: the next generation of gravitational wave detectors. *Classical and Quantum Gravity*, 27(8):084006, 2010.

[9] Advanced Virgo Baseline Design. `https://pub3.ego-gw.it/itf/tds/file.php?callFile=VIR-0027A-09.pdf`.

[10] J. Abadie et al. Predictions for the rates of compact binary coalescences observable by ground-based gravitational-wave detectors. *Class. Quantum Grav.*, 27:173001, March 2010.

[11] K. Belczynski, V. Kalogera, and T. Bulik. A comprehensive study of binary compact objects as gravitational wave sources: Evolutionary channels, rates, and physical properties. *Astrophysical J.*, 572:407, 2002.

[12] Bruce Allen, Warren G. Anderson, Patrick R. Brady, Duncan A. Brown, and Jolien D. E. Creighton. FINDCHIRP: An Algorithm for Detection of Gravitational Waves from Inspiraling Compact Binaries. 2005.

[13] Warren G. Anderson, Patrick R. Brady, Jolien D. E. Creighton, and Eanna E. Flanagan. An excess power statistic for detection of burst sources of gravitational radiation. *Phys. Rev. D*, 63:042003, 2001.

[14] B. P. Abbot et al. Searches for gravitational waves from known pulasrs with science run 5 ligo data. *ApJ*, 713:671, 2010.

[15] The LIGO Scientific Collaboration and the Virgo Collaboration. An upper limit on the stochastic gravitational-wave background of cosmological origin. *Nature*, 460:990, 2009.

[16] B. Abbott et al. Analysis of LIGO data for gravitational waves from binary neutron stars. *Phys. Rev. D*, 69:122001, 2004.

[17] B. Abbott et al. Search for gravitational waves from galactic and extra-galactic binary neutron stars. *Phys. Rev. D*, 72:082001, 2005.

[18] B. Abbott et al. Search for gravitational waves from primordial black hole binary coalescences in the galactic halo. *Phys. Rev. D*, 72:082002, 2005.

[19] B. Abbott et al. Search for gravitational waves from binary black hole inspirals in LIGO data. *Phys. Rev. D*, 73:062001, 2006.

[20] B. Abbott et al. Search for gravitational waves from binary inspirals in S3 and S4 LIGO data. *Phys. Rev. D*, 77:062002, 2008.

[21] B. Abbott et al. Search for Gravitational Waves from Low Mass Binary Coalescences in the First Year of LIGO's S5 Data. *Phys. Rev. D*, 79:122001, 2009.

[22] B. Abbott et al. Search for Gravitational Waves from Low Mass Compact Binary Coalescence in 186 Days of LIGO's fifth Science Run. *Phys. Rev. D*, 80:047101, 2009.

[23] Luc Blanchet, Bala R. Iyer, Clifford M. Will, and Alan G. Wiseman. Gravitational wave forms from inspiralling compact binaries to second-post-Newtonian order. 13:575, 1996.

[24] Serge Droz, Daniel J. Knapp, Eric Poisson, and Benjamin J. Owen. Gravitational waves from inspiraling compact binaries: Validity of the stationary-phase approximation to the fourier transform. *Phys. Rev. D*, 59:124016, 1999.

[25] Luc Blanchet. Gravitational radiation from post-Newtonian sources and inspiralling compact binaries. *Living Rev. Rel.*, 5:3, 2002.

[26] Alessandra Buonanno, Gregory B. Cook, and Frans Pretorius. Inspiral, merger and ring-down of equal-mass black-hole binaries. *Phys. Rev. D*, 75:124018, 2007.

[27] Michael Boyle et al. High-accuracy comparison of numerical relativity simulations with post-Newtonian expansions. *Phys. Rev. D*, 76:124038, 2007.

[28] Mark Hannam et al. Where post-Newtonian and numerical-relativity waveforms meet. *Phys. Rev. D*, 77:044020, 2008.

[29] Yi Pan et al. Data-analysis driven comparison of analytic and numerical coalescing binary waveforms: Nonspinning case. *Phys. Rev. D*, 77:024014, 2008.

[30] Michael Boyle, Duncan A. Brown, and Larne Pekowsky. Comparison of high-accuracy numerical simulations of black-hole binaries with stationary phase post-Newtonian template waveforms for Initial and Advanced LIGO. *Class. Quantum Grav.*, 26:114006, 2009.

[31] Duncan A. Brown. *Searching for Gravitational Radiation from Binary Black Hole MACHOs in the Galactic Halo*. PhD thesis, University of Wisconsin-Milwaukee, December 2004.

[32] Bruce Allen. A χ^2 time-frequency discriminator for gravitational wave detection. *Phys. Rev. D*, 71:062001, 2005.

[33] Andres Rodríguez. Reducing false alarms in searches for gravitational waves from coalescing binary systems. Master's thesis, Louisiana State University, 2007.

[34] C. A. K. Robinson, B. S. Sathyaprakash, and Anand S. Sengupta. Geometric algorithm for efficient coincident detection of gravitational waves. *Phys. Rev. D*, 78(6):062002, 2008.

[35] Charles W. Misner, Kip S. Thorne, and John Archibald Wheeler. *Gravitation*. Freeman, New York, New York, 1973.

[36] Bernard F. Schutz. *A First Course in General Relativity*. Cambridge University Press, Cambridge, 1985.

[37] R. D. Blandford and K. S. Thorne. Online lecture notes, version 06*.1.k.

[38] Sean Carroll. *Spacetime and Geometry*. Addison Wesley, San Francisco, 2004.

[39] Luc Blanchet. Gravitational radiation from post-Newtonian sources and inspiralling compact binaries. *Living Rev. Rel.*, 9:3, 2006.

[40] Peter R. Saulson. *Fundamentals of Interferometric Gravitational Wave Detectors*. World Scientific, Singapore, 1994.

[41] J. Abadie et al. Calibration of the ligo gravitational wave detectors in the fifth science run. *Nuclear Instruments and Methods in Physics Research*, A624:223–240, 2010.

[42] I. Bartos et al. Calibration for Big Dog; LIGO-T1000580-00-R. 2011.

[43] T. Accadia et al. Calibration and sensitivity of the Virgo detector during its second science run. *Class. Quantum Grav.*, 28:025005, 2011.

[44] R. A. Hulse and J. H. Taylor. Discovery of a pulsar in a binary system. *Astroph. J. Lett.*, 195:L51–L53, January 1975.

[45] J. H. Taylor and J. M. Weisberg. A new test of general relativity : gravitational radiation and the binary pulsar psr 1913 +16. 253:908, 1982.

[46] Patrick R. Brady and Stephen Fairhurst. Interpreting the results of searches for gravitational waves from coalescing binaries. *Class. Quantum Grav.*, 25(10):105002, 2008.

[47] P. R. Brady, J. D. E. Creighton, and A. G. Wiseman. Upper limits on gravitational-wave signals based on loudest events. *Class. Quantum Grav.*, 21:S1775–S1781, 2004.

[48] Rahul Biswas, Patrick R. Brady, Jolien D. E. Creighton, and Stephen Fairhurst. The Loudest Event Statistic: General Formulation, Properties and Applications. *Class. Quantum Grav.*, 26:175009, 2009.

[49] L. S. Finn. Detection, measurement, and gravitational radiation. *Physical Review D*, 46(12):5236–5249, December 1992.

[50] L. S. Finn and D. F. Chernoff. Observing binary inspiral in gravitational radiation: One interferometer. *Phys. Rev. D*, 47:2198–2219, March 1993.

[51] L A Wainstein and V D Zubakov. *Extraction of signals from noise.* Prentice-Hall, Englewood Cliffs, NJ, 1962.

[52] D. S. Sivia and J. Skilling. *Data Analysis: A Bayesian Tutorial.* Oxford University Press, Oxford, 2nd edition, 2006.

[53] Eric W. Weisstein. Modified bessel function of the first kind. From *MathWorld–A Wolfram Web Resource.*

[54] Benjamin J. Owen. Search templates for gravitational waves from inspiraling binaries: Choice of template spacing. *Phys. Rev. D*, 53:6749–6761, 1996.

[55] Benjamin J. Owen and B. S. Sathyaprakash. Matched filtering of gravitational waves from inspiraling compact binaries: Computational cost and template placement. *Phys. Rev. D*, 60:022002, 1999.

[56] T. Tanaka and H. Tagoshi. Use of new coordinates for the template space in a hierarchical search for gravitational waves from inspiraling binaries. *Phys. Rev. D*, 62(8):082001–+, October 2000.

[57] Stas Babak, Balasubramanian, David Churches, Thomas Cokelaer, and B.S. Sathyaprakash. A template bank to search for gravitational waves from inspiralling compact binaries i: physical models. *Class. Quantum Grav.*, 23:5477–5504, 2006.

[58] Thomas Cokelaer. Gravitational wave from inspiralling compact binaries: hexagonal template placement and its efficiency in detecting physical signals. *Phys. Rev. D*, 76:102004, 2007.

[59] R. N. Manchester, G. B. Hobbs, A. Teoh, and M. Hobbs. The australia telescope national facility pulsar catalogue. *Astronom. J.*, 129(4):1993–2006, 2005.

[60] Theocharis A. Apostolatos, Curt Cutler, Gerald J. Sussman, and Kip S. Thorne. Spin-induced orbital precession and its modulation of the gravitational waveforms from merging binaries. *Phys. Rev. D*, 49:6274, 1994.

[61] J. Abadie et al. Search for Gravitational Waves from Low Mass Compact Binary Coalescence in LIGO's Sixth Science Run and Virgo's Science Runs 2 and 3. *In preparation for Submission to PRD*, 2011.

[62] Drew Keppel. *Signatures and dynamics of compact binary coalescences and a search in LIGO's S5 data*. PhD thesis, Caltech, Pasadena, CA, 2009.

[63] Duncan A. Brown. Using the inspiral program to search for gravitational waves from low-mass binary inspiral. *Classical and Quantum Gravity*, 22:S1097, 2005.

[64] John R. Taylor. *An Introduction to Error Analysis*. University Science Books, Sausalito, CA, 2nd edition, 1997.

[65] Edward Harrison. *Darkness at Night: A Riddle of the Universe*. Havard University Press, Cambridge, MA, 1987.

[66] Todd Tannenbaum, Derek Wright, Karen Miller, and Miron Livny. Condor – a distributed job scheduler. In Thomas Sterling, editor, *Beowulf Cluster Computing with Linux*. MIT Press, October 2001.

[67] J. Slutsky et al. Methods for Reducing False Alarms in Searches for Compact Binary Coalescences in LIGO Data. *Class. Quant. Grav.*, 27:165023, 2010.

[68] N. Christensen (for the LIGO Scientific Collaboration and the Virgo Collaboration). Ligo s6 detector characterization studies. *Classical and Quantum Gravity*, 27:194010, 2010.

[69] M. Ito. glitchmon: A DMT monitor to look for transient signals in selected channels. Developed using the LIGO Data Monitoring Tool (DMT) library.

[70] Duncan Brown et al. S6 segment database infrastructure proposal. *Technical document LIGO-T0900005-00-Z*, 2010.

[71] Roy Williams. Ldas lightweight data format specification. Technical Report T990023-01-E, LIGO Scientific Collaboration, 2005.

[72] `https://confluence.pegasus.isi.edu/display/pegasus/Home`, July 2010.

[73] Clifford M. Will and Alan G. Wiseman. Gravitational radiation from compact binary systems: Gravitational waveforms and energy loss to second post-Newtonian order. *Phys. Rev. D*, 54:4813–4848, 1996.

[74] C Cutler and E Flanagan. Gravitational waves from merging compact binaries: How accurately can one extract the binary's parameters from the inspiral waveform? *Phys. Rev. D*, 49:2658, 1994.

[75] Alessandra Buonanno, Yi Pan, Harald P. Pfeiffer, Mark A. Scheel, Luisa T. Buchman, and Lawrence E. Kidder. Effective-one-body waveforms calibrated to numerical relativity simulations: Coalescence of nonspinning, equal-mass black holes. *Phys. Rev. D*, 79:124028, 2009.

[76] A. S. Sengupta, J. A. Gupchup, and C. A. K. Robinson. Trigscan: Using parameter correlations to cluster triggers in binary inspiral gravitational wave search pipeline. 2008. In Preparation.

[77] Kipp C Cannon. A proposal to add coincidence tables, April 2007.

[78] `http://www.sqlite.org`.

[79] Drew Keppel. Private communication, February 2010.

[80] Larne Pekowsky. *Characterization of Enhanced Interferometric Gravitational Wave Detectors and Studies of Numeric Simulations for Compact-Binary Coalescences.* PhD thesis, Syracuse University, August 2011.

[81] Omega scan. `https://trac.ligo.caltech.edu/omega/wiki/Documentation/omega/scan`, January 2011.

[82] B. Abbott et al. LIGO: The Laser Interferometer Gravitational-Wave Observatory. 2007.

[83] F Acernese et al. The Virgo status. *Classical and Quantum Gravity*, 23(19):S635–S642, 2006.

[84] Masaru Shibata, Koutarou Kyutoku, Tetsuro Yamamoto, and Keisuke Taniguchi. Gravitational waves from black hole-neutron star binaries I: Classification of waveforms. *Phys. Rev. D*, 79:044030, 2009.

[85] Kenta Kiuchi, Yuichiro Sekiguchi, Masaru Shibata, and Keisuke Taniguchi. Longterm general relativistic simulation of binary neutron stars collapsing to a black hole. 2009.

[86] K S Thorne. Gravitational radiation. In S. W. Hawking and W. Israel, editors, *Three hundred years of gravitation*, chapter 9, pages 330–458. Cambridge University Press, Cambridge, 1987.

[87] B S Sathyaprakash and S V Dhurandhar. *Phys. Rev. D*, 44:3819, 1991.

[88] B. Abbott et al. Tuning matched filter searches for compact binary coalescence. Technical Report LIGO-T070109-01, 2007.

[89] Ravi Kumar Kopparapu, Chad Hanna, Vicky Kalogera, Richard O'Shaughnessy, Gabriela Gonzalez, Patrick R. Brady, and Stephen Fairhurst. Host Galaxies Catalog Used in LIGO Searches for Compact Binary Coalescence Events. *Astrophysical J.*, 675(2):1459–1467, 2008.

[90] V. Kalogera, R. Narayan, D. N. Spergel, and J. H. Taylor. The coalescence rate of double neutron star systems. *Astrophysical J.*, 556:340, 2001.

[91] V. Kalogera, C. Kim, D. R. Lorimer, M. Burgay, N. D'Amico, A. Possenti, R. N. Manchester, A. G. Lyne, B. C. Joshi, M. A. McLaughlin, M. Kramer, J. M. Sarkissian, and F. Camilo. The cosmic coalescence rates for double neutron star binaries. *Astrophysical J.*, 601:L179–L182, 2004. Erratum-ibid. 614 (2004) L137.

[92] V. Kalogera et al. Erratum: "The Cosmic Coalescence Rates for Double Neutron Star Binaries (ApJ, 601, L179 [2004]). *Astrophysical J.*, 614:L137–L138, October 2004.

[93] R. O'Shaughnessy, C. Kim, V. Kalogera, and K. Belczynski. Constraining Population Synthesis Models via Empirical Binary Compact Object Merger and Supernova Rates. *Astrophysical J.*, 672:479–488, January 2008.

[94] R. O'Shaughnessy, C. Kim, T. Fragos, V. Kalogera, and K. Belczynski. Constraining Population Synthesis Models via the Binary Neutron Star Population. *Astrophysical J.*, 633:1076–1084, November 2005.

[95] Andrew Lundgren. Private communication, July 2011.

[96] Josh Smith. hveto documentation. `https://wiki.ligo.org/foswiki/bin/view/DetChar/HvetoDoc`.

[97] Tomoki Isogai. Used percentage veto for ligo and virgo binary inspiral searches. *J. Phys. Conf. Ser.*, 243:012005, 2010.

[98] `https://wiki.ligo.org/DetChar/H1LVEASEISZS6Cflag`, June 2010.

[99] `https://wiki.ligo.org/DetChar/SeisVeto`, September 2010.

[100] Duncan Macleod et al. Removing the effect of seismic noise from ligo data by targeted veto generation. Technical Report LIGO-P1100031-v8, LIGO Scientific Collaboration, 2011.

[101] S. Fairhurst. Triangulation of gravitational wave sources with a network of detectors. *New Journal of Physics*, 11:123006, December 2009.

[102] Stephen Fairhurst. Localization through triangulation. `https://www.lsc-group.phys.uwm.edu/ligovirgo/cbcnote/aLIGOaVirgo/100223081702SouthLocalization%20through%20triangulation`.

Vita

Name of Author

Collin D. Capano

Place and Date of Birth

Glens Falls, New York   February 15, 1983

Colleges and Universities   Degrees Awarded

Syracuse University   B.S. Physics, 2005

Syracuse University   M.S. Physics, 2008

Publications

1. Abadie J et al., *Search for Gravitational Waves from Low Mass Compact Binary Coalescence in LIGO's Sixth Science Run and Virgo's Science Runs 2 and 3*, In preparation for Submission to PRD (2011)

2. Abbott B et al., *Search for Gravitational Waves from Low Mass Compact Binary Coalescence in 186 Days of LIGO's fifth Science Run*, Phys Rev D **80** 047101 (2009)

3. Abbott B et al., *Search for Gravitational Waves from Low Mass Binary Coalescences in the First Year of LIGO's S5 Data*, Phys Rev D **79** 122001 (2009)

Awards and Fellowships

2001 Syracuse University Founders' Scholarship