LIGO-SURF Summer 2006: Project Status and Progress Report-II for "Searching for Gravitational-Wave Bursts of Arbitrary Waveform"

Research Mentor: Dr. Shourov K. Chatterji

Primary Researcher: Rubab Khan, Columbia University

Project Status:

As per the outcome of the procedure described in the previous project report, density based clustering had been considered to be the best clustering algorithm for the purpose of this project. A complete density based clustering function has been implemented in Matlab, and is currently being tested using simulated burst signals of various waveforms and strength injected into real LIGO noise taken from the fifth LIGO Science Run (S5). Upon completion of training and testing the clustering function using various injections, it will be ready for integration with the standard Q pipeline. Please refer to the project proposal and first progress report, both attached at the end of this report, for details on project motivations, background, objectives, previous progress, and references.

Progress Report:

After implementing the clustering function in Matlab, a test program has been setup using various modules of the standard Q pipeline and the density based clustering function. It facilitates production of ROC curves (explained later) for the clustering function for various injections in order to evaluate it's performance, and helps to train and test the clustering algorithm extensively. This section explains the technical details of density based clustering algorithm, evaluates its performance so far, and identifies the issues that are yet to be decisively concluded.

Density Based Clustering:

The Density Based clustering has two major advantages for the purpose of this project: first, it is very efficient in finding arbitrarily shaped regions in the time-frequency signal space; and while most other clustering algorithm would include noise or simply any data point in some cluster or other, density based algorithm keeps noise, or data points that could not be put together with significantly many other data points, out of all clusters and identify them as noise. Thus, it is possible to identify and isolate much of the noise in the time-frequency signal space and more efficiently find the most significant clusters. These advantages can be extremely useful since it allows both to search for unknown shapes of signals, and to pick up only significant clusters over a large set of data without cluttering the output report with a list of numerous noise clusters that contain one or just a few data points. Briefly: density based clustering is a simple and computationally economic algorithm that looks for certain data density around arbitrary point to start clustering, and then through a number of different steps based upon the idea of density-link

between points, picks up the whole cluster out of a given data set. It does not require any detailed domain knowledge, and this makes it further suitable for this project's purpose since instead of looking for a specifically shaped signal, the algorithm simply disregards the issue of signal shape.

The algorithm picks a data-point that has “enough” data-points “near” itself, that we call neighbors; and for all such neighbors that individually has enough neighbors the algorithm find their neighbors; and so on; until it reaches a point that doesn't have enough neighbors. Thus, all point that are related through this neighbor relation become one cluster. If some neighbor already belongs to a different cluster, the current cluster is merged with that cluster. Thus, regardless of at which data-point the algorithm starts clustering from, it will always find the same clusters though for speed optimization our density based clustering function picks the more “significant” data-points first. How many neighbors is “enough” (neighbor number) and how close is “near” (neighborhood radius) are the two clustering parameters for this function. According to the recommendation of the authors of [12], and from our experimentation, the neighbor number of 4 seems to be ideal for nearly all distance metrics. The exact numerical value of neighborhood radius is determined using a 4-distance graph as will be explained later in the evaluation section.

Avoiding the language specific technicalities, a simplistic yet accurate pseudo-code of the density based clustering function that we have implemented is given here. It has two modules. The main function calls the “expandCluster” function, and the latter recursively calls itself. Clustering starts at the most significant data-point first and then proceeds to the next significant data point that is not in a cluster, considering only such points that has enough neighbors as cluster seeds to ensure that least number of loops are not executed.

Density Based Clustering Function Pseudo -Code:

Clustering Function:

1. Measure distance from each data-point to each other data-point.
2. Count the number of neighbors within the given neighborhood radius around each data point.
3. Mark every data-point with at least N neighbors as potential cluster seeds.
4. Sort the potential cluster seeds according to significance (normalized energy) in descending order.
5. Initiate clusterNumber at 0 .
6. i) Start loop over significance sorted potential cluster seeds (most significant potential cluster seed first),
   ii) if data-point is not already in a cluster then,
      a) increment clusterNumber by 1,
      b) assign the clusterNumber to the data point,
      c) execute “expandCluster” function for the data-point,
   iii) else, go to next data-point,
   iv) end if,
   v) end loop over potential cluster seeds.

expandCluster Function:

1. Mark neighbors of the given data-point as members of current cluster.
2. i) Start loop over the neighbors,
   ii) if neighbor was originally an unused potential cluster seed then,
   execute “expandCluster” function for the neighbor (recursive call),
   iii) else if the neighbor was originally a member of another cluster,
   merge current cluster with that cluster,
   iv) else, go to next neighbor,
   v) end loop over neighbors.

Evaluation:

It had been expected that the density based clustering will significantly reduce the effect of noise and thus decrease the false rate of detection, and at the same time will magnify the significance of injections or glitches by clustering the most significant tiles with other tiles, and thus increase the detection efficiency. In order to evaluate for such improvements, we first discuss how it improves the situation from where we left at the previous project report. We were analyzing S5 data that included simulated signal injections, and were analyzing 8 seconds windows centered on the injection time to visually evaluate how well a clustering algorithm is performing.

The attached figures show the unique, non-overlapping, and significant time-frequency tiles produced by the Q pipeline for the case of a 5 Mpc inspiral injected during S5. Without clustering, the Q pipeline detects the “most significant tile” in the signal space thus detecting the central time and frequency tile of a signal. But each tile in this case represents an “event”, and we do not automatically derive much information about the shape and nature of the signal from here (Figure-1). Using Matlab built in hierarchical clustering function along with a customized distance metric and manually optimized numerical clustering threshold, much of the injection is clustered together (Figure-2). While this shows the potential advantage of clustering, there are a lot of noise clusters that makes identifying the most significant cluster statistically difficult. A considerable amount of noise is clustered together with the injection making determination of its significance and duration confusing. Also, it splits the injection cluster into two parts, that leads to two detections for one injection, meaning an additional false detection.

In order to use the density based clustering, we produce a “4-distance graph” (Figure-3) that has the distance of the fourth distant neighbor of every point along y axis for every corresponding point on x axis; points are sorted according to descending order of their 4-distance value. Close observation of the 4-distance graph as well as some experimentation says that a neighborhood radius of 8 is a good choice for the current distance metric. Thus, using neighborhood radius of 8 and neighbor number of 4 (as explained before), we get Figure-4 which shows how density based clustering has clustered together the most significant part of the injection successfully. Though it loses the high-frequency end of the injection that contains very little energy and does not significantly contribute to the duration or significance estimation or the detected trigger. There is only one noise cluster on the signal space, while almost all the noise is

removed. Thus, our density based clustering function passes this initial qualitative singe test case pretty nicely. However, for a more formal and sensible method of evaluation we use a more extensive test program.

The test program executes the following tasks: It loads segments of LIGO S5 noise data, runs clustering over the noise only signal space, injects signals of specific waveforms at random times with random strength, and clusters the noise and injection data over the same signal space again. Every “detection” on the noise-only space is considered a false detection, while every injection that is successfully detected is considered a correct detection. The receiver operating characteristic curve or ROC curve that is produced from this data has false-rate on the x axis, and detection efficiency on the y axis. Details of the rigorous calculations performed to produce the ROC's is skipped here. Figures 5 through 8 shows ROC curves produced for inspiral, noise-burst, Gaussian, and sinusoidal Gaussian waveform injections. While there is tremendous improvement for the more significant waveforms such as inspirals and noise-bursts, there is no improvement noticed for Gaussian injections, and there is a noticeable decline in performance of Q pipeline for sinusoidal Gaussian detection. The improvement for inspiral and noise-bursts are pretty encouraging, though it would much better if that happened without affecting the performance of Q pipeline for other waveforms. It needs to be noted that the Q pipeline is optimized to find all sorts of sinusoidal Gaussians, and thus it is not a big surprise that clustering could not do better than optimal, but made Q pipeline's efficiency higher and false rate lower for astrophysically significant known waveform injections. It would be pretty satisfactory if the density based clustering were improving Q pipeline's performance for certain waveform injections while keeping the performance unchanged for all other ones, though in this case we are observing decline in performance for some waveform injections which we would like to avoid. We have some idea about how to improve the performance of the code for sinusoidal Gaussian waveforms, but they are still being investigated.

Apart from the ROC curves, as general observations about the the clustering function we should mention it's speed, memory-efficiency, and overall simplicity. The dominant time consuming module of a clustering embedded Q pipeline will still be qtransform, compared to which the clustering takes nearly no time at all. Since the information about distance from each data-point to each other is not carried in the recursion based clustering section, rather the address of four neighbors for each data-point is carried, the recursive call does not tend to overwhelm the system memory. During running the test program jobs that takes 32 seconds clips at a time hardly exceeded 3% of system memory usage. The clustering algorithm does not use any information about waveforms, and and thus it finds clusters regardless of it's shape using very simple and intuitive logic.

Problems and Challenges:

Improving the performance of the clustering function for injections of sinusoidal Gaussian waveform is definitely a top priority at the moment. Also, we are currently re-assessing the post-processing that evaluates the performance of the test program. Aside from this specific issue, two more basic points that need further attention are the distance metric and definition of a detection. Currently, the distance metric inflates the distance of frequency scale in order to

promote clustering across time bins, and also compensates for tiles that are elongated on either time or frequency scale. Though this distance metric is performing pretty well and we have not yet been able to come up with a better performing one despite rigorous efforts in this regard, this distance metric was developed more through a trial-and-error method rather than a more theoretically and methodically sound one. The definition of a positive detection also gets literally blurry as definitions of cluster-center, cluster-width, and cluster-significance all become heavily dependent on their definitions as developed during the course of this project. Though all those definitions so far have carefully been adopted to be possibly most neutral, further effort to rationalize their usage or modifying them is necessary.

Research Goal:

While the original goal of the project was to add a clustering module to the standard Q pipeline, and run a triple co-incident search, realistically reassessing the goals at this point of the project indicates that the initial goal, while being the rational overall aim of this project, was slightly too optimistic to fit withing the project's time-scale. Developing a clustering module that is compatible with most components of the Q pipeline is a somewhat separate issue from rebuilding the Q pipeline including the clustering module, not to mention that the latter is a bit more time consuming task as well. As clustering makes the definition of center time and durations of candidate events literally blurred, defining a multiple detector co-incident is quite an extravagant ambition, at least at this point. A revised goal is thus determined to deliver an expanded Q pipeline code complete with the clustering module that would be ready for running triple coincident searches in near future as the work in this regard carried on beyond this specific project. This should result in a professional journal publication withing a few months, and the results will be formally presented at the December session of Gravitational Wave Data Analysis Workshop (GWDAW) due to take place in Germany.

Interactions with Mentor:

Interactions with my mentor has been very lively, frequent, and elaborate. Since both of us are situated on the same floor just a few doors away, we have always been in close contact with each other in order to evaluate project status, carry out implementations, and of course, to brain-storm and optimally utilize the white-board space. The personal supervision and guidance provided by my mentor in spite of the busy schedule of a post-doctoral researcher that he has to maintain has been of a great help throughout the project, as well as being a pleasant academic experience.

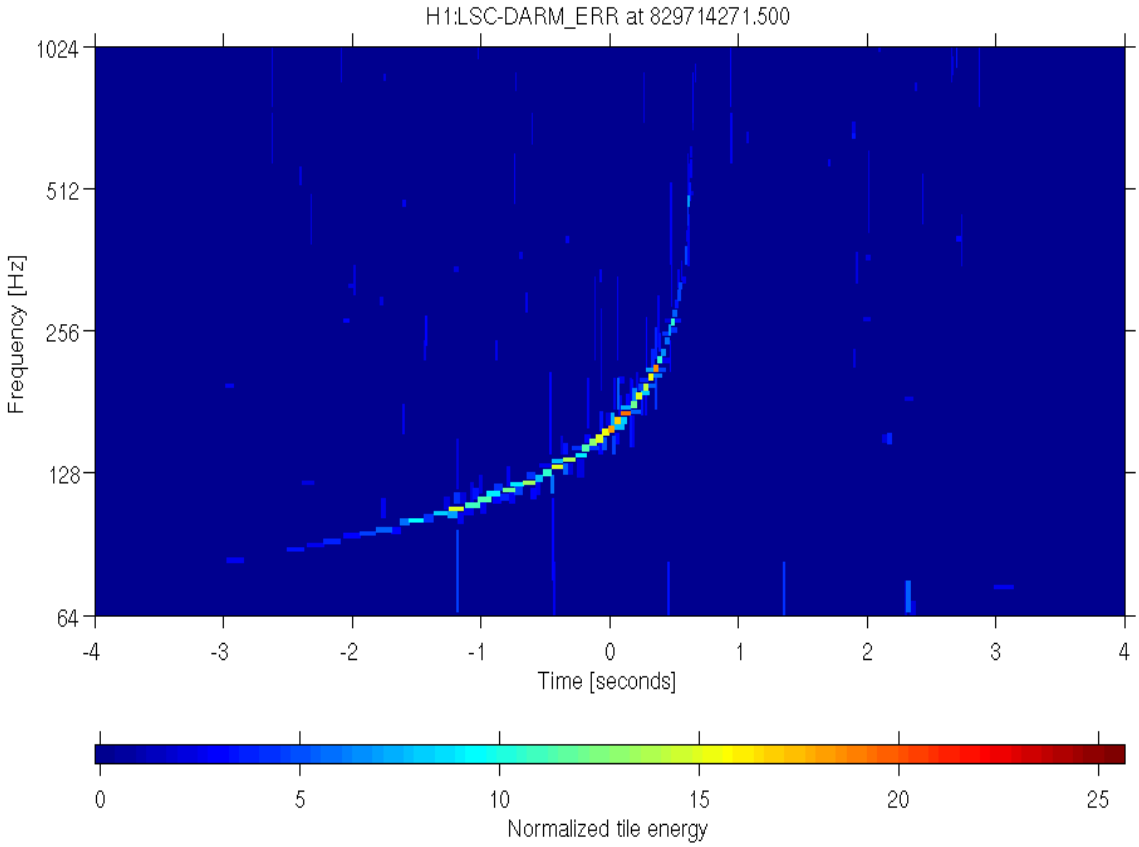

Figure 1: Unique, non-overlapping, and significant time-frequency tiles on q-planes as produced by Q pipeline without clustering for a inspiral injection during S5. Each tiles represents an "event" and there is a "most significant tile" of highest normalized energy.

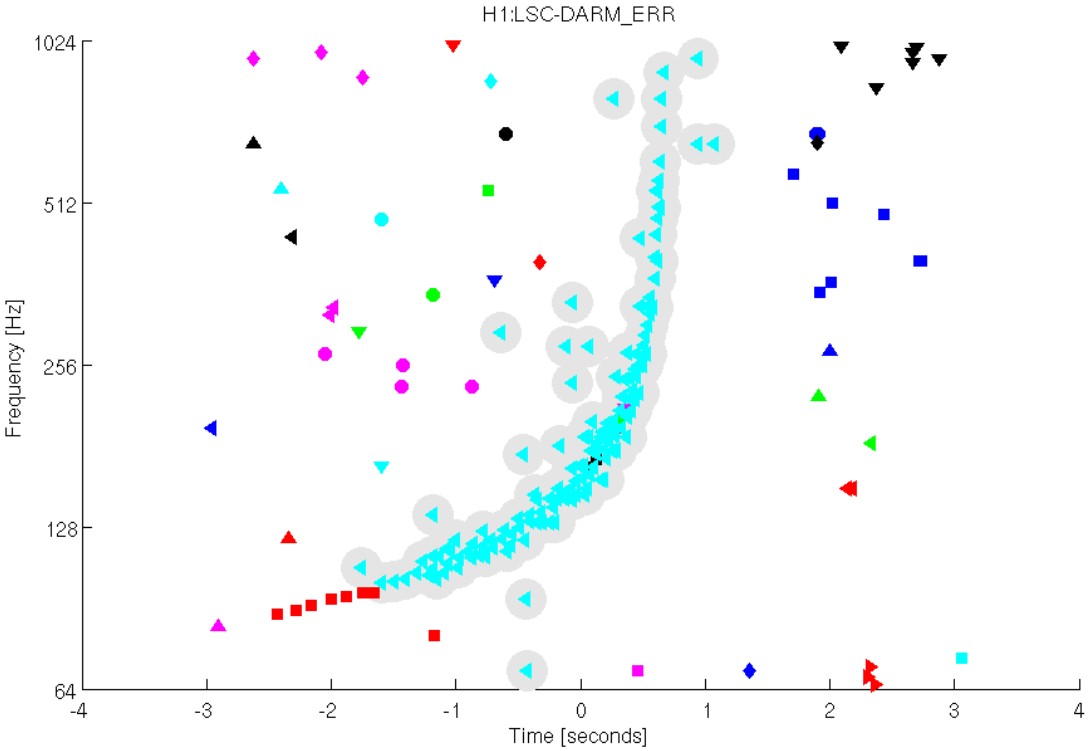

Figure 2: The inspiral injection in Figure 1, after hierarchical clustering using Matlab clustering function, customized distance function, and manually optimized clustering threshold. There are a lot of noise clusters in the figure, and a lot of noise has been considered to be part of the injection.

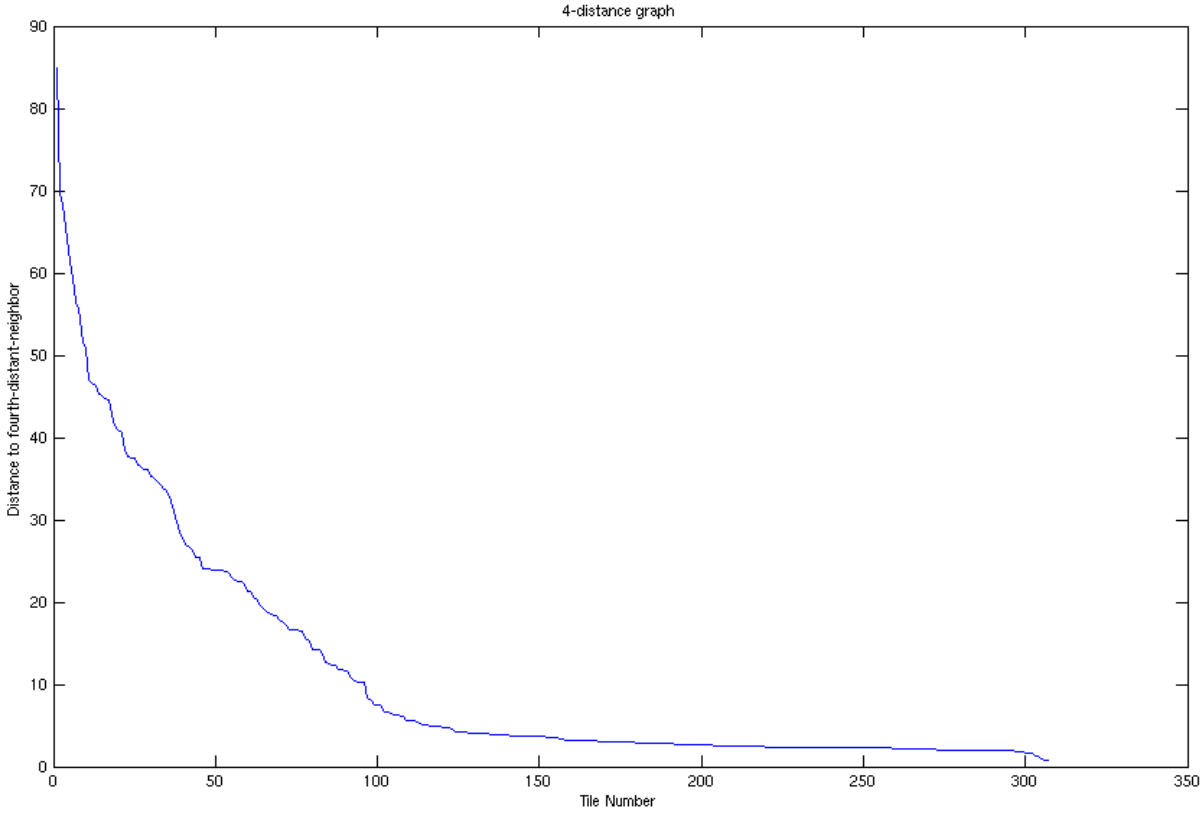

Figure 3: A 4-distance graph that helps determine clustering threshold for density based clustering.

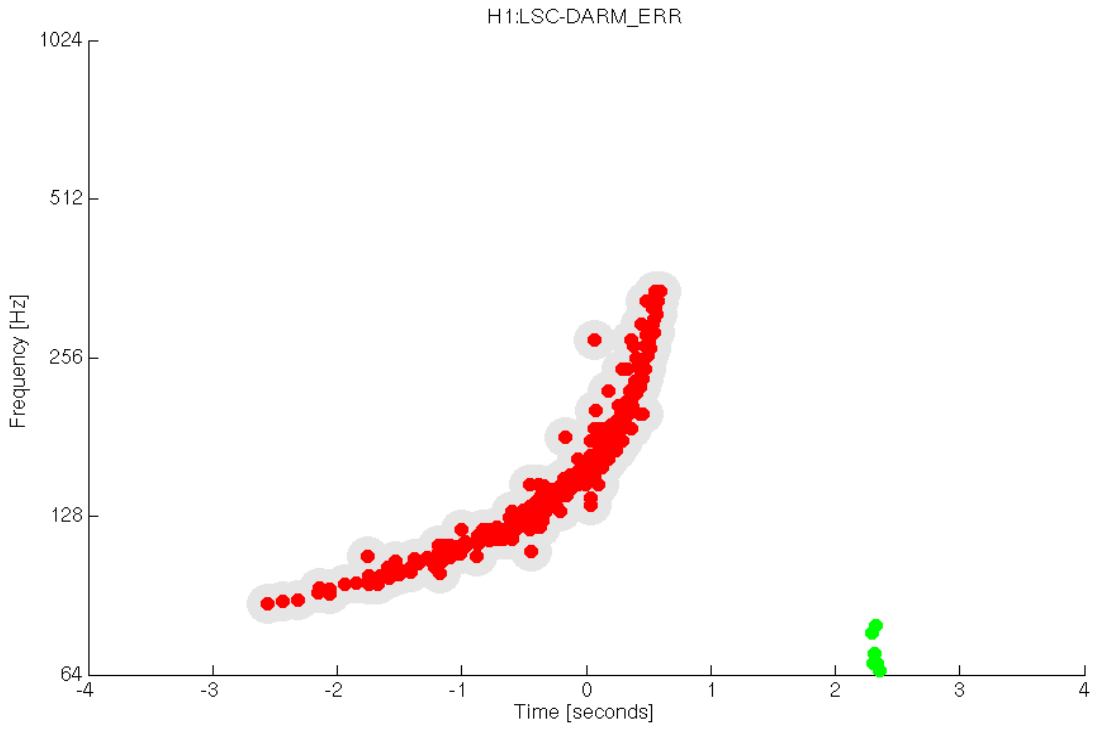

Figure 4: Same injection as of Figures 1 and 2, after density clustering. The clustering algorithm has successfully identified the injection picking off almost all the noise tiles except for a few in one noise cluster. Though the high frequency top portion of the injection has not been identified, that part contains very little energy, and almost all the normalized-energy of the injection is contained in the portion identified by density clustering.

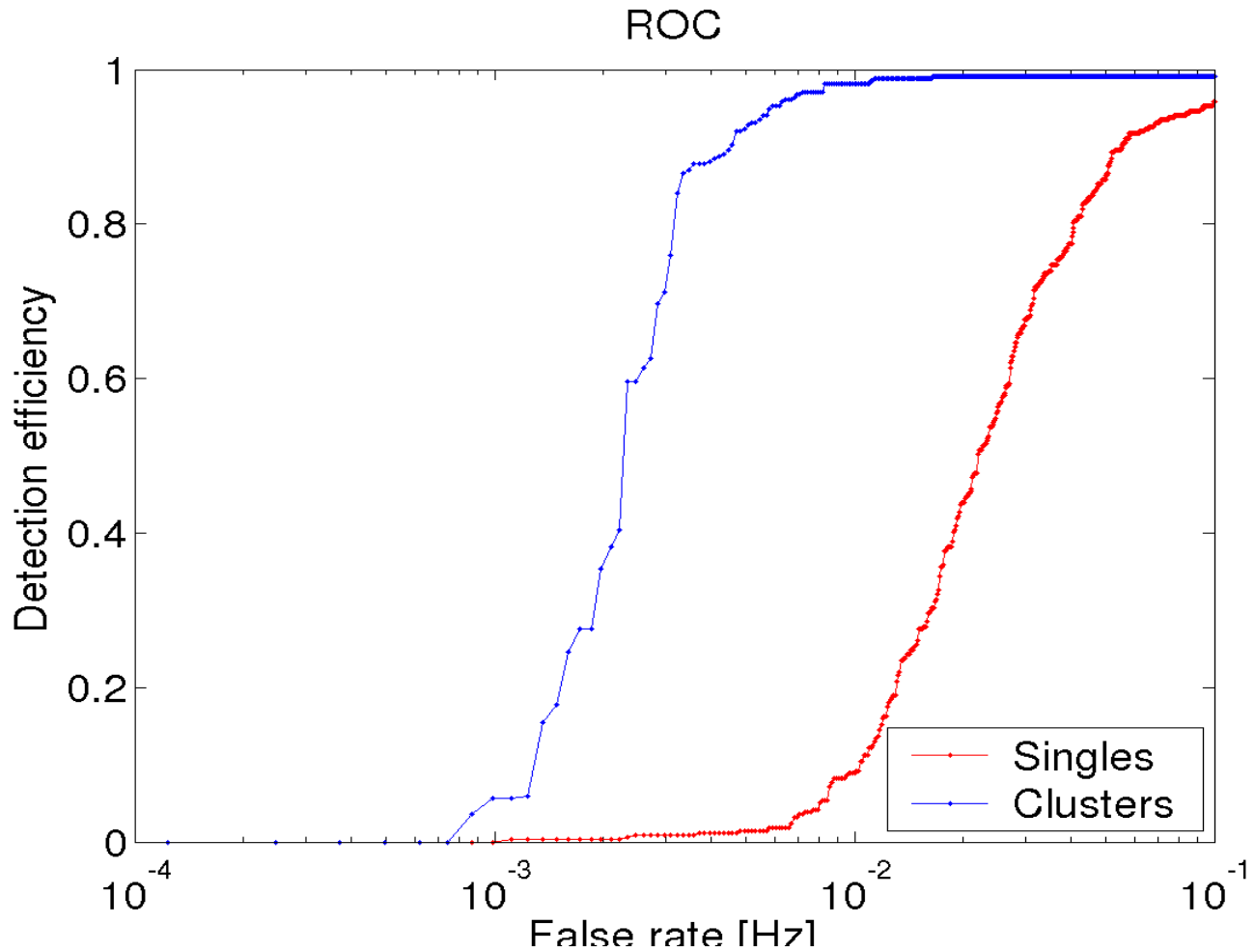

Figure 5: ROC curve inspiral injections. The significant development achievment in terms of higher detection efficiency and lower false rate is evident.

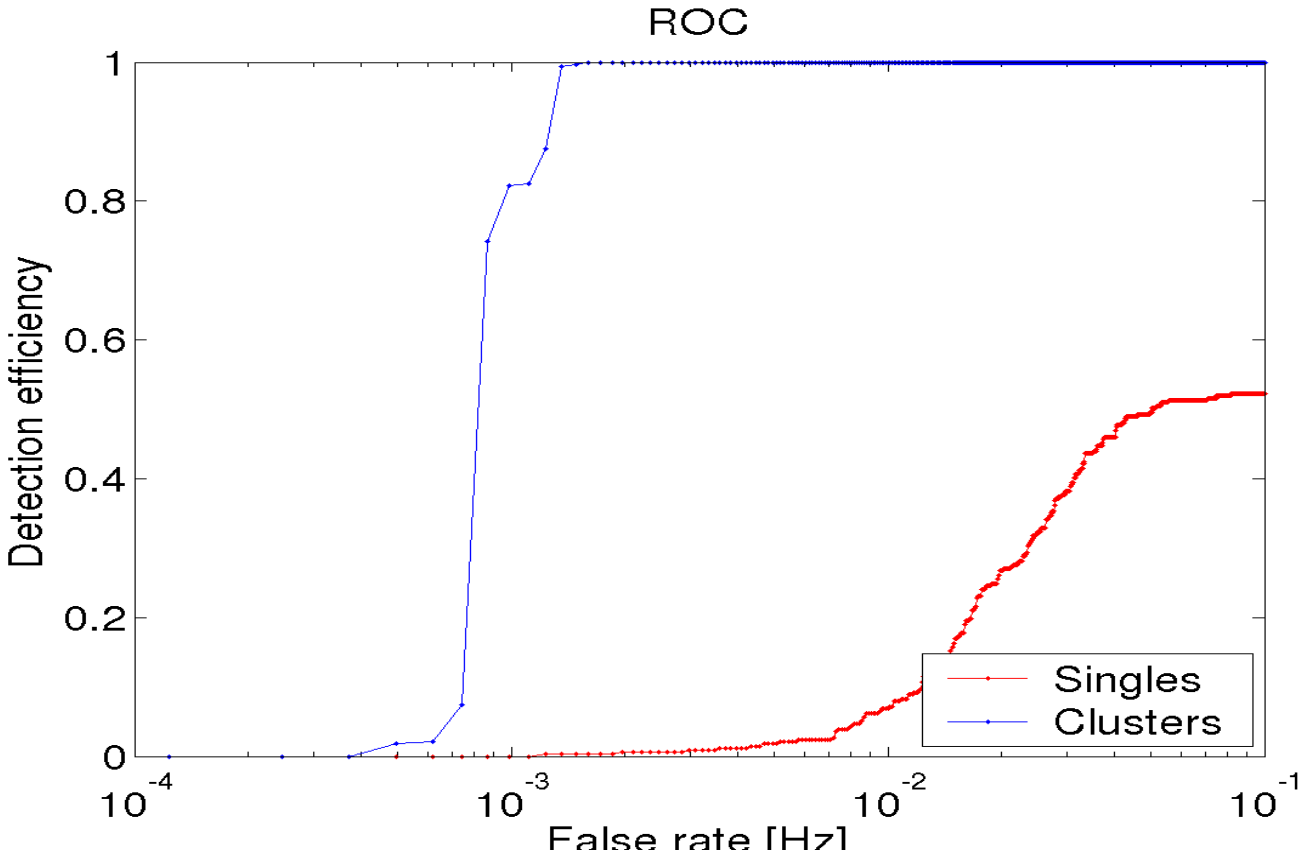

Figure 6: ROC curver for noise-burst injections. The significant development achievment in terms of higher detection efficiency and lower false rate is evident.

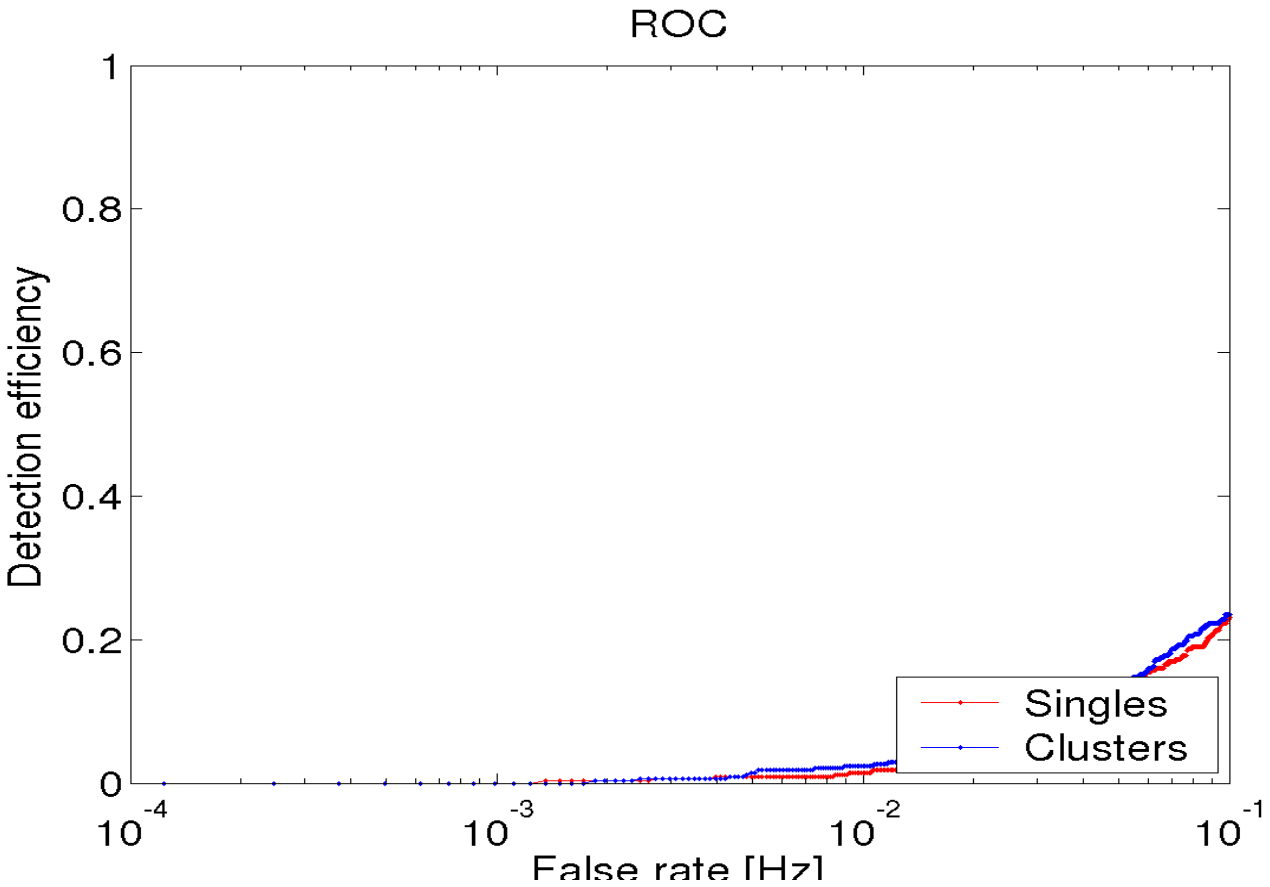

Figure 7: ROC curve for gaussian injections. No significant improvement in terms of efficiency and false rate noticeable.

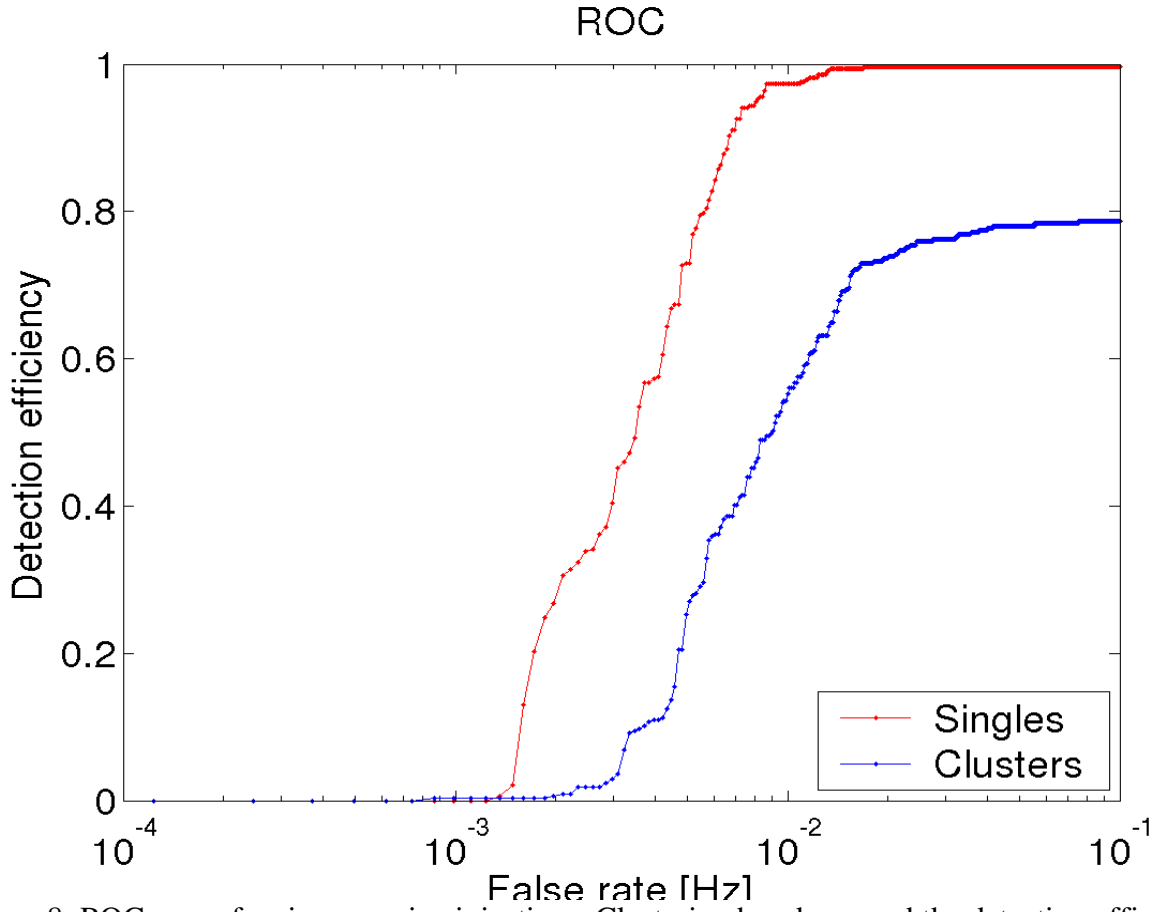

Figure 8: ROC curve for sine-gaussian injections. Clustering has decreased the detection efficiency and increased the false rate.

Time-line:

Weeks	*Task*	*Status*
<1	Prior to project start: i) Familiarize with Q pipeline algorithm and codes. ii) Familiarize with various clustering methods. iii) Create a shortlist of clustering algorithms to try.	Completed
1, 2	i) Write simple Matlab scripts to experiment with some of the short listed algorithms. ii) Evaluate their relative performance and choose one algorithms to implement as the Q pipeline extension.	Completed
3, 4, 5	i) Implement the chosen clustering algorithm in Matlab. ii) Prepare a test code to evaluate the performance of the chosen clustering algorithm.	Completed
6, 7, 8	Test the performance of the extended Q pipeline on a variety of simulated bursts, and compare it to the performance of the Q pipeline without clustering.	Ongoing
9	i) Test the performance of the extended Q pipeline on real LIGO data collected during S5, and compare its efficiency in identifying candidate-events with that of the Q pipeline without clustering. ii) Prepare the final presentation.	
10	i) Document the work and its results. ii) Prepare the final technical paper.	

Date: August 01, 2006.

(Mentor)

Shourov K. Chatterji
Post-doctoral Researcher, LIGO-Caltech.

LIGO-SURF Summer 2006: Project Status and Progress Report with Updated Proposal for

"Searching for Gravitational-Wave Bursts of Arbitrary Waveform"

Research Mentor: Dr. Shourov K. Chatterji

Primary Researcher: Rubab Khan, Columbia University

Project Status:

After considering various aspects of different data clustering methods, the candidate algorithms list has been narrowed down to existing Matlab hierarchical clustering functions, TFclusters, Density Based clustering, Graphmon, and Spectrum Analysis. A skeleton code has been prepared in Matlab that can take in various clustering modules, and makes evaluating their clustering efficiency easier. Apart from progress towards developing the core module for clustering, the overall structure of the final code has been set up that has allowed to develop some further basic idea about how the whole program will work. Please refer to the updated project proposal for details on project abstract, background, objectives, references.

Progress Report:

The TFclusters algorithm consists of three steps: pixels of the time-frequency representation of the data that have power above a fixed threshold are first identified, clusters of such pixels that conform to a set of rules on their size and their proximity to other clusters are formed, and a final threshold is applied on the power integrated over all pixels in such clusters (9). Thresholding over power is already a part of the standard Q pipeline implementation, and thus it would not yield any additional significant leverage. Thresholding on clusters based on their shapes is against the primary goal of this project that is to find burst signals of unknown shape, and identifying shapes at an early stage would bias the code towards certain forms of signals. A final threshold over the clusters, despite still being an option, would require deeper considerations since simply summing up the energy of the tiles included in a cluster does not properly represent it's significance. Considering all these aspects, Density Based Algorithm is currently being considered to be the first algorithm to try implementing and evaluating. Other clustering algorithms, such as Graphmon and Spectrum Analysis may be tried later depending on how well Density Based Algorithm works for this project's purpose.

The Density Based clustering has two major leverage for the purpose of this project: first, it is very efficient in finding arbitrarily shaped regions on a signal space through clustering; and while most other clustering algorithm would put noise or simply any data point in some cluster or other, density based algorithm keeps noise, or data points that could not be put together with significantly many other data points, out of all clusters and identify them as noise. Both of these can be extremely useful since it allows both to search for unknown shapes of signals, and to pick up only significant clusters over a large set of data without cluttering the output report with a list of numerous noise clusters that contain one or just a few data points. Briefly: density based clustering is a simple and computationally

economic algorithm that looks for certain data density around arbitrary point to start clustering, and then through a number of different steps based upon the idea of density-link between points, pick up the whole cluster out of a given data set. It does not require any detail domain knowledge which makes it further suitable for this project's purpose. The critical question on defining density on the q-plane in this case may either be “How much energy is there in a certain area around this point” or “How many data-points are there within a certain energy-space around this point.” On the downside, this algorithm requires a hard-coded threshold to be given as input which will be very inconvenient for a search of signals from various distances and sources. As will be explained in the overview of the whole data analysis program in next paragraphs, density based algorithm probably will be very useful in clustering data points in the initial cluster building step, but to get the whole signal a further step of clustering-the-clusters will be necessary as the whole data analysis program is being planned now.

The way the current skeleton program has been coded, it at first reads background noise and injects signal on that noise before running this data through the standard Q pipeline modules that produce a list of events characterizing them with their central time, frequency, q, and normalized energy. Taking these four parameter as variables, the clustering function is called which currently is built on Matlab Statistical Toolbox clustering functions. After the clustering function produces small clusters in the first step, a further clustering module will run repeatedly to cluster the clusters together without merging signals into noise. The aim is to create a generalized enough clustering module that would run iteratively by calling itself within it, and take the output of every iterative step as the input of the next until a certain statistical criteria is fulfilled, such as, until the number of clusters do not change significantly in a iterative run.

While simply using spatial distance functions built into Matlab does not yield much meaningful gain as noise and signal are always clustered together in such case, using custom distance function with Matlab's pdist distance measurement function has produced much better results. Depending on the threshold that is currently set manually, the code can already pick out inspiral signals isolating it from most noise. However, it still clusters a considerable amount on noise with the signal, and tends to break off the lower frequency tail of the inspiral signal into pieces. The problem would most possibly increase for other more complicated signal shapes.

As can be seen in the attached figures, the unique, non-overlapping, and significant time-frequency tiles on q-planes as produced by Q pipeline without clustering for a 5 Mpc inspiral injection during S5 detects a “most significant tile” in the signal space thus detecting the central time and frequency tile of a signal. But each tile in this case represents an “event”, and we do not automatically derive much information about the shape and nature of the signal from here (Figure-1). Applying Matlab built in distance and clustering functions on this signal space, the resulting cluster does not tell us much about the signal either, and it includes too many noise tiles clustered with signal tiles in the largest cluster to reach any meaningful conclusion (Figure-2). Also, it builds a large noise cluster which undermines the reliability of the code for detecting signals with desirable certainty. However, using a customized distance function with Matlab built in clustering functions improves the scenario significantly as it picks up the signal shape pretty nicely, and does not build any large noise cluster that is comparable to the main clusster (Figure-3). Despite the fact that it breaks off the low frequency tail of the inspiral injection in a separate cluster, and that the main cluster includes a considerable amount of noise, this figure shows the obvious advantage and leverage that is gained by using clustering over Q pipeline output. If we raise the cutoff

threshold for clustering just high enough to include the “tail”, it includes even more noise in the main cluster which is an undesirable change. Thus, the goal at this point becomes building a clustering program that would find out the signal accurately, and not include any noise in the main cluster or build too many large noise clusters.

Plan for Coming Weeks:

At this point of the project, as it has become clear that manipulating Matlab clustering functions, despite already showing the advantage of using clustering, might not optimally yield the desired result of the project, the obvious next step is to develop a clustering module using Density Based clustering algorithm. The step next to that would be to develop a separate module to cluster the clusters, and to ultimately merge these two modules into one that would be capable of running iteratively to build upon the output of density based clustering, and reprocess it's own output data until a certain statistical criteria is fulfilled. On the theoretical side, defining data density and data distance on the time-frequency plane, and defining cluster significance remains to be major challenges at hand. Moreover, at some point sooner or later, setting up a self sufficient and automated thresholding system based on definition of cluster significance will also be a vital factor to consider.

As various clustering modules are developed based on different algorithms, they will be plugged into the skeleton program already developed, and the quantitative reports based on results from them will be used to produce detection-efficiency against false-rate ROC curves that will help in determining which clustering algorithm is most efficient in detecting unmodelled bursts of unknown shapes on the time-frequency plane. At first each method will be validated using various types of noise such as white noise, simulated noise, and real detector noise with various sorts of injections such as sine-Gaussian, Gaussian, white-noise burst, and inspirals or ring-downs at different distances. Then the completed code built using the chosen clustering algorithm will be ran on real S5 data. The most significant aspect to watch out for during this evaluation process will be the performance of the completed code near detector sensitivity limit, and that will be the actual meaningful parameter of success for this project.

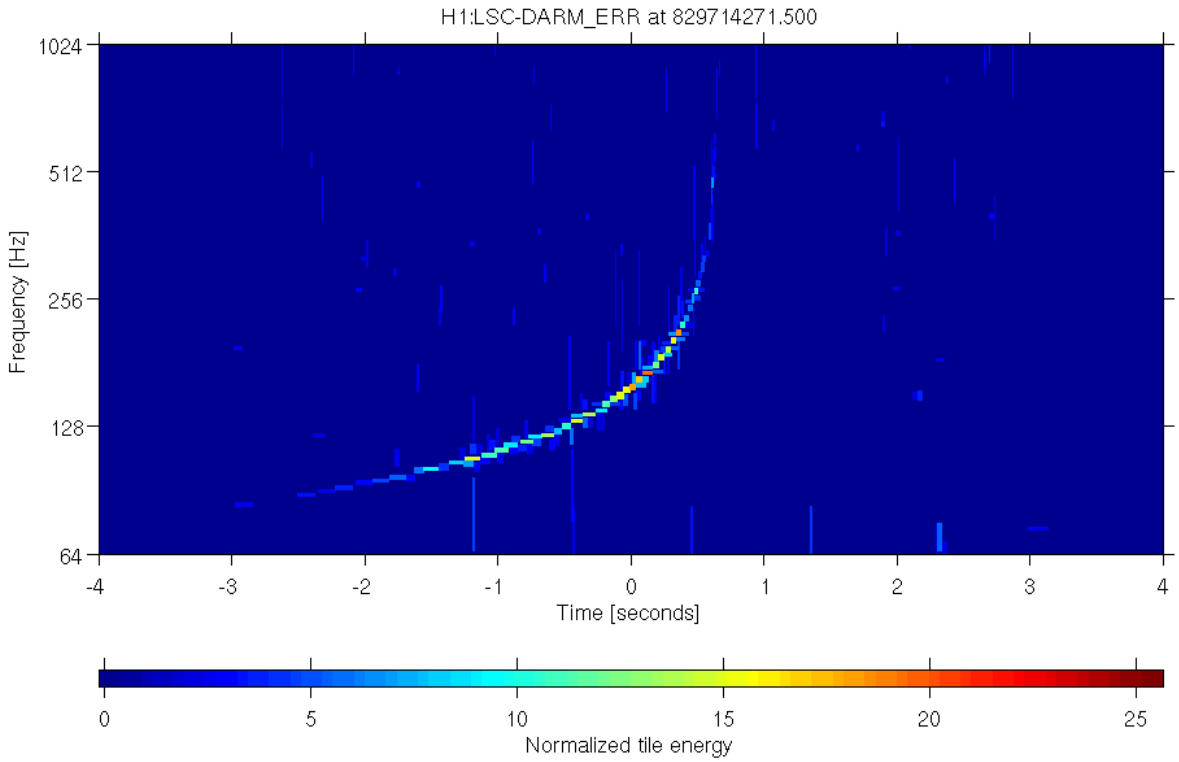

Figure 1: Unique, non-overlapping, and significant time-frequency tiles on q-planes as produced by Q pipeline without clustering for a inspiral injection during S5. Each tiles represents an "event" and there is a "most significant tile" of highest normalized energy.

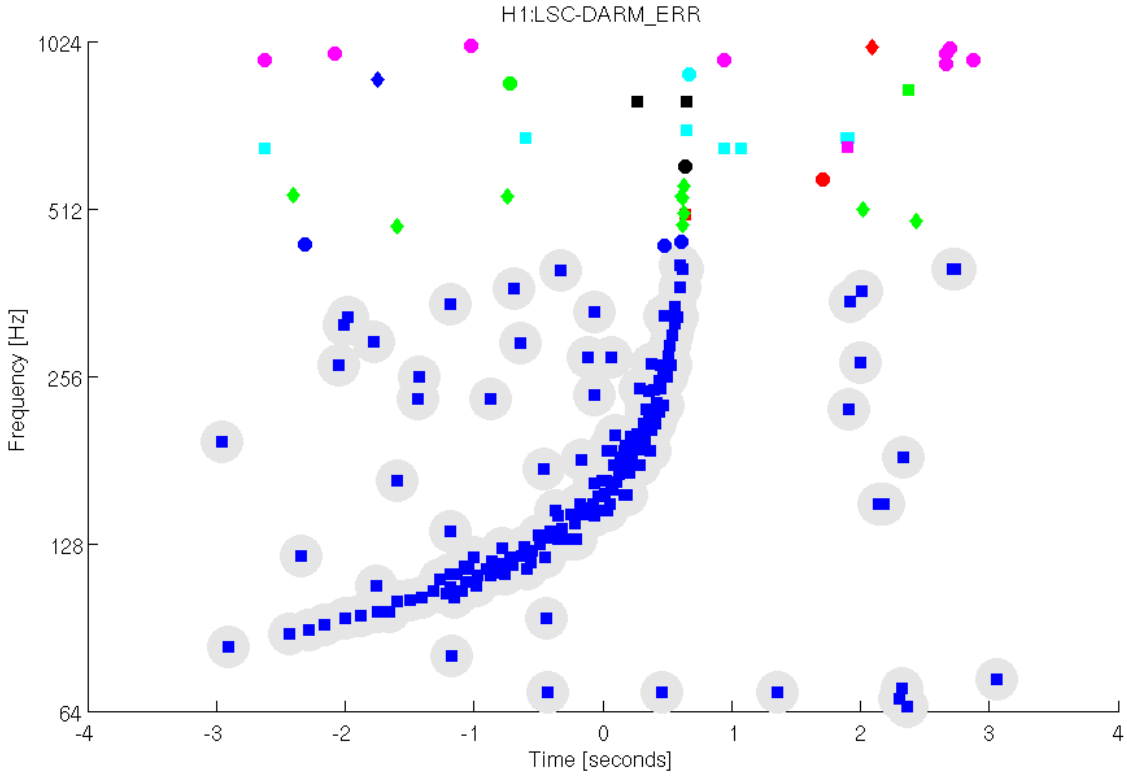

Figure 2: Same injection; clustered using built in Matlab clustering and distance functions with manually set threshold. Too many noise is clustered with signal, and signal shape is very difficult to determine from here.

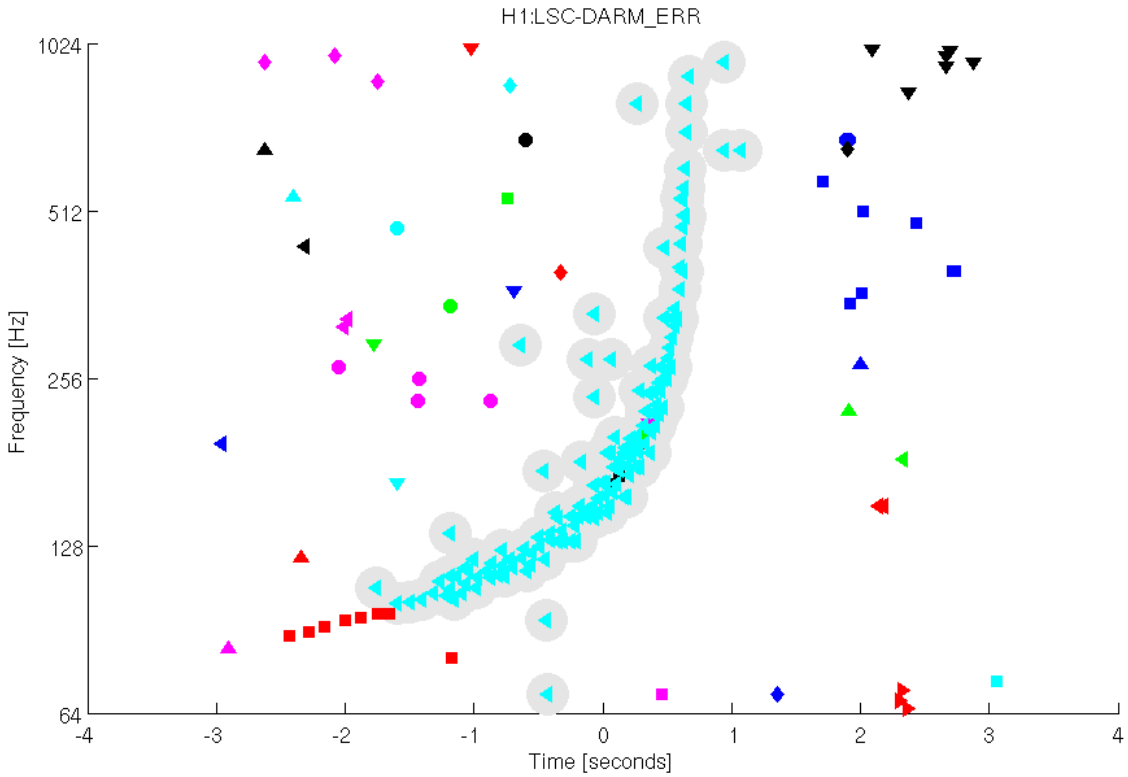

Figure 3: Same injection; clustered using built in Matlab clustering function, and a manually optimized custom distance function with manually set threshold. The low frequency "tail" of the signal has been excluded from the main clusters, and there are some noise clustered with the signal tiles.

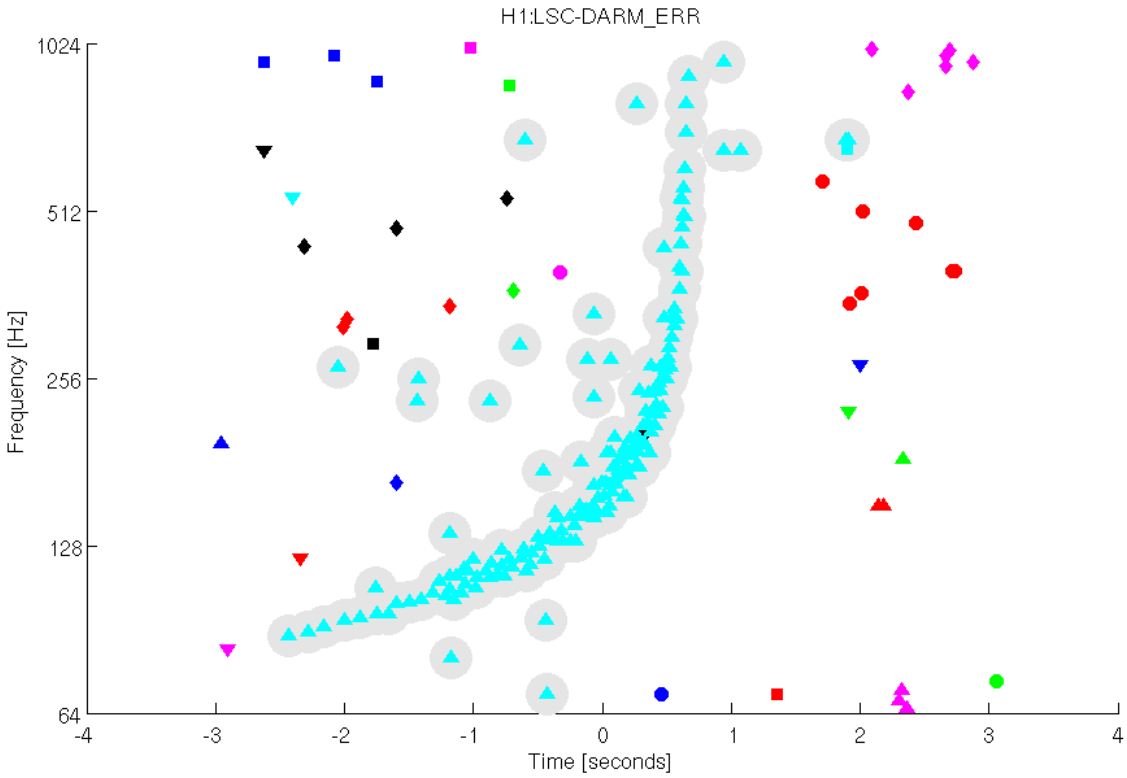

Figure 4: Same injection and functions as in previous figure at a higher threshold. The "tail" is now a part of the main cluster, but so is much more noise.

LIGO-SURF Summer 2006 Updated Project Proposal for

"Searching for Gravitational-Wave Bursts of Arbitrary Waveform"

Research Mentor: Dr. Shourov K. Chatterji

Primary Researcher: Rubab Khan, Columbia University

Abstract:

One class of signal LIGO is searching for consists of short duration gravitational wave bursts of a priori unknown waveform. Potential sources include core collapse supernovae and the coalescence of binary black holes. To detect such events, existing search algorithms project the LIGO data stream onto various time-frequency bases and then search for regions of excess signal energy. One of these search algorithms, the Q pipeline, determines the statistical significance of events based solely on the peak signal observed in the time-frequency plane. This project will investigate extensions to this approach that also consider the statistical significance of arbitrarily shaped regions in the time-frequency plane. Such approaches offer the prospect of improved performance for a variety of sources of both known and unknown waveform.

Background:

Einstein's General Theory of Relativity describes gravity as a space-time curvature due to the presence of mass-energy. One prediction of GR is that as concentrations of mass-energy rapidly changes shape (i.e. supernovae explosion, merger of astronomical binary systems, star-quake etc.) they create dynamically changing space-time warpage or ripples in spacetime that propagates throughout the universe at the speed of light [1,2]. When they reach Earth, gravitational waves are extremely weak perturbations of local flat spacetime, and the detection of these elusive waves to gain knowledge of their sources becomes a tremendous challenge. Unlike electromagnetic waves which propagate through spacetime after being created by the incoherent motion of atoms and molecules and have a wavelength much smaller than their sources, gravitational waves are propagated as perturbation of spacetime itself after being created by massive astronomical sources and have wavelengths similar to the size of their sources. Most significantly, while electromagnetic waves interact with most objects, the universe is mostly transparent gravitational waves. This is both a blessing and a curse since it ensures that the gravitational waves that reach us have not been meddled with since they were created, while this also makes their direct observation more difficult [3]. While the existence of gravitational waves have been observationally confirmed through the discovery of binary stars' (PSR 1913+16) spiraling together just at the rate predicted by GR due energy loss through gravitational radiation [2], efforts to directly detect gravitational waves are yet to achieve decisive success. While the very first direct detection of gravitational waves will be a very significant, the payoff will come when analyzing the detected waves from which it will be possible to extract information about the physics of the extreme conditions where the waves had originated – such as very strong gravity and nuclear densities – information that is not easily accessible or totally inaccessible from electromagnetic observations.

Current efforts of building gravitational wave observatories focus on using interferometry to make extremely precise observations of the distance between two sets of test masses. A global network of interferometric detectors is currently at various stages of data-collection, commissioning, construction, or planning phase. They include the Laser Interferometer Gravitational-wave Observatory (LIGO; three detectors; two in Hanford, Washington and one in Livingstone, Louisiana), Virgo (Pisa, Italy), GEO600 (Hanover, Germany), TAMA300 (Tokyo, Japan), and ACIGA (Perth, Australia) [3]. Each LIGO detector (or every interferometric detector) is basically an ultra-sensitive giant Michelson interferometer having two arms in L-shaped structure. Naively, it can be imagined that as a gravitational wave passes through stretching one arm and squeezing the other, and then reversing this effect periodically, LIGO measures this change of length and thus convey information about the passing waves. If waves of similar nature are detected at multiple detectors nearly simultaneously, it further enhances the level of confidence about the detection and value of the information collected. Moreover, the time difference (<10ms) between detection at two sites give the LIGO detectors directional sensitivity that helps to better determine exactly from which direction of sky a wave is coming in from. The LIGO detectors have now reached their design sensitivity as a result of relentless efforts on the part of numerous engineers and scientists pushing the limits of technology, and currently LIGO is collecting data as it is undergoing its yearlong science run.

It is expected that observation of signals at LIGO will occur near the limit of detector sensitivity, and searching for and identifying such small signals in the presence of various detector noise is a daunting task [4]. Depending on what algorithm we use to search for gravitational waves in data collected at detectors, sources of gravitational waves are classified into four major groups. The inspiraling of a star into its compact binary partner (i.e. neutron star or black hole) causes chirp signals. This process is sufficiently well understood and there are special tools to search for gravitational waves produced by them. Signals from very short-lived events of which we do not have sufficient understanding to predict an expected waveform, such as merger of binary objects, core collapse supernovae, gamma ray bursts, and even unexpected sources, are classified as Burst Sources. Stochastic signals can be either relic gravitational-waves from the very early universe or the cumulative effect of many low amplitude sources that can give rise to a correlated random noise in multiple LIGO detectors as co-incident events. Periodic signals from spinning compact objects (pulsars) can produce a signal if they have asymmetric shape or mass distribution [5].

If the waveform of GW burst is known, then matched filtering can be used which first whitens the data under test by a filter whose magnitude response is the inverse of the detector noise spectrum, next forms the projection of the data onto the waveforms that are to be detected, and then looks for times when the projection is large. However, for bursts of unmodeled waveform, the data under test are typically projected onto a convenient basis of abstract waveforms that are chosen to cover a targeted region of signal space, and one looks for large projections. These searches can be time-domain searches in which the primary basis consists of delta functions in time, and time-frequency searches in which the typical basis consists of windowed complex exponentials or wavelets [6]. The focus of this project is the expansion of the gravitational wave burst search algorithm titled Q pipeline that looks for unmodeled bursts. The Q pipeline is a comprehensive end-to-end analysis pipeline for the detection of gravitational-wave bursts in data from a single interferometric detector. It consists of whitening by zero-phase linear prediction, application of the discrete Q transform, thresholding on the white noise significance of Q transform coefficients, identification of the most significant set of non-overlapping time-frequency tiles, and a final

stage that excludes all but the most significant time-frequency tile within a specified time window in order to prevent the redundant reporting of candidate events [7]. The analysis tool of Q pipeline is the Q-transform which is a modification of the standard short time Fourier transform in which the analysis window duration varies inversely with frequency such that the time frequency plane is covered by tiles of constant "Q" [7] which can be naively interpreted as a dimensionless quality factor for bursts which is the ratio of the center frequency to the bandwidth of a burst [6].

Objectives:

As the Q pipeline projects the data into small regions of the time-frequency plane, a Heisenberg like uncertainty relation applies and bursts cannot have both a well-defined frequency and a well-defined time. The minimum uncertainty signal is a "sine-Gaussian", that is a sinusoid with a Gaussian amplitude envelope: $h(t) = \exp(-(t - t0)^2 / (4 \text{ sigmat}^2)) * \sin(2 * \pi * f * t)$. The algorithm is therefore optimal for signals that have this waveform. For signals that are less localized in the time frequency plane, their detectability is currently determined by their maximum projection onto the space of sinusoidal Gaussians. Q pipeline's treatment has been somewhat limited for bursts that are poorly localized in the time-frequency plane. In particular, it only considers the statistical significance of single most significant tile with minimum time-frequency uncertainty. In searching for statistically significant events, methods of clustering the measurements from neighboring or overlapping basis functions to more optimally detect signals that are not well represented by the particular choice of basis can be employed [6]. An improvement in the detectability of poorly localized bursts should be possible if it would consider the combined statistical significance of clusters of time-frequency tiles by clustering together projections that are nearby in time and frequency[8]. In addition, as described in [7], when evaluating the statistical significance of clusters of time-frequency tiles or testing for time-frequency coincidence between detectors, a more accurate treatment of the overlap between time-frequency tiles can be obtained by applying the mismatch formalism [8]. This should significantly improve the detectability of bursts with arbitrary waveform.

This project will investigate extensions to Q pipeline that would consider the statistical significance of arbitrarily shaped regions in the time-frequency plane by utilizing the advantages of clustering algorithms, and thus would significantly improve detectability of bursts with arbitrary waveform that are less localized in time frequency plane.

Approach:

There has been some work done already on clustering algorithms in the past, and the initial stage of this project will be to identify among them an appropriately promising one, which can be utilized as an extension to Q pipeline. Some algorithms that should be considered can include:

1. Windowed Clustering: The simplest approach would be to slide a "window" of duration T and bandwidth F over the mosaics and record whenever the total significance inside the window exceeds a given threshold. This is of course sensitive to the choice of T and F. It may not be as powerful as some other approaches, but it is conceptually and computationally very simple.

2. TFClusters: This is one of the first algorithms that were applied to search for gravitational-wave bursts in LIGO data. It is similar to the Q pipeline in that it identifies regions of statistically significant excess signal energy in the time-

frequency plane. However, instead of attempting to test multiple time-frequency resolutions, it clusters the results from a single time-frequency resolution. The clustering algorithm is well documented [9,10] and it may be possible to adapt to the Q pipeline.

3. Graph Analysis: This is a method developed to identify "glitches" in the auxiliary and environmental data channels from LIGO, but the method may be very applicable to searching for gravitational waves as well [11]. This method is specially applicable to search for supernovae bursts.

4. Spectral Clustering: It is a partitional clustering algorithm. Given a set of data points, a similarity matrix is defined that measures the similarity between any two data points. Spectrum of the similarity matrix is used to cluster the data. This approach is also used for dimensionality reduction to cluster data in fewer dimensions.

5. TrackSearch: This algorithm is specifically meant for gravitational-wave bursts that produce ridge-like features in the time-frequency plane. It is currently under development and will be considered if time allows.

6. Density Based Algorithm: If time allows, we will also consider algorithms that cluster based on the density of nearby significant tiles[12].

At first, there shall be a short list of clustering approaches to try out as possibilities. Next, we should write some simple Matlab scripts to experiment with these algorithms, and select a method that seems promising and realistically simple to implement correctly. Then it has to be added as an extension to the Q pipeline so that we may test its performance on a variety of simulated bursts, and compare it to the performance of the Q pipeline without clustering. We should then repeat this last step for real LIGO data collected during S5 and compare expanded Q pipeline's efficiency in identifying-candidate events with that of the Q pipeline without clustering.

References:

[1] D. Sigg, “gravitational waves”, LIGO-P980007-00-D, 1998.

[2] K. S. Thorne, “gravitational waves”, arXiv:gr-qc/9506086 v1, 1995.

[3] S. A. Hughes et al. “New physics and astronomy with the new gravitational wave observatories”, arXiv:astro-ph/0110349 v2, 2001.

[4] S. K. Chatterji, “Chapter 1: Introduction”, emvogil-3.mit.edu/~shourov/thesis/chapter1.pdf, 2005.

[5] B. C. Barish and R. Weiss, “LIGO and the Detection of gravitational waves”, LIGO-P99039-00-R, 1999.

[6] S. K. Chatterji, “Chapter 3: Burst Detection”, emvogil-3.mit.edu/~shourov/thesis/chapter3.pdf, 2005.

[7] S. K. Chatterji, “Chapter 5: The Q Transform”, emvogil-3.mit.edu/~shourov/thesis/chapter5.pdf, 2005.

[8] S. K. Chatterji, “Chapter 8: Conclusion”, emvogil-3.mit.edu/~shourov/thesis/chapter8.pdf, 2005.

[9] J. Sylvestre, “Time-frequency detection algorithm for gravitational wave bursts”, arXiv:gr-qc/0210043, 2002.

[10] J. Sylvestre, “Upper Limits for Galactic Transient Sources of Gravitational Radiation from LIGO First Observations”, LIGO-P020007-00-R, 2002.

[11] H. Bantilan, “Graph Analysis”, virgo.physics.carleton.edu/Hans/index.html, 2005.

[12] M. Ester, “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”, ifsc.ualr.edu/xwxu/publications/kdd-96.pdf.