Data Analysis Lab

The purpose of this lab is to provide an insider look in the field of Data Analysis, using tools such as Computing Cluster to work with large sets of experimental data, such as gravitational wave data. The main goal of the lab is to detect a signal injected in Gaussian noise. For this purpose the student will have to make usage of computing resources. The lab yields to a basic understanding of:

- time series data with stationary noise and signals
- statistical concepts (moments, median, higher moments)
- template placement, mismatch statistics, ROC curve
- sensitivity computational cost analysis
- paralelization with a computing cluster using Condor

The practical work is divided into four exercises. A guide complemented with useful links and documentation is given. In order to proceed to the next exercise, the student is asked to demonstrate solutions to short set of problems.

The lab is targeted to Bachelor or Master students in Physics. The student should have knowledge of:

- C programming language
- scripting language (Python is particularly useful)
- basic statistical working knowledge (mean, variance, higher moments), time series analysis

Exercise 3: Template Banks

As you may have seen in the last exercise, the statistical variables (S_1, S_2, S_3) vary with the signal parameters, having a maximum at the right ones. Knowing the waveform of the signal we can compare our data with a discrete set of templates, each of which represents a possible signal. The set of parameters defining a signal is known as a template, and the set of different templates is called a template bank. When using template banks we have to answer the next three questions:

1. Which parameters significantly affect the waveform?
2. How should the spacing of the template parameters be chosen?
3. How many templates are needed to cover the parameter space?

Because we have a finite number of templates, each point in the parameter space will have a finite distance to the nearest template. This is also the case for possible signals. Therefore we can define a mismatch between points of the parameter space and the nearest template:

$$M_i = 1 - \frac{S_i(\Delta\vec{l})}{S_i(0)}, \quad i = 1, 2, 3. \tag{1}$$

Where $\Delta\vec{l}$ is the distance vector between the point in the parameter space an the nearest template, and $S(0)$ is the value of S on the right parameters of the signal.

The maximum mismatch between any point in the parameter space and the closest template must not exceed a given limit, which depends on the particular signal. This only can be achieved by a minimum number of templates. On the other hand, we try to keep the number of templates as low as possible. This problem is the same than the sphere covering problem, since we are assuming that each template covers a given radius of the parameter space. We want to cover the whole space of parameters, in order not to miss the signal, with the smallest number of spheres. That is, we want to reduce its overlap lowering then the computational cost as well, since we are not using almost the same template more than once.

There are several kinds of template banks. For simplicity we will use square lattices, but it is important to know that for higher dimensional parameter spaces it is usually better to use a random or an stochastic one. A random template bank consists of a given number of templates placed randomly with uniform probability distribution over the parameter space. While a stochastic template bank adds a filter to the last method. The basic procedure is to select a point randomly of the parameter space, if its distance to any other point is greater than a certain value add this point to the template bank (which is initially empty). Then this procedure is repeated until the list of templates stops growing or any other criterion is met.

Goal

Retrieve the right parameters of the signal injected using template banks.

Guidance

1. Inject multiple signals with different parameters and arrive to a mismatch graph (that is plot the probability of mismatch in a histogram) to evaluate template bank properties. You can introduce template banks files into *prober* (for more information use $--$ *help* command). Using Condor is highly recommended.

2. Inject a random signal setting -r to 1 with *generate_source*. Arrive to a ROC curve analysing multiple signals.

3. Repeat with coarser and finer template banks. Obtain the parameters of the injected signal and compare the results using different tests and different template banks.

Useful links

- Stochastic template placement algorithm for gravitational wave data analysis - I.W. Harry, B. Allen and B.S. Sathyaprakash, `http://journals.aps.org/prd/pdf/10.1103/PhysRevD.80.104014`

- Random template banks and relaxed lattice coverings - C. Messenger and R. Prix, `http://journals.aps.org/prd/pdf/10.1103/PhysRevD.79.104017`