Final Assignment: Project Blind Search

There are different kinds of searches depending on the amount of a priori knowledge. When the possible values for each parameter are well known, we can afford a well-resolved and deep search. The parameter space increases with the amount of unknown or weakly narrowed down parameters, and so does the computational power necessary to do the search. The worst cases are the so-called *blind searches*, where only vague physical constraints are set for the parameters, resulting in a vast parameter space and an enormous cost of computation.

In this project we want to do a blind search on data with unknown injections. There are only 2 parameters, frequency f and phase ϕ, for which we know the upper and lower bounds only: $f \in [10, 1000]\,\mathrm{Hz}$, $\phi \in [0, 2\pi]$.

In the case of blind searches it is especially important to make good use of the available computational power. We do this by doing a prior analysis of runtime and expected signal loss, the *mismatch*, due to our search grid, the *template bank*. Therefore we optimize our search to have a reasonable runtime while not being too coarse in resolution to miss any signal.

In the previous exercises we investigated the mismatch of our signal for being off-target due to, for example, a discrete grid. With these studies we could estimate our loss due to different grid spacings which immediately gave us the template count of the full search. With a runtime analysis of our analysis tool *prober* we could then estimate how long our search will perform for a given grid spacing.

In Exercise 1 we saw among other things what statements we can infer from a given detection statistics value, the *prober* return value, and how likely it was due to noise or due to a signal.

Combining all these ingredients we can set up and do our blind search.

Goal

Find all injections and make statements about possibility of a real detection or false alarm.

Guidance

1. You'll be given one instance of data with known properties ($T_{obs} = 180.0$, dt $= 1/4096$, $N = \frac{180}{\mathrm{dt}}$, Sigma $= 25.0$) and an unknown number of injections with unknown parameters. We have very little knowledge on our signal model, so we expect signals in the range of $f \in [10, 1000]\,\mathrm{Hz}$, $\phi \in [0, 2\pi]$ and $A \in [0.5, 2]$, where A is the amplitude.

2. The data will likely have different (N, dt, Sigma) to your previous datasets, thus firstly you have to determine a suitable setup. For this do the Monte-Carlo studies as in Exercise 3: Iteratively do multiple injections, search around a small box and notice the differences in frequency, phase and the mismatch $(df, dp, M)_i$ between these points and an optimal recovery on-target.

3. After you have the list of $(df, dp, M)_i$, you can calculate with above parameter ranges the total amount of templates N_{stage0}.

4. You have to do a runtime analysis. We saw there is a linear increase with number of templates, so a few points with reasonable large template counts will suffice to get a good prediction. After that you can calculate the predicted runtimes for the average mismatches and choose the best found setup for a runtime you can afford to spend.

5. Now take a look at Exercise 1: Do Monte-Carlo studies to arrive at a ROC curve where you can gain insight into your expected thresholds in terms of *false negatives* and *true positives*.

6. Choose threshold $\mathcal{T}_c$ for follow-up and cluster all candidates above threshold. Since we deal with only 2 parameters and we expect our recovery in phase to be very unsharp, we may cluster only in frequency, that is define a start- and end-frequency $f_{c,\mathrm{s}}, f_{c,\mathrm{e}}$ for each cluster. Our *follow-up* stages search space will thus be the full $\phi \in [0, 2\pi]$ and the union of all cluster-frequency-regions $f \in \bigcup_c (f_{c,\mathrm{s}}, f_{c,\mathrm{e}})$ in frequency. This is much smaller than our initial search region.

7. Since we don't have any more data and our refinement is probably maxed out, we note the loudest candidate per cluster and determine with our ROC curve how probable a detection is. Note the chance of *false negatives* and *true positives*.

8. Publish results: Prepare a document in which you give the necessary information:

   (a) Each stages search setup (df, dp, Mismatch&Runtime)

   (b) ROC curve

   (c) Threshold used

   (d) Properties of candidates above threshold (f, ϕ, $\pm$ errors): Please include a text file in the following form: "#Frequency FrequencyError Phase PhaseError" which will be used to determine the amount of *true positives* and *false negatives*. For the errors just plot images of each cluster and eyeball your errors.

9. Upload your Paper and source code (jupyter notebook or otherwise) to your github repository before the end of the semester: **23rd August 2022**. After upload email me at alexandra.botnariuc@aei.mpg.de with your full name, study course and immatriculation number.