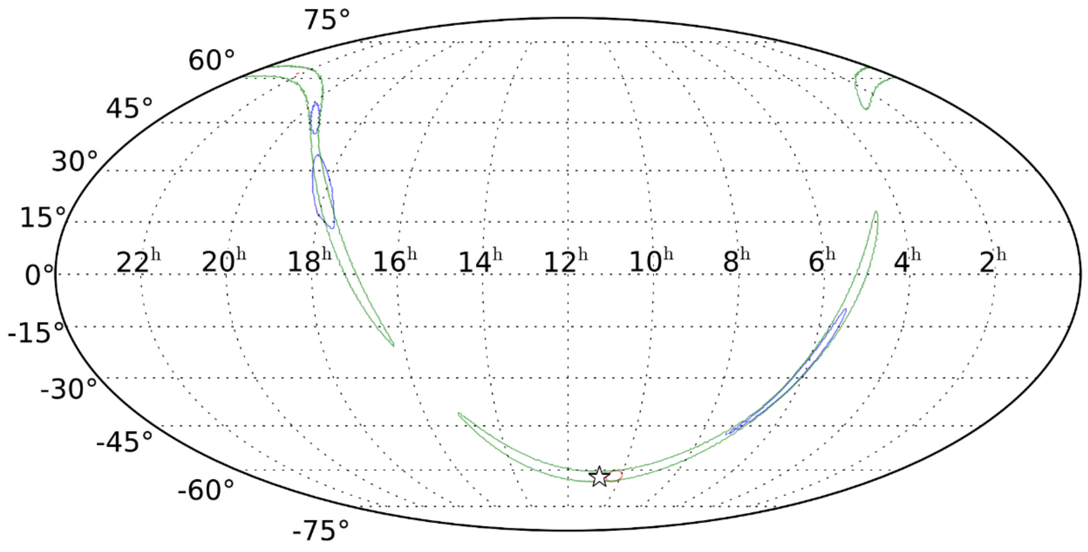
Localization of coalescing binaries with a hierarchical network of gravitational wave detectors

Work report at LAPP

Yoshinori Fujii

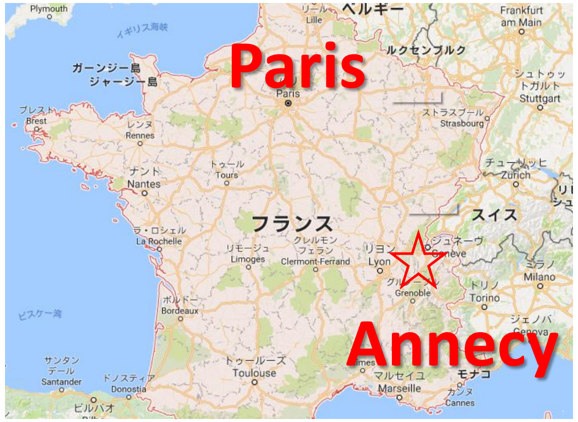
This work is mainly supported by Frederique Marion, Thomas Adams and MBTA team (especially in LAPP)

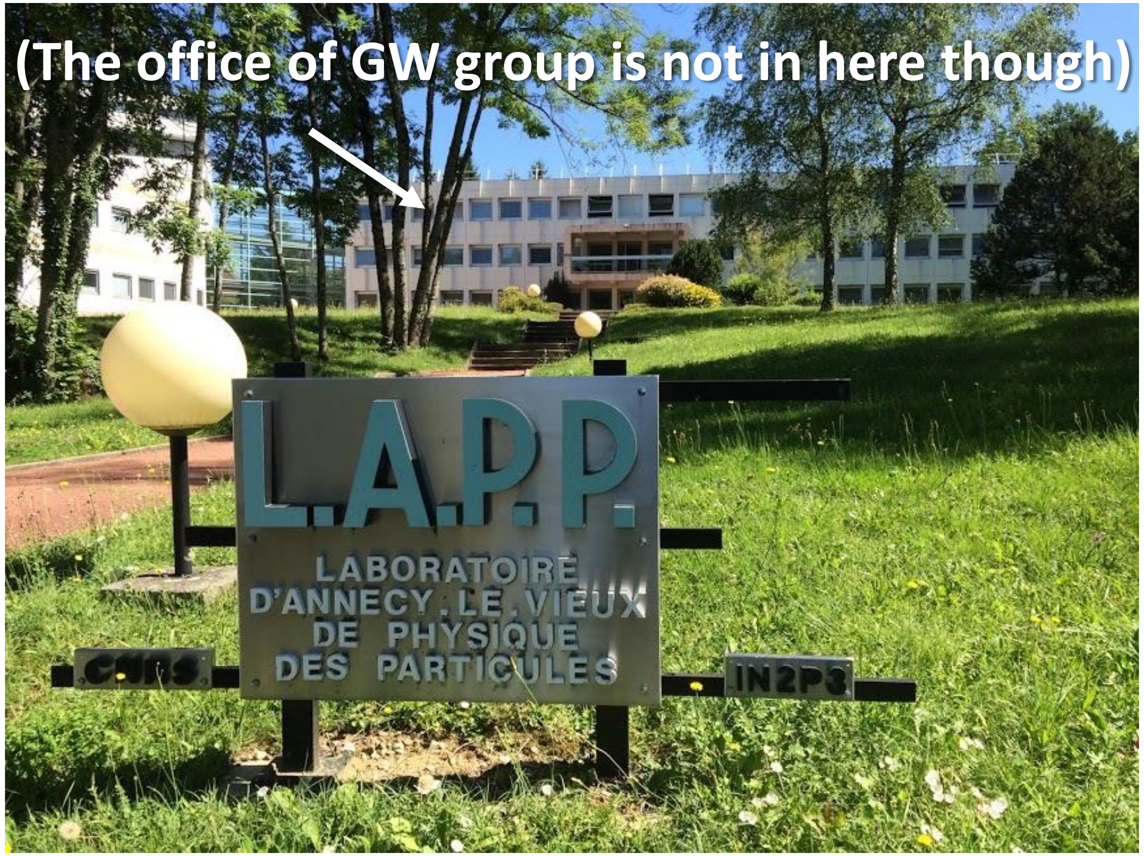


Contents

Introduction : LAPP

Laboratoire d'**A**nnecy-le-vieux de **P**hysique des **P**articules





ATLAS | LHCb | Neutrino

AMS | H.E.S.S. / CTA | Virgo

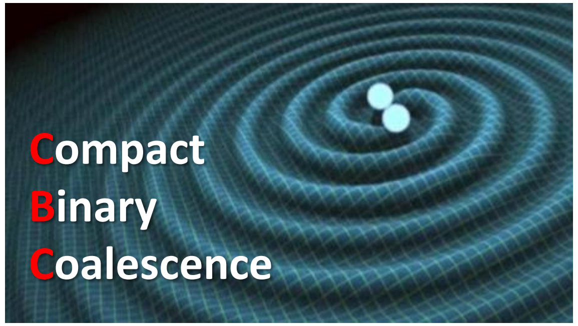
Development of low-latency GW search pipeline etc..

My work at LAPP was mainly about data analysis (not Vibration Isolation System)

Introduction :

Laboratoire d'Annecy-le-vieux de Physique des Particules

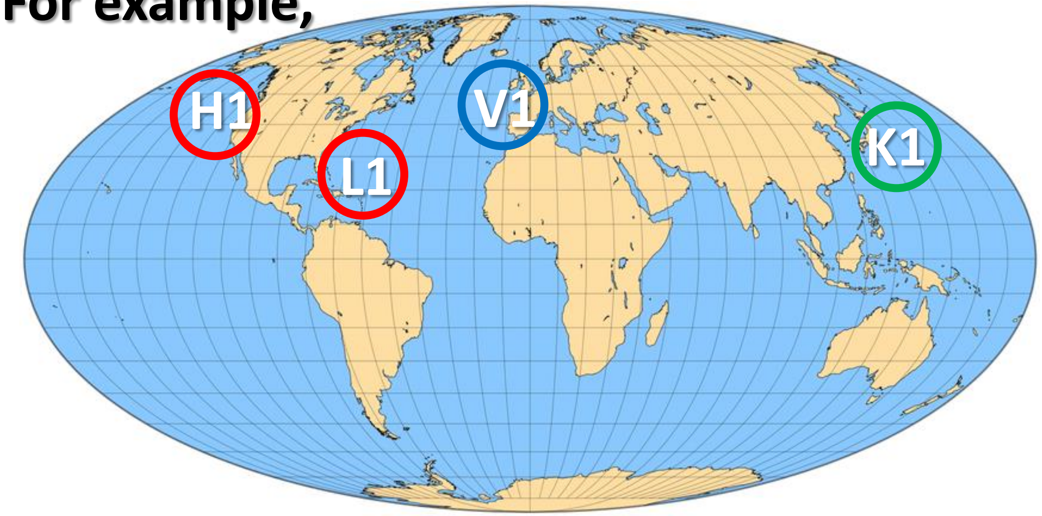
Topic :
how newly constructed detectors should enter the detection network?
(in low-latency CBC search)

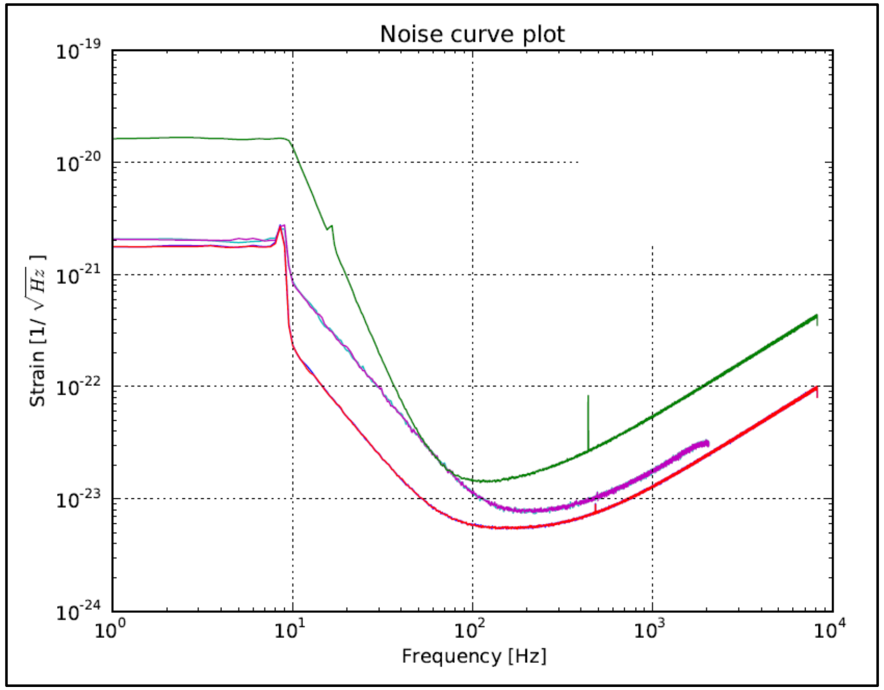


Introduction :

Several detectors are needed for source localization (detection network)

For example,





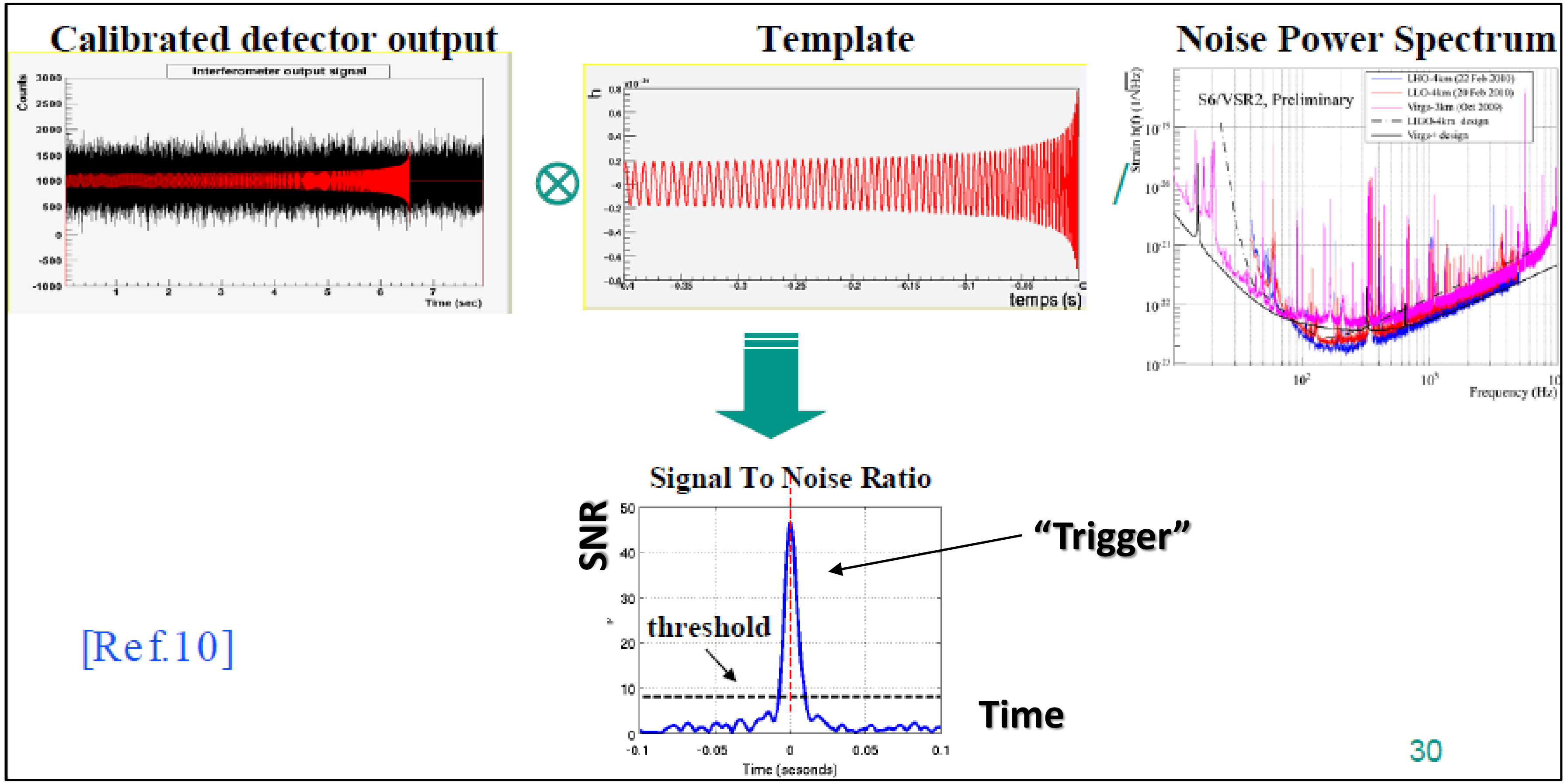
The sensitivities of these detectors would be different from each other, especially just after their construction.

(ex. in observation 2 (O2), the higher sensitive 2 LIGOs, and the less sensitive Virgo)

➡ **In the Virgo or KAGRA, GW signals can be buried into noise easier than in LIGOs!**

Introduction :

Especially, in the low-latency search for Compact Binary Coalescence (CBC)



According to
matched filtering,
time series of
Signal to Noise Ratio (SNR)
are generated.

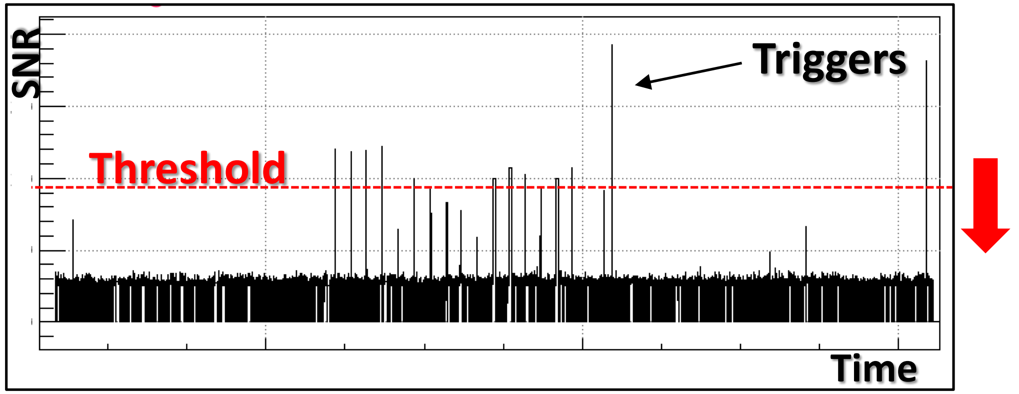
The generated SNR
from less sensitive detectors
whould be smaller than
the SNR from high sensitive
detectors.

http://old.apctp.org/conferences/2011/NRG2011/NRGPDF/CBC_DA_Korean_School_2011.pdf

The detection threshold SNR of less sensitive detectors are wanted to be lowered..

Introduction :

However,
if the threshold SNR is purely lowered,
we have to handle tons of the triggers



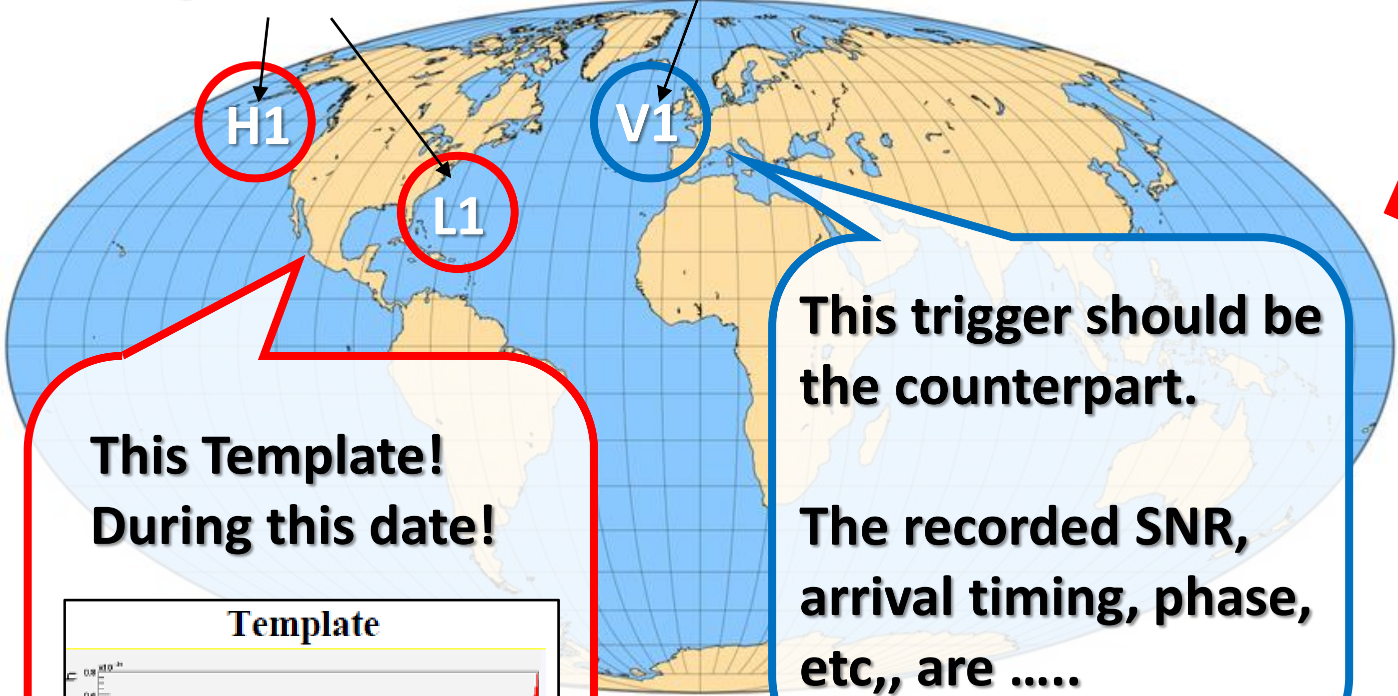
→ Computational cost and time cost get large.. Not low-latency, anymore!

How about including the less sensitive detectors into the network,
1. with lower threshold SNR than that of higher sensitive detectors, but
2. only when we search triggers, generated from
higher sensitive detector's coincidences.

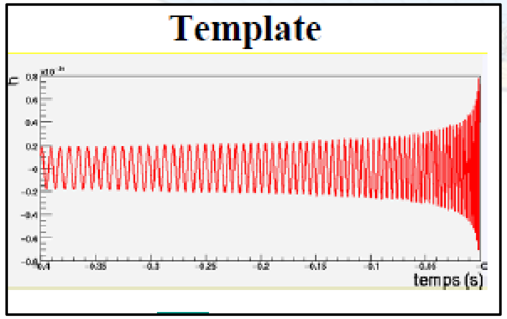
Introduction : “hierarchical search”

High sensitive

Low sensitive



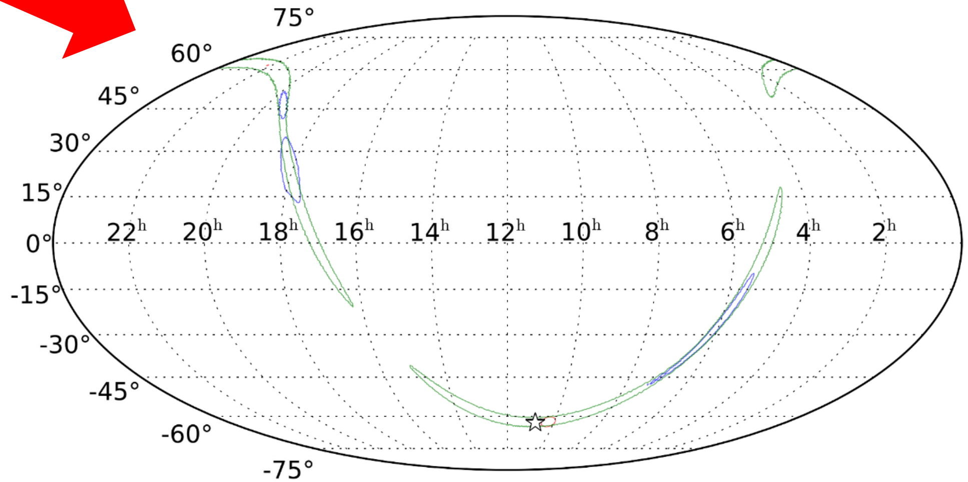
This Template!
During this date!



This trigger should be the counterpart.

The recorded SNR, arrival timing, phase, etc,, are …..

Combining
3 (or more) detector information



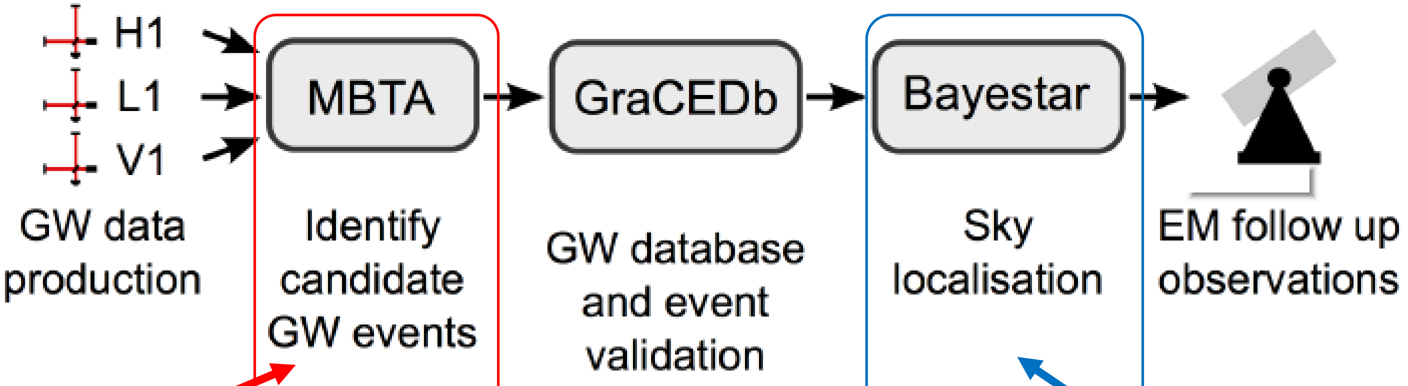
EM partners

Even if the threshold is lowered,
no need to look for all the triggers → saving time

Contents

GW-EM follow up pipe line for low-latency CBC search :

https://arxiv.org/pdf/1512.02864.pdf



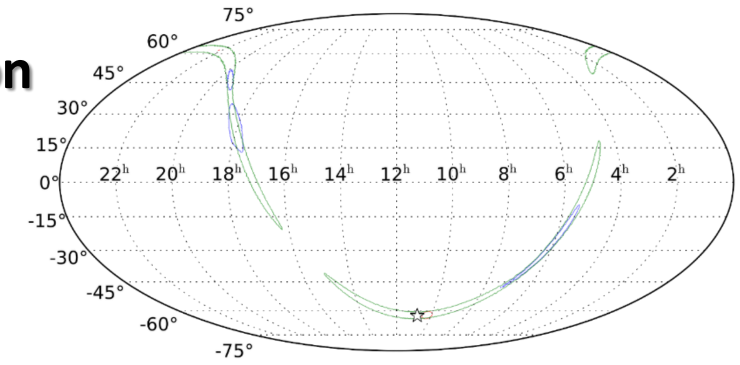
Multi-Band Template Analysis

Report results of
1. Matched filtering
2. Veto cut
3. Data quality check
4. Identification of coincident triggers

...

BAYESian TriAngulation and Rapid localization

Plot “sky map” from MBTA output information



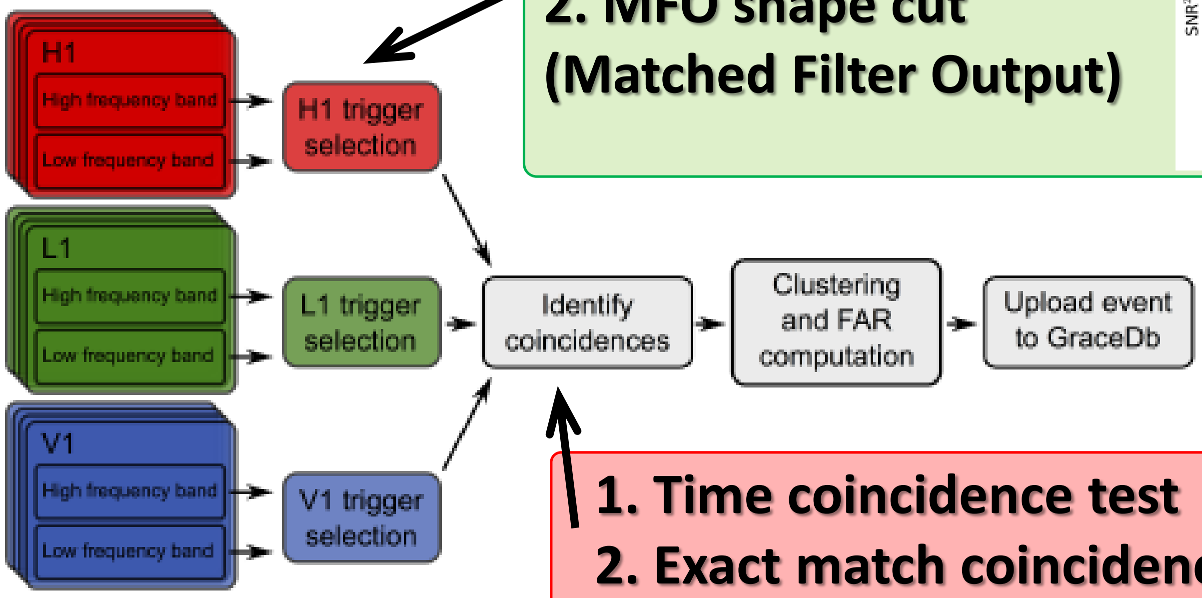
(Detailed information is following in next page.)

Multi-Band Template Analysis (MBTA)

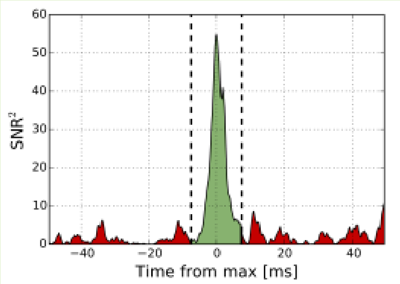
→ Split the matched filter across two (or more) frequency bands.

→ Shorter templates in each frequency band
→ Phase of the signal is tracked over fewer cycles.
⇒ Computational cost reduction

→ Smaller sampling rate for low frequency band
⇒ FFT cost reduction



1. χ^2 cut
2. MFO shape cut
(Matched Filter Output)



1. Time coincidence test
2. Exact match coincidence test
(masses, spin are same in all the detectors?)

BAYESian TriAngulation and Rapid localization (BAYESTAR)

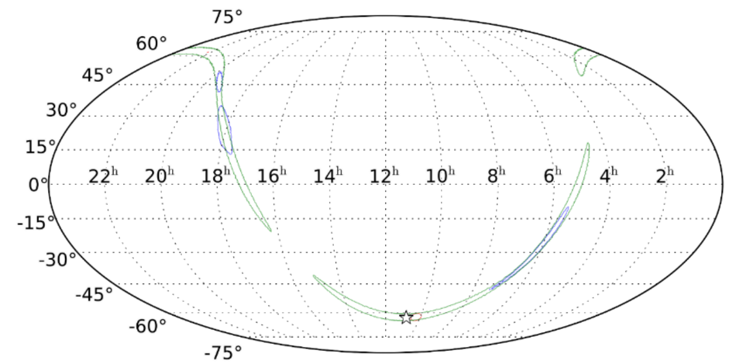
1. Arrival timing
2. SNR
3. Phase

4. Noise curve

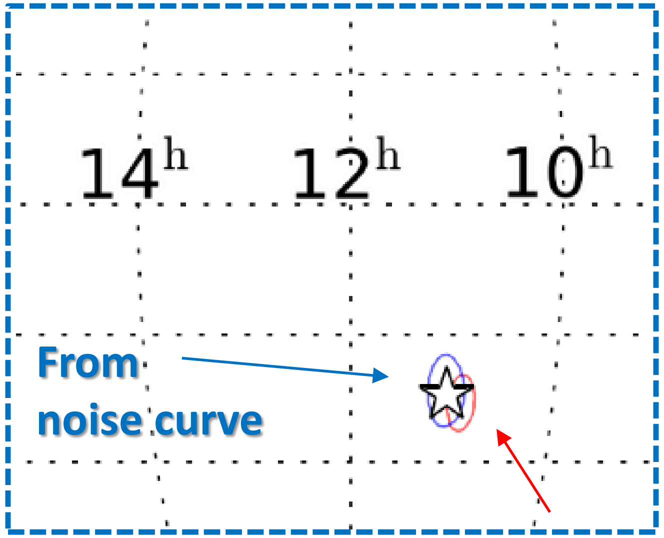
(of each detector)

⇨ **BAYESTAR** ⇨

"Sky map"



Plot probability



If arrival timing is not correct.

Contents

Purpose of this work : in the hierarchical search with HLV,

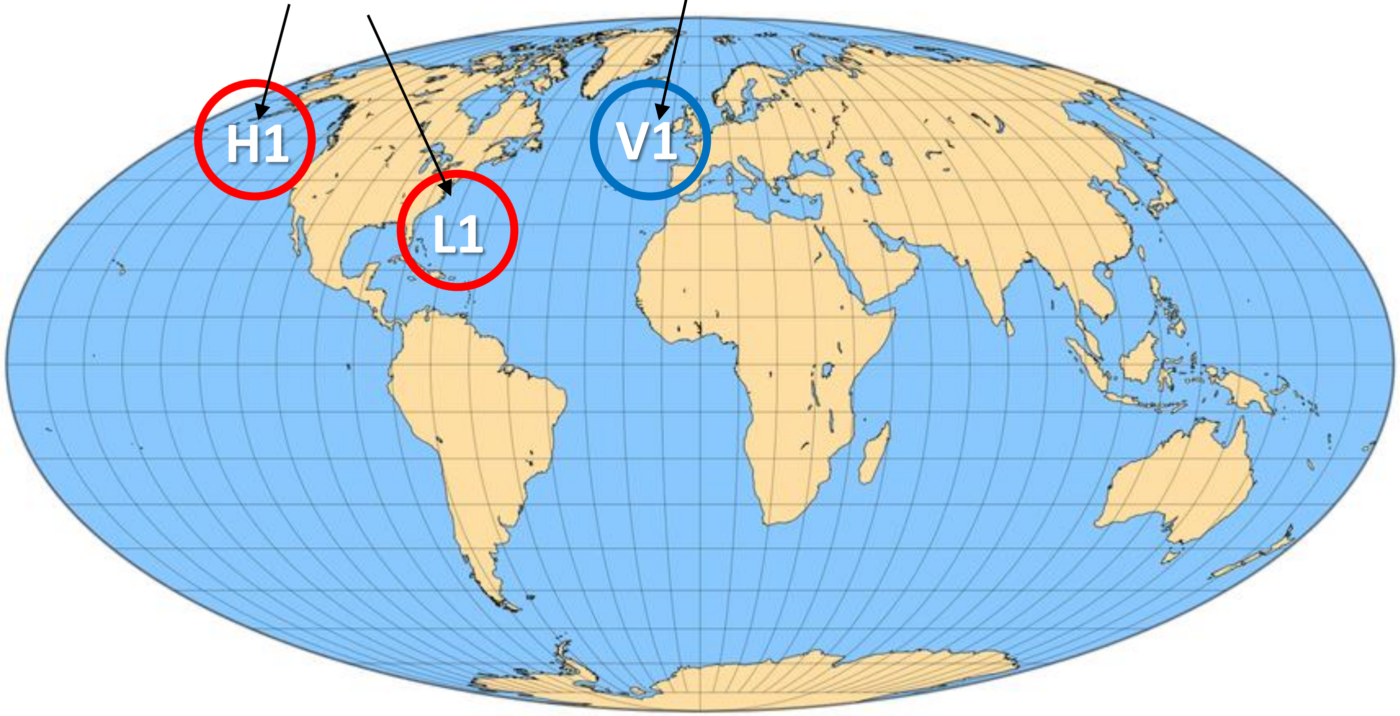
1. What is the optimal threshold for the V1?
2. How the localization gets improved at the threshold?

High sensitive

H1, L1 $\equiv$ 70 Mpc (*)

Low sensitive

V1 $\equiv$ 20 Mpc (*)



(* at $1.4M_\odot - 1.4M_\odot$ Binary Neutron Star)

To answer these questions,

1) Prepare injection set
2) Suppose inputting them
3) Investigate re-constructed sky map

Measures of localization performance:

→ "Offset angle", "Searched area"

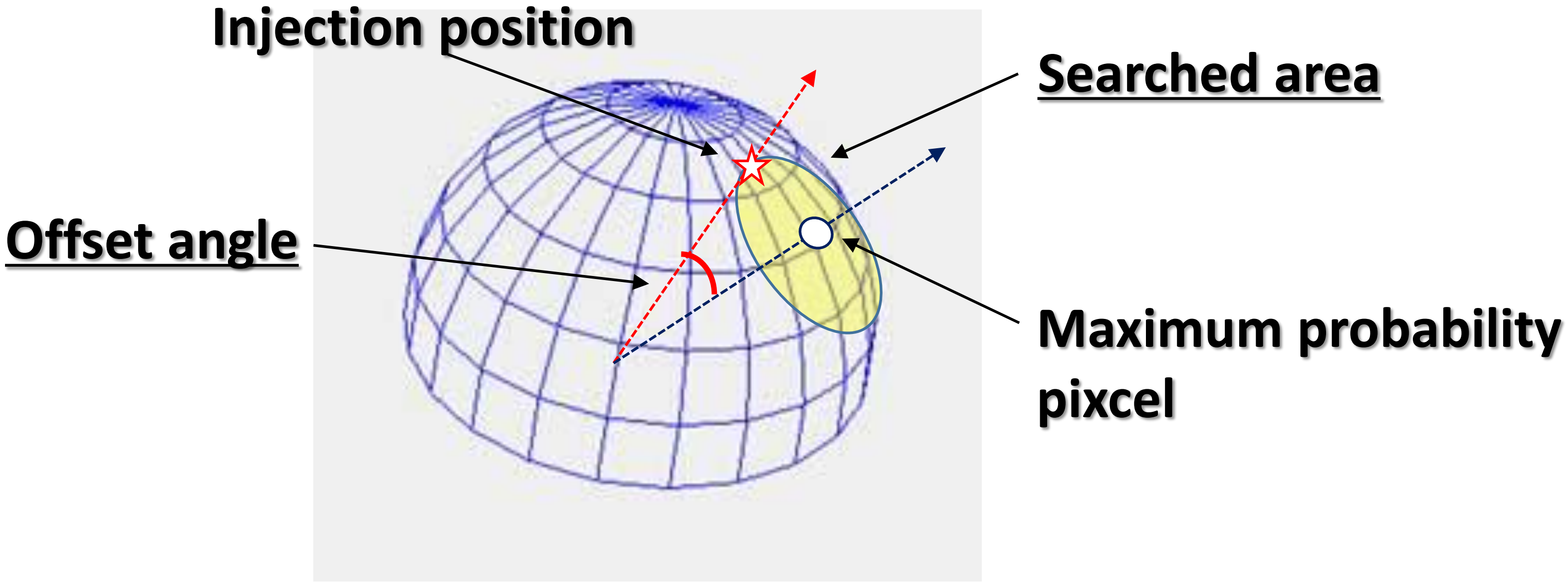
Definitions of the offset angle and the searched area :

1. Offset angle:

Angle between the sky localization of the injected signal, and the reconstructed max probability pixel.

2. Searched area :

The smallest area of the highest confidence region around the max probability pixel, to include the sky location of the injected signal.

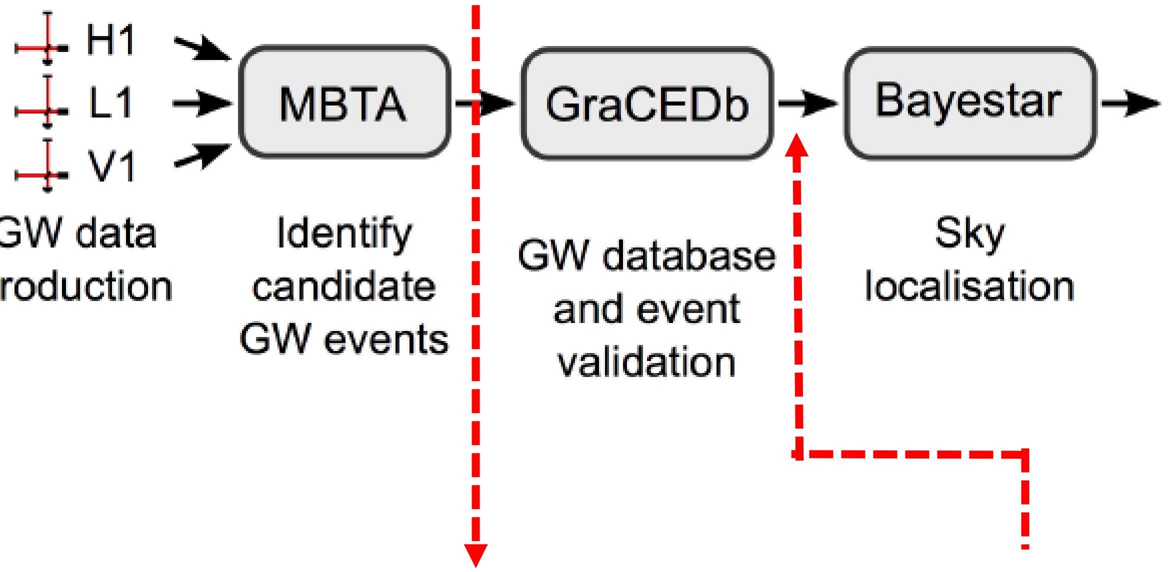


Searched area

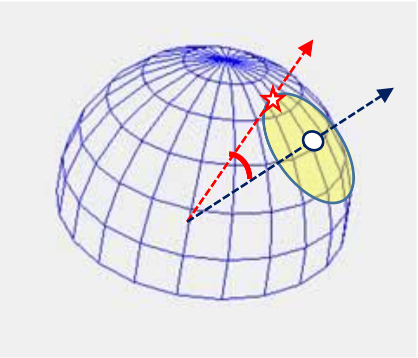
Ex. If the injected position is at a pixel of probability 0.7, the searched area is all the sum of the pixels which larger probability than 0.7.

Contents

Calculation setup : Main flow



3. Re-construct sky map



Localization performance

How the localization gets improved?

1. Prepare injection set

Existed 248 MBTA outputs, obtained from HL double coincidences (generated from previous software - injection test)

2. Suppose inputting them

Transform HL → HLV triggers adding artificial V1 information (SNR, timing, phase)

Calculation setup : How to transform the triggers, HL → HL or HLV

Considered 3 patterns :

Case 1 var. : HL → HL or HL + random V

If p > FAP, otherwise

Suppose the V1 triggers from noises

→ Worst case

Case2 : HL → HL, or HL + V based on injection

If V1 SNR < threshold, otherwise

Suppose the V1 triggers from signals

→ Best case

Case 3 : HL → HL, or HL + random V, or HL + V based on injection

If p > FAP and V1 SNR < threshold, If p < FAP , If p > FAP and V1 SNR > threshold

Suppose the V1 triggers from both of noises and signals

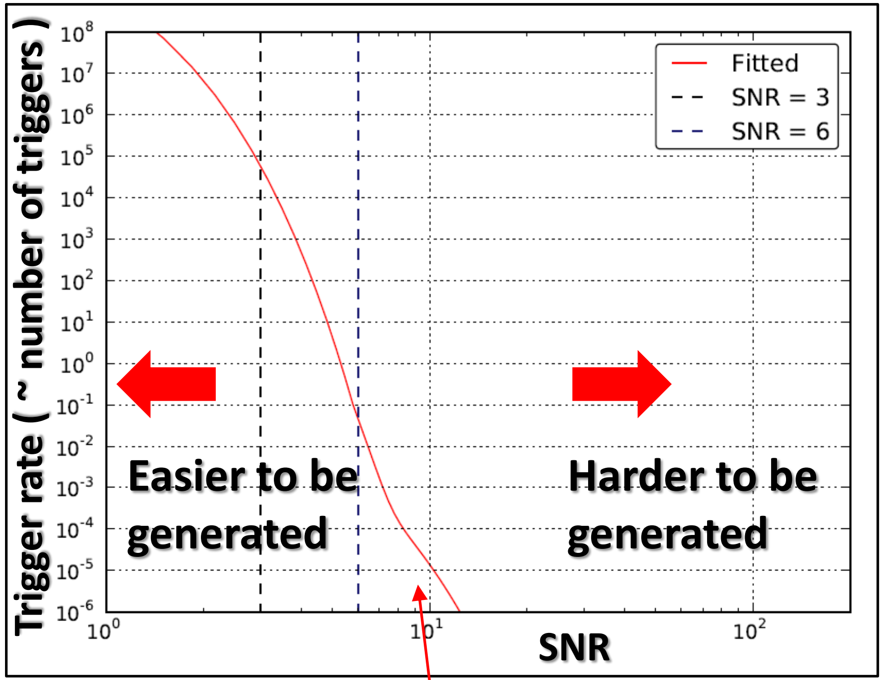
→ More realistic case

(How to generate the FAP, random V, V based on injection, is following)

Calculation setup : How to generate SNR, arrival timing, phase of the V1

1. "random V trigger : Vr"

1. SNR = Random above a threshold SNR, following measured O1 SNR distribution



Obtained from O1 measurement (next page)

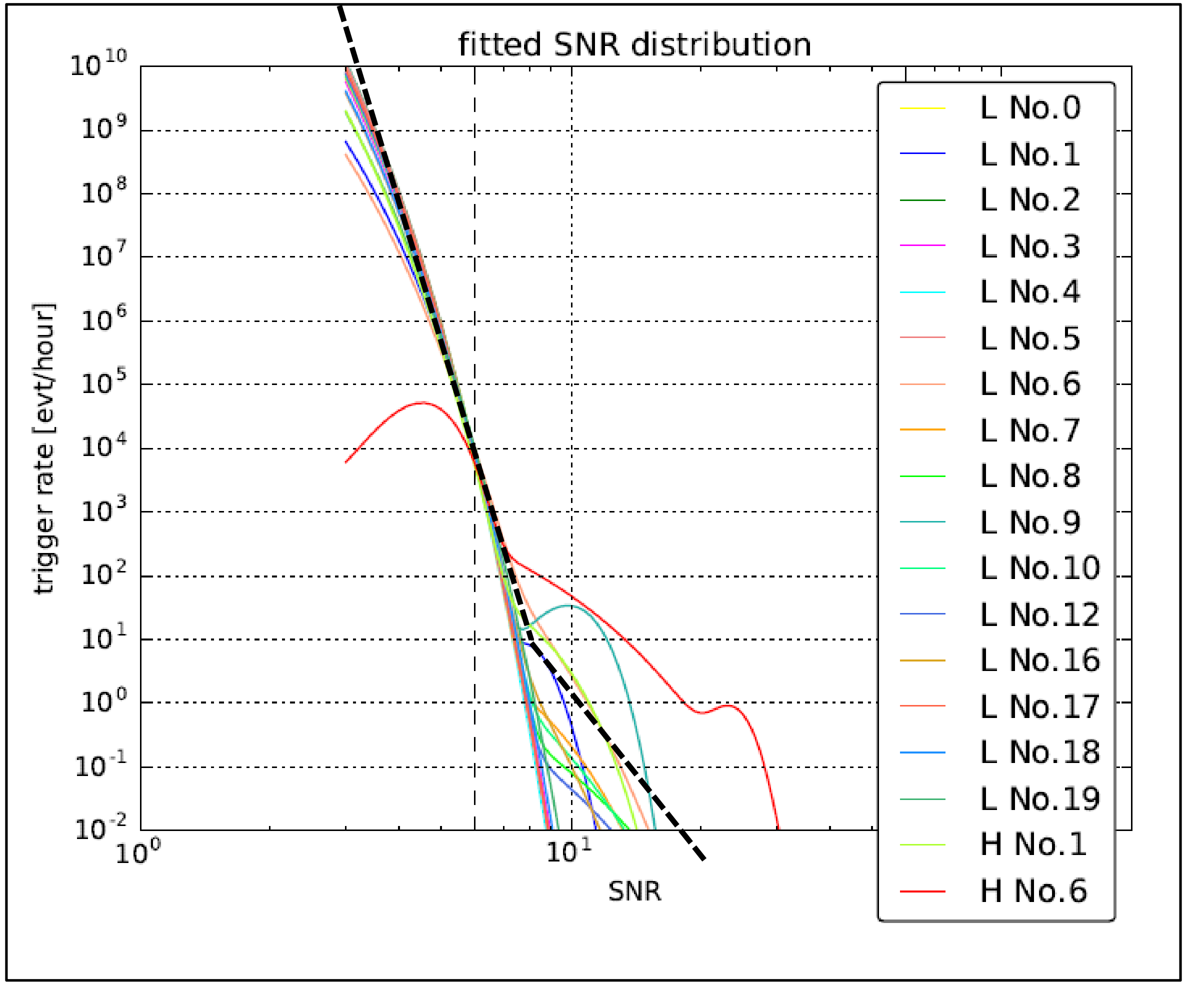
2. Timing = $t_0 + \Delta t$

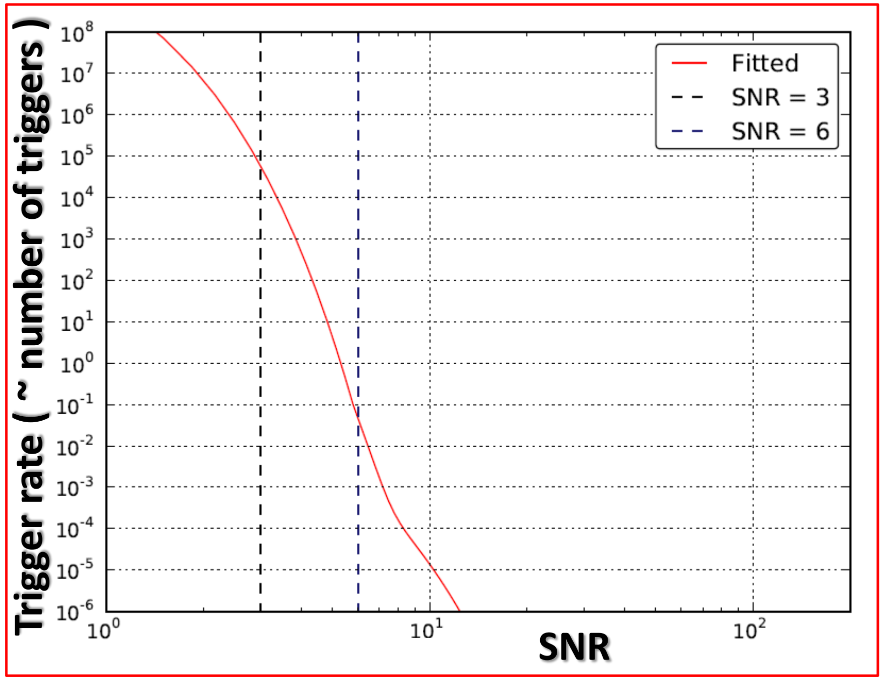
$t_0 = t_{H1}$ if $\mathrm{SNR_{H1}} > \mathrm{SNR_{L1}}$, otherwise $t_0 = t_{L1}$.

Δt = random uniform number from -35 ms to 35 ms.

3. Phase = random uniform number from 0 rad to 2 π rad.

Calculation setup : How to generate SNR, arrival timing, phase of the V1





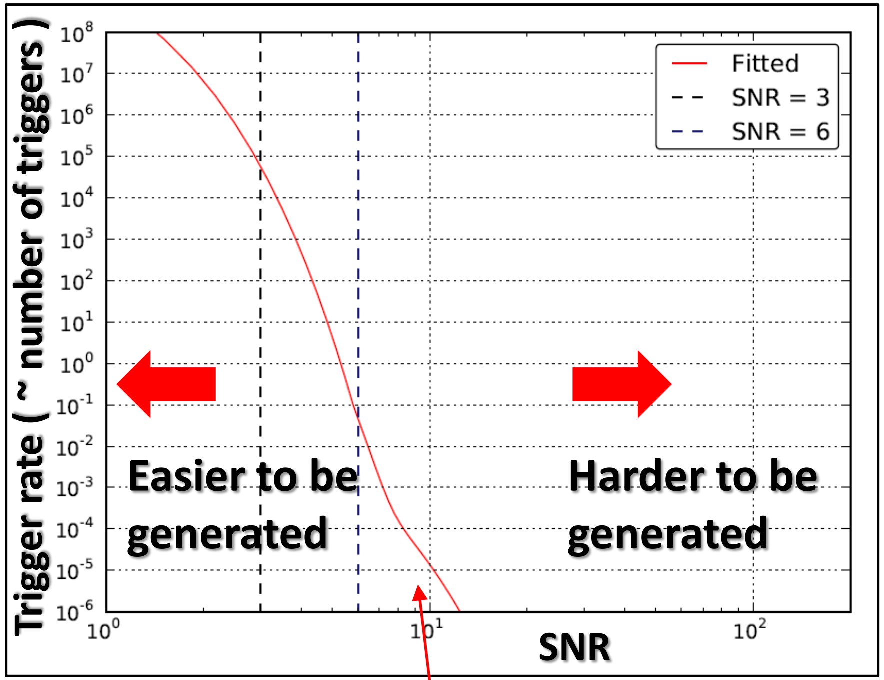
Plot SNR distribution from ~ about 20 hours data

→ Choose typical curve ("quiet")

Calculation setup : How to generate SNR, arrival timing, phase of the V1

1. “random V trigger : Vr ”

1. SNR = Random above a threshold SNR, following measured O1 SNR distribution



2. Timing = $t_0 + \Delta t$

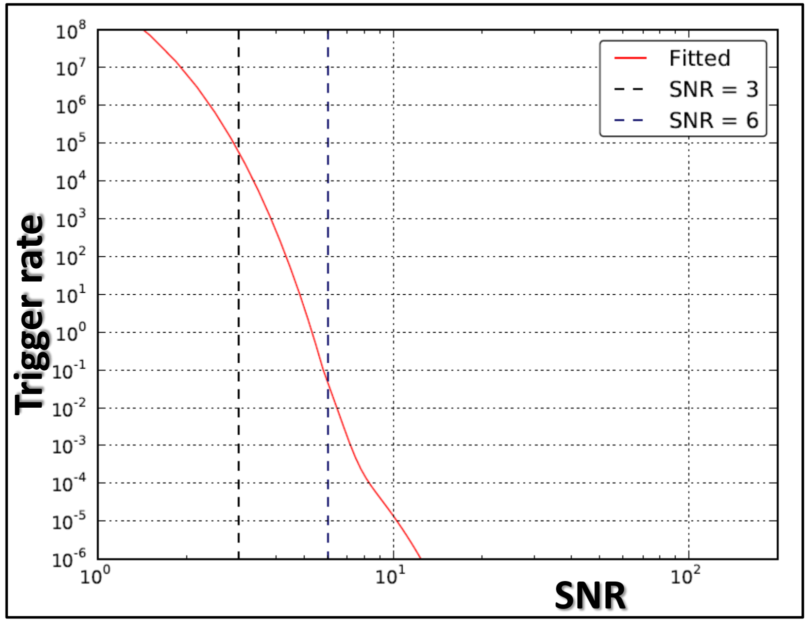
$t_0 = t_{H1}$ if $\mathrm{SNR_{H1}} > \mathrm{SNR_{L1}}$, otherwise $t_0 = t_{L1}$.
Δt = random uniform number from -35 ms to 35 ms.

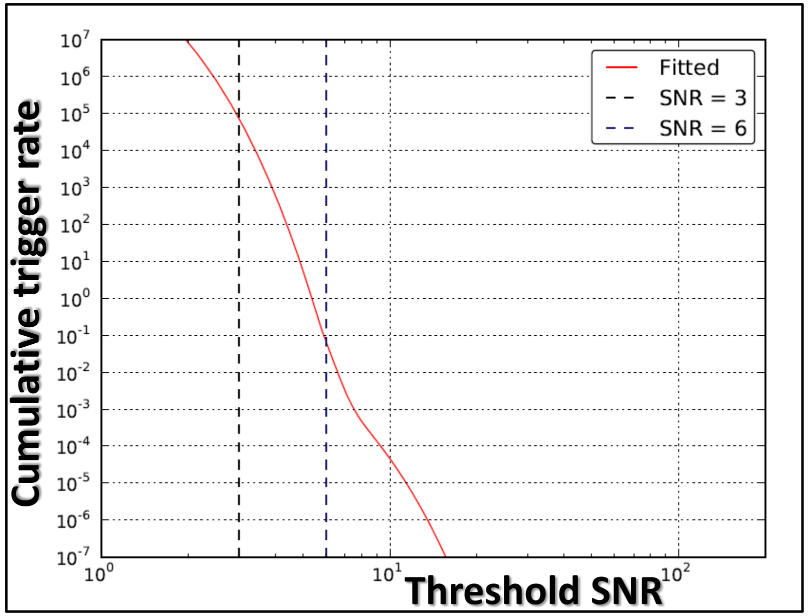
3. Phase = random uniform number from 0 rad to 2 π rad.

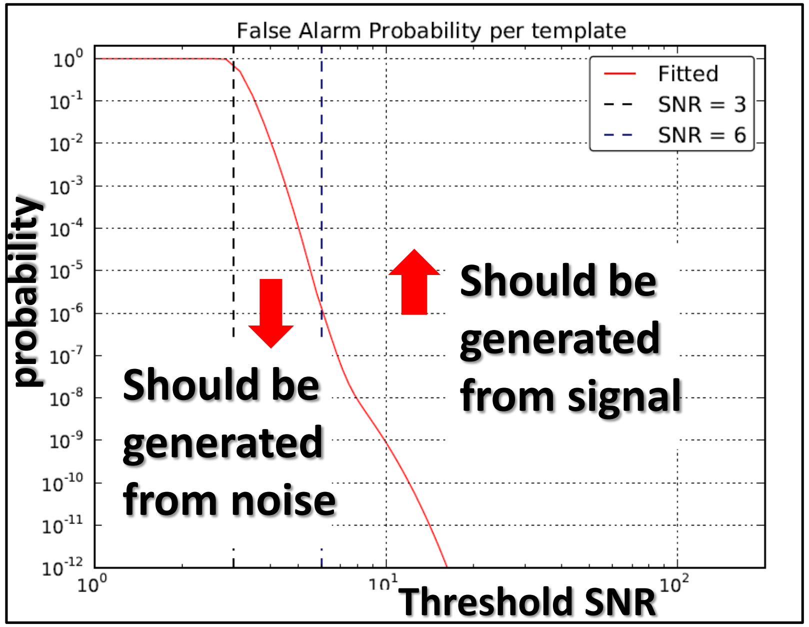
Obtained from O1 measurement (next page)

Calculation setup : False Alarm Probability (FAP)

SNR distribution (per template) ⇨ **Cumulative SNR distribution (per template)** ⇨ **False Alarm Probability (per template)**







$$\text{FAP} = 1 - \exp(-R \times T)$$

R = cumulative rate of background triggers per template, above a given threshold, per template,

T = analyzing time for the V1 (less sensitive detector)

70 ms for V1

Calculation setup : How to generate SNR, arrival timing, phase of the V1

2. “V based on injection : Vi ”

1. SNR = $\boxed{\mathrm{SNR}^{\mathrm{expected}}} + \boxed{\Delta\mathrm{SNR}}$

   $\mathrm{SNR}^{\mathrm{expected}}$ = from injection metadata

   $\Delta\mathrm{SNR} =$ Gaussian(0, 1)

2. Timing = $\boxed{t^{\mathrm{expected}}} + \boxed{\Delta t}$

   $t^{\mathrm{expected}} =$ injection meta data

   $\Delta t =$ Gaussian(0, 1 ms)

3. Phase = $\boxed{\phi^{\mathrm{expected}}} + \boxed{\Delta\phi}$

   $\phi^{\mathrm{expected}} =$ injection meta data

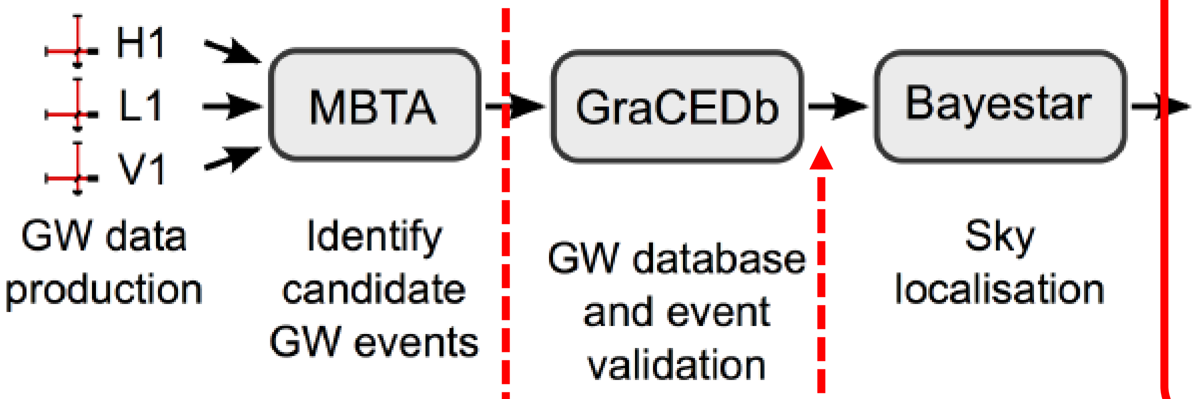
   $\Delta\phi =$ Gaussian(0, 0.25 rad)

These uncertainties are added to simulate more from realistic performance.
The typical values are used.

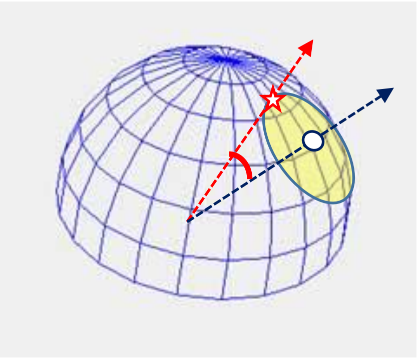
Contents

Review

Calculation setup : Main flow



3. Re-construct sky map



Localization performance

How the localization gets improved?

1. Prepare injection set

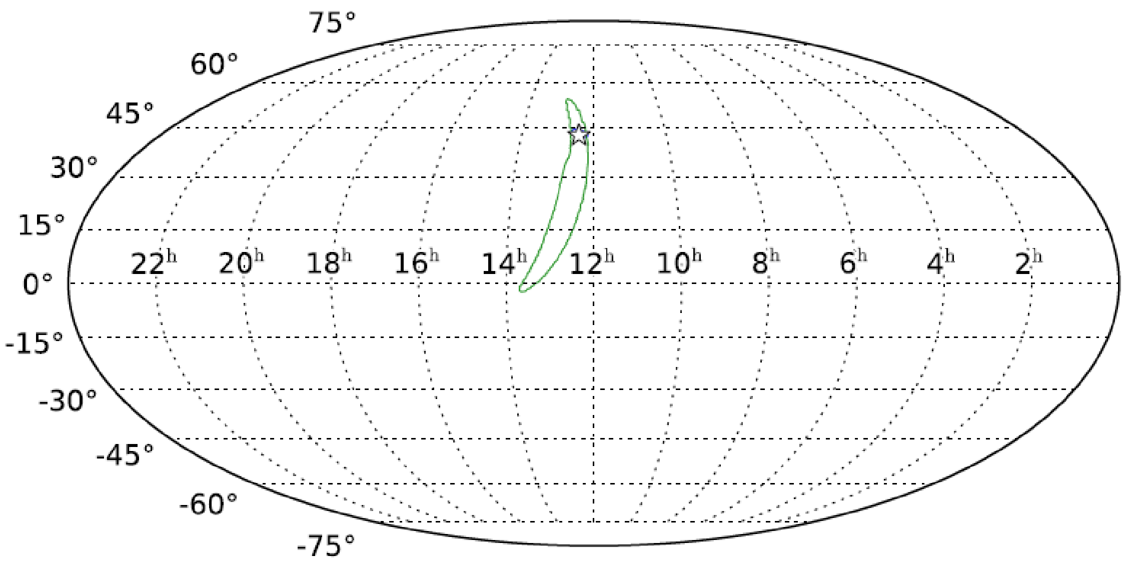
Existed **248** MBTA triggers, obtained from **HL** double coincidences (generated from previous injection test)

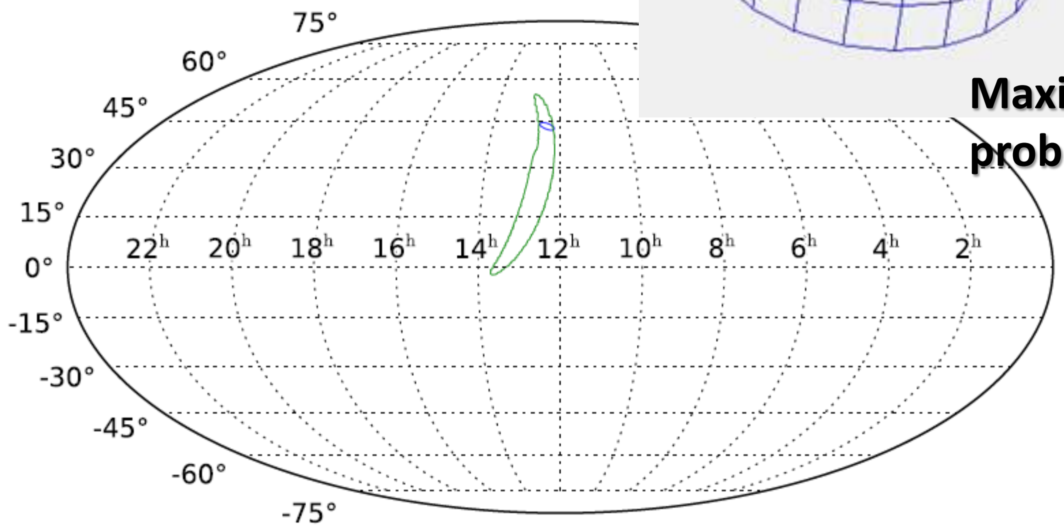
2. Suppose inputting them

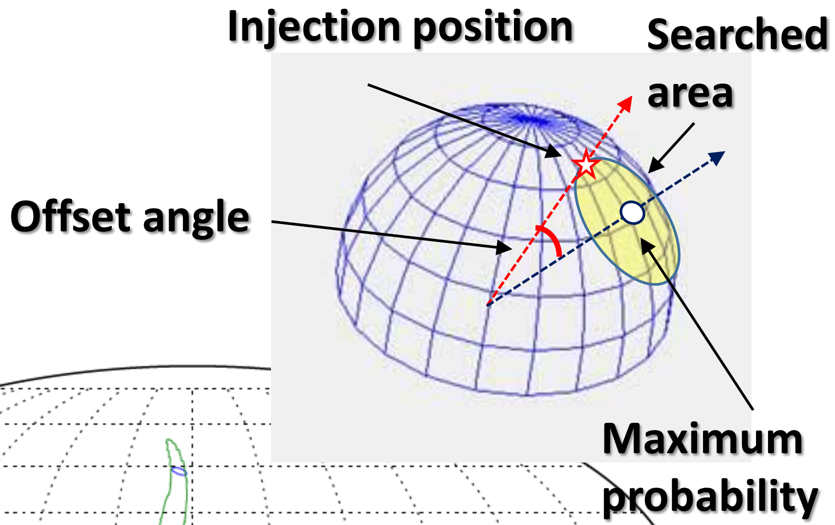
Transform HL → **HLV** triggers adding artificial V1 information (SNR, timing, phase)

Optimization of Virgo threshold :

Typical result : sky map





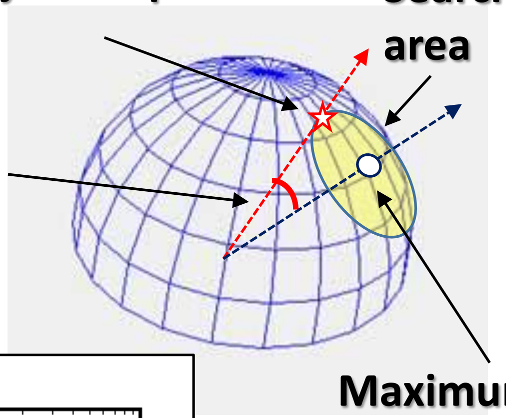


	Offset (deg)	:	Searched area (deg**2)
Double coincidence	24.64	:	98.96
Triple coincidence	0.781	:	5.27

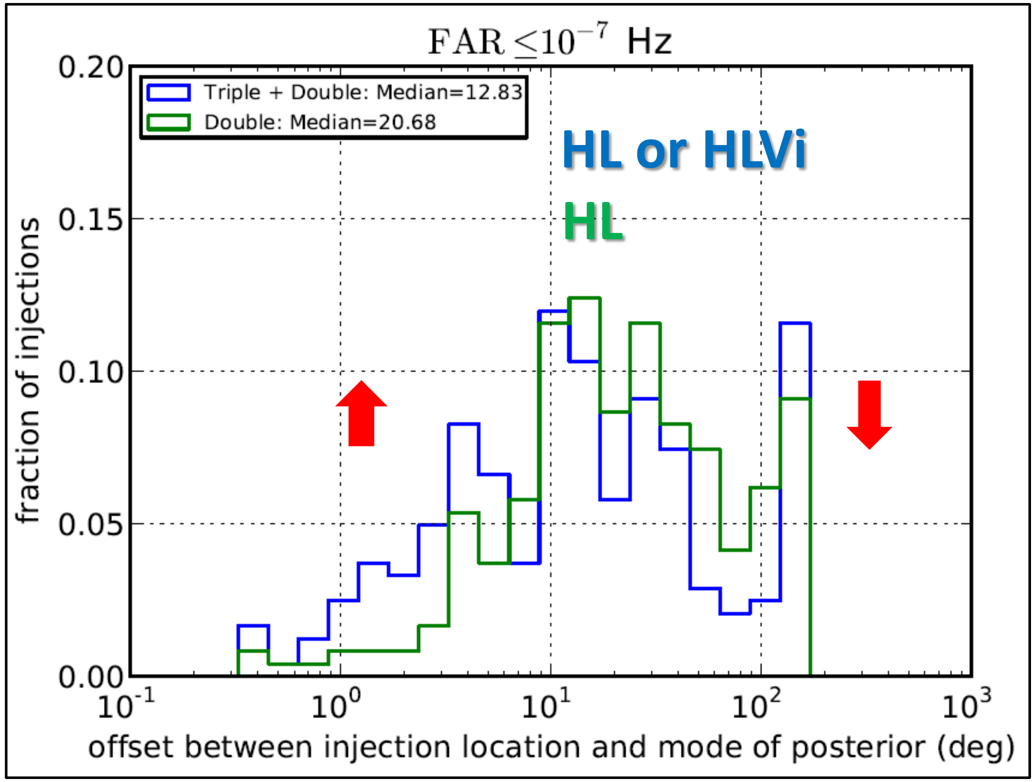
➡ Repeat 248 times, and collect the statistics of the offset angle and the searched area

Optimization of Virgo threshold :

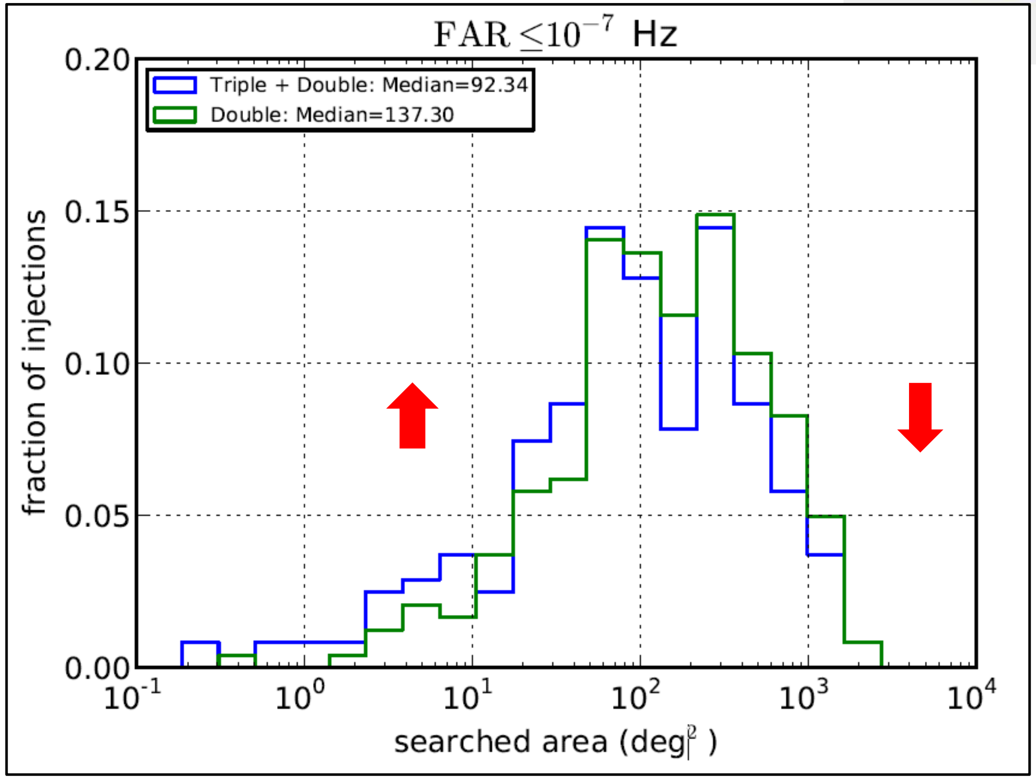
Typical result : statistics



For the “case 2” (Best case), when the V1 threshold SNR is set at 3.0.



Offset angle



Searched area

⇨ Seems to be improved.

Optimization of Virgo threshold :

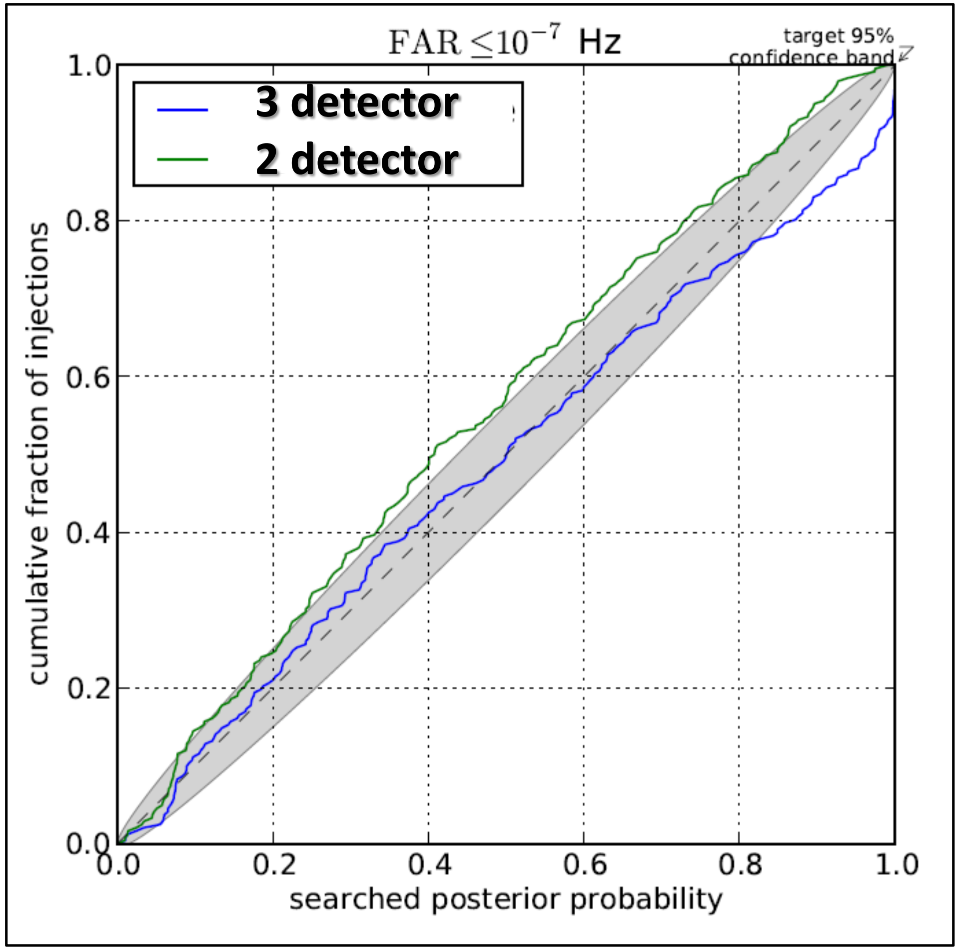
Typical result : Self-consistency test

→ **Probability - Probability Plot :**

90 % confidence area → 90% of injections should be included.

For the “case 2”, V1 threshold SNR is set at 3.0.

→ Number of injections



→ Certain confidence area

→ Localization depends on :
1. arrival timing difference
2. phase difference
3. relative SNR.

→ If the added uncertainties are properly, the curve should along with the diagonal line.

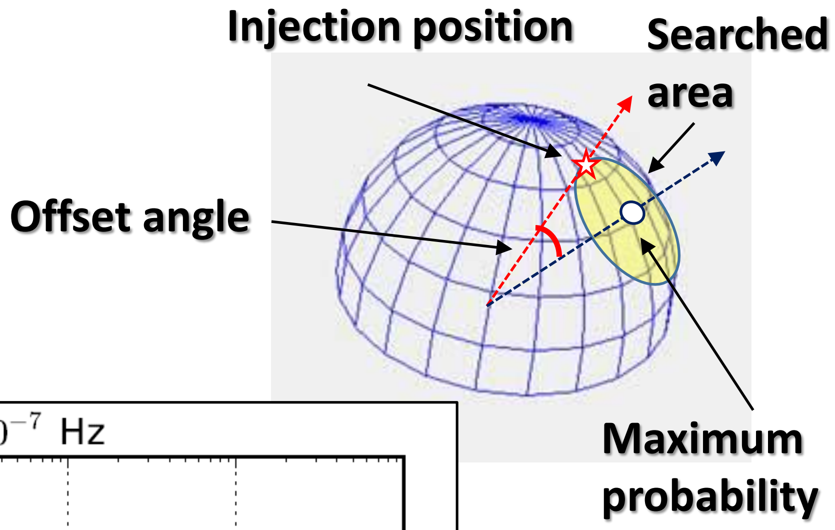
In this HLV search (blue), the curve gets below the diagonal line a little bit.
→ The added uncertainties are not crazy (though a little bit not realistic).

2. Timing = $t^{\text{expected}} + \boxed{\Delta t}$

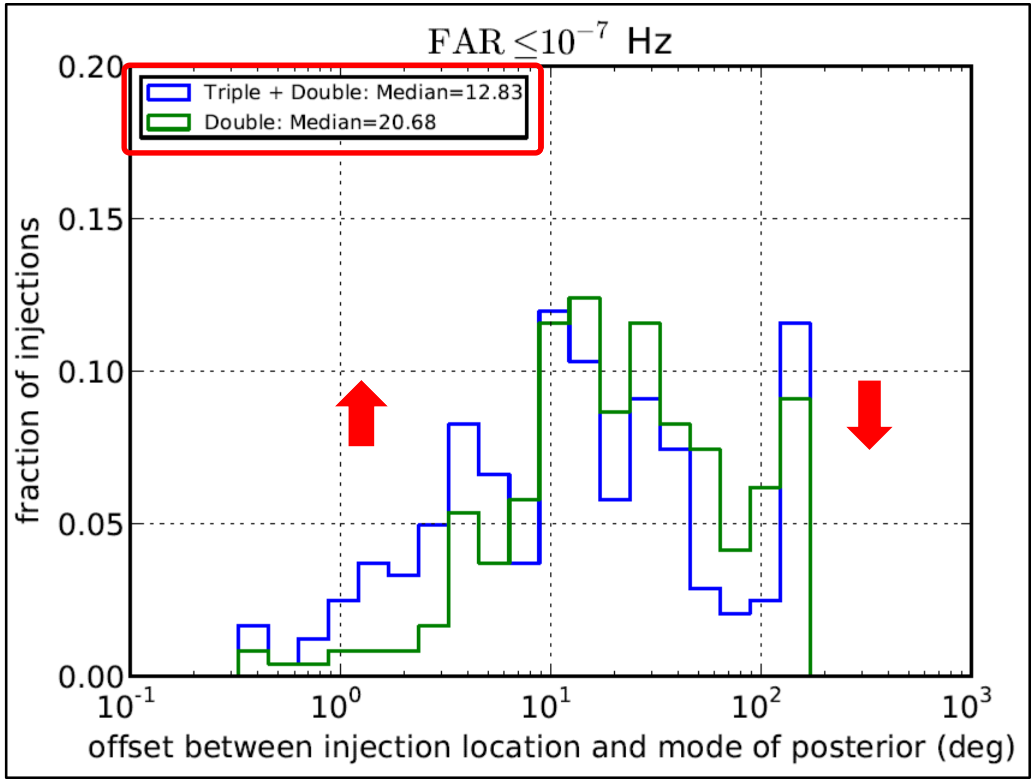
Δt = Gaussian(0, 1 ms) → 1 ms $\times \frac{6}{\text{V1 SNR}}$ etc. ?

Optimization of Virgo threshold :

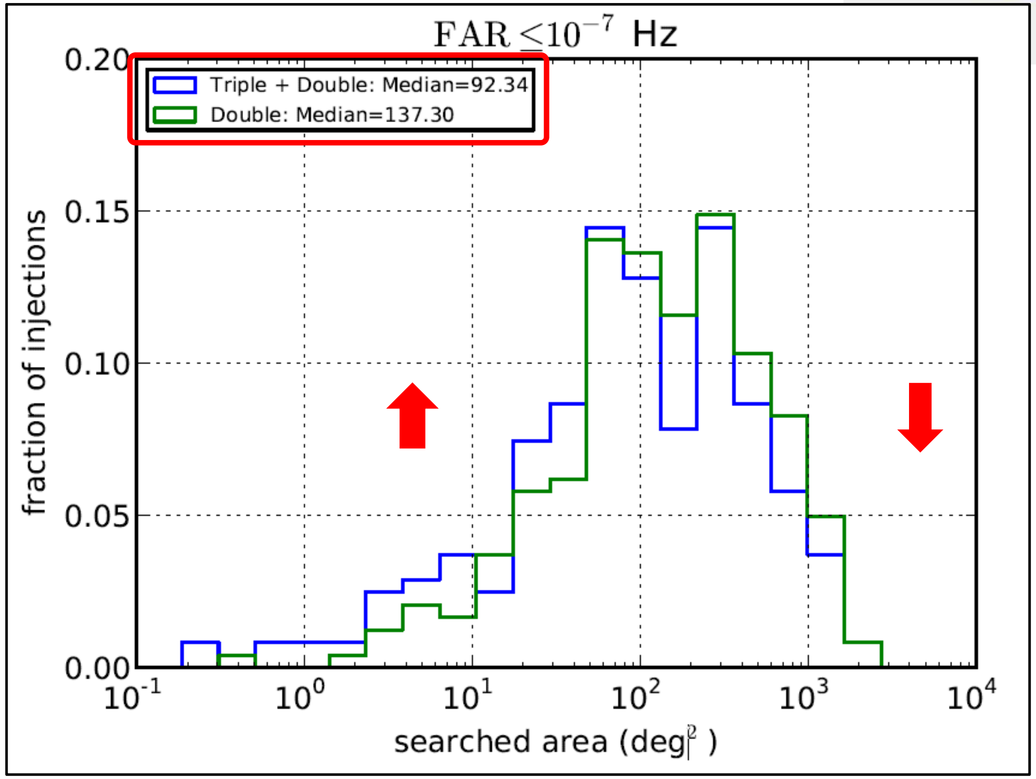
Typical result : statistics



For the "case 2" (Best case), V1 threshold SNR is set at 3.0.



Offset angle



Searched area

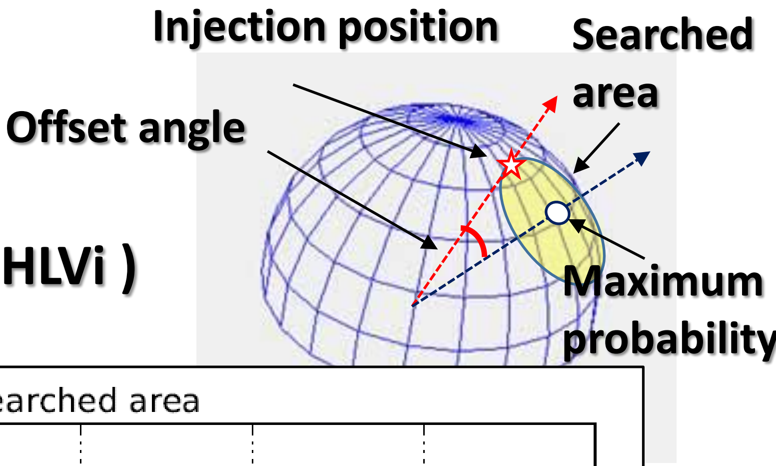
⇨ **Collect the median values, with changing V1 threshold SNR**

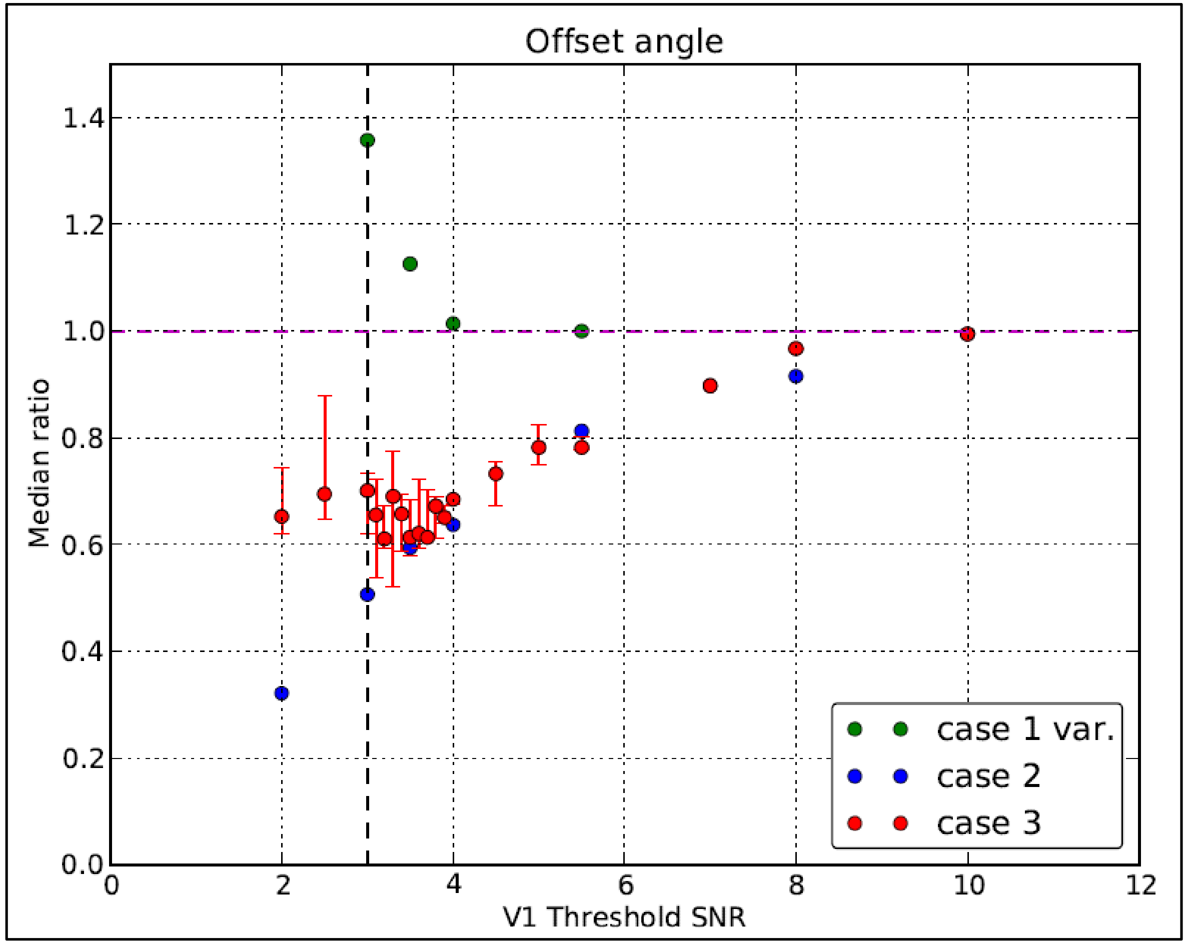
Optimization of Virgo threshold :

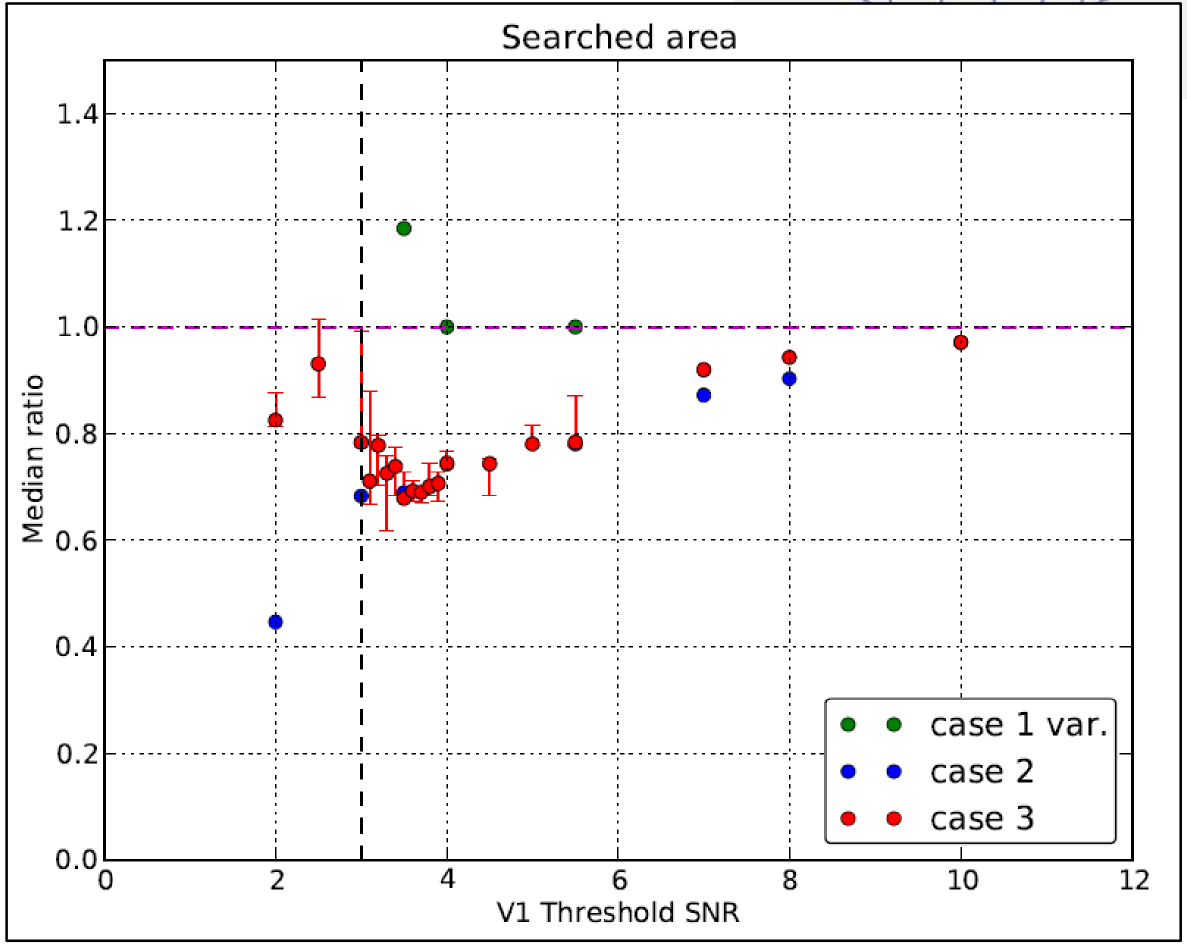
⇨ **Collect the median values, with changing V1 threshold SNR.**

(**Case 1 var.** : HL or HLVr / **Case 2** : HL or HLVi / **Case3** : HL or HLVr or HLVi)

Worst **Best** **More realistic**





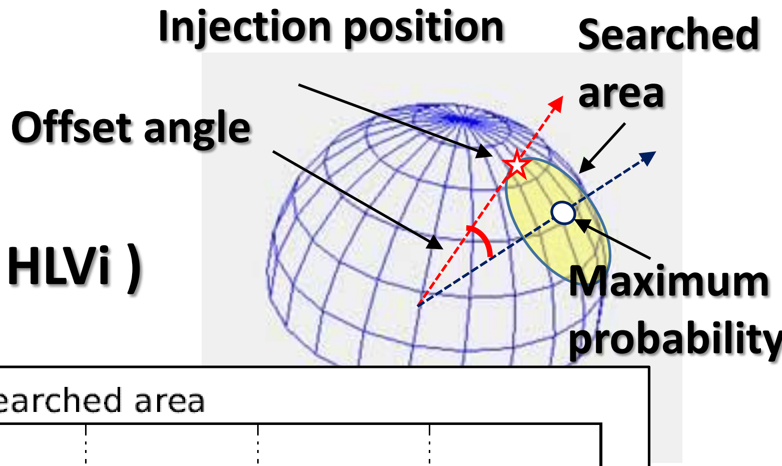


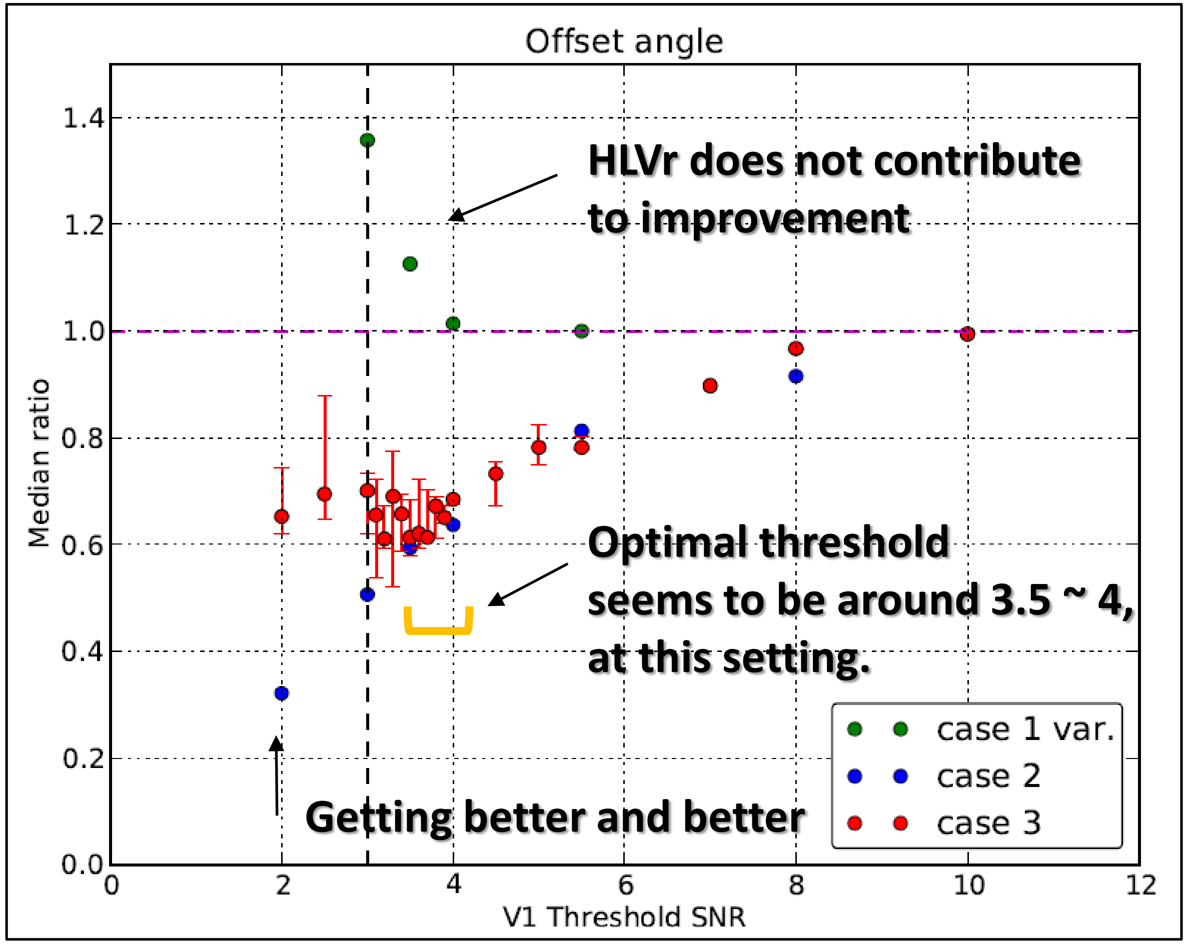
Optimization of Virgo threshold :

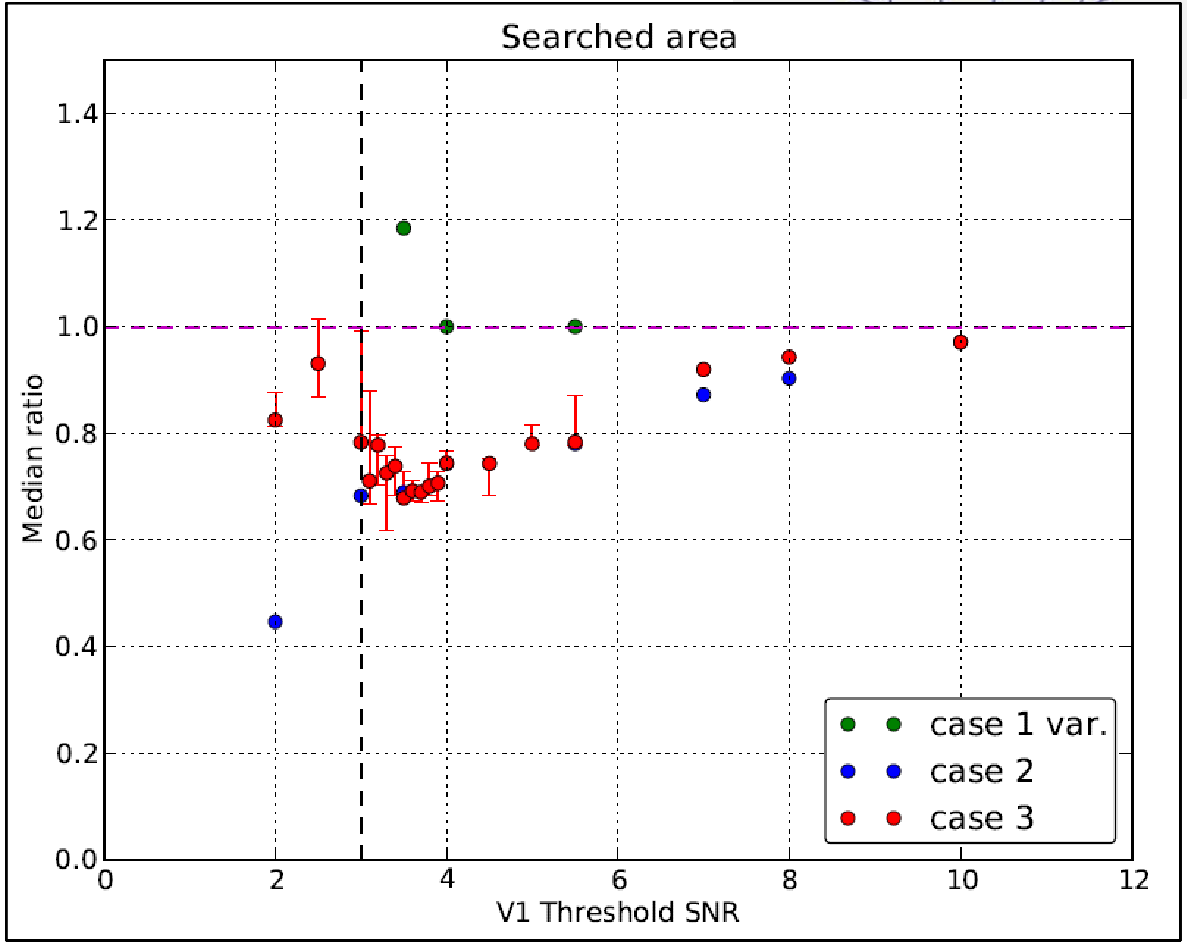
Collect the median values, with changing V1 threshold SNR.

(Case 1 var. : HL or HLVr / Case 2 : HL or HLVi / Case3 : HL or HLVr or HLVi)

Worst — Best — More realistic





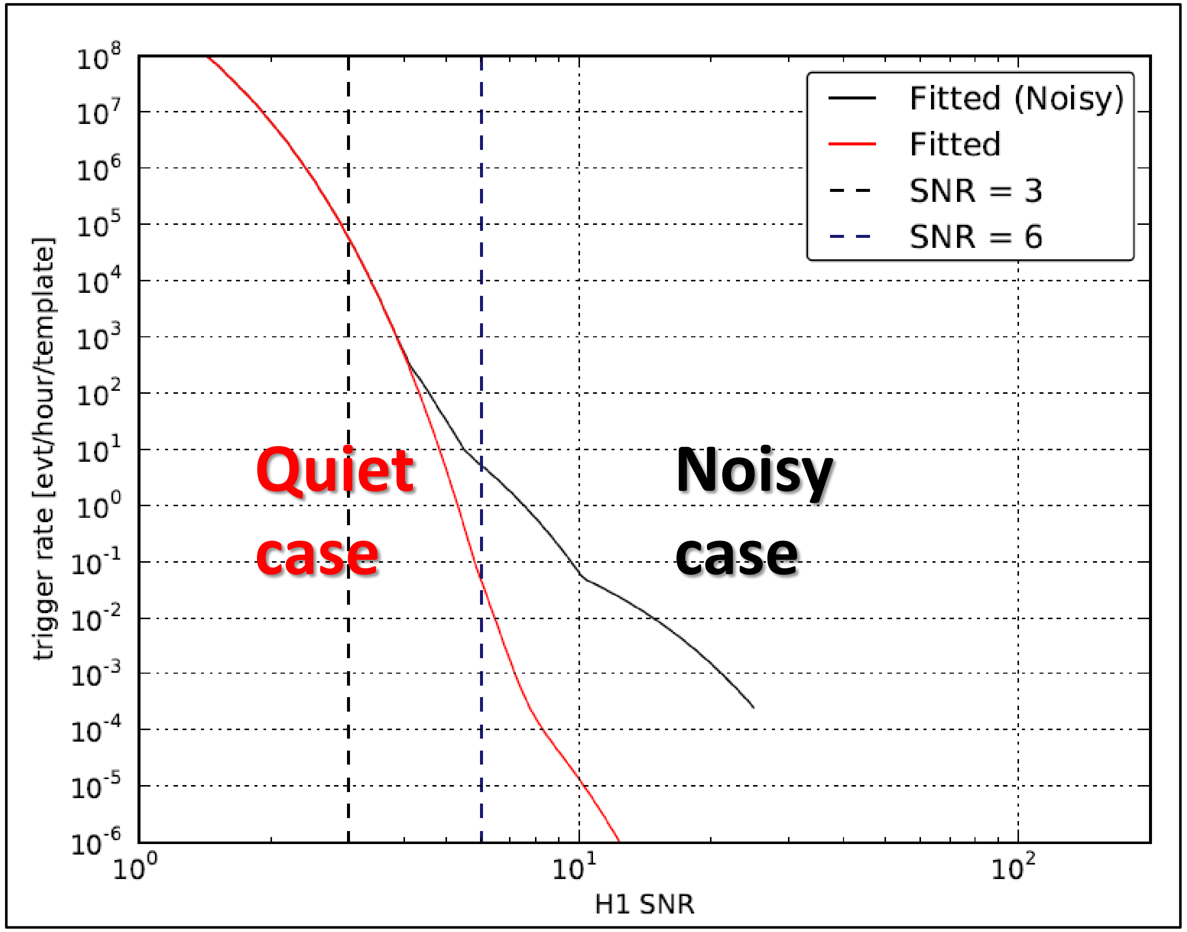


The optimal threshold SNR for V1 is at around 3.5 ~ 4.0. (Threshold for H1, L1 = 5.0)

Optimization of Virgo threshold :

Is the optimal threshold still valid for the noisy case?

SNR distribution



What is happen if noisier SNR distribution, FAP are used?

Calculation setup : How to transform the triggers, HL → HLV

Changed points :

Case 1 var. : HL → HL or HL + random V (Worst case)

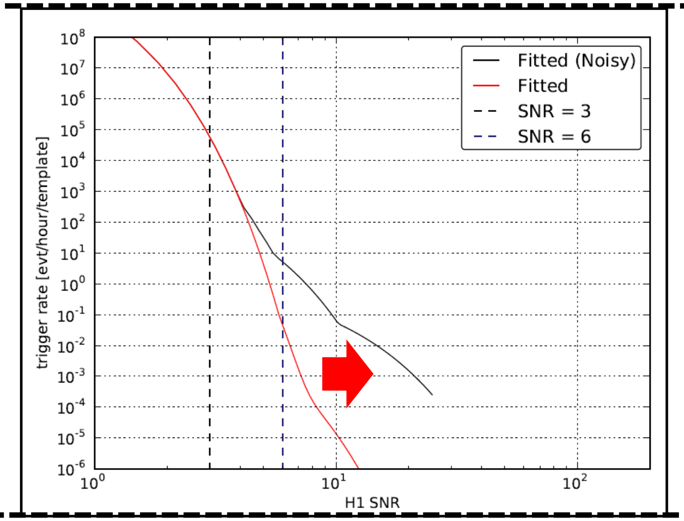
If p > $\boxed{\text{FAP}}$, otherwise

Case 3 : HL → HL, or HL + random V, or HL + V based on injection (More realistic case)

If p < $\boxed{\text{FAP}}$, If p > $\boxed{\text{FAP}}$ and V1 SNR < threshold , If p > $\boxed{\text{FAP}}$ and V1 SNR > threshold

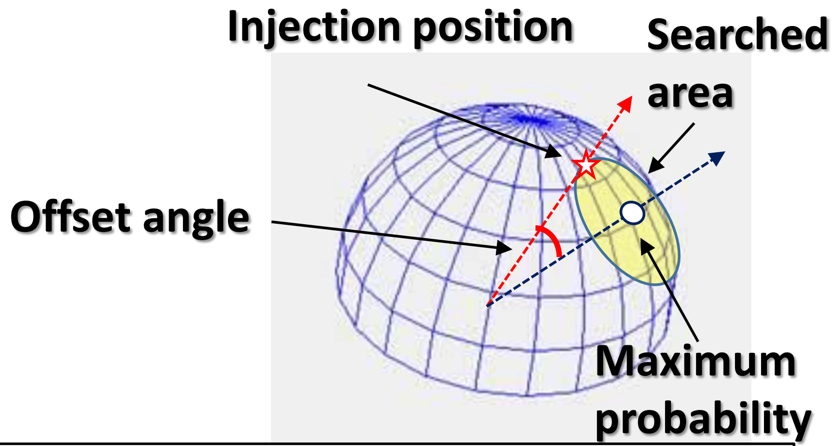
1. "random V trigger : Vr "

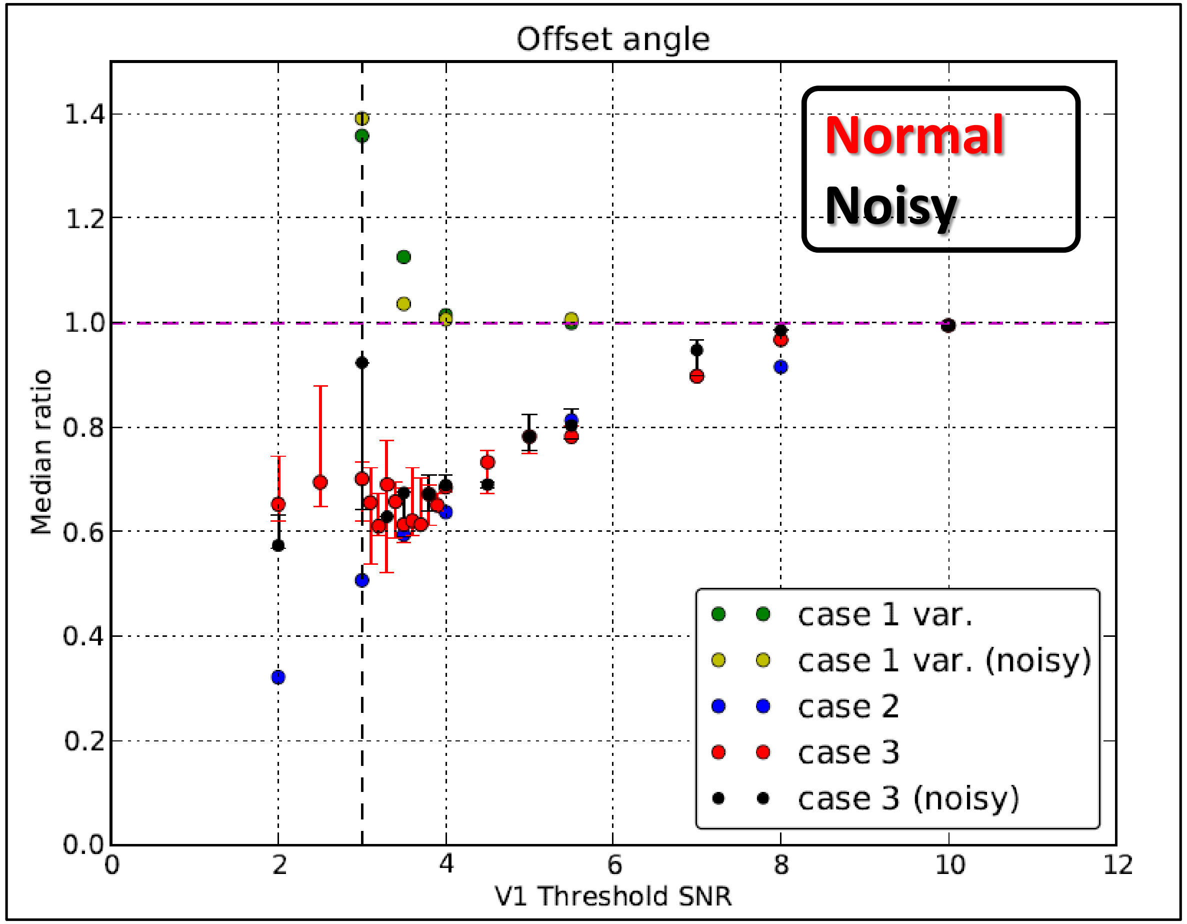
1. SNR = Random above a threshold SNR, following measured O1 SNR distribution

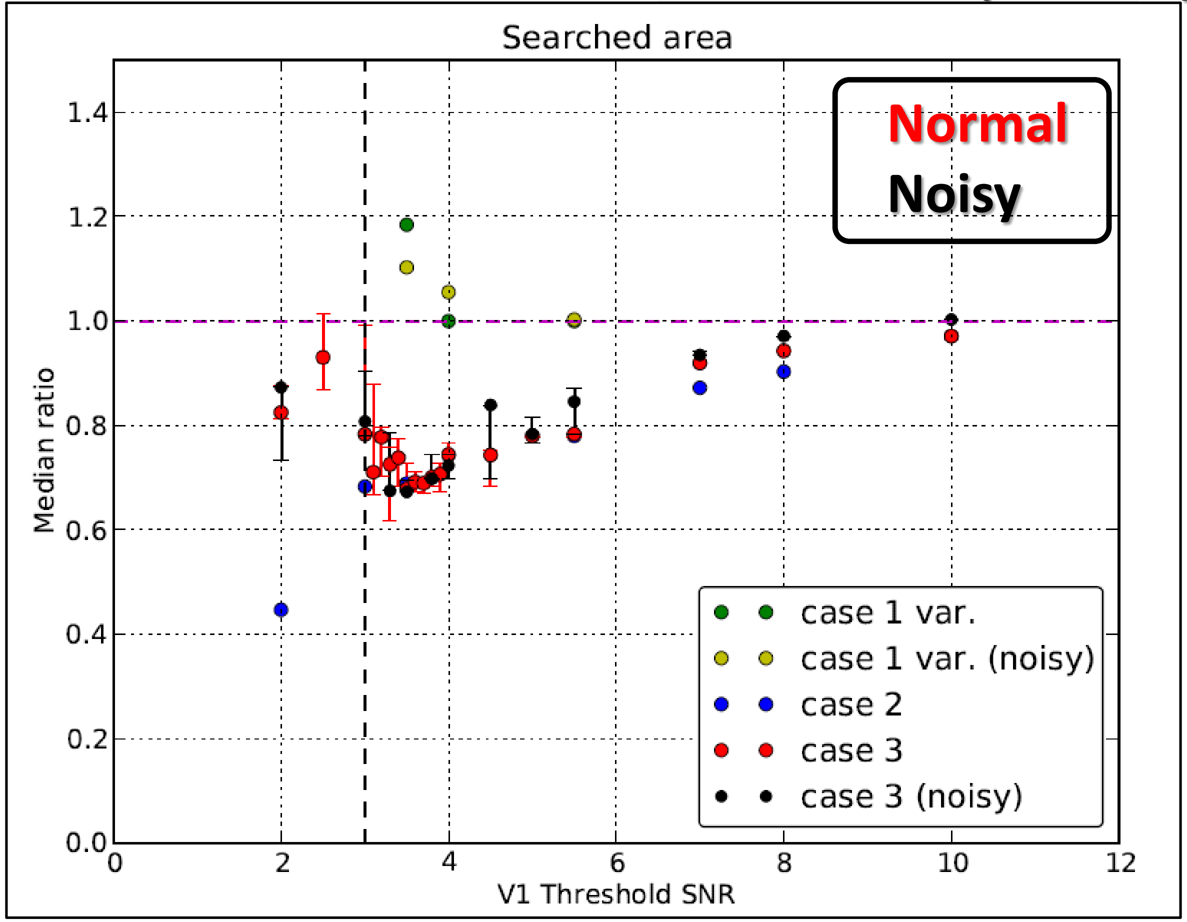


Optimization of Virgo threshold :

⇨ If the background triggers are noisy, the localization can be worse. However, the optimal threshold for V1 still works.







Contents

Summary

Investigated sky localization performance in "hierarchical search" with 3 detectors HLV, and look for the optimal threshold for V1

1. What is the optimal threshold for the V1?

→ **Optimal threshold for V1 is around 3.5 ~ 4.**

2. How the localization gets improved at the threshold?

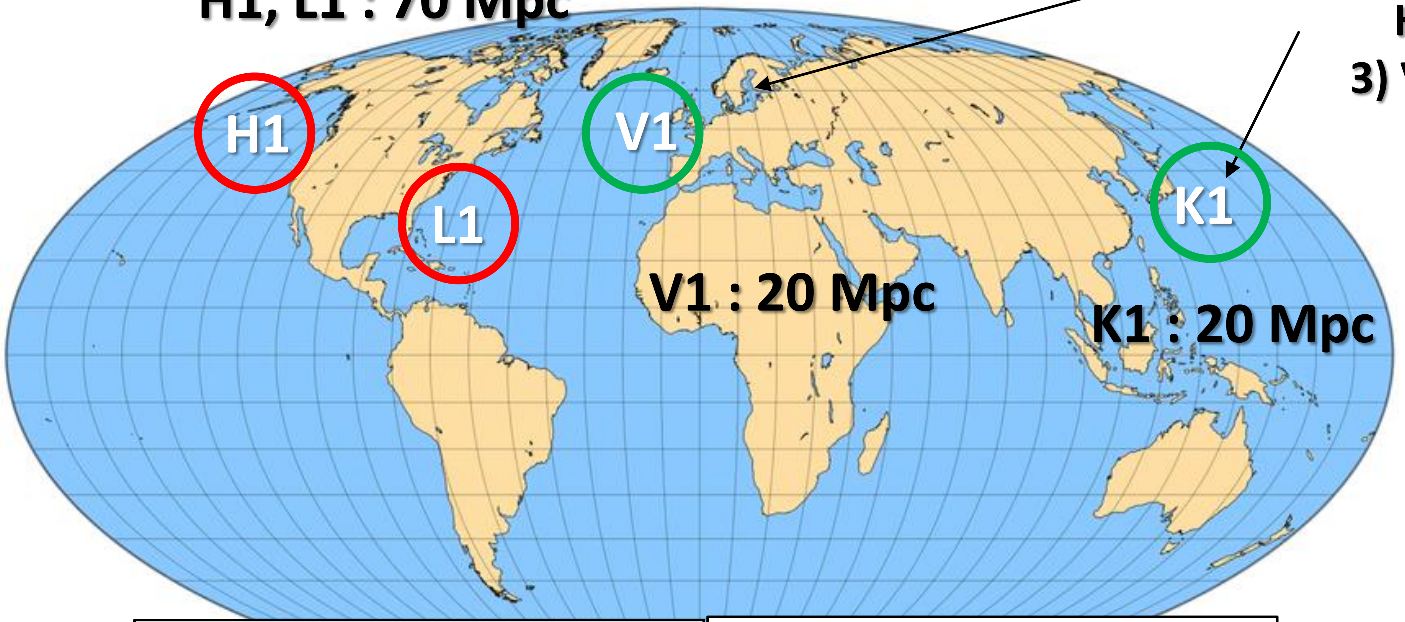
→ **Offset angle, searched area are reduced to ~70 % at the threshold, according to the setup.**

→ **even if the V1 is less sensitive than H1, L1, in the hierarchical search, V1 improves the sky localize performance, comparing to the double detector search.**

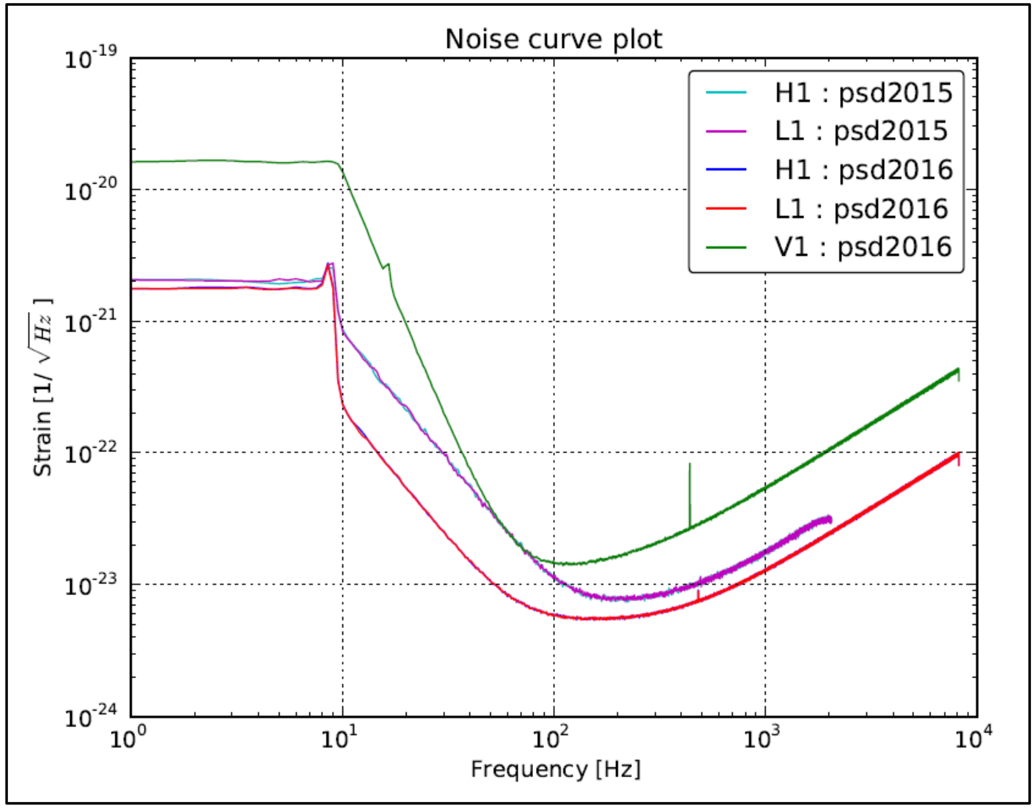
→ **The hierarchical search is useful to enter newly constructed detectors into the network.**

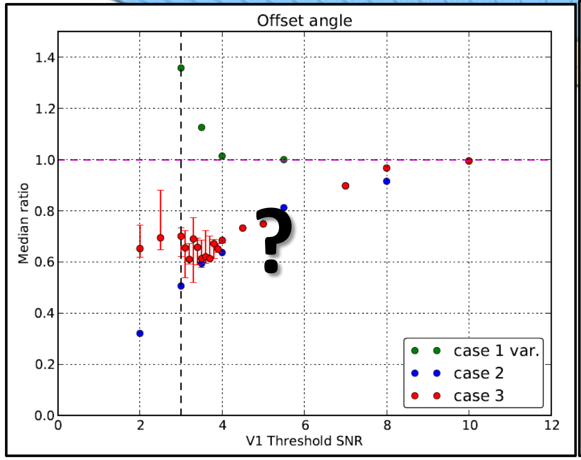
… How about the "Hierarchical search" with 4 detectors HLVK ?

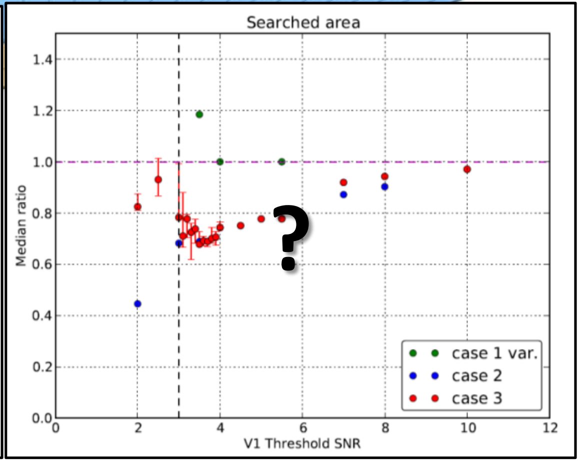
KAGRA related topic (Just for introducing)



1) K1 Noise curve
2) K1 Horizon distance are same as V1:
H1, L1 = 70 Mpc, V1, K1 = 20 Mpc.
3) V1, K1 thresholds are set as same.





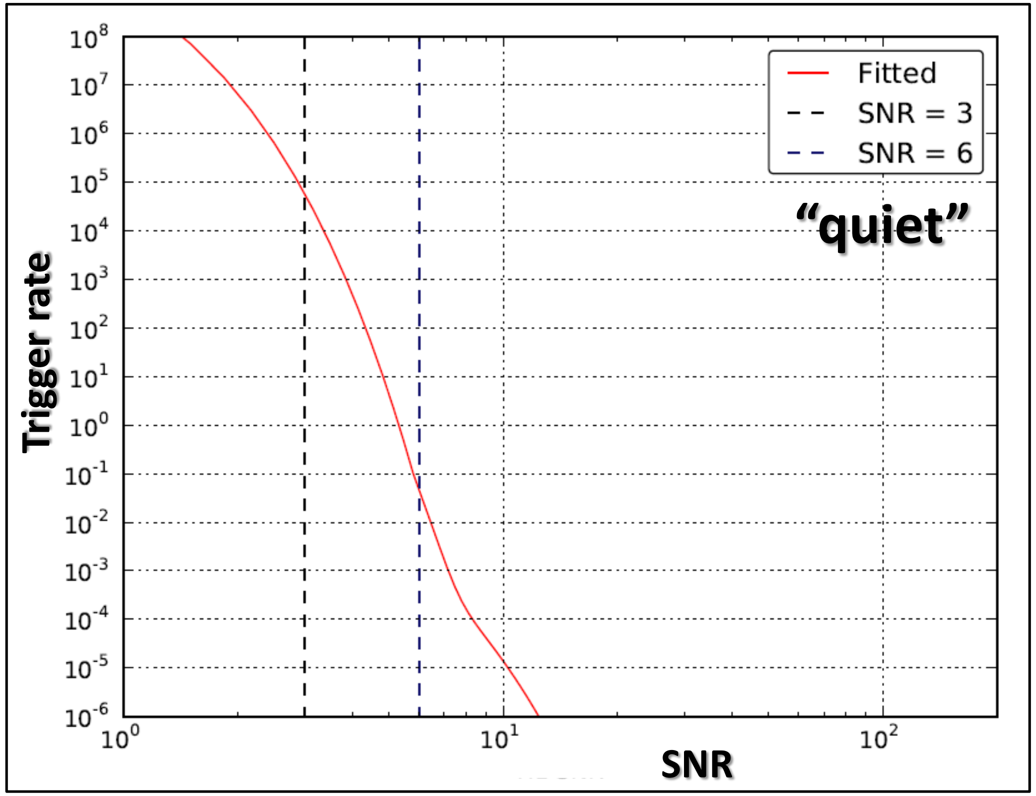


Look for the optimal threshold SNR for V1, K1, in this search.

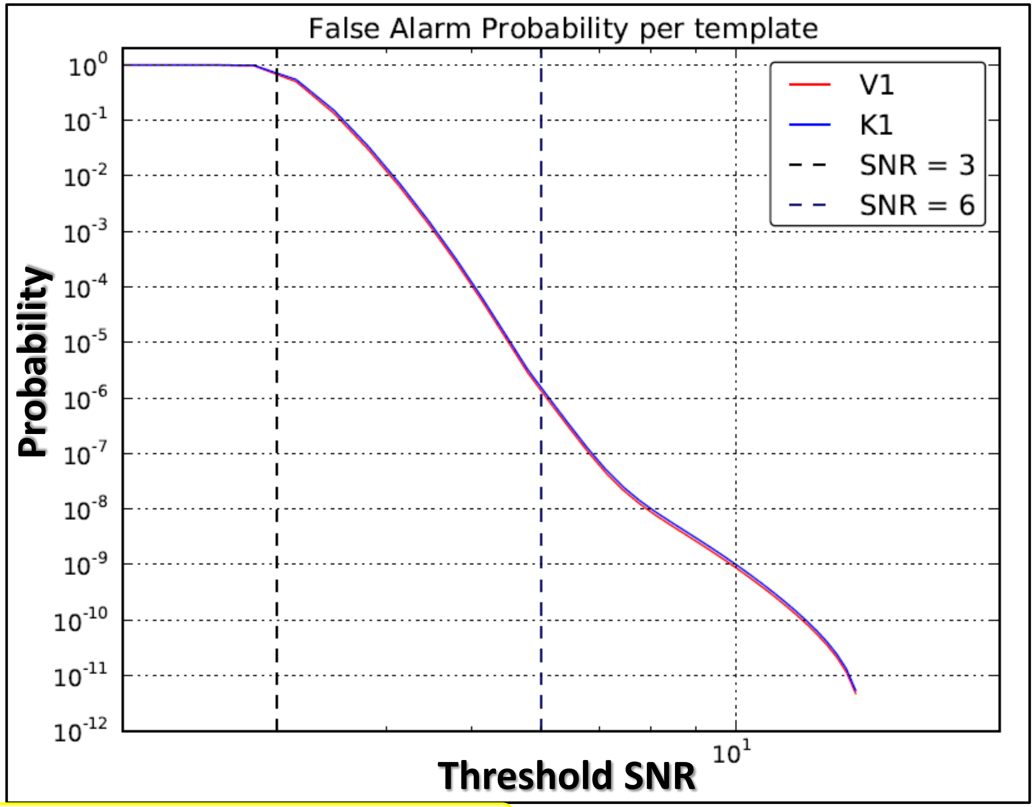
KAGRA related topic : Setup

(Parameters for V1, K1 are mostly same in each other.)

SNR distribution (per template)



False Alarm Probability (per template)



FAP = $1 - \exp(-R \times T)$

R **= cumulative rate of background triggers per template, above a given threshold, per template,**

T = analyzing time for the V1 (less sensitive detector)

70 ms for V1
80 ms for K1

Calculation setup : How to transform the triggers, HL → HLV or HLK or HLVK

➡ **Transforming concept is same as the 3-detector search**

2 Procedure

p_{V1}, p_{K1} = random uniform number from 0 to 1.

Case 1 : V1, K1 triggers are random

$$\mathrm{HL} + \mathrm{V_{random}} + \mathrm{K_{random}}$$

Case 1 var : V1, K1 triggers are random ⇨ **Worst case**

$$\begin{array}{lll}
p_{V1} < FAP_{V1} \text{ and } p_{K1} < FAP_{K1} & \Rightarrow & \mathrm{HL} + \mathrm{V_{random}} + \mathrm{K_{random}} \\
p_{V1} > FAP_{V1} \text{ and } p_{K1} < FAP_{K1} & \Rightarrow & \mathrm{HL} + \quad + \mathrm{K_{random}} \\
p_{V1} < FAP_{V1} \text{ and } p_{K1} > FAP_{K1} & \Rightarrow & \mathrm{HL} + \mathrm{V_{random}} + \\
p_{V1} > FAP_{V1} \text{ and } p_{K1} > FAP_{K1} & \Rightarrow & \mathrm{HL} + \quad +
\end{array}$$

Case 2 : V1, K1 triggers are based on injection parameters ⇨ **Best case**

$$\begin{array}{lll}
\mathrm{SNR}_{V1} > \mathrm{Threshold}_{V1} \text{ and } \mathrm{SNR}_{K1} > \mathrm{Threshold}_{K1} & \Rightarrow & \mathrm{HL} + \mathrm{V_{inj}} + \mathrm{K_{inj}} \\
\mathrm{SNR}_{V1} < \mathrm{Threshold}_{V1} \text{ and } \mathrm{SNR}_{K1} > \mathrm{Threshold}_{K1} & \Rightarrow & \mathrm{HL} + \quad + \mathrm{K_{inj}} \\
\mathrm{SNR}_{V1} > \mathrm{Threshold}_{V1} \text{ and } \mathrm{SNR}_{K1} < \mathrm{Threshold}_{K1} & \Rightarrow & \mathrm{HL} + \mathrm{V_{inj}} + \\
\mathrm{SNR}_{V1} < \mathrm{Threshold}_{V1} \text{ and } \mathrm{SNR}_{K1} < \mathrm{Threshold}_{K1} & \Rightarrow & \mathrm{HL} + \quad +
\end{array}$$

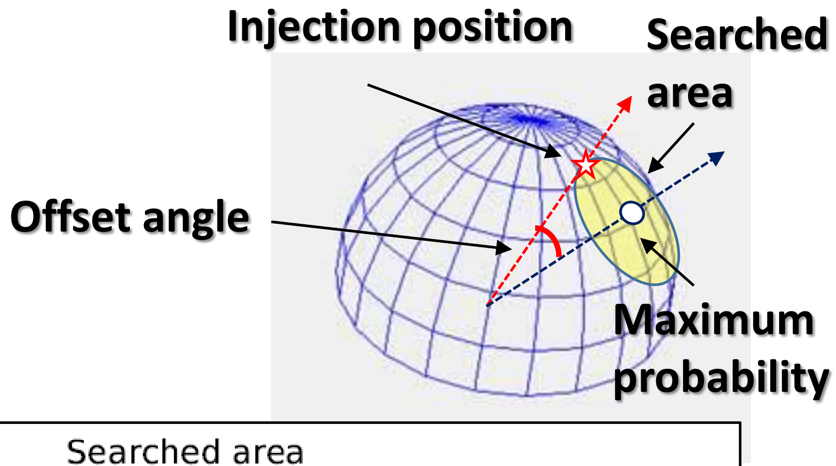
Case 3 : V1, K1 triggers are either random or based on injection parameters

$FAP = FAP(\mathrm{SNR})$ if SNR > Threshold, otherwise $FAP = FAP(\mathrm{Threshold})$

- $p_{V1} < FAP_{V1}$ and $p_{K1} < FAP_{K1}$ $\Rightarrow$ $\mathrm{HL} + \mathrm{V_{random}} + \mathrm{K_{random}}$
- $p_{V1} < FAP_{V1}$ and, $p_{K1} > FAP_{K1}$ and $\mathrm{SNR}_{K1} > \mathrm{Threshold}_{K1}$ $\Rightarrow$ $\mathrm{HL} + \mathrm{V_{random}} + \mathrm{K_{inj}}$
- $p_{V1} > FAP_{V1}$ and $\mathrm{SNR}_{V1} > \mathrm{Threshold}_{V1}$ and $p_{K1} < FAP_{K1}$ $\Rightarrow$ $\mathrm{HL} + \mathrm{V_{inj}} + \mathrm{K_{random}}$
- $p_{V1} > FAP_{V1}$ and $\mathrm{SNR}_{V1} > \mathrm{Threshold}_{V1}$ and $p_{K1} > FAP_{K1}$ and $\mathrm{SNR}_{K1} > \mathrm{Threshold}_{K1}$ $\Rightarrow$ $\mathrm{HL} + \mathrm{V_{inj}} + \mathrm{K_{inj}}$
- $p_{V1} < FAP_{V1}$ and $p_{K1} > FAP_{K1}$ and $\mathrm{SNR}_{K1} < \mathrm{Threshold}_{K1}$ $\Rightarrow$ $\mathrm{HL} + \mathrm{V_{random}} +$
- $p_{V1} > FAP_{V1}$ and $\mathrm{SNR}_{V1} < \mathrm{Threshold}_{V1}$ and $p_{K1} < FAP_{K1}$ $\Rightarrow$ $\mathrm{HL} + \quad + \mathrm{K_{random}}$
- $p_{V1} > FAP_{V1}$ and $\mathrm{SNR}_{V1} > \mathrm{Threshold}_{V1}$ and $p_{K1} > FAP_{K1}$ and $\mathrm{SNR}_{K1} < \mathrm{Threshold}_{K1}$ $\Rightarrow$ $\mathrm{HL} + \mathrm{V_{inj}} +$
- $p_{V1} > FAP_{V1}$ and $\mathrm{SNR}_{V1} < \mathrm{Threshold}_{V1}$ and $p_{K1} > FAP_{K1}$ and $\mathrm{SNR}_{K1} > \mathrm{Threshold}_{K1}$ $\Rightarrow$ $\mathrm{HL} + \quad + \mathrm{K_{inj}}$
- $p_{V1} > FAP_{V1}$ and $\mathrm{SNR}_{V1} < \mathrm{Threshold}_{V1}$ and $p_{K1} > FAP_{K1}$ and $\mathrm{SNR}_{K1} < \mathrm{Threshold}_{K1}$ $\Rightarrow$ $\mathrm{HL} + \quad +$

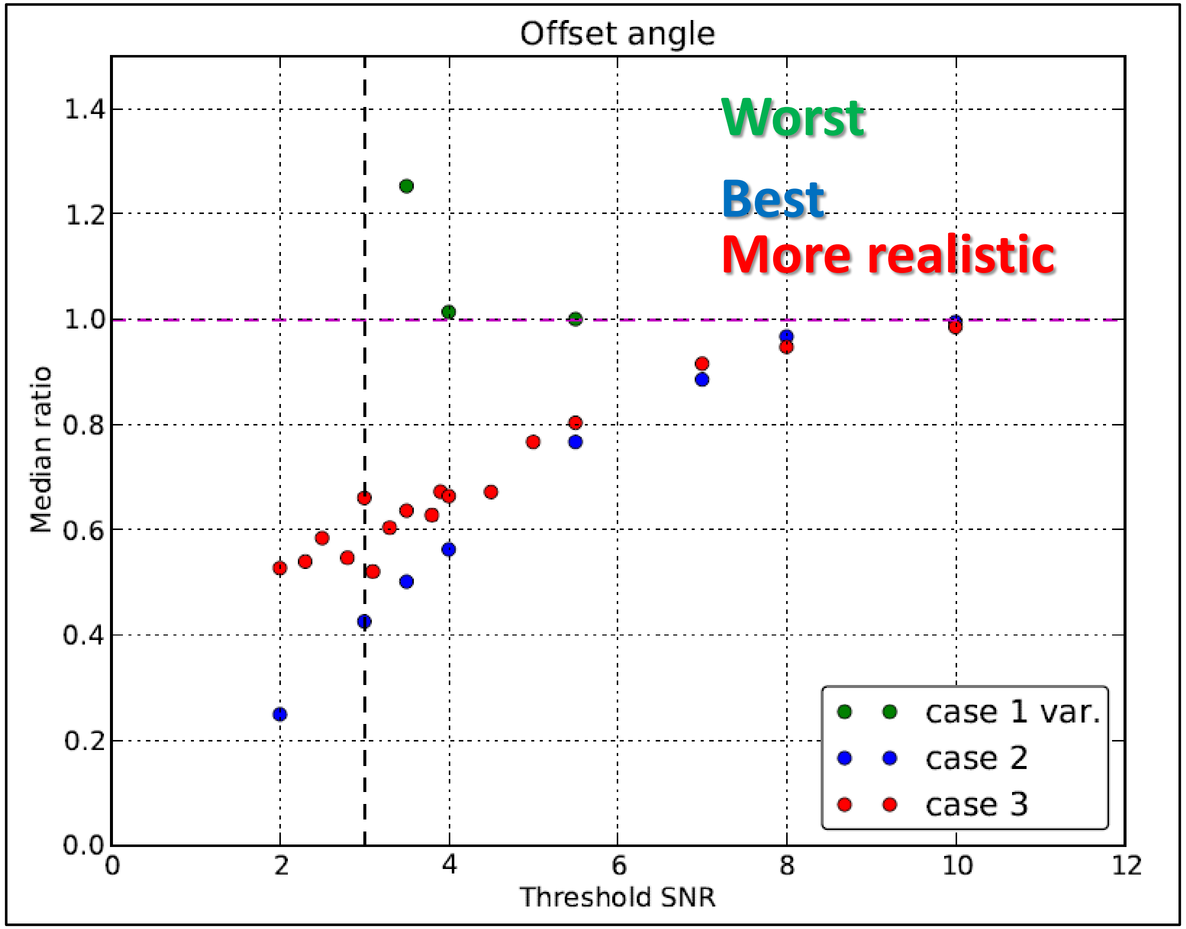
⇨ **More realistic case**

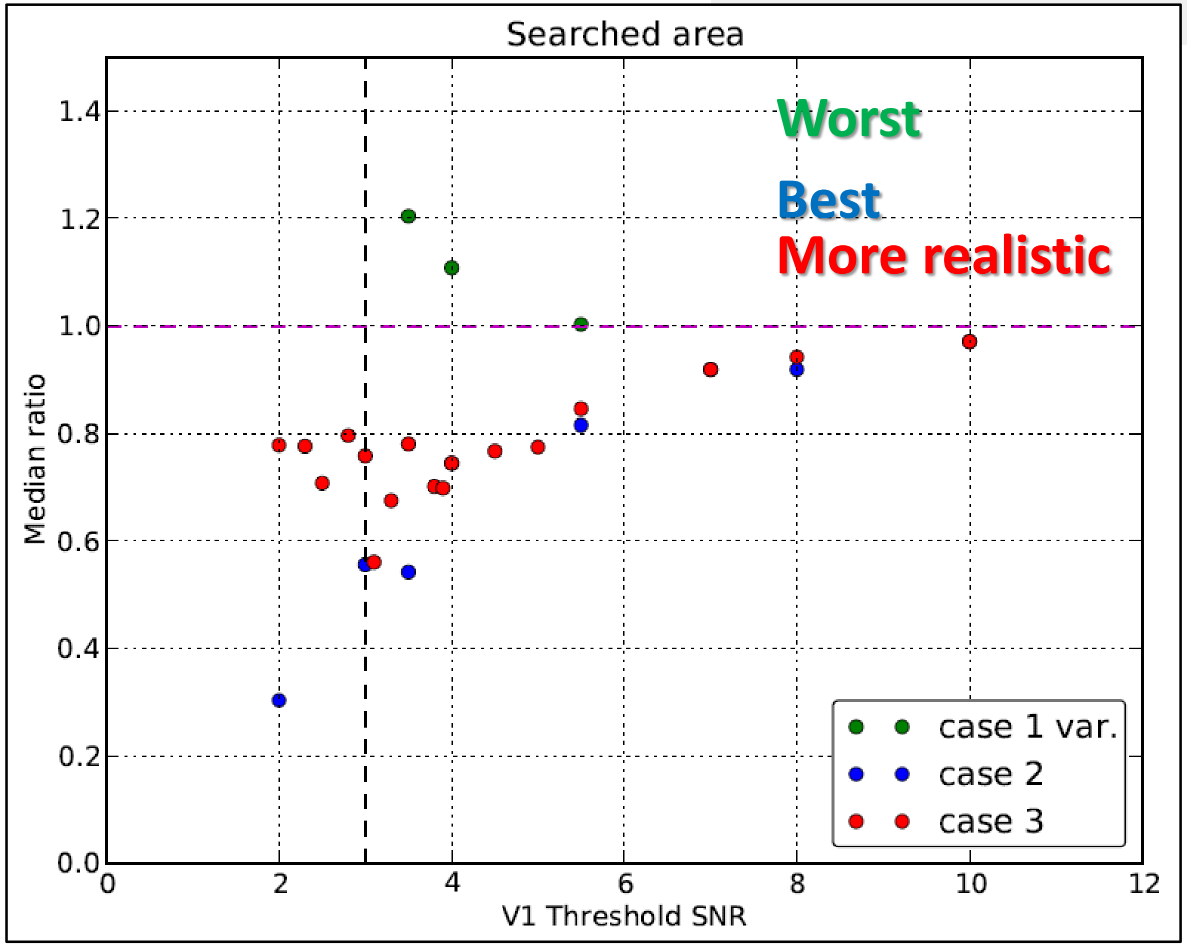
Optimization of V1, K1 threshold :



Now calculation of HLVK is ongoing..

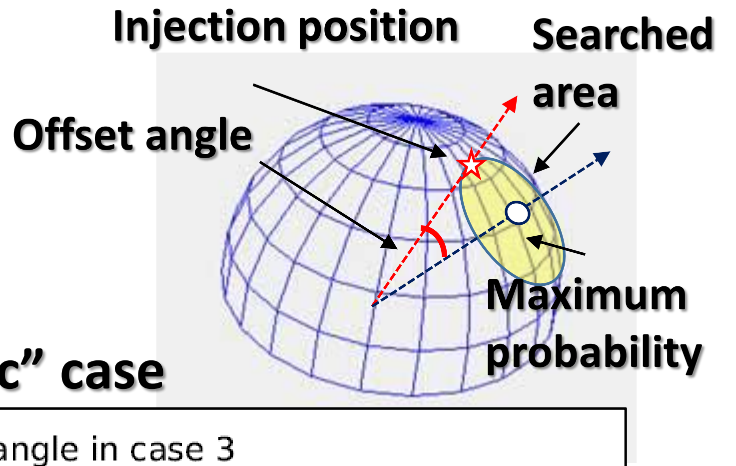
(Uncertainties of the red points are to be investigated.)



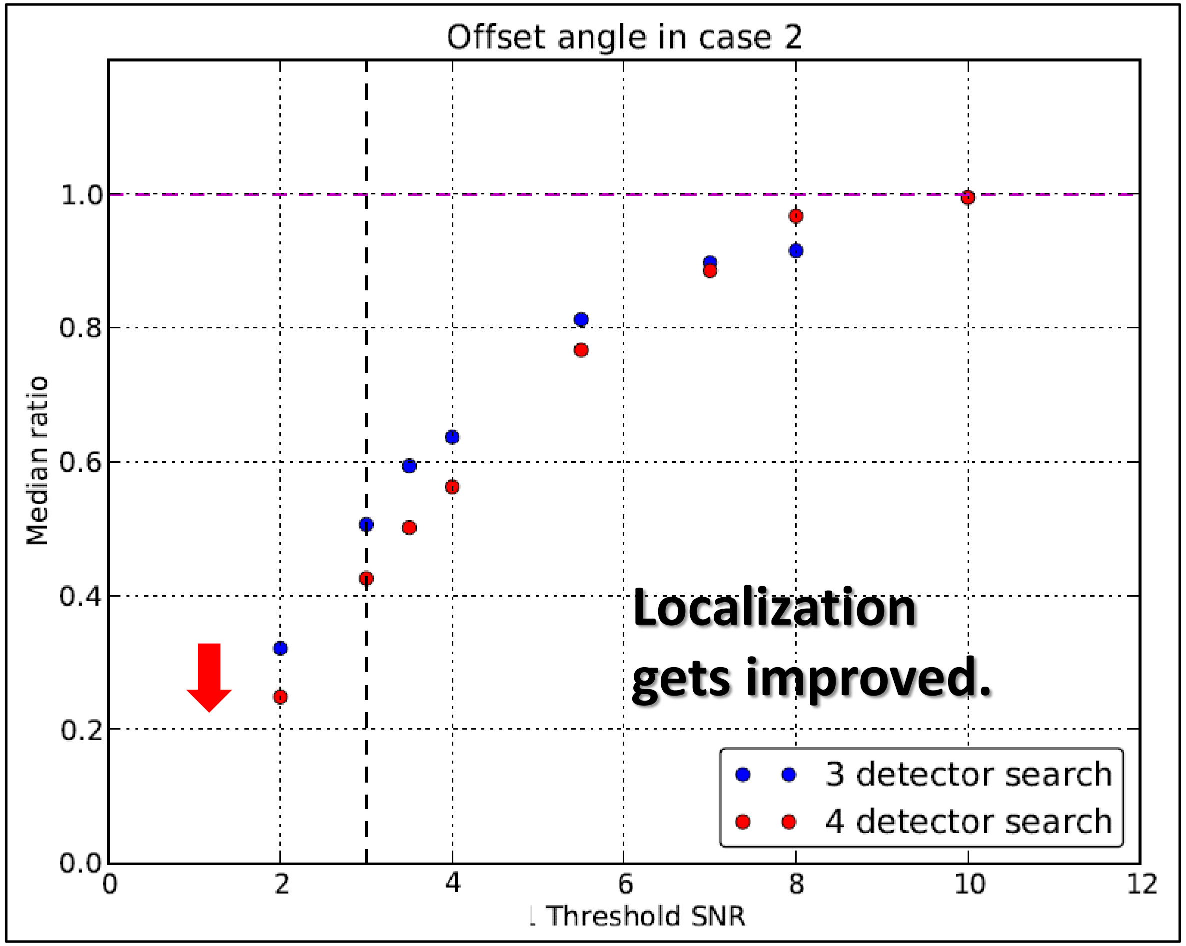


Optimization of V1, K1 threshold : Offset angle

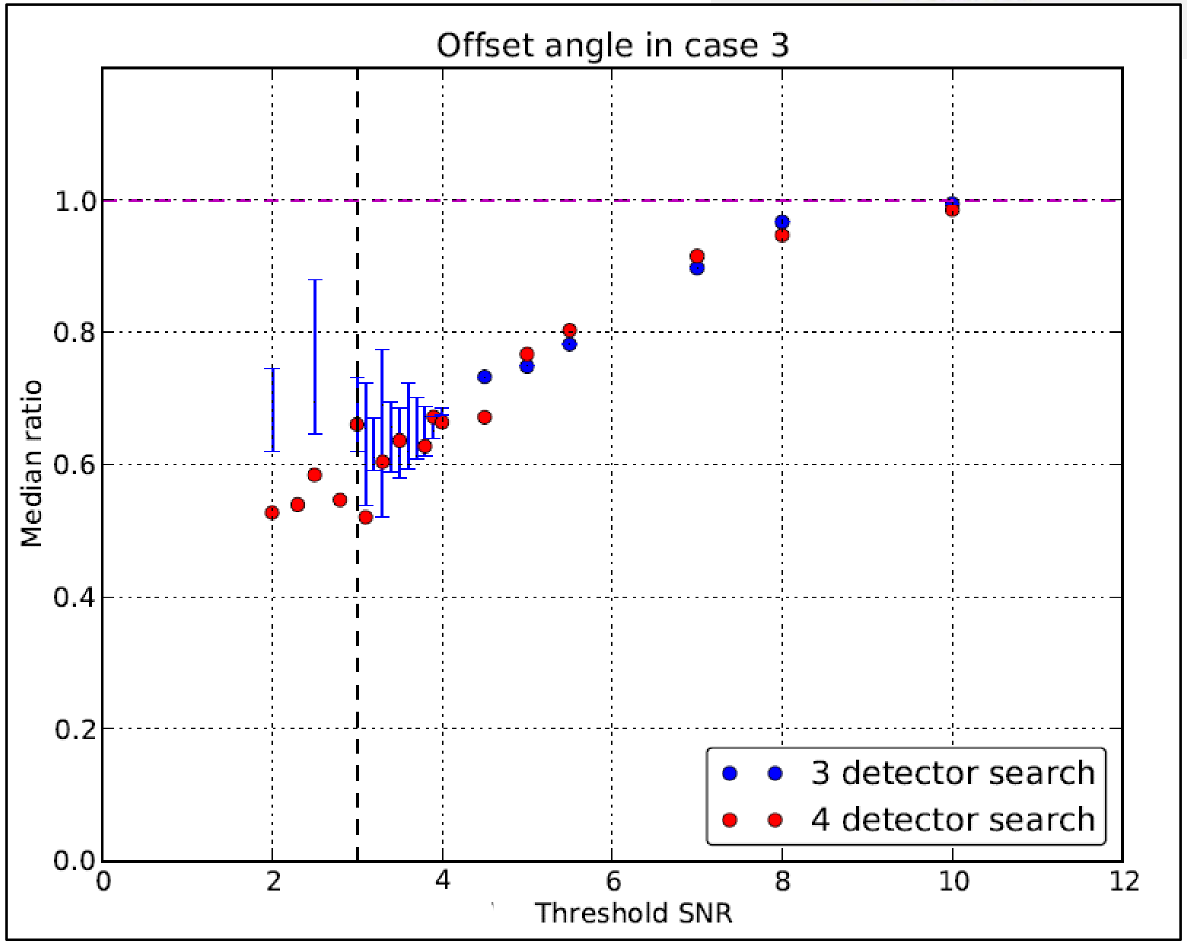
Now calculation of HLVK is ongoing..



Case2 = “Best” case

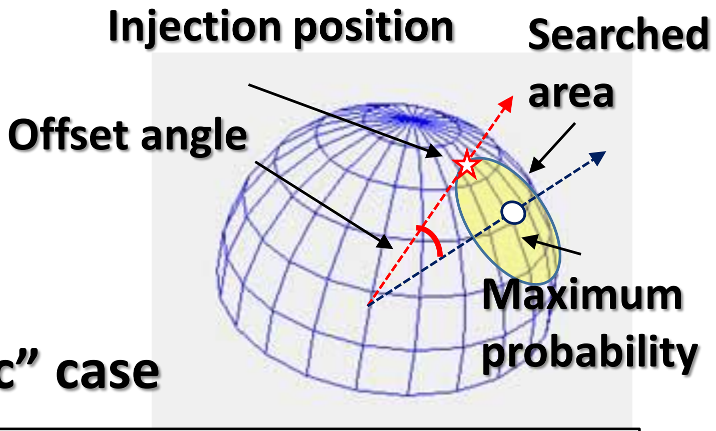


Case3 = “More realistic” case

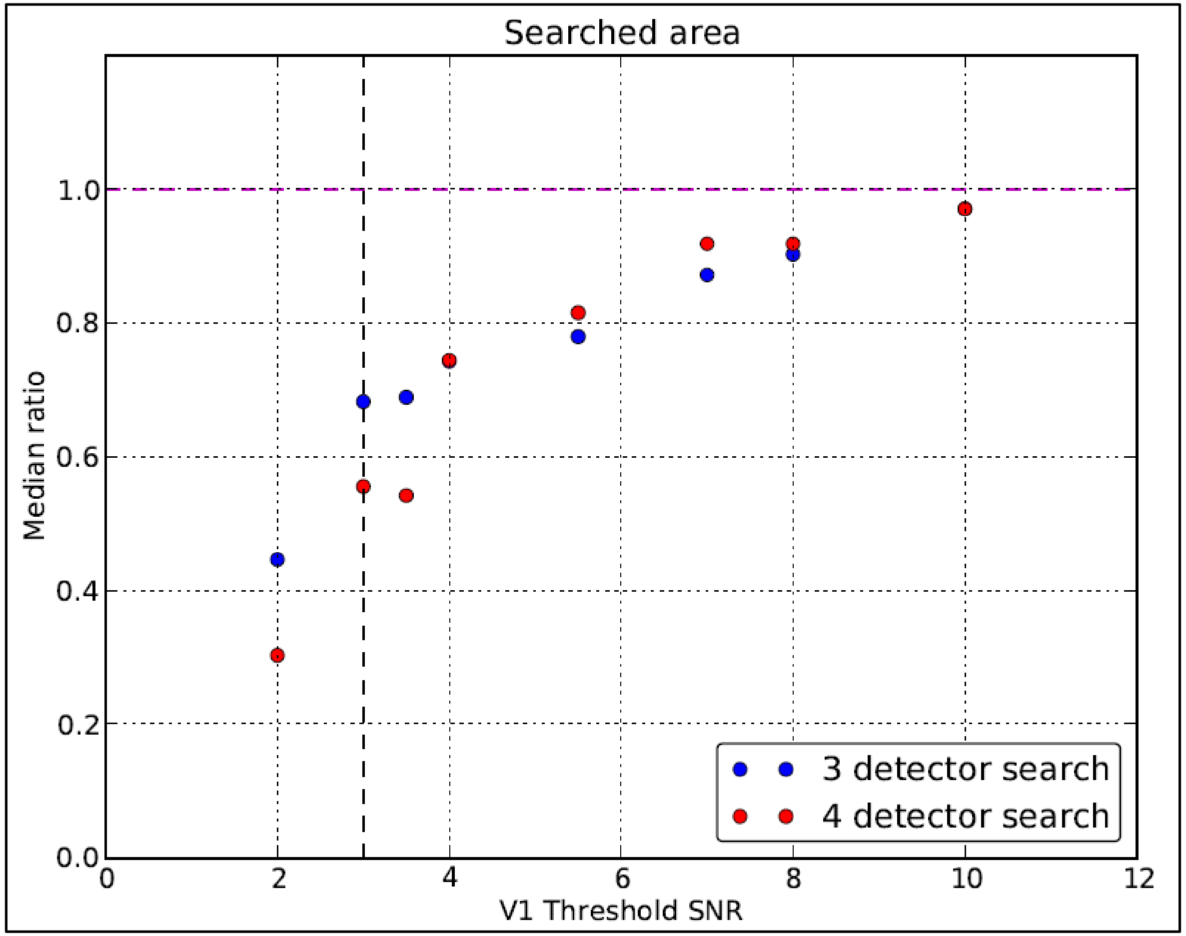


Optimization of V1, K1 threshold : Searched area

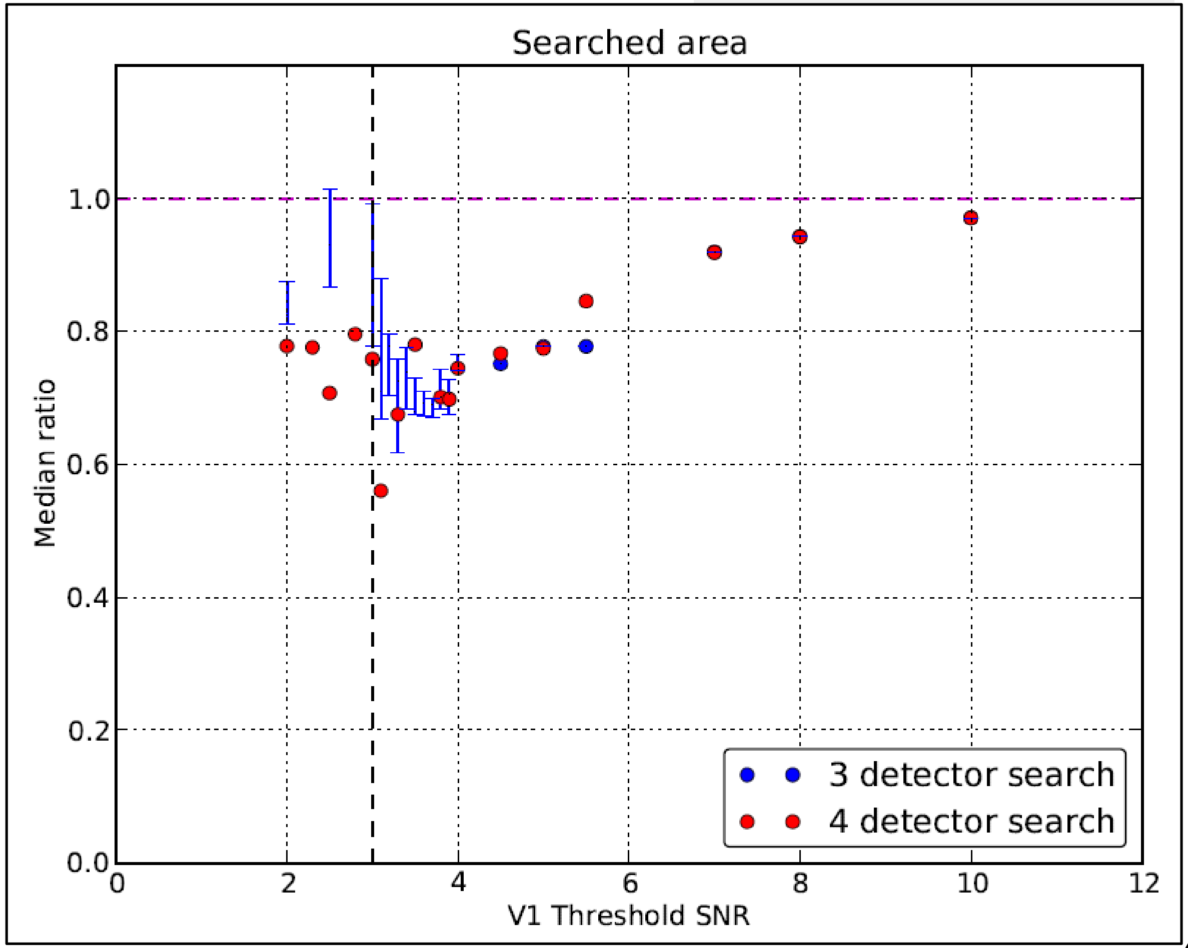
Now calculation of HLVK is ongoing..



Case2 = “Best” case



Case3 = “More realistic” case



Next step (ongoing)

1. To get more realistic results,
   simulate the localization performances with changing the added timing uncertainties.

2. Continue the calculation about the hierarchical search with 4 detectors HLVK

…


Hotel du Palais de l'Isle


NISSIN
Nouilles Japonaises
Saveur Poulet 'Teriyaki'




End

Tools which I learned : Vega

vega : plotting tool based on ROOT.

Mainly (in my case)

*** Plot histograms, such as SNR distribution.**

*** Fit**

*** Edit MBTA output files (.gwf), or Bayestar input files (.xml)**

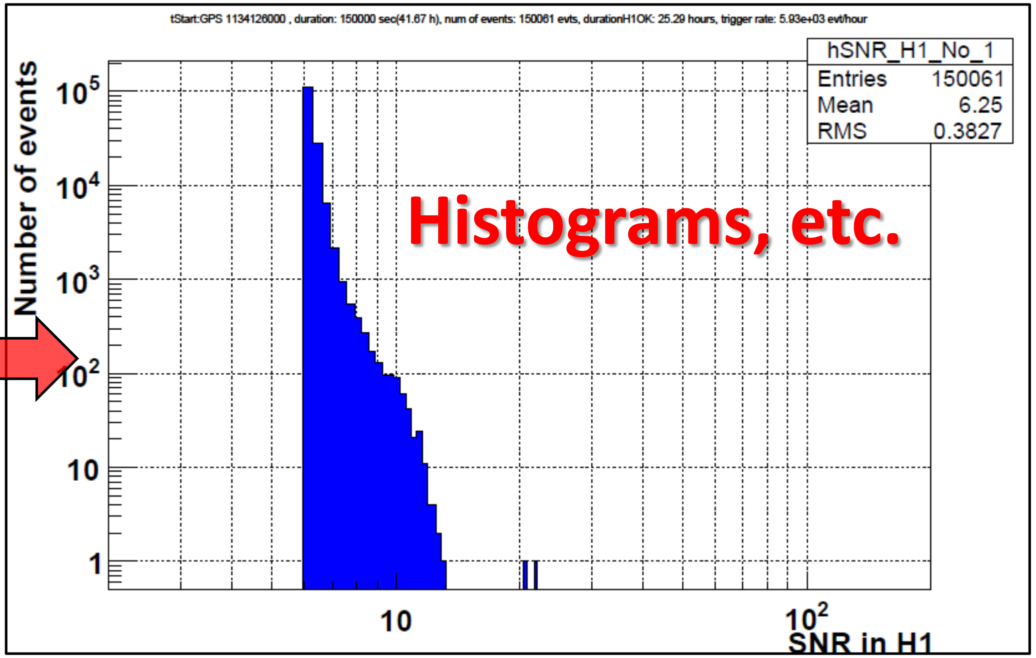
errormessage.txt × SNR_InjectionMDC160714v1.C × MbtaHLV-Chi2MCutOK-clustered-969622856-997_th35.xml × MbtaHLV-Chi2M

```c
  char *eventName = "SpinTaylorT4threePointFivePN";
  printf("-->Event name:%s tStart=%.0f duration=%.0f\n",eventName, tStart, duration);
  FrSimEvent *eventsDet  = FrSimEventReadT(mbtaFile,eventName, tStart, duration, 0., 1.e10);
  FrSimEventDump(eventsDet, stdout, 2);
  //---------------------------------------------build graphs

  TH1F *hSNR_H1 = new TH1F("SNR_H1","", 600, 1e0, 300);
  TH1F *hSNR_L1 = new TH1F("SNR_L1","", 600, 1e0, 300);
  TH1F *hSNR_V1 = new TH1F("SNR_V1","", 600, 1e0, 300);
  TH1F *hSNR_com = new TH1F("SNR_combined","", 600, 1e0, 300);

  int nEvt = 0;
  for(evt = eventsDet; evt != NULL; evt = evt->next) {
    FrSimEventDump(evt,stdout,3);
    double snrH1 = 8*RH_H1/FrSimEventGetParam(evt, "eff_dist_h");
    double snrL1 = 8*RH_L1/FrSimEventGetParam(evt, "eff_dist_l");
    double snrV1 = 8*RH_V1/FrSimEventGetParam(evt, "eff_dist_v");
    double snrCom = sqrt(snrH1**2 + snrL1**2);
    if(snrCom > 0. && snrH1 > 0 && snrL1 > 0){
    hSNR_H1->Fill(snrH1);
    hSNR_L1->Fill(snrL1);
    hSNR_V1->Fill(snrV1);
    hSNR_com->Fill(snrCom);
    }
  nEvt++;}
```

"C" language



Histograms, etc.

Tools which I learned : Bayestar

Bayestar : mainly

bayestar_localize_coinc :

*** Generate files to plot skymaps (this process needs long time : ~ one night)**

bayestar_aggregate_found_injections :

*** Generate files to plot offsets angles, searched area, 90 % confidence area ,,,**

bayestar_plot_allsky :

*** Generate skymaps**

```
olserver59[~]: bayestar_
bayestar_aggregate_found_injections   bayestar_plot_found_injections
bayestar_bin_samples                  bayestar_plot_pileup
bayestar_lattice_tmpltbank            bayestar_prune_neighborhood_tmpltbank
bayestar_littlehope                   bayestar_realize_coincs
bayestar_localize_coincs              bayestar_sample_model_psd
bayestar_localize_lvalert             bayestar_sim_to_tmpltbank
bayestar_plot_allsky
olserver59[~]: bayestar_
```

Bayestar has more functions.
what I'm using is only these ones.

*** Except for them, I'm using "ligolw", some python codes etc.**

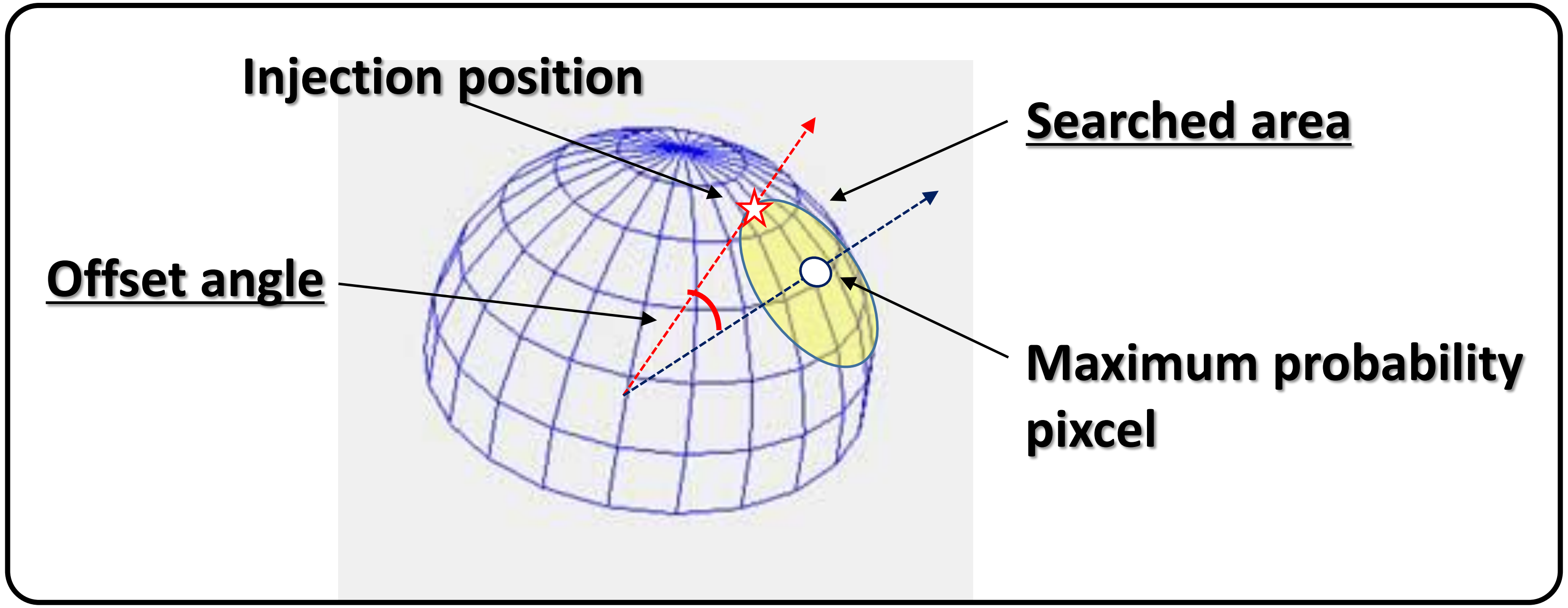
Definitions of the offset angle and the searched area :

1. Offset angle
2. Searched area

⇨ How far the localization is from the true injected position

3. Certain confidence area
   (ex. 90 % confidence area)

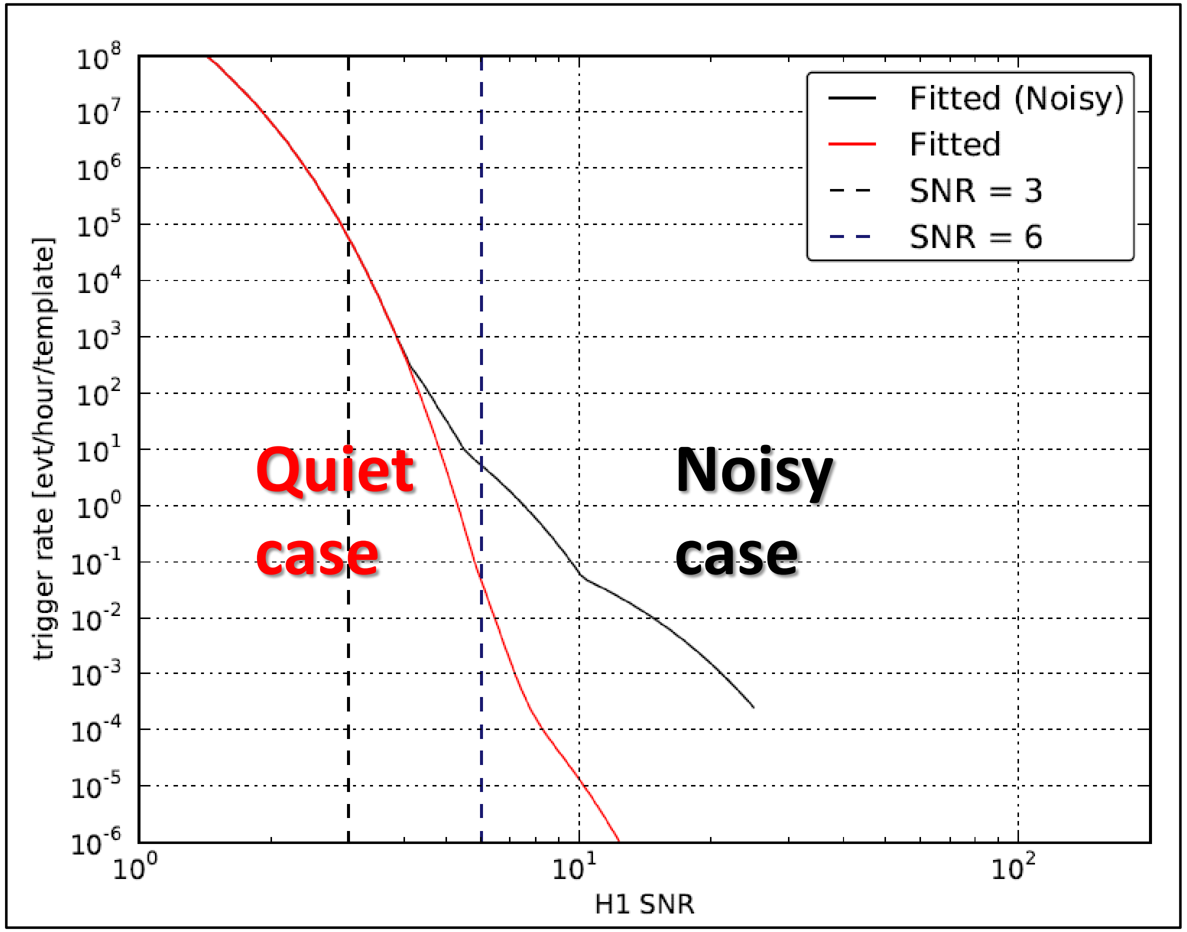
⇨ How spread or concentrated each probability is



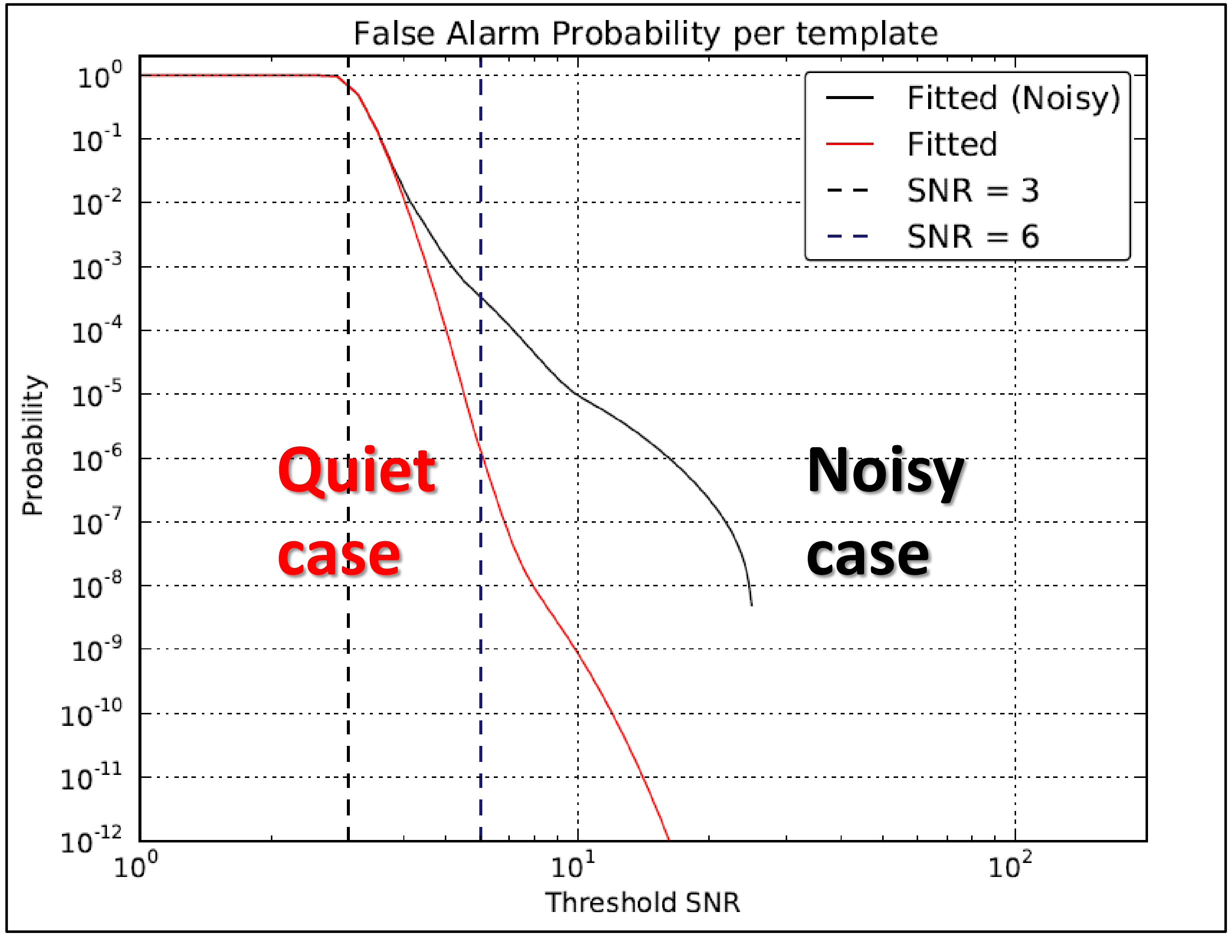
Optimization of Virgo threshold :

Is the optimal threshold still valid for the noisy case?

SNR distribution

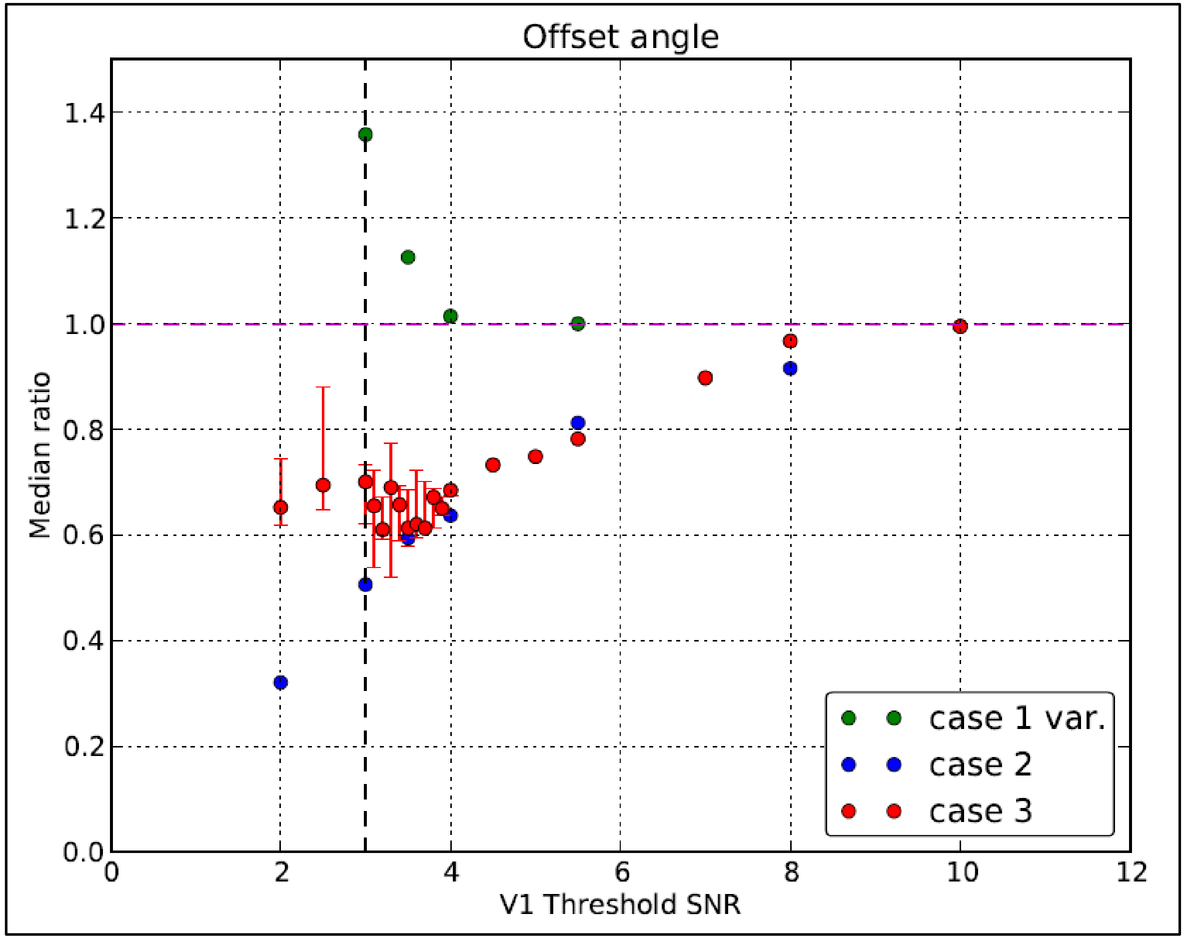


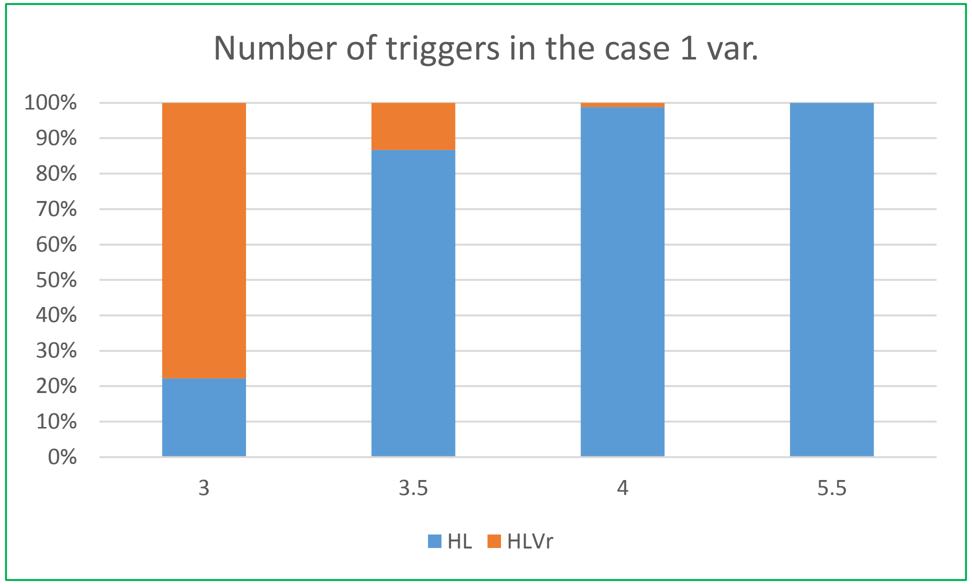
False Alarm Probability

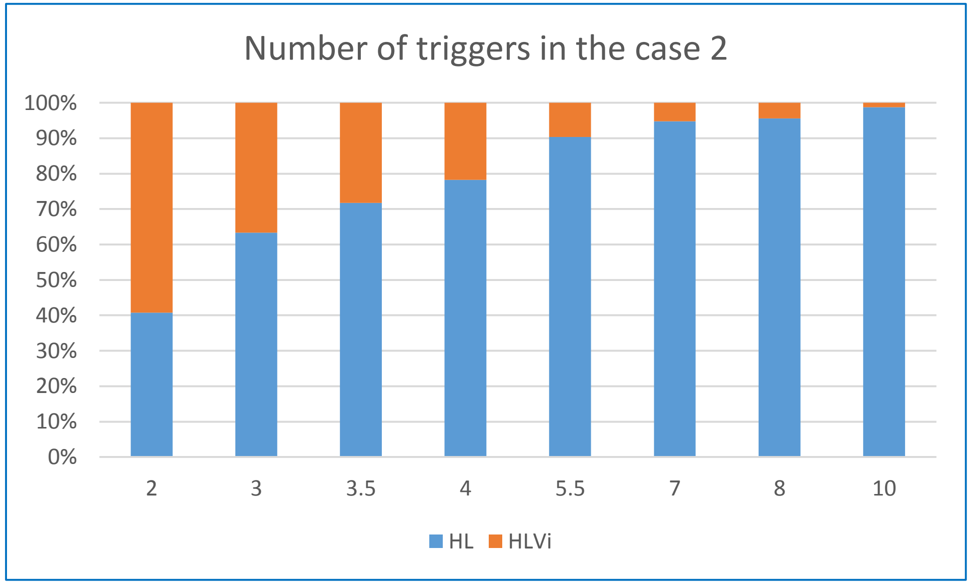


Update the sky localization performance in the case 3 :
Summary of sky localization performance

HLVr = HL + Vrandom
HLVi = HL + Vinjection

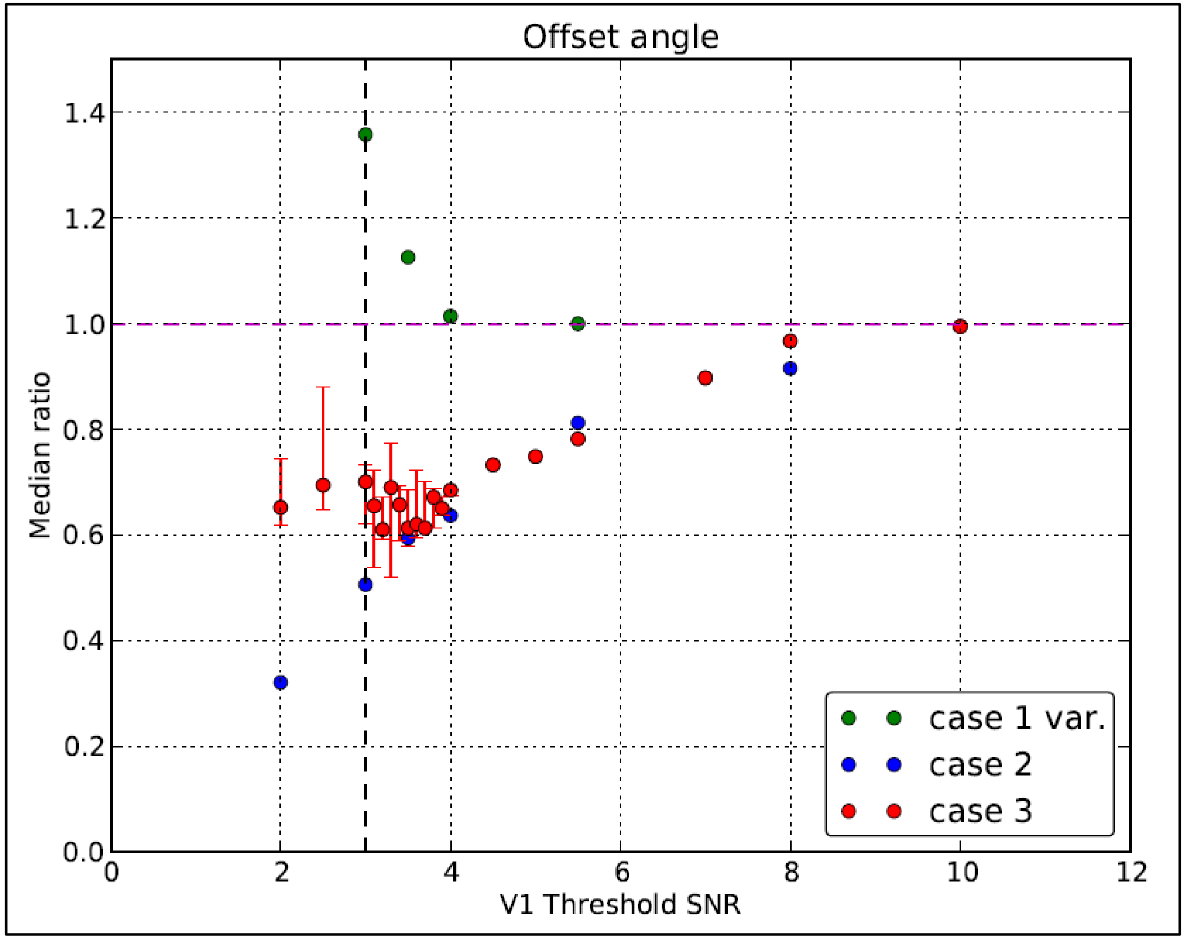


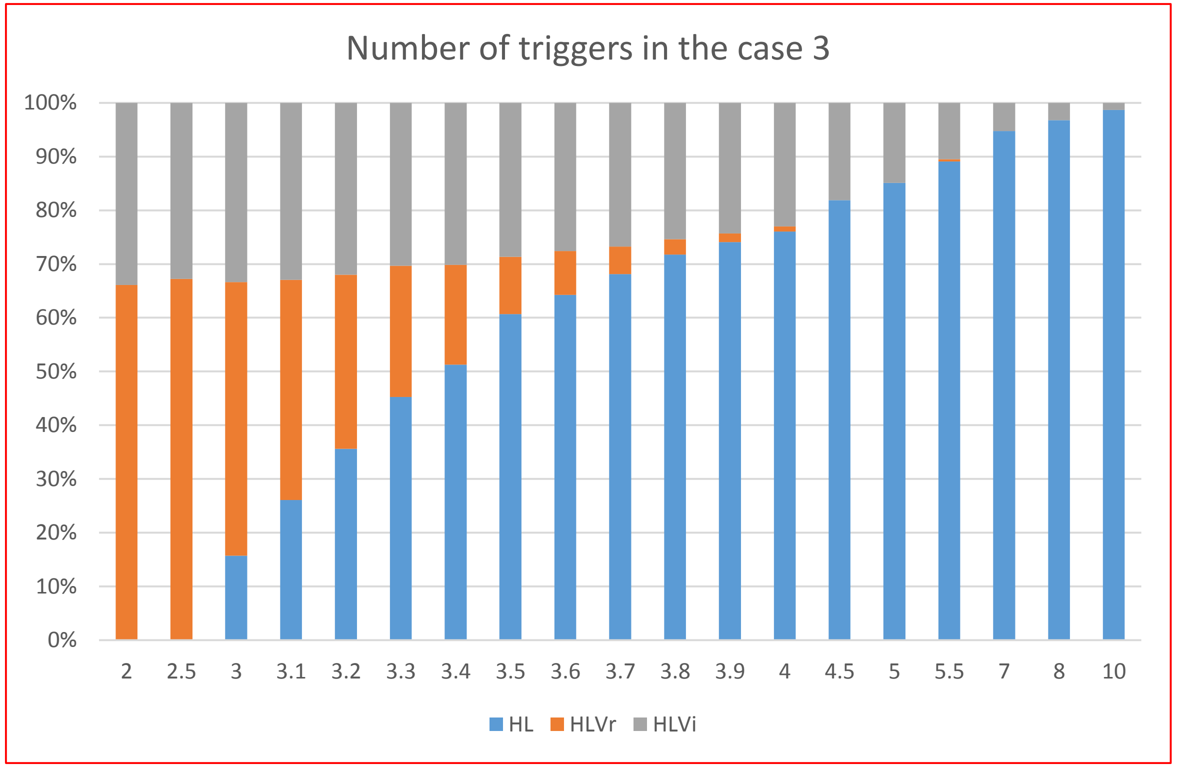




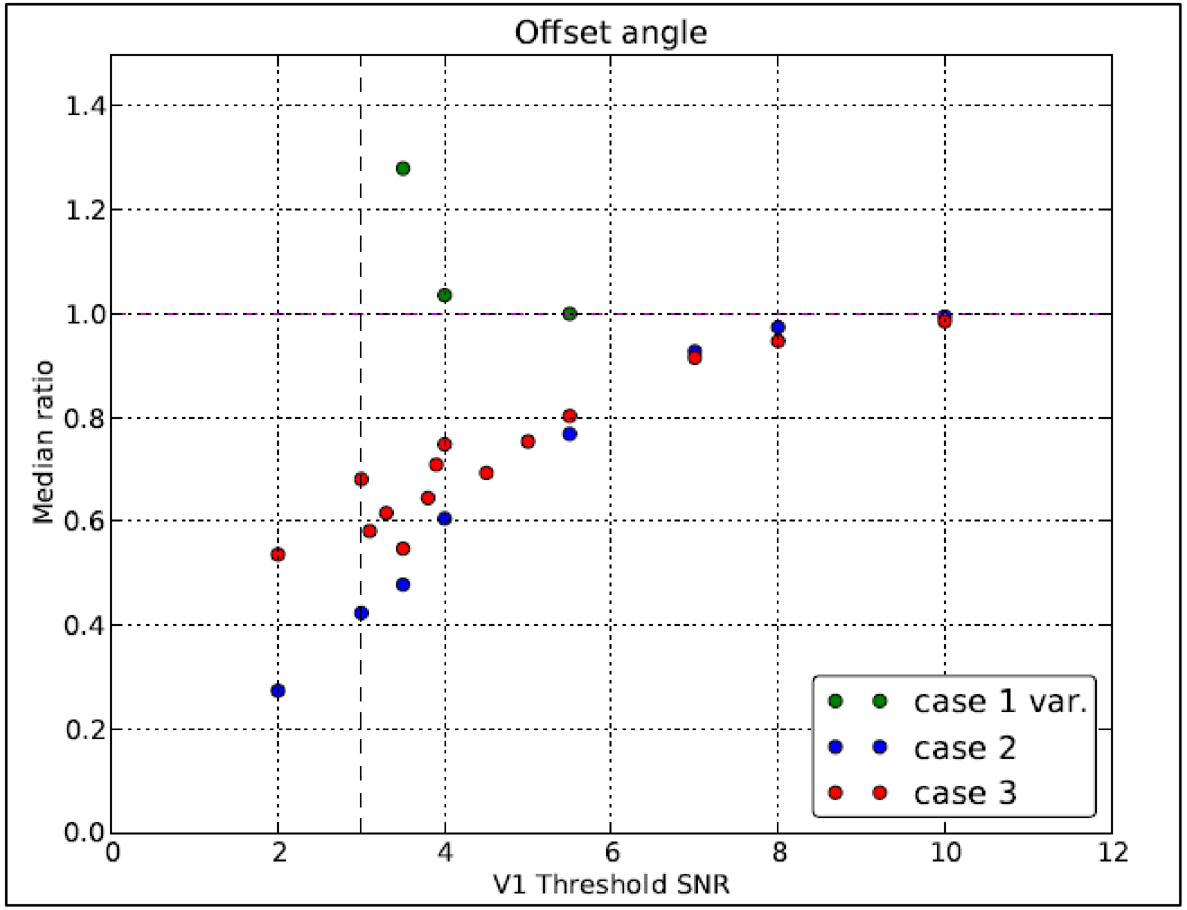
Update the sky localization performance in the case 3 :
Summary of sky localization performance

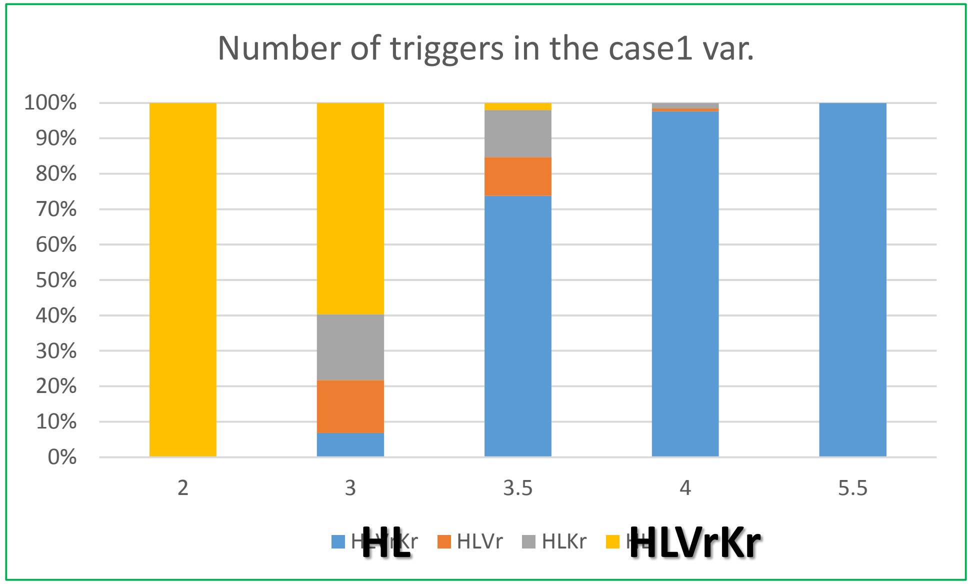
HLVr = HL + Vrandom
HLVi = HL + Vinjection

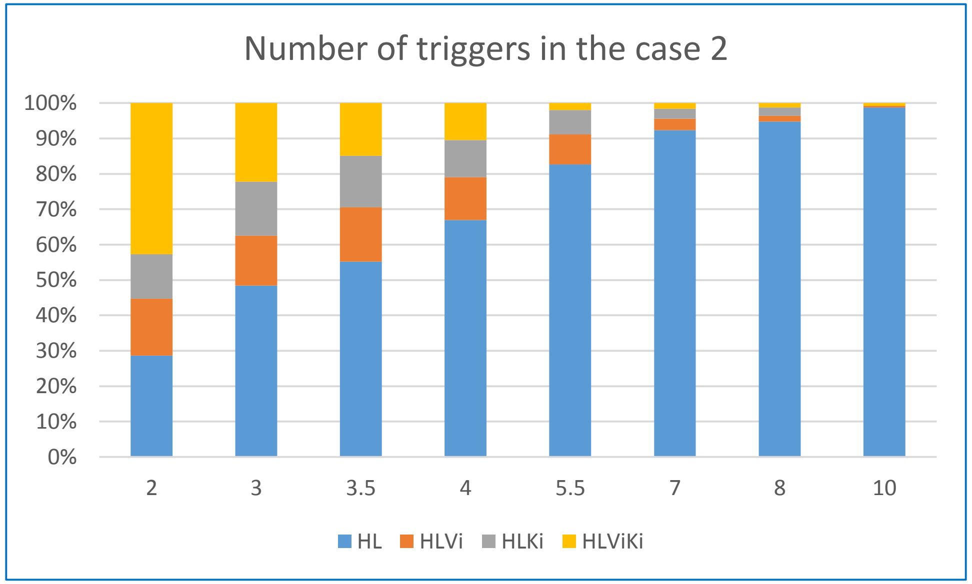




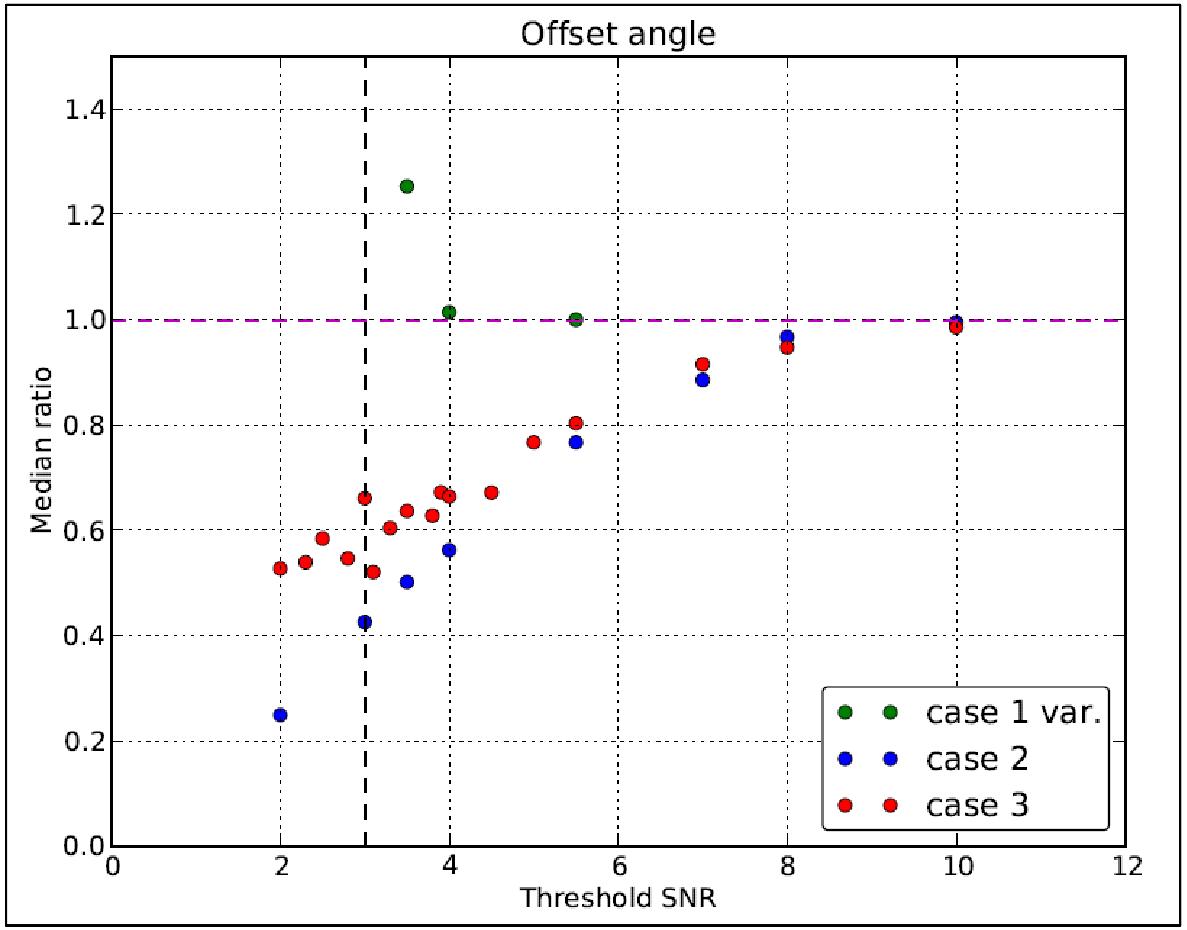
Start to generate skymaps with 4 detectors (one-template search)

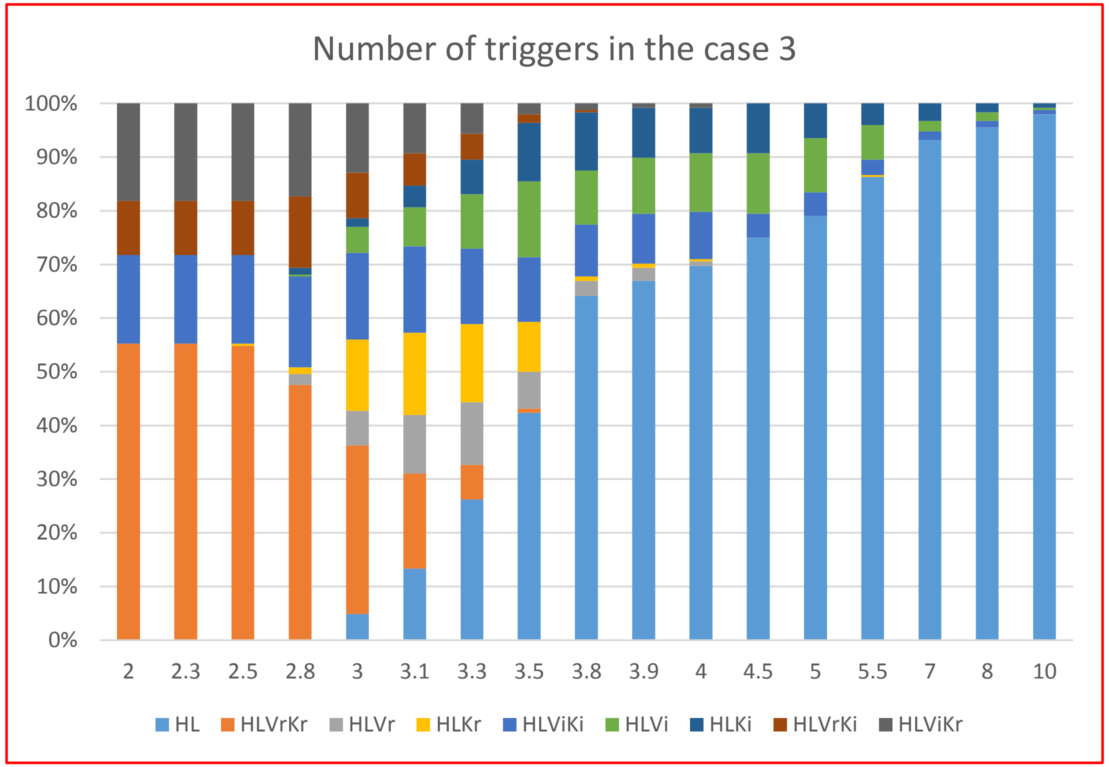






Start to generate skymaps with 4 detectors (one-template search)

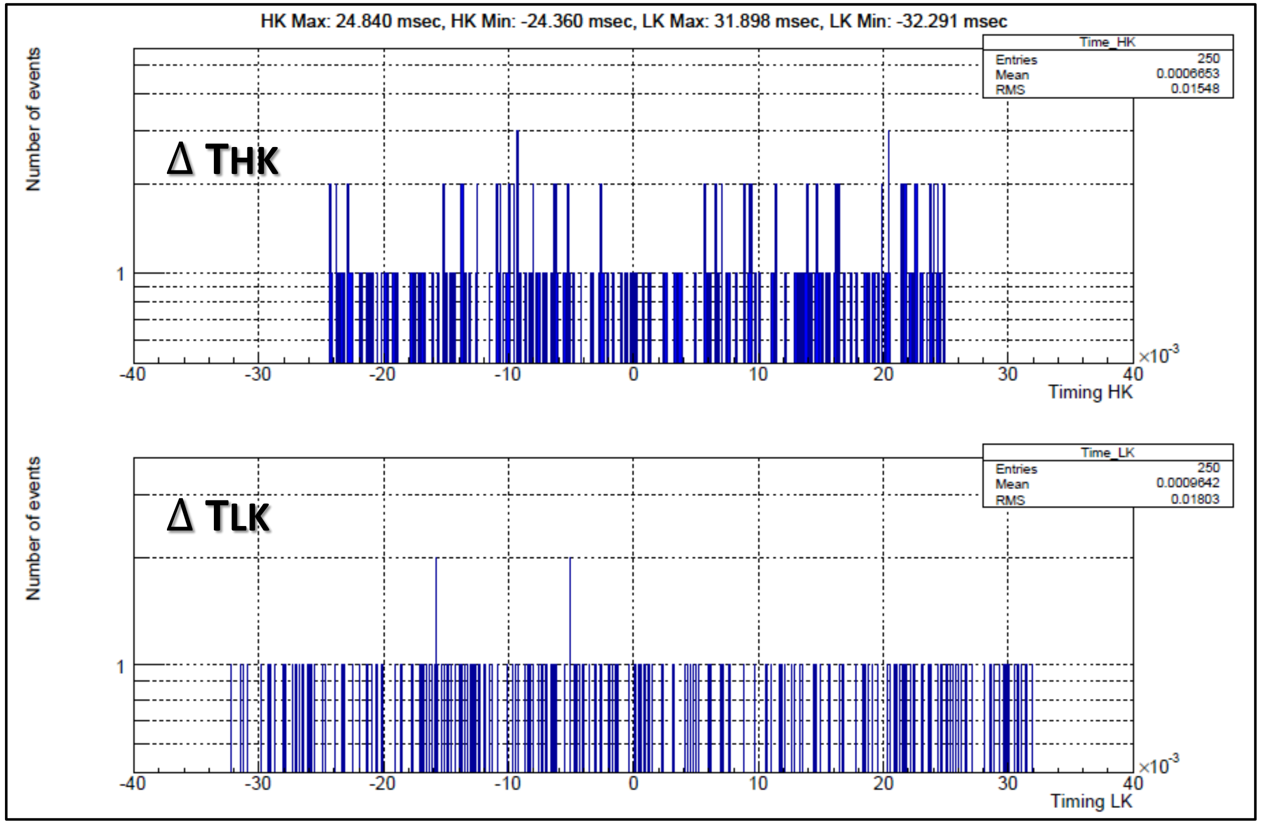


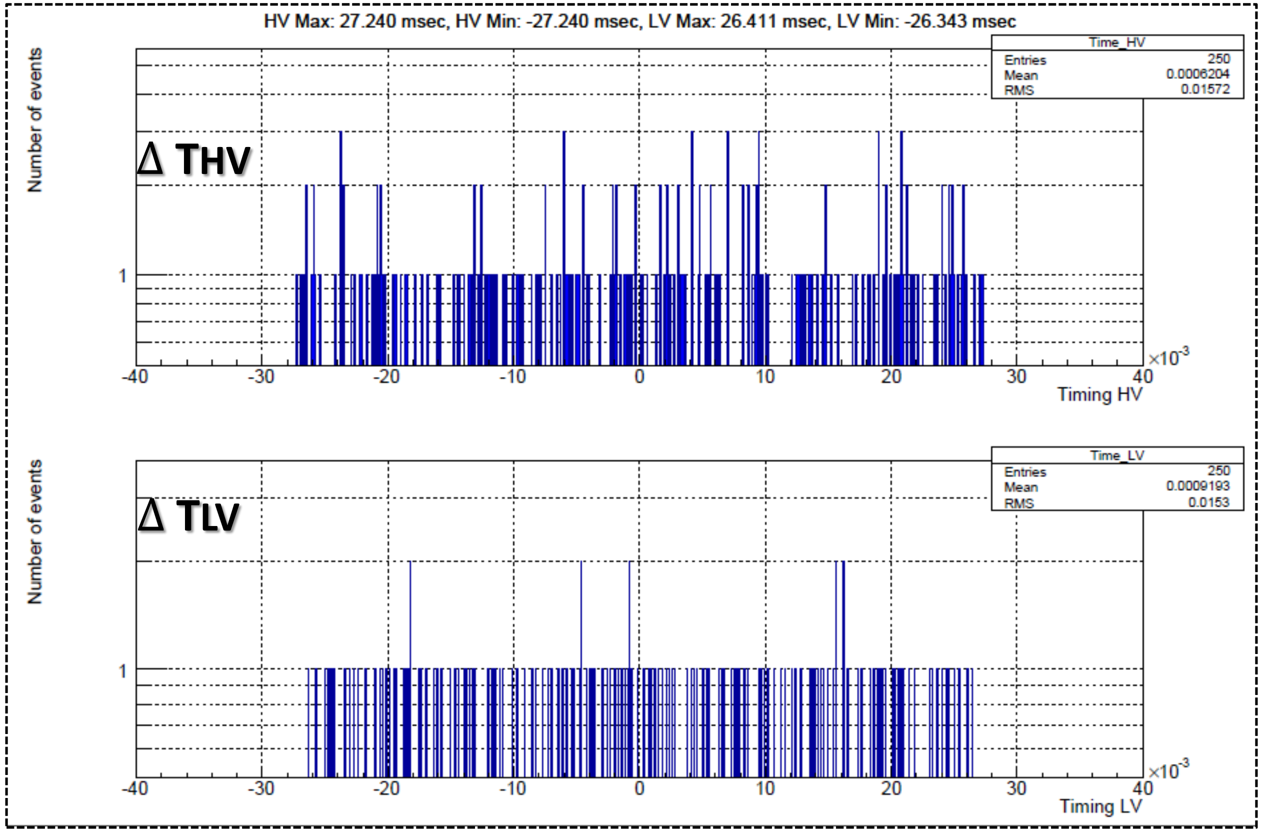


Vr = Vrandom
Vi = Vinjection

Trigger population seems to be strange...

* **Start to generate skymaps with 4 detector**





$\Delta T_{HK} \equiv$ **30 msec**

$\Delta T_{LK} \equiv$ **40 msec**

$T \equiv$ **80 msec**

(T is Time window for searching K1 trigger)

$\Delta T_{HV} \equiv$ **35 msec**

$\Delta T_{LV} \equiv$ **35 msec**

$T \equiv$ **70 msec**

(Time window for searching v1 trigger)

HL → HL or HLV or HLK or HLVK

1.1 Genetaing random triggers : $\mathrm{V_{random}}$, $\mathrm{K_{random}}$

- SNR = Raodom above a threshold SNR, following measured O1 SNR distribution.
- Time = $t_0 + \Delta t$
  - $t_0 = t_{\mathrm{H1}}$ if $\mathrm{SNR_{H1}} > \mathrm{SNR_{L1}}$, otherwise $t_0 = t_{\mathrm{L1}}$
  - Δt = random uniform number:
    from −35 ms to 35 ms, for V1.
    from ms to ms, for K1.
- Phase = random uniform number from 0 rad to 2π rad.
- Effective distance $D_{\mathrm{eff}} = 2.26 \times$ detection range $\times$ 8 / SNR

1.2 Generating triggers based on injection parameters : $\mathrm{V_{inj}}$, $\mathrm{K_{inj}}$

- $\mathrm{SNR} = \mathrm{SNR}^{\mathrm{expected}} + \Delta\mathrm{SNR}$
  - $\mathrm{SNR}^{\mathrm{expected}} = 2.26 \times$ detection range $\times$ 8 / D_{eff}
  - $\Delta\mathrm{SNR}$ = random Gaussian(0, 1).
  - D_{eff} = injection meta data
  - detection range for V1 = 54 Mpc × 20 Mpc / 70 Mpc
  - detection range for K1 = 54 Mpc × 20 Mpc / 70 Mpc
- Time = $t^{\mathrm{expected}} + \Delta t$
  - t^{expected} = injection meta data
  - Δt = random Gaussian(0,1 ms).
- Phase = $\phi_0 + \Delta\phi$
  - $\phi_0 = \phi_{\mathrm{H1}} - \Delta\phi^{\mathrm{expected}}_{\mathrm{HV}}$ if $\mathrm{SNR_{H1}} > \mathrm{SNR_{L1}}$, otherwise $\phi_0 = \phi_{\mathrm{L1}} - \Delta\phi^{\mathrm{expected}}_{\mathrm{LV}}$, for V1
  - $\phi_0 = \phi_{\mathrm{H1}} - \Delta\phi^{\mathrm{expected}}_{\mathrm{HK}}$ if $\mathrm{SNR_{H1}} > \mathrm{SNR_{L1}}$, otherwise $\phi_0 = \phi_{\mathrm{L1}} - \Delta\phi^{\mathrm{expected}}_{\mathrm{LK}}$, for K1
    $\phi_{\mathrm{H1}}, \phi_{\mathrm{L1}}$ = injection metadata
    $\Delta\phi^{\mathrm{expected}}_{\mathrm{HV}}, \Delta\phi^{\mathrm{expected}}_{\mathrm{LV}}, \Delta\phi^{\mathrm{expected}}_{\mathrm{HK}}, \Delta\phi^{\mathrm{expected}}_{\mathrm{LK}}$ are generated from injection metadata.
  - $\Delta\phi$ = random Gaussian(0, 0.25 rad).

Note that the Gaussian(μ, σ) corresponds to this function:

$$\mathrm{Gaussian}(\mu,\sigma) \equiv \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

2 Procedure

p_{V1}, p_{K1} = random uniform number from 0 to 1.

Case 1 : V1, K1 triggers are random

$$\mathrm{HL} + \mathrm{V_{random}} + \mathrm{K_{random}} \quad (2)$$

Case 1 var : V1, K1 triggers are random

$$p_{V1} < FAP_{V1} \text{ and } p_{K1} < FAP_{K1} \Rightarrow \mathrm{HL} + \mathrm{V_{random}} + \mathrm{K_{random}} \quad (3)$$
$$p_{V1} > FAP_{V1} \text{ and } p_{K1} < FAP_{K1} \Rightarrow \mathrm{HL} + \quad + \mathrm{K_{random}} \quad (4)$$
$$p_{V1} < FAP_{V1} \text{ and } p_{K1} > FAP_{K1} \Rightarrow \mathrm{HL} + \mathrm{V_{random}} + \quad (5)$$
$$p_{V1} > FAP_{V1} \text{ and } p_{K1} > FAP_{K1} \Rightarrow \mathrm{HL} + \quad + \quad (6)$$

Case 2 : V1, K1 triggers are based on injection parameters

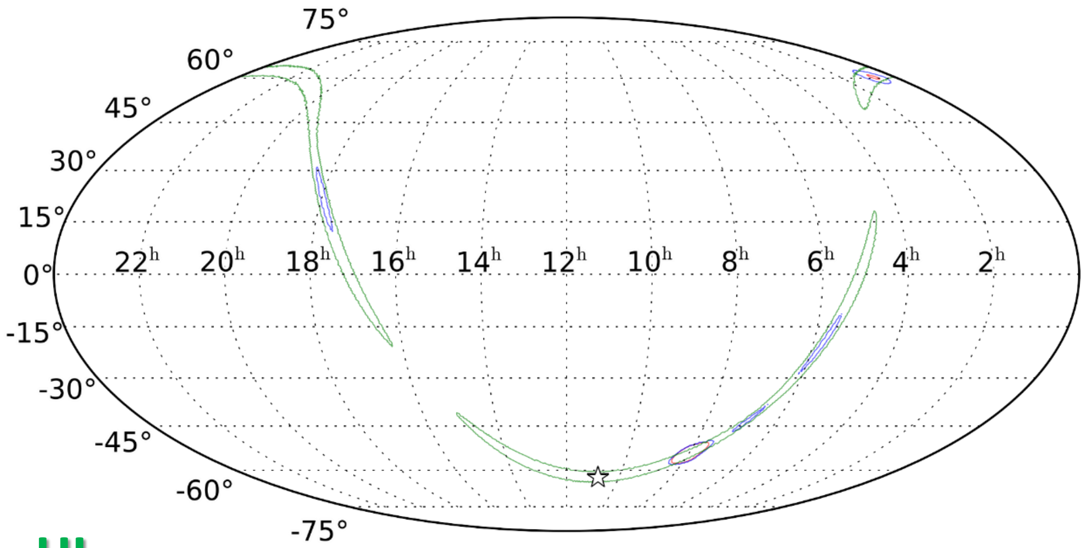
$$\mathrm{SNR}_{V1} > \mathrm{Threshold_{V1}} \text{ and } \mathrm{SNR}_{K1} > \mathrm{Threshold_{K1}} \Rightarrow \mathrm{HL} + \mathrm{V_{inj}} + \mathrm{K_{inj}} \quad (7)$$
$$\mathrm{SNR}_{V1} < \mathrm{Threshold_{V1}} \text{ and } \mathrm{SNR}_{K1} > \mathrm{Threshold_{K1}} \Rightarrow \mathrm{HL} + \quad + \mathrm{K_{inj}} \quad (8)$$
$$\mathrm{SNR}_{V1} > \mathrm{Threshold_{V1}} \text{ and } \mathrm{SNR}_{K1} < \mathrm{Threshold_{K1}} \Rightarrow \mathrm{HL} + \mathrm{V_{inj}} + \quad (9)$$
$$\mathrm{SNR}_{V1} < \mathrm{Threshold_{V1}} \text{ and } \mathrm{SNR}_{K1} < \mathrm{Threshold_{K1}} \Rightarrow \mathrm{HL} + \quad + \quad (10)$$

Case 3 : V1, K1 triggers are either random or based on injection parameters

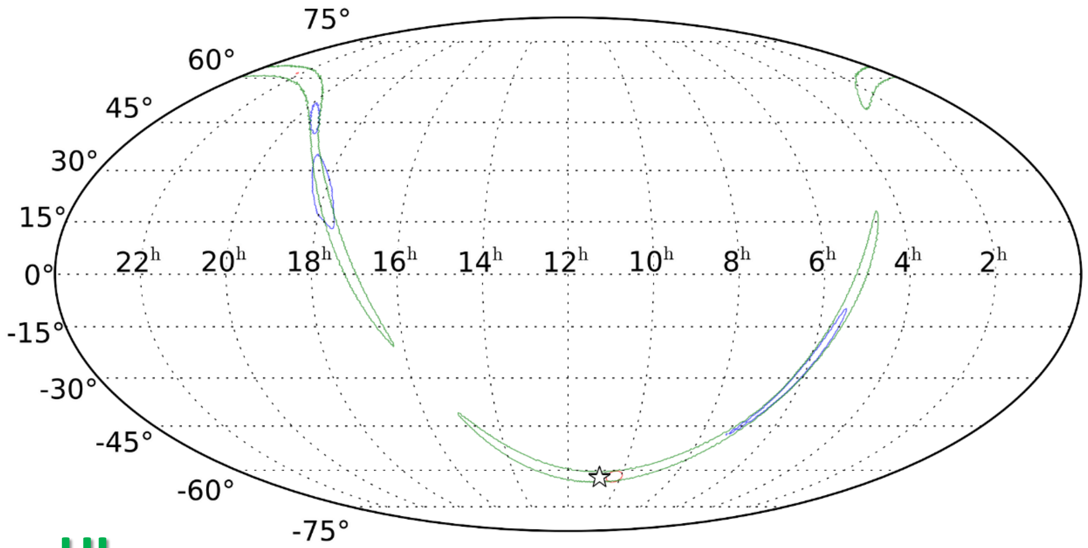
$FAP = FAP(\mathrm{SNR})$ if SNR > Threshold, otherwise $FAP = FAP(\mathrm{Threshold})$

- $p_{V1} < FAP_{V1}$ and $p_{K1} < FAP_{K1}$ $\Rightarrow \mathrm{HL} + \mathrm{V_{random}} + \mathrm{K_{random}}$ (11)
- $p_{V1} < FAP_{V1}$ and,
  $p_{K1} > FAP_{K1}$ and $\mathrm{SNR}_{K1} > \mathrm{Threshold_{K1}}$ $\Rightarrow \mathrm{HL} + \mathrm{V_{random}} + \mathrm{K_{inj}}$ (12)
- $p_{V1} > FAP_{V1}$ and $\mathrm{SNR}_{V1} > \mathrm{Threshold_{V1}}$ and
  $p_{K1} < FAP_{K1}$ $\Rightarrow \mathrm{HL} + \mathrm{V_{inj}} + \mathrm{K_{random}}$ (13)
- $p_{V1} > FAP_{V1}$ and $\mathrm{SNR}_{V1} > \mathrm{Threshold_{V1}}$ and
  $p_{K1} > FAP_{K1}$ and $\mathrm{SNR}_{K1} > \mathrm{Threshold_{K1}}$ $\Rightarrow \mathrm{HL} + \mathrm{V_{inj}} + \mathrm{K_{inj}}$ (14)
- $p_{V1} < FAP_{V1}$ and
  $p_{K1} > FAP_{K1}$ and $\mathrm{SNR}_{K1} < \mathrm{Threshold_{K1}}$ $\Rightarrow \mathrm{HL} + \mathrm{V_{random}} +$ (15)
- $p_{V1} > FAP_{V1}$ and $\mathrm{SNR}_{V1} < \mathrm{Threshold_{V1}}$ and
  $p_{K1} < FAP_{K1}$ $\Rightarrow \mathrm{HL} + \quad + \mathrm{K_{random}}$ (16)
- $p_{V1} > FAP_{V1}$ and $\mathrm{SNR}_{V1} > \mathrm{Threshold_{V1}}$ and
  $p_{K1} > FAP_{K1}$ and $\mathrm{SNR}_{K1} < \mathrm{Threshold_{K1}}$ $\Rightarrow \mathrm{HL} + \mathrm{V_{inj}} +$ (17)
- $p_{V1} > FAP_{V1}$ and $\mathrm{SNR}_{V1} < \mathrm{Threshold_{V1}}$ and
  $p_{K1} > FAP_{K1}$ and $\mathrm{SNR}_{K1} > \mathrm{Threshold_{K1}}$ $\Rightarrow \mathrm{HL} + \quad + \mathrm{K_{inj}}$ (18)
- $p_{V1} > FAP_{V1}$ and $\mathrm{SNR}_{V1} < \mathrm{Threshold_{V1}}$ and
  $p_{K1} > FAP_{K1}$ and $\mathrm{SNR}_{K1} < \mathrm{Threshold_{K1}}$ $\Rightarrow \mathrm{HL} + \quad +$ (19)

*** Start to generate skymaps with 4 detector (V1, K1 threshold = 3.5)**



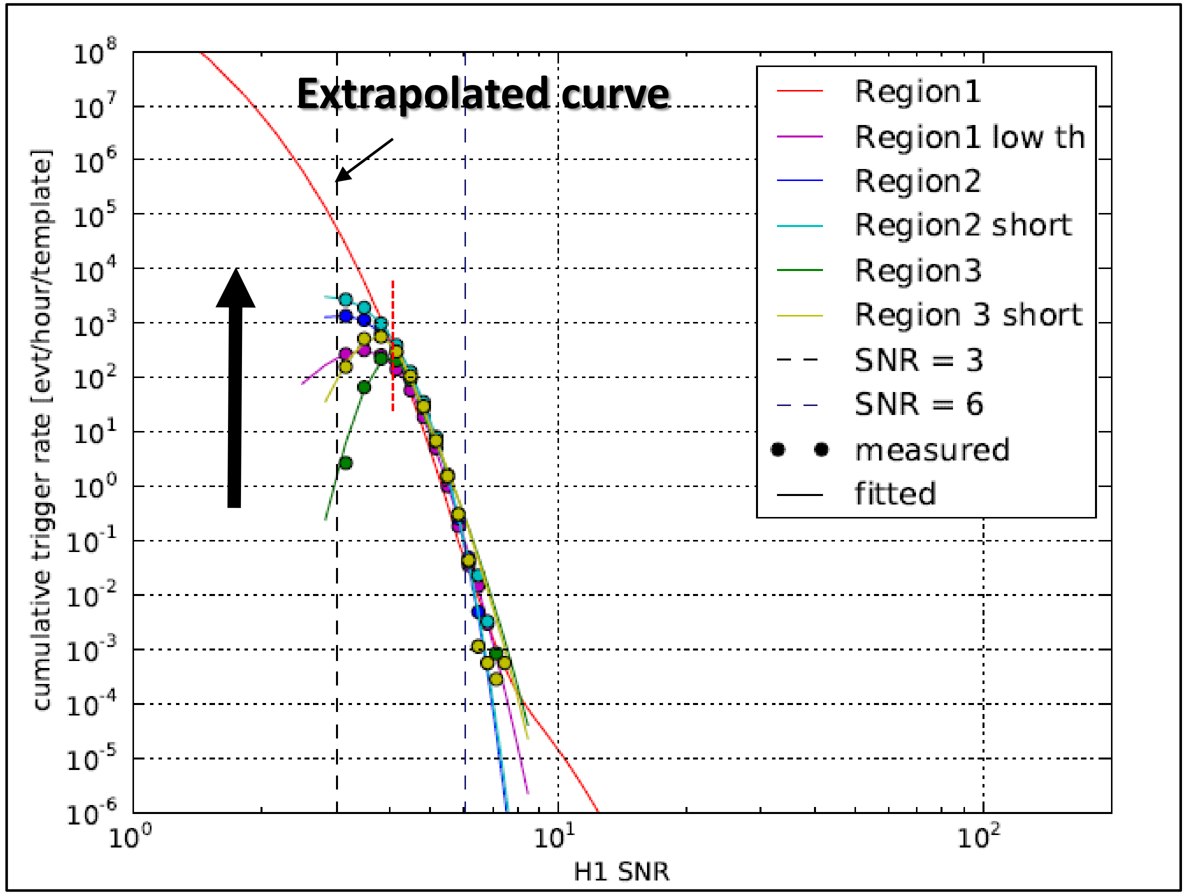
HL
HL + Vrandom
HL + Vrandom + Krandom



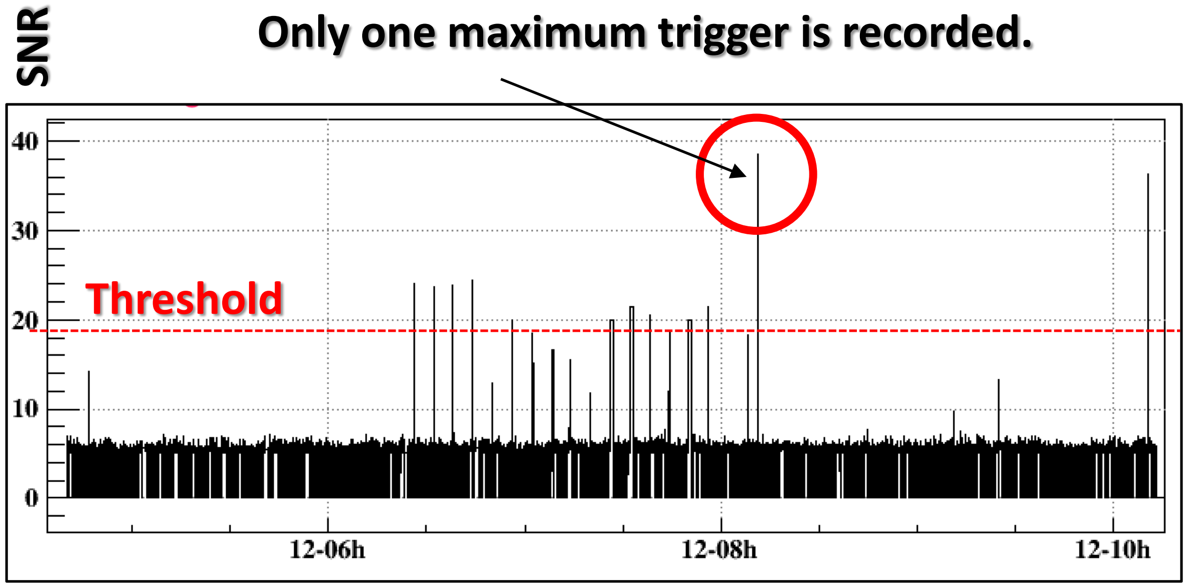
HL
HL + Vinj
HL + Vinj + Kinj

* Investigate the SNR distribution at low SNR distribution

SNR distribution



Only one maximum trigger is recorded.



Certain collecting time length

According to this, collecting time effect contributes to SNR distribution at low SNR.
Sometimes FAP has pseudo saturation before getting to 1.

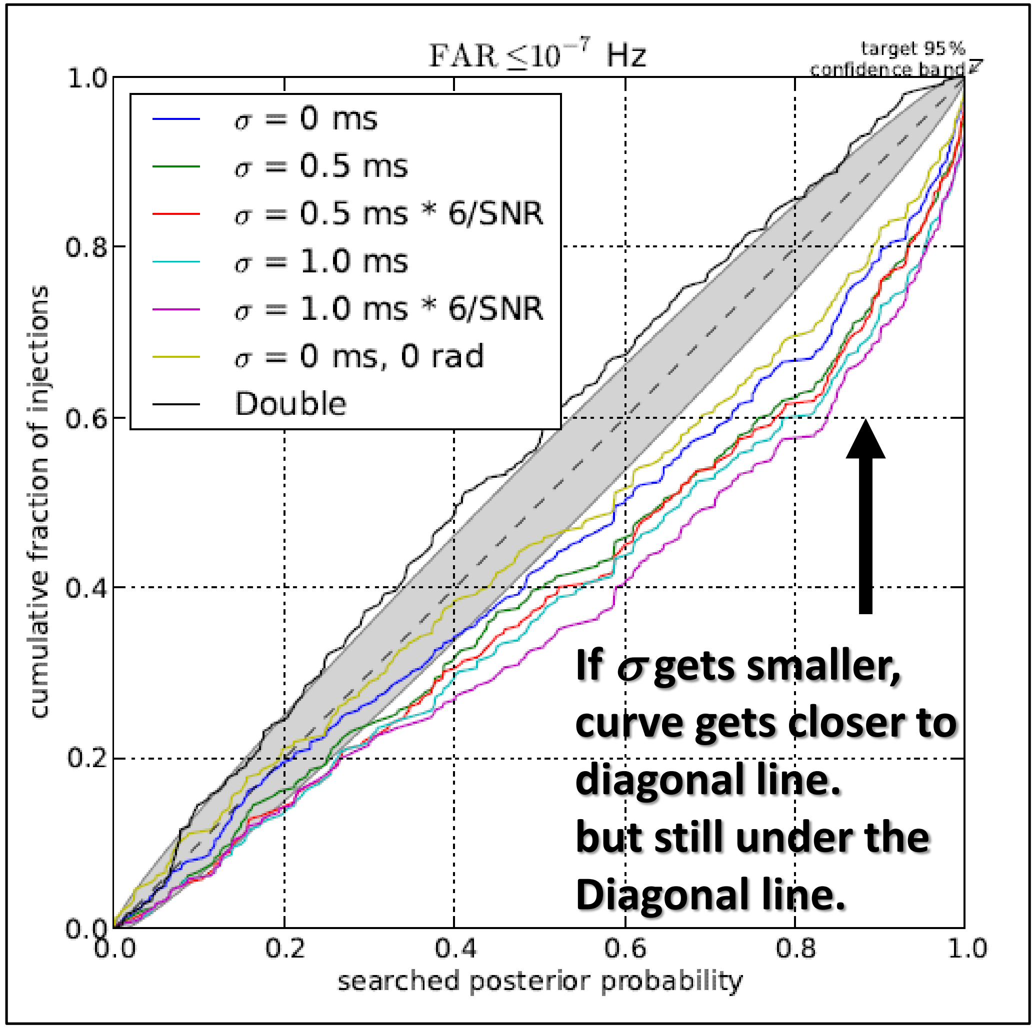
At low SNR, if collecting time gets shorter,
1) the saturation gets better, and 2) curves get close to red line(extrapolated one)

At High SNR, there are mostly no differences → distributions don't depend on how to analyze, and templates.

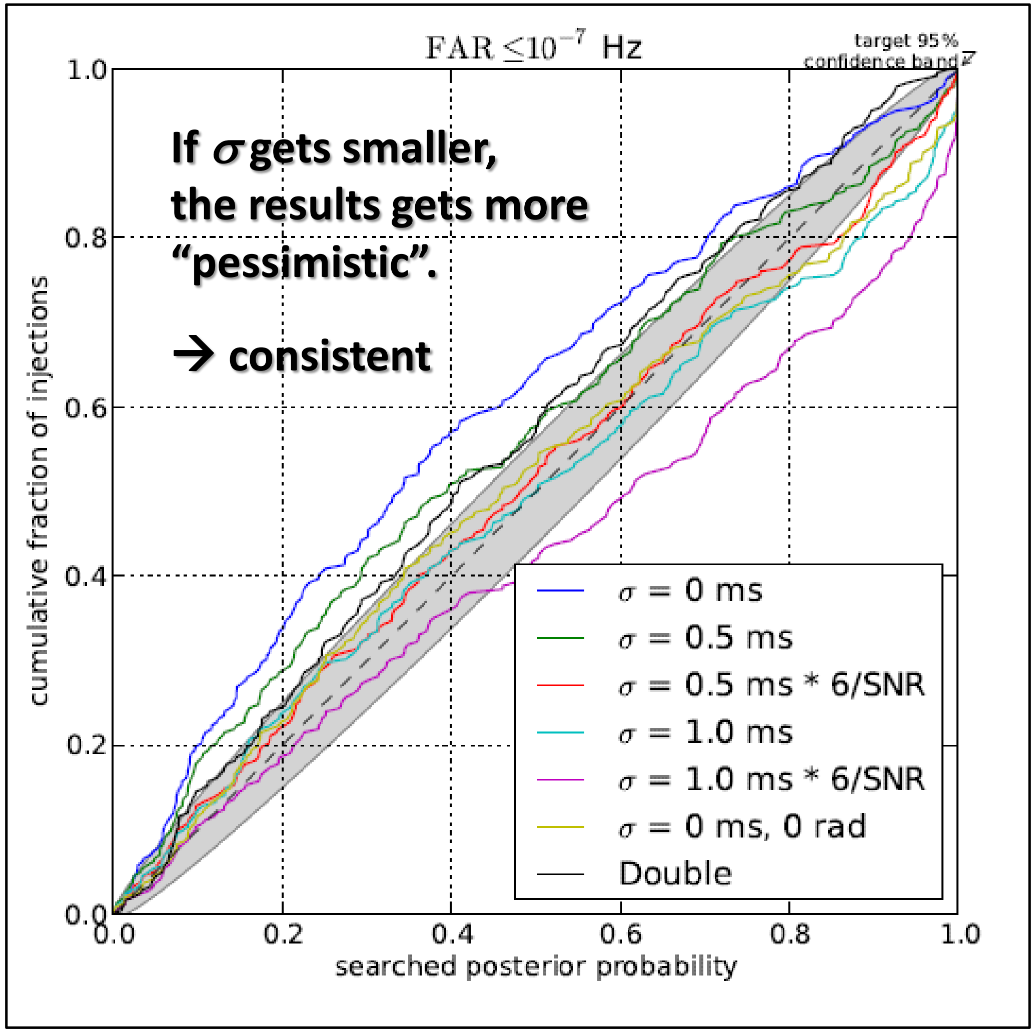
* Investigate relation between the P-P plot and timing fluctuation

* (Arriving time) = (meta data) + (Gaussian)

Gaussian $\sigma_{\mathrm{Time}} = 1$ ms (Const.) → $\sigma_{\mathrm{Time}} = Time$ or $Time \times \frac{6}{\mathrm{SNR}}$



Case 3 (HL, HLVr, or HLVi)



Case 2 (HL, or HLVi)

We cannot judge if the statistics is optimistic or pessimistic, from this P-P plot.