Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
ICSC Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing

ICSC Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing

Algorithm optimization to improve continuous gravitational-wave searches WP3 – Flagship Use Case 2.3.1

Lorenzo Pierini

Spoke 2 annual meeting, CINECA, 19/12/2023

Introduction

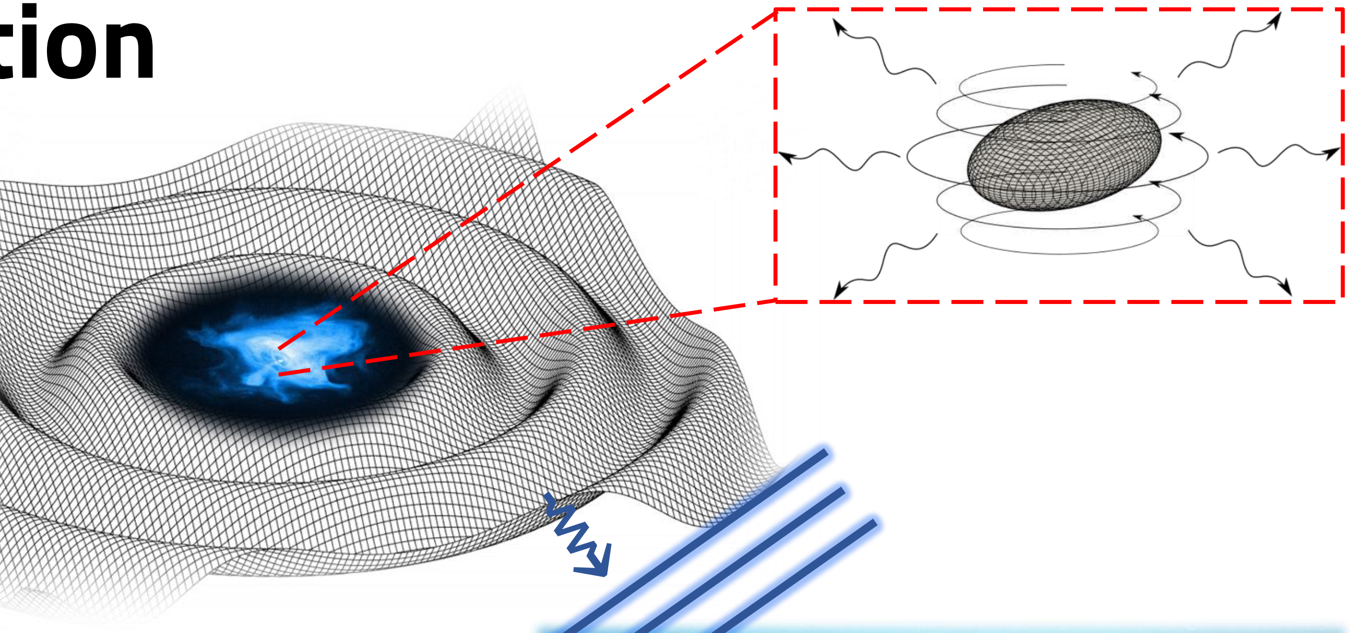
About me:

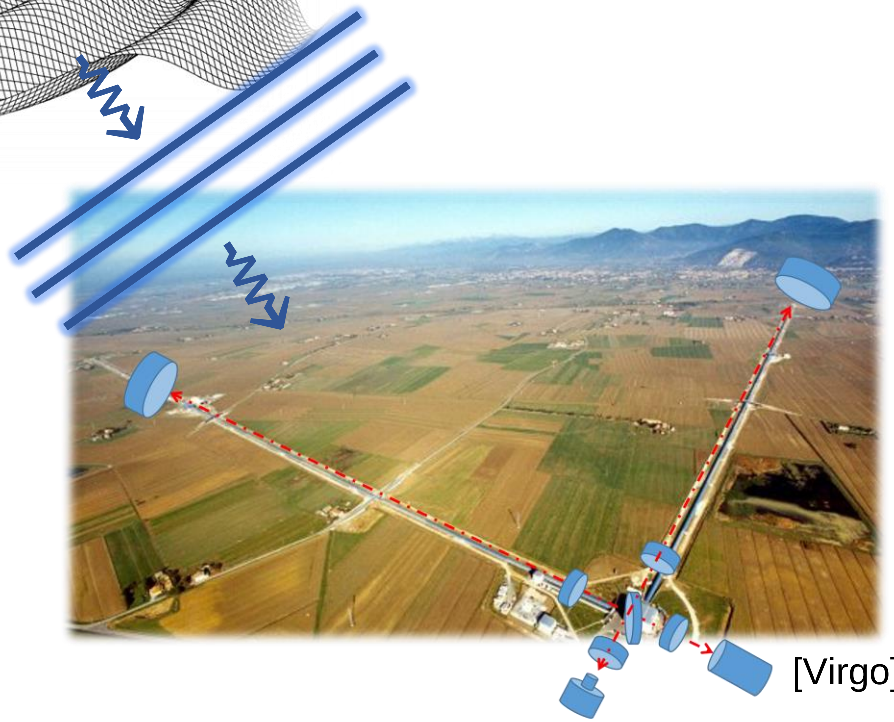
Lorenzo Pierini
PhD in physics, defended in May 2023
INFN technologist 100% ICSC since May 2023

About us: Virgo Rome group

Topic: Continuous gravitational waves

- Perturbations of the space-time, predicted by General Relativity.
- Emitted by rotating, deformed neutron stars (and more exotic sources).
- Can be detected by Earth-based detectors: LIGO, Virgo, KAGRA.
- Not yet detected so far.





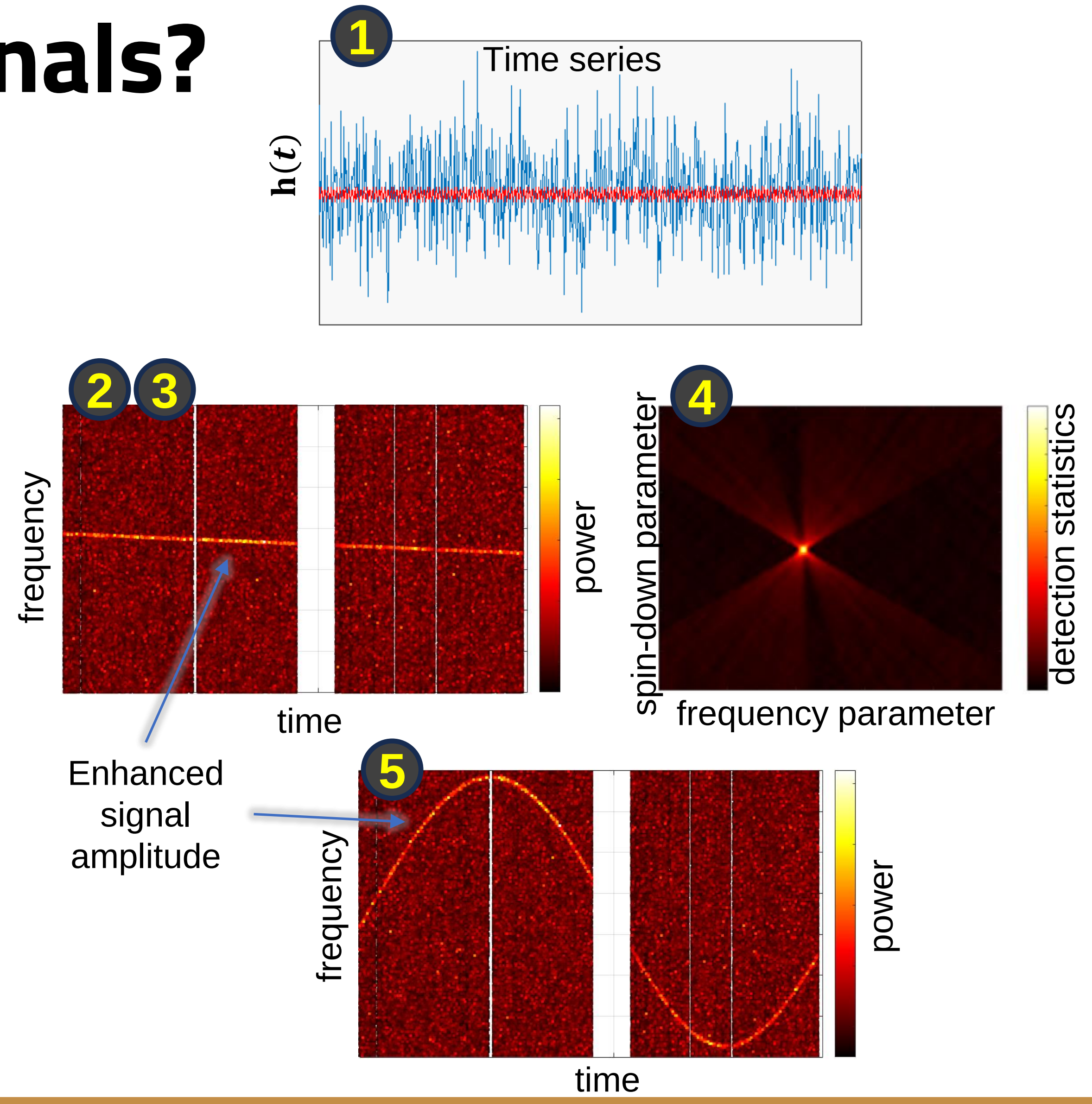
[Virgo]

People involved:

- Pia Astone (**PI**) (INFN Roma)
- Lorenzo Pierini (INFN Roma)
- Marco Serra (INFN Roma)
- Stefano Dal Pra (INFN CNAF)
- Cristiano Palomba (INFN Roma)

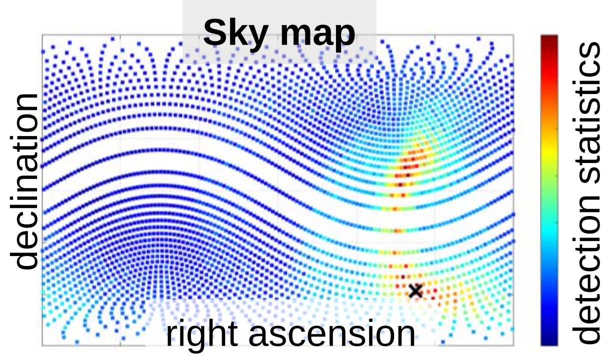
How do we search those signals?

1) Detector output: calibrated time series. Weak signals deeply embedded in noise.
2) Data processed to obtain time-frequency maps.
3) The expected signal is nearly monochromatic, with a slow frequency variation (spin-down).
4) To recover the signal parameters, we apply the Hough transform to the map. Most significant outliers identify a possible signal.
5) However, the observed signal is distorted by the Doppler effect due to the Earth motion! This distortion changes for any sky location.

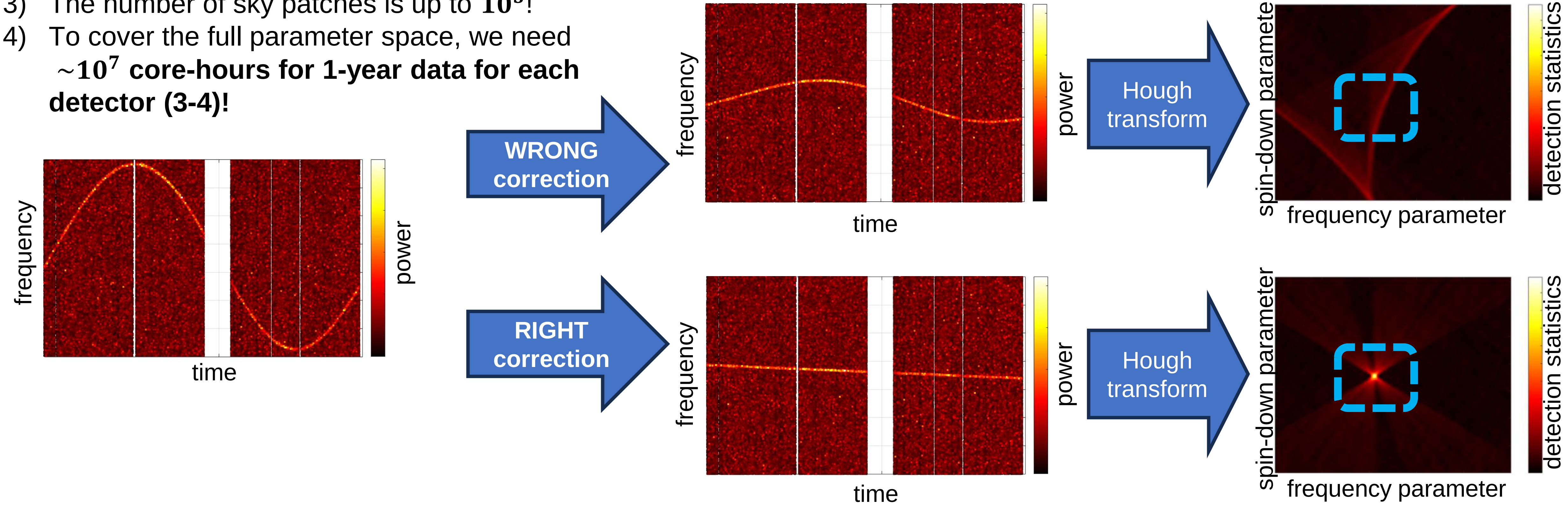


Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
ICSC Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing

The computational problem

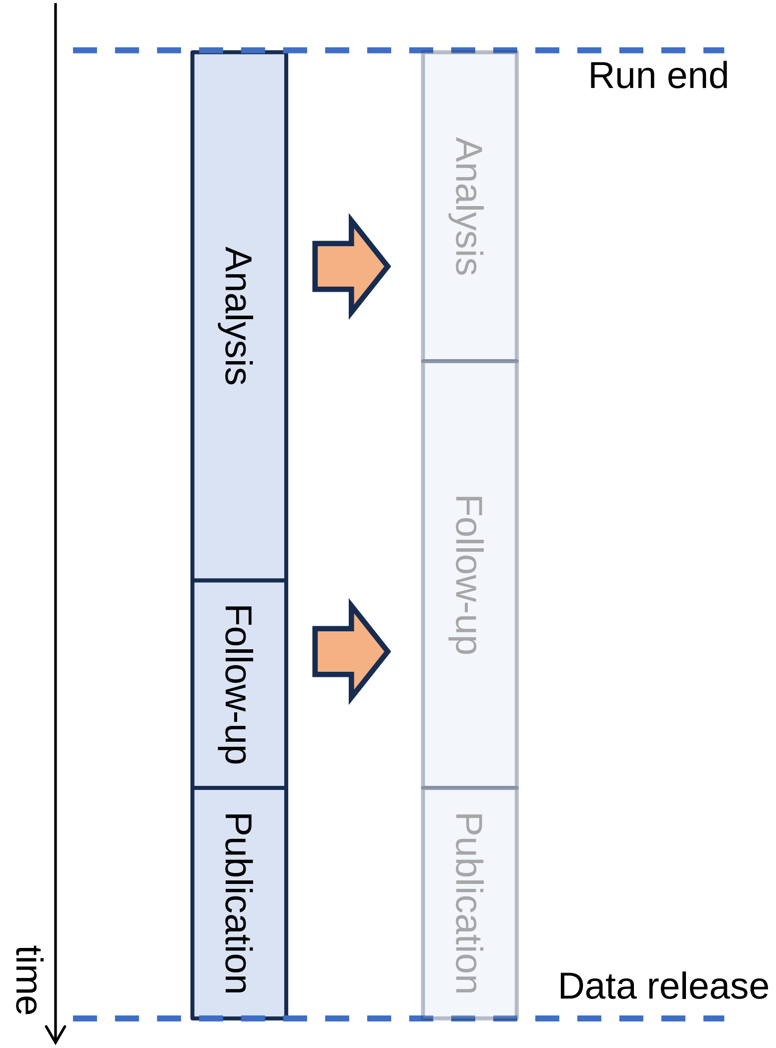


1) According to a discretized sky map, we correct the Doppler effect for ANY POINT in that map.
2) Only the correction that matches the right position of the source maximizes the detection statistics.
3) The number of sky patches is up to $\mathbf{10^5}$!
4) To cover the full parameter space, we need **$\sim 10^7$ core-hours for 1-year data for each detector (3-4)!**



The computing bounds

1) An all-sky & all-frequencies search on 1-year data for 3 detectors requires $\sim 30 \cdot 10^6$ core hours.
   (A typical 10-14 HS06/core is considered)
2) Follow-up: the goal is to select up to 10^9 signal candidates from the Hough maps, to be further processed and verified.
3) The follow-up itself is a refined search based on the Hough transform: another computationally heavy step!
4) There is a tight schedule to publicly release the data too: long computing time limits the number of signal candidates that we are able to verify.
5) Optimizing the algorithm is crucial: shortening the computing time leaves room to analyze a higher number of candidates, thus increasing the overall search sensitivity.



The opportunity of ICSC

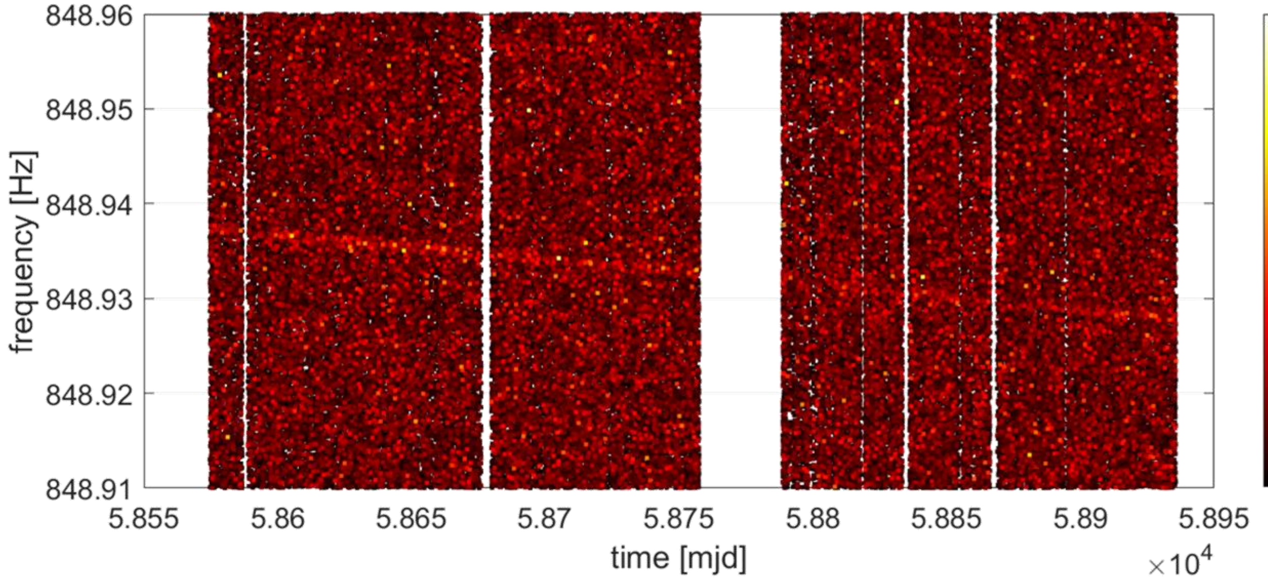
- First part: code optimization.
  - The heavy part of the algorithm is the calculation of the Hough transform.
  - The Hough transform is implemented and optimized in a serial code that avoids parallelization.
  - The main goal is to implement different versions of the Hough transform that exploit parallel architectures, in particular GPU devices.
  - Then, to adapt the analysis code to fully run on GPUs and CPUs.

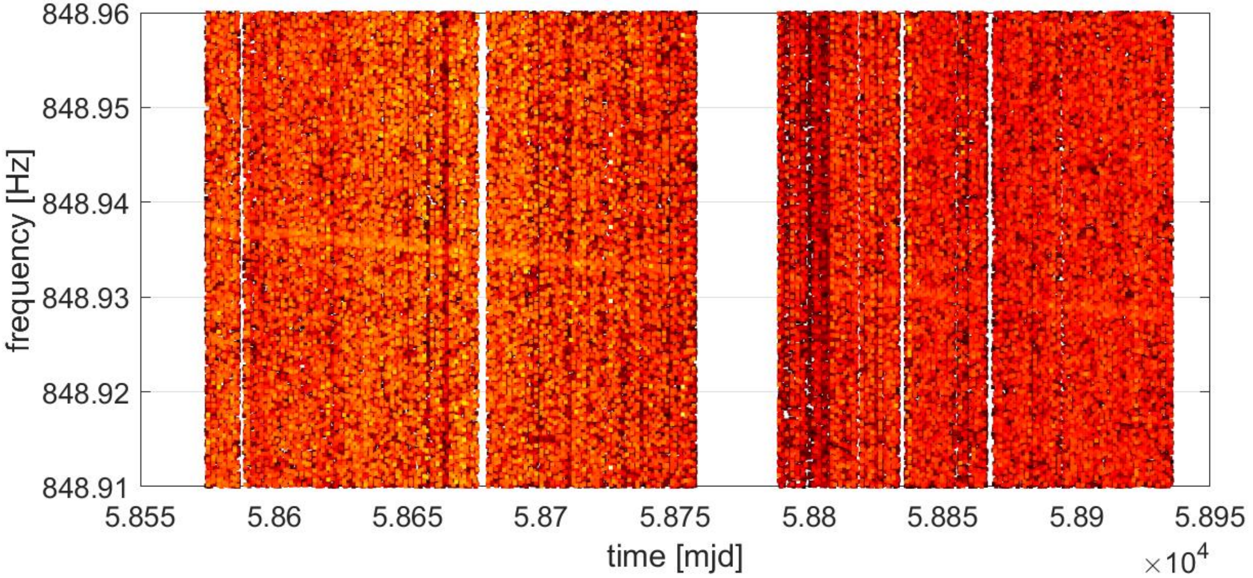
- Second part: extensive tests.
  - The large scale of ICSC is an exceptional opportunity to test the new algorithms: the amount of available resources matches the typical infrastructure needed to obtain results according to out tight time scales.
  - Extensive tests: perform long timescale analysis to test the stability of the code running on a high number of cores and for long jobs.
  - We hope that in future the ICSC resources will be available for next years research: gravitational wave searches are planned to go on with new detectors!

Spare slides
-
Technical details

Input: peakmap (sparse time-frequency map)

Normalized power spectral density:
NOT USED
Not robust with respect to spectral lines



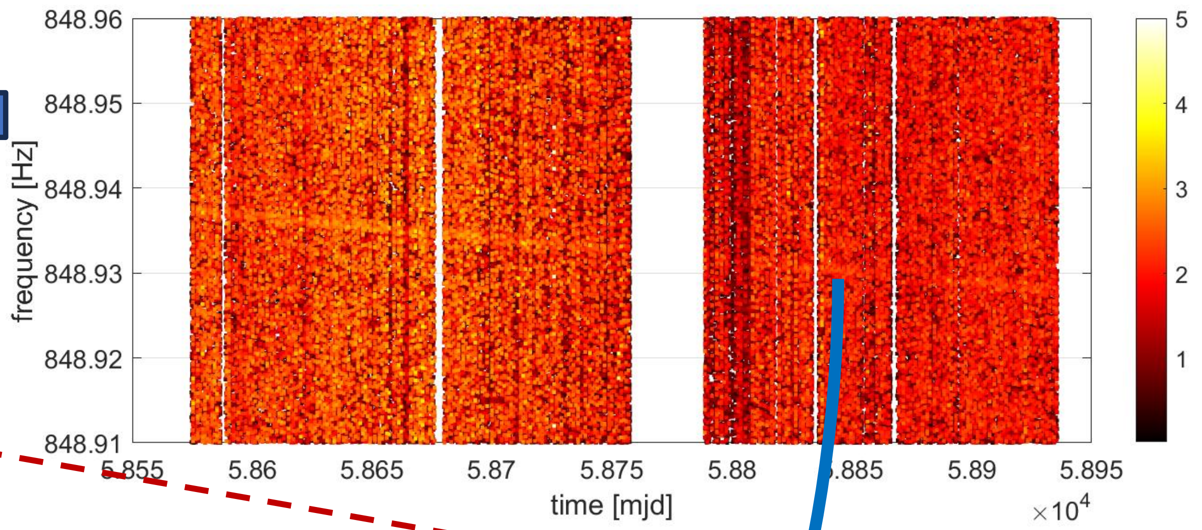


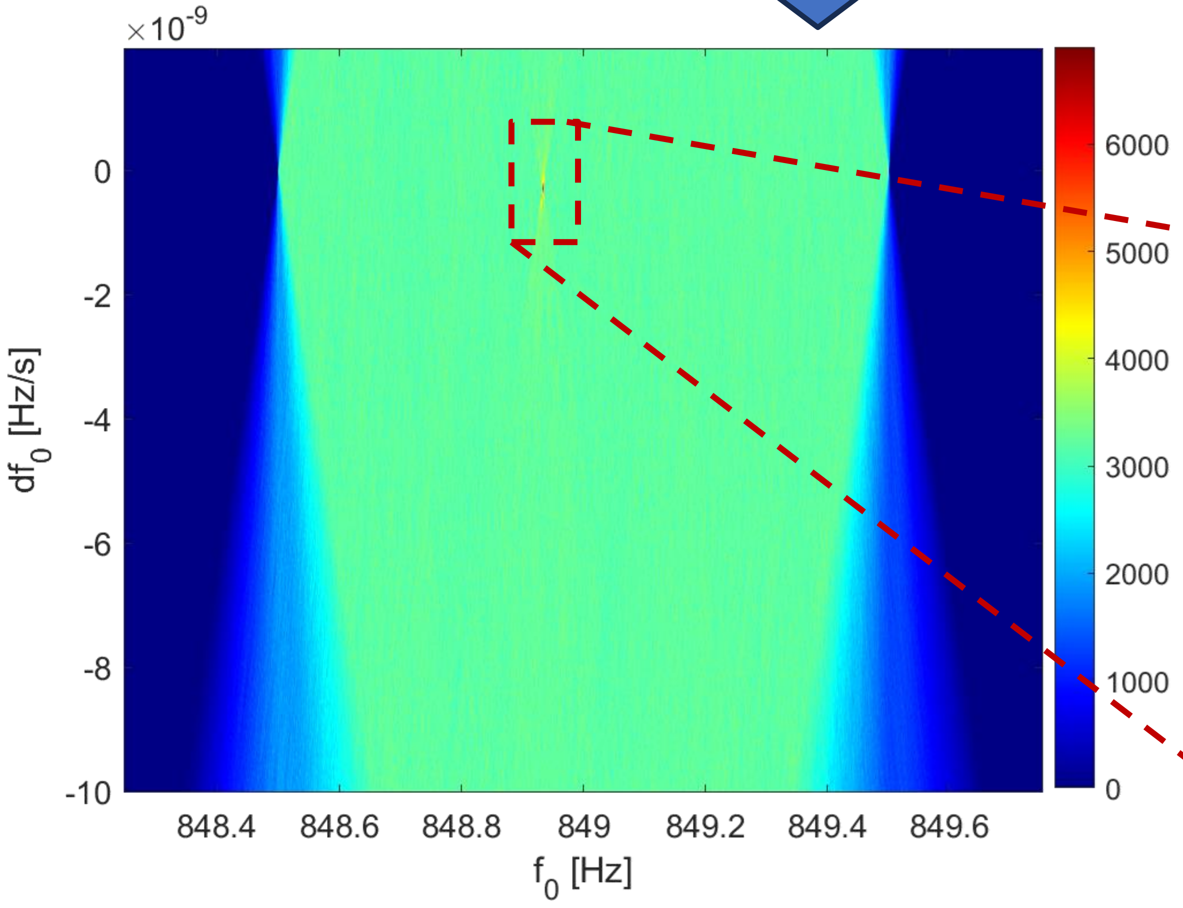
Instead, typically used
WEIGHTS

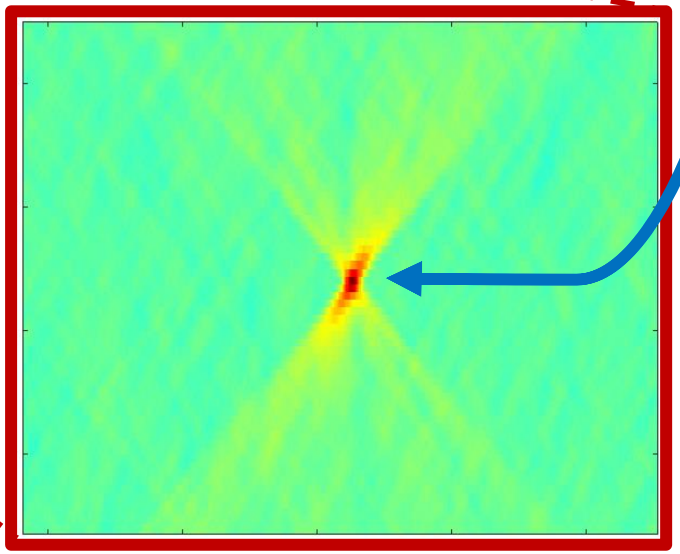
$$w_i \propto \frac{F(\mathrm{sky}\,;t_i)}{n(\nu\,;t_i)}$$

$F(\mathrm{sky};t_i)$: detector power response for a given sky position at a given time t_i.
$n(\nu;t_i)$: noise power at the frequency ν at the given time t_i.

Output: Hough map



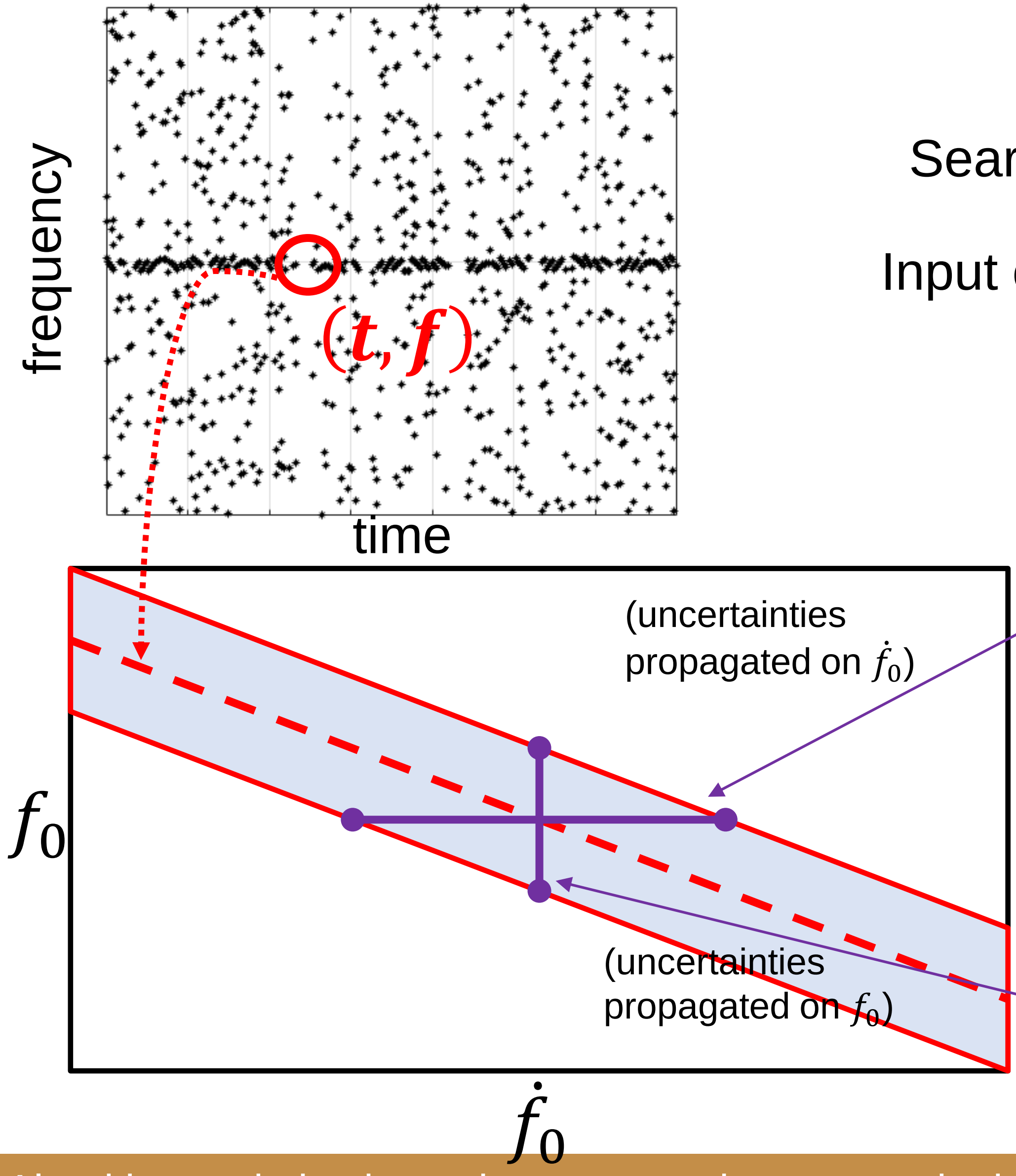




If a signal is present, it generates an excess of counts at the location corresponding to its parameters

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
ICSC Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing

FH idea (single point)



Searched pattern: $f = f_0 + \dot{f}_0(t - t_0)$

Input coordinates: $\left\{t_k \; ; \; f_k \pm \frac{\delta f_H}{2}\right\}$ → output space $(f_0 \, , \dot{f}_0)$

$$\dot{f}_0 = -\frac{f_0}{t - t_0} + \frac{f}{t - t_0}$$

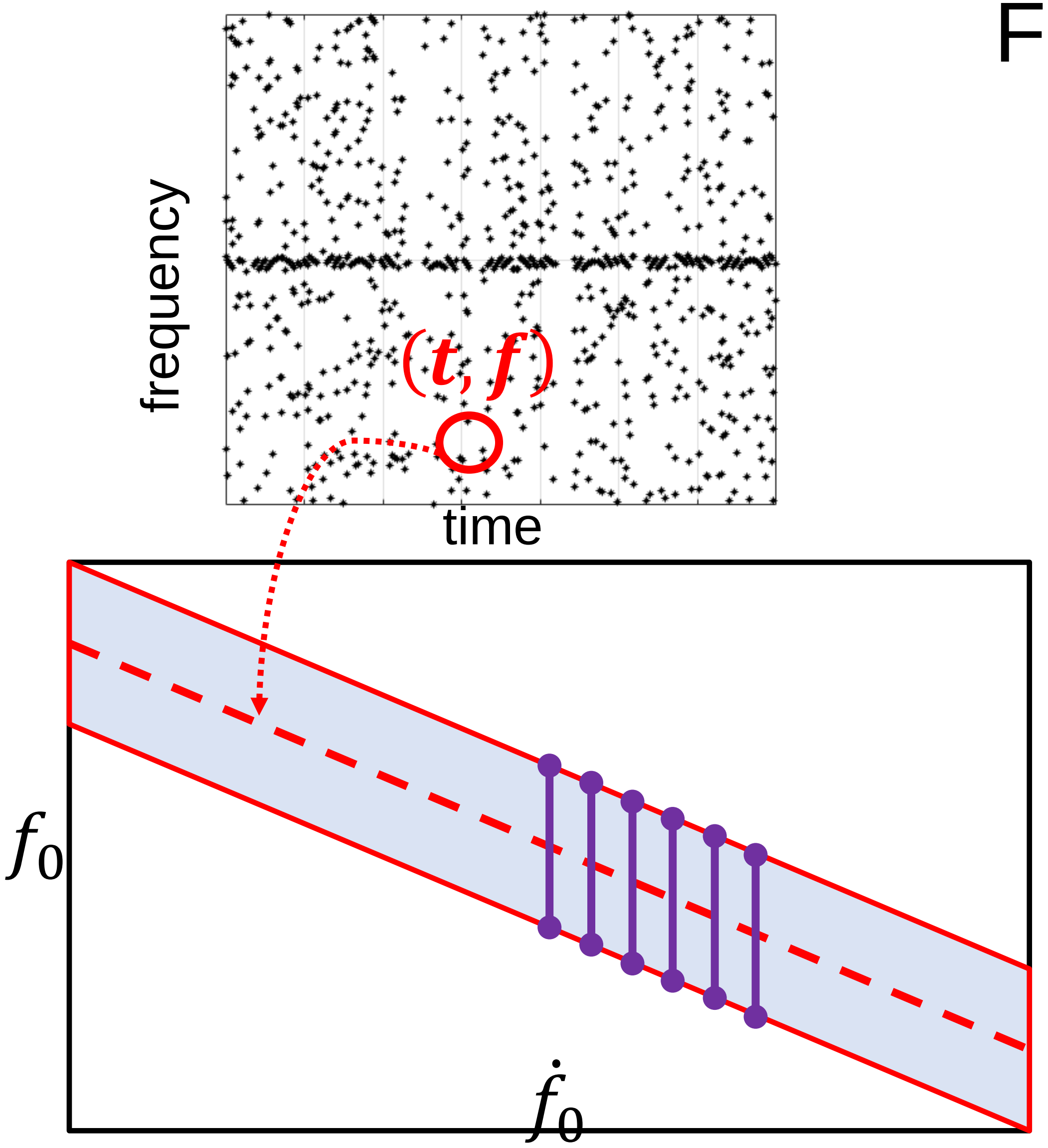
$$-\frac{f_0}{t - t_0} + \frac{f + \delta f_H/2}{t - t_0} < \dot{f}_0 < -\frac{f_0}{t - t_0} + \frac{f + \delta f_H/2}{t - t_0}$$

Actually implemented:

$$f_0 = f - \dot{f}_0(t - t_0)$$

$$f - \dot{f}_0(t - t_0) - \frac{\delta f_H}{2} < f_0 < f - \dot{f}_0(t - t_0) + \frac{\delta f_H}{2}$$

FH CPU implementation (single point)



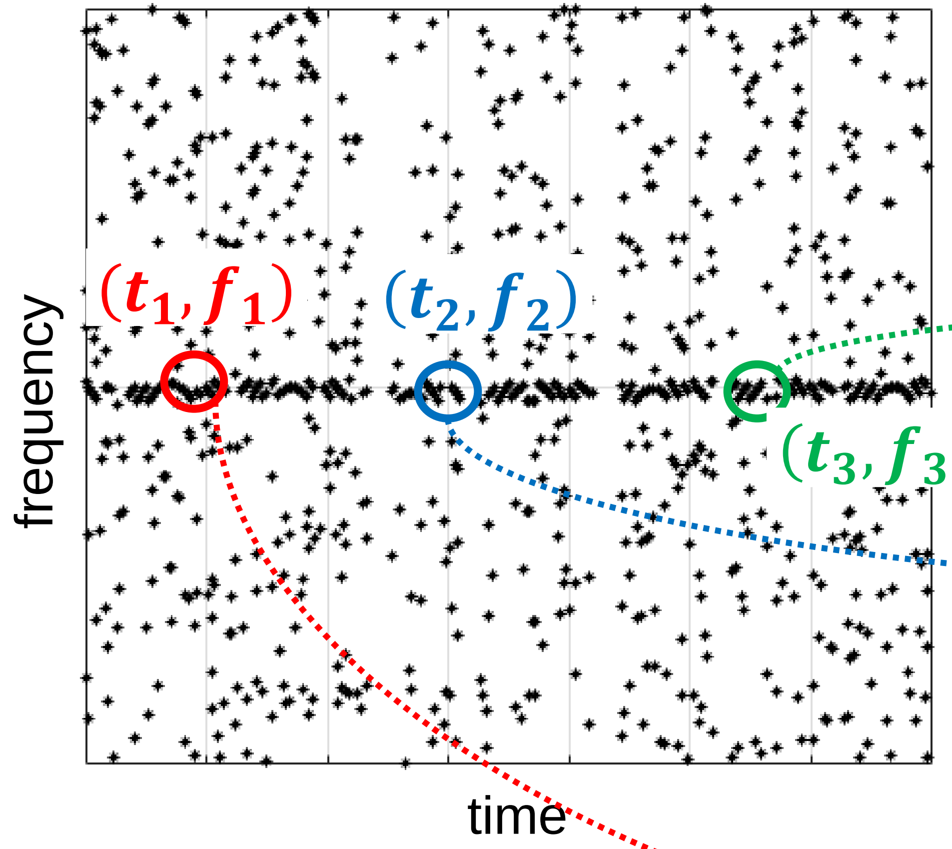
$$f_0 = f - \dot{f}_0(t - t_0)$$

- Loop over the whole spin-down grid
- For each s-d value $\dot{f}_0^*$ compute the frequency indexes in f_0 such that

$$f - \dot{f}_0^*(t - t_0) - \frac{\delta f_{\mathrm{H}}}{2} < f_0 < f - \dot{f}_0^*(t - t_0) + \frac{\delta f_{\mathrm{H}}}{2}$$

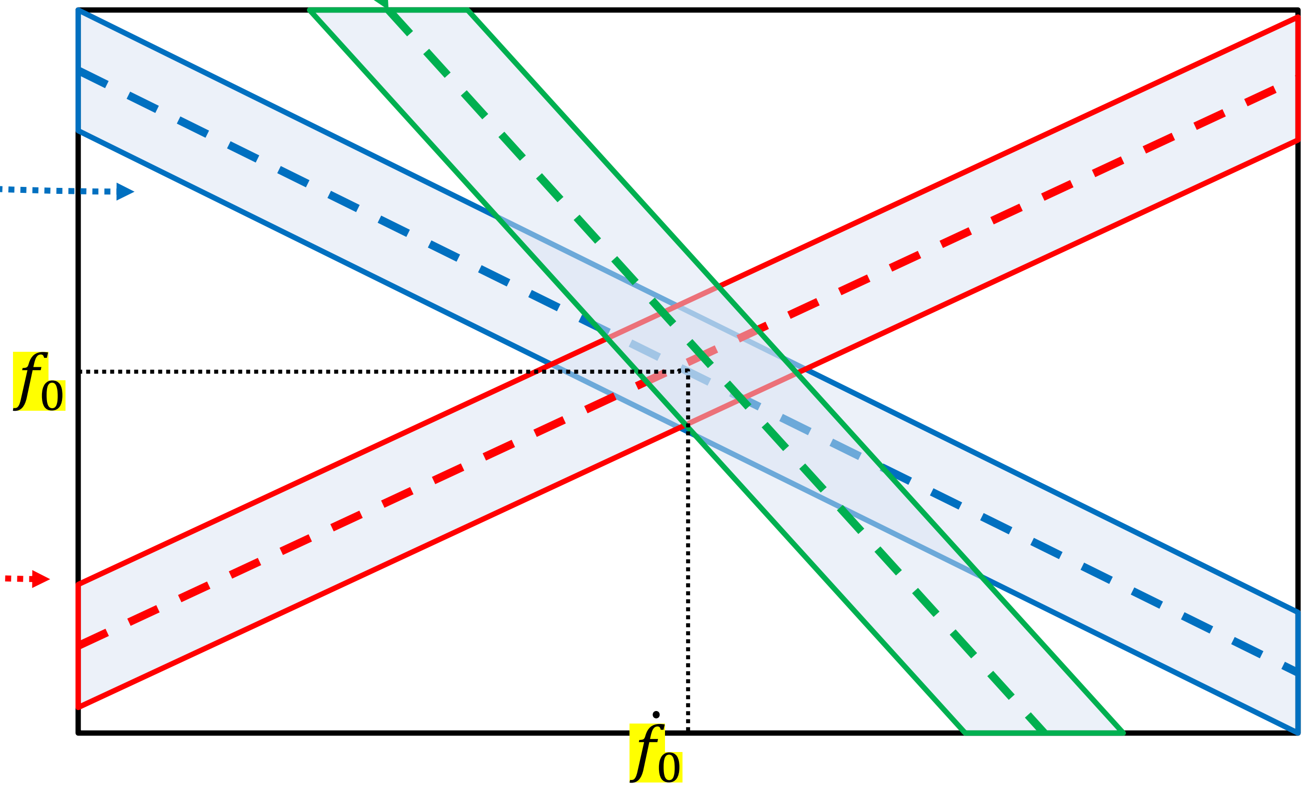
- Increase by 1 (or by the weight) the corresponding points in the Hough map
- A single (t, f) point corresponds to a uniformly-valued stripe in the Hough map

FH CPU implementation (single point)



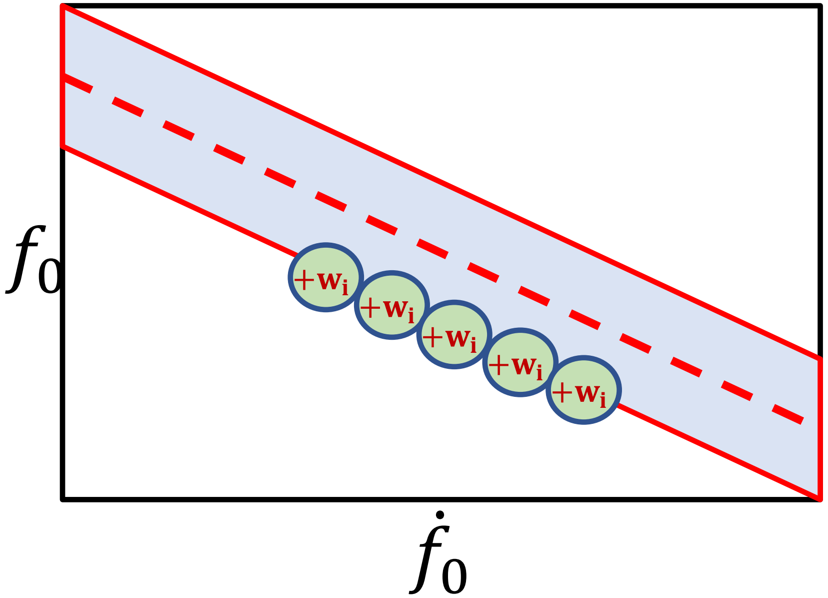
Different points from a same line produce stripes that intersect all on the ''true'' signal parameters

$(\dot{f}_0, f_0) \rightarrow f = f_0 + \dot{f}_0 (t - t_0)$



The area where the stripes intersect gets an excess of counts/weights

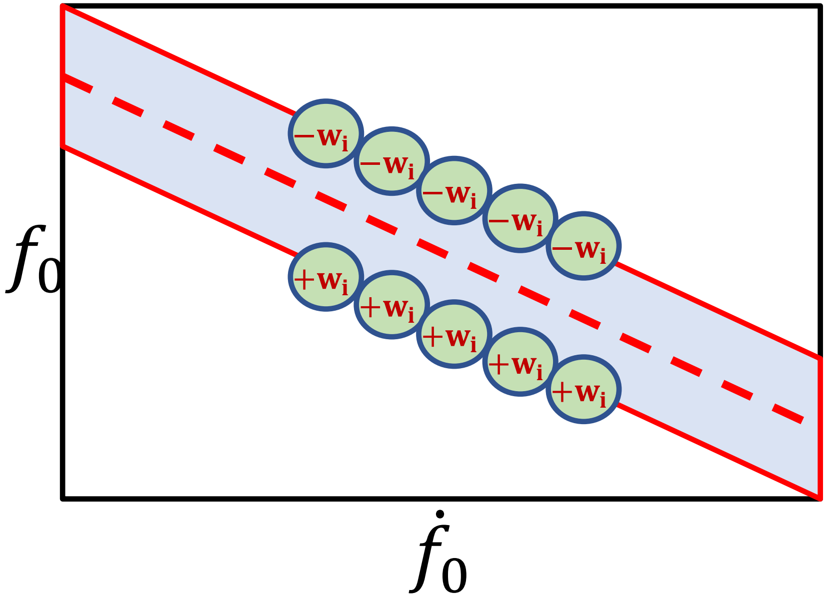
Differential FH: Carl Sabottke idea to deal with over-resolution



In the loop, only the pixels on the lower bound of the stripe are increased by the weights.

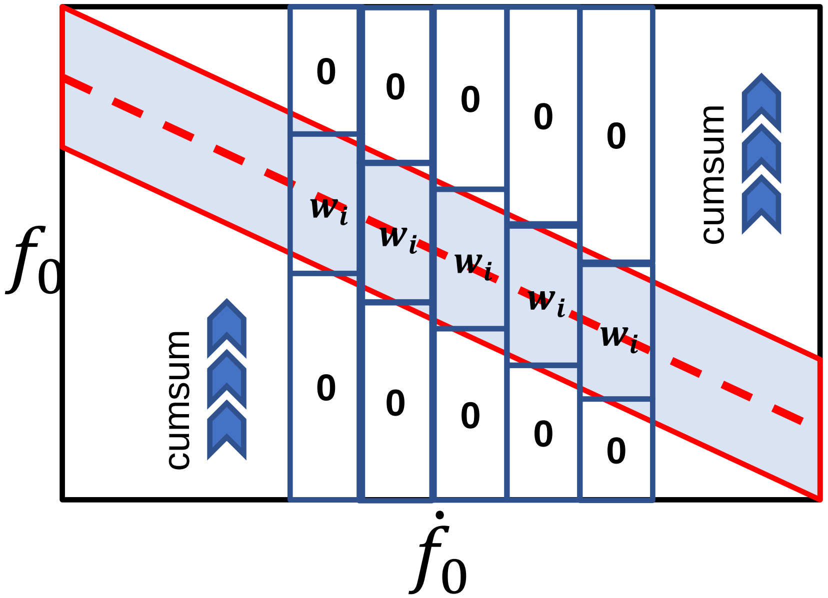
$$f_0 = f - \dot{f}_0(t - t_0) - \frac{\delta f_H}{2}$$

= half differential map



After the loop, the values at the lower bound are symmetrically subtracted at the upper bound.

= full differential map

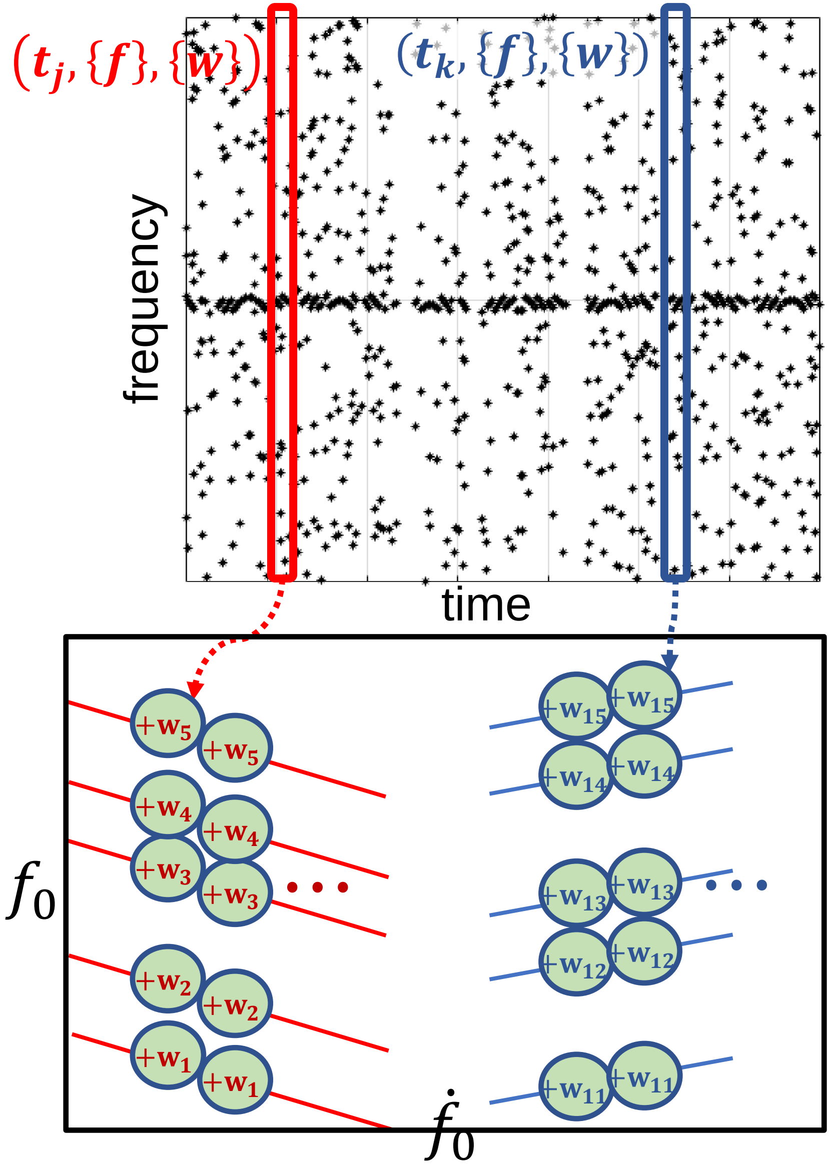


Then, the map is integrated through cumulative sum from low to high frequencies.

- The whole stripe is valued with the weights.

= Hough map

FH CPU full implementation



- ❖ Loop over different times $\{t_i\}$.

For each time t_k select the vectors of all frequencies $\{f_k\}$ and the weights $\{w_k\}$.

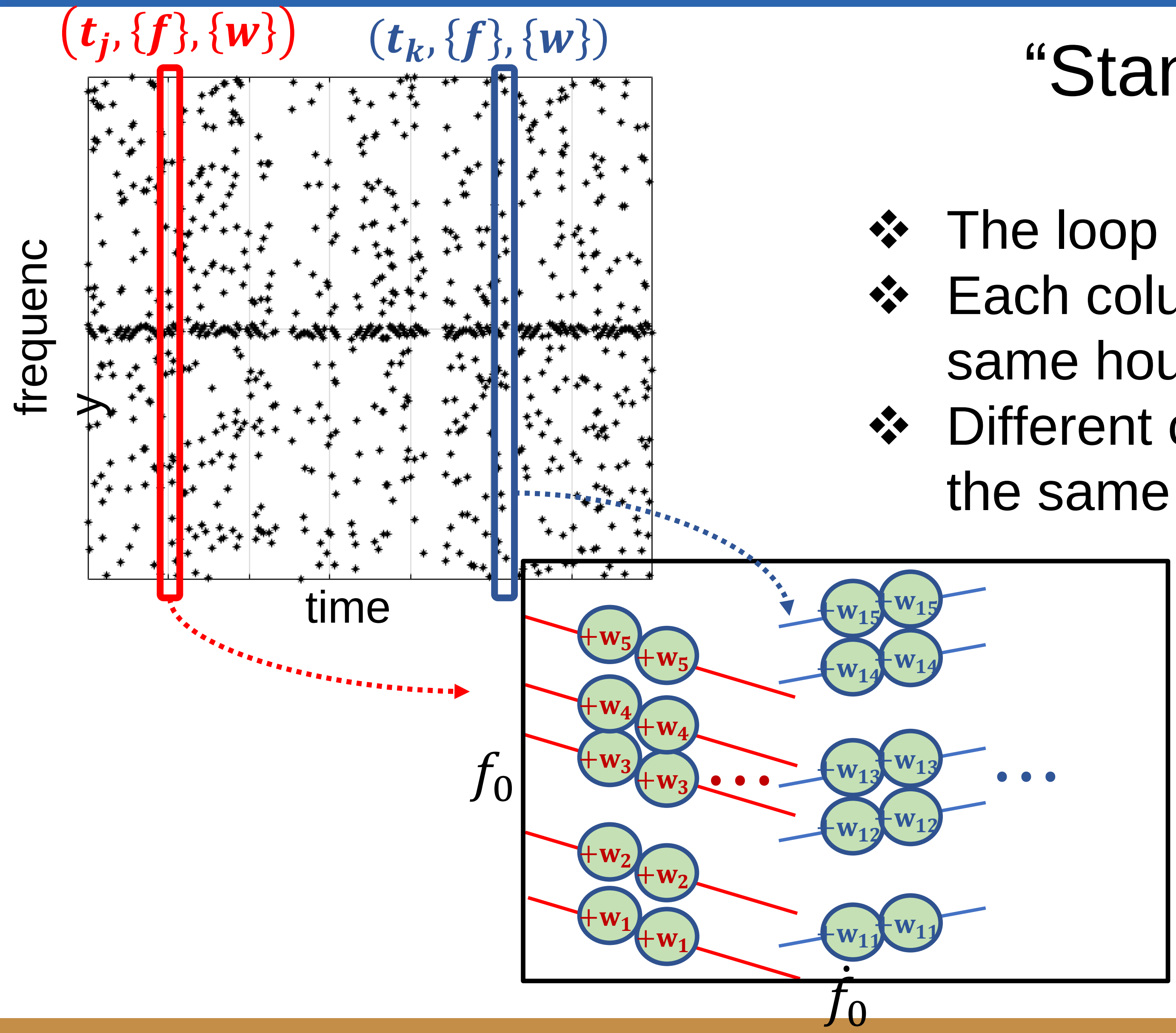
- ❖ Loop over the whole spin-down grid.
  - For each s-d value $\dot{f}_0^*$ compute the frequency indexes of the lower bounds of the stripes in f_0 such that

$$\{f_0\} = \{f\} - \dot{f}_0^*(t_k - t_0) - \frac{\delta f_{\mathrm{H}}}{2}$$

  - Increase by $\{w_k\}$ the corresponding points in the Hough map.

- ❖ After the loops completion: subtract symmetrically the values shifted by δf_{H} and then perform the cumulative sum.

"Standard" FH limitation

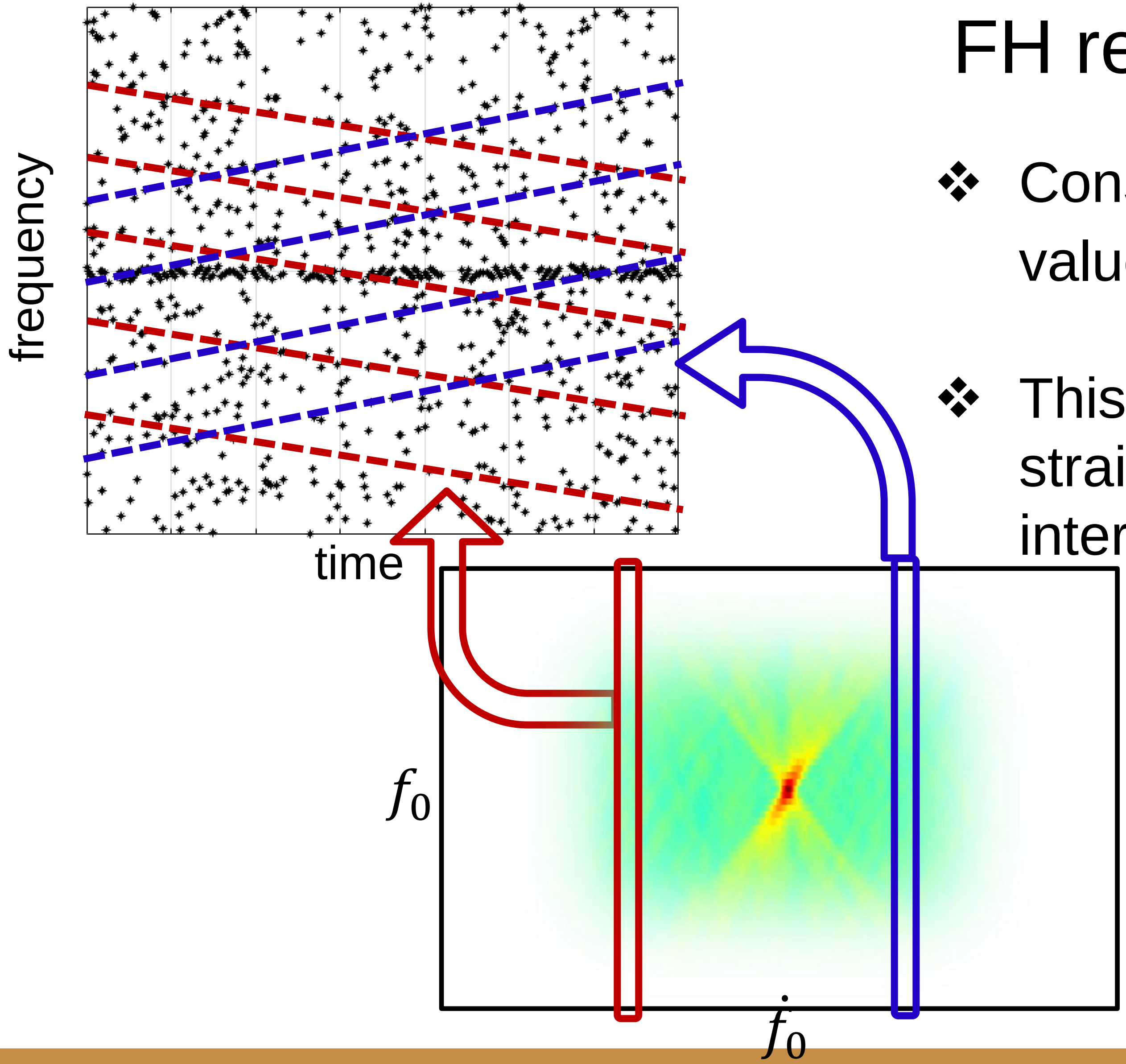


- The loop is done over different times $\{t_i\}$
- Each column $(t_k, \{f\}, \{w\})$ is transformed over the same hough map.
- Different columns are likely to write partially on the same memory locations.

➢ <u>The loop cannot be parallelized</u>!

OR: it could be parallelized at a strong memory cost (each column writes on one hough map copy, then sum all together)

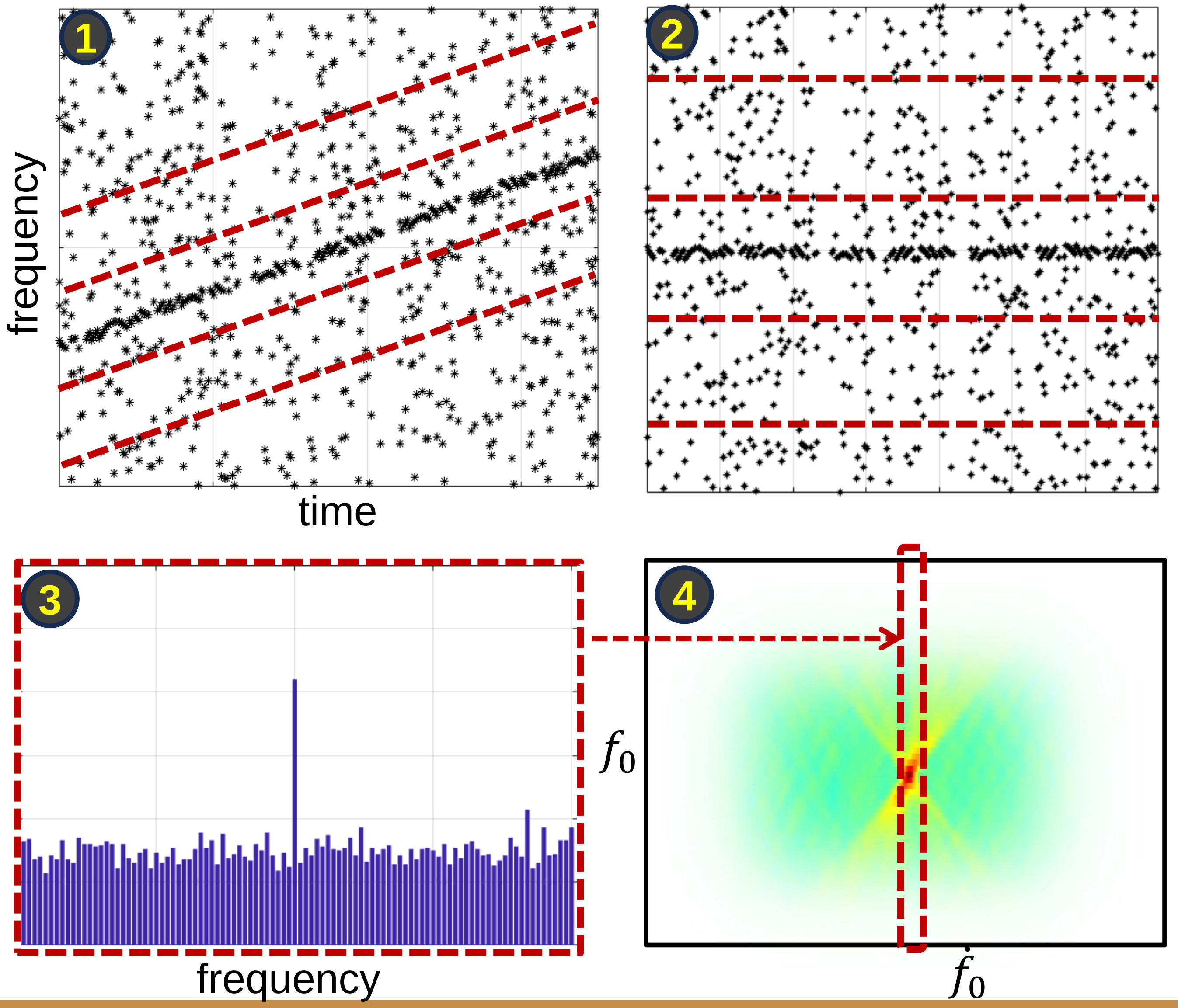
FH reformulation: loop inversion



- Consider a column in the hough map: one s-d value $\dot{f}_0$ and all the initial frequencies $\overrightarrow{f_0}$.
- This column corresponds to a set of parallel straight lines with same slope and different intercept: $\left(\dot{f}_0, \overrightarrow{f_0}\right) \rightarrow \vec{f} = \overrightarrow{f_0} + \dot{f}_0 (t - t_0)$
  - By inverting the relation, the whole peakmap can be mapped on a hough column for a given s-d value $\dot{f}_0$:

$$\overrightarrow{f_0} = \vec{f} - \dot{f}_0 (t - t_0)$$

Loop inversion implementation



1) Select a single s-d value $\dot{f}_0$: it corresponds to a given slope in the peakmap.
2) Shift the peaks depending on their time value according to $\overrightarrow{f_0} = \overrightarrow{f} - \dot{f}_0(t - t_0) - \frac{\delta f_H}{2}$
3) Compute the (weighted) histogram of the shifted peakmap.
4) Put the result in the Hough map at the $\dot{f}_0$ column.

➢ Repeat for all $\{\dot{f}_0\}$ values of the Hough map **(in parallel!)**

Thank you for your attention!