Getting started with line investigations and lines lists

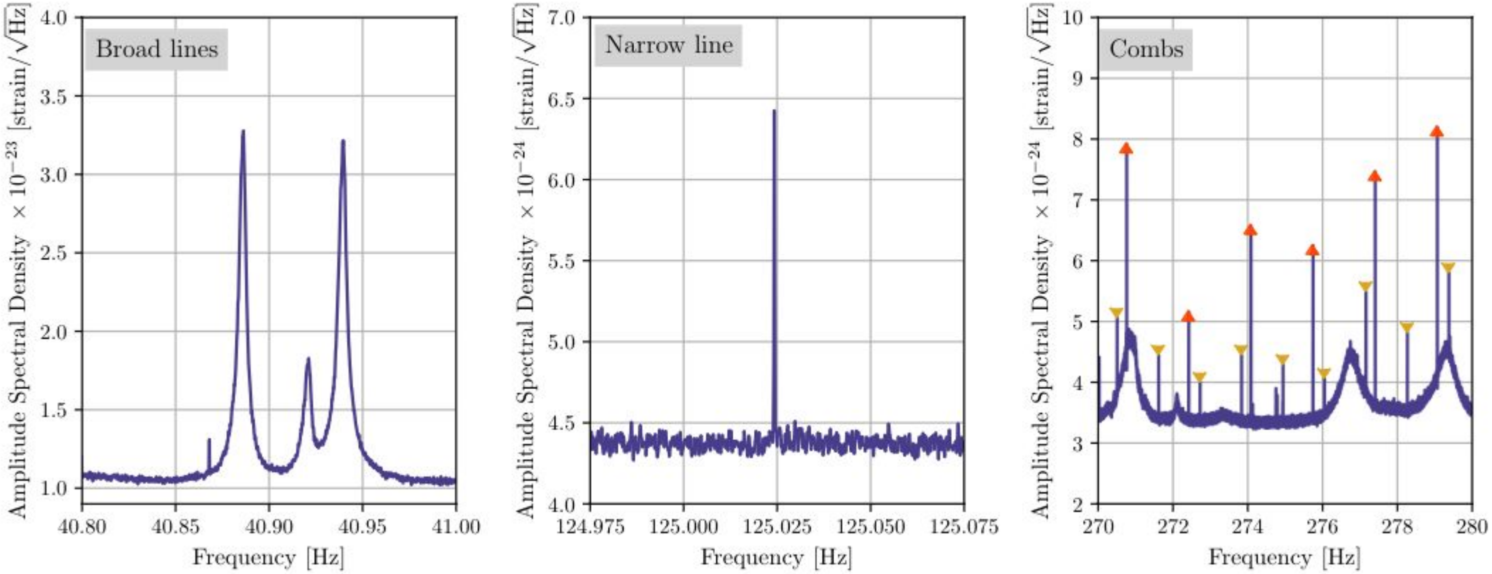
Evan Goetz
University of British Columbia

DetChar workshop, IUCAA - Dec 2025

DCC: LIGO-G2502647

Narrow spectral artifact “classes”

LIGO detector characterization in the first half of the fourth Observing run Soni, et al. 2024



Part 1 roadmap

- Continuous waves and data analyses
- Data quality products and line investigations
- Line characteristics
- Generating lines lists and starting investigations
- Basic tools
- Spectra and spectrograms
- Coherence and persistency
- Mitigation efforts
- Tools for investigations and examples

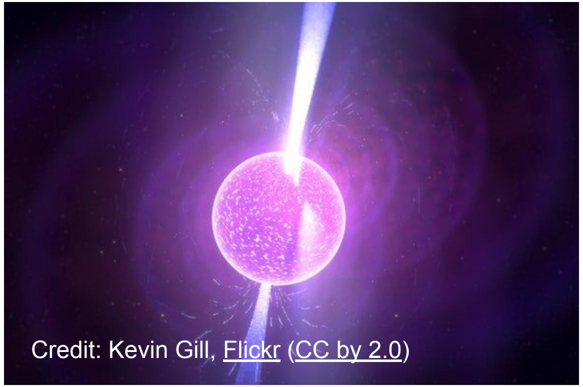
Part 2 roadmap

- Create SFTs (intermediate data products)
- Calculate spectra, coherence
- Interactive bokeh plots using Fscan code
- Run a simple Fscan workflow example
- Experiment with some of the bespoke scripts

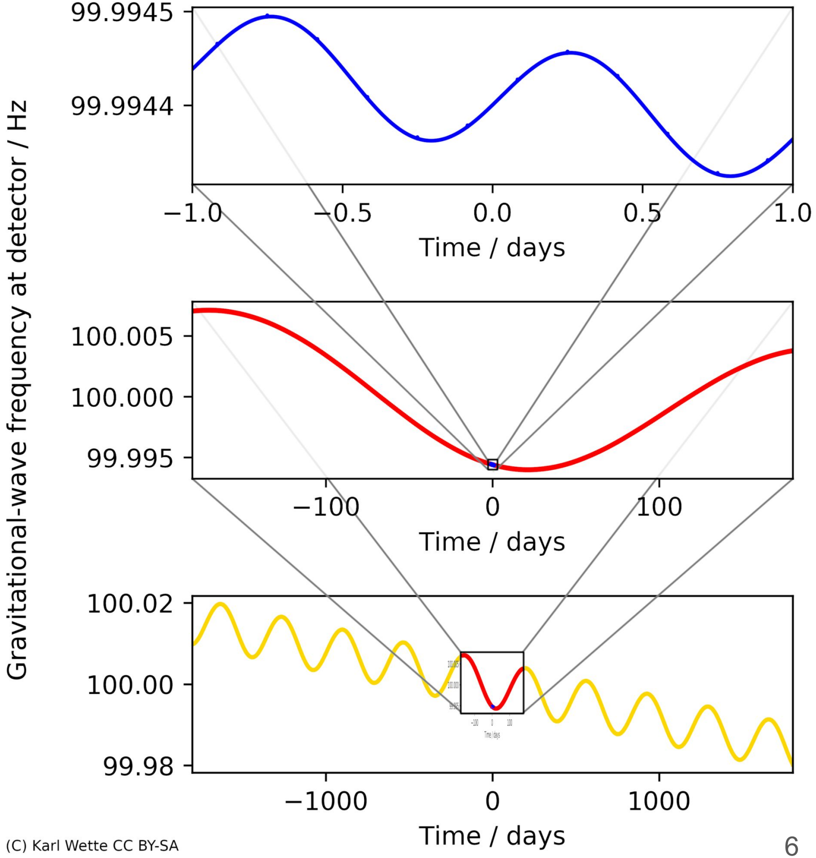
[time permitting]

Part 1: overview and scope

Continuous gravitational waves


Credit: Kevin Gill, Flickr (CC by 2.0)

- Characteristics: weak, persistent, amplitude modulated and Doppler modulated by motion of the detector ⇒ some sky locations will have minimal Doppler modulation
- Detector artifacts that have similar behaviour will obscure or degrade searches ⇒ artifact mimics a real signal



CW analyses

- Analyses most impacted by spectral artifacts are the all-sky searches (“blind”/wide-parameter space)
- O(10^{15}+) templates ⇒ many chances for narrowband artifacts to mimic a signal
- Power-based or F-statistic based searches with coherent baseline O(1) hr accumulate statistic over observing run
- “Coherent baseline” for power-based searches = Fourier transform O(1) hr data (these Fourier transforms are often called Short Fourier Transforms, SFTs)
- Useful strategy: make run-average spectra with O(1) h SFTs
  - Mimics the actual CW searches, allows us to visualize what the analyses “see”
- Longer coherence is better for signal versus noise rejection but this changes the signal model (signal phase coherence must be maintained for longer; reduced or no spin wandering)

A note on narrowband stochastic gravitational wave searches

- Complementary searches to CW analyses, but with far less restrictive signal model - could provide a detection if source does not follow signal models
- Nominally a less restrictive model would cause the analysis to be more prone to narrowband artifacts but stochastic searches rely on cross-correlating data between detectors
- As long as artifacts are not coherent between detectors, the cross-correlation analysis is less impacted by artifacts (but coherent artifacts are very bad) in part due to the coarser frequency resolution
- Stochastic group maintains a “notch list”, a list of frequency bins to exclude from analyses. Does not include everything from the lines lists

Data quality products

- CW: lines lists

- Stochastic: notch list

- Notch list may partially or closely align with lines lists
- This presentation will focus primarily on lines lists

Examples of line contamination

- Purposeful:
  - Calibration lines
- Unintended (unavoidable?):
  - Interferometer mirror suspension violin resonances, bounce/roll modes
  - Power mains harmonics (60 Hz)
- Unintended (avoidable?):
  - Calibration line intermodulation
  - DuoTone lines
  - Dither lines
  - OM2 heater driver comb (1.66 Hz comb)
  - HWS near-integer combs (~1, ~2, ~5, ~7 Hz)
  - A lot of unidentified lines…

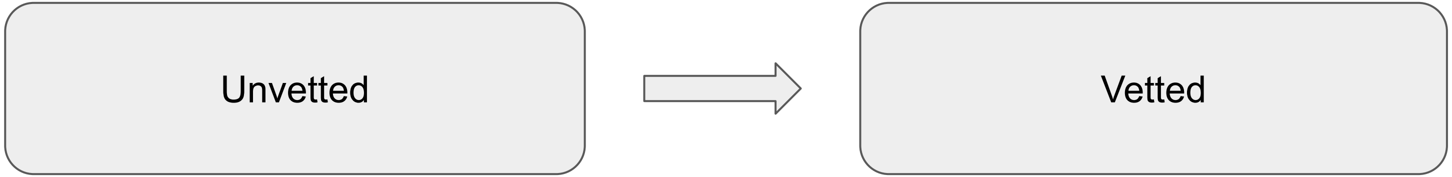
Narrow artifact characteristics

- “Ideal” behaviour: fixed amplitude, frequency
- Typical behaviour: time-varying amplitude, frequency

- Line artifacts that are single bin (to high resolution), do not wander, and have “human”-like frequencies (integer frequency, or frequency with tenths or hundredths values and no further) are likely **GPS-synchronized** or **digital**
- Line artifacts that do not exhibit these characteristics may be **analog** (some physical object moving, e.g., motor, mirror suspension resonance, etc.); might have a quartz crystal oscillator or not for frequency stabilization

Lines lists

- Run average spectra ⇒ narrow peaks in spectra
- Divide these artifacts into two categories: **unvetted and vetted lists**
- Unvetted = unknown origin (anything not vetted)
- Vetted = known non-astrophysical in origin
    - Needs an instrumental cause; evidence for some terrestrial cause
    - Amplitude of the artifact insufficient to prove non-astrophysical
    - Correlation with auxiliary channel also insufficient to prove non-astrophysical
    - Combs almost always can be immediately added to vetted list even if cause is unclear
    - Used by CW searches for vetoing analysis outliers
- Lists (usually unvetted) are used as starting points for **investigations**



What is in a lines list?

Unvetted list contents:

```
<frequency>,<label> !<tag>
```

Example:

```
15.0,Unidentified peak; suspicious frequency !1 working
```

Vetted list contents:

```
<frequency or spacing>,<type>,<frequency offset>,<first visible harmonic>,<last visible harmonic>,<left width>,<right width>,<comments>,<segments known to be present - new for O4>
```

Example:

```
4.9842518,1,0,3,133,0.0003,0.0003,HWS (ITMY); segments determined from aLOGs/Fscans; 1-day resolution,"[(1368975618, 1385856018)]"
```

Vetted list contents and implications

Frequency or frequency spacing [Hz]	Type (0:line; 1:comb; 2:comb with scaling width; 3:band with time-limits)	Frequency offset [Hz]	First visible harmonic	Last visible harmonic	Left width [Hz]	Right width [Hz]	Comments	Segments known to be present (empty or * should assume line is always present)
11.475	0	0	1	1	0.00111	0.00111	Calibration line (thermalization line DARM_EXC)	[(1371427218, 1371513618), (1374278418, 1375660818)]

Type

0 $$[frequency - leftwidth, frequency + rightwidth]$$

1 $$[harmonic * frequency + offset - leftwidth, harmonic * frequency + offset + rightwidth]$$

2 $$[(harmonic * frequency + offset) * (1 - leftwidth/(firstharmonic * frequency + offset)), (harmonic * frequency + offset) * (1 + rightwidth/(firstharmonic * frequency + offset))]$$

3 $$[frequency - leftwidth, frequency + rightwidth]$$

Starting the lines list

- To generate spectra:
  - Need to know the segments of data used for analyses, typically ANALYSIS_READY - CAT1 vetoes
  - *h(t)* frame data, typically need **gated** frame data
- DetChar group provides CAT1 vetoes and veto definer files
- If using gated data, need to have frame cache file
- Using the previous observing run lines list may be a place to start

Understanding data from an analysis perspective

- Require a way to characterize the data as an analysis would “see”
- Create a run-average spectra where each entry into the average is weighted based on the noise in the neighbouring frequency bins (lalpulsar_spec_avg_long)

$$\langle P(f_n)\rangle = \frac{2}{T_{\mathrm{SFT}}}\frac{1}{M}\sum_{m=0}^{M-1}\frac{\sum_{n=0}^{N-1} w_m(f_n)P_m(f_n)}{\sum_{n=0}^{N-1} w_m(f_n)} \qquad w_m(f_n) = \frac{1}{N+1}\sum_{i=-N/2}^{N/2}\frac{1}{P_m(f_{n+i})}$$

$i = \text{window frequency bin index}$

$m = \text{SFT index}$

$n = \text{frequency bin index}$

- This effectively deweights noisy SFTs for a given frequency bin
- Note that this can be impacted by loud lines since the weight for a frequency bin is the mean of inverse of the surrounding noise power

Checklist

- Segments for analyses
- (Gated) $h(t)$ data
- Run-average spectra

Line catalogues (data quality product)

Steps:

1. Identify peaks in spectrum
2. Peak vetting
3. Estimate frequency band impacted by each vetted peak
4. Vetted peaks go in vetted list; everything else in unvetted list

- Most work done “by hand” - difficult to automate but welcome efforts!
- **Partial automation**: peak identification (still requires human verification/modifications)
- **Manual**: peak vetting, band estimation, list creation

Identifying peaks in spectra

- Historically done by eye, but tedious and prone to error
- Ansel Neunzert has explored using scipy functionality (`scipy.signal.peak_prominances`) to identify peaks in spectrum
- Pretty good for first pass, but still may make mistakes
- Require people to check that there were no missed peaks or that the peaks are correctly found
- Use Fscan interface (`fscan.process.linefinder`) fo find peaks

- **Important: we are primarily concerned about narrow lines, which is a subjective term; CW is usually concerned with lines $\lesssim \text{few mHz}$ in width, but loud broad lines (violin resonances, power mains, etc.) are also problematic**

Checklist

- Segments for analyses
- (Gated) *h(t)* data
- Run-average spectra
- Peaks in spectrum + review

Now we start vetting lines

- Reminders:
  - Needs an instrumental cause; evidence for some terrestrial cause
  - Amplitude of the artifact insufficient to prove non-astrophysical
  - Correlation with auxiliary channel also insufficient to prove non-astrophysical
  - Combs almost always can be immediately added to vetted list even if cause is unclear

Identifying combs

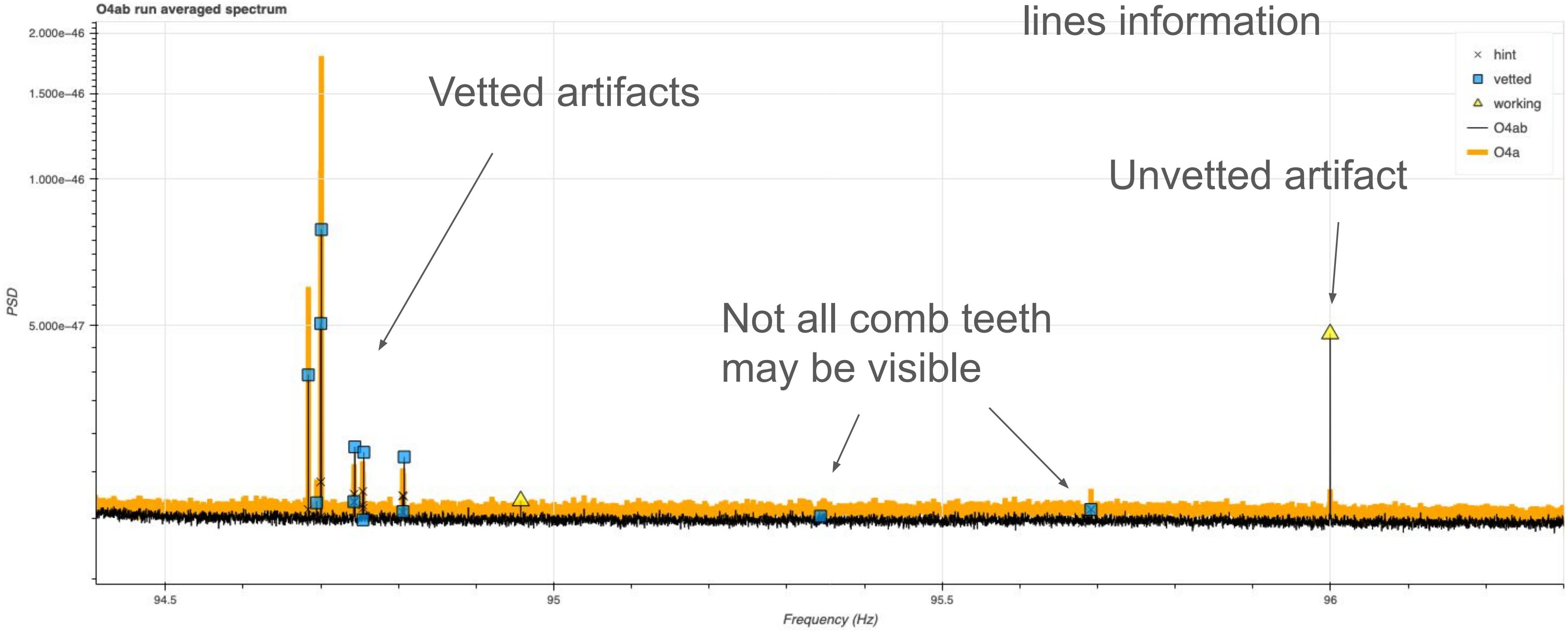
- Fscan contains functionality to search for patterns in the list of peaks (`fscan.process.combfinder`)
- As with peak finder method, this is imperfect but is a good place to start when searching for combs

Checklist

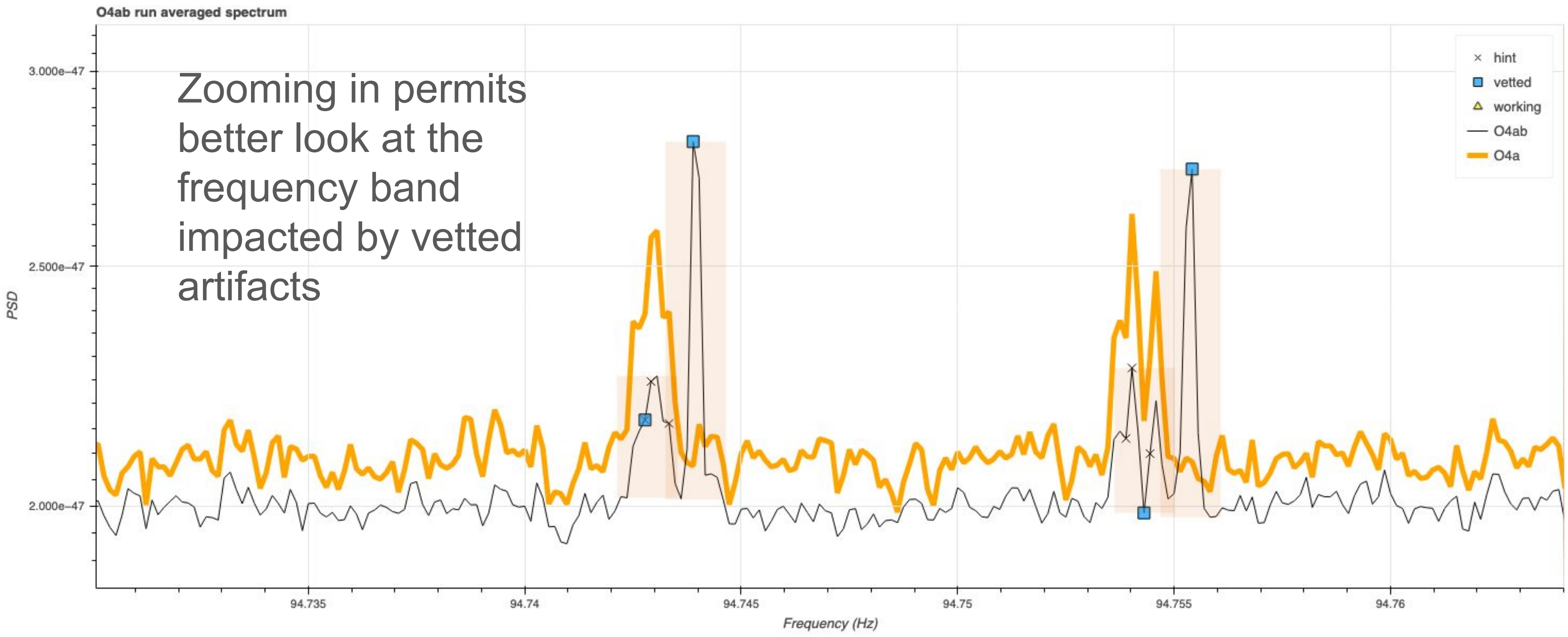
- Segments for analyses
- (Gated) *h(t)* data
- Run-average spectra
- Peaks in spectrum + review
- Initial list from previous run + early checks for combs

Example spectra during vetting

We make this spectra by providing options to the plotter with the peaks and lines information



Example spectra during vetting



Line investigations

- Likely many unvetted lines remain so we need to do more investigations
- Objectives:
  - Identify cause of noise or the coupling pathway and attempt to mitigate (remove the artifacts from the data)
  - Identify the cause of noise in order to put artifacts into the “vetted” category (analyses must remove offending data ahead of time, or veto outliers after analyses are carried out)
- First objective is the best one as it makes the data better overall, but if it cannot be done, then we fallback to vetoes

- “Where do I start?”

First checks of specific line

- Do an aLOG search of the line frequency to see if anyone has noticed this before
- Check previous lines lists

- These may provide clues to the origin of artifact
- If not we may have to dig deeper to determine origin of lines

Basic tools

- Power spectral density
- Spectrogram
- Coherence
- Persistency figure of merit

Terminology

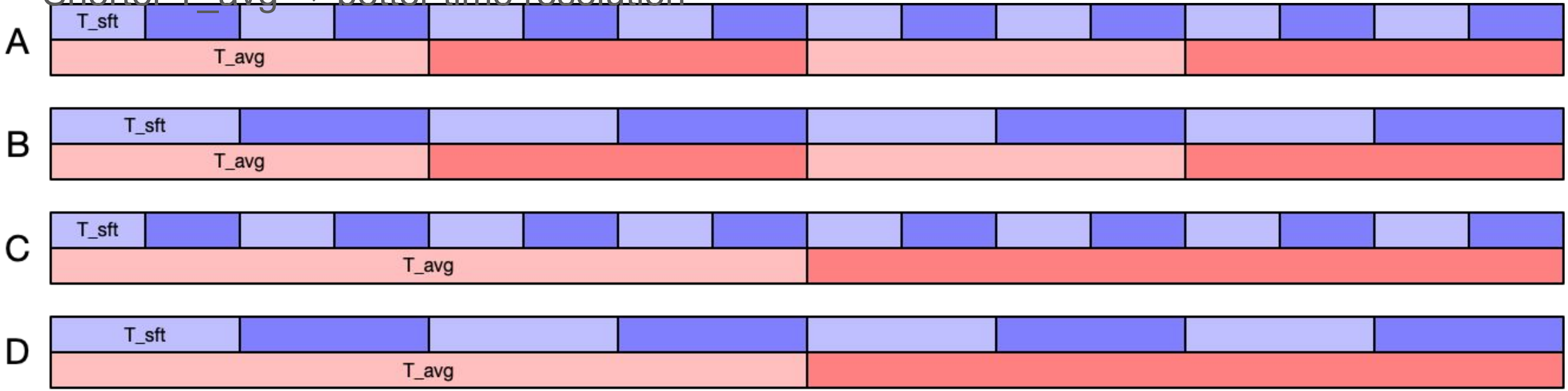
- Short Fourier Transform (**SFT**): synonym for Fourier transform; a specific file format used by CW analyses
- **T_sft**: time duration of an SFT - usually 1800 s or 7200 s, but can be customized
- **T_avg**: time duration to compute the mean value of each frequency bin in SFTs. T_avg >= T_sft. Usually 1 day, 1 week, or 1 month, but can be customized
- SFT windowing and overlap: typically use **Hann windowing and 50% overlap**. Other overlaps are supported; only rectangular, Hann, and Tukey windows are supported by the SFT file specification
    - Hann window is a good, standard, multipurpose window function

Timescales for PSDs

Longer T_sft ⇒ higher frequency resolution, increased peak height relative to background

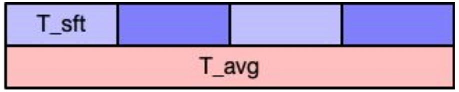
More SFTs per average ⇒ background noise variance decreases

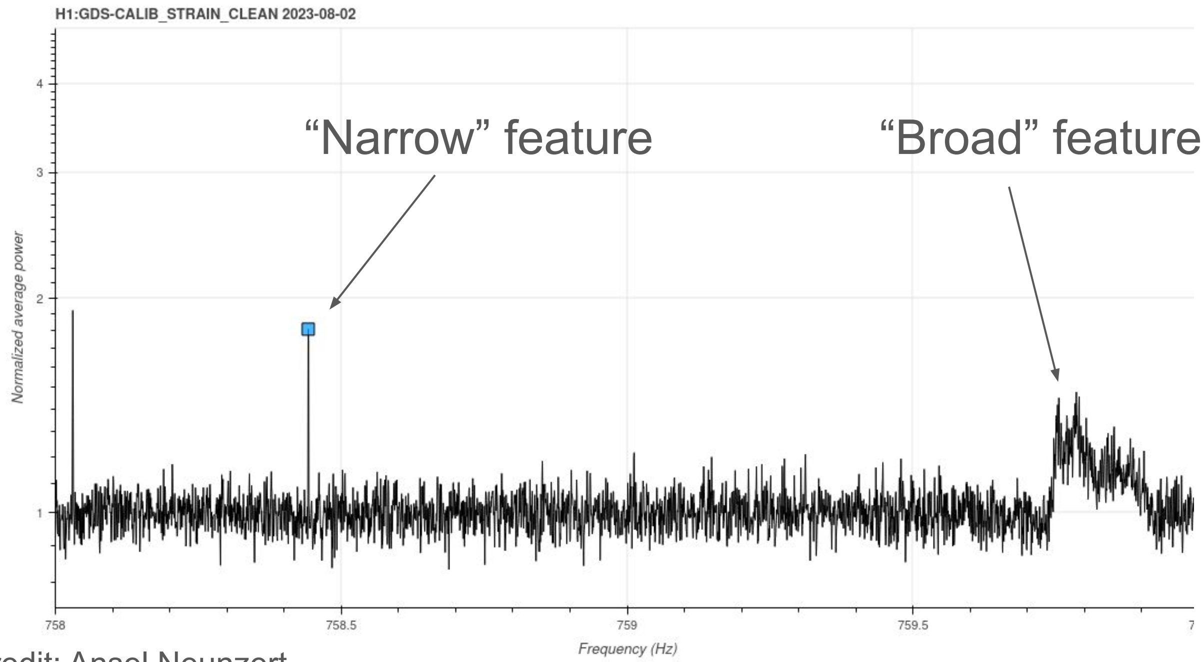
Shorter T_avg ⇒ better time resolution



Credit: Ansel Neunzert

Spectra and spectrograms



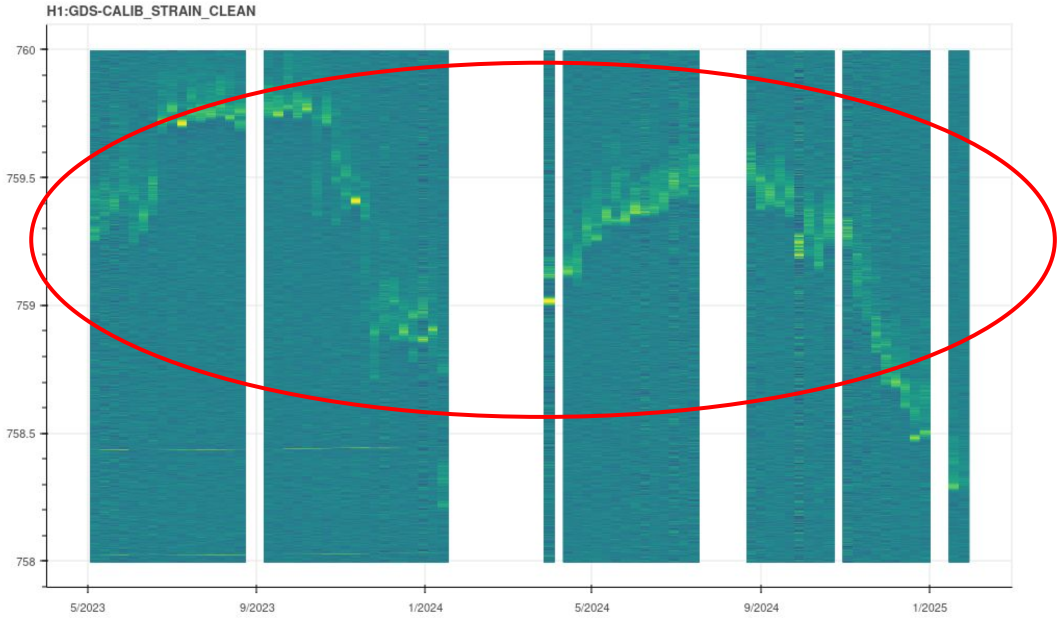


Credit: Ansel Neunzert

One averaged spectrum

- h(t) data
- 1800 s SFTs
- Low noise data only
- 1 week average
- Spectrum is "normalized" with a running median

Spectra and spectrograms

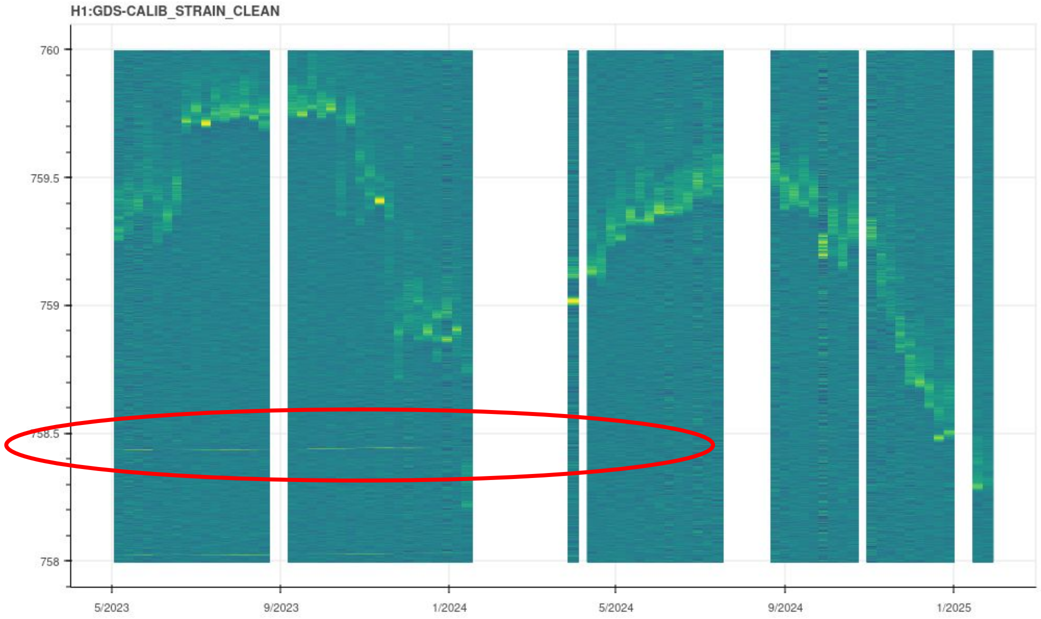


Looking at many spectra together in a spectrogram, we can see that the “broad” feature wanders in frequency over the timescale of many months

https://alog.ligo-wa.caltech.edu/aLOG/index.php?callRep=82632

Credit: Ansel Neunzert

Spectra and spectrograms



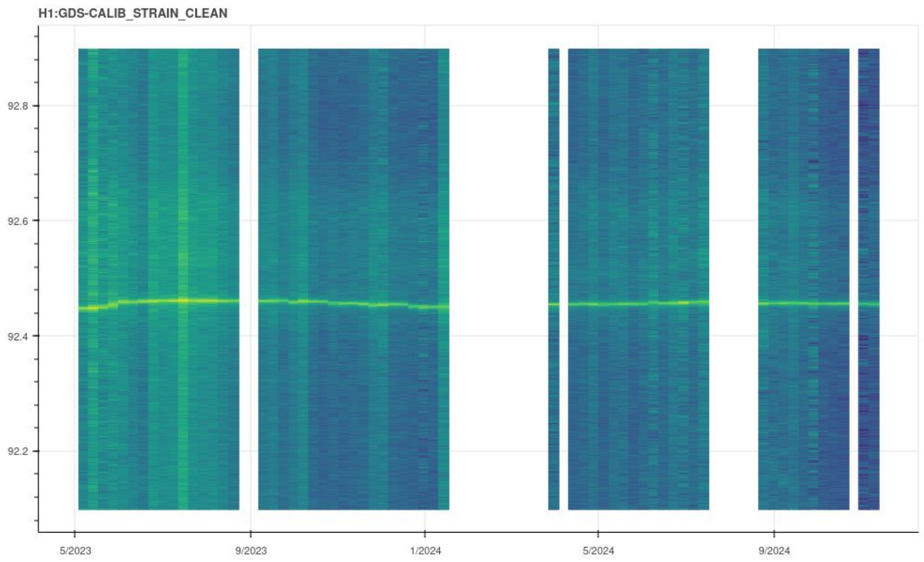
https://alog.ligo-wa.caltech.edu/aLOG/index.php?callRep=82632

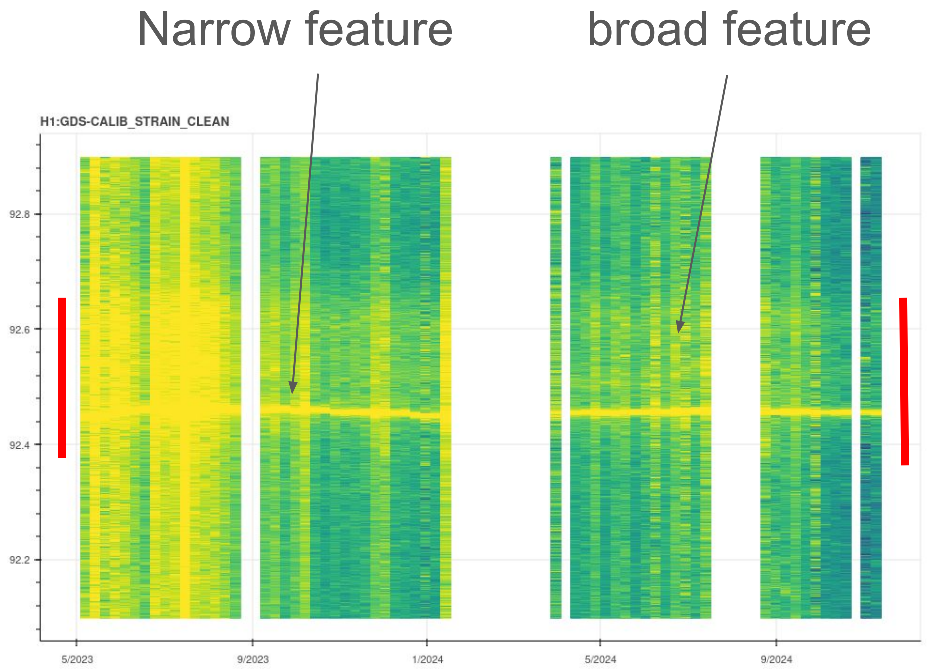
The "narrow" feature moved less and eventually disappeared

Note: hard to see narrow features on spectrograms with this y-axis scale

Credit: Ansel Neunzert

Spectra and spectrograms





Color bar adjustments may be necessary depending on the data and what you are interested in

Credit: Ansel Neunzert

Spectra and spectrograms

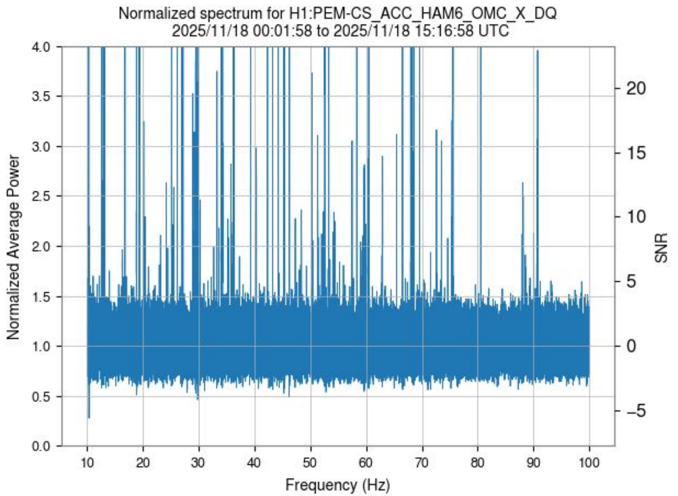
Spectrum	Spectrogram
✅ **Better for making peak height comparisons**	Color bar not ideal for peak height comparisons
✅ **Better for seeing fine details alongside large artifacts**	Color bar not ideal for covering many orders of magnitude in peak height
✅ **Can make a full-run spectrum**	Requires multiple T_avg time periods to be meaningful
✅ **Can make interactive plots over a larger frequency span**	Interactive plots over a large frequency span are slow
Can't see time evolution	✅ **Can see time evolution**

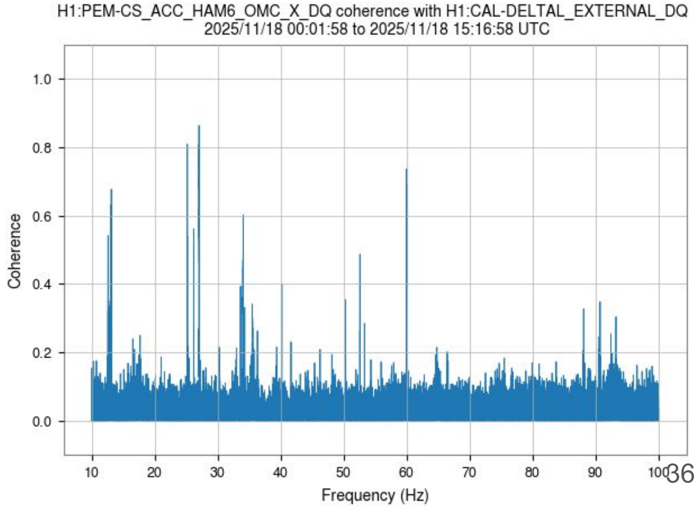
Credit: Ansel Neunzert

Coherence

$$C_{xy}(f) = \frac{|G_{xy}(f)|^2}{G_{xx}(f)G_{yy}(f)}$$

- High resolution, long averaging coherence with auxiliary channels
  - Similar to Brute force coherence (BruCo), but works over longer time scales
- Coherence is more effective at measuring a direct relationship between two channels than correlation
  - Correlation can happen by random chance, and may happen often in noisy channels
- Usually a cause for concern when coherence measured between h(t) and an auxiliary channel yields values close to 1



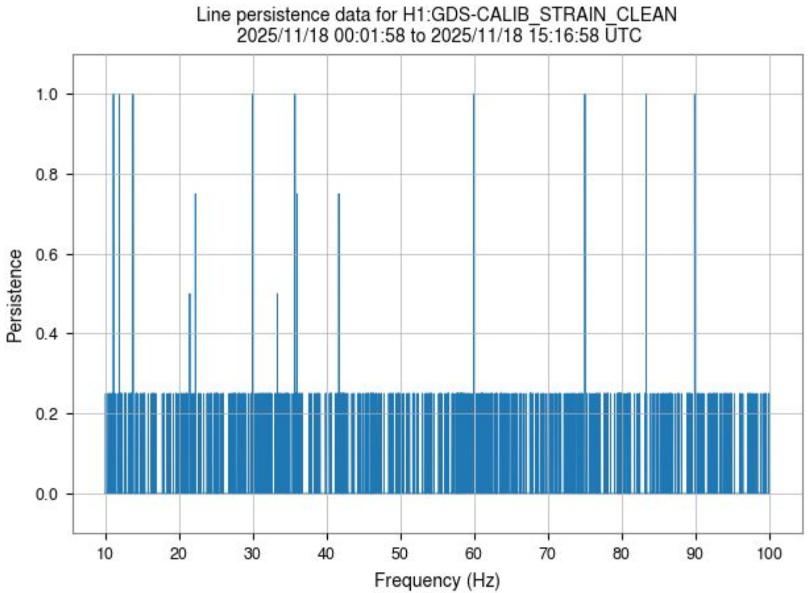


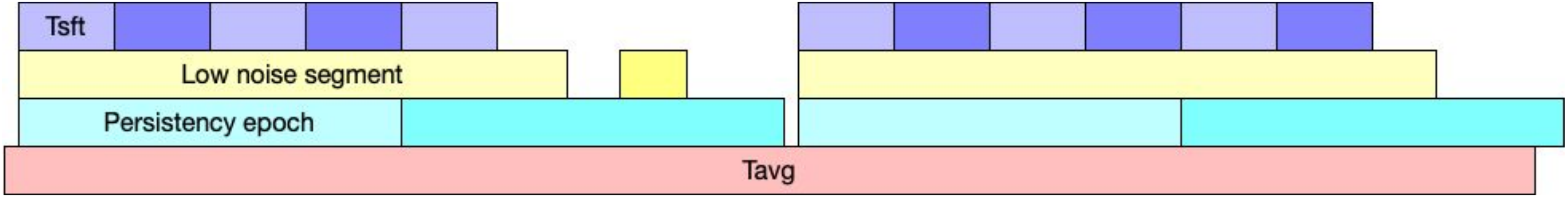
Persistency figure of merit

Measurement of how frequent over an interval of time a frequency bin is sticking out above the noise

$$Z(f) = \frac{P(f) - \mu(f)}{\sigma(f)}$$

$$\mathcal{P}(f) = \frac{1}{N_e} \sum_{n=0}^{N_e - 1} 1(Z_n(f) \geq Z_{\text{thresh}})$$





Coherence and persistency

Coherence	Persistency
Evidence might be degraded if coupling changes	✅ **Can track information about line over long term**
✅ **Effective at finding possible noise influencing *h(t)***	Does not inform about an external noise source
Caution advised for *unsafe* channels	✅ **May be used for any channel**
Lots of auxiliary channels and none may witness the noise influencing *h(t)*	Lots of auxiliary channels and none may witness the noise influencing *h(t)*
More investigations needed for threshold value that coherence is “bad”	For unaided persistency metric, need to compare against a background
May be challenging to track coherence if line frequency is time-varying	May be challenging to track persistency if line frequency is time-varying

Mitigation efforts examples

- O1-O2 - 1 Hz comb - blinking LEDs in electronics (greatly reduced)
  - Magnetometers
- O1-O2 - 8/16 Hz comb - DAC issue (greatly reduced)
  - Correlation with changing OMC dither lines
- O1-O2 - 11.11 Hz comb - vacuum electronics (mitigated but has returned)
  - Magnetometers
- O1-O2 - near 2 Hz with offset - CPS timing fanout electronics (mitigated)
  - Magnetometers
- O2 - 0.5, 2.24 Hz - illuminator remote control electronics (mitigated)
  - Magnetometers
- O2 - 1 Hz with offset - digital camera ethernet adapter (mitigated)
  - Correlation with configuration change
- O2 - 86 Hz, 119.9 Hz lines - Pcal DAC artifacts (mitigated)
  - Correlation with configuration change
- O4 - 1.11, 1.66 Hz comb OM2 heater driver electronics (mitigated)
  - Correlation with configuration changes, magnetometers
- O4 - 9.5 Hz comb - PSL flow meter sensor electronics (mitigated)
  - Magnetometers
- O4 - near 1, 5, 7 Hz combs - HWS shutter electronics (mitigated)
  - Correlation with configuration changes
- O4 - near 30, 100 Hz combs with various offsets - under investigation, maybe PSL electronics (not yet mitigated)
  - Magnetometers
- O4 - near 750 Hz lines - aliasing artifacts (greatly reduced)
  - Fixed unintentionally :)

Tools that we use

- **SFTs, spectra, spectrograms, etc.**
- **Fscan**
  - Production daily, weekly, monthly jobs on summary pages
  - Custom Fscan can be run via terminal on LDG clusters
- **Line investigation scripts**
  - Often utilize Fscan data products
  - Typically more bespoke / specialized investigations
  - Code not always mature enough for inclusion in Fscan (though popular/useful codes should!)
  - People can reuse code, contribute/improve/add their own codes
- Carleton line tool
- STAMP-PEM

* Bold items are highlighted in this tutorial

Line investigation scripts

- When-where
- Clustering
- Line prioritization

- Other personal scripts for doing specific investigation or tasks
    - Example: I have scripts that do comb finding with more knobs by using the Fscan library and another script that does some vetted lines statistics

Investigation example - unvetted artifact

- A peak is observed in the run-averaged spectrum

- Review history in aLOG and lines lists
- Check if visible on Fscan production pages
- Use line investigation script (whenwhere) to see if line behaviour changes over time - did it get louder or quieter at any specific time?
- If yes, is there anything in the aLOGs or do commissioners know if something changed that day? Was it a Tuesday maintenance time?
- If not, or uncertain, may need to carry out custom Fscans

Investigations example - commissioner request

- Commissioners change something, ask for feedback from DetChar

- Need to collect data, 1 day, 1 week, etc.
- Use Fscan line counter to identify significant changes
- Spectral ratio plots (FineToothPlot)
- Can be harder to spot smaller feature changes, especially across a full band
- Other detector changes can make it hard to interpret spectral changes

Investigations example - DetChar/commissioner observe something

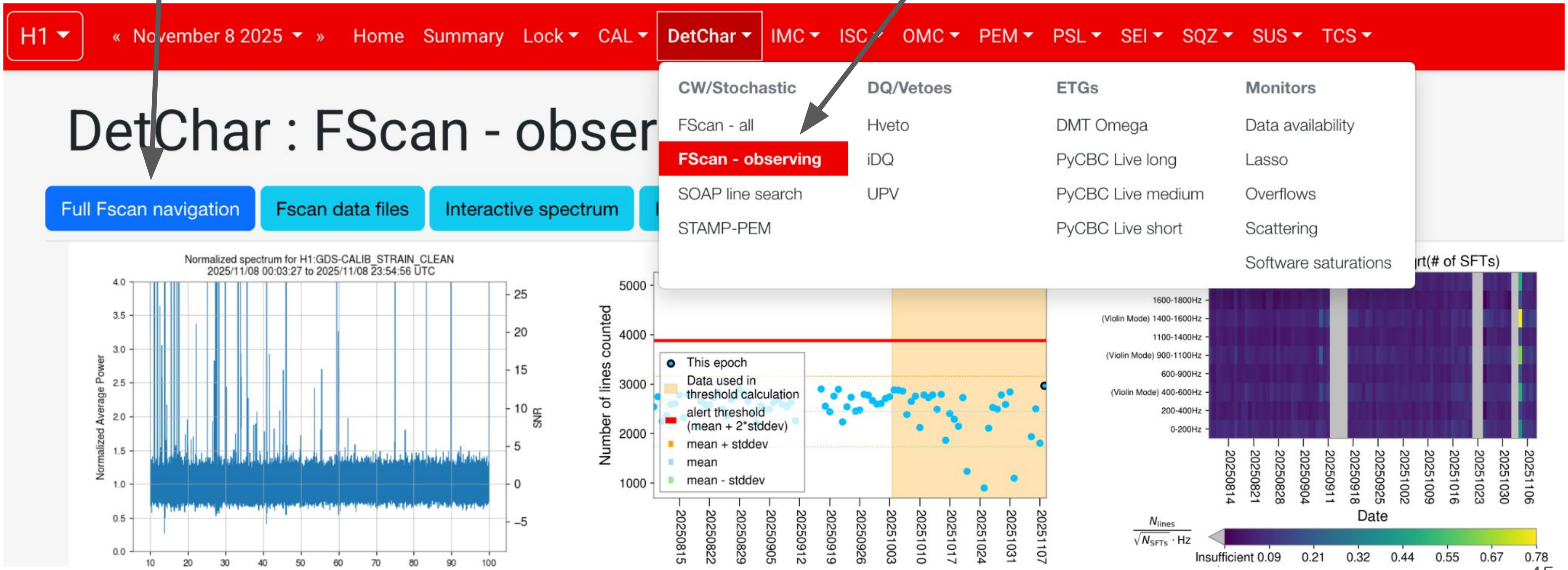
- A comb has been spotted in daily, weekly, and/or monthly *h(t)* Fscans
- Commissioners report a new narrow line found in *h(t)* spectra

- When did it appear? Is it observable in any auxiliary channels? Coherence with auxiliary channels?
- Using line investigation script (whenwhere) we can track the comb / line behaviour over time
- Can we link it to configuration changes listed in aLOGs?

Fscan summary pages - “production Fscan”

Full Fscan output

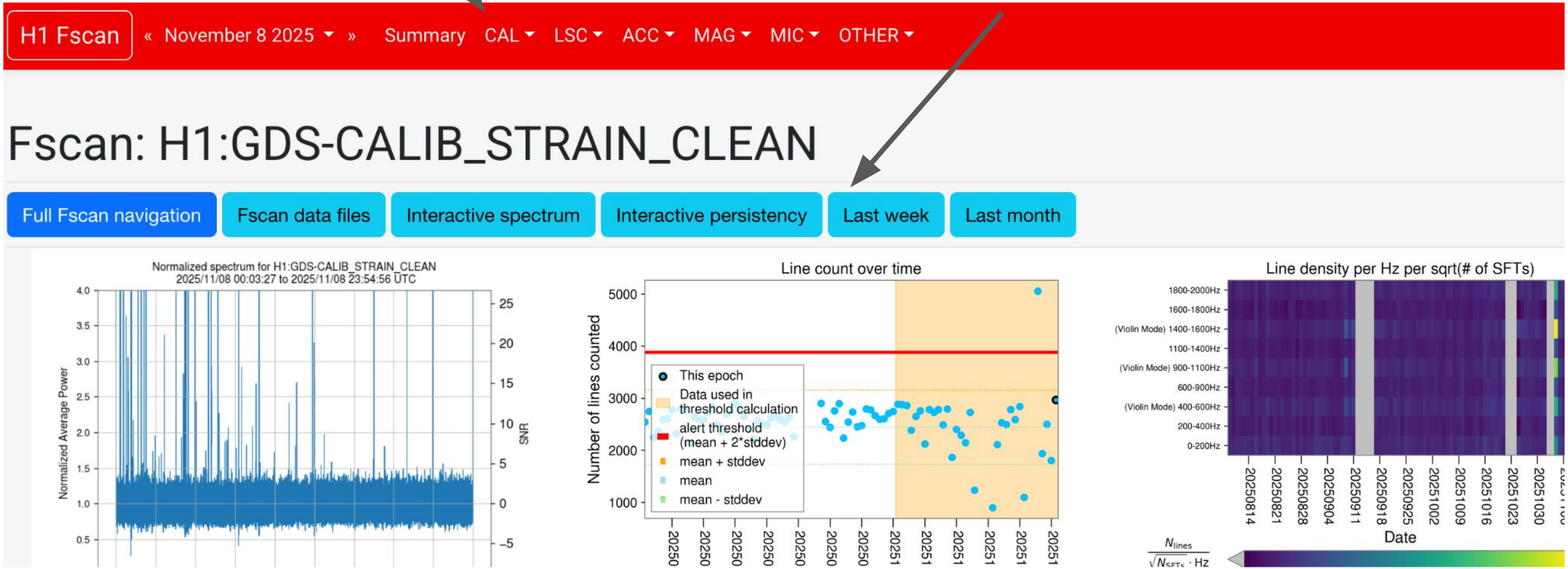
Dropdown to access Fscan page



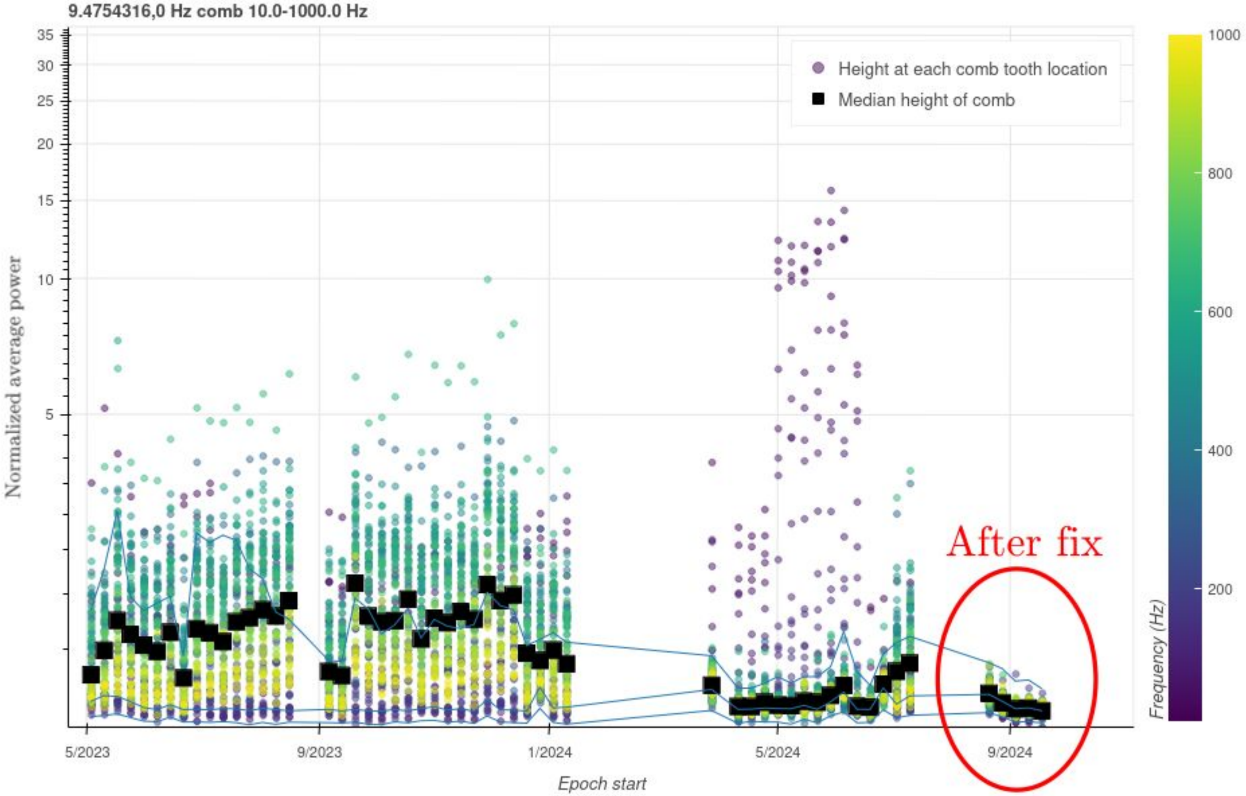
Fscan summary pages - “production Fscan”

Subsystems > channels

Buttons to access more information about this channel



Tracking combs - whenwhere.py

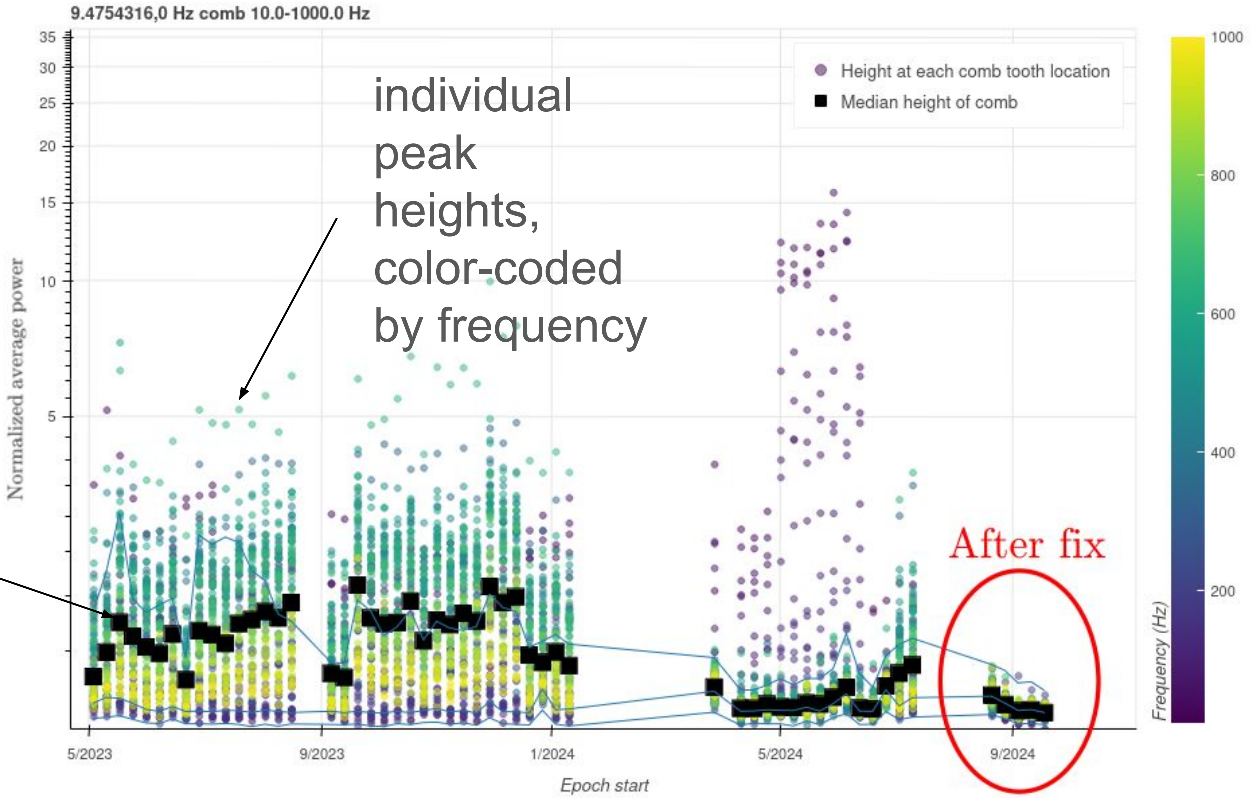


https://alog.ligo-wa.caltech.edu/aLOG/index.php?callRep=79822

Credit: Ansel Neunzert

Tracking combs - whenwhere.py

Median height of all comb peaks

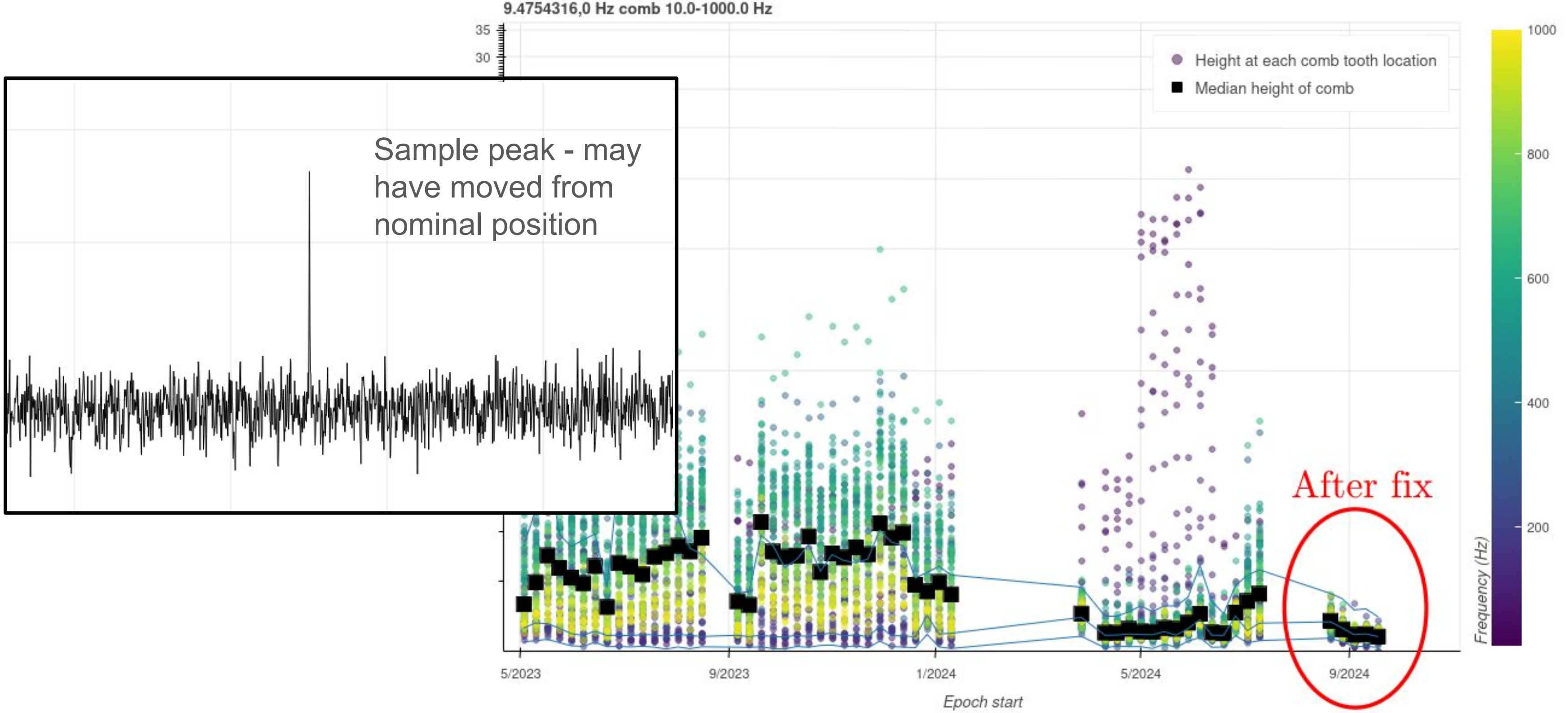


individual peak heights, color-coded by frequency

Credit: Ansel Neunzert

https://alog.ligo-wa.caltech.edu/aLOG/index.php?callRep=79822

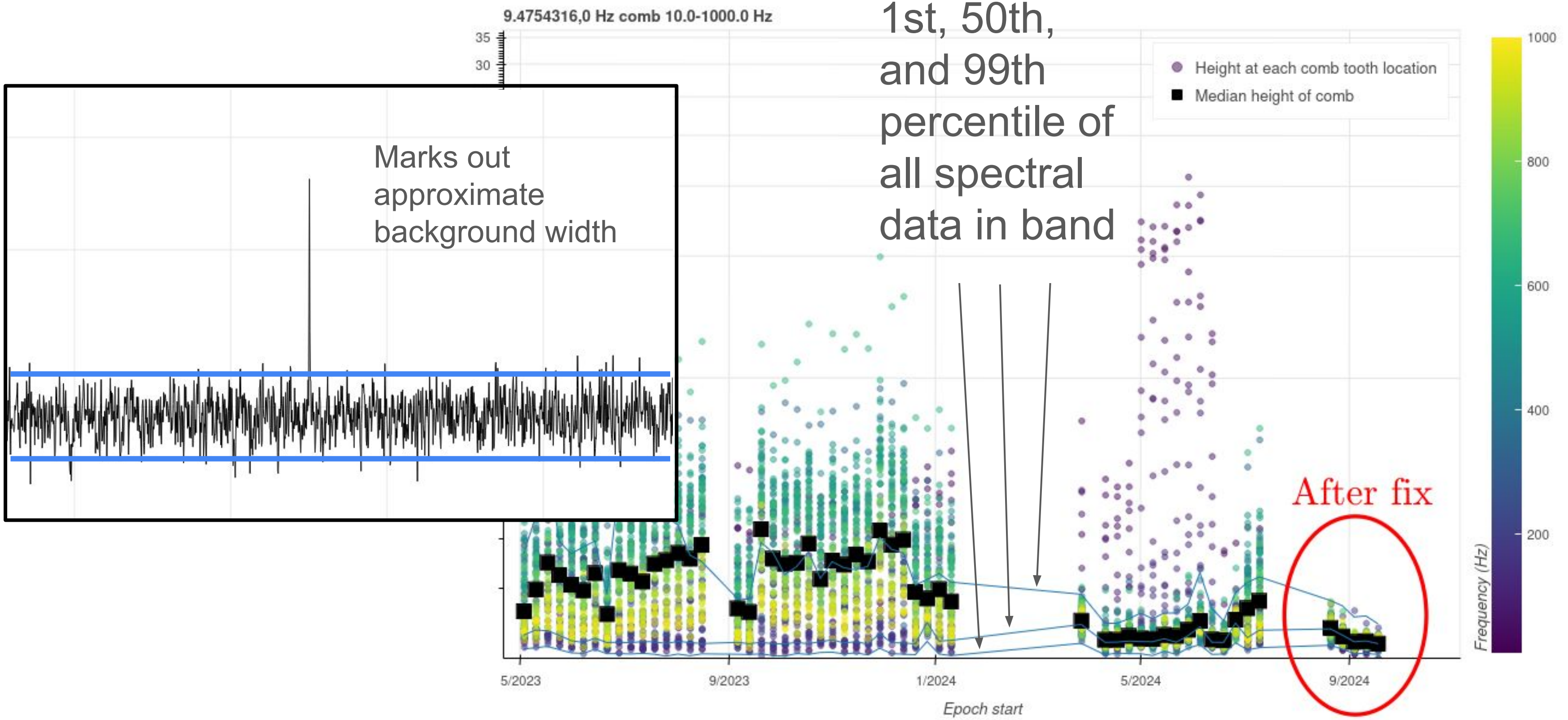
Tracking combs - whenwhere.py



Credit: Ansel Neunzert

https://alog.ligo-wa.caltech.edu/aLOG/index.php?callRep=79822

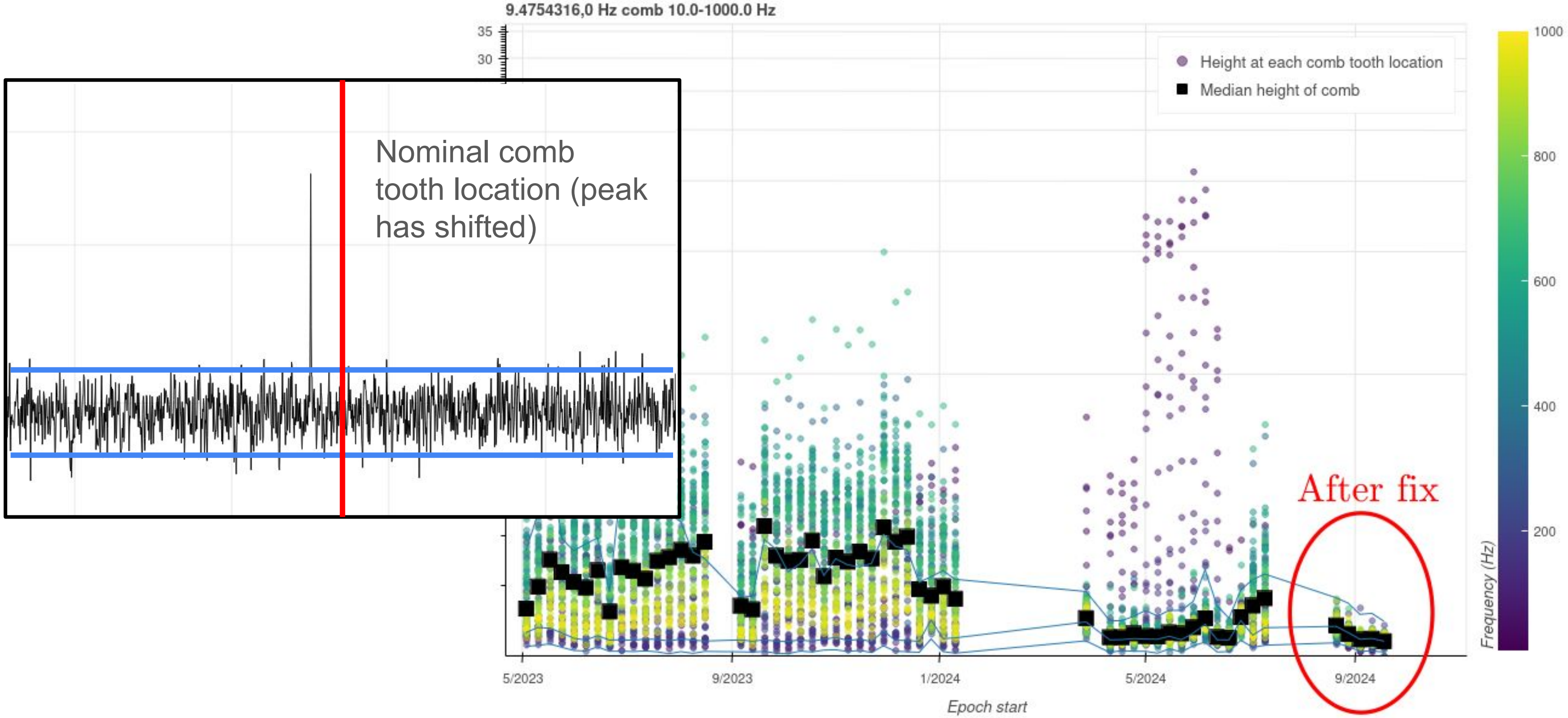
Tracking combs - whenwhere.py



Credit: Ansel Neunzert

https://alog.ligo-wa.caltech.edu/aLOG/index.php?callRep=79822

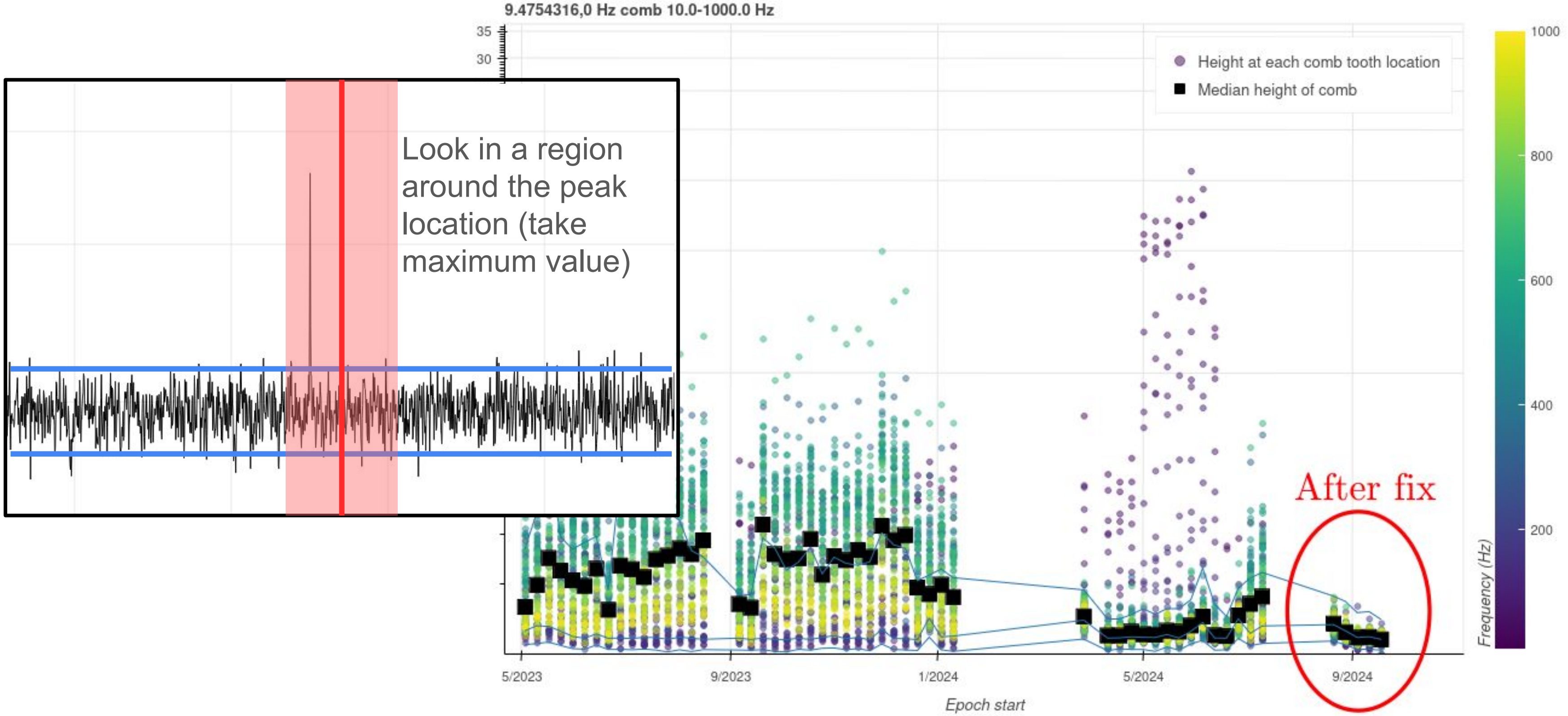
Tracking combs - whenwhere.py



https://alog.ligo-wa.caltech.edu/aLOG/index.php?callRep=79822

Credit: Ansel Neunzert

Tracking combs - whenwhere.py



https://alog.ligo-wa.caltech.edu/aLOG/index.php?callRep=79822

Credit: Ansel Neunzert

Checklist

- Segments for analyses
- (Gated) *h(t)* data
- Run-average spectra
- Peaks in spectrum + review
- Initial list from previous run + early checks for combs
- Longer list of vetted artifacts that have known causes

Vetted line artifact entries in lines lists

- All vetted lines need to have a frequency and a width for frequency band to exclude from analyses
  - Not automated, and can be subjective
- Combs need first harmonic visible and last harmonic visible
  - Note: there may be missing teeth of the comb so careful checks of higher frequencies is required
- If a known source is causing artifact, the segments of time when the source is active should be added

- All vetted artifacts should be reviewed by visual inspection of the spectra

Checklist

- Segments for analyses
- (Gated) *h(t)* data
- Run-average spectra
- Peaks in spectrum + review
- Initial list from previous run + early checks for combs
- Longer list of vetted artifacts that have known causes
- Reviewed and vetted list

Future needs

- How to generate lines lists faster? In O4, took ~few months after end of observing intervals
- Dependency on upstream data products (gated *h(t)* data, CAT1 vetoes)
- Automating steps to generate lines lists - likely we will still need human vetting of any automated lines list
- Improving automation tools that we do have
- A lot of O4 lines still unvetted - how can we improve this?
- General improvements to Fscan software, integration into LDVW
- Move mature bespoke tools that are broadly useful into Fscan (e.g., whenwhere.py)

Resources

- Detchar-requests
- Fscan / Fscan-configuration / Fscan documentation
- Line-investigations
- Summary pages
- LIGO line catalog repository

- Line investigations mailing list (aligo-lines)
- Line-investigations mattermost (chat.ligo.org)

End of part 1

Part 2: tutorial exercises

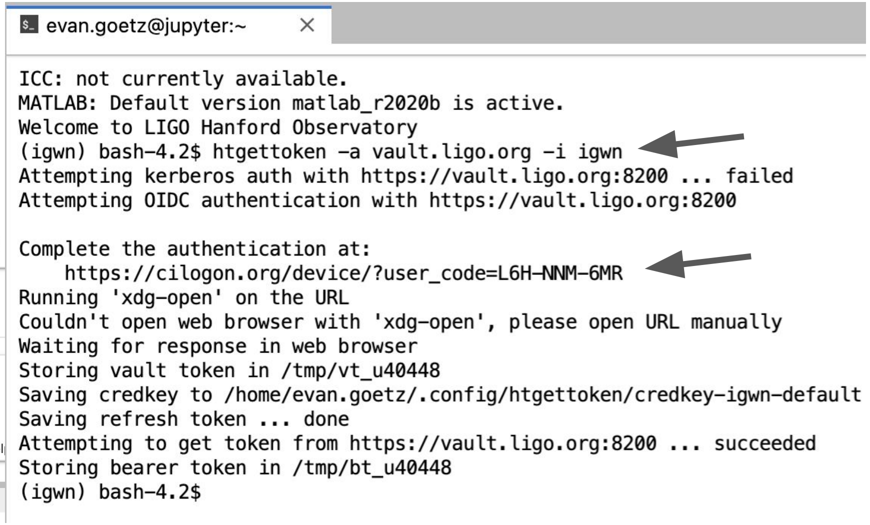
What we are hoping to accomplish

- Create SFTs (intermediate data products)
- Calculate spectra, coherence
- Interactive bokeh plots using Fscan code
- Run a simple Fscan workflow example
- Experiment with some of the bespoke scripts

Let’s see how far we get!

Logging in and data access

- This tutorial requires access to data/programs on LHO cluster, so you need an account to log in at LHO
- Use jupyter.ligo-wa.caltech.edu or ssh
- Fscan will need scitoken for access to data quality segment database (DQsegDB)



Command lines used in this tutorial

```
(igwn) bash-4.2$ cat
~evan.goetz/public_html/CW/IUCAAWorkshop/Lines_HandsOn_CommandLines.txt

Working on ligo-wa.caltech.edu
==============================

Obtain scitoken for DQsegDB:
----------------------------

(igwn) $ htgettoken -a vault.ligo.org -i igwn
Open website in browser when prompted and sign in using your LIGO.ORG access

Running lalpulsar_MakeSFTDAG:
-----------------------------
...
```

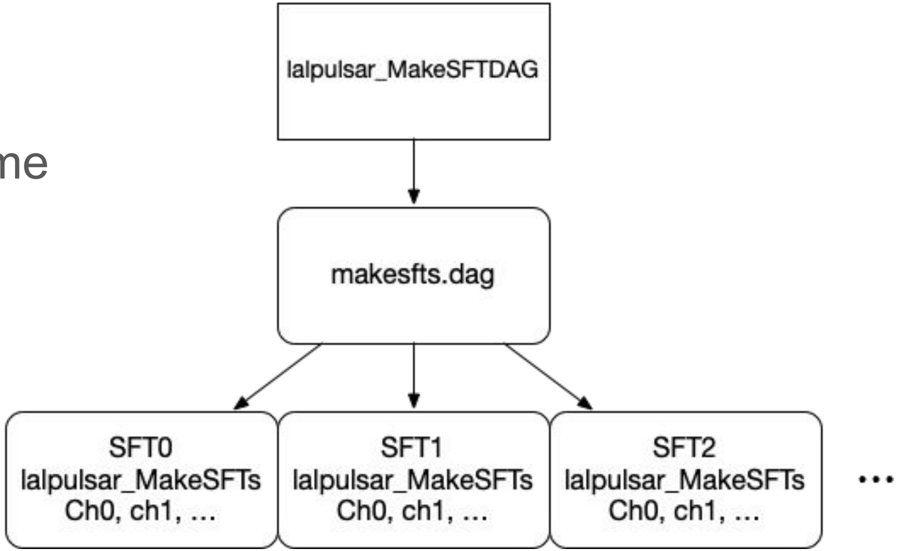
https://ldas-jobs.ligo-wa.caltech.edu/~evan.goetz/CW/IUCAAWorkshop/Lines_HandsOn_CommandLines.txt

A note of caution

- It is useful to explore possibilities for these programs, but you might find the particular choices of arguments do not work, the program crashes, or the output is unexpected
- Basically, I am not promising that every corner case of parameter choices have been tested. If you find something is broken, please let me know
- Bespoke scripts are not as well documented and are not as mature

Creating SFTs

- Remember: SFTs are just a specific file format of a Fourier transform
- Requirements: LALSuite, access to frame data **



** Should work using OSDF

Running lalpulsar_MakeSFTDAG **

```
[Log in to <jocelyn.bell>@ldas-grid.ligo-wa.caltech.edu]

(igwn) $ ~pulsar/gitrepos/lalsuite/_inst/bin/lalpulsar_MakeSFTDAG -f test.dag
-G test -a 1447286418 -b 1447372818 -L 86400 -J
~pulsar/gitrepos/lalsuite/_inst/bin --movesfts-path
~pulsar/gitrepos/lalsuite/_inst/bin -A ligo.dev.o4.detchar.linefind.fscan -U
${USER} -d H1_R -O 4 -K AUX -R 1 -k 7 -T 1800 -p . -N H1:IMC-F_OUT_DQ -F 10 -B
90 -w hann -P 0.5 --no-validate
(igwn) $ ls *.sft
…
(igwn) $ ~pulsar/gitrepos/lalsuite/_inst/bin/lalpulsar_dumpSFT -i
H-1_H1_1800SFT_O4AUX+R1+CIMCFOUTDQ+WHANN-1447371018-1800.sft | head
…
(igwn) $ ~pulsar/gitrepos/lalsuite/_inst/bin/lalpulsar_MakeSFTDAG --help
```

** Using pulsar account installation from LALSuite source due to delays in LALSuite package release

Generating spectra

- Different LALSuite tools read the SFT data
  - lalpulsar_spec_avg
    - Averaged, normalized spectra
    - Spectrograms
  - lalpulsar_spec_avg_long
    - Averaged, weighted/non-weighted spectra
    - Persistency
  - lalpulsar_spec_coherence
    - Coherence

Running LALSuite spectral tools **

```
[Log in to <jocelyn.bell>@ldas-grid.ligo-wa.caltech.edu]

(igwn) $ ~pulsar/gitrepos/lalsuite/_inst/bin/lalpulsar_spec_avg_long
--SFTs="/home/pulsar/public_html/fscan/1800s/H1_DMT-ANALYSIS_READY/day/2025111
6/H1_CAL-DELTAL_EXTERNAL_DQ/sfts/*.sft" --IFO=H1 --startGPS=1447286418
--endGPS=1447372818 --fMin=10 --fMax=100 --blocksRngMean=101
--timeBaseline=1800 --persistAvgSeconds=14400 --autoTrack=0.5
--outputDir=/home/${USER}/public_html/CW/tmp
(igwn) $ ls /home/${USER}/public_html/CW/tmp
spec_10.00_100.00_H1_1447286418_1447372818_line_times.csv
spec_10.00_100.00_H1_1447286418_1447372818.txt
spec_10.00_100.00_H1_1447286418_1447372818_PWA.txt

(igwn) $ ~pulsar/gitrepos/lalsuite/_inst/bin/lalpulsar_spec_avg_long --help
```

** Using pulsar account installation from LALSuite source due to delays in LALSuite package release

Running LALSuite spectral tools **

[Log in to <jocelyn.bell>@ldas-grid.ligo-wa.caltech.edu]

```
(igwn) $ ~pulsar/gitrepos/lalsuite/_inst/bin/lalpulsar_spec_coherence
--ChASFTs="/home/pulsar/public_html/fscan/1800s/H1_DMT-ANALYSIS_READY/day/2025
1116/H1_CAL-DELTAL_EXTERNAL_DQ/sfts/*.sft"
--ChBSFTs="/home/pulsar/public_html/fscan/1800s/H1_DMT-ANALYSIS_READY/day/2025
1116/H1_IMC-F_OUT_DQ/sfts/*.sft" --startGPS=1447286418 --endGPS=1447372818
--fMin=10 --fMax=100 --timeBaseline=1800
--outputDir=/home/${USER}/public_html/CW/tmp
(igwn) $ ls /home/${USER}/public_html/CW/tmp
spec_10.00_100.00_1447286418_1447372818_coh.txt
spec_10.00_100.00_H1_1447286418_1447372818.txt
spec_10.00_100.00_H1_1447286418_1447372818_line_times.csv
spec_10.00_100.00_H1_1447286418_1447372818_PWA.txt

(igwn) $ ~pulsar/gitrepos/lalsuite/_inst/bin/lalpulsar_spec_coherence --help
```

** Using pulsar account installation from LALSuite source due to delays in LALSuite package release

Plotting data

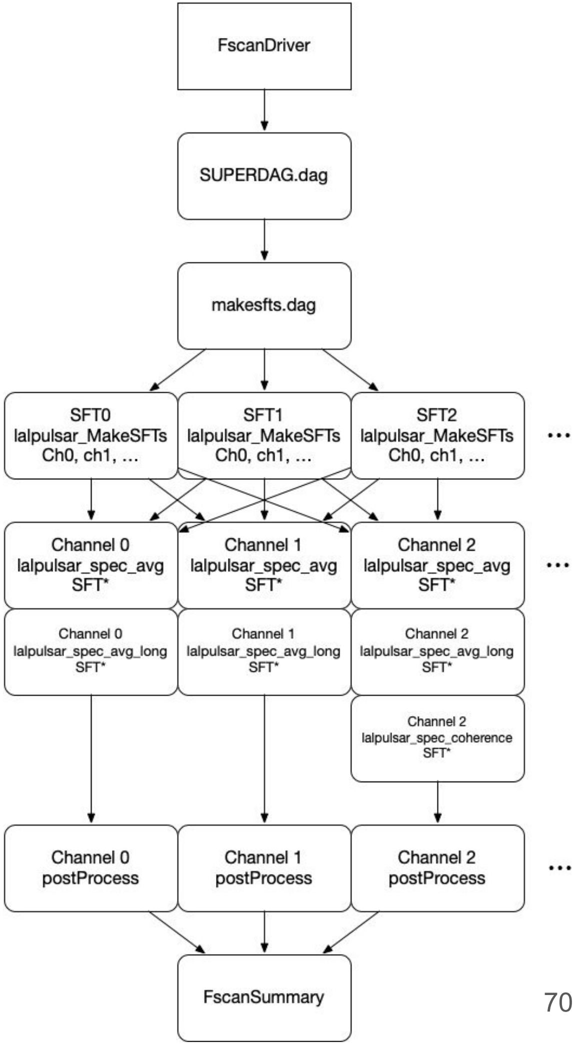
- Fscan package provides two ways to generate plots from the spectra:
    - Static plots: FscanStaticPlot (fscan.plot.static)
    - Interactive plots: FscanFineToothPlot (fscan.plot.finetooth)

```
[Log in to <jocelyn.bell>@ldas-grid.ligo-wa.caltech.edu]

(igwn) $ conda activate ~pulsar/.conda/envs/fscan-py3.10
(fscan-py3.10) $ FscanFineToothPlot
--spectfile=/home/${USER}/public_html/CW/tmp/spec_10.00_100.00_H1_1447286418_1
447372818.txt --fmin=10 --fmax=100
--outfile=/home/${USER}/public_html/CW/tmp/test.html --dtype=speclong

(fscan-py3.10) $ FscanFineToothPlot --help
```

https://ldas-jobs.ligo-wa.caltech.edu/~evan.goetz/CW/tmp/test.html

Fscan

- Produces spectra, spectrograms*, persistency, coherence, etc.; presented on summary page style webpages
- Fairly mature, but bugs are sometimes encountered–please report issues. Package maintained by EG, still undergoing development
- Production running Fscan on LHO and LLO: daily, weekly, monthly averages of *h(t)* and a subset of interesting channels monitored by LIGO DetChar
- Custom Fscans may be necessary for other averaging durations or different channel selection



* Fscan spectrogram is slightly different than usual spectrogram. Power in SFT vs time

Running Fscan

```
[Log in to <jocelyn.bell>@ldas-grid.ligo-wa.caltech.edu]

(igwn) $ conda activate ~pulsar/.conda/envs/fscan-py3.10
(fscan-py3.10) $ htgettoken -a vault.ligo.org -i igwn
(fscan-py3.10) $ FscanDriver --create-sfts=1 --plot-output=1
--accounting-group=ligo.dev.o4.detchar.linefind.fscan --full-band-avg=1
--analysisStart=2025-06-25-22:00:00 --analysisDuration=17h
--averageDuration=17h --seek-existing-sfts=0
--chan-opts=/home/${USER}/lscrepos/fscan-eg/configuration/production/hoft_only
.yaml --SFTpath=/home/${USER}/public_html/fscan/ --intersect-data=1
--delete-ascii=1 --run=0 --allow-skipping=1 --accounting-group-user=${USER}
(fscan-py3.10) $ FscanDriver --help
```

https://ldas-jobs.ligo-wa.caltech.edu/~evan.goetz/fscan/1800s/H1_DMT-ANALYSIS_READY/summary/17hours/20250625-220000/

Fscan channel configuration file

- Requirement to run Fscan workflow
- Contains information on channel specific options:
  Channel name
  Frame type
  Plot subband bandwidth
  Segment type
  Spectrogram frequency resolution
  Coherence A channel
  Persistency averaging length
  Auto track lines option
  Including extra subband plots

Production Fscan output directory structure

/home/pulsar/public_html/fscan/<Tsft>s/<DQ flag>/<avg dur>/<epoch>/

Within this folder:

<channel>/*

<frametype>_SFT_GEN/*

/home/pulsar/public_html/fscan/<Tsft>s/<DQ flag>/summary/<avg dur>/<epoch>/

Within this folder:

index.html

<channel>.html

Working with bespoke tools

- Make whenwhere trend plot (line-investigations/ansel-scripts/whenwhere.py)

```
[Log in to <jocelyn.bell>@ldas-grid.ligo-wa.caltech.edu]

(igwn) $ conda activate ~pulsar/.conda/envs/fscan-py3.10
(fscan-py3.10) $ python
/home/evan.goetz/lscrepos/line-investigations-eg/ansel-scripts/whenwhere.py
--analysisStart=20230524 --analysisEnd=20250129 --averageDuration=1day --greedy=1
--parent-path=/home/pulsar/public_html/fscan/1800s/H1_DMT-ANALYSIS_READY/
--channels=H1:GDS-CALIB_STRAIN_CLEAN --datatype=normpow
--outfile=/home/${USER}/public_html/temp.html --yaxlog --verbose --allpts --guidelines
--comb=4.9842193,0 --fmin=10 --fmax=700
(fscan-py3.10) $ python ~/lscrepos/line-investigations-eg/ansel-scripts/whenwhere.py --help
```

https://ldas-jobs.ligo-wa.caltech.edu/~evan.goetz/temp.html

- Make whenwhere spectrogram (https://ldas-jobs.ligo-wa.caltech.edu/~evan.goetz/tempspect.html)

```
(fscan-py3.10) $ python
/home/evan.goetz/lscrepos/line-investigations-eg/ansel-scripts/whenwhere.py
--analysisStart=20230524 --analysisEnd=20250129 --averageDuration=1day --greedy=1
--parent-path=/home/pulsar/public_html/fscan/1800s/H1_DMT-ANALYSIS_READY/
--channels=H1:GDS-CALIB_STRAIN_CLEAN --datatype=normpow
--outfile=/home/${USER}/public_html/temp.html --verbose -fmin=89.5 --fmax=90
```

Thank you

Working with the tools

- Make Fscan spectral comparison plot (fscan.plot.FineToothPlot)

```
[Log in to <jocelyn.bell>@ldas-grid.ligo-wa.caltech.edu]

(igwn) $ conda activate ~pulsar/.conda/envs/fscan-py3.10
(fscan-py3.10) $ FscanFineToothPlot
--spectfile=/home/pulsar/public_html/fscan/1800s/H1_DMT-ANALYSIS_READY/day/20250706/H1_GD
S-CALIB_STRAIN_CLEAN/fullspect_10.0000_2000.0000_1435810417_1435879419_speclong.npz
--desc=20250706 --freqcolname=f --datacolname=psdwt
--spectfile-ref=/home/pulsar/public_html/fscan/1800s/H1_DMT-ANALYSIS_READY/day/20250705/H
1_GDS-CALIB_STRAIN_CLEAN/fullspect_10.0000_2000.0000_1435720688_1435792406_speclong.npz
--desc-ref=20250705 --freqcolname-ref=f --datacolname-ref=psdwt --ylog=1
--outfile=/home/${USER}/public_html/test.html --yaxlabel="Weighted PSD" --fmin=10
--fmax=100 --legend=1
(fscan-py3.10) $ FscanFineToothPlot --help
```

https://ldas-jobs.ligo-wa.caltech.edu/~evan.goetz/test.html