

# JMonitor1Ldt plots for time offset study

Feifei Huang

# Recap

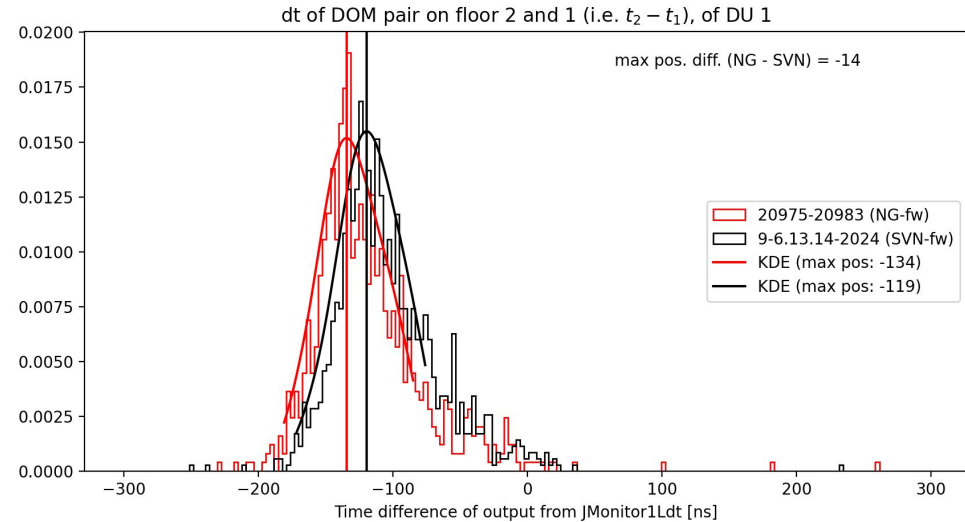
- Time offset seen in ARCA28 data with NG-fw running (~ 11 months), triggered by DOM rebooting
- Git issue:  
[https://git.km3net.de/working\\_groups/calibration/-/issues/99#note\\_71946](https://git.km3net.de/working_groups/calibration/-/issues/99#note_71946)
- Dorothea produced plots using the output of JMonitor1Ldt:
  - Use SN hits as input
  - Checks the L1 time difference between nearby DOMs in one DU (the hits are from muons)
  - Do not depend on fits

## Steps I did:

- Run JMonitor1Ldt on runs with SVN-fw and NG-fw
- Plot the output of the JMonitor1Ldt (L1 time correlations between DOMs)
- Run a KDE (kernel density estimation) to fit the distribution and get the max position
- Plot the max position difference for the same DOM pairs with NG-fw and SVN-fw
  - Expect it to be around +/-16 ns, and 0
- Try to answer this question: Could this max position difference be used as an observable for detecting DOMs in “offset” mode?

# difference of max position

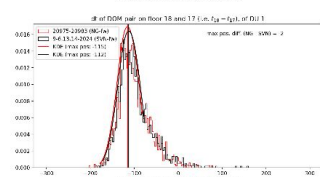
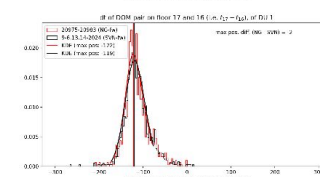
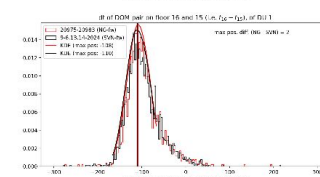
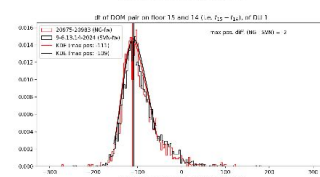
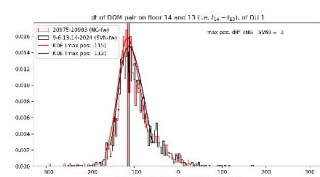
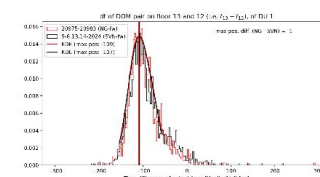
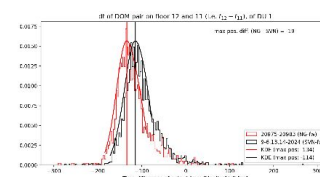
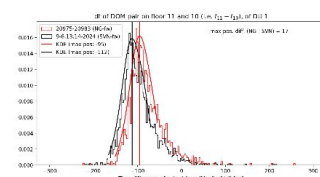
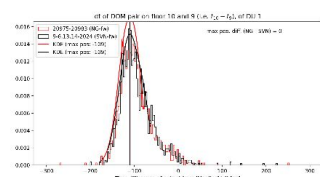
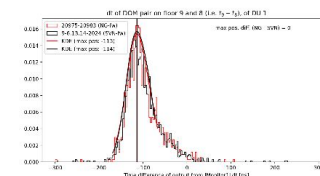
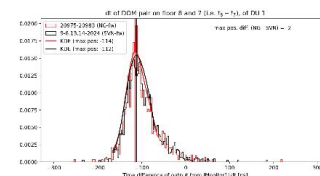
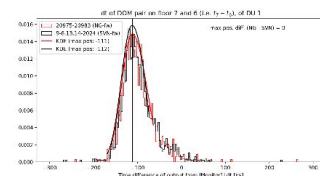
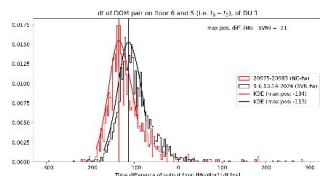
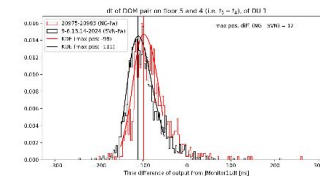
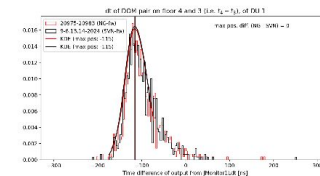
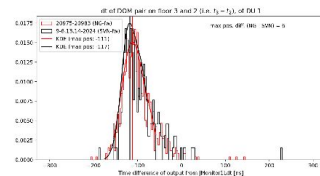
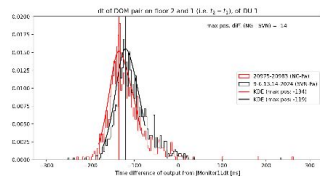
- 8 runs for **NG-fw**: run 20975-20983
- 21 runs for **SVN-fw** (Sep. 6,13,14)
- KDE done with points around +/- 50 ns around the max point (reduce effect of tails)
- DOM 1 & 2 on DU1, their KDE max position difference (NG - SVN) is **-14 ns**
  - Close to the 16 ns seen in lab



# DU1:

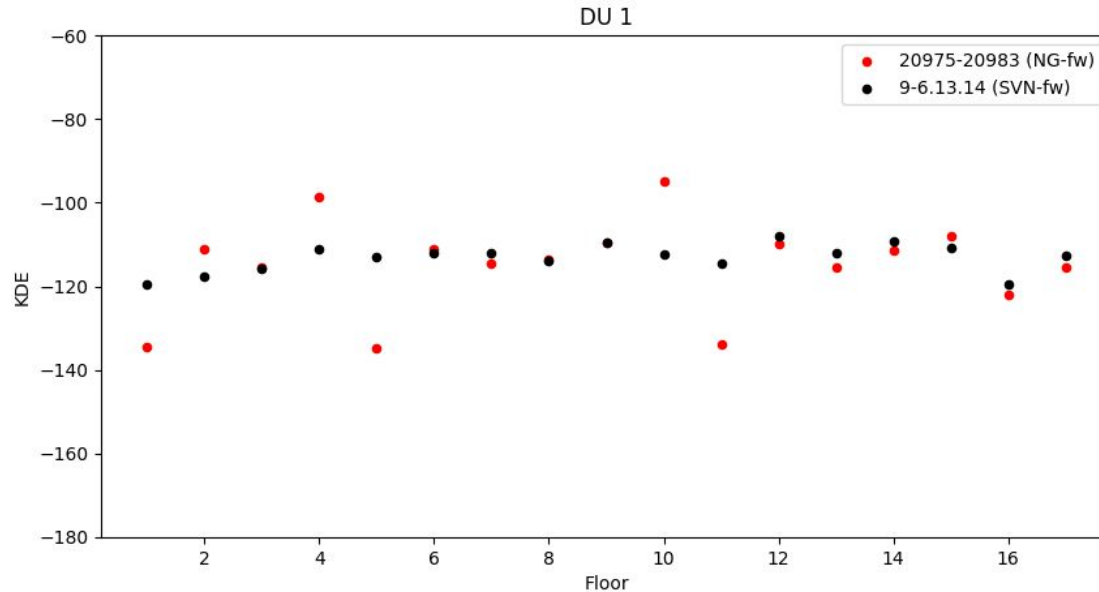
17 DOM pairs of  
DU1.

See more plots in  
[this google folder](#)



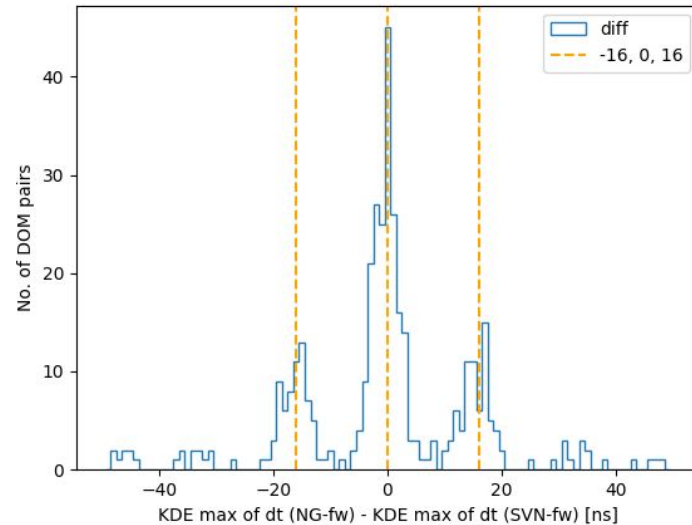
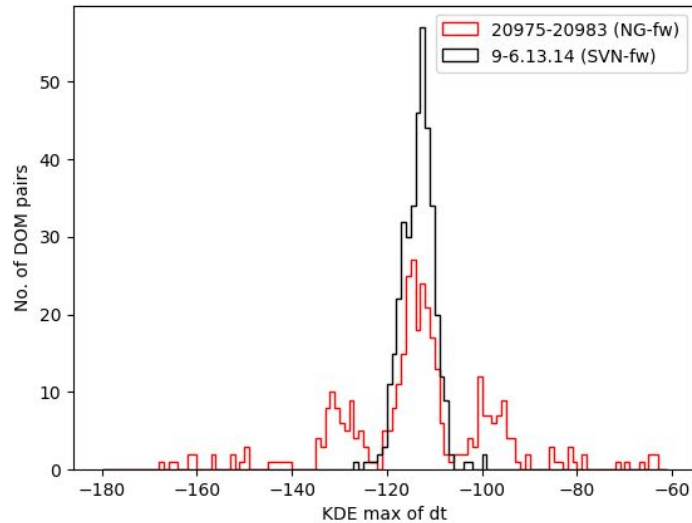
# max pos. vs floor

KDE max position of DOM pair (floor, floor+1) vs floor:



# max position & max position difference

- 1D distribution for the max position for all DUs (left):
- Plot the difference between them (right):
  - Indeed see 3 major groups centered around 0, -16, +16 ns, with a spread
  - However also see a few large values > 40 ns. Not sure why?



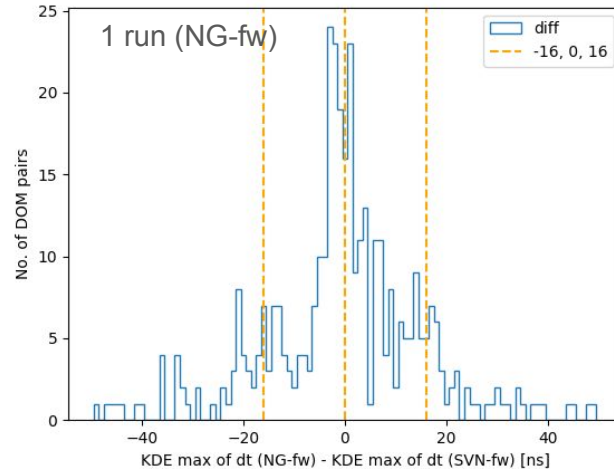
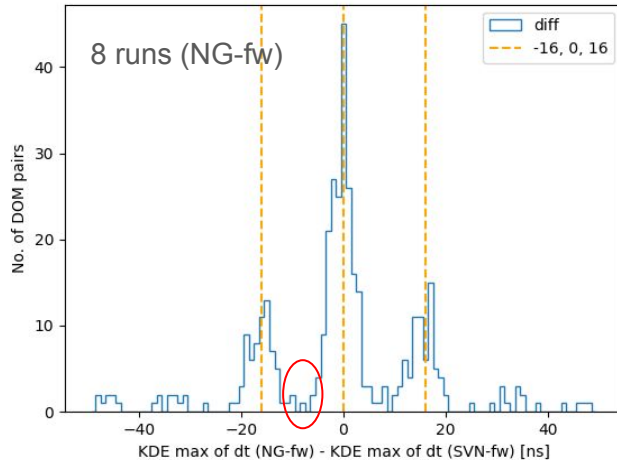
# Ideally, one can deduce the DOM(s) with an offset

- We know a DOM can be either at the baseline (“good”) or have a time offset of +16 ns compared to the baseline after a reboot
- $dt = t_2 - t_1$ 
  - if dt shifts positively ( $\sim +16$  ns), then: DOM1 “good”; DOM 2 “offset”
  - if dt shifts negatively ( $\sim -16$  ns), then: DOM1 “offset”; DOM2 “good”
  - if dt does not shift ( $\sim 0$ s), can be one of the following situations:
    - Both are good;
    - Both in “offset”.
- We have 17 dt values for the 17 DOM pairs inside a DU:
  - if at least see one non-zero dt (actually it’ll be at least two, one positive one negative):
    - one can deduce the state of the DOM one by one, using the logic above
  - but if all 17 DOM pairs are all 0
    - either all good or all “offset” (but the latter should happen with a very small probability)



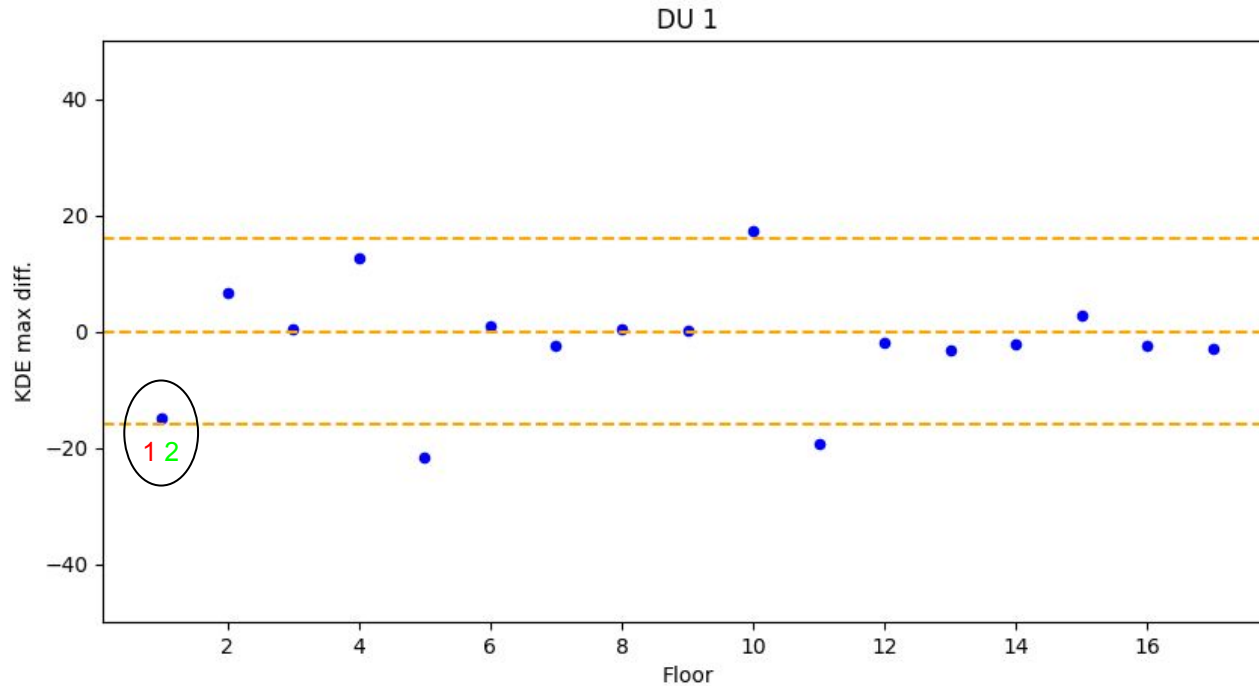
# Issues in practice

- A few points in the middle  $\rightarrow$  but can be still enough with complementary info from other pairs in the DU (or other methods)
- Worse with low statistics (see 8-run vs 1-run comparison)  $\rightarrow$  need more runs to accumulate statistics
- Some DOMs are lost  $\rightarrow$  less info to help disentangle the possible scenarios



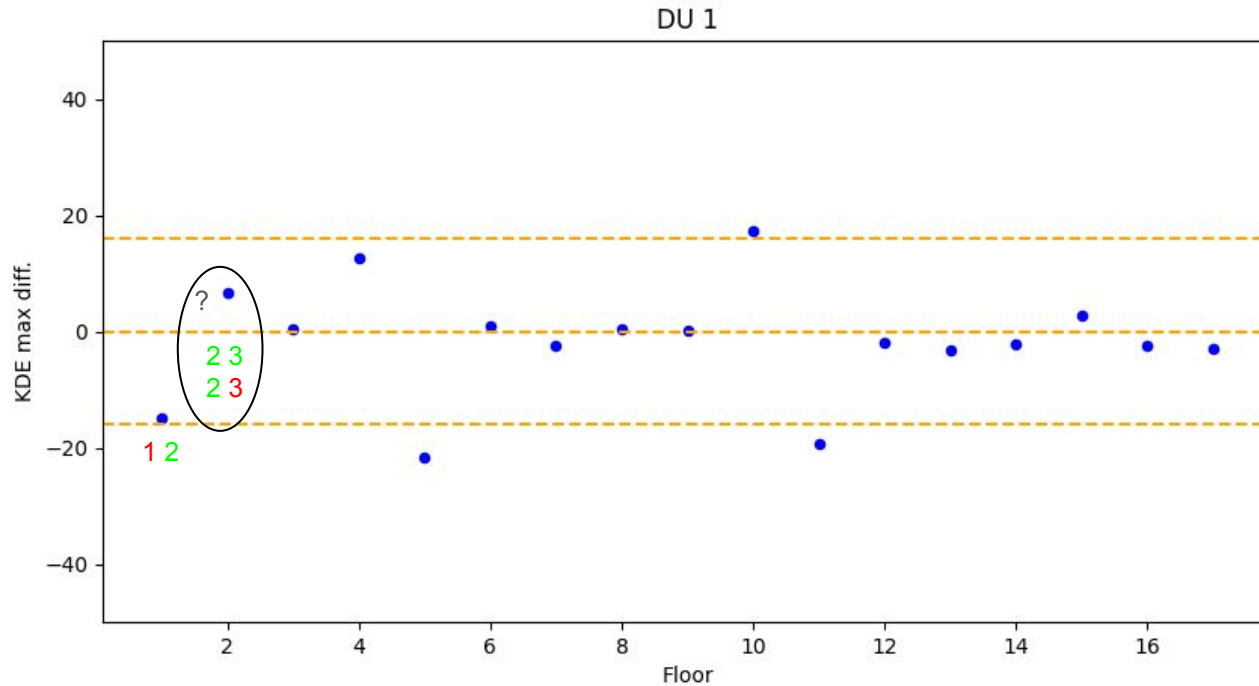
# Deduction example 1

Green: good; Red: offset state. Start with DOM pair (1, 2):



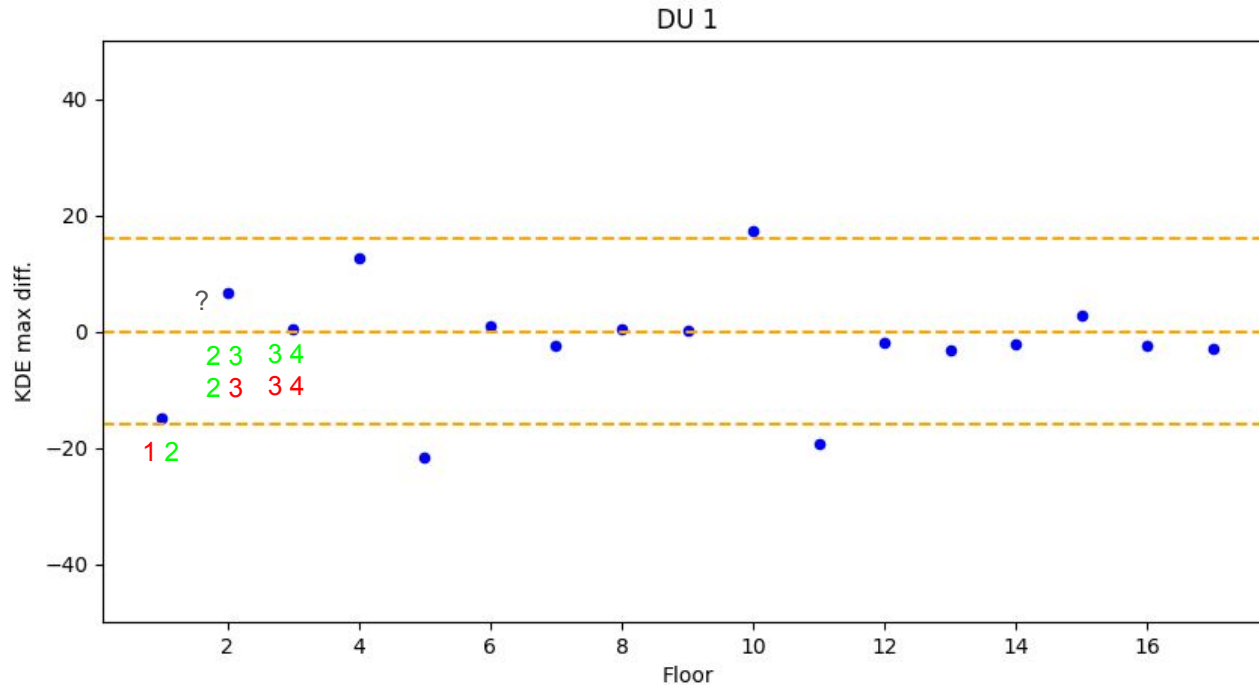
# Deduction example 1

Not sure if (2, 3) should be considered as 0 or shifted



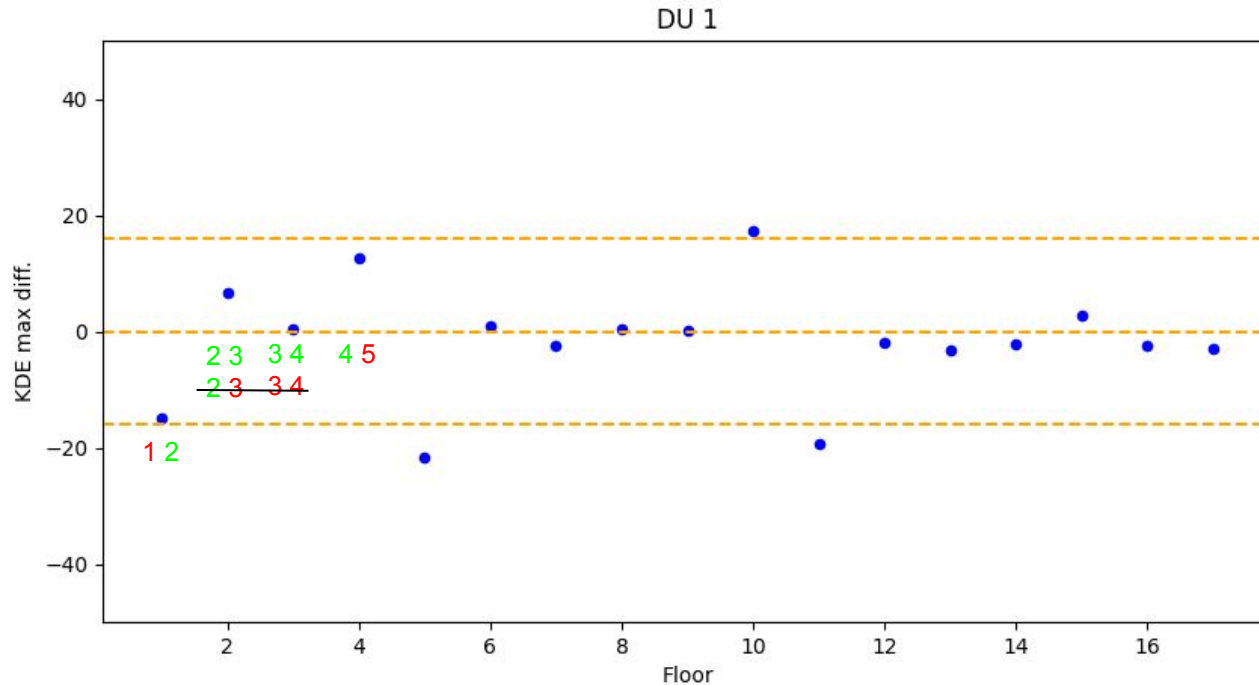
# Deduction example 1

Looking at (3, 4) still doesn't solve the problem. Let's keep going...



# Deduction example 1

(4, 5) is positive, so 4 is good, 5 is “offset”, this disentangles the scenarios: 3 has to be good

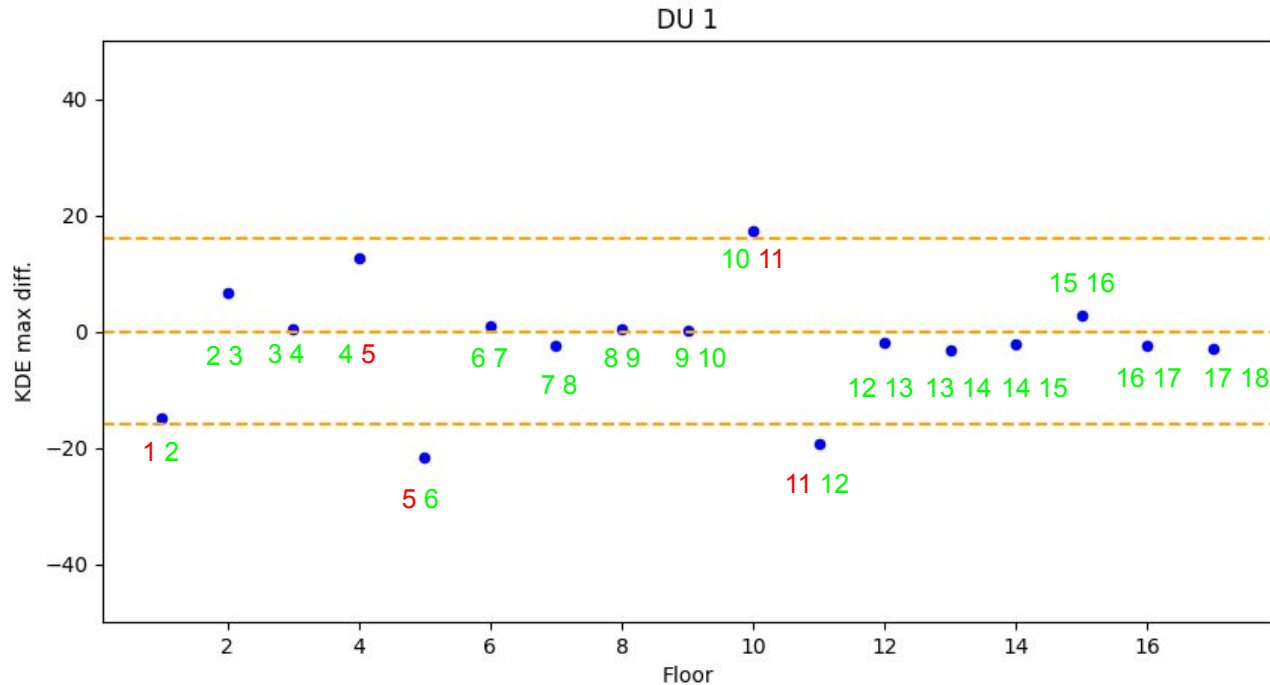


# Deduction example 1

It's easy to continue the deduction for the rest of the DOMs. In the end, we know:

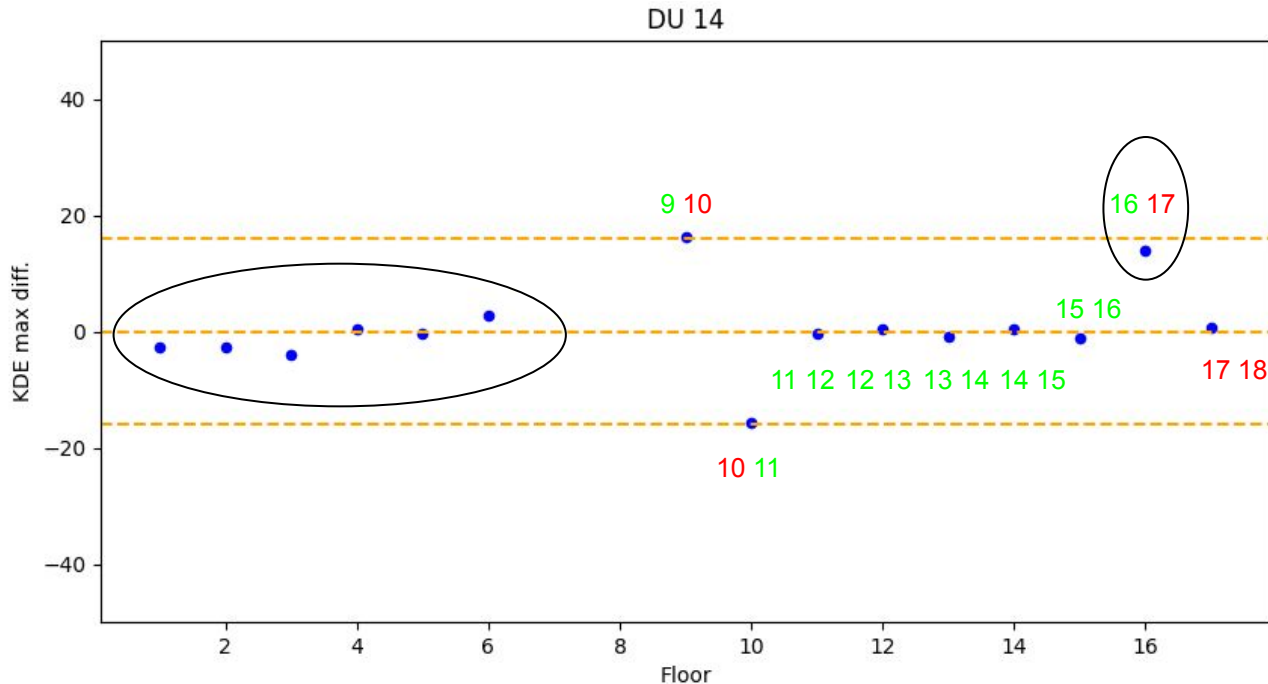
Offset: DOM **1, 5, 11**

Others: all good



# Deduction example 2

Start with (16,17). No data point at (7,8) & (8,9), i.e. DOM 8 is lost, thus our chain of logic is broken here. Also, the first 6 pairs are all 0: are DOMs 1-7 all good, or all “offset”?



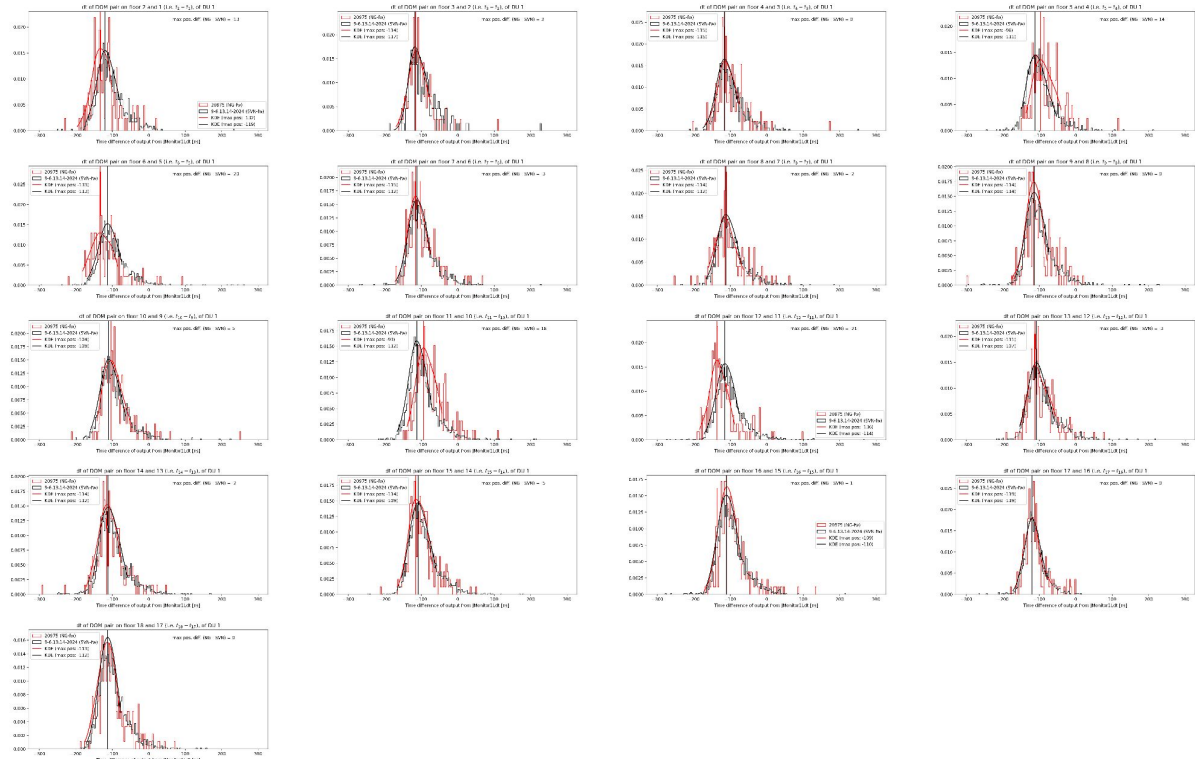
Offset: DOM 10, 17, 18

DOM 1-7: more likely to be all good.

- All 7 offset → low probability
- Use other methods to verify?

# Can one run provide enough statistics?

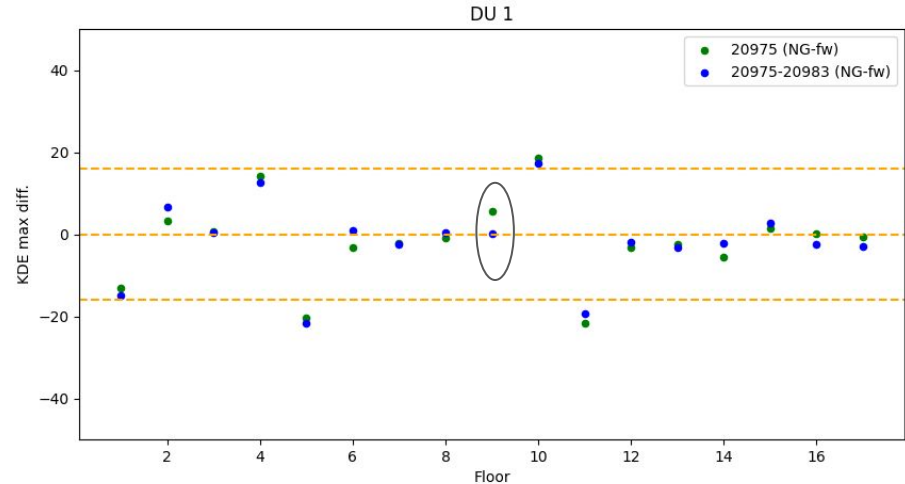
Plots in this [folder](#)





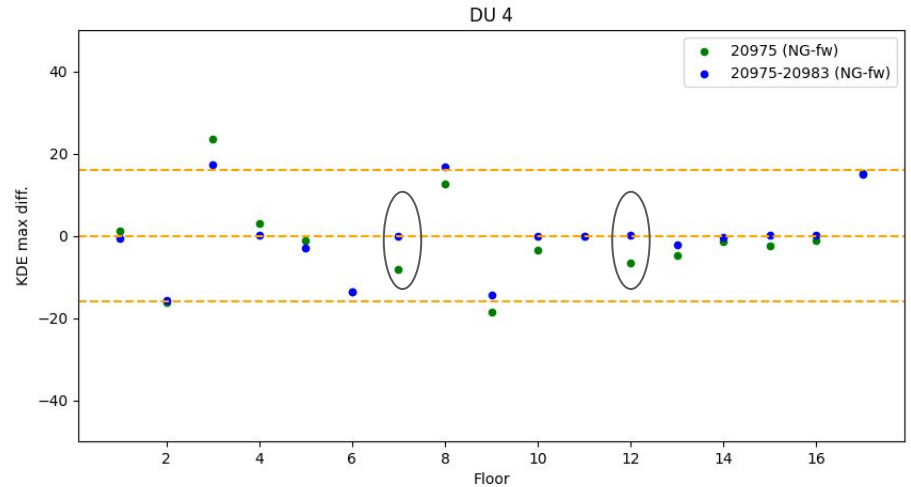
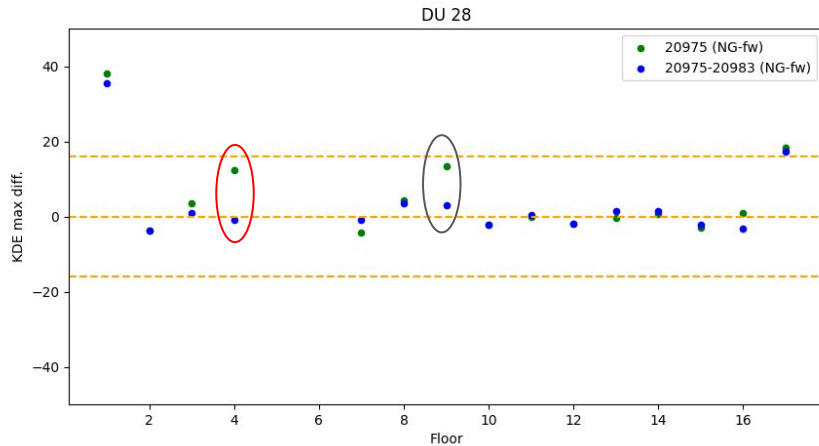
# Can one run provide enough statistics?

- Green: with 1 run, Blue: 8 runs
- Most of the points close enough
- a few points may wander into the middle region → info from other pairs could help resolve the confusion, but still too much uncertainty



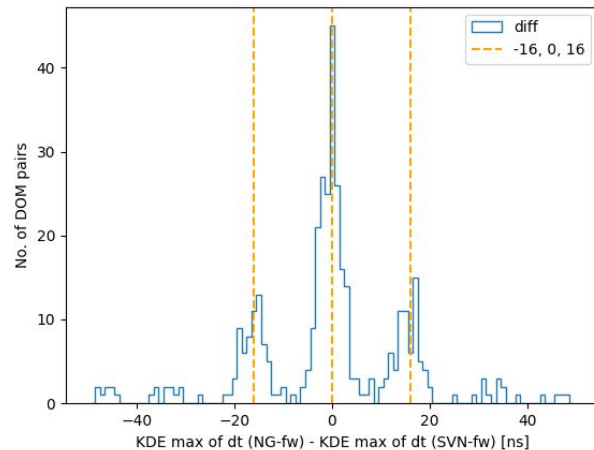
# Can one run provide enough statistics?

- Variations can be almost as large as the 16 ns range (the points in red circle), this could mean a misidentification



# Summary

- In the max position difference of the output **dt** (from JMonitor1Ldt) between data with NG-fw & SVN-fw, we see 3 major groups centered around -16, 0, 16 ns
  - see a few large values > 40 ns. Not sure why?
- Could potentially use this dt max position difference (w.r.t SVN-fw) as an observable for identifying the DOMs in “offset” mode after a reboot
  - Need to do a little deduction
  - Info from other pairs can help resolve confusion of mid-area points
  - All 0 values may indicate all-good OR all-offset DOMs (though I assume the latter has lower probability)
  - Need to collect more than a few hours of data to accumulate statistics for accuracy
    - Variations too large with 1 run



# Backup

See > 40 ns  
difference:

