Data Quality Check: Version & Server Updates, Spectrum Check & Period Division

Seung-mok Lee

Sunkee kim's lab, Seoul Nat'l Univ.

COSINE Collaboration

Jul., 21st, 2021

Purpose of Data Quality Check

- We encountered the type 2 (Gaussian) noise at low energy.
- To remove those events, we are going to discard sub-runs having too high event rate at low energy region.

Procedure of Data Quality Check

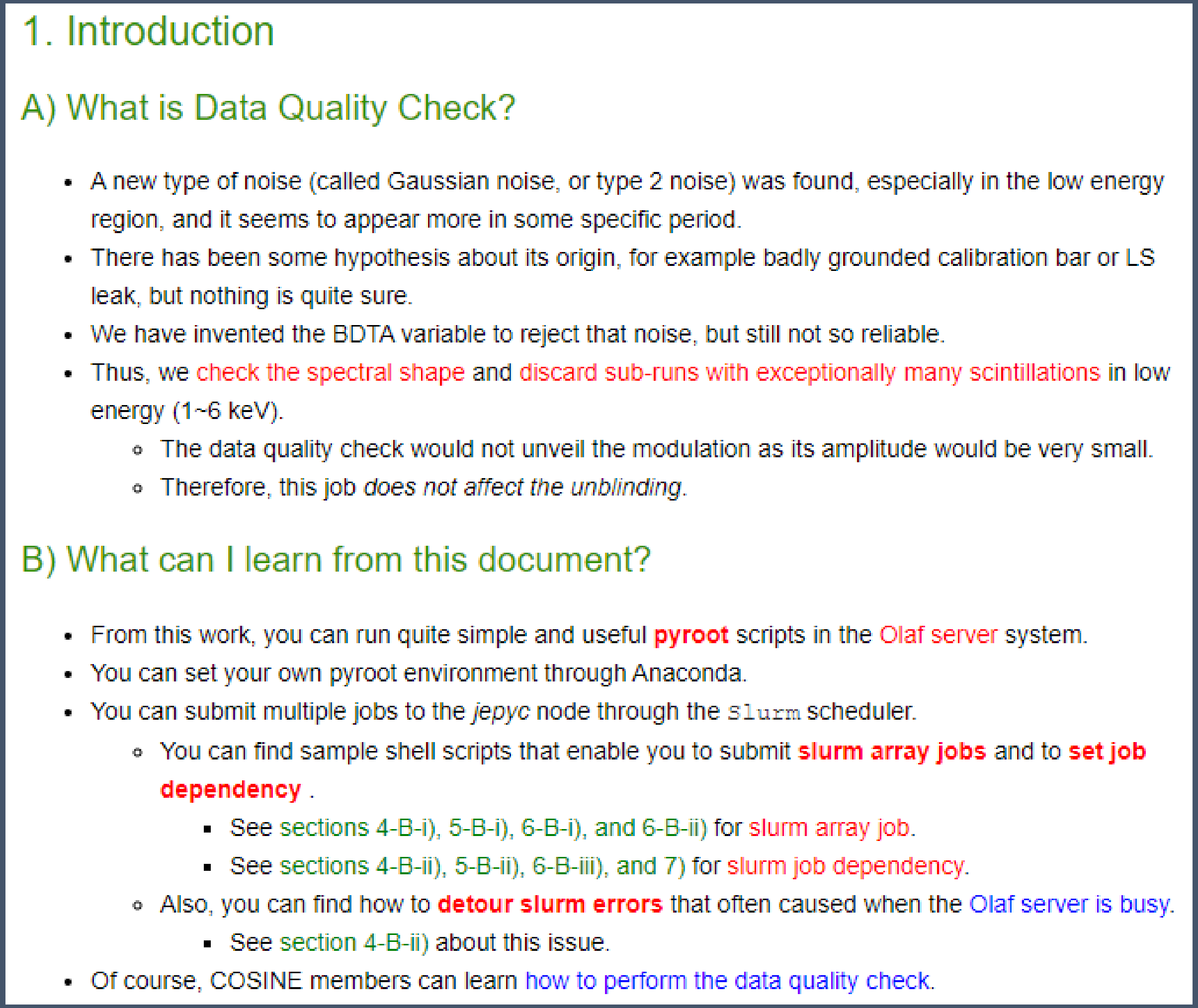
- First, rate vs time behavior checked to identify any periods of sustained high rate.
- Next, for each sub-run count number of events in $1-6\ \mathrm{keV}$ region after event selection and histogram results.
    - Without noise, the rate distribution should be Poissonian.
    - Fit histogram to the Poissonian and determined number of events in a sub-run that excludes a sub-run at various confidence levels.
    - Until the previous version (V00-04-15), 99.9% C.L. was used to remove sub-runs.
        - William, Feb 3rd, 2020
        - William, July 28th, 2020

Updates on Data Quality Check

- Data quality check is outdated.
  - Last update was July 2020, run 1873, production V00-04-15. Performed at cup cluster.
  - Used old BDT cuts for event selection.
- First, the manual was written.
- Second, scripts were modified for Olaf server and Slurm scheduler.
- Third, the event selection was updated. Tighter BDT and additional ES.
- Fourth, data quality check is now up to date.
  - V00-04-19, run 1939.
- Fifth, period division for crystal 4 and spectrum comparison were performed.
- The preliminary version of COSINE database is prepared.

Manual on Data Quality Check

- The manual deals with from how to set PyRoot environment, how to write and execute Slurm job submission shell scripts, to how to perform data quality check.
  - It also illustrates about Slurm array job and job dependency options.
  - Codes at
    (Olaf): /mnt/lustre/ibs/seungmok/Share/CosineDQC/
- I hope it helps you to start research with Olaf server.



1. Introduction

A) What is Data Quality Check?

- A new type of noise (called Gaussian noise, or type 2 noise) was found, especially in the low energy region, and it seems to appear more in some specific period.
- There has been some hypothesis about its origin, for example badly grounded calibration bar or LS leak, but nothing is quite sure.
- We have invented the BDTA variable to reject that noise, but still not so reliable.
- Thus, we check the spectral shape and discard sub-runs with exceptionally many scintillations in low energy (1~6 keV).
  - The data quality check would not unveil the modulation as its amplitude would be very small.
  - Therefore, this job *does not affect the unblinding*.

B) What can I learn from this document?

- From this work, you can run quite simple and useful **pyroot** scripts in the Olaf server system.
- You can set your own pyroot environment through Anaconda.
- You can submit multiple jobs to the *jepyc* node through the `Slurm` scheduler.
  - You can find sample shell scripts that enable you to submit **slurm array jobs** and to **set job dependency** .
    - See sections 4-B-i), 5-B-i), 6-B-i), and 6-B-ii) for slurm array job.
    - See sections 4-B-ii), 5-B-ii), 6-B-iii), and 7) for slurm job dependency.
  - Also, you can find how to **detour slurm errors** that often caused when the Olaf server is busy.
    - See section 4-B-ii) about this issue.
- Of course, COSINE members can learn how to perform the data quality check.

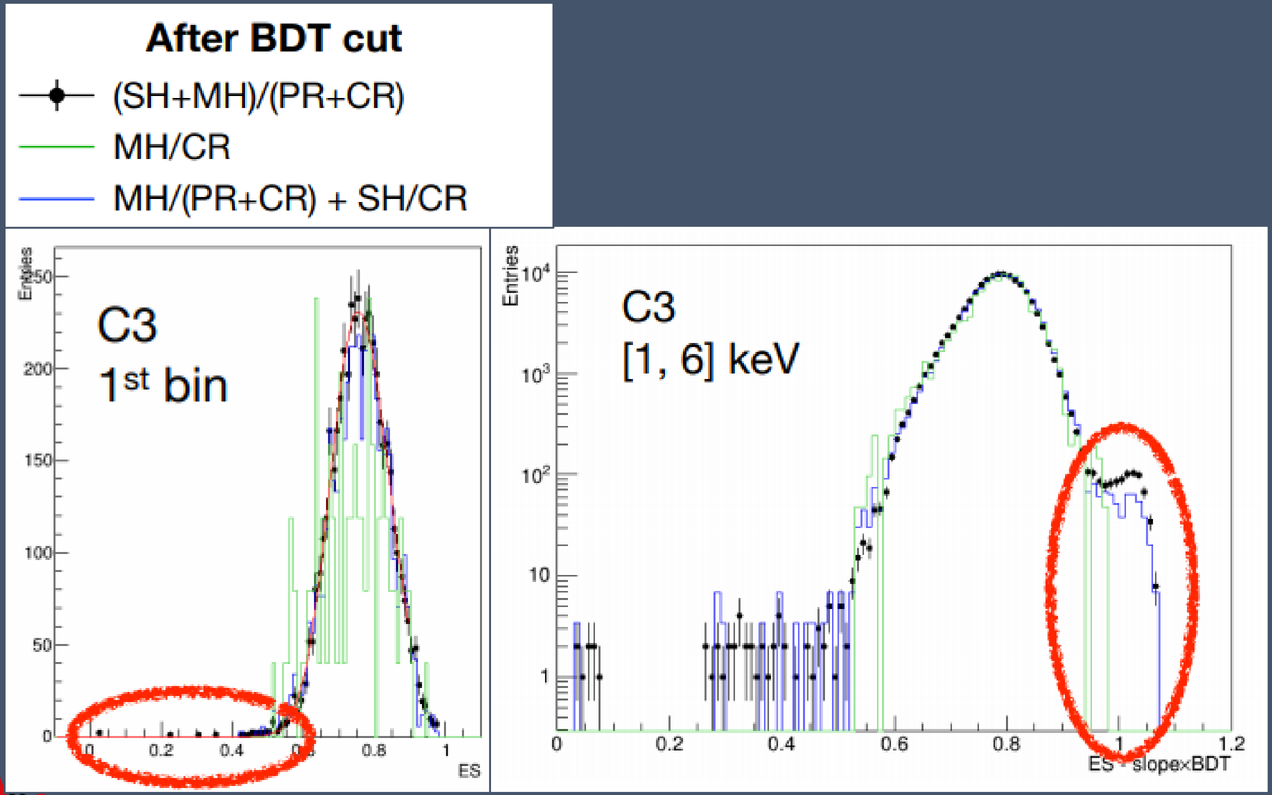
Job Flow of Data Quality Check in Olaf

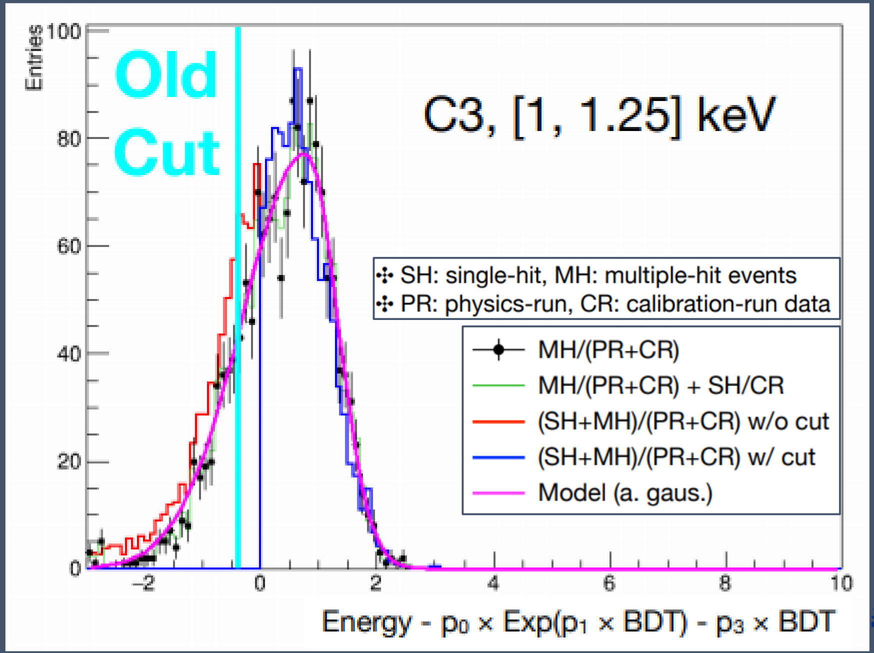
- Using the Slurm scheduler in Olaf server, I could design the work structure so that a single command line execution performs the whole data quality check study.
  - Parallel running of data trimming.
  - Calculate event rate.
  - Draw plots and tag bad sub-runs.
- Only 1~2 days took.

```
────TIME────TIME────TIME────→
(trial_)perform_end_to_end.sh
│
├(1) (trial_)submit_trim_singlehit.sh
│  │    at $MAIN/sources/4.TrimmingData/
│  ├(1-a) perform_trim_singlehit_onerun.sh 1544
│  ├(1-b) perform_trim_singlehit_onerun.sh 1545
│  ├...
│  └(1-z) perform_trim_singlehit_onerun.sh 1935
│
├─(2) submit_graph_rate_vs_time.sh
│   │    at $MAIN/sources/5.ExtractRate/
│   │      after (1)
│   ├(2-a) perform_graph_rate_vs_time.sh 2
│   │        after (1-a~z)
│   ├(2-b) perform_graph_rate_vs_time.sh 3
│   │        after (1-a~z)
│   ├...
│   └(2-e) perform_graph_rate_vs_time.sh 7
│            after (1-a~z)
│
└──(3) submit_draw.sh
     │    at $MAIN/sources/6.DrawPlots/
     │    after (2)
     ├(3-a) perform_draw_rate_vs_hist.sh
     │        after (2-a~e)
     └(3-b) perform_draw_rate_hist.sh
              after (2-a~e)
```

Criteria Update on Data Quality Check

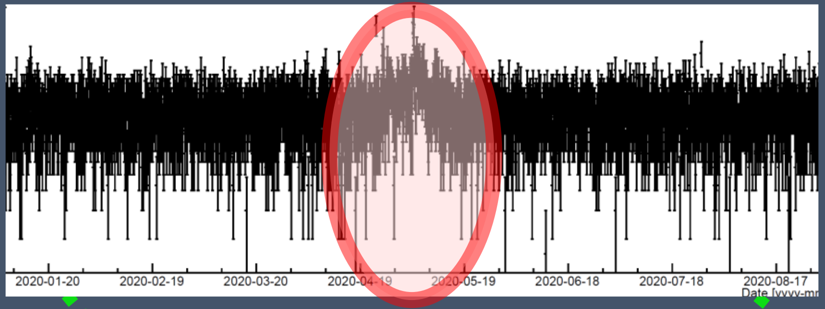
- BDT cut was updated on last October.
  - YJ Ko, Oct 21st, 2020
  - BDT criteria became tighter, and ES (DAMA variable) cut was added.

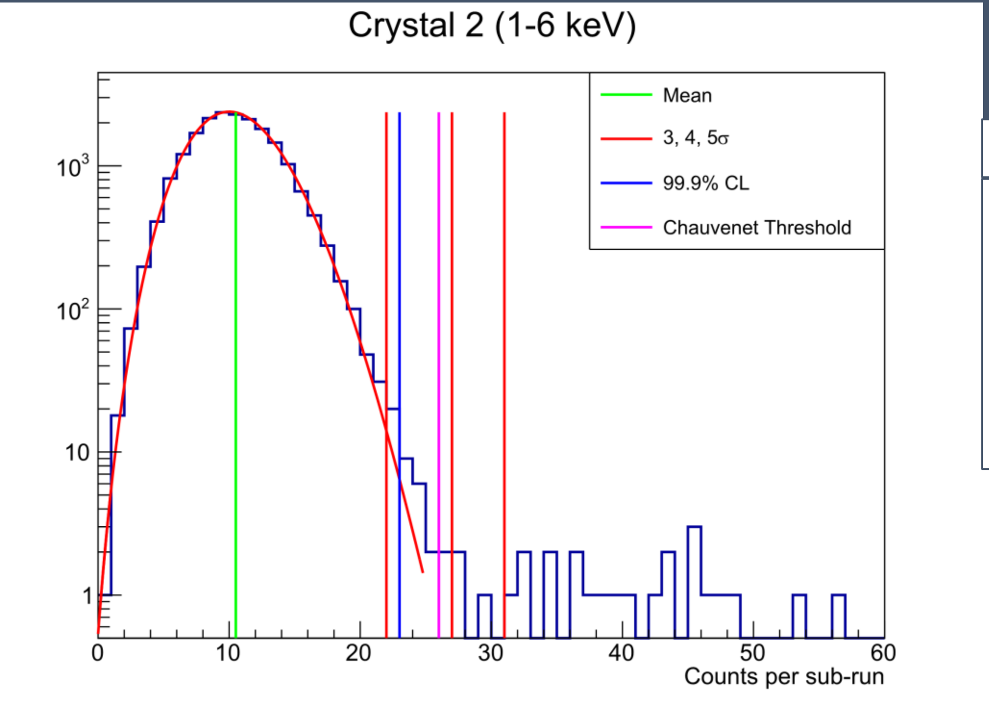


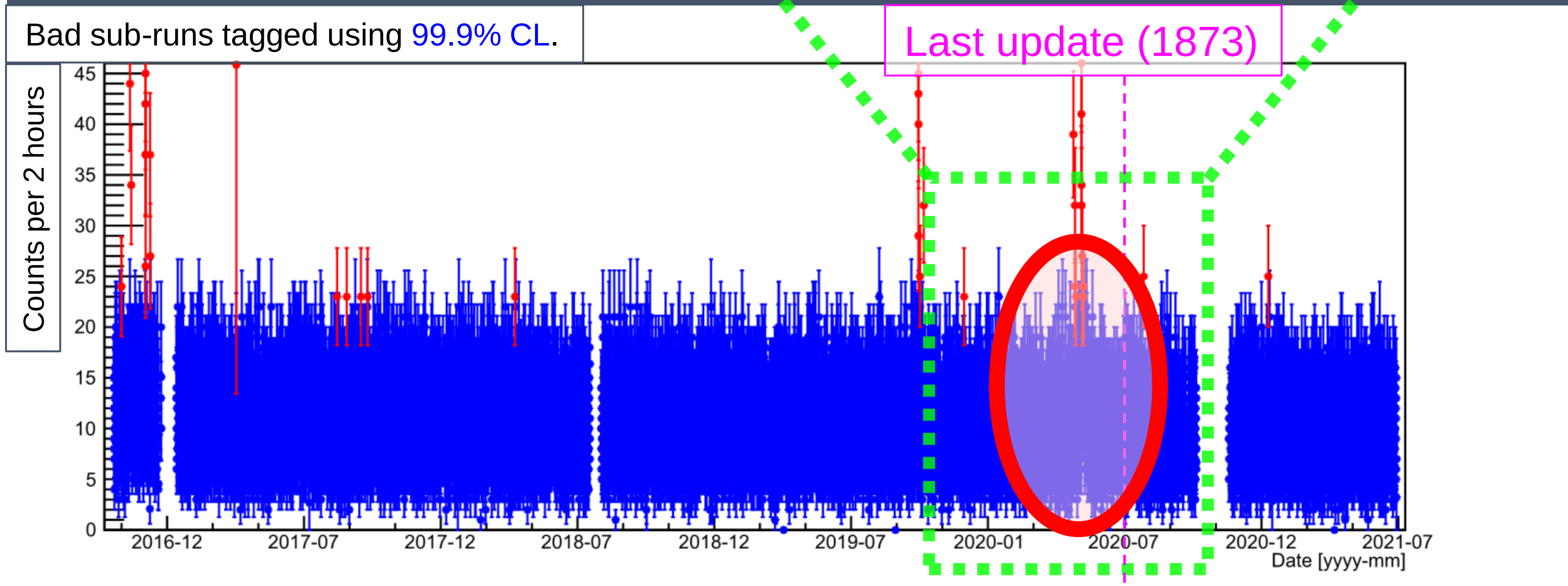


Crystal 2 Result

- Long term instability from 20-04-18 (1873.061) to 20-05-18 (1873.410).

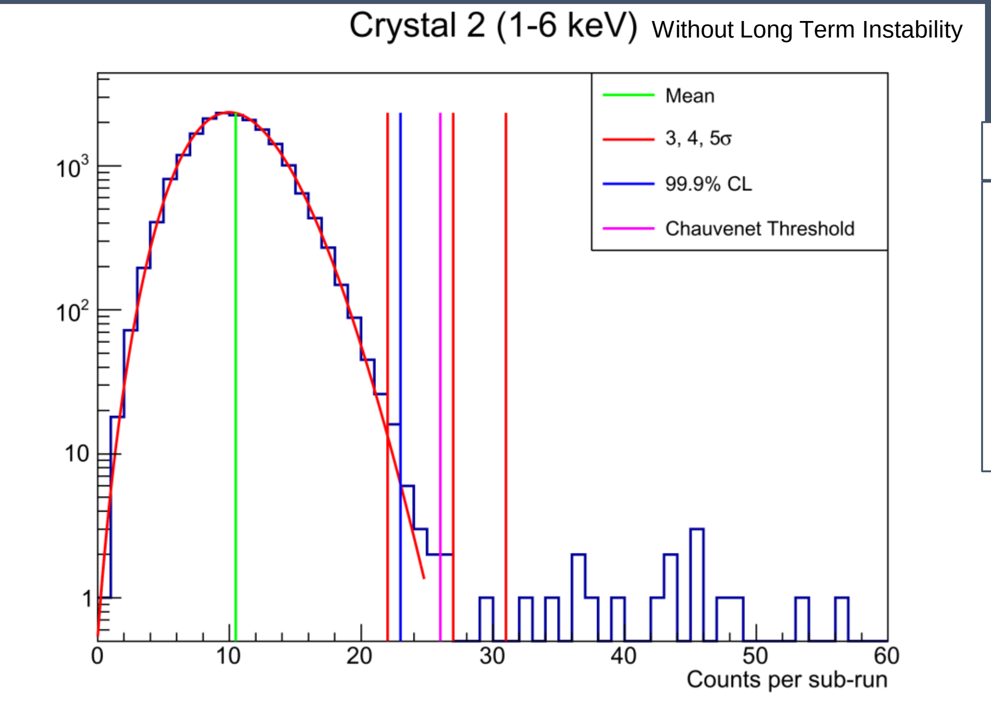


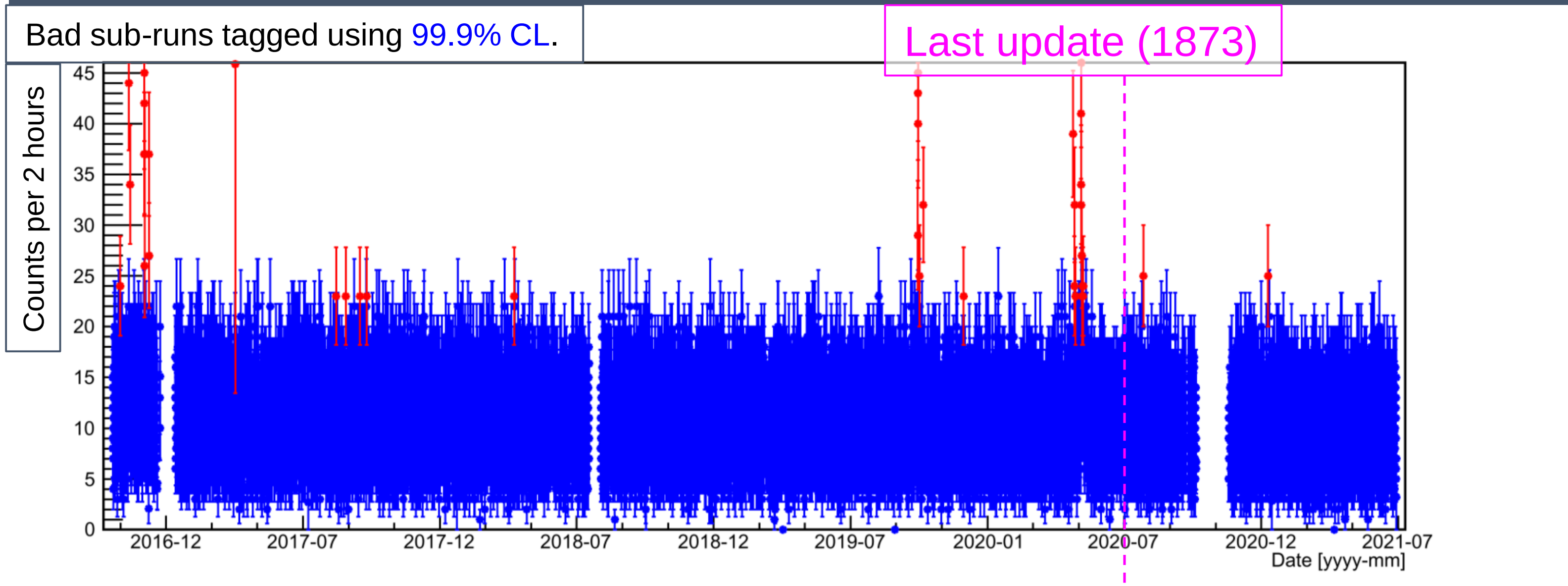




Crystal 2 Result

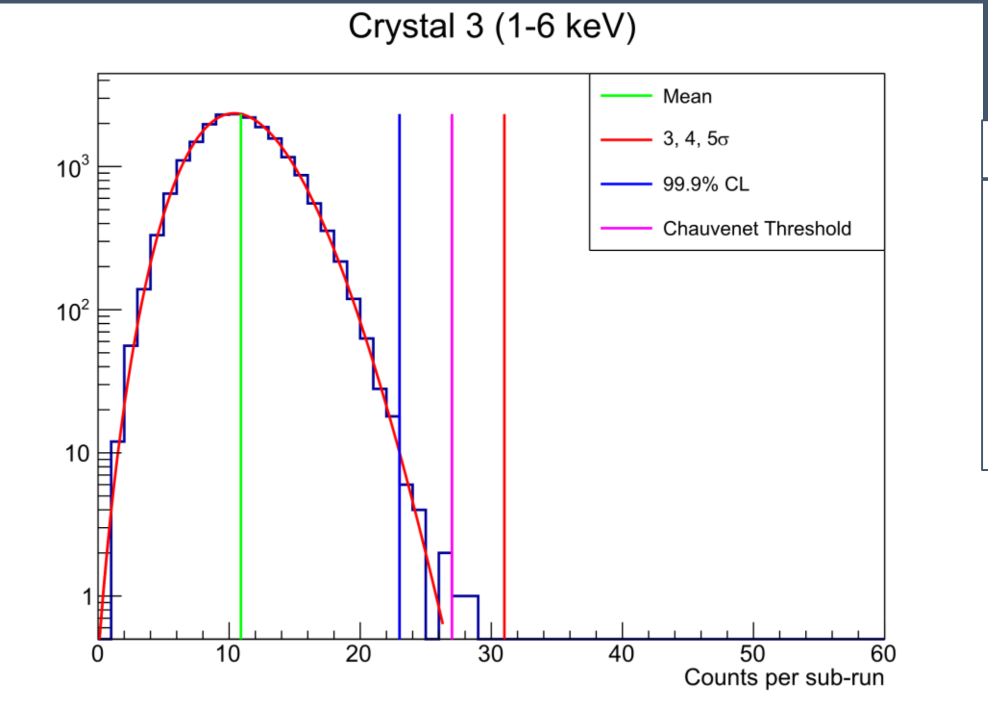
- Long term instability from 20-04-18 (1873.061) to 20-05-18 (1873.410).
- Removing that instability, we get the following well-fitted plot.
- To be sure for noise rejection, 99.9% CL was adopted.

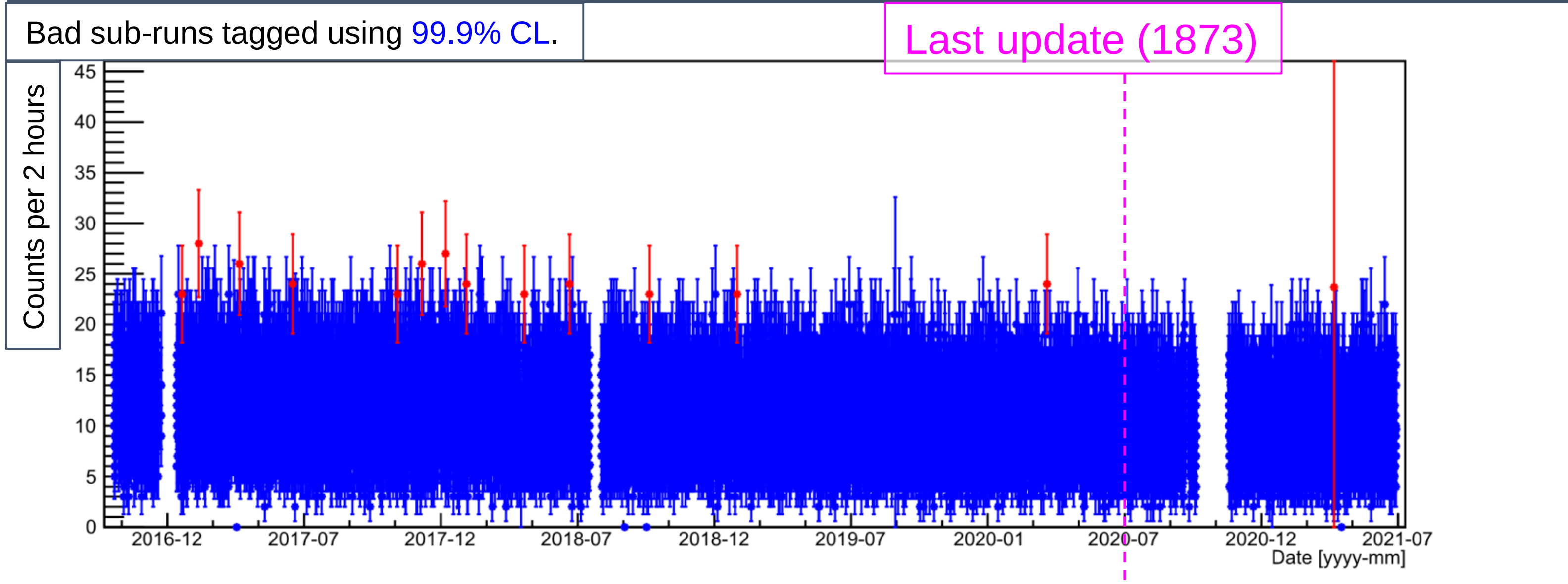




Crystal 3 Result

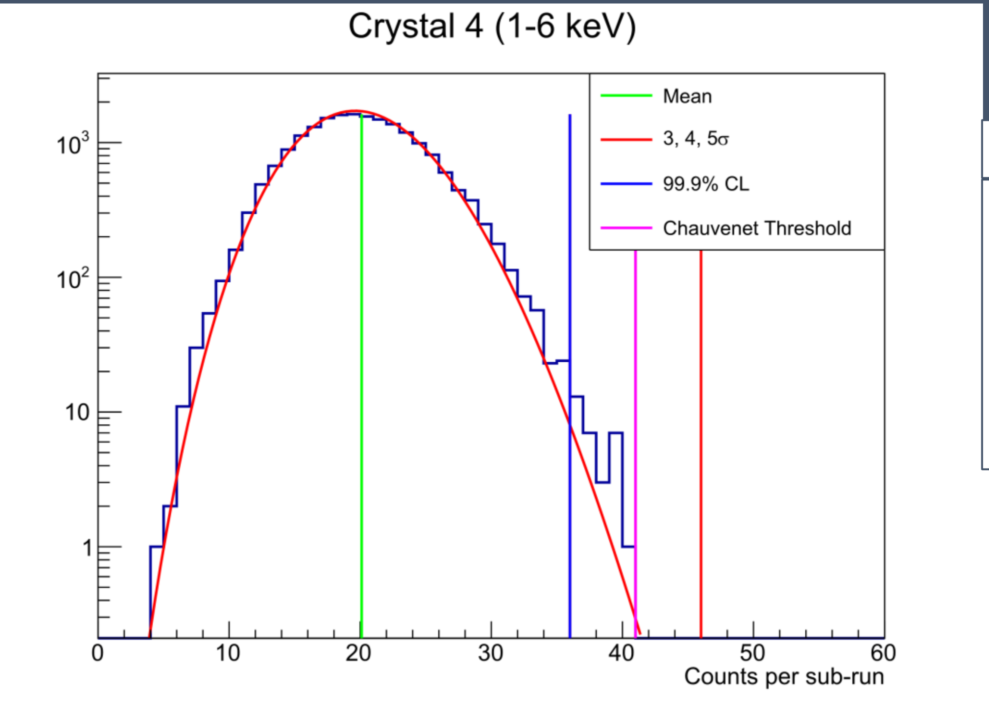
- Distribution looks good. Some sub-runs were tagged.

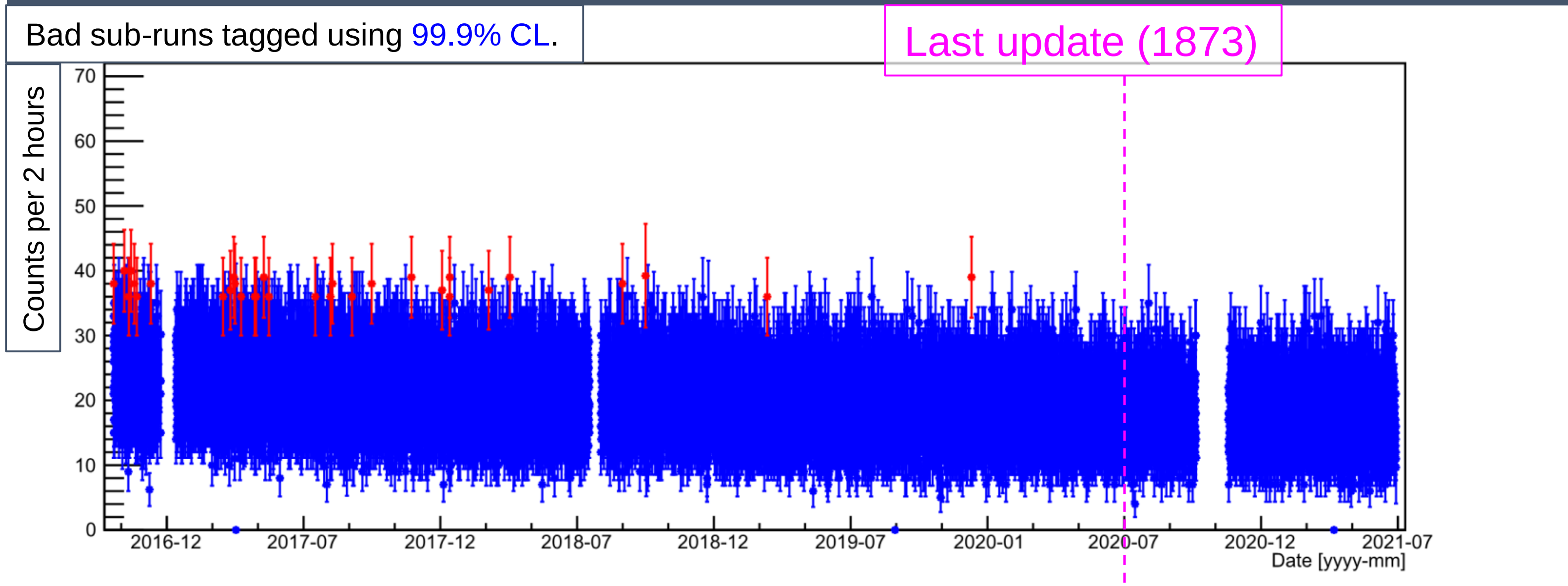




Crystal 4 Result

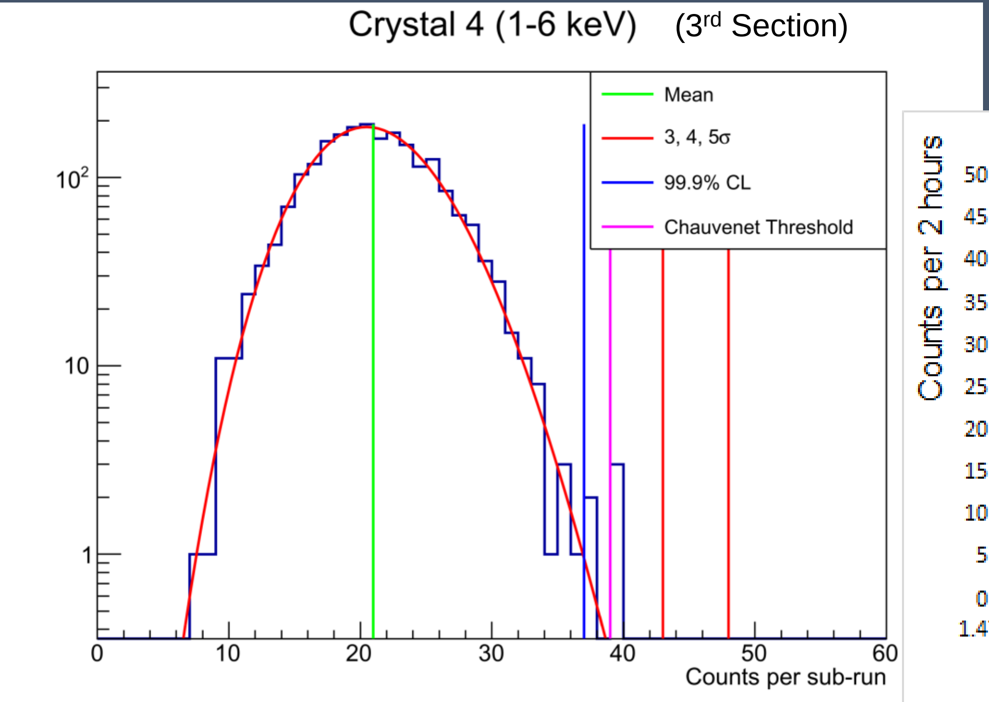
- Count is decreasing. Without considering it, most bad sub-runs were tagged at early time.

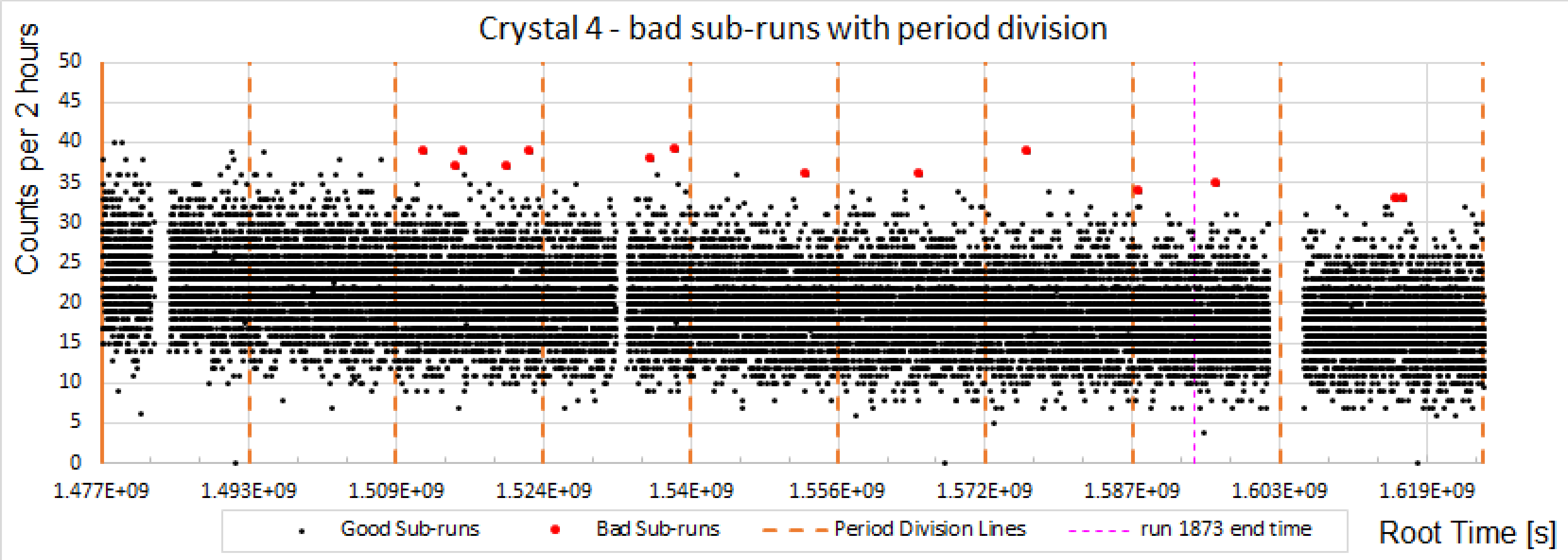




Crystal 4 Result

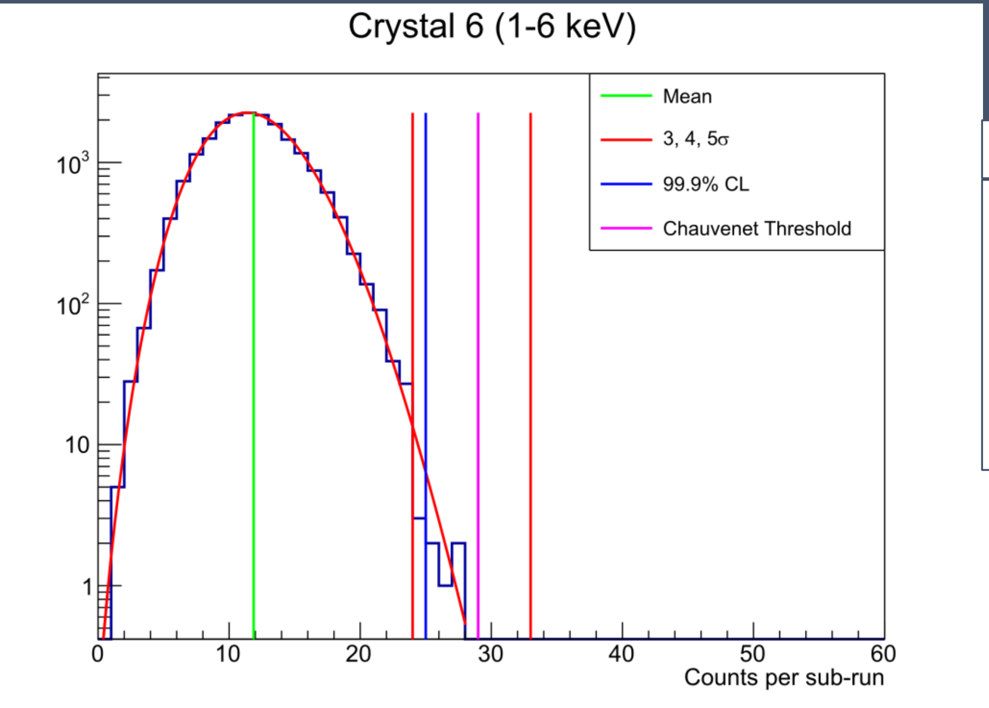
- Dividing the period by 6 months, and detecting outliers in each section, the following bad sub-runs are detected.
  - Last section contains the remaining period.

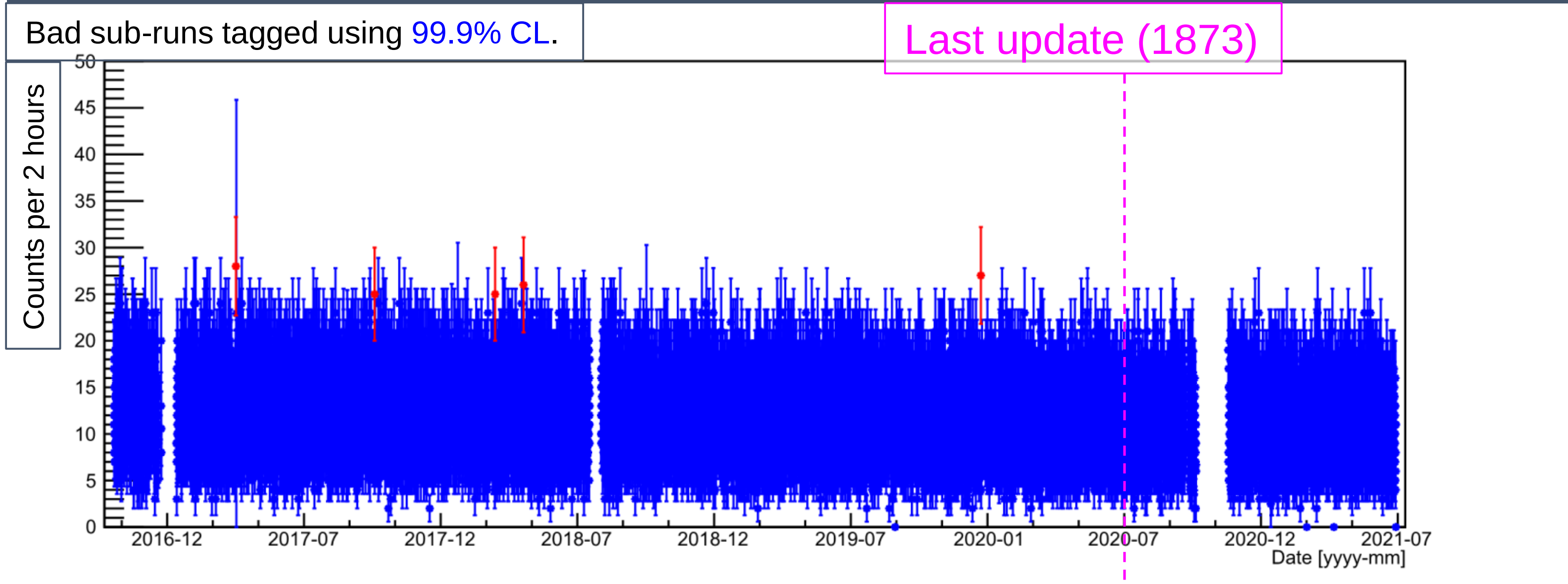




Crystal 6 Result

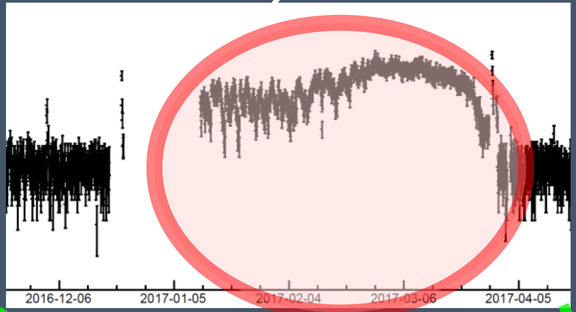
- Distribution looks good. Some sub-runs were tagged.

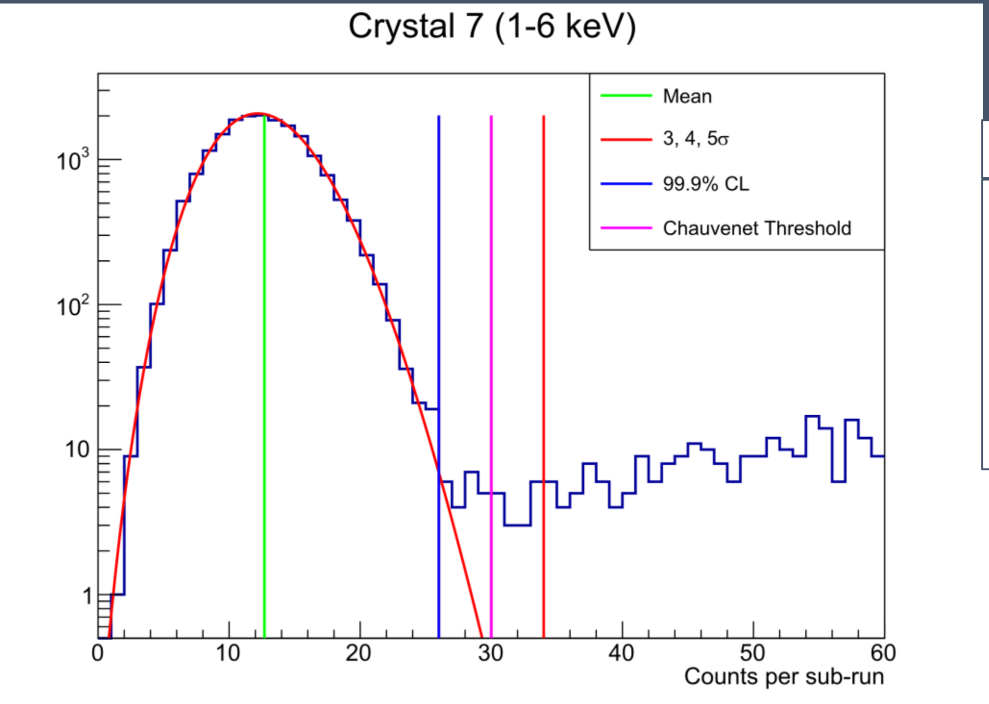


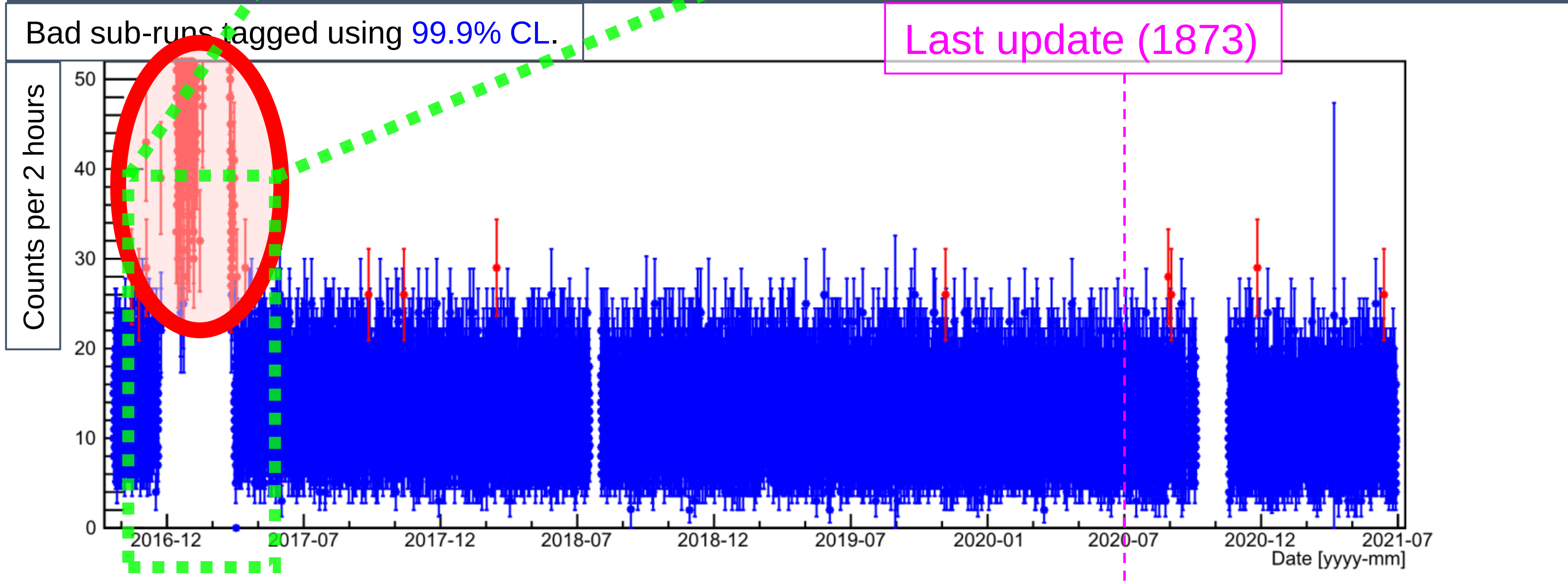


Crystal 7 Result

- Long term instability from 16-12 (1555.000) to 16-04 (1625.008).

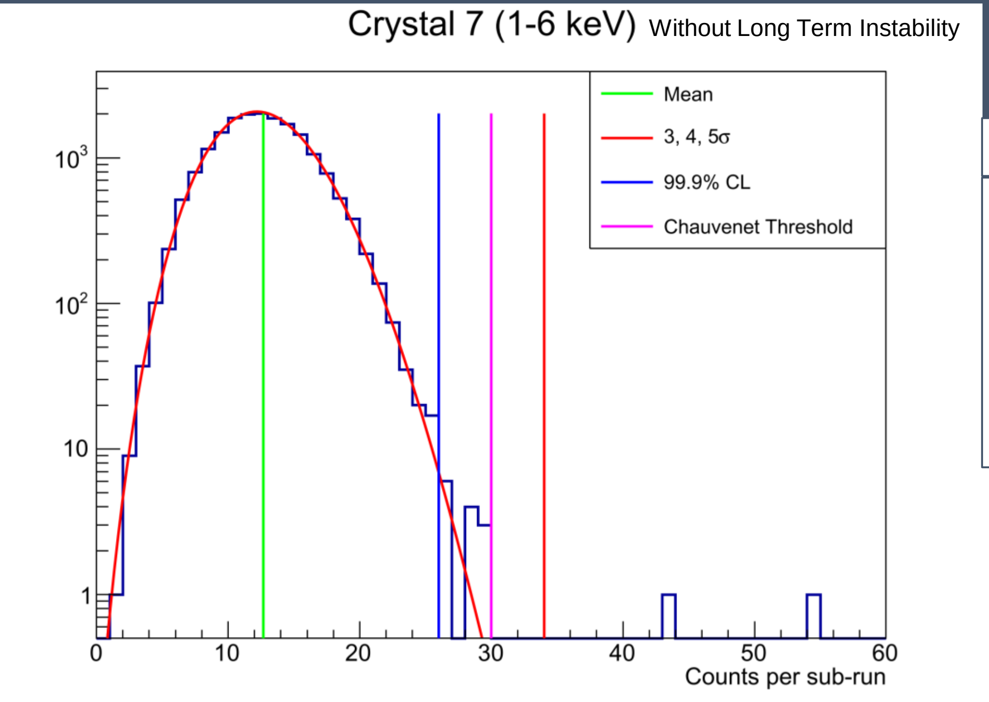


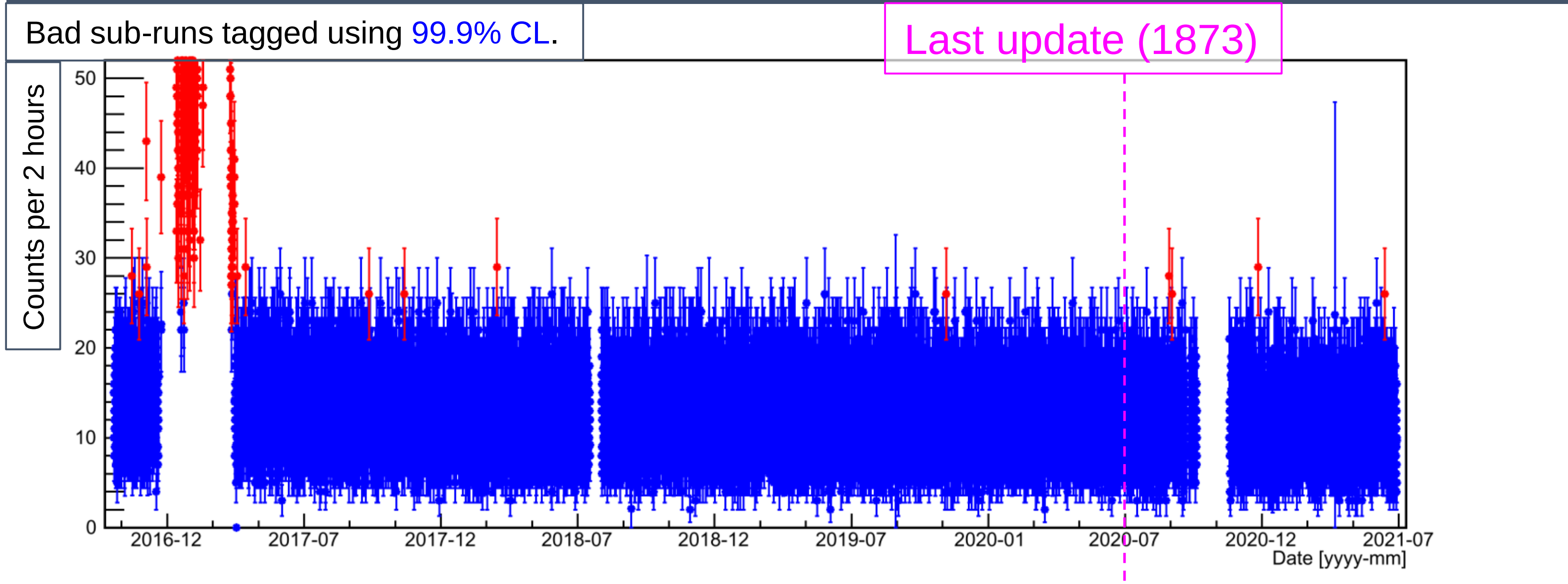




Crystal 7 Result

- Long term instability from 16-12 (1555.000) to 16-03 (1625.008).
- Removing that instability, we get the following well-fitted plot.
- To be sure for noise rejection, 99.9% CL was adopted.





Bad Sub-run Candidates

Crystal 2

Run	Sub-runs	Run	Sub-runs
1544	122-123 260-281 507-515	1718	837
		1862	366-370 382 453
1546	000-002		
1626	026	1863	099
1671	138 293	**1873**	**061-410** (long term)
1672	216 324	1875	252
		1919	632

Crystal 3

Run	Sub-runs	Run	Sub-runs
1616	088 358	1718	060
1634	006	1719	037 763
1652	756	1858	046
1672	802	1859	452
1678	230 612	1868	704
		1935	996

Crystal 4

Run	Sub-runs	Run	Sub-runs
1678	072 562 686	1859	942
		1861	619
		1863	224
1718	424 764	1873	173
		1875	339
1771	346	1935	684 789
1777	229		

Bad Sub-run Candidates

Crystal 6

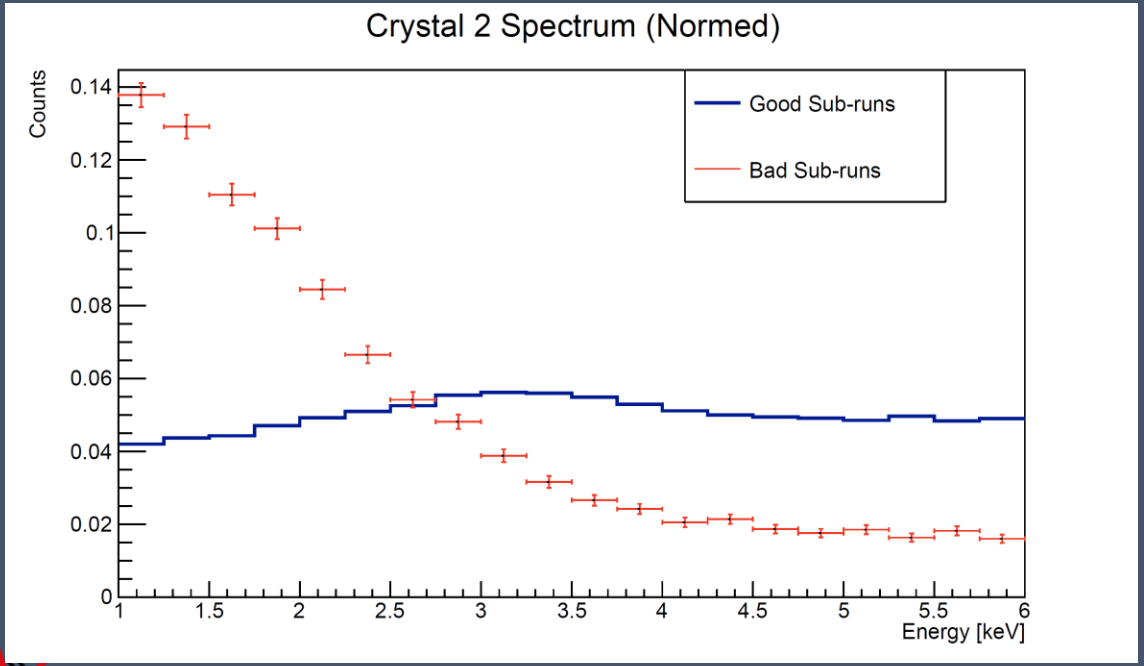
Run	Sub-runs	Run	Sub-runs
1626	018	1719	030
1672	436	1865	055
1718	522		

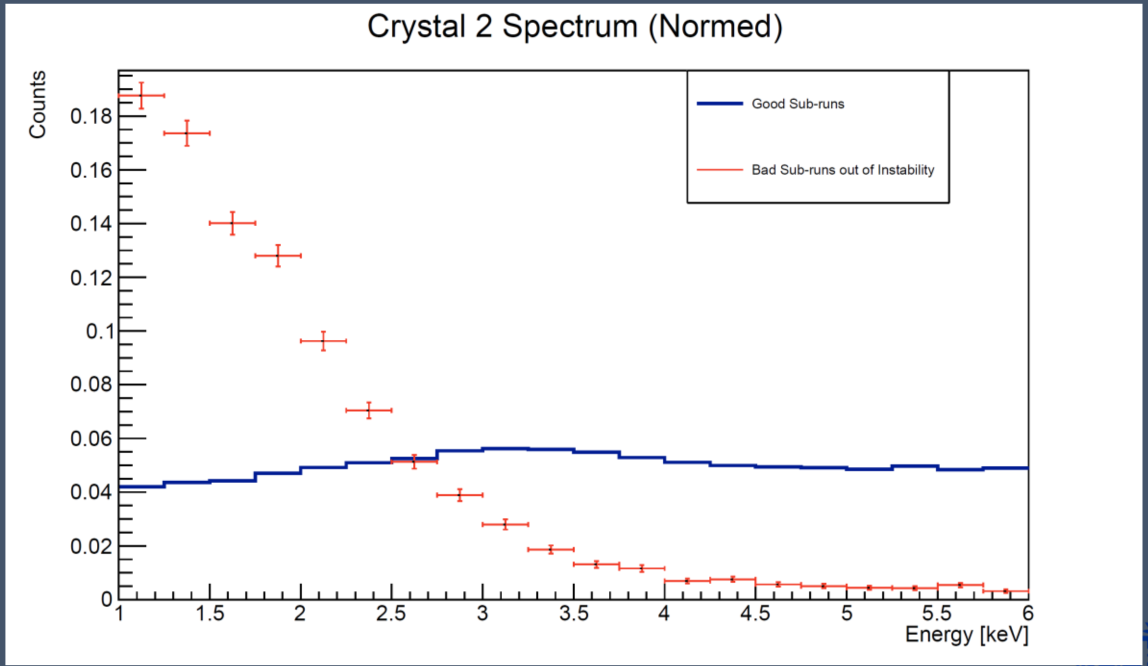
Crystal 7

Run	Sub-runs	Run	Sub-runs
1544	288	1627	002
	406	1652	002
	521-522	1672	343
	526		909
1555	*	1718	548
1616	*	1862	806
1617	*	1875	650
1618	*		699
1625	*	1919	462
(long term)		1939	798

Spectrum Comparison Crystal 2

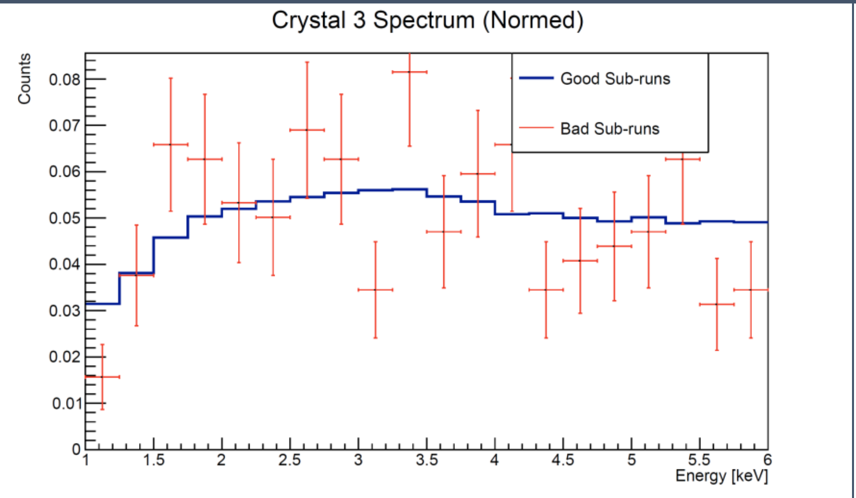
- Spectrum of bad sub-run candidates in crystal 2 is very noisy.
- Those sub-runs were tagged as bad.

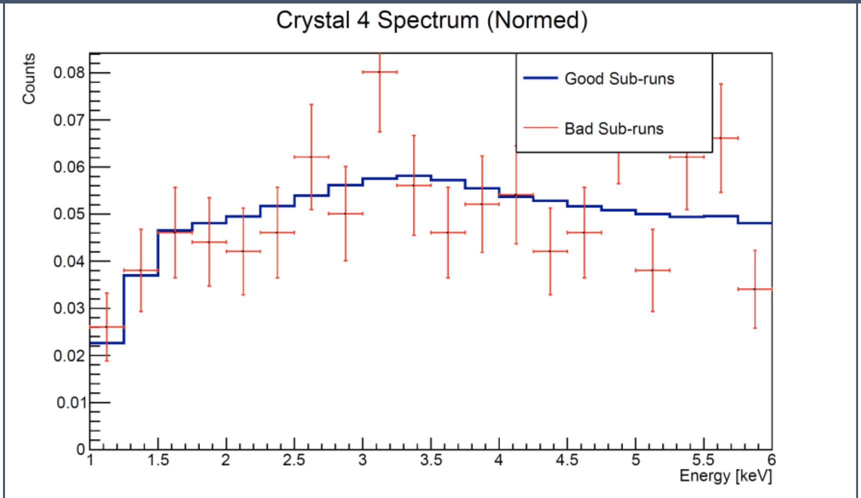


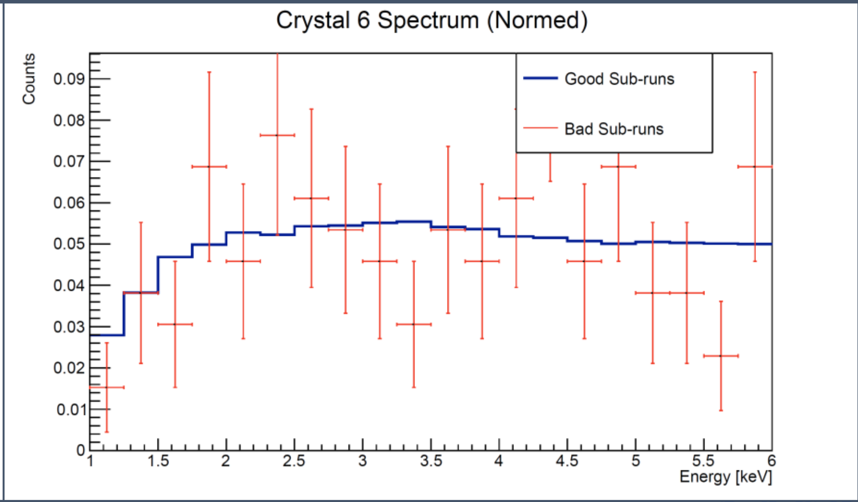


Spectrum Comparison Crystal 3, 4, 6

- For crystal 3, 4 and 6, spectrums of bad sub-run candidates show no significant difference from good sub-runs.
  - These crystals don't have any clear long term instability.
- No bad sub-runs were tagged for those crystals.

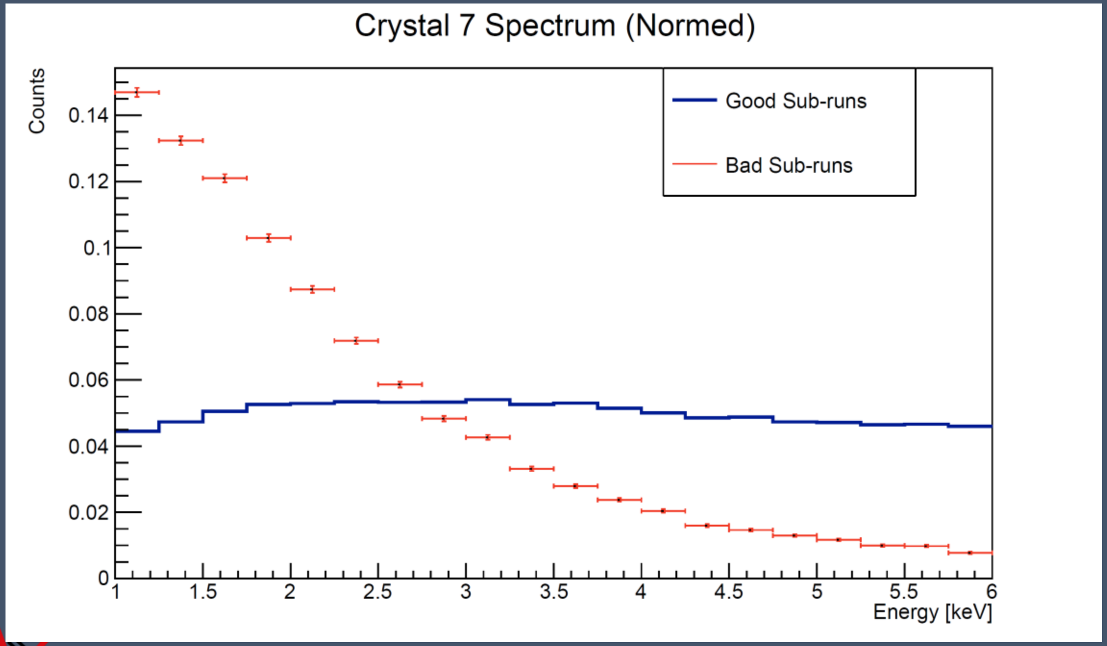


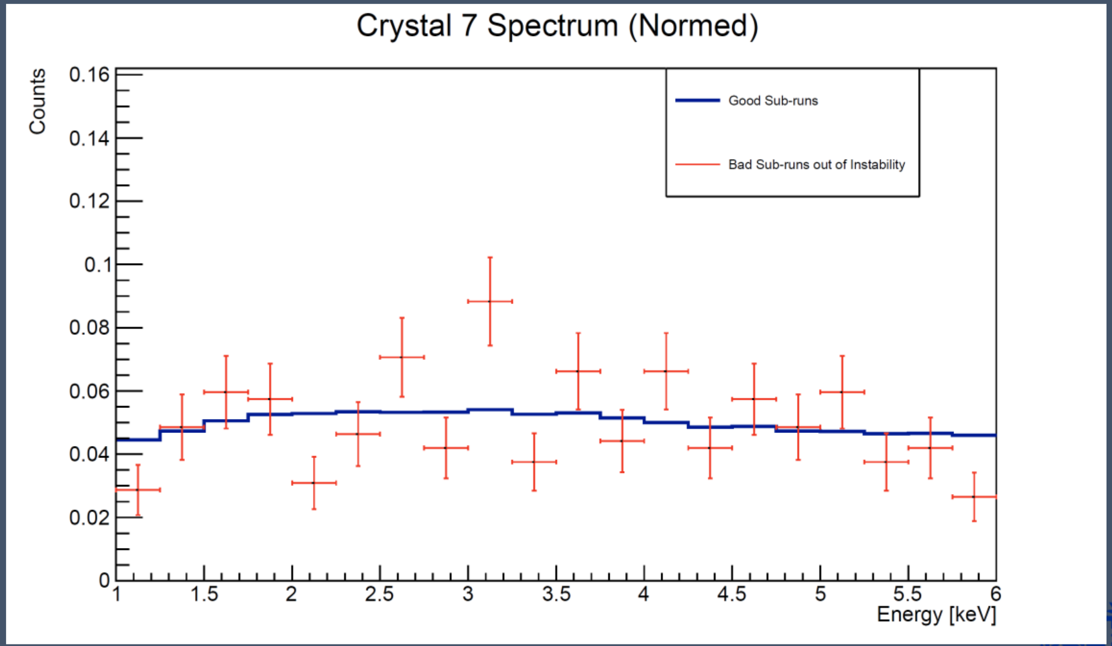




Spectrum Comparison Crystal 7

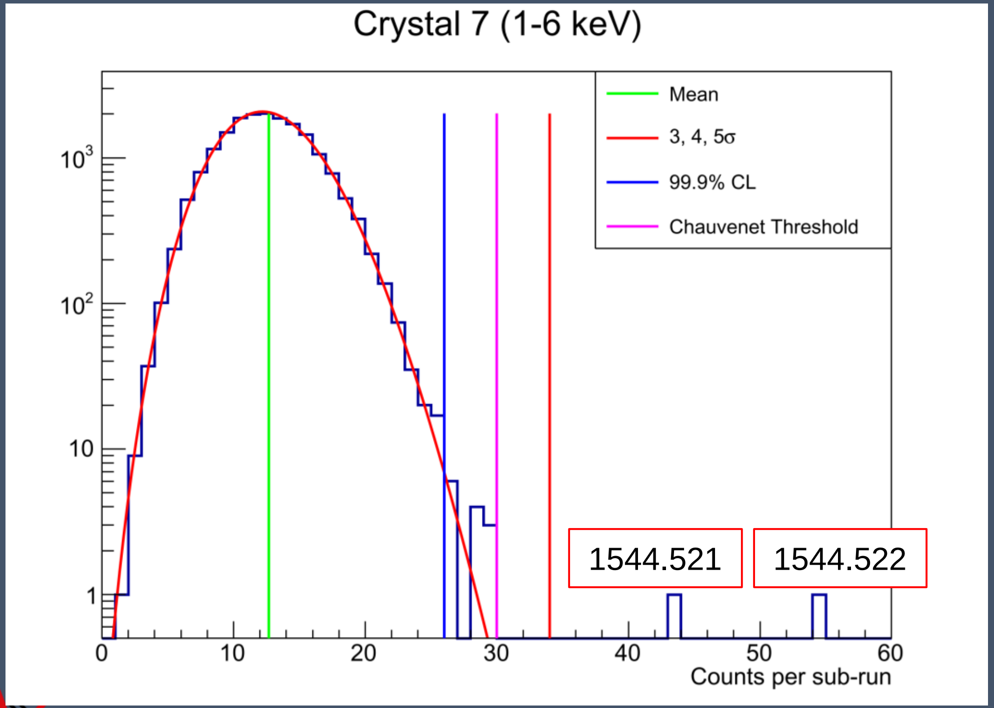
- Spectrum of long term instability in crystal 7 is very noisy.
- Bad sub-run candidates out of the instability showed no significant difference from good sub-runs.
- Only the long term instability in crystal 7 was tagged as bad.

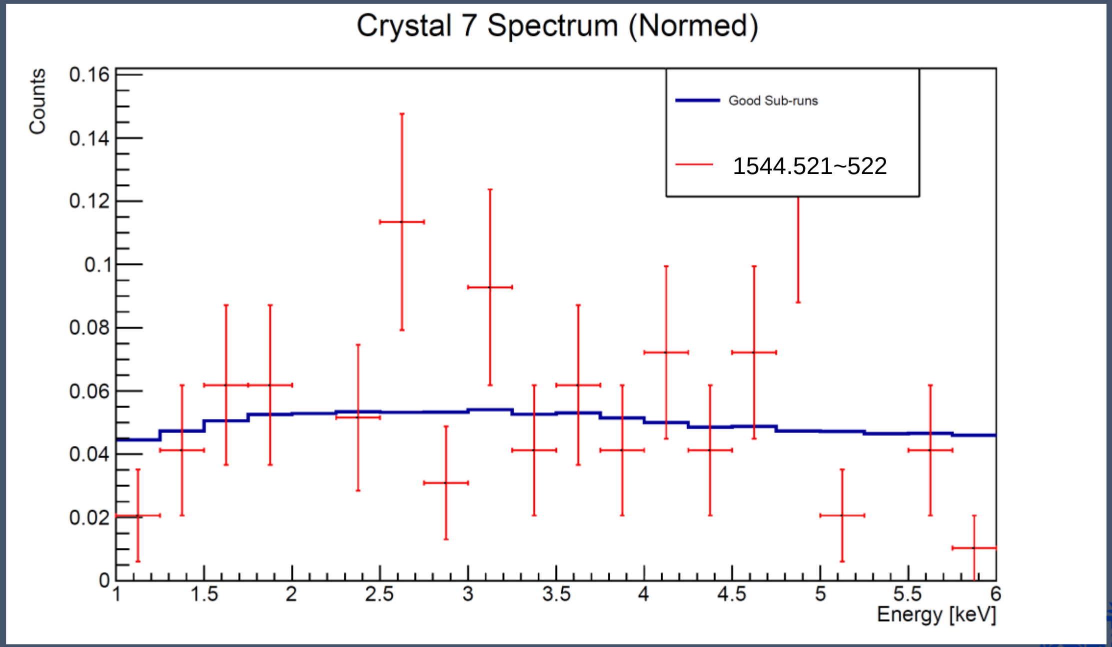




Spectrum Comparison Crystal 7 – 1544.52#

- Two sub-runs in 1544 had very high event rate.
- But they also did not show some problematic spectrum.





Conclusion Bad Sub-run List

- The bad sub-runs list for V00-04-19, run ~1939 was determined as right tables.
- Crystal 4 decaying component could be dealt by dividing the period by 6 months.
- Bad/good sub-runs were validated by observing their energy spectrums.
- The preliminary version of database is now prepared.
  - https://cosine.cup.re.kr/app2/offline

Crystal 2

Run	Sub-runs	Run	Sub-runs
1544	122-123 260-281 507-515	1718	837
		1862	366-370
			382
1546	000-002		453
1626	026	1863	099
1671	138	**1873**	**061-410**
	293	(long term)	
1672	216	1875	252
	324	1919	632

Crystal 3, 4, 6

No bad sub-runs were tagged.

Crystal 7

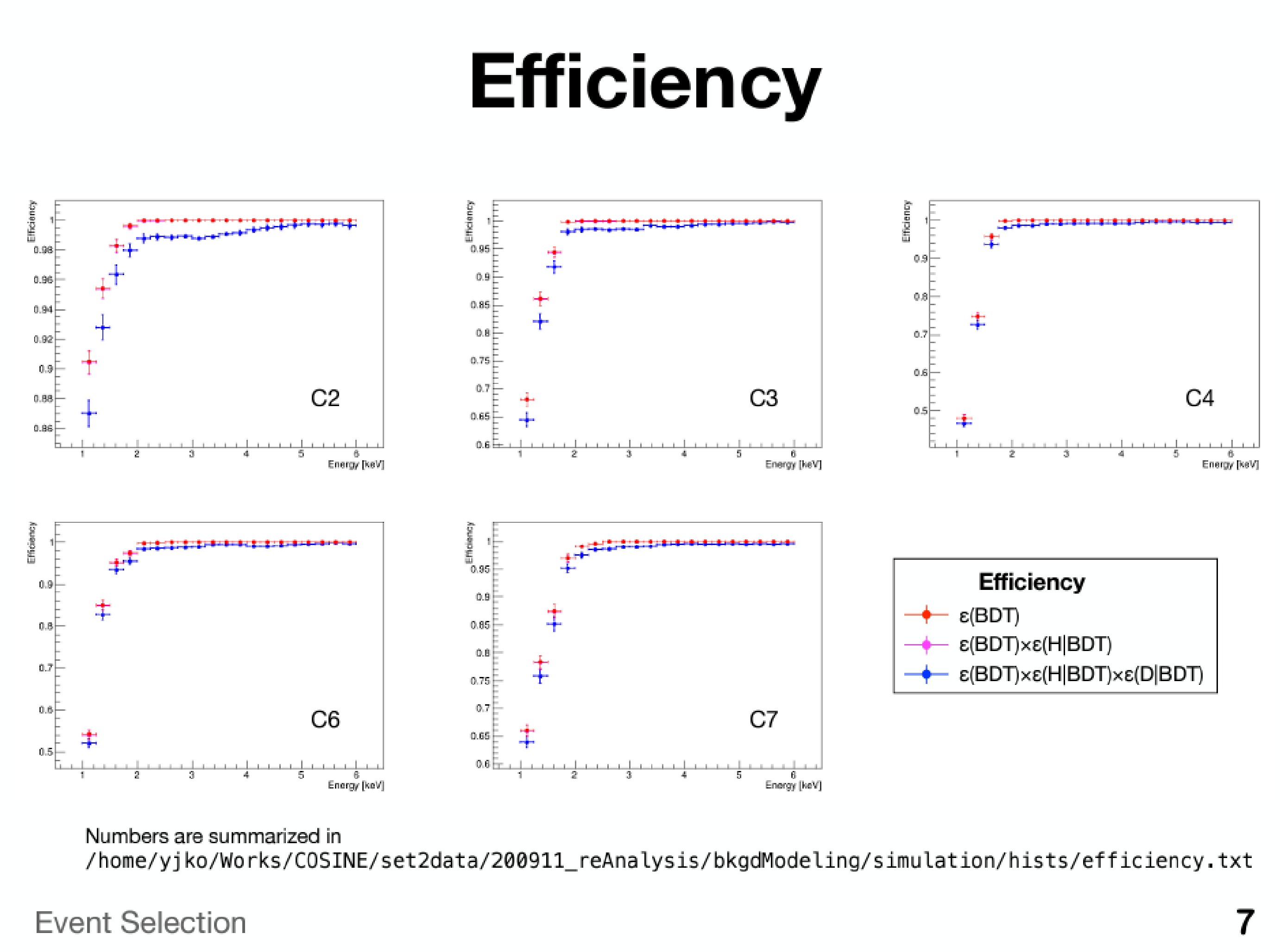
Run	Sub-runs
1555	*
1616	*
1617	*
1618	*
1625	*
(long term)	

BACK UP

Codes for Data Quality Check

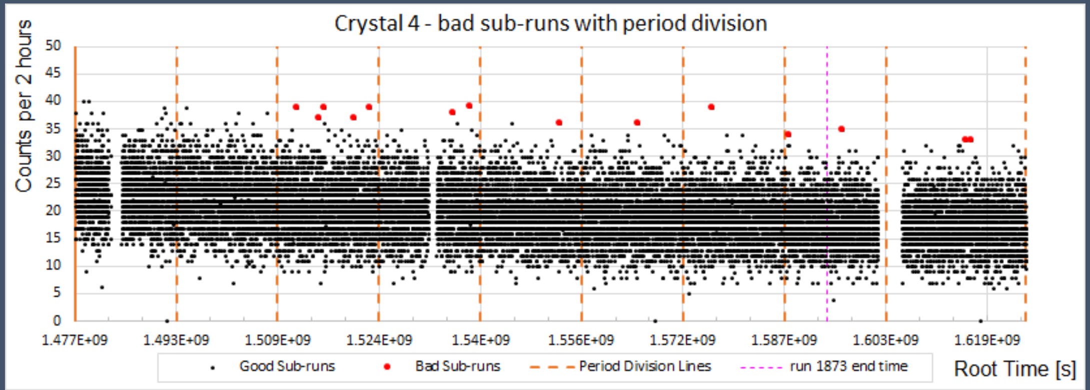
- You can find my source codes in the directory below.
  - **(Olaf): /mnt/lustre/ibs/seungmok/Share/data_quality_2021/**
  - Or (GitLab): gitlab.com/cosine100/data_quality_2021/
- Results are in the directory below.
  - **(Olaf): /mnt/lustre/ibs/seungmok/Research/2106_DataQualityCheck2/CosineDQC/**
  - ‘**plots**’ and ‘**result**’ folders are containing the plots and bad sub-run list.

New Event Selection Efficiency

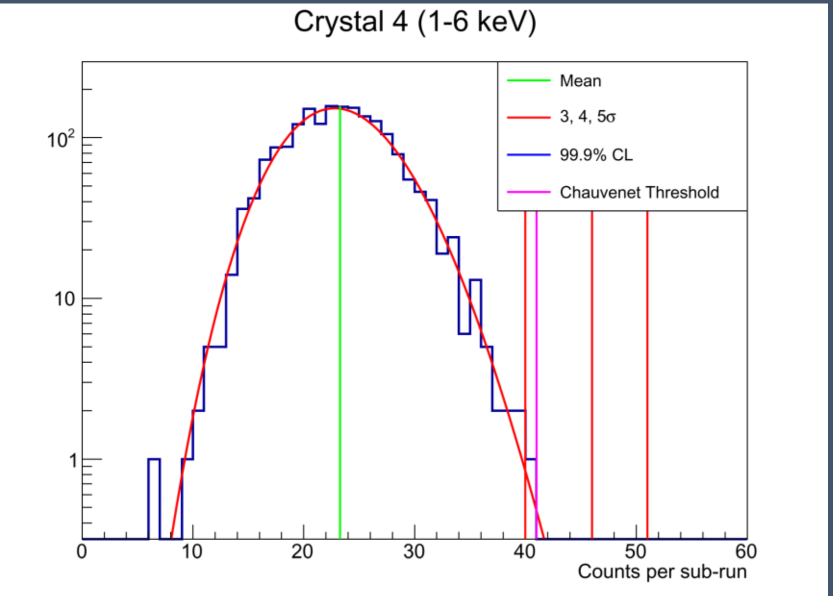


YJ Ko, Oct 21st, 2020

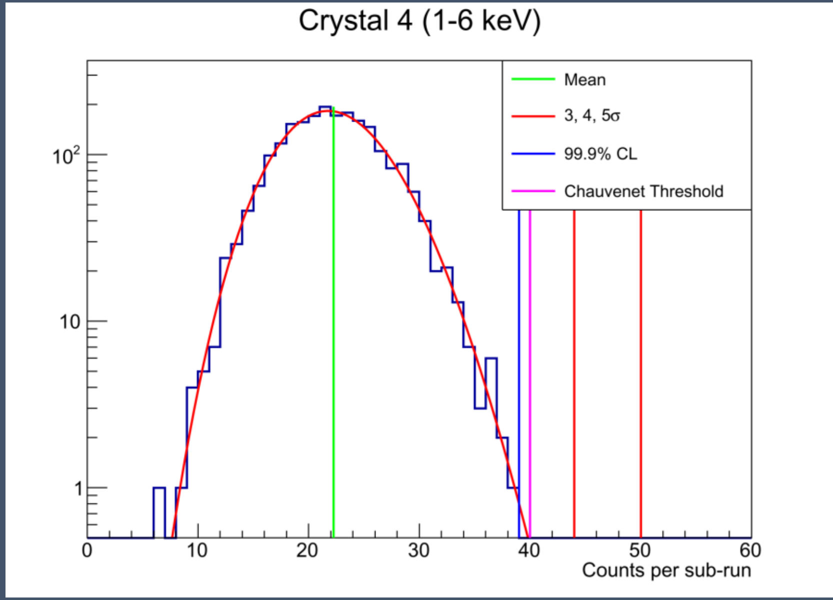
Crystal 4 Histograms



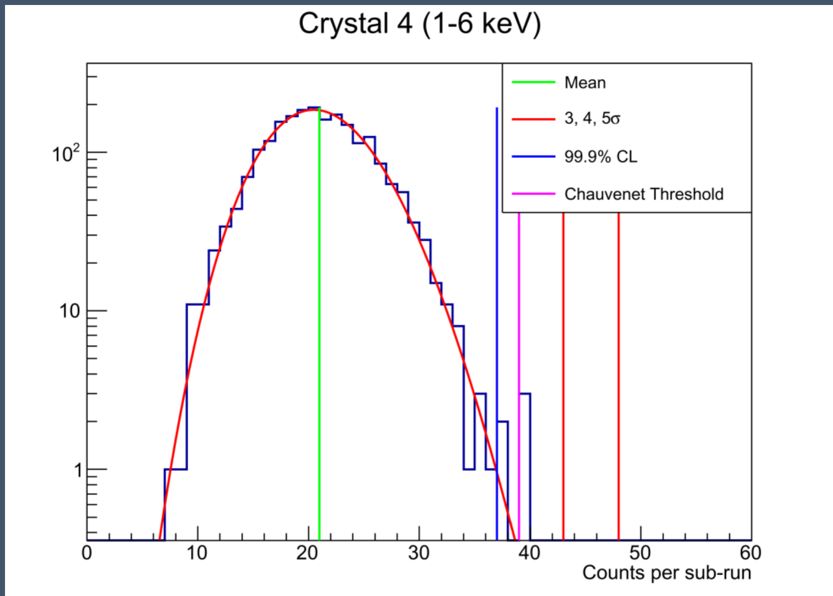
(1st Section)



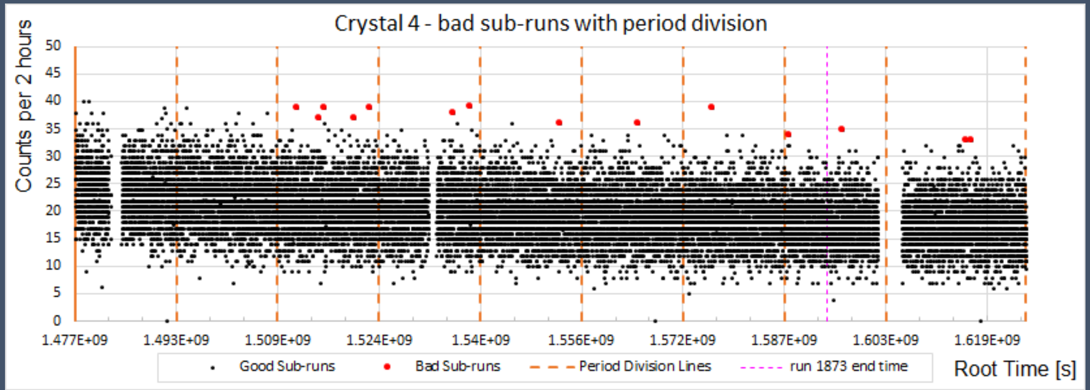
(2nd Section)



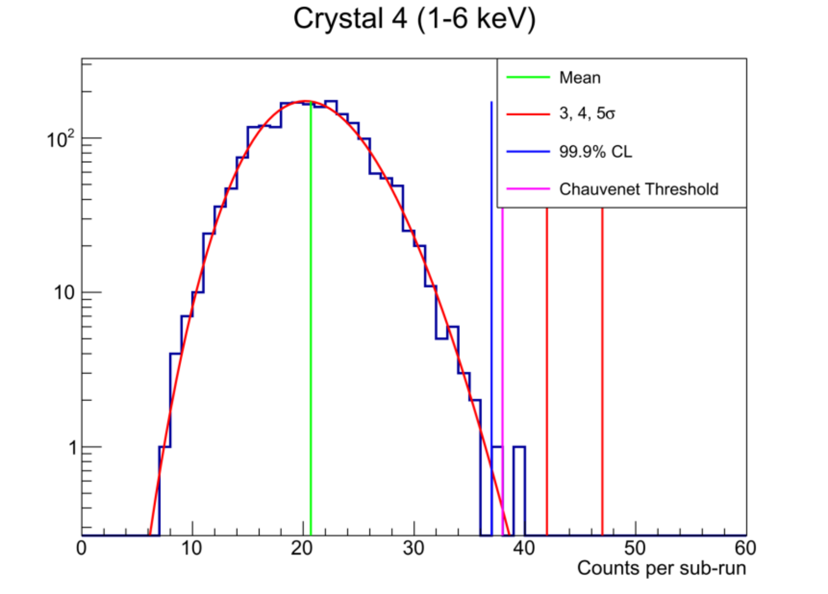
(3rd Section)



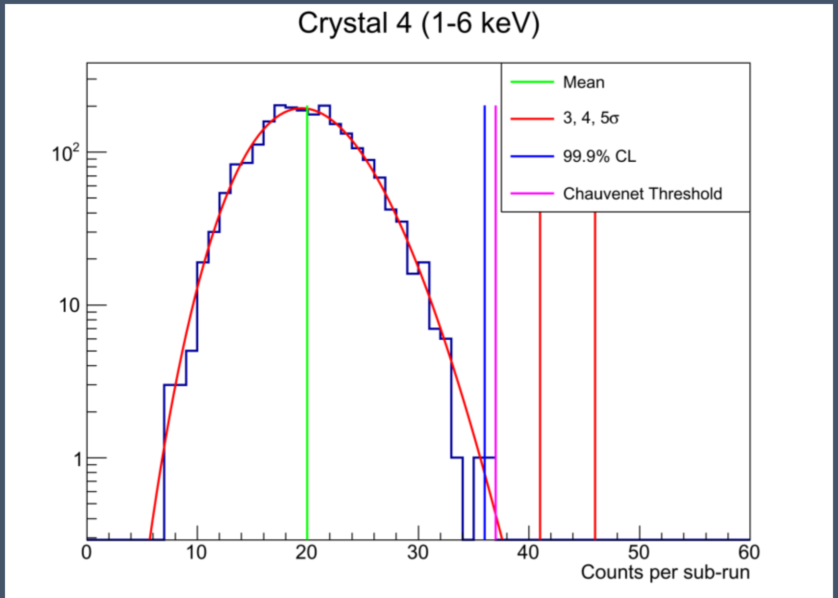
Crystal 4 Histograms



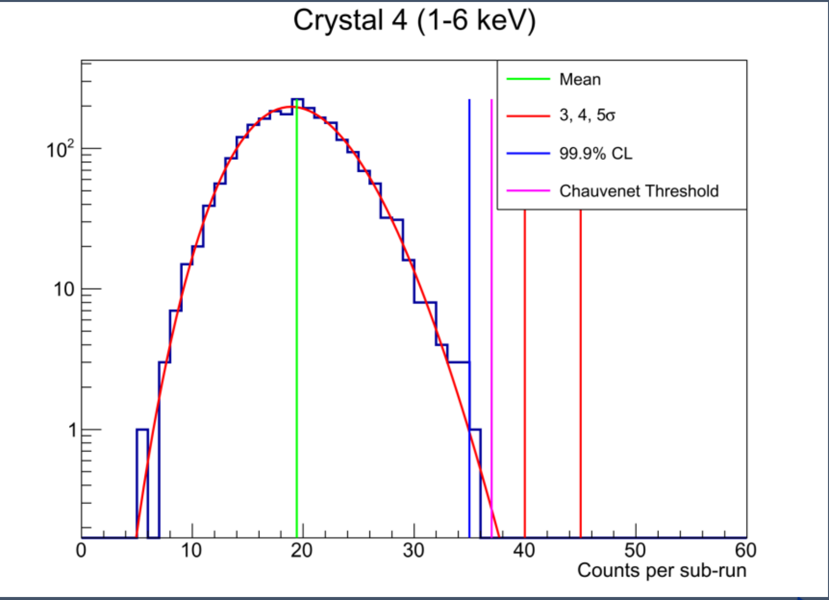
(4th Section)



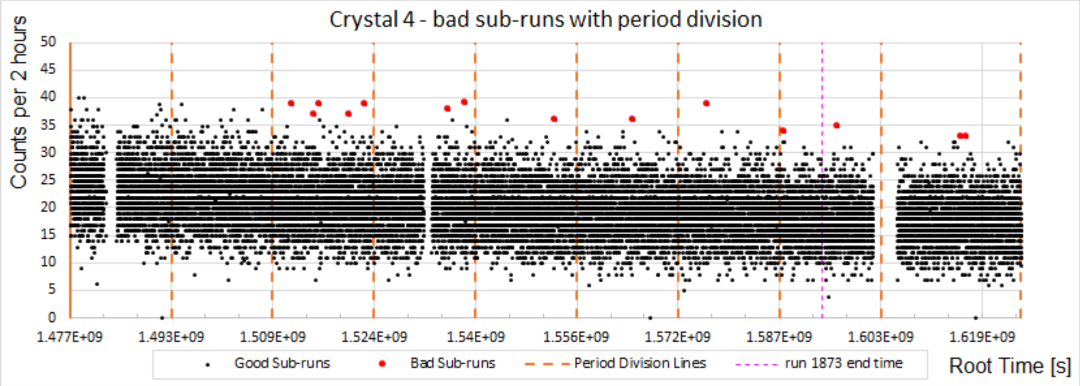
(5th Section)



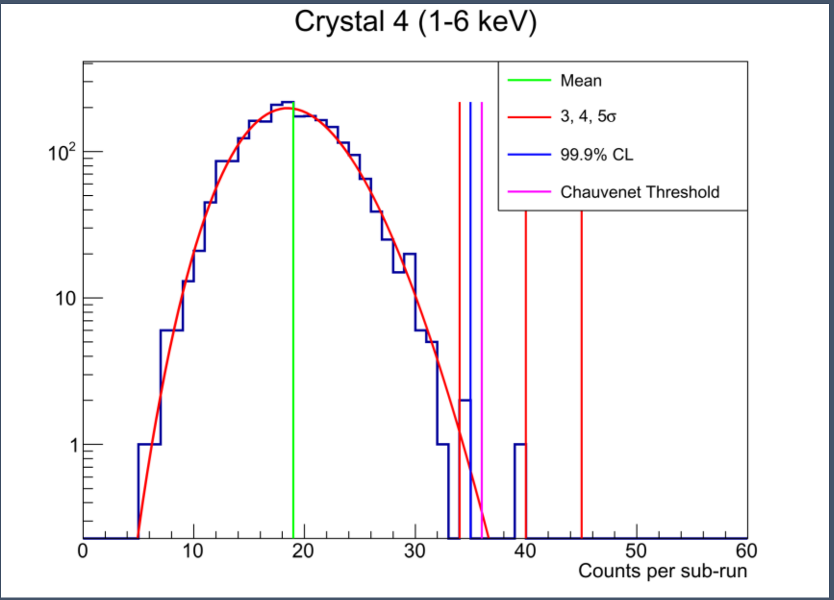
(6th Section)



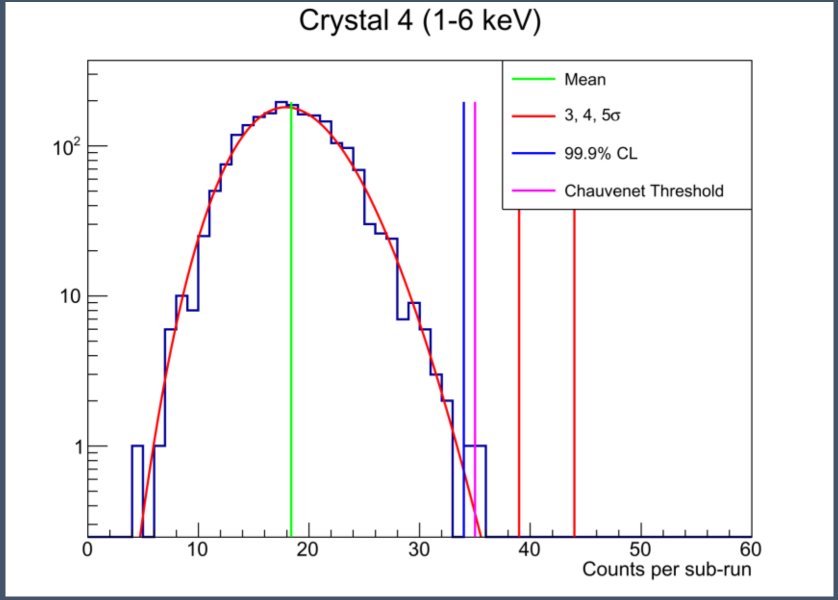
Crystal 4 Histograms



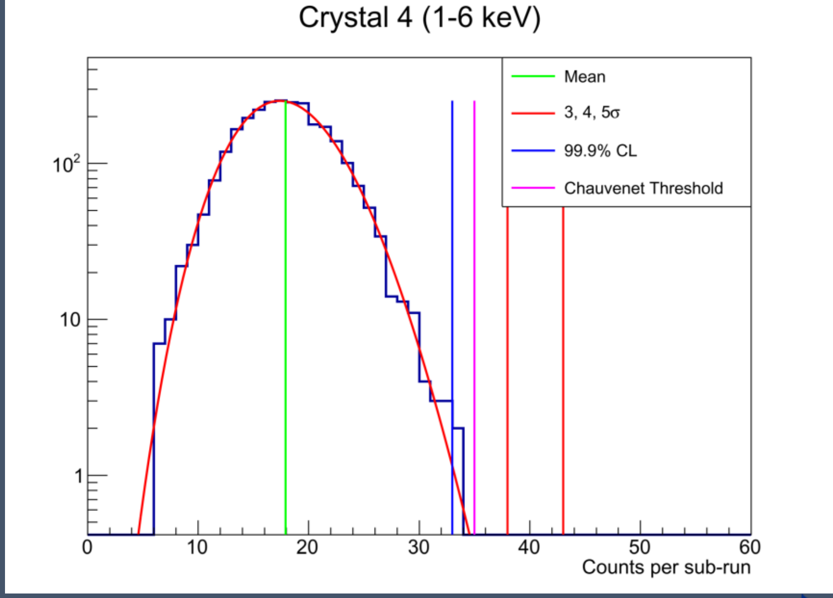
(7th Section)



(8th Section)



(9th Section, Last)



LAST SLIDE