FXT Event-by-event Vertex Correction
[and more…]

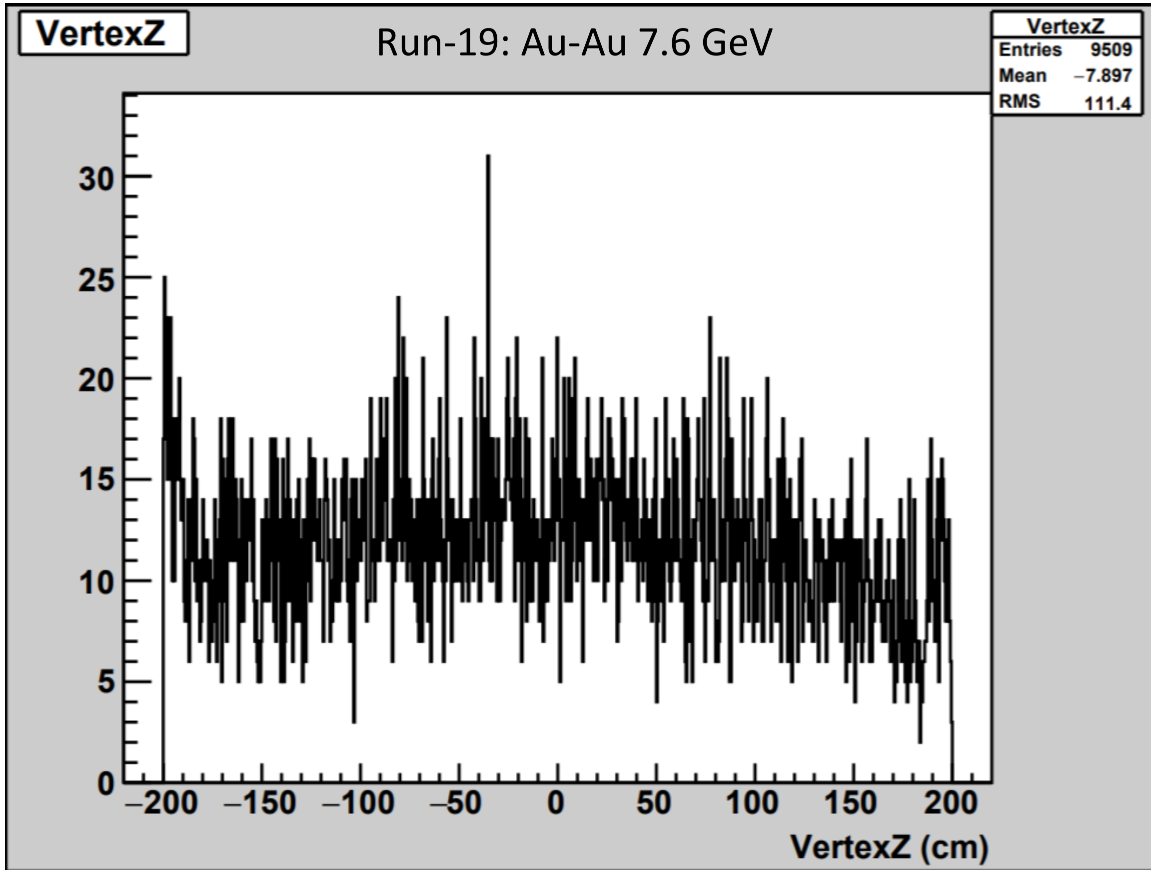
16 February 2021

Irakli Chakaberia

Lawrence Berkeley National Laboratory

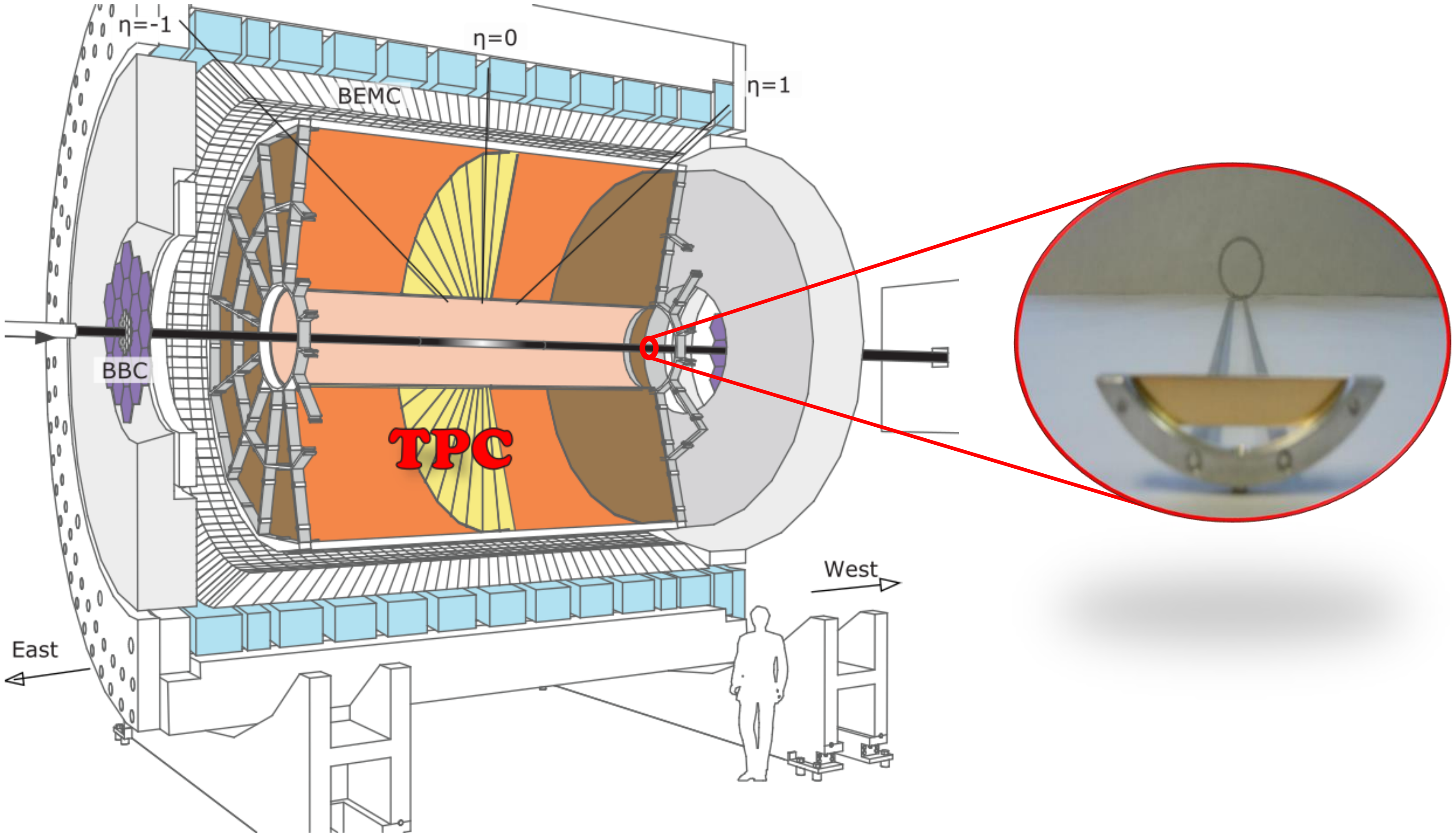
Overview

- BES-II is pushing the limits of the RHIC to provide very low energies while keeping the intensities as high as possible
- The solution is optimizing bunch parameters to have high rate of collisions while keeping background reasonably low
- As a result, we get very long bunches and collisions smeared almost uniformly along the entire TPC
- It is dealt with on the level of the data-analysis, but it poses issues on the level of the level of reconstruction and calibration as well



STAR FXT Program

- Single beam hitting the target located on the west side of the STAR detector

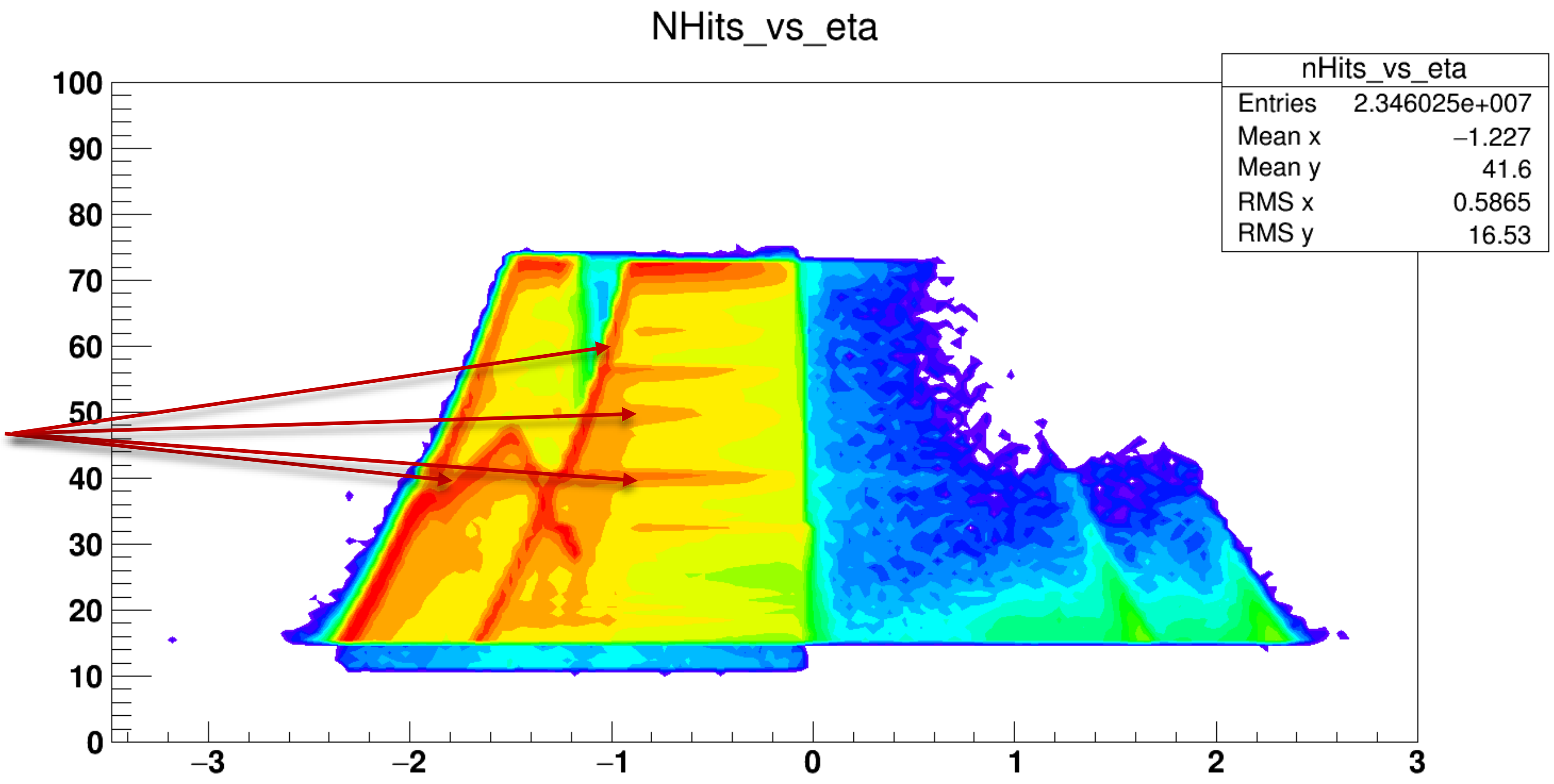


Introduction

- In Run-19 the problem with the FXT distributions became obvious

- Symptoms looked similar to those of wrong T0 for the TPC

- Till Run-19 there has been no practice of introducing additional T0 for FXT runs, but BES-II made it necessary

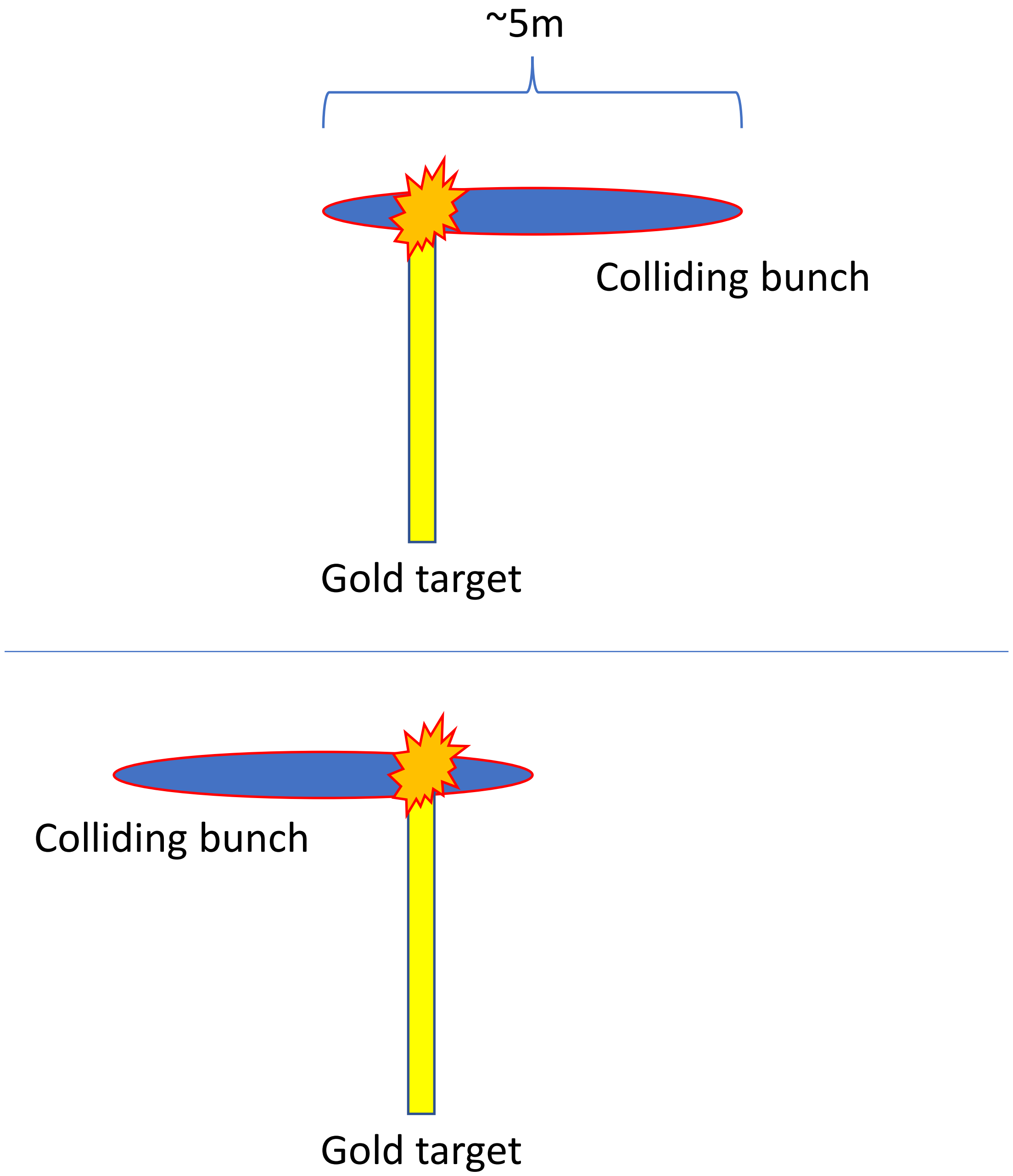


Description of timing

- There are many parameters related to the timing of the event in the STAR detector
- Global Time: start time of the event / trigger that happened in the middle of the TPC
- This time is set by the “overlords of the run” to have vertex centered and is based on the energy of the beam
- TPC T0 for each sector: this is the “average” time measured by the TPC relative to the global time for every sector based on the “prompt hits”

Timing Offset

- The global timing setup for the detector is based on the beam energy
- It is therefore same for the FXT and Collider modes for that Energy
- This introduces the `2 m/c=~7 ns` shift in timing, which translates into `~0.7 mm` in vertex separation
- But for some events larger separation was observed which means there is additional distance/time to consider
- Idea: it is due to the collision happening not in the middle of the very long bunches and not always at the same place
- This idea was tested, with the EPD in mind as a good candidate to provide timing of the collision

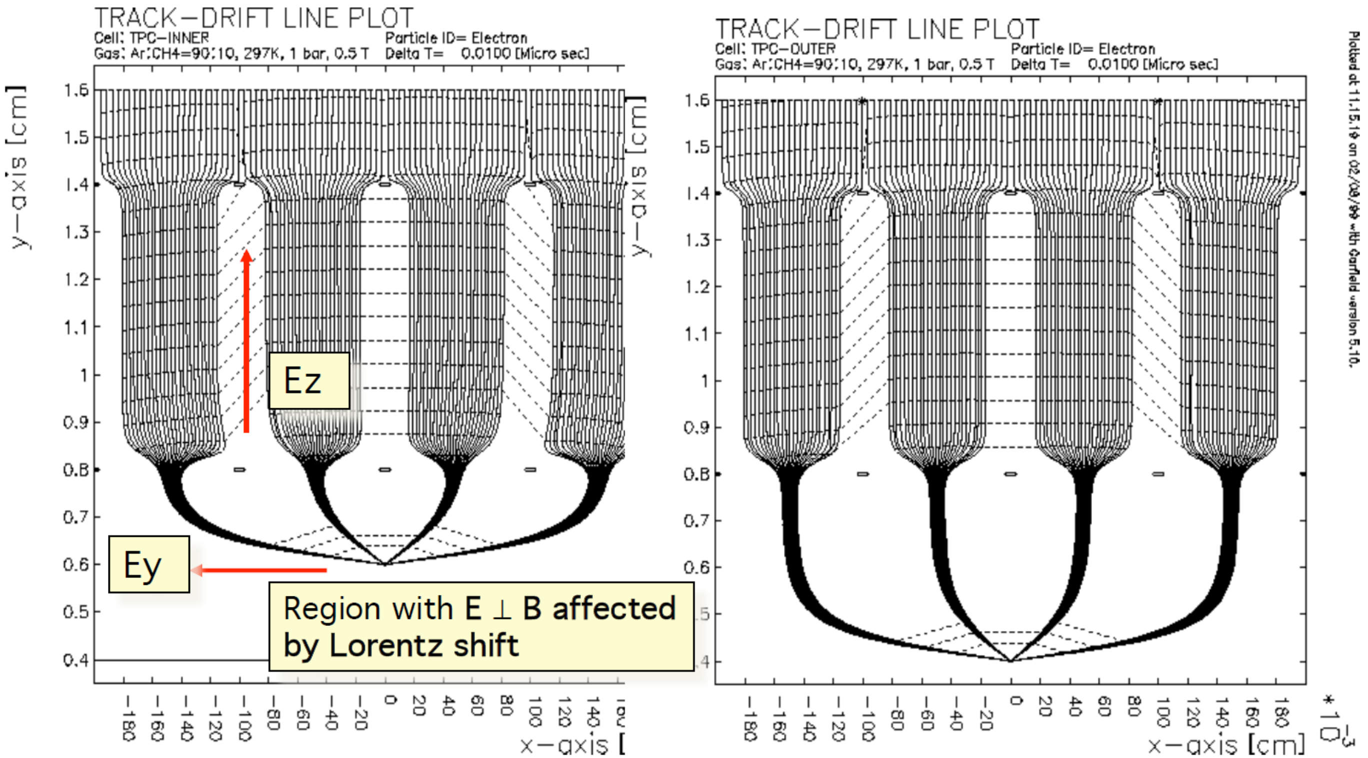


Prompt Hits

- Ionization right around the anode wires is registered as the prompt hit
- Timing of this hit is used as correction to the global time
- The geometry of the inner and outer sector are different therefore collection time of the ionization is slightly different
- This difference in time between inner and outer sectors is obtained from the GARFIELD simulation
- There is only one timing based on collision data for the detector setup of that Run

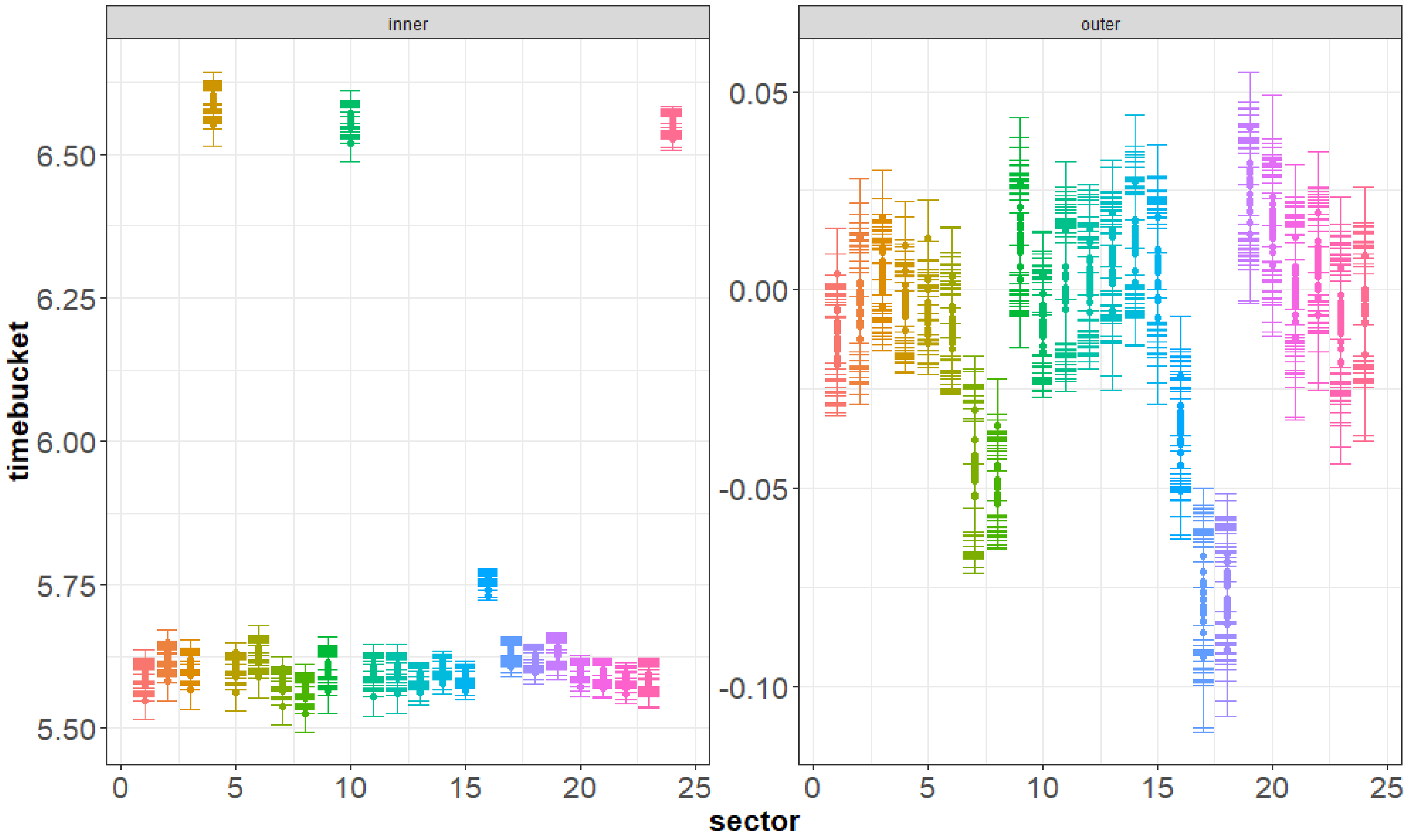
tpcSectorT0Offset

entryTime	beginTime	endTime
2021-02-08 01:12:30	2018-12-15 00:00:00	2037-01-01 00:00:00
2020-04-27 21:06:48	2019-01-01 01:00:00	2037-01-01 00:00:00
2021-02-08 01:12:15	2019-11-15 00:00:00	2037-01-01 00:00:00
2021-02-08 01:13:46	2019-11-20 18:00:00	2037-01-01 00:00:00
2021-02-08 01:11:55	2020-12-15 00:00:00	2037-01-01 00:00:00
2021-02-08 01:13:34	2020-12-20 00:00:00	2037-01-01 00:00:00

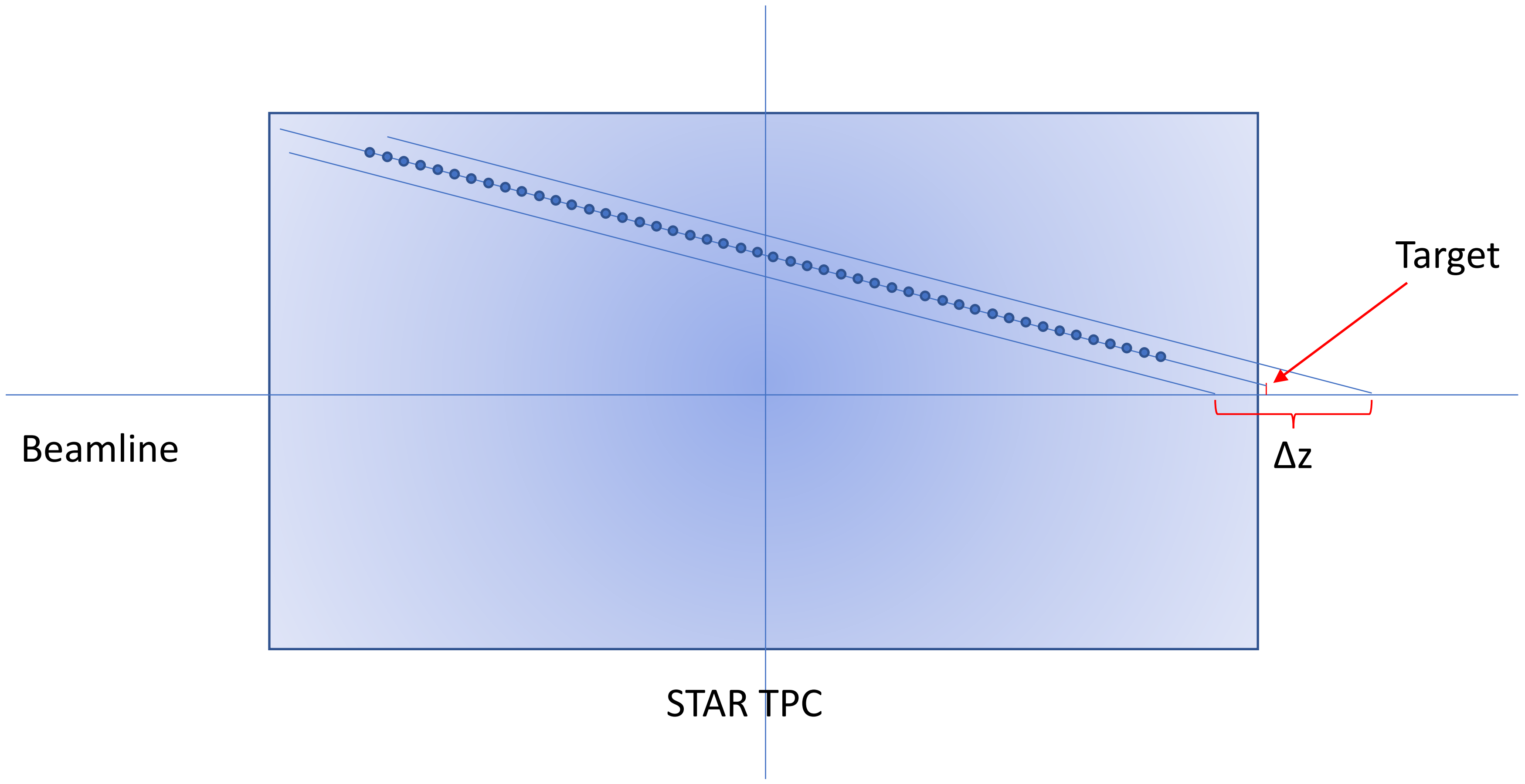


Prompt hits T0 offset for the run

- These plots show the final "corrected" timing for the prompt hits during the 19.6 GeV run of the Run-19
- At this point the differences between electronics are also taken into consideration (you can see some sectors having a timebucket offset in the readout in the iTPC)
- These times are pretty stable throughout the run

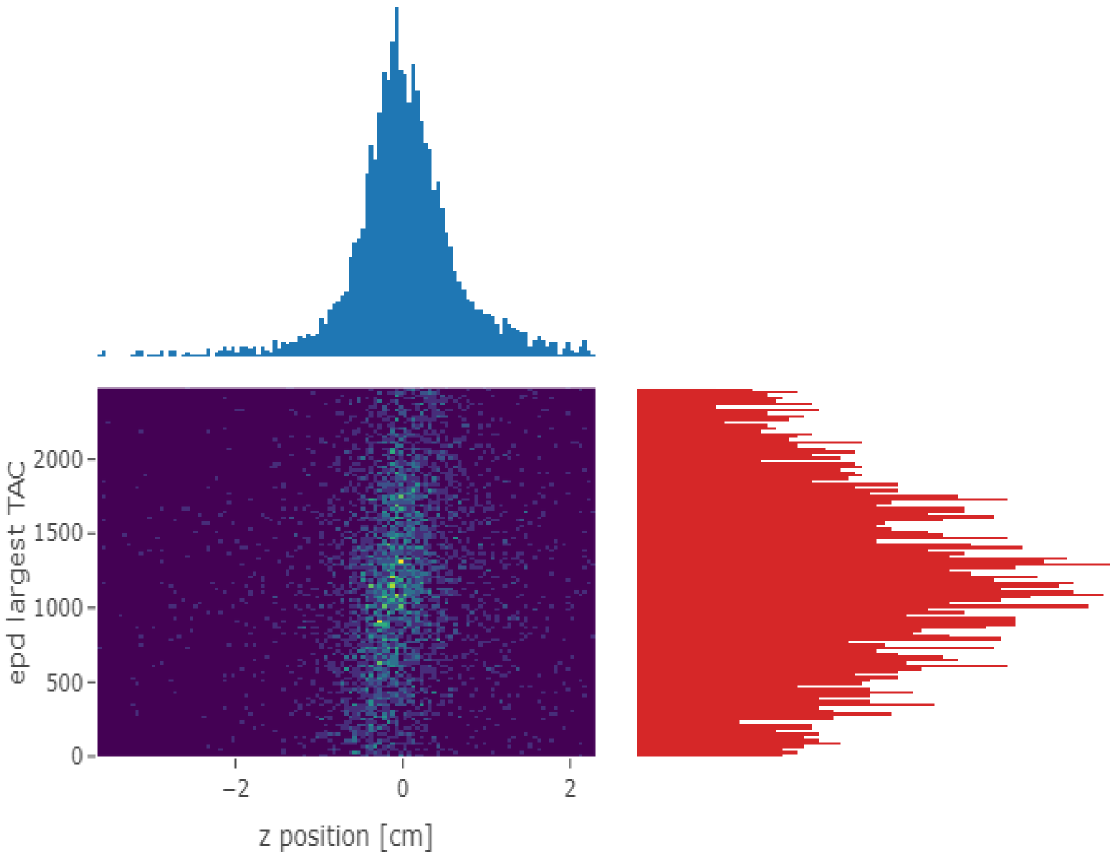


Effect of the T0



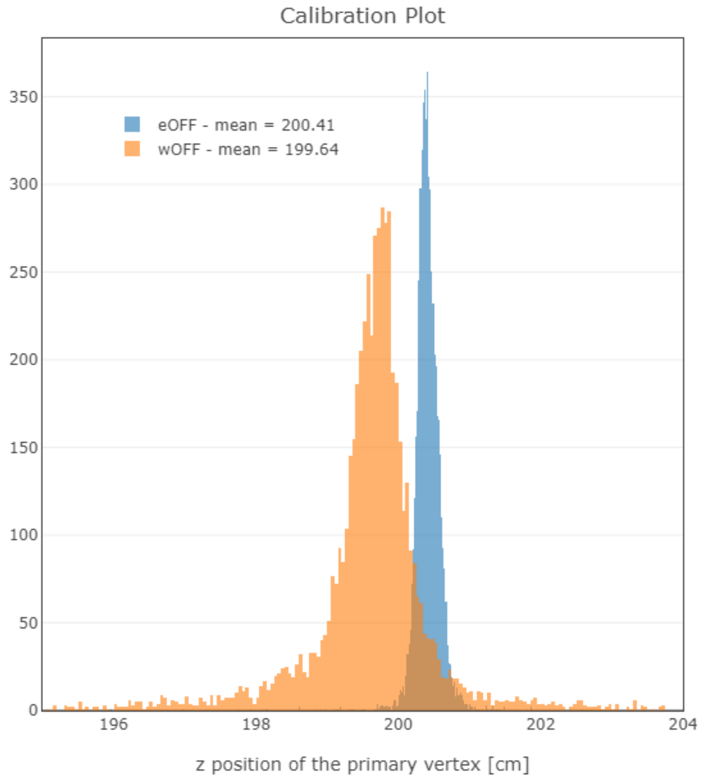
Event-by-event correction

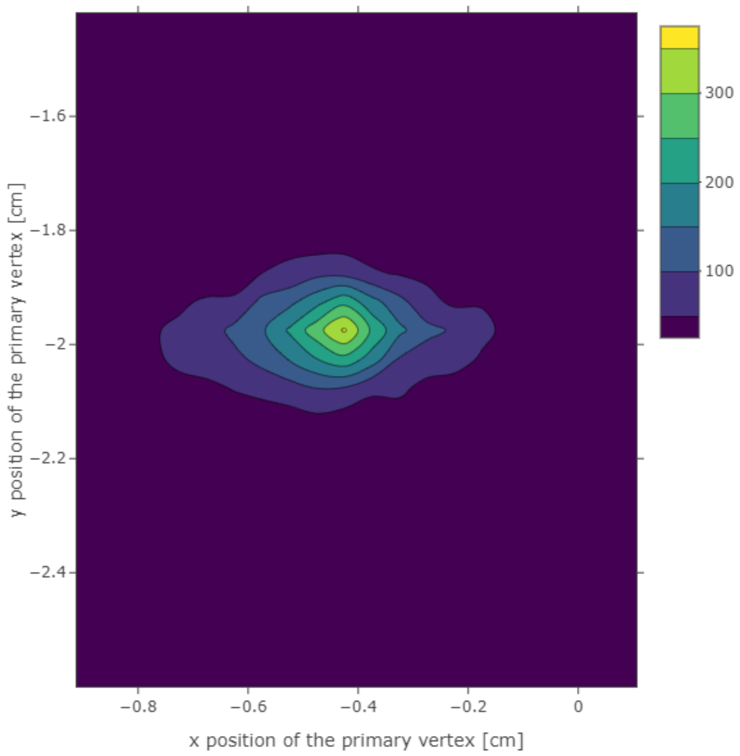
- We need proper time of every event
- ZDC and BBC do not provide a reliable information for the FXT mode, but EPD does
- Event timing is obtained from the EPD readout (side away from the target)
- The first hit registered I the EPD (in the allowed window and selection) could be considered a good measure of the event-by-event timing offset
- The vertex can be reconstructed from the east and west TPC hits separately, therefore giving us the measure of the vertex separation
- EPD timing showed a correlation with the offset between the vertices that we used for the calibration

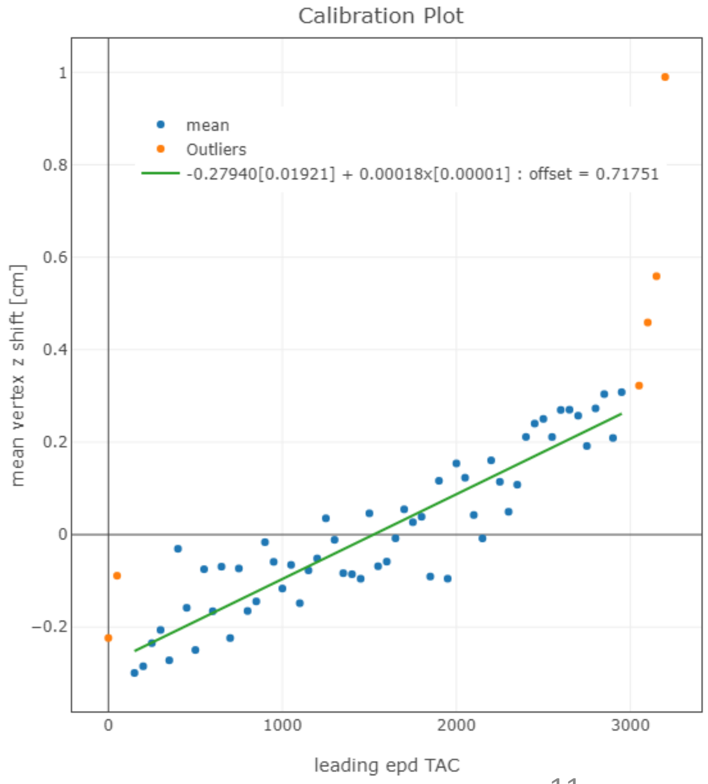


Before the Calibration

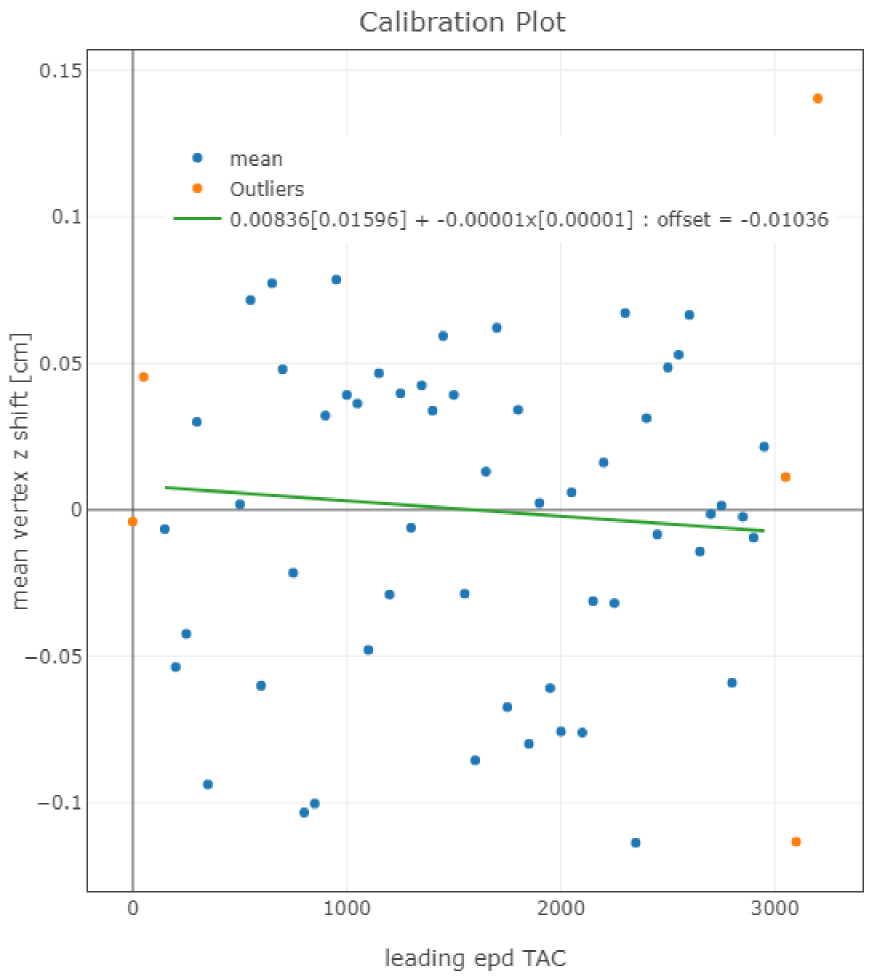
- Calibration is done for every FXT energy separately

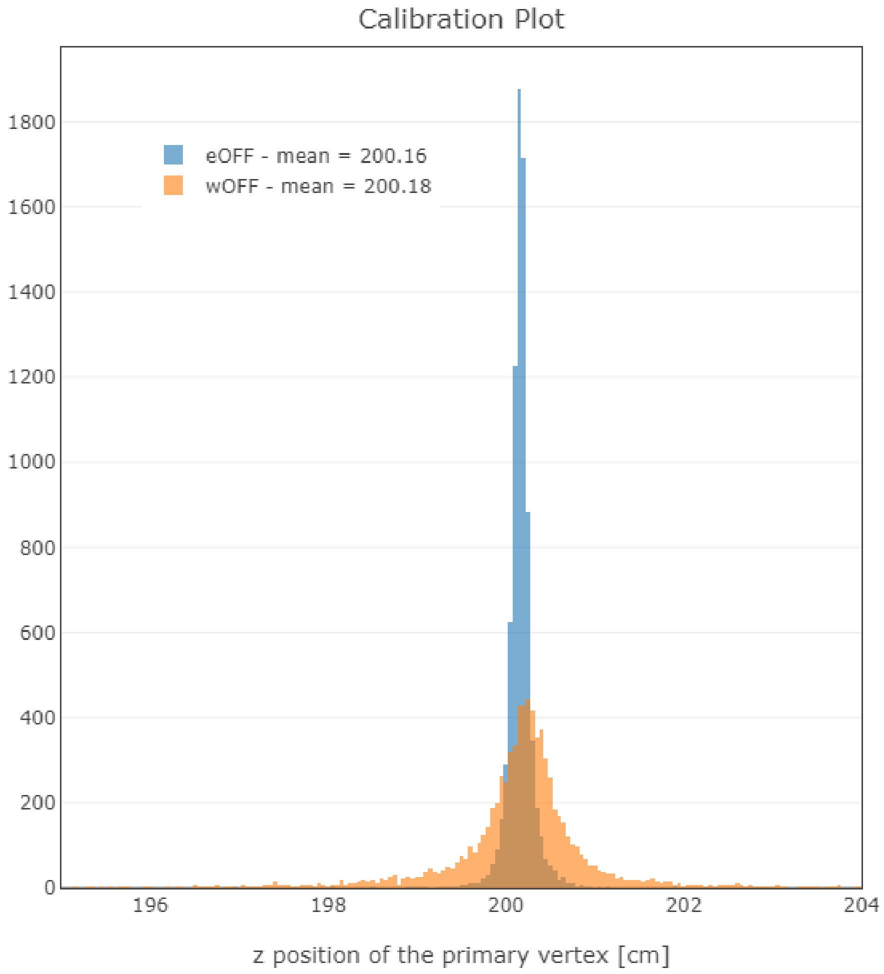






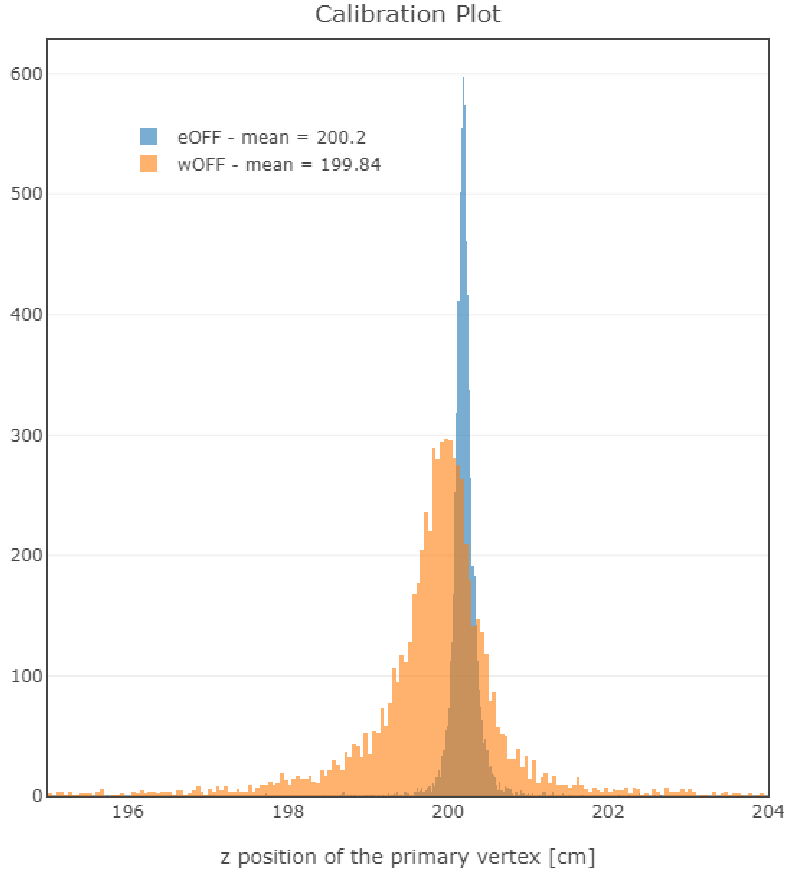
After the Calibration

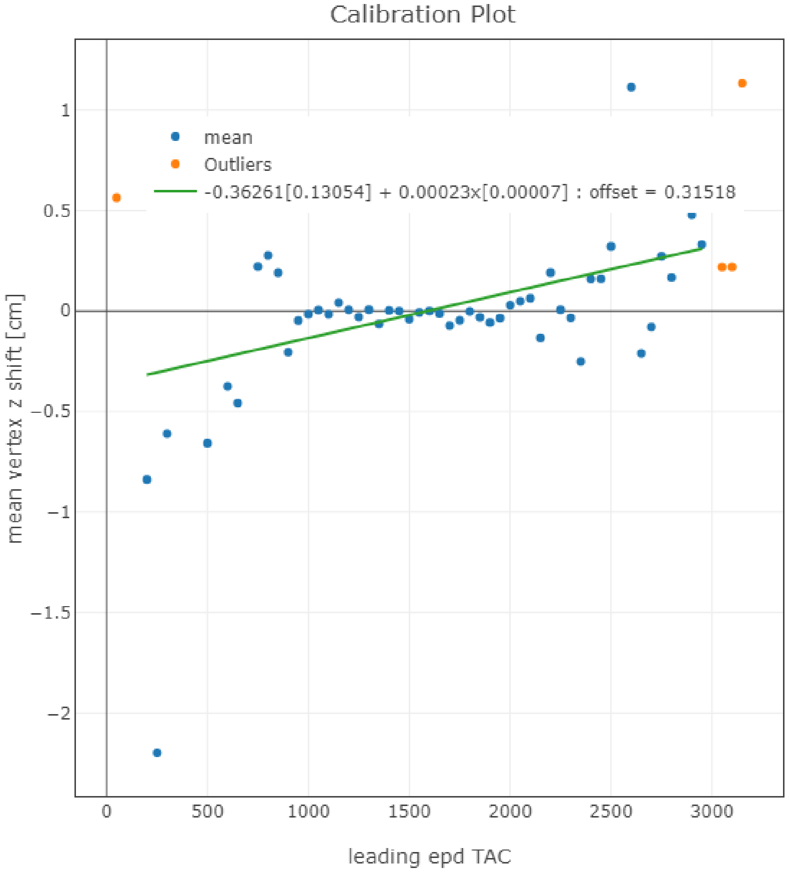




Higher Energy FXT Runs [31 GeV is shown]

- Higher energy FXT runs do not show such a good EPD event-by-event timing / correlation, but correction is still done based on the average separation between vertices





Correction Implementation

- Actual implementation is pretty simple and as non-invasive as possible

```
//      EPD based event-by-event correction for the hit timing
        double mTimeBinWidth = 1./StTpcDb::instance()->Electronics()->samplingFrequency();

        int ew = 0;
        int TAC = 0;
        int maxTAC = -1;

        int doEPDT0Correction = StTpcBXT0CorrEPDC::instance()->nrows();

        if (doEPDT0Correction) {
                StEpdCollection * epdCol = pEvent->epdCollection();
                if (epdCol) {
                        StSPtrVecEpdHit &epdHits = epdCol->epdHits();
                        int nEpdHits = epdHits.size();

                        for(int i = 0; i < nEpdHits; i++) {
                                StEpdHit * epdHit = dynamic_cast<StEpdHit*>(epdHits[i]);
                                TAC = 0;
                                if (epdHit->tile() > 9)  continue; // only tiles 1 - 9 have timing info
                                if (epdHit->id() < 0) ew = -1; // tile is on the east
                                else ew = 1;
                                if (epdHit->adc() < 100) continue;
                                TAC = epdHit->tac(); // this is the timing
                                if (TAC > maxTAC) maxTAC = TAC;
                        }
                }
        }
//      ======================================================
```

```
//              THIS IS A BLOCK TO CORRECT TIMING IN FXT MODE FOR DATA
                if (doEPDT0Correction) time += StTpcBXT0CorrEPDC::instance()->getCorrection(maxTAC, driftVelocity, mTimeBinWidth);
//              ======================================================
```

```
class StTpcBXT0CorrEPDC : public St_tpcBXT0CorrC {
        public:
                static StTpcBXT0CorrEPDC * instance();
                StTpcBXT0CorrEPDC(St_tpcBXT0Corr * table = 0) : St_tpcBXT0CorrC(table) {}
                virtual ~StTpcBXT0CorrEPDC() {fgInstance = 0;}

                double getCorrection (double epdTAC, double driftVelocity, double timeBinWidth) {
                        double timeBucketShiftScale = 500000/(driftVelocity*timeBinWidth);
                        double generalOffset = a(0)[0];
//                      printf("%f, %f, %f, %f, %f\n, ", epdTAC, driftVelocity, timeBinWidth, timeBucketShiftScale, generalOffset);
                        if (epdTAC == -1) return timeBucketShiftScale*generalOffset;
                        else return timeBucketShiftScale*(generalOffset + a(0)[1] + a(0)[2]*epdTAC);
                }
        private:
                static StTpcBXT0CorrEPDC * fgInstance;
                ClassDef(StTpcBXT0CorrEPDC, 1)
};
```

Calibration Procedure

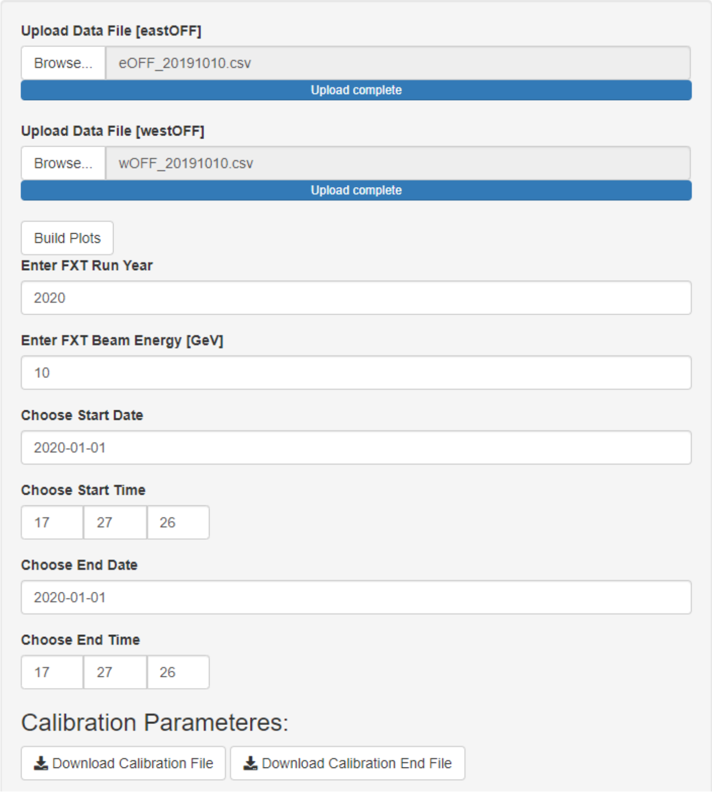
- Run data reconstruction for about 100k events separately for east and west sides of the TPC

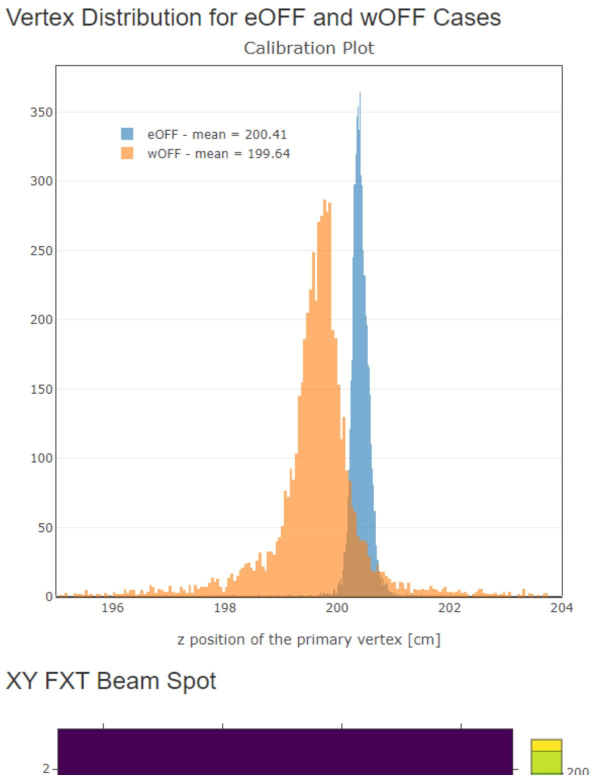
- Note the start and end time of the FXT run under calibration

- Run the analyzer on the reconstructed data to extract necessary information

- Produced files could be plugged into a handy GUI (or one can simply used them to calculated number as well) https://iraklic.shinyapps.io/LiveLumi/

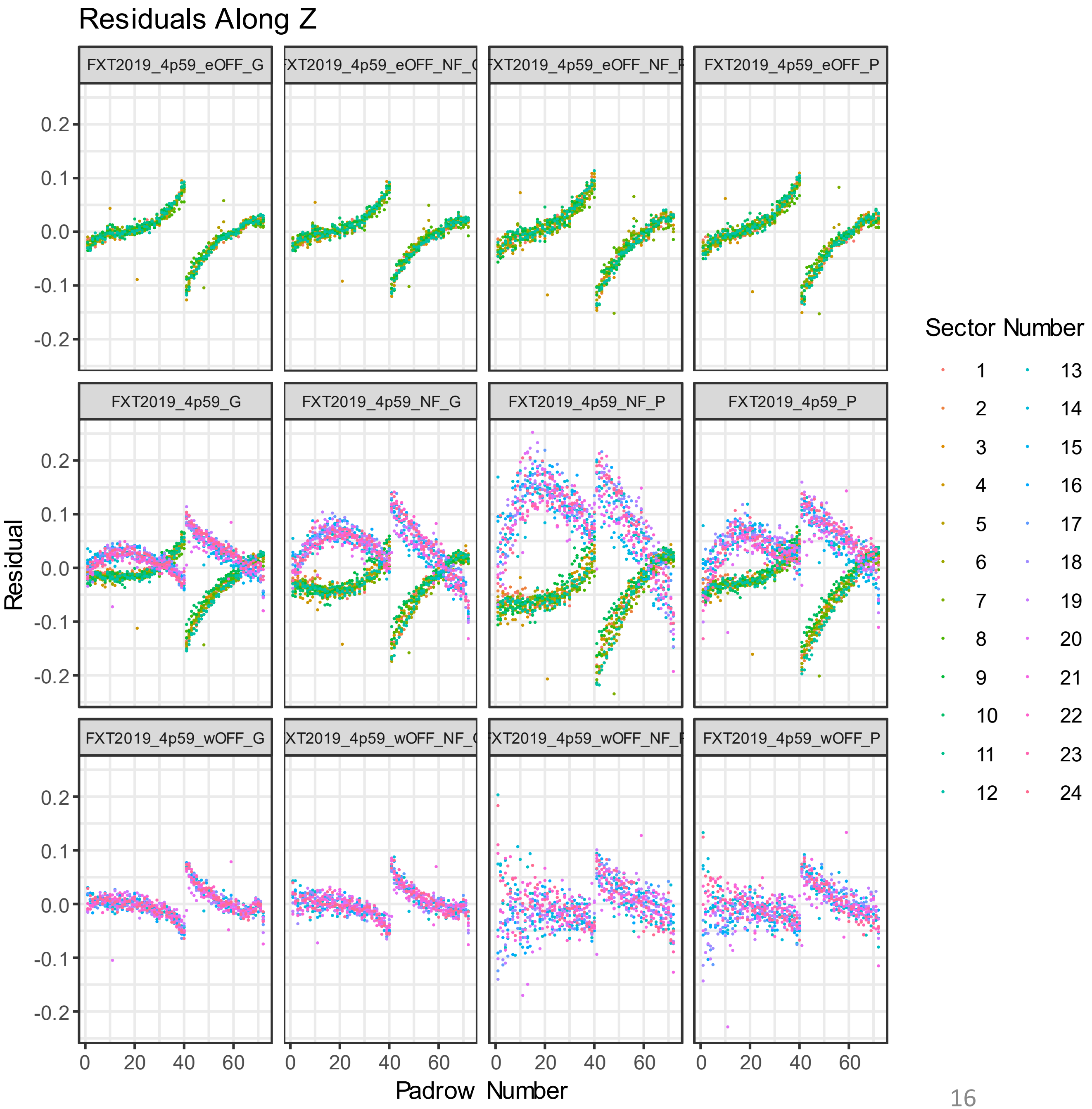
- Simply download the calibration files





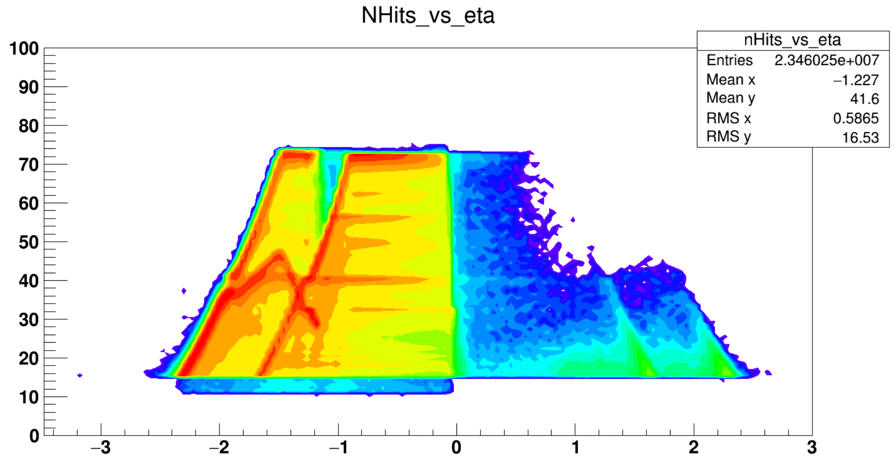
Residuals - 2019

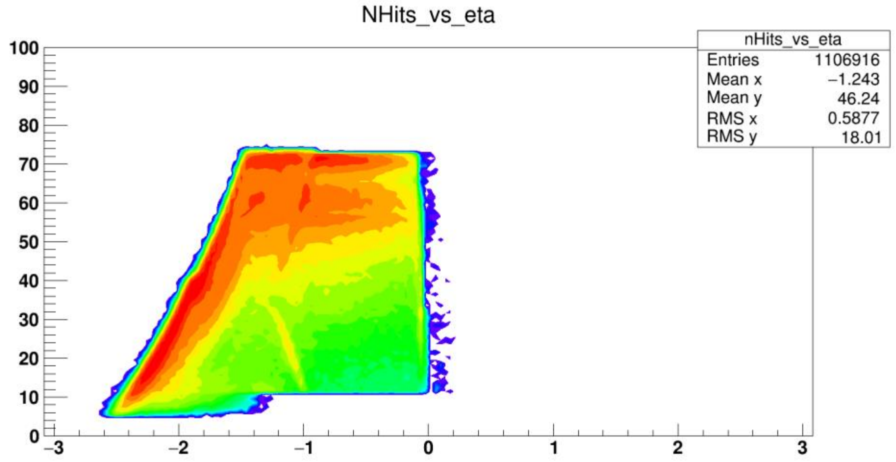
- The residuals (distance between hit and the track) are a good way to look at the effect
- NF on the plots stand for No Fix; P – primary tracks; G – Global tracks; eOFF/wOff – reconstruction from west/esat side hits only
- Plots clearly show the improvement due to the new calibration in the case when all hits are used



Calibration Works

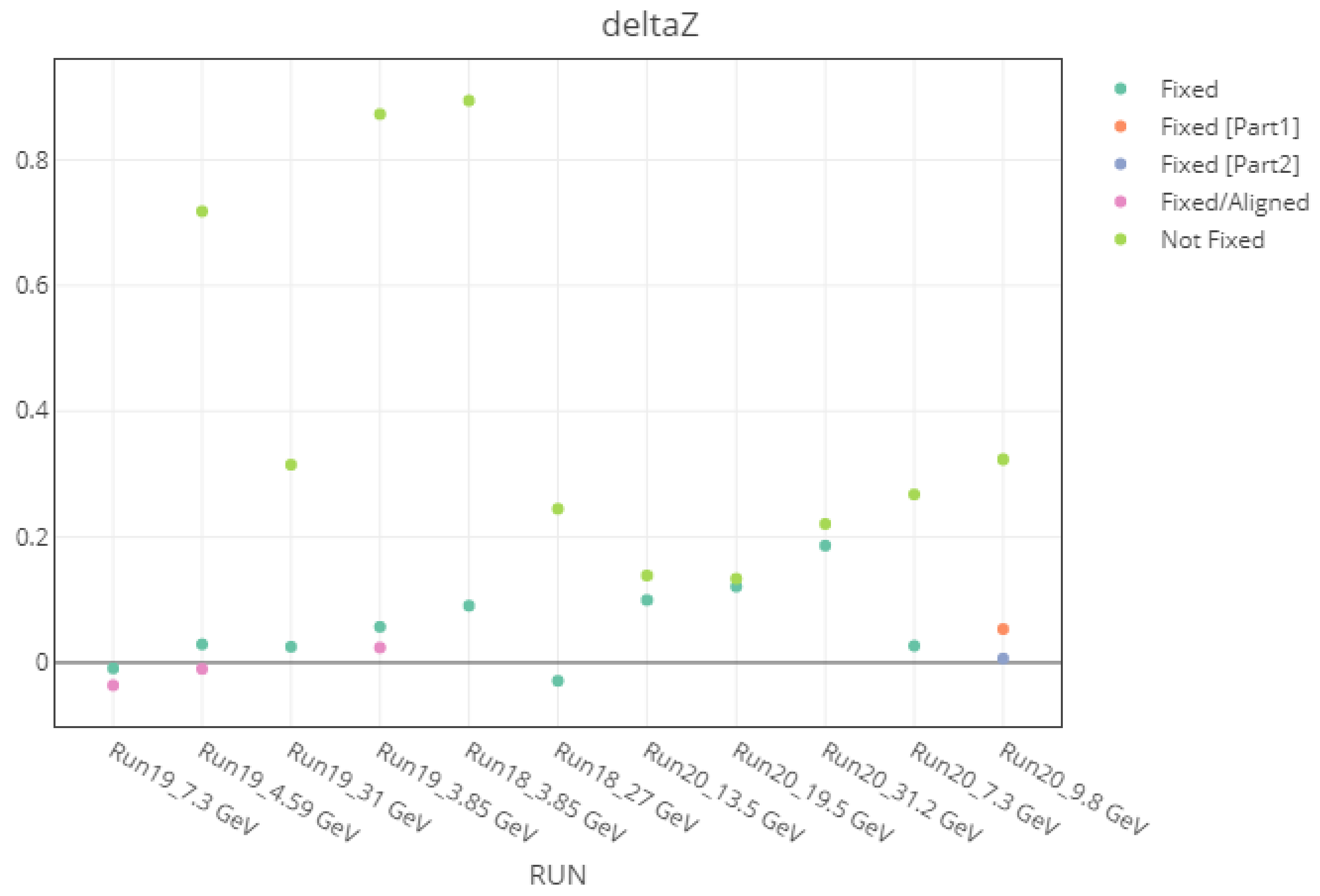
QA groups reported noticeable improvement on their side after the recalibration of the data





Run-20 FXT Calibration

- Run-20 saw probably the greatest number of FXT runs
- The calibration of the FXT timing was handled completely by Dan's graduate student Matthew Harasty (thank you Matt)
- Matt completed these calibrations in a very timely manner – they were ready by the end of the run

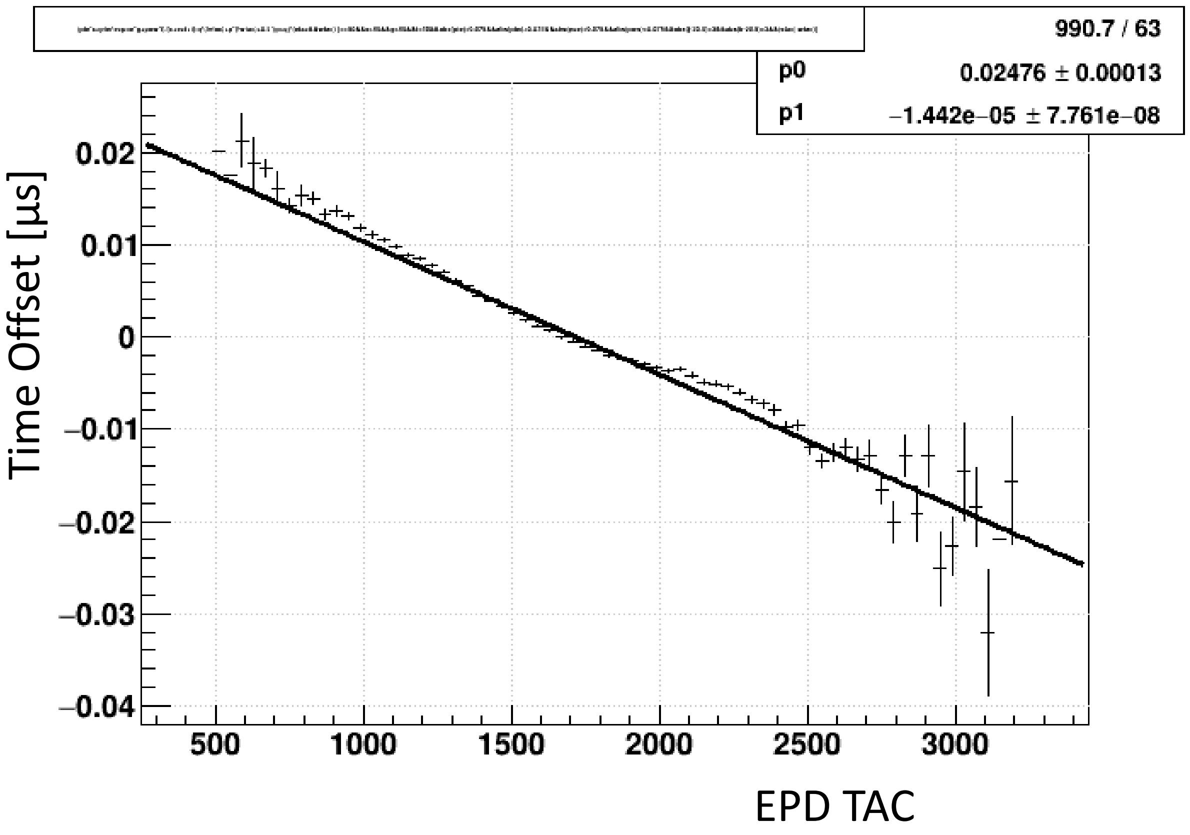


Summary

- It was demonstrated during Run-19 that event timing for the FXT needs to be corrected
- The correction procedure was established based on the EPD measured timing of the event
- The calibration procedure is well defined and was done by a student for the Run-20

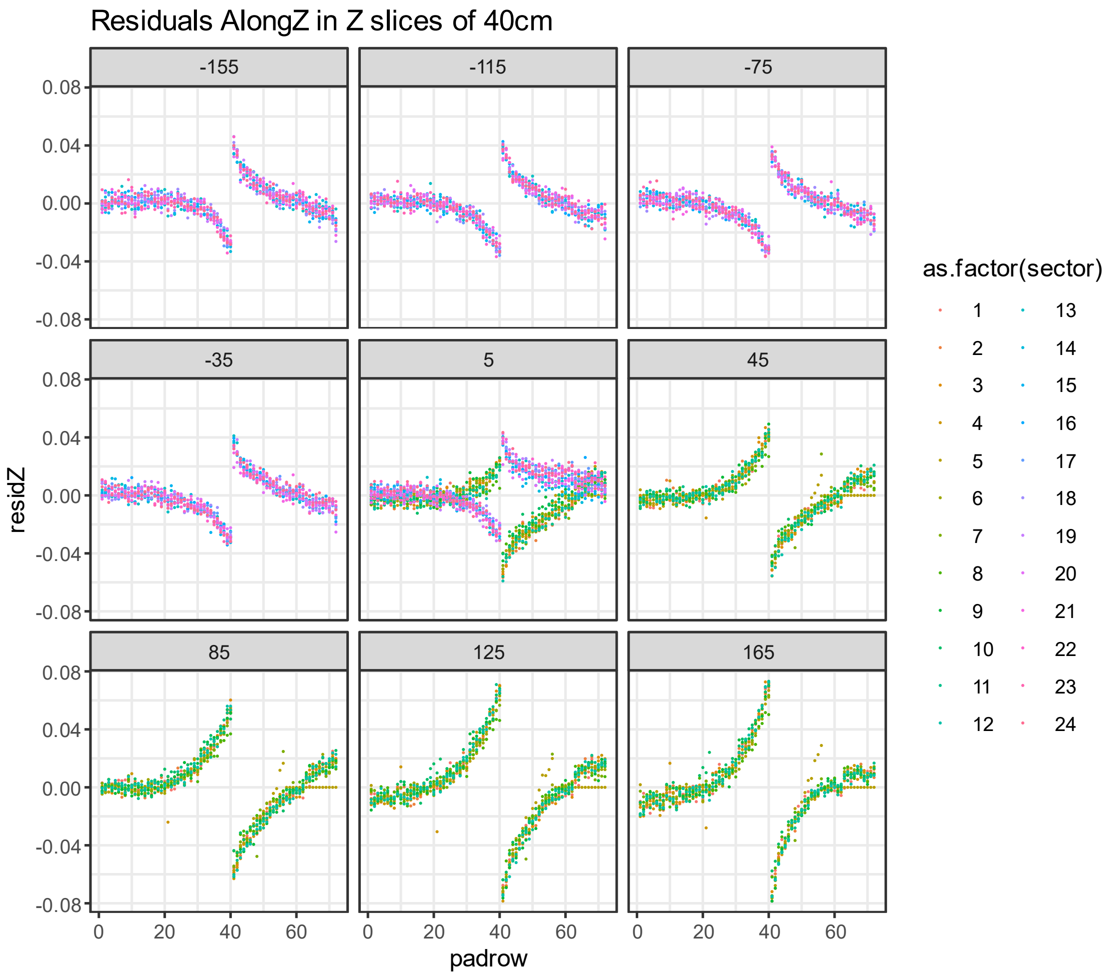
Outlook

- Off-center collision in ideal situation needs to be always corrected;
- In collisions this effect:
  - used to be much smaller pre-BES-II
  - is averaged out because of the symmetric distribution of an actual vertex
  - Is harder to catch unlike FXT where we know exactly where the vertex should be
- Nevertheless:
  - Long bunches might be producing noticeable effect (which might be swept under other calibration, or making them harder)
  - EPD can prove useful for event-by-event correction in the collider mode as well
  - Gene showed that 97% of events have EPD readout either on both sider or at least on one side and could be corrected for



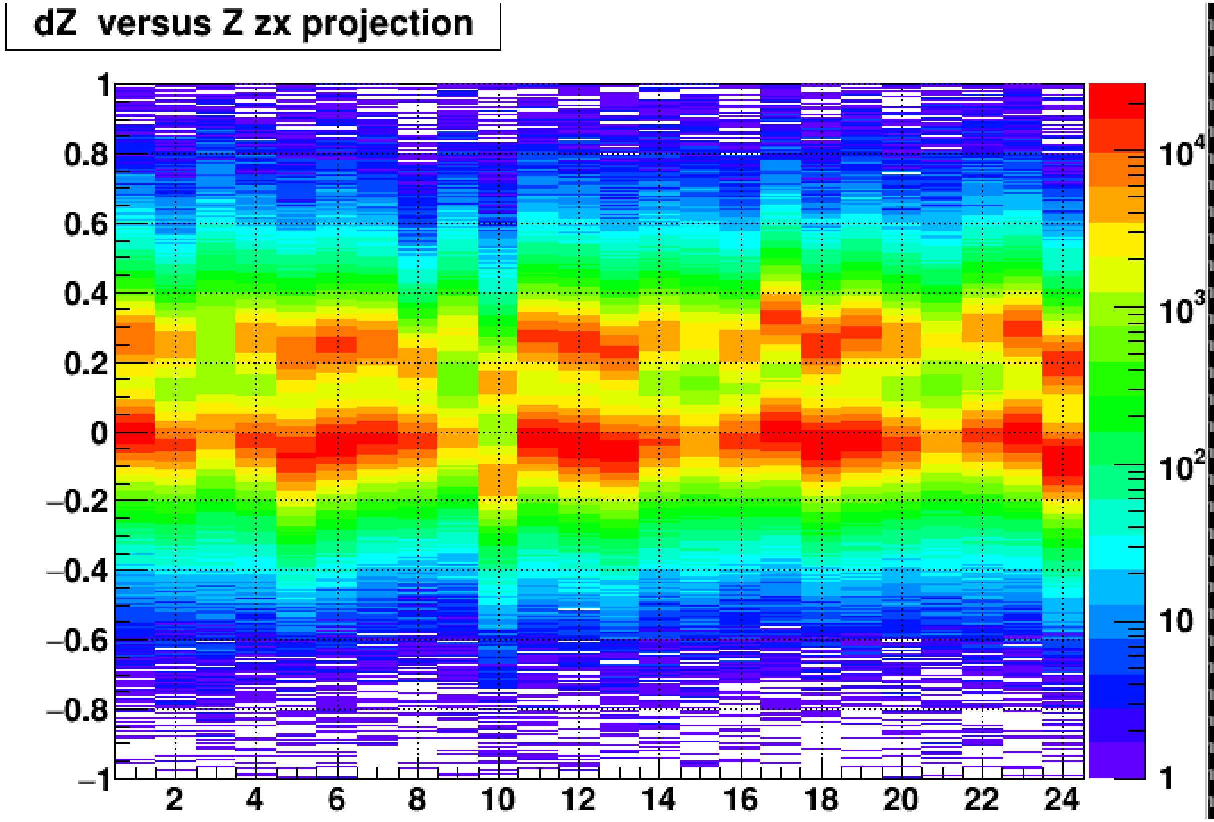
Backup [Alignment I]

- FXT Study showed that there were non-zero residuals (in z direction) even after the vertex correction
- Residuals were independent of the z position therefore unlikely to be from the dynamic distortions but from the misalignment between inner and outer sectors
- This triggered the revision of the Run-19 alignment, which had a lot of cosmics due to it being the first run after complete iTPC instalation



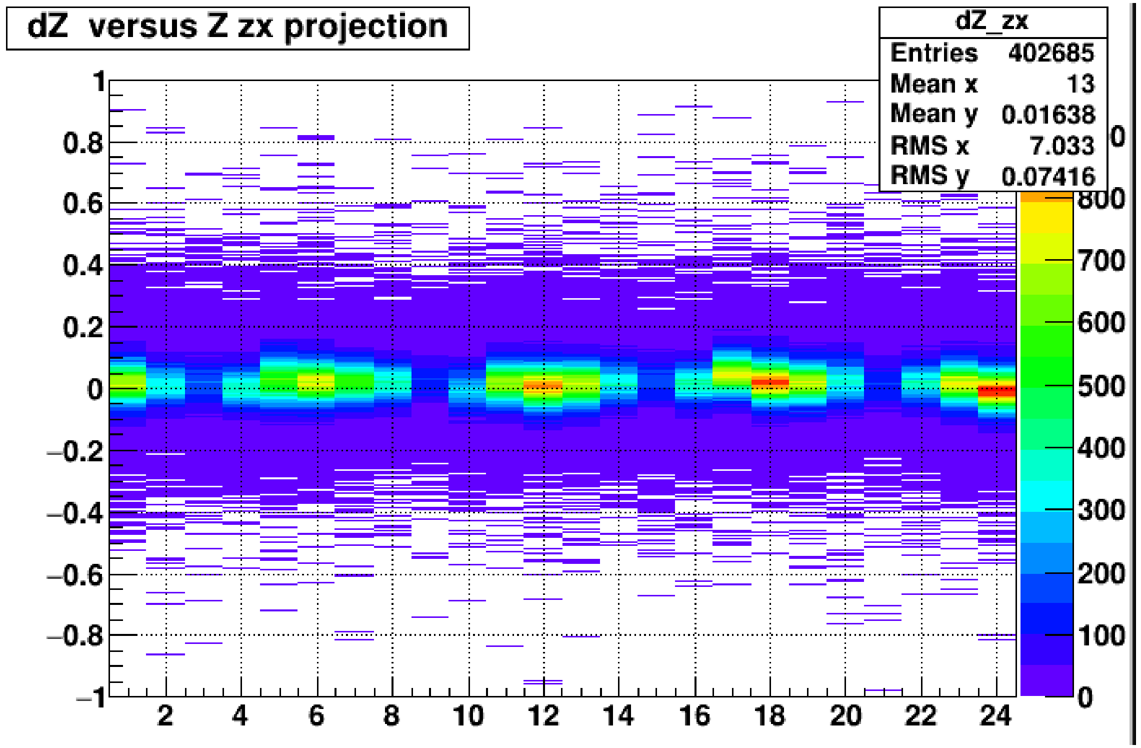
Backup [Alignment II]

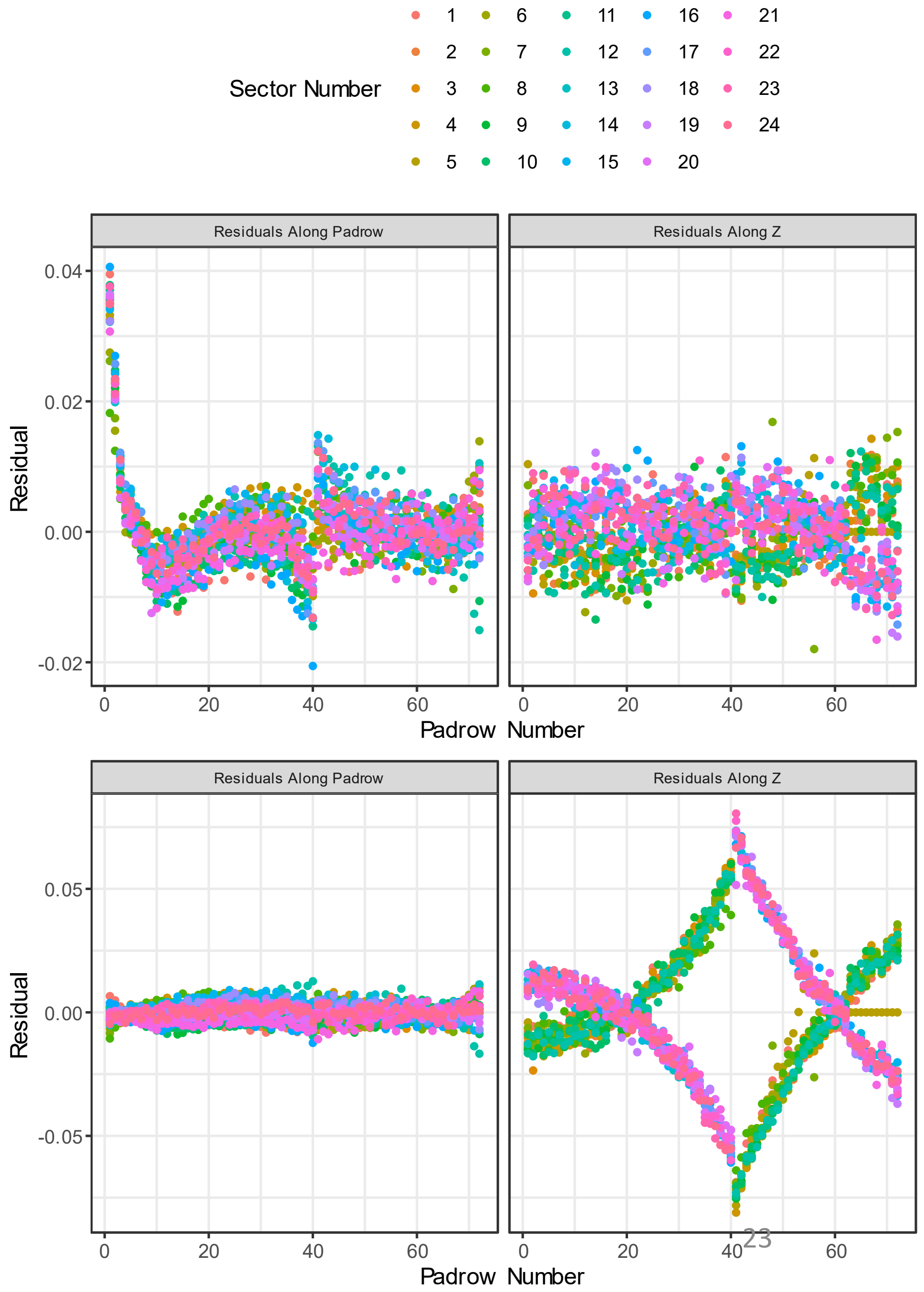
- Culprit was found to be mismatch in the tpcSectorT0 used for the alignment during cosmics run and post-cosmics data-taking
- Problem was rectified and alignment re-done



Backup [Alignment III]

- Result showed that this was clearly necessary





Backup [Alignment IV]

- In order to complete the picture, the supersector alignment was also revisited and it was found that there is substantial split in the primary vertex DCA from the vertex reconstructed from each sector (as shown on the plot)
- Alignment was re-done but the new parametrization caused a split tin the reconstructed m^2
- A lot of efforts were made to understand the problem since but the problem is not quite understood/agreed upon at the moment

Full story on my blog:

https://drupal.star.bnl.gov/STAR/blog/iraklic/run19-cosmics-and-supersector-alignment