Truncated Mean: BugFix, Completion service task

Florian Herrmann

September 24, 2018

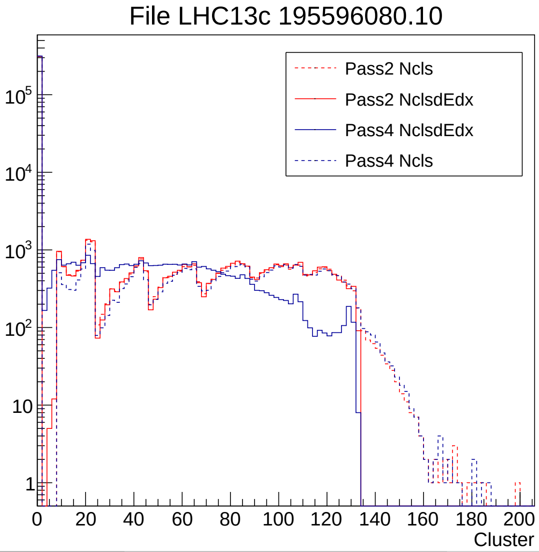
BUG

- **Problem**
  - significant cluster loss in truncated mean method (in LHC15 compared to LHC13)
  - bad resolution

BUG

- **BUG**
  1. **CODING: double baseline subtraction in truncated mean**
     Truncated mean expects uncalibrated Signal + Baseline
     But in **AliTRDclusterizer** the logic is (since 2012!):
     - IF (!OnlineCalibrationTable)
       $\Rightarrow$ RawSignal (includes Baseline);
     - ELSE
       $\Rightarrow$ (RawSignal-Baseline)/OnlineGainCalibration **ERROR**
  2. **MISSING ONLINE CALIBRATION**
     Up to LHC13b_pass3 the OCDB snapshot does not include the right online correction table (Krypton 2012-1)
     $\rightarrow$ truncated mean correct (apart from missing online deconvolution)
     Since LHC13b_pass4 the OCDB snapshot includes the correct table
     $\rightarrow$ truncated mean goes wrong
  3. **UNFORTUNATELY we used, improved and checked the truncated mean only with LCH13bc_pass3 data**

BUG


File LHC13c 195596080.10
Pass2 Ncls
Pass2 NclsdEdx
Pass4 NclsdEdx
Pass4 Ncls
10^5
10^4
10^3
10^2
10
1
0
20
40
60
80
100
120
140
160
180
200
Cluster

BUG

- **CONSEQUENCES**
  1. **Likelihood Method**
     Fortunately, the Likelihood method is not affected
  2. **Truncated Mean**
     In principle all DATA and so far created PARAMETER sets are NOT USABLE
     $\Rightarrow$CPass0/CPass1 and FullPass have to be rerun
  3. **Others**

BUGFIX

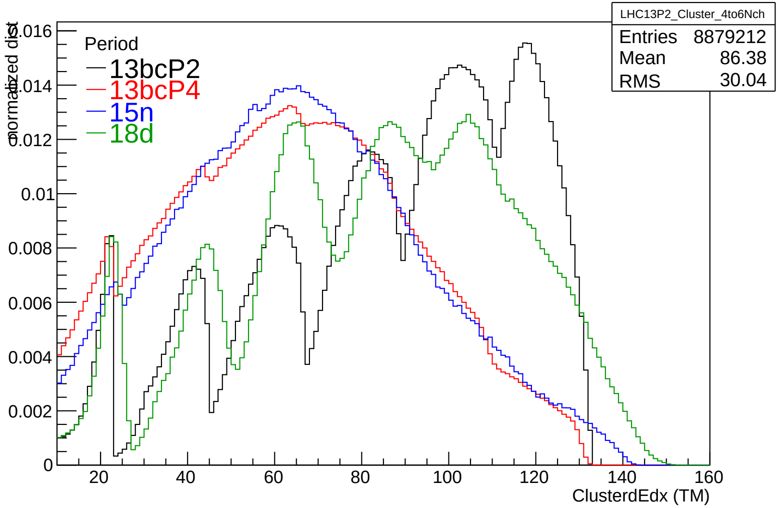
BUGFIX

- Strategy: Simple as possible
- Therefore new logic in **AliTRDclusterizer**

```
IF (!OnlineCalibrationTable)
    ⇒ RawSignal-Baseline;
ELSE
    IF (RawSignal==0)
        ⇒ 0
    ELSE
        ⇒ (RawSignal-Baseline)/OnlineGainCalibration
```

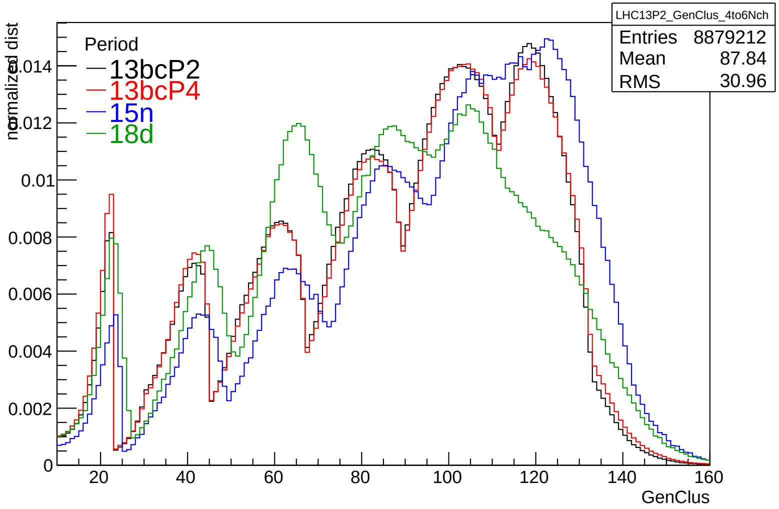
- Remove baseline subtraction in truncated mean class (AliTRDdEdxReconUtils)

BUGFIX

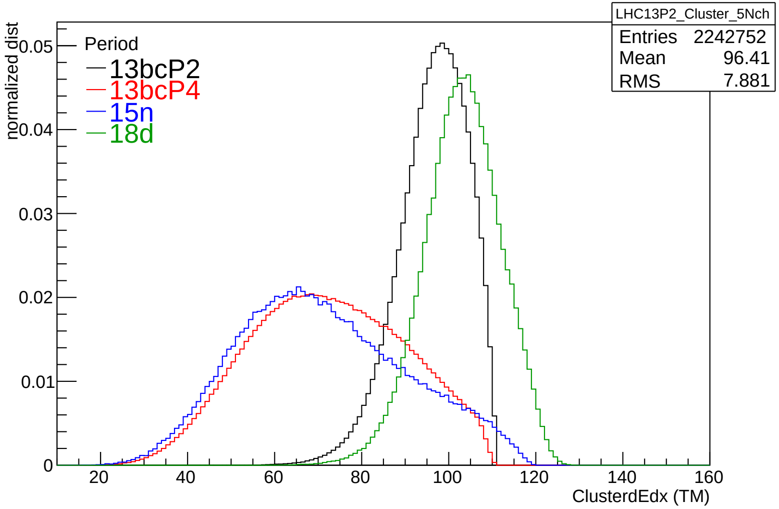


⇒ two effects: modified read_out (increase max Ncls), more inactive chambers

BUGFIX

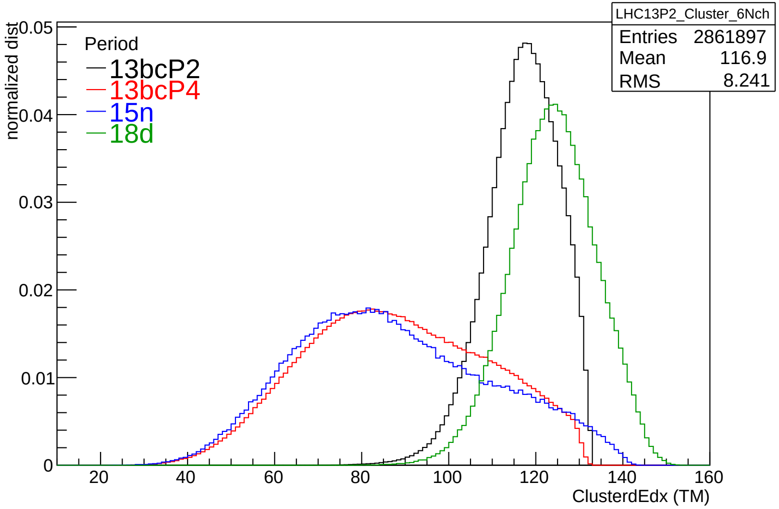

LHC13P2_GenClus_4to6Nch
Entries 8879212
Mean 87.84
RMS 30.96
Period
13bcP2
13bcP4
15n
18d
normalized dist
0.014
0.012
0.01
0.008
0.006
0.004
0.002
0
20
40
60
80
100
120
140
160
GenClus

BUGFIX



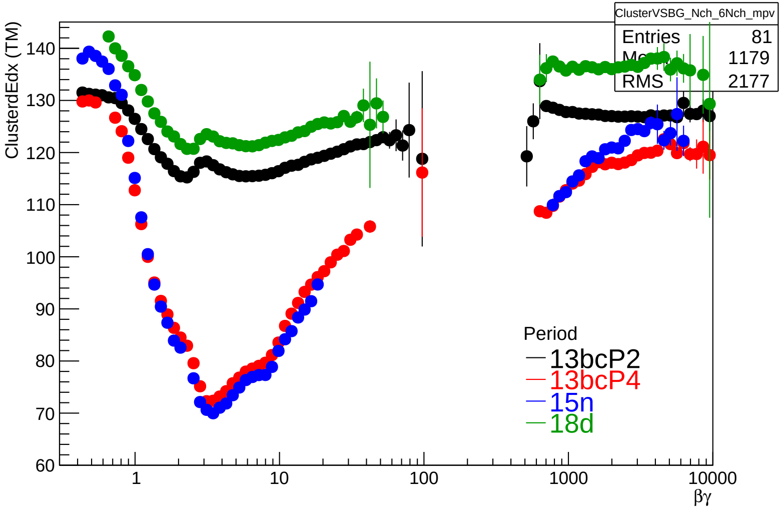
normalized dist
Period
13bcP2
13bcP4
15n
18d
LHC13P2_Cluster_5Nch
Entries 2242752
Mean 96.41
RMS 7.881
0
0.01
0.02
0.03
0.04
0.05
20
40
60
80
100
120
140
160
ClusterdEdx (TM)

BUGFIX



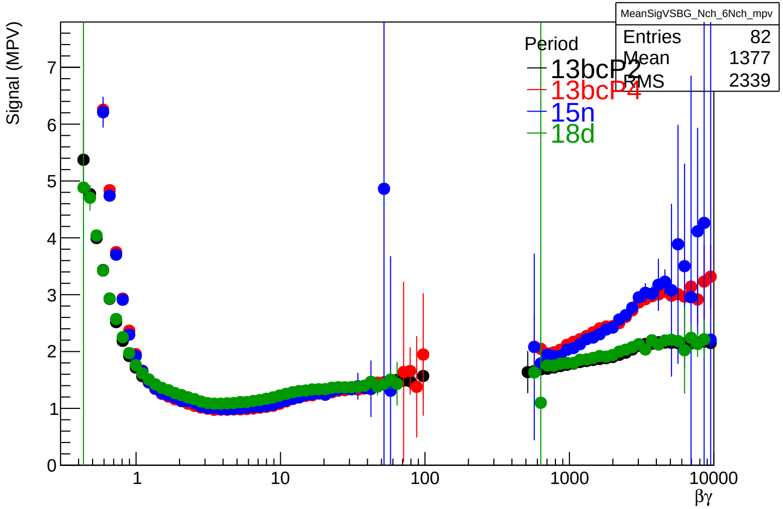
LHC13P2_Cluster_6Nch
Entries 2861897
Mean 116.9
RMS 8.241
Period
13bcP2
13bcP4
15n
18d
normalized dist
0.05
0.04
0.03
0.02
0.01
0
20
40
60
80
100
120
140
160
ClusterdEdx (TM)

BUGFIX



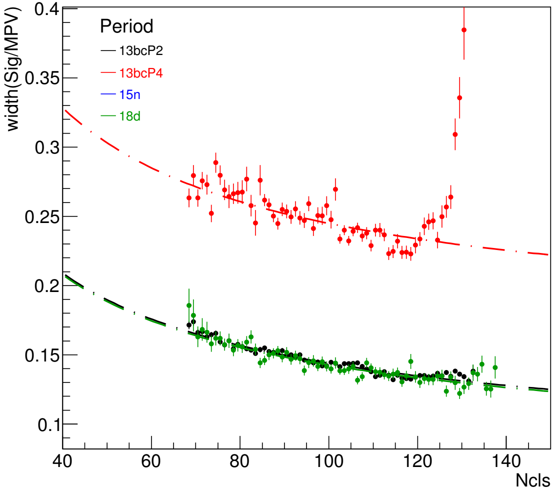
ClusterVSBG_Nch_6Nch_mpv
Entries 81
Mean 1179
RMS 2177
ClusterdEdx (TM)
140
130
120
110
100
90
80
70
60
1
10
100
1000
10000
βγ
Period
13bcP2
13bcP4
15n
18d

BUGFIX



MeanSigVSBG_Nch_6Nch_mpv
Entries 82
Mean 1377
RMS 2339
Period
13bcP2
13bcP4
15n
18d
Signal (MPV)
0
1
2
3
4
5
6
7
1
10
100
1000
10000
βγ

BUGFIX



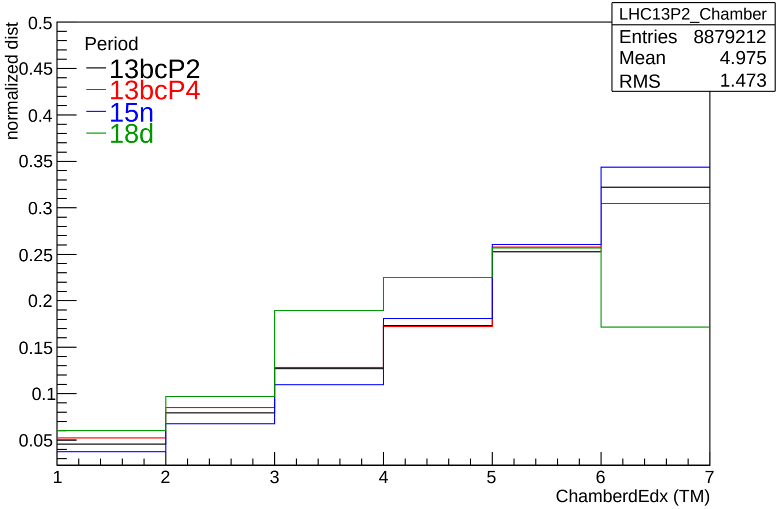
width(Sig/MPV)
0.4
0.35
0.3
0.25
0.2
0.15
0.1
Period
13bcP2
13bcP4
15n
18d
40
60
80
100
120
140
Ncls

Summary

- Successful BugFix
- Concluding tasks:
  - finish documentation in trd-wiki: code, improvements, BugFix
  - proposal to add code to AliPhysics

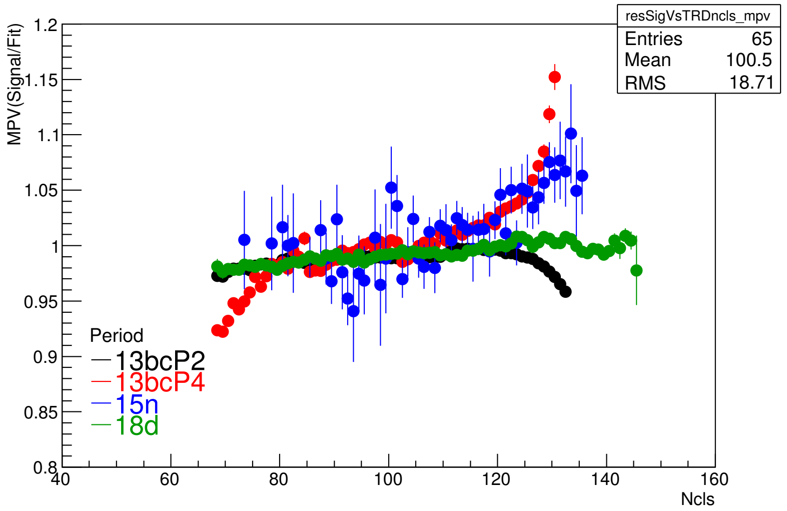
backup slides

BUGFIX



LHC13P2_Chamber
Entries 8879212
Mean 4.975
RMS 1.473
Period
13bcP2
13bcP4
15n
18d
normalized dist
ChamberdEdx (TM)
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
1
2
3
4
5
6
7

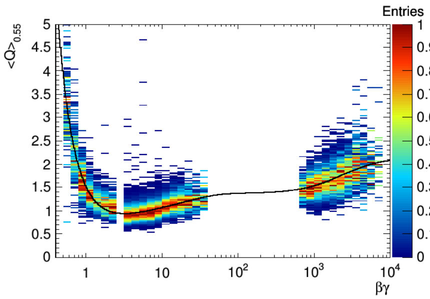
BUGFIX

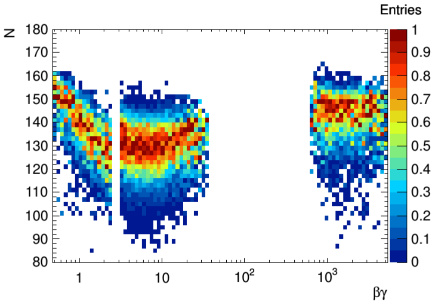


resSigVsTRDncls_mpv
Entries 65
Mean 100.5
RMS 18.71
MPV(Signal/Fit)
1.2
1.15
1.1
1.05
1
0.95
0.9
0.85
0.8
40
60
80
100
120
140
160
Ncls
Period
13bcP2
13bcP4
15n
18d

Starting Point: Xiangou

- determine most probable TM signal by fitting gauss for each $\beta\gamma$ slice
- interpolate missing $\beta\gamma$ slices by fitting Aleph+TR function to this MPV





Starting Point: Xiangou

- Width of signal depends dominantly on number of cluster $\approx \frac{1}{\sqrt{N}}$, therefore fit gaussian to deviation from MPV

$$\frac{TMSignal(\beta\gamma, \eta, NCluster, Centrality, ...)}{MPVFit(\beta\gamma)}$$

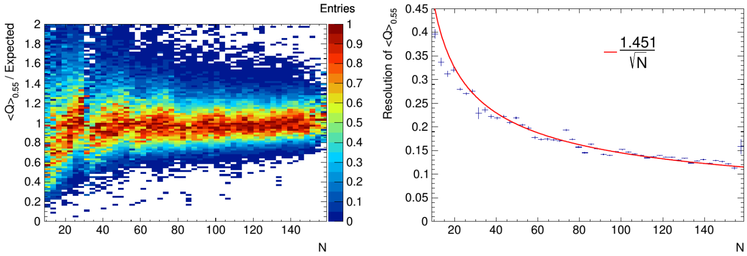


Figure 5.9: (*Left*) Scaled truncated mean signal and (*right*) the signal resolution as a function of the number of clusters.

PhD thesis Xiangou

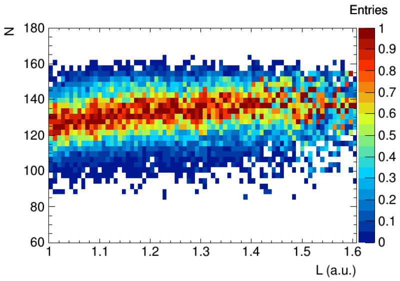
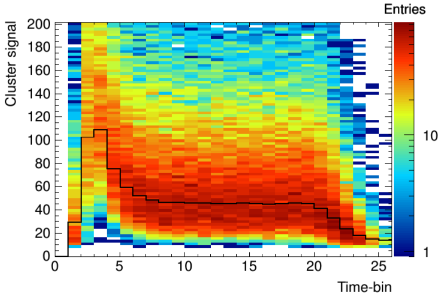
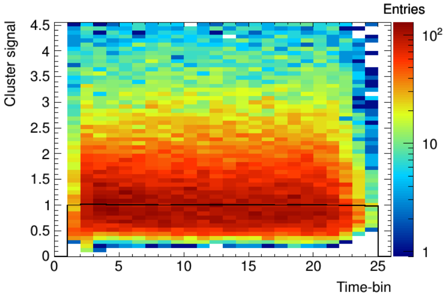


Figure 5.8: Number of clusters as a function of the particle path length in the TRD.

PhD thesis Xiangou





TM Cluster signal

Cluster signal used for Likelihood. Calculated in TRDclusterizer.cxx and accessed via GetQ

$$padCharge = \frac{RawPadSignal - fbaseline}{OnlCalRoc} \times \frac{1}{CalDet \times CalPad}$$

$$clsCharge = padCharge(max-1) + padCharge(max) + padCharge(max+1)$$

TM Cluster signal

For TM the signal will be calculated without new calibration
TRDClusterizer::CreateClusters (if fCalOnlGainRoc = true)

$$padCharge = \frac{RawPadSignal - fbaseline}{OnlCalRoc} + 0.5f$$

And stored as Short_t in 7 dim RawSignal array

$$RawSignal = \{padCharge(max - 3), \ldots, padCharge(max), \ldots\}$$

TM Cluster signal

Afterwards in AliTRDdEdxReconUtils.cxx new calculation

$$\mathit{clsCharge} = \sum(\mathit{ifRawSignal} > 0)\frac{\mathit{RawSignal}[i] - \mathit{basline}(= 10)}{\mathit{CalPad}} \quad (1)$$

Note: double baseline subtraction!

MissingDetCalibration – because of scaling with TPCsignal!

Cls are counted if clsCharge>0

Signal is scaled with path length and QScale=50

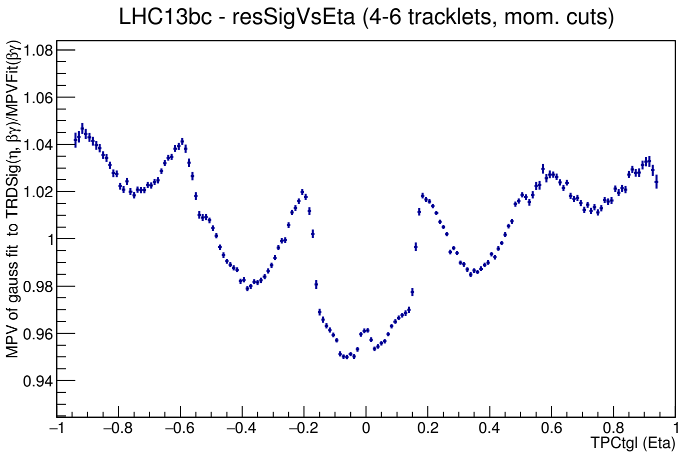
Afterwards Signal is scaled with GainCalibration for TM (missing cluster only if gain factor = 0)

TM Cluster signal

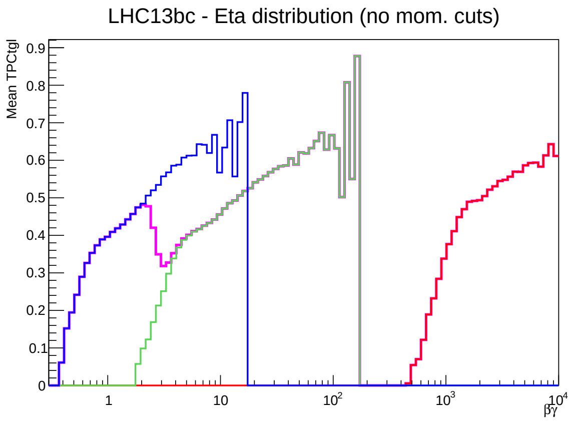
```
 Double_t AliTRDdEdxReconUtils::GetRNDClusterQ(AliTRDcluster *cl, const Double_t baseline)
{
  //
  //get cluter q from GetRawQ, apply baseline and Kr pad-calibration
  //

  const Int_t det     = cl->GetDetector();
  const Int_t pad3col = cl->GetPadCol();
  const Int_t padrow  = cl->GetPadRow();

  Double_t rndqsum = 0;
  for(Int_t ii=0; ii<7; ii++){
    if(cl->GetSignals()[ii] < EPSILON){//bad pad marked by electronics
      continue;
    }

    const Int_t icol = pad3col+(ii-3);
    const Double_t padgain = GetPadGain(det, icol, padrow);
    if(padgain<0){//indices out of range, pad3col near boundary case
      continue;
    }

    const Double_t rndsignal = (cl->GetSignals()[ii] - baseline )/(AliTRDdEdxBaseUtils::IsPadGainOn()? padgai

    //sum it anyway even if signal below baseline, as long as the total is positive
    rndqsum += rndsignal;
  }

  return rndqsum;
}
```

Eta dependence (Lukas, Yvonne, Florian)

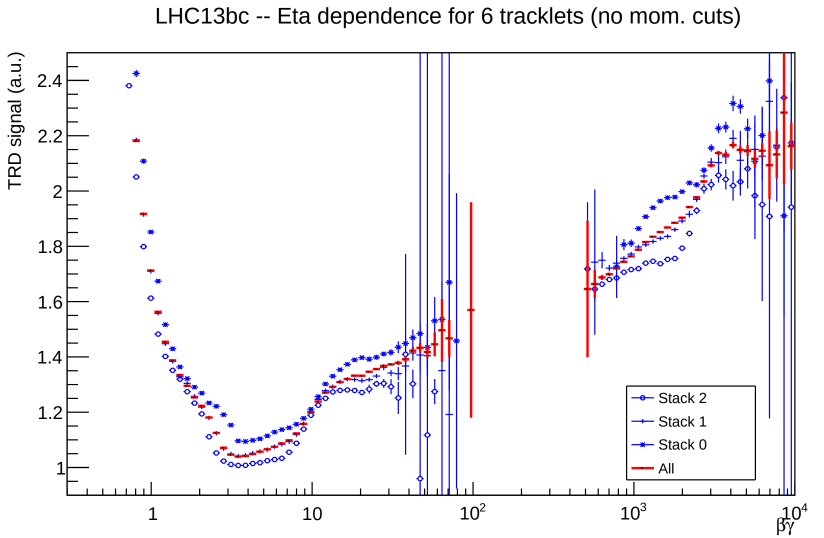
- eta dependence in signal (around 5%)



Eta Dependence

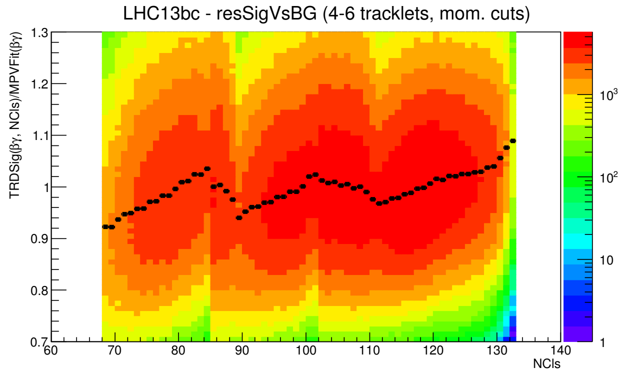


Eta Dependence



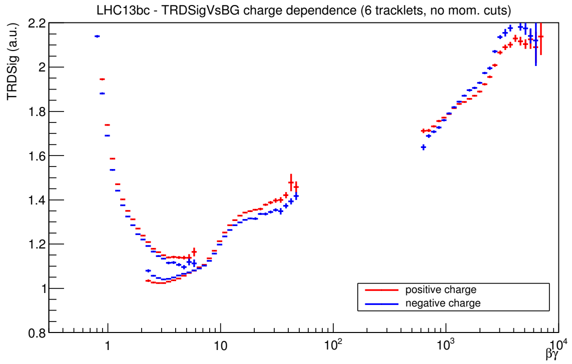
Cluster dependence

- signal increases in each chamber with increasing cluster number (TRD meeting 22.06)



Additional Observations

- charge dependence



CORRECTION MAPS

- **Some Insights**
  - Up to now: EtaCorrection, ClusterCorrection, CentralityCorrection
  - Corrections in Code
    - PadGain Correction - but no chamber gain correction (due to TPCSignal calibration)
    - Path length correction
    - Time bin calibration using truncated mean of TRDSignal/TPCSignal
  - Possible sources of remaining deviations:
    - Scaling to TPCSignal corrects for multiple effects (like chamber calibration), but we introduce all deviations from the TPC (e.g. eta dependence)
    - Scaling to TPCSignal introduces bias if TRDSignal/TPCSignal shows $\beta\gamma$ dependence
    - time bin calibration: different particle compositions in calibration bins
  - However, improvements seems to be very time-consuming