Truncated Mean: Bug, BugFix, CorrectionMaps

Florian Herrmann

January 16, 2017

Graduiertenkolleg 2149
Research Training Group

DFG Deutsche
Forschungsgemeinschaft

Westfälische
Wilhelms-Universität
Münster

ALICE

BUG

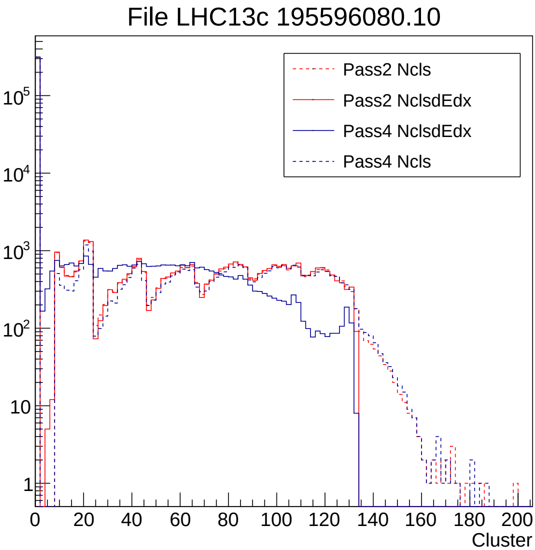
- **PROBLEM**
  - significant cluster loss in truncated mean method (in LHC15 compared to LHC13)
  - bad resolution

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)
       ⇒ RawSignal (includes Baseline);
     - ELSE
       ⇒ (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)
     → truncated mean correct (apart from missing online deconvolution)
     Since LHC13b_pass4 the OCDB snapshot includes the correct table
     → 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 by that in general. However, LHC13b (and others) are affected if the right online gain calibration table is missing
     $\Rightarrow$ Only small effect on efficiencies etc. but using LHC13b_pass4 data with parameter from LHC13b_pass3 may have an effect
     $\Rightarrow$ Yvonne is doing further checks
  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
     $\Rightarrow$We hope to join the next reconstruction with bug fix

BUGFIX

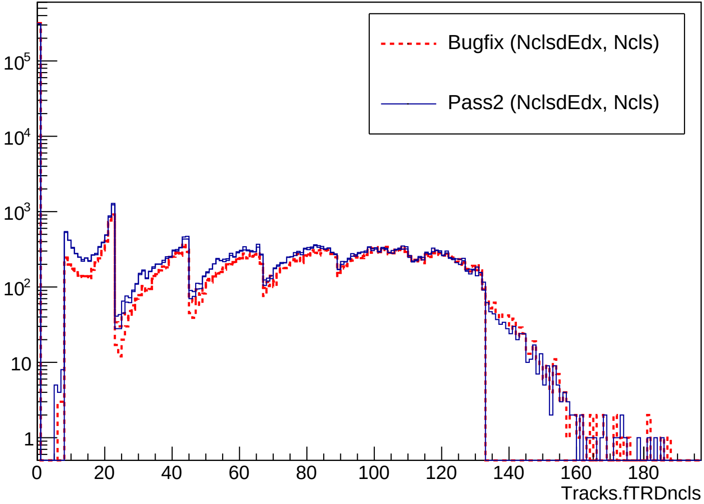
BUGFIX

- Strategy: Simple as possible (not doing it worse)
- Other classes make use of this signal variable, but as far as I could find out all expect a baseline subtracted signal (if not, they should have already realized)
- Therefore new logic in **AliTRDclusterizer**
  - IF (!OnlineCalibrationTable)
    ⇒ RawSignal-Baseline;
  - ELSE
    - IF (RawSignal==0)
      ⇒ RawSignal-Baseline
    - ELSE
      ⇒ (RawSignal-Baseline)/OnlineGainCalibration
- Remove baseline subtraction in truncated mean class (AliTRDdEdxReconUtils)

BUGFIX

- TESTS
  - General problem: Need correct time bin calibration for total check of right cluster number and resolution (hard to run locally)
  - Running local reconstruction even with wrong calibration should return correct cluster number
    - One file test successful (see next slice)
    - Larger sample possible
  - Since there is no danger of doing things worse, I propose to join the next reconstruction and see what happens
  - Is there a chance to rerun LCH3b/c?

BUGFIX



Bugfix (NclsdEdx, Ncls)
Pass2 (NclsdEdx, Ncls)
10^5
10^4
10^3
10^2
10
1
0
20
40
60
80
100
120
140
160
180
Tracks.fTRDncls

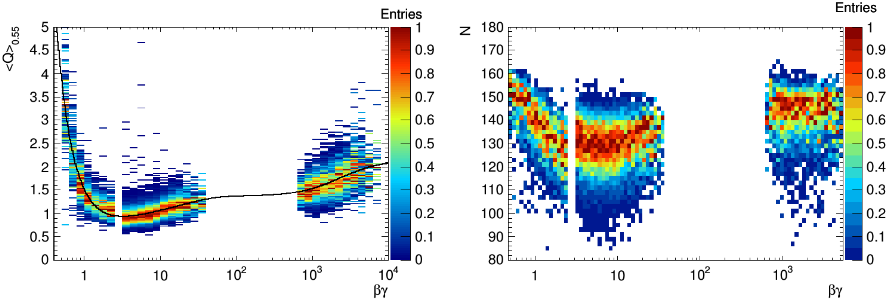
CORRECTION MAPS

- **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
    $\Rightarrow$ Priority now: bugfix

backup slides

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

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

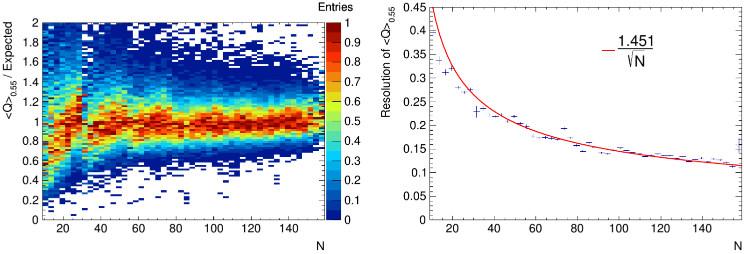


Figure 5.9: (*Left*) Scaled truncated mean signal and (*right*) the signal resolution as a function of the number of clusters.

PhD thesis Xiangou

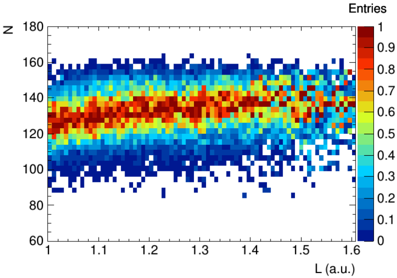
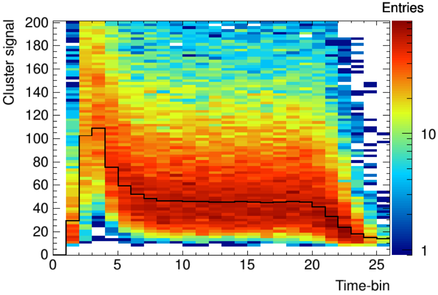
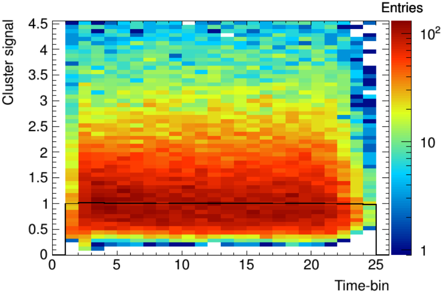


Figure 5.8: Number of clusters as a function of the particle path length in the TRD.

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

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

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

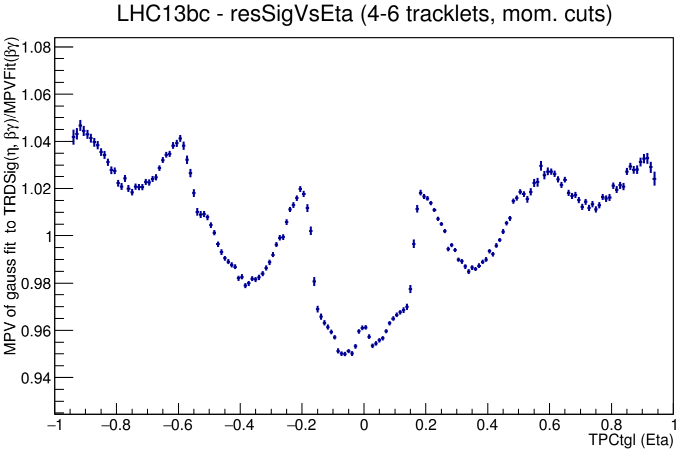
Afterwards Signal is scaled with GainCalibration for TM (missing cluster only if gain factor = 0)

TM Cluster signal

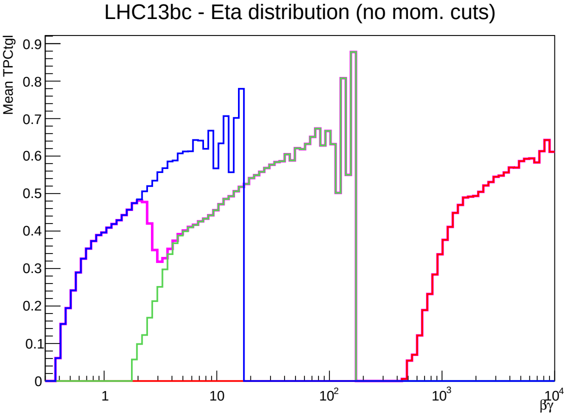
```
 Double_t AliTRDdEdxReconUtils::GetRNDClusterQ(AliTRDcluster *cl, const Double_t baseline)
{
  //
  //get cluter q from GetRawQ, apply baseline and Kr pad-calibration
  //

  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)

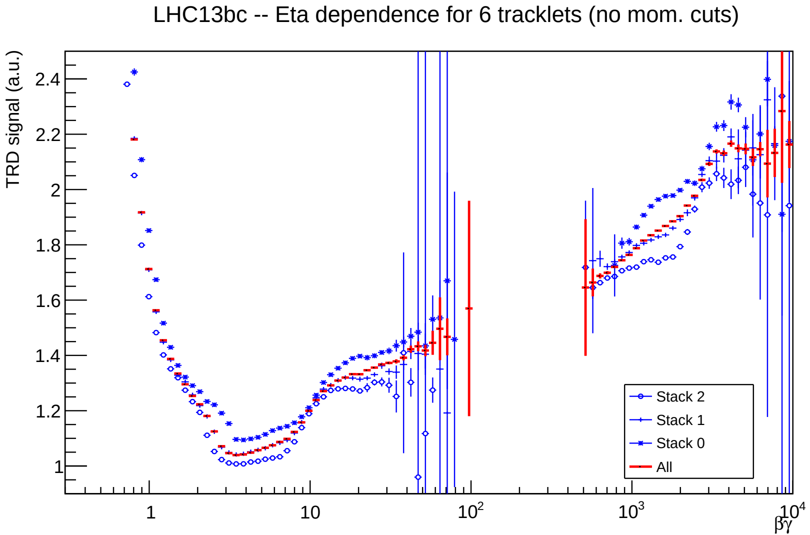
- eta dependence in signal (around 5$\%$)



Eta Dependence

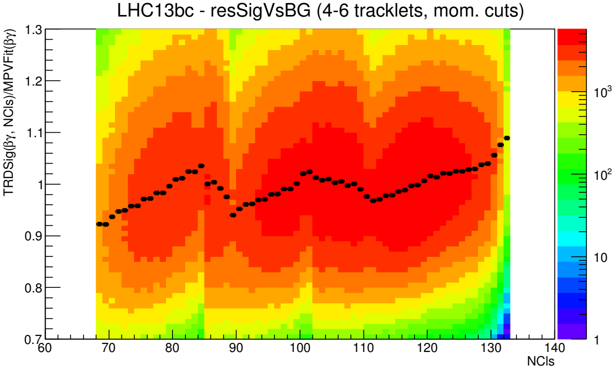


Eta Dependence



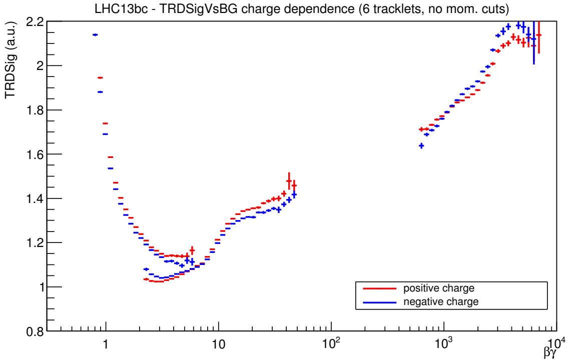
Cluster dependence

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



Additional Observations

- charge dependence


LHC13bc - TRDSigVsBG charge dependence (6 tracklets, no mom. cuts)
TRDSig (a.u.)
10^2
10^3
10^4
βγ
positive charge
negative charge