Analysis of CTRUE-B events for pPb @8.16TeV

Guillermo Contreras

Faculty of Nuclear Sciences and Physical Engineering,
Czech Technical University in Prague,
Prague, Czech Republic

September 18, 2017

Thanks a lot to Evgeny for suggestions, cross-checks, discussions ...
and an apparently infinite amount of patience!

Introduction

- The triggers used in UPC analyses have inputs which veto activity in the detector:
  p-Pb: CMUP14-B = 0MSL *0VBA *0UBA
  Pb-p: CMUP23-B = 0MUL *0UBC *0UGC *0VBA *0VGA *0SH2 *0VC5
- Furthermore, other vetoes are applied offline:
  - AD and V0 decision kEmpty in the lead side (A for p-Pb, C for Pb-p)
  - AD decision kEmpty in the proton side (C in p-Pb, A in Pb-p)
  - No activity in the neutron ZDC.
- The idea here is to study if these vetoes reject events that should be accepted, and in that case to compute a correction factor.
- To this end, events triggered by CTRUE-B have been analysed.
  (CTRUE-B is a trigger that is randomly fired when bunches from both sides cross the IP.)

Data used in the analyses

- Format: AOD.
- Periods:
  p-Pb in LHC16r: muon_calo_pass2.
  Pb-p in LHC16s: muon_calo_pass3.
- Runs according to

  https://twiki.cern.ch/twiki/bin/view/ALICE/MuonpPbQA2016.

  - LHC16r, 58 runs:

    265589, 265594, 265596, 265607, 265669, 265691, 265694, 265696, 265697, 265698, 265700, 265701, 265709, 265713, 265714, 265740, 265741, 265742, 265744, 265746, 265754, 265756, 265785, 265787, 265788, 265789, 265792, 265795, 265797, 265840, 266022, 266023, 266025, 266034, 266074, 266076, 266081, 266084, 266085, 266086, 266117, 266187, 266189, 266190, 266193, 266196, 266197, 266208, 266234, 266235, 266296, 266299, 266300, 266304, 266305, 266312, 266316, 266318.
  - LHC16s, 81 runs:

    266405, 266437, 266438, 266439, 266441, 266470, 266472, 266479, 266480, 266487, 266514, 266516, 266518, 266520, 266522, 266523, 266525, 266533, 266534, 266539, 266543, 266549, 266584, 266587, 266588, 266591, 266593, 266595, 266613, 266614, 266615, 266618, 266621, 266630, 266657, 266658, 266659, 266665, 266668, 266669, 266674, 266676, 266702, 266703, 266706, 266708, 266775, 266776, 266800, 266805, 266807, 266857, 266878, 266880, 266882, 266883, 266885, 266886, 266912, 266915, 266940, 266942, 266943, 266944, 266988, 266993, 266994, 266997, 266998, 267020, 267022, 267062, 267063, 267067, 267070, 267072, 267077, 267109, 267110, 267130, 267131.

Selection of CTRUE events

- Event in one of the good runs.
- Event triggered by CTRUE-B events

Numbering of trigger inputs

0VBA = 1;
0VGA = 4;
0UBA = 6;
0VBC = 2;
0UGC = 12;
0UBC = 7;
0VC5 = 8;
0SH2 = 9 (LHC16r); = 17 (LHC16s);

Checked for one run in LHC16r and one in LHC16s,
assumed to be valid for all other runs in LHC16r and LHC16s ...

Methods

- For each run find the fraction of events triggered by CTRUE-B that also fulfil another requirement denoted below by Cut. (See 'y-axis' of plots.)
- This fraction was plotted against μ, the probability of a random hadronic inelastic interaction.
- The error on each fraction was computed with the binomial formula used by ROOT.
- For some selections, data shows a two band structure. One band (blue circles in next plots) corresponds to runs where the TRD trigger was active. The runs where the TRD trigger was not active are shown with red circles in the next plots.
- The data from runs where there was no TRD trigger were fit to a function $f(\mu) = p_1\mu$ where p_1 was the fit parameter.

Runs where the TRD trigger was active

- LHC16r, 46 runs:

  265589, 265594, 265596, 265607, 265669, 265696, 265697, 265698, 265700, 265701, 265709, 265713, 265714, 265741, 265742, 265744, 265746, 265754, 265756, 265788, 265789, 265792, 265795, 265797, 266034, 266074, 266076, 266081, 266084, 266085, 266086, 266117, 266187, 266189, 266190, 266193, 266196, 266197, 266208, 266296, 266299, 266300, 266304, 266305, 266316, 266318.

- LHC16s, 19 runs:

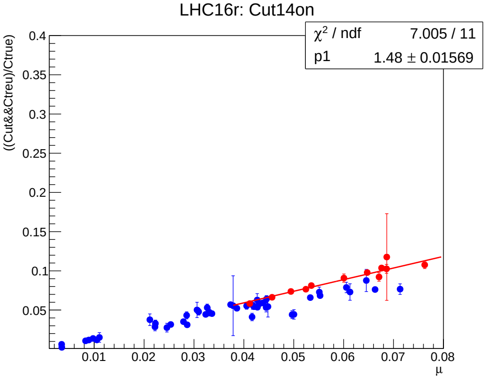
  266439, 266441, 266487, 266543, 266549, 266630, 266882, 266883, 266885, 266886, 266944, 266993, 266994, 266997, 266998, 267070, 267072, 267077, 267110.

Cuts used to analyse p-Pb data

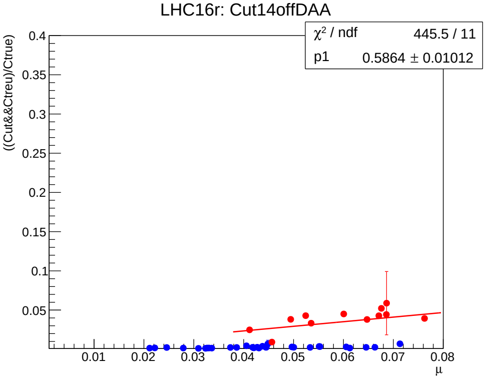
- Cut14on: to study the behaviour of the online tigger.
  Cut = 0UBA .or. 0VBA.
- **Cut14offDAA**: to study the ADA decision.
  Cut = .not.(0UBA .or. 0VBA) .and. ((ADADecision != 0)).
- **Cut14offDVA**: to study the V0A decision.
  Cut = .not.(0UBA .or. 0VBA .or. (ADADecision != 0)) .and. (V0ADecision != 0).
- **Cut14offDVC**: to study the V0C decision.
  Cut = .not.(0UBA .or. 0VBA .or. (ADADecision != 0) .or. (V0ADecision != 0)) .and. (ADCDecision != 0).
- **Cut14offZN**: to study the ZN decision.
  Cut = .not.(0UBA .or. 0VBA .or. (ADADecision != 0) .or. (V0ADecision != 0) .or. (ADCDecision != 0)) .and. ((ZNAEnergy>400) .or. (ZNCEnergy>18)).
- Cut14off: to study all offline vetoes.
  Cut = .not.(0UBA .or. 0VBA) .and. ((ADADecision != 0) .or. (V0ADecision != 0) .or. (ADCDecision != 0) .or. (ZNAEnergy>400) .or (ZNCEnergy>18)).
- Cut14: all vetoes at the same time.
  Cut = .0UBA .or. 0VBA .or (ADADecision != 0) .or. (V0ADecision != 0) .or. (ADCDecision != 0) .or. (ZNAEnergy>400) .or (ZNCEnergy>18).

The cut to be actually used in the analysis is Cut14.
Cuts Cut14on and Cut14off are used for a cross check.
Other cuts are used only for illustration.

Cut14on and Cut14offDAA

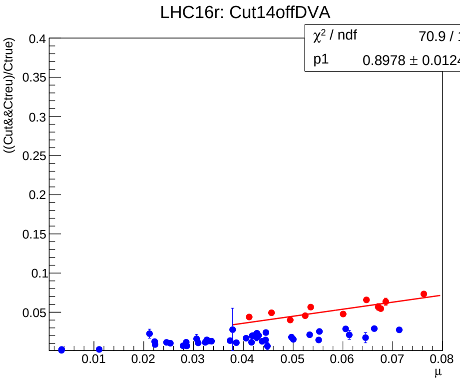


Probability that in an arbitrary bunch crossing either 0UBA or 0VBA fire in a run with a given value of μ.

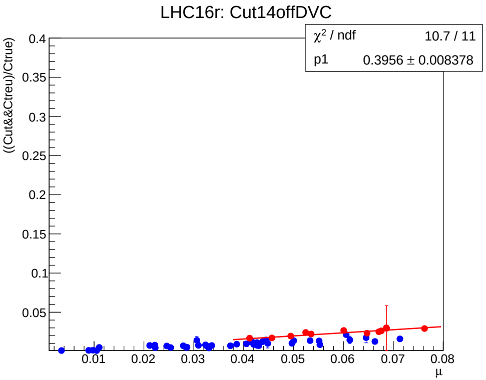


Probability that in an arbitrary bunch crossing ADA produces a decision different from kEmpty when neither 0UBA nor 0VBA fired.

Cut14offDVA and Cut14offDVC

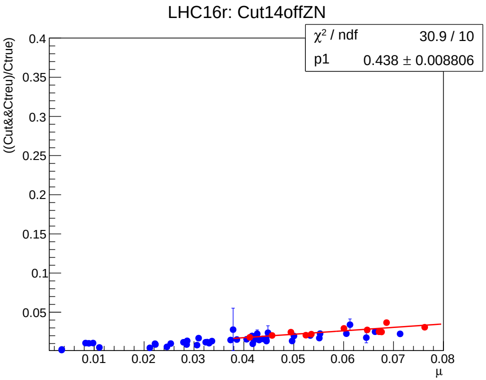


Probability that in an arbitrary bunch crossing V0A produces a decision different from kEmpty when neither 0UBA nor 0VBA fired and ADA decision was kEmpty.

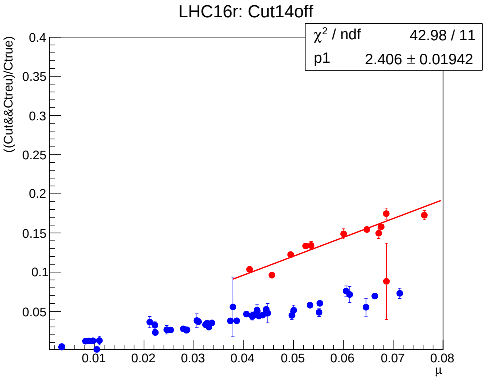


Probability that in an arbitrary bunch crossing V0A produces a decision different from kEmpty when neither 0UBA nor 0VBA fired and both ADA and V0A decisions were kEmpty.

Cut14offZN and Cut14off

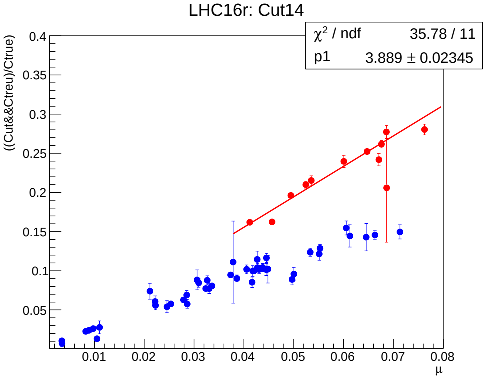


Probability that in an arbitrary bunch crossing ZN produces a signal above the selection threshold when neither 0UBA nor 0VBA fired and ADA, V0A and V0C decisions were kEmpty.

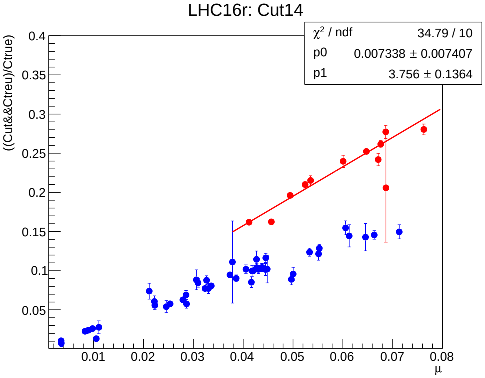


Probability that in an arbitrary bunch crossing any of the offline vetoes is rejected when neither 0UBA nor 0VBA fired.

Cut14



Probability that in an arbitrary bunch crossing any of the variables used as a veto would reject the UPC event.



The same, but fitting to $f = p_0 + p_1\mu$. Note that p_0 is consistent with being zero and both p_1 are consistent.

Comments

- Last page shows that it is justified to use $f = p_1\mu$ as the fitting function.
- The χ^2/dof is larger than one, Thus the fit error has been scaled by $\sqrt{\chi^2/\mathrm{dof}}$, resulting in a value for $p_1^{\mathrm{cut14}} = 3.889 \pm 0.042$ to be used below.
- Note that $p_1^{\mathrm{cut14}} = 3.889$ is compatible with $p_1^{\mathrm{cut14on}} + p_1^{\mathrm{cut14off}} = (2.406 \pm 0.019) + (1.48 \pm 0.016)$, so it is close to equivalent to correct first the online and then the offline contribution or to use directly cut14.
- The term $p_1\mu$ is the probability that an arbitrary interaction, in a run characterised by μ masks one of the veto elements. The probability that no veto element is masked, assuming that the effect from different collisions are not correlated, is given by

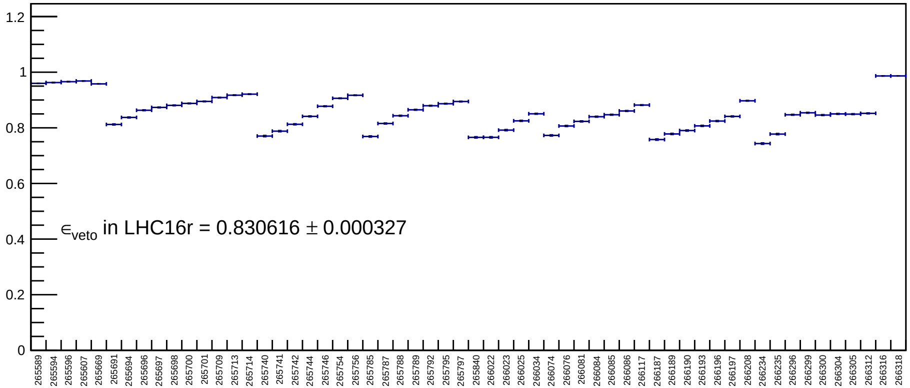
$$\epsilon^r_{\mathrm{veto}} = \exp(-p_1\mu(r)),$$

  where r denotes a run. This is the fraction of events that we see after the selection.
- The veto efficiency ϵ_{veto} of the full data sample is given then by

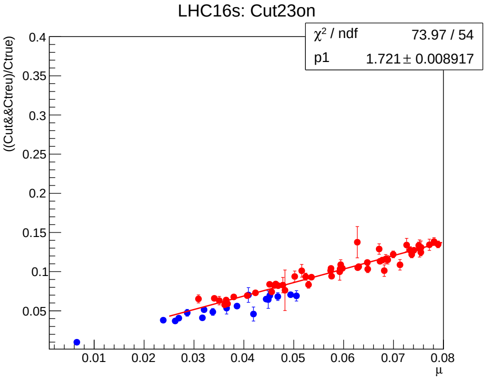
$$\epsilon_{\mathrm{veto}} = \sum_r w^r \epsilon^r_{\mathrm{veto}},$$

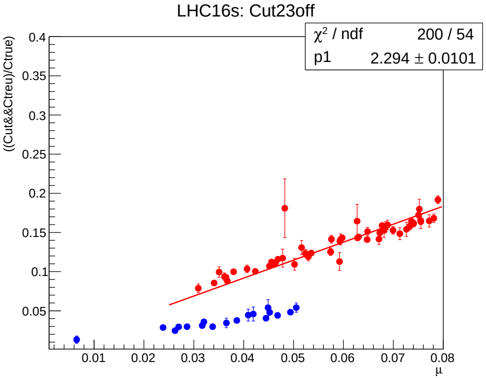
  where the weight w^r is the ratio of the luminosity in run r to the luminosity of the full sample in the given period. The error is given by using the fit errors from p_1.

Veto efficiency LHC16r

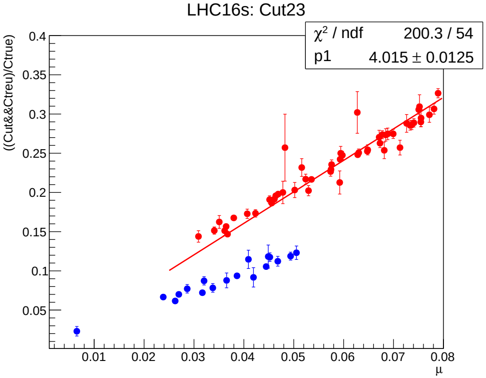


Cut23on and Cut23off

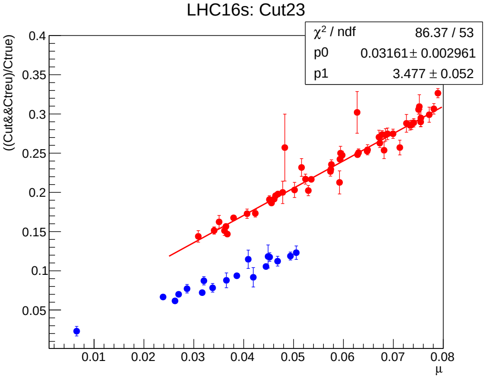




Cut23

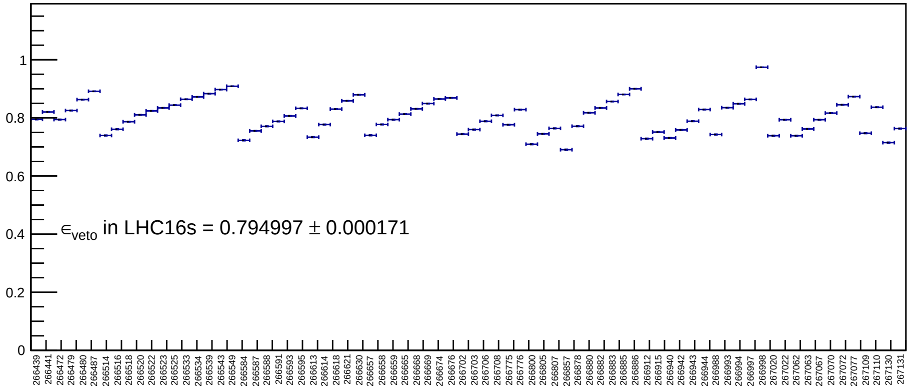


Probability that in an arbitrary bunch crossing any of the variables used as a veto would reject the UPC event.



The same, but fitting to $f = p_0 + p_1\mu$. Note that p_0 is consistent with being zero and both p_1 are consistent.

Veto efficiency LHC16s



Conclusions

- Large correction factors found from CTRUE analysis ...
- Unfortunately,
  - for LHC16r, this would make the cross section some 10% larger than expected from HERA and our on previous measurement,
  - and for LHC16s, the new factor is not enough to bring it to the expected levels (after applying this correction the cross section would still be some 20% lower than what we measured in Run 1 ...)
  - (Note that this are very rough estimates ... take them with a grain of salt ...)
- More work is needed to understand the data ...