

Status Weekly Ξ^0_c analysis meeting, Mar. 10, 2021

- **Status**
 - 1st progress report in D2H
 - a. No major/critical problem
 - b. Comments received
 - Preparing to update AliAnalysisTask code again
 - a. Plan to start from most updated version
 - b. List of updates

Report at D2H

- **D2H report at Mar. 5**

- Overall positive response: no major problem

- Comments received:

- a. **Xinye:** ratio to Jinjoo's result for cross-checked x-section (MB + [0, 100])

- * Planned to request bin by bin value to Jinjoo, but it looks QA required beforehand

- b. **Fabrizio:** # of events for normalization

- b-1. The # of events I use (about 1.83 B, obtained via personal method using 2D plot) should be smaller (~1.6 B) than the one obtained by *AliNormalizationCounter* (~1.83 B)

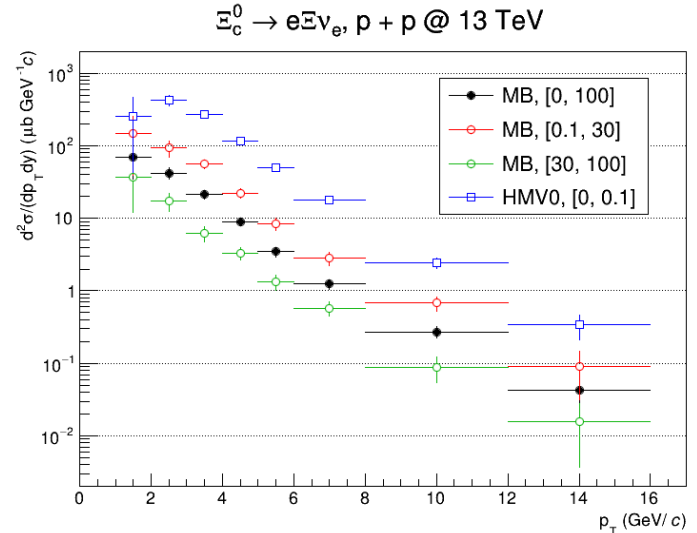
- b-2. Plan to check by using another *AliNormalizationCounter* object in main task

- (* checked evaluation by multiplicity is possible: only need to separate by trigger)

- c. **Andrea:**

- c-1. Mainly related to the “dropping point” at $1 < p_T < 2$, for HMV0 + [0, 0.1]

- c-2. Check eXi pair's inv. Mass distribution by RS/WS/RS-WS



Update Plan for AliAnalysisTask

- **List of items to be updated**

- **Minor:**

- a. Don't save triggerbit (assign 0 value) for MC
- b. Few garbage entries exist in train output: their run numbers are NOT makes sense at all:
Add hard-coded run number cut (252000 - 295000)

- **Add a boolean variable “fINEL” under EventTree: for later “INEL > 0” judgment**

- a. data: AliPPVsMultUtils::IsINELgtZERO(event) – should be equivalent to “kINT7 + # of SPD tracklets $\geq 1 + |\eta| < 1$ ”
- b. MC: require “IsPhysicalPrimary + IsCharged + $|\eta| < 1$ ”
 - * Quote from Junlee's code thanks to Prof. Lim
 - * **To use INEL > 0 condition on MC, must use “general purpose MC” rather than “Xic0 enriched MC”**

- **AliNormalizationCounter update**

- a. To use proper normalization factor
- b. Each counter object can have multiplicity info: plan to add a couple of counter more,
each for MB and HMV0 (keep Jinjoo's original object untouched)

- **pT binning unification for efficiency calculation histograms:**

- a. Denominator: hMCGenLevXic0_inc<W> (= hTrueXic0 at main task, binning = 7, bin)
- b. Numerator: hMCRecoLevXic0_<CUT>_<CUTFLAG> (= hGenXic0pT, binning = 9, widebin)

- **Debug: update all tree's float object definition from /f to /F (next page)**

Update Plan for AliAnalysisTask

- **Debug: /f to /F**

- Currently all trees in the AliAnalysisTask defined like this:

```
for (Int_t ivar=0; ivar<(Float_t)fTreeVariableName.size(); ivar++)  
{  
    fMCTree->Branch(fTreeVariableName[ivar].Data(), &fMCTreeVariable[ivar],  
    Form("%s/f",fTreeVariableName[ivar].Data()));  
}
```

- But according to ROOT homepage,

- a. F : a 32 bit floating point (Float_t)
- b. f : a 24 bit floating point with truncated mantissa (Float16_t)

- They actually make different between “actual number” and “saved number” - for example,

pTe = 1.29126 (saved) <-> 1.29116 (printed, actual data in the AOD)

echarge = -1 <-> -1

TOFnSigma = -0.168732 <-> -0.168728

TPCnSigma = 1.21021 <-> 1.21031

- It looks this data type makes difference starting from 4th - 5th digit under the point

* Silver lining is, according to Jinjoo, **most sensitive cut variable uses 3 digits level**

- We need crosscheck with new train run