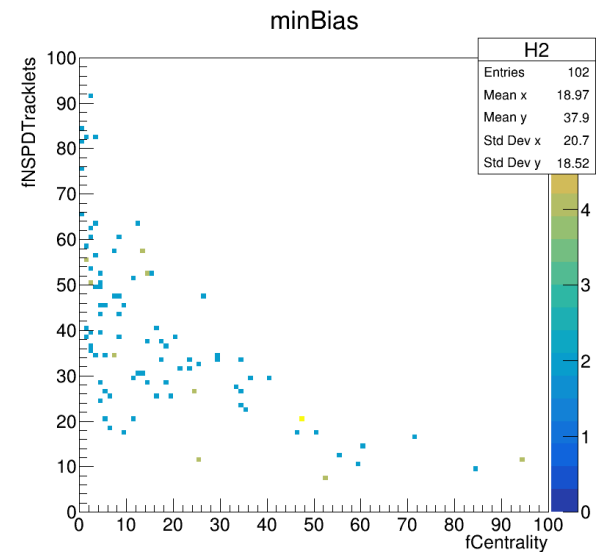
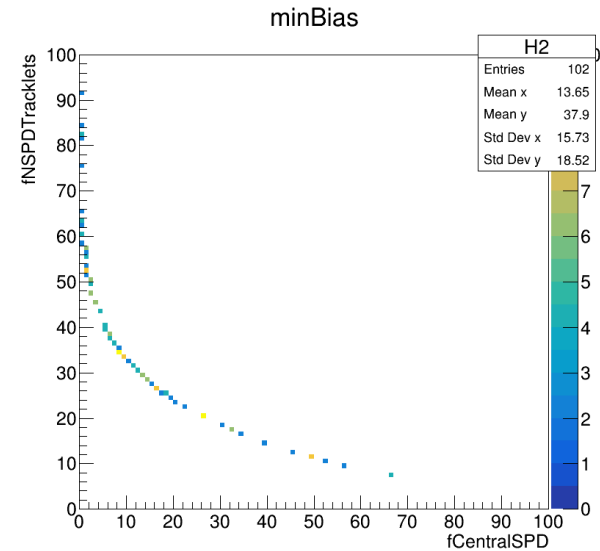
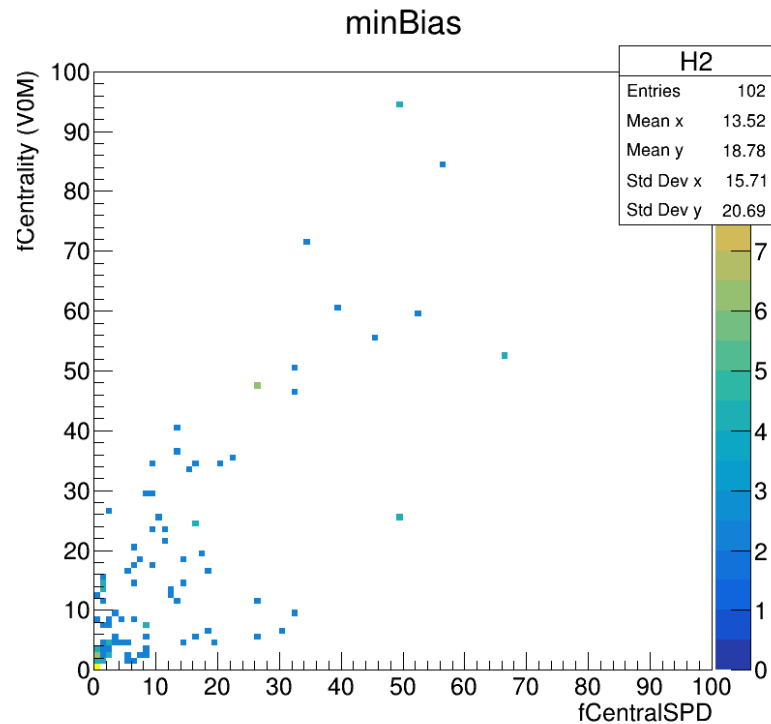


Status Weekly Ξ_c^0 analysis meeting, Sep. 3

- **Current status**

- Updated main task code (AliAnalysisTaskSEXic0SemiLeptonic.*):
 - a. Pushed to Git repository, but additional updates/debugging required
 - b. Updated items: trigger related features, SPD tracklets, zVtx (not pushed yet)
- Minor concern about ROOT versions being used among analyzers:
 - a. A couple of ROOT versions being used simultaneously:
ROOT5 for Grid/LegoTrain, ROOT6 (required for unfolding) for macro level analysis
 - b. I'll check ROOT5/6 consistency in Grid by picking a few data files up randomly
- No event-wise info in conventional tree
 - a. Current codes apply all event-wise cuts in Grid/LegoTrain level
 - b. Main tree for analysis (eXiTree) does NOT contains event related info:
 - b-1. It contains e-Xi pairs which passed cuts – not a problem at all for MB only analysis
 - b-2. Problematic if one want to distinguish each event at tree/macro level
 - b-3. Considering SPDtracklets based correction (raised by Prof. Lim),
double counting must be avoided (ex. 3 e-Xi pairs in an event)

Code Update Sanity check for SPD related variables



– Variables (saved under EventTree):

* Results from single AOD file of run 294200

- fCentrality: multiplicity percentile calculated by VOM
- fCentralSPD: mult. percentile calculated by SPD
(* I assume this should be used for HMSPD triggered events)
- fNSPDTracklets: # of SPD tracklets

Eventwise info problem A makeshift measure

- **Necessity of event-wise distinction**

- I plan to separate each event by fired trigger and its multiplicity percentile:
but, at the same time, I don't want to modify existing codes as much as I can
- I thought “running multiple LegoTrain jobs using only one trigger” as a solution, but
 - a. How about multiplicity percentile? Provide them as hard coded value and re-run?
 - b. To do the SPD tracklet based correction,
such event by event info (sync w/ e-XI pair) required after all
- As a makeshift solution,
 - a. Fill EventTree only when e-Xi pair is filled up
 - b. Add a variable (fNeXiPair) under EventTree which counts e-Xi pairs filled for the event:
 - b-1. If only one e-Xi pair filled for the event, the fNeXiPair will be always 0
 - b-2. If fNeXiPair is NOT 0, it means multiple pairs saved for the event

ex. If 3 pairs saved, same fCentrality, SPDTracklets, etc will be repeated w/ increasing fNeXiPair from 0 to 2

Plan for final level analysis

- **All modification should be compatible w/ existing codes**

- The updates I applied so far should NOT interfere w/ existing codes:

- a. The existing macros should be compatible to the newly generated output ROOT files

- b. [Now I re-writing Jinjoo's MakeROOTResultsXic0.C](#)

- b-1. It's de facto copy & paste of existing code but I'll add trigger / percentile feature

- b-2. Plan to produce output w/ applied options, for instance,

Generating ROOT file in following setup:

Input type: Data

Trigger: HMSPD (bit 8)

Multiplicity percentile: [0.1, 30.0]

Weight fit parameters: [1.0000, 1.0000]

-> output: out_Data_HMSPD_0.1to30.0.root

- b-3. No change in analysis histograms' name, etc

- b-4. Existing final level analysis macros should be compatible –

- repeat analysis w/ multiple output, then collect results by script

Backup Conventional tree (e-Xi) fillup

- **In AliAnalysis...SemiLeptonic.cxx ,**

```
void AliAnalysisTaskSEXic0Semileptonic::UserExec(Option_t*)
{
    AliVEvent *event = InputEvent();
    ...
    //Event Selection Ends

    ...
    auto nCascs = fEvt->GetNumberOfCascades();
    for (int icasc=0; icasc<nCascs; icasc++)
    {
        AliAODcascade *casc = ((AliAODEvent*)fEvt)->GetCascade(icasc);
        ... //casc cuts

        for (Int_t itrk=0; itrk<nTracks; itrk++)
        {
            AliAODTrack *trk = (AliAODTrack*) fEvt->GetTrack(itrk);
            ... //track cuts
            FillPairEleXi(casc, trk); //This Fill function has own cuts
        }
        ...
    }
    ...
}
```

Backup Multiple e-Xi pairs in an event example

```
root [1]
root [1]
root [1]
root [1] EventTree->Scan("fRunNumber:fCentrality:fCentralSPD:fNSPDTracklets:fNeXiPair");
*****
* Row * fRunNumbe * fCentrali * fCentrals * fNSPDTrac * fNeXiPair *
*****
* 0 * 294852 * 0.3849999 * 0.4879832 * 64 * 0 *
* 1 * 294852 * 0.5249999 * 0.1138210 * 77 * 0 *
* 2 * 294852 * 0.0515000 * 0.8419871 * 59 * 0 *
* 3 * 294852 * 0.0515000 * 0.8419871 * 59 * 1 *
* 4 * 294852 * 0.0515000 * 0.8419871 * 59 * 2 *
* 5 * 294852 * 0.0515000 * 0.8419871 * 59 * 3 *
* 6 * 294852 * 8.8500003 * 1.7128884 * 52 * 0 *
* 7 * 294852 * 8.8500003 * 1.7128884 * 52 * 1 *
* 8 * 294852 * 0.0335000 * 0.0081062 * 99 * 0 *
* 9 * 294852 * 0.0335000 * 0.0081062 * 99 * 1 *
* 10 * 294852 * 2.3499999 * 0.2842426 * 69 * 0 *
* 11 * 294852 * 0.9350000 * 0.0901937 * 79 * 0 *
* 12 * 294852 * 0.5350000 * 0.3537535 * 67 * 0 *
* 13 * 294852 * 0.0505000 * 0.1019716 * 78 * 0 *
* 14 * 294852 * 1.75 * 0.0371813 * 86 * 0 *
* 15 * 294852 * 6.4499998 * 2.2968947 * 49 * 0 *
* 16 * 294852 * 0.8650000 * 0.4377365 * 65 * 0 *
* 17 * 294852 * 0.0225000 * 0.2264022 * 71 * 0 *
* 18 * 294852 * 6.75 * 0.7565677 * 60 * 0 *
* 19 * 294852 * 6.75 * 0.7565677 * 60 * 1 *
* 20 * 294852 * 2.1500001 * 0.3537535 * 67 * 0 *
* 21 * 294852 * 2.1500001 * 0.2264022 * 71 * 0 *
* 22 * 294852 * 4.5500001 * 0.0627875 * 82 * 0 *
* 23 * 294852 * 0.3950000 * 0.0066161 * 101 * 0 *
* 24 * 294852 * 5.5500001 * 1.4048814 * 54 * 0 *
Type <CR> to continue or q to quit ==>
* 25 * 294852 * 3.5499999 * 0.1793265 * 73 * 0 *
* 26 * 294852 * 7.1500001 * 0.4377365 * 65 * 0 *
* 27 * 294852 * 7.1500001 * 0.4377365 * 65 * 1 *
* 28 * 294852 * 2.8499999 * 0.4377365 * 65 * 0 *
* 29 * 294852 * 0.2549999 * 0.8419871 * 59 * 0 *
* 30 * 294852 * 0.1749999 * 0.6788611 * 61 * 0 *
* 31 * 294852 * 2.4500000 * 0.3537535 * 67 * 0 *
* 32 * 294852 * 0.6549999 * 0.3537535 * 67 * 0 *
* 33 * 294852 * 0.0935000 * 0.8419871 * 59 * 0 *
* 34 * 294852 * 0.6050000 * 0.0712871 * 81 * 0 *
* 35 * 294852 * 0.7950000 * 0.1793265 * 73 * 0 *
* 36 * 294852 * 0.1850000 * 0.3938317 * 66 * 0 *
* 37 * 294852 * 0.1850000 * 0.3938317 * 66 * 1 *
* 38 * 294852 * 0.5450000 * 0.6788611 * 61 * 0 *
* 39 * 294852 * 6.4499998 * 0.4879832 * 64 * 0 *
* 40 * 294852 * 6.4499998 * 0.4879832 * 64 * 1 *
* 41 * 294852 * 0.8149999 * 0.2264022 * 71 * 0 *
* 42 * 294852 * 0.3549999 * 0.0421464 * 85 * 0 *
* 43 * 294852 * 0.1550000 * 0.8419871 * 59 * 0 *
* 44 * 294852 * 0.1049999 * 0.1276373 * 76 * 0 *
* 45 * 294852 * 0.1049999 * 0.1276373 * 76 * 1 *
* 46 * 294852 * 2.0499999 * 0.3171443 * 68 * 0 *
* 47 * 294852 * 2.0499999 * 0.3171443 * 68 * 1 *
* 48 * 294852 * 2.3499999 * 0.3537535 * 67 * 0 *
* 49 * 294852 * 0.7350000 * 0.3537535 * 67 * 0 *
Type <CR> to continue or q to quit ==>
```