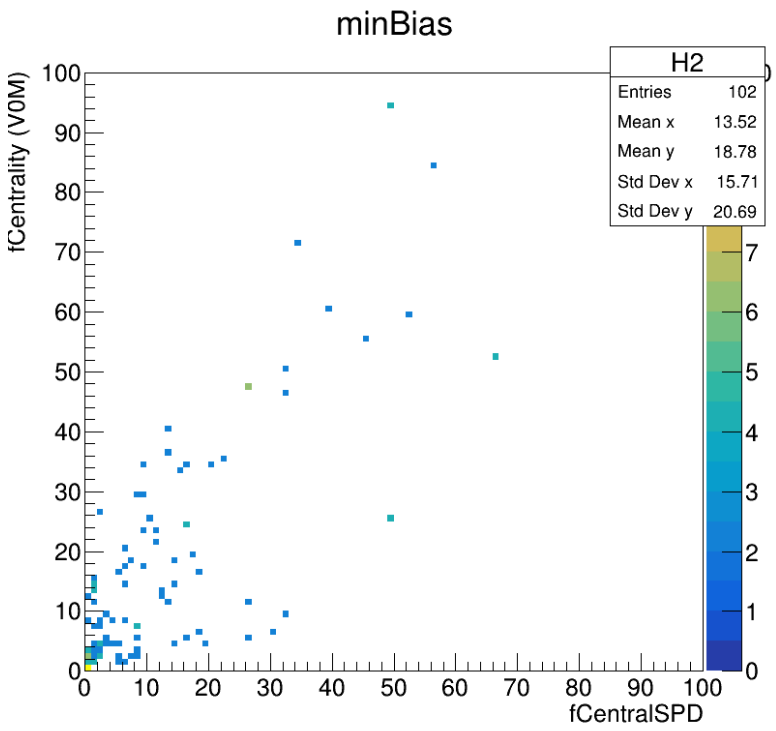
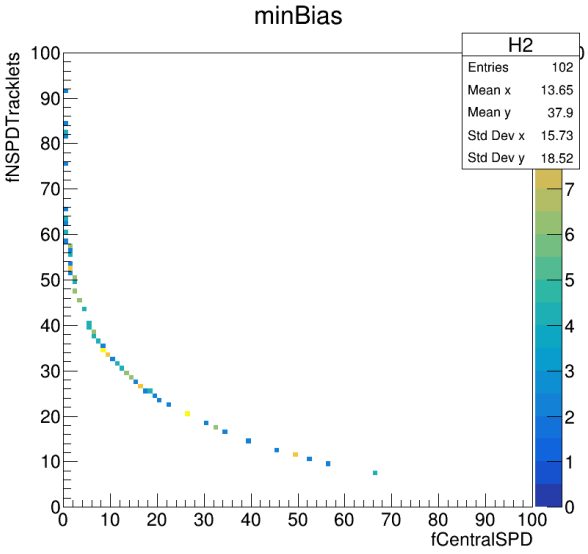
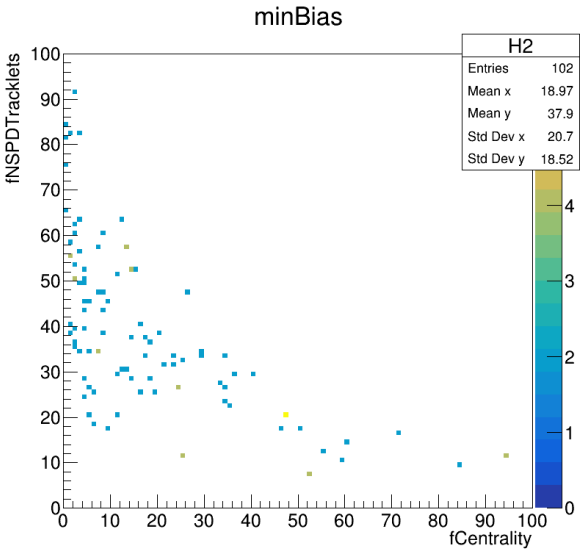
Status Weekly Ξ^0_c analysis meeting, Sep. 3

- **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

a. fCentrality: multiplicity percentile calculated by V0M

b. fCentralSPD: mult. percentile calculated by SPD

(* I assume this should be used for HMSPD triggered events)

c. 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 corded 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

LAST SLIDE

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