Status Weekly Ξ^0_c analysis meeting, Sep. 30, 2021, CKim

- **Crosscheck to Jinjoo's result**
  - Updates in last two weeks
    a. *AliAnalysisTask*: pileup rejection updated by new info based on SPD contributors (link)
       → pushed the code to Xic0 Git repository: I believe Jinjoo tested this code on her end
       → Jinjoo received another method (different function)
    b. Year by Year weighting factor calculation to deal with 2016-2018 dataset
       b-1. Each year's V0 xSec is slightly different, thus proper weighting required
       b-2. 1st method: get eXi pair's run number distribution and use it to calculate weighting factor (old and obsolete, but currently sitting on Git repo)
       b-3. 2nd method: get weighting factor from norm. factor from ANC object in each year's data (e.g., *AnalysisResults_2016.root->AliNormalizationCounter->GetNeventsForNorm()*)
  - Status and Short term plan
    a. Waiting for Jinjoo's train run finishes
    b. Once the new train run available, resume xCheck with Jinjoo's results, then report to D2H
    c. Items plan to report in next D2H:
       c-1. xCheck to Jinjoo's results
       c-2. Sanghoon's study on low pT: unstable behavior sue to BG dominance, thus discard the point
       c-2. INEL>0 effect on xSec (decomposed by trig + multiplicity) + Xic0/Lc vs. pT

V0 xSec function

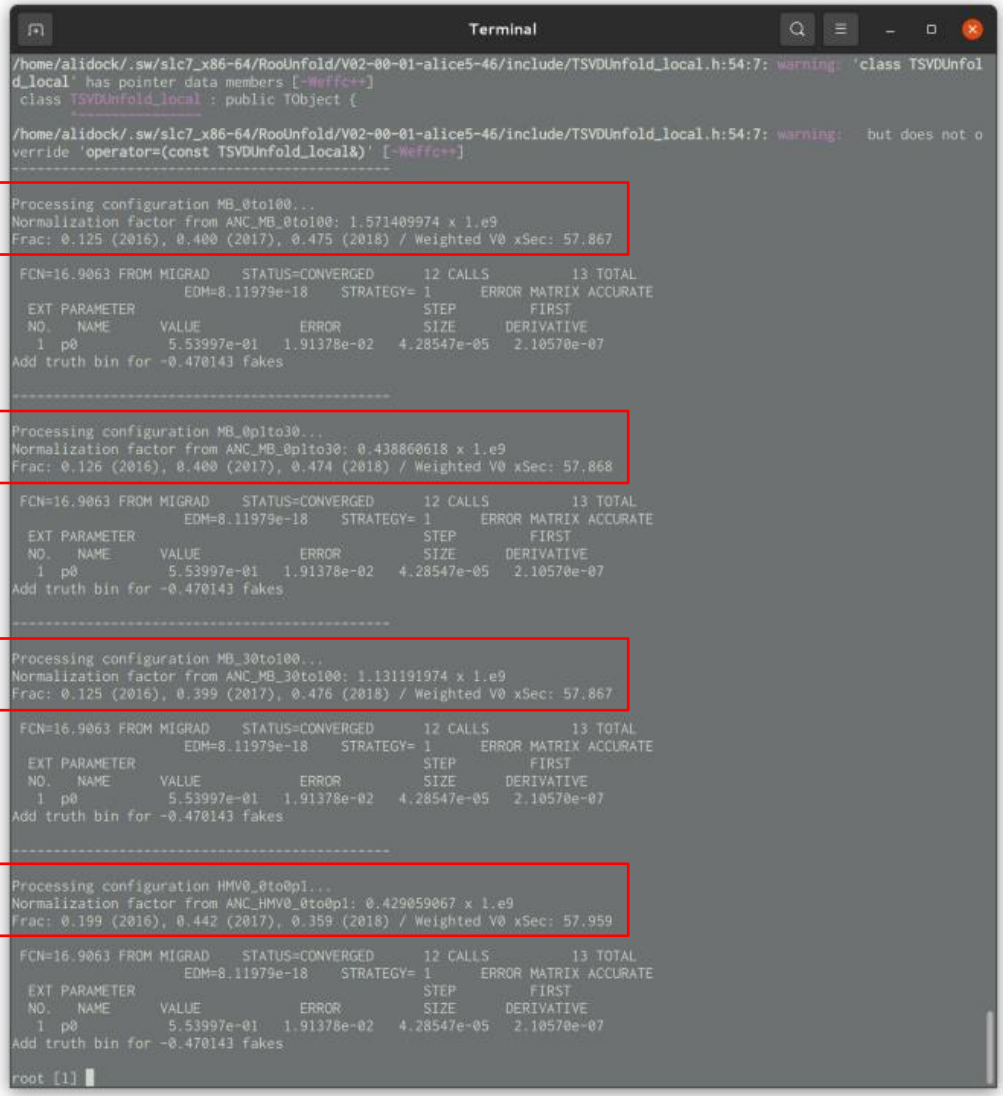
Hard corded, but wrapped in function

```
//------------------------------------------------
double GetV0xSec(int Year, const char* Type = "pp")
{
    double xSec = -999;

    // https://twiki.cern.ch/twiki/bin/viewauth/ALICE/EventNormalization#proton_proton_at_sqrt_s_13_T_AN1
    // Last checked Sep. 27, 2021

    // V0 xSec @ pp 13 TeV
    // pp 2016: 58.44 (mb) +- 1.0/1.6/1.9 (%, correlated, uncorrelated, and total)
    // pp 2017: 58.10      +- 1.2/2.4/2.7
    // pp 2018: 57.52      +- 1.0/1.8/2.1

    // V0 @ p-Pb 5.02 TeV (2013) (* no vdM in 2016 - use 2013 value)
    // p-Pb: 2.09 (b) +- 3.7 (%)
    // Pb-p: 2.12     +- 3.4

    if (Year==2016 && !strcmp(Type, "pp")==true) xSec = 58.44;
    if (Year==2017 && !strcmp(Type, "pp")==true) xSec = 58.10;
    if (Year==2018 && !strcmp(Type, "pp")==true) xSec = 57.52;

    if (Year==2016 && !strcmp(Type, "pPb")==true) xSec = 2.09;
    if (Year==2016 && !strcmp(Type, "Pbp")==true) xSec = 2.12;

    return xSec;
}//GetV0xSec
```

```
//Get normalization factor and year weighted V0xSec
const double NormF = GetNormFac("AnalysisResults_data.root", Form("ANC_%s", Suffix), true);
const double Frac2016 = GetNormFac("AnalysisResults_data_2016.root", Form("ANC_%s", Suffix))/NormF;
const double Frac2017 = GetNormFac("AnalysisResults_data_2017.root", Form("ANC_%s", Suffix))/NormF;
const double Frac2018 = GetNormFac("AnalysisResults_data_2018.root", Form("ANC_%s", Suffix))/NormF;

const double xSec2016 = GetV0xSec(2016, "pp");
const double xSec2017 = GetV0xSec(2017, "pp");
const double xSec2018 = GetV0xSec(2018, "pp");
const double xSecWgt = Frac2016*xSec2016 + Frac2017*xSec2017 + Frac2018*xSec2018;
const double V0xSec = xSecWgt * 1.E3; // multiplying 1.E3 - to match the scale? (from legacy code)
cout <<Form("Frac: %4.3f (2016), %4.3f (2017), %4.3f (2018) / Weighted V0 xSec: %5.3f\n\n",
        Frac2016, Frac2017, Frac2018, xSecWgt);
```

Weighting factor

Output by 2nd method (uses ANC object)



* May train output