All this has been copied to
/pnfs/uboone/persistent/users/markross/MiniBooNEDatasets2023/

Data files (Post CCQE selection)

- miniboone_data_2012_dataset.root (6.46e20 POT)
- miniboone_data_2018_dataset.root (6.38e20 POT)
  - This has been “corrected” referring to a 2% energy correction made on 2018 dataset, on “RecoEnuQE” only.
- Miniboone_data_combined_2012_2018_dataset.root

MC files (Corresponds to 41.10e20 POT I believe)

- miniboone_mc_filtered_passosc.root
  - All Miniboone MC that passes CCQE selection
- miniboone_mc_all_nue_nuebar.root
  - Full nue/nuebar MC file.
  - Contains branch “PassOsc” which if 1, corresponds to passing full CCQE selection
- miniboone_mc_dirt_postccqe.root
  - Dirt component AFTER ccqe selection (i.e PassOsc==1)
- output_osc_mc_detail_1..10
  - A series of 10 MC files containing all miniboone MC post 200tank hit cut with no additional cuts.
  - (note we pretty much exclusively worked with the filtered files, so not as much validation of this one)

Background “Types”

- DIRT
- π^0 Mis-id
- $\Delta \rightarrow N\gamma$
- Intrinsic ν_e from μ
- Intrinsic ν_e from K^+
- Intrinsic ν_e from K^0
- Other

The actual definition of these is annoyingly complex. We ported over code from MiniBooNE’s fortran “CombinedFit”. These functions can be found in

- CombinedTypes.h (just some enum’s)
  - They define the StackedBkgdType_t corresponding to the above 7 BKG’s
- CombinedFunctions.h/.cxx
  - The core “StackHistoBkgd” returns a StackedBkgdType_t given some input. This is the official way of catagorizing the backgrounds as seen in MiniBooNE publications

    - “Pi0Details” given a bunch of input, is the event a Pi0? You’d think this would be easy but no.
    - Also contains Enu_CCQE functions
- CombinedUtil.h/.cxx
    - Some convenience functions.

Then if you load up the MC MiniBooNE_CCQE as tree

```
For (int entry = 0; entry < (size_t)tree->GetEntries(); ++entry) {
        tree->GetEntry(entry);
        if(!PassOsc) continue;

            //
        // check if gamma is pi0
        //
        unsigned isPi0 = sp::Pi0Details(NFSP, FSPType_v,
             VertexX_v, VertexY_v, VertexZ_v,
             MomX_v, MomY_v, MomZ_v, MomT_v);

        //
        // get the background type ID
        //
        sp::StackedBkgdType_t bkg_type = sp::StackHistoBkgd(0,
             (bool)isPi0,
             NUANCEChan,
            NuType,
            NuParentID);


       //Then fill the event with “Weight” weight into the appropriate background category
bkg_type
}
```

Where the corresponding branches in MiniBooNE_CCQE ttree are

- NFSP : Number of final state particles
- FSPType : Final state particle type
- VertexX,Y,Z : Vertex position
- MomX,Y,Z,T : Four momentum
- NUANCEChan: Nuance channel of MC
    - CCQE==1, 93-97 DeltaRad, Can see all in CombinedTypes.h and nuance-instructions.txt
- NuType: Neutrino Flavour
    - 1 numu, 2 numubar, 3 nue, 4 nuebar

- NuParentID: Some neutrino parentage ID, i.e is is from muon or Kaon
    - These are NOT PDG codes, but Geant3 types. Can see codes in CombinedTypes.h
- Weight: MC weight for various MiniBooNE corrections