Energy Reconstruction Guide

Chrysovalantis Karagiannis

February 2023

For the energy reconstruction we use the EnergyRecoPredict toolchain. For the latest tools for the track length in the water and the energy reconstruction you can check my github https://github.com/KValantis/ToolAnalysis in the directories UserTools/DNNTrackLengthPredict/ and UserTools/BDTMuonEnergyPredict/ and UserTools/FindTrackLengthInWater/ respectively. For this toolchain we use the _pred scripts in these directories. Supposing that we have already done the training and testing we will focus on the reconstruction in this guide. For the EnergyRecoPredict toolchain we can check in the configfiles/EnergyReco/Predict directory to see the settings of the configfiles and open the ToolsConfig file to see which tools are being used. We observe that the first two tools being used are the LoadWCSim and LoadWCSimLAPPD tools which are cpp scripts which load the data of two root files we have set as the inputs in the LoadWCSimConfig and LoadWCSimLAPPDConfig files respectively. These tools eventually pass the data to the Store to be loaded by the next tool.

1 Vertex Reconstruction Tools

The next tools are the tools that do the vertex reconstruction. These tools are being used in the following order:

- MCParticleProperties

- MCRecoEventLoader
- DigitBuilder
- HitCleaner
- ClusterFinder
- EventSelector
- VtxSeedGenerator
- VtxSeedFineGrid(this tool is going to be added after Franklin's vertex reconstruction tools are pushed to the ANNIEsoft/Application)
- VtxExtendedVertexFinder

These tools load data from the Store and after some selection cuts or some calculations they upload the required data to the Store to be loaded by the next tool. You can add two extra tools in the toolchain to get the reconstructed MRD variables. These tools are the TimeClustering and FindMrdTracks tools. The TimeCLustering tool needs to be placed before the FindMrdTracks one in the ToolsConfig file because the MRDTimeClusters are created first in order to then find the MRD tracks in each cluster.

2 Track Length and Energy Reconstruction Tools

For the track length and energy reconstruction we use the following tools which are written in the ToolsConfig file after the vertex reconstruction tools in the following order since we want the variables from the vertex reconstruction in order to be able to reconstruct the track length in the water:

- FindTrackLengthInWater
- DNNTrackLengthPredict
- BDTMuonEnergyPredict
- BDTNeutrinoEnergyPredict

Here, DNNTrackLengthPredict, BDTMuonEnergyPredict and BDTNeutrinoEnergyPredict are all tool names and their tool type is PythonScript (check the ToolsConfig file). The FindTrackLengthInWater tool loads data from Store such as the reconstructed vertex in order to calculate the track length as the distance between the first and last Cherenkov photon emission point along the track and eventually stores the variables for the track length reconstruction using a DNN in the Store. More specifically for every hit, we have a reconstructed vertex and we use the hit position to estimate the distance between the first and last Cherenkov photon emission point along the track. In this tool we also have cuts to select well reconstructed events (with recovtxFOM>0) and events that stop in the MRD or pass through it (with TrueTrackLengthInMRD>0). This tool also creates the input .csv file with all the variables that we want to use for the track length reconstruction with the DNNTrackLengthPredict tool.

The name of the output .csv file from the FindTrackLengthInWater tool is specified in the corresponding config file. We need to remember the name in order to use this file as input for the DNNTrackLengthPredict tool. We define the input file of this tool in its config file named DNNTrackLengthPredictConfig as well as other variables such as the weights file name and the output file name. This tool creates a .csv file that has the same variables as the input one but also has the reconstructed track length in the water tank. This .csv file will be used as input for the next two tools BDTMuonEnergyPredict and BDTNeutrinoEnergyPredict.

In the config file BDTMuonEnergyPredictConfig and BDTNeutrinoEnergyPredictConfig we can set the name and path of the input file and also the name and path of the weights files and the output files. The output files are two .csv files with the energies of the muons and the neutrinos for each event. To run the EnergyRecoPredict toolchain you have to create a symlink and then run ./Analyse as seen below:

```
ln -s configfiles/EnergyReco/Predict/ToolChainConfig EnergyRecoPredict
./Analyse EnergyRecoPredict
```

Any of the variables in the config files can be accessed via m_variables after the recent cppyy integration and you can also access the DataModel via m_data. Check the python scripts to see exactly how.

The training of the ML algorithms has already been done using events from WCSim files so we already have the weight files. You can get them from my github in the directories UserTools/DNNTrackLength/stand_alone/weights/ and UserTools/EnergyReco/ stand_alone/weights. They are named as weights_bets.hdf5 and finalized_BDTmodel _forMuonEnergyWithRecoMRD.sav respectively. When the changes are pushed to the ANNIEsoft/Application branch you will just have to create the toolchain in the config files as described above.