OpDetAnalyzer

Variables at the particle generation stage

Branch name	Type	Description
nuvX, nuvY, nuvZ	std::vector<double>	True neutrino interaction vertex (in cm)
nuvT	std::vector<double>	True neutrino interaction time (in ns)
nuvE	std::vector<double>	True neutrino energy (in GeV)

They store information regarging all the primary neutrino interactions in a given readout window. Loop over the std::vector to get the energy, interaction vertex and interaction time corresponding to the i_{th} neutrino interaction.

Variables at the LArG4 stage (particle propagation+ionization+scintillation)

Variables regarding the MC particles propagated by G4

Branch name	Type	Description
E	std::vector<double>	Primary energy of each MC particle (in GeV)
process	std::vector<std::string>	Primary process of MC each particle
trackID	std::vector<int>	MC particle ID
motherID	std::vector<int>	ID of the mother MC particle
PDGCode	std::vector<int>	PDG code
InTimeCosmics	bool	Returns true if there is a cosmic interaction during the BNB spill

Example: Consider a 1207 MeV beam ν_μ. It undergoes a charged current interaction, leading to a μ^- (E=574 MeV) and a proton (E=1543 MeV). The first entry in the previous vectors will have the following values: E[0]=0.574, process[0]=primary, trackID[0]=1, motherID[0]=0, PDGCode[0]=13, E[1]=1.543, PDGCode[1]=2212... If e.g. the muon decays to a Michel electron, there will be an entry with PDGCode=11, motherID=1 and process=Decay.

Variables regarding the deposited energy

Branch name	Type	Description
energydep	std::vector<std::vector<double>>	Energy deposition (in MeV) at each G4 tracking step. It's saved for each MC particle
energydepX, energydepY, energydepZ	std::vector<std::vector<double>>	Location (in cm) of each energy deposition.
dEpromX, dEpromY, dEpromZ	std::vector<double>	Average X, Y, Z (in cm) location of the energy depositions. It's saved for the two TPCs (vector size is 2)
dEspreadX, dEspreadY, dEspreadZ	std::vector<double>	X, Y, Z standard deviation of the energy depositions

Following the previous example, to read the energy deposition values and their locations induced by the primary proton you can take energydep[1], energydepX[1]...

Variables regarding the scintillation photons

Branch name	Type	Description
SimPhotonsperOpChVUV	std::vector<double>	Number of true photons at each PD (VUV wavelength)
SimPhotonsperOpChVIS	std::vector<double>	Number of true photons at each PD (visible wavelength)
SimPhotonsLiteVUV	std::vector<std::vector<double>>	Photon arrival times at G4 stage (VUV). In ns.

Branch name	Type	Description
SimPhotonsLiteVIS	std::vector<std::vector<double>>	Photon arrival times at G4 stage (VIS). In ns.

SBND has 312 PDs, hence the size of the SimPhotonsperOpChVUV(VIS) is 312. You can obtain the number of VUV photons reaching the coated PMT with ID 144 by taking SimPhotonsperOpChVUV[144]. The size of the SimPhotonsLiteVUV(VIS) is also 312. Each vector in the 'vector of vecrtors' contains the times (in ns) in which each photon gets to the given PD. Imagine 567 VUV photons reach the coated PMT with ID 144. The size of the vector SimPhotonsLiteVUV[144] will be 567.

Variables at the digitization stage

Branch name	Type	Description
SignalsDigi	std::vector<std::vector<double>>	Digitized signals (ADC values)
StampTime	std::vector<double>	Start time of each digitized waveform (in μs)
OpChDigi	std::vector<int>	Associated PD ID

The PMT/XARAPUCA output signals (including electronic response) are stored in the previous vectors. We only save the regions of the waveforms going above a certain thereshold (region of interest or ROIs). The ADC values of each identifeid ROI correspond to an entry in the SignalsDigi "vector of vectors". Note that we may have more than one ROI per PD, so the size of the SignalsDigi branch will be in general different than the number of PDs (312). To get the start time and the channel corresponding to the i_{th} ROI get the StampTime and OpChDigi with index i_{th}.

Variables at the reconstruction stage

Deconvolution

Branch name	Type	Description
SignalsDeco	std::vector<std::vector<double>>	Deconvolved signals
StampTimeDeco	std::vector<double>	Start time of each digitized waveform (in μs)

Branch name	Type	Description
OpChDeco	std::vector<int>	Associated PD ID

Same as above, but storing the deconvolved signals.

Pulse finder (a.k.a. OpHits)

Branch name	Type	Description
nophits	int	Total number of reconstructed OpHits
ophit_opch	std::vector<int>	Optical channel corresponding to the reconstructed OpHit
ophit_peakT	std::vector<double>	Waveform bin in which the OpHit gets the maximum value (in μs)
ophit_width	std::vector<double>	With of the OpHit (in μs)
ophit_amplitude	std::vector<double>	Amplitude of the OpHit (in #PE units)
ophit_area	std::vector<double>	Area of the OpHit (in μs x #PE units)
ophit_pe	std::vector<double>	Reconstructed number of PE

Clustering among different PDs (a.k.a. OpFlash)

Legend

- PD: Photon Detector
- PE: Photoelectron
- BNB: Booster Neutrino Beam