RICERCA SU METODI DI CLUSTERING

OBIETTIVO: da un blob, una nuvoletta, che è una ricostruzione di una o più traccia, l’idea è fare un fit per capire che quella è una traccia e poi estrarne i parametri

CODICE PER RICOSTRUZIONE TRACCIA

Creare oggetto lpcsolver

Fare setpoints

E poi solve

E mi restituisce in output i punti che sono i punti medi dei voxel a cui è associato un valore di fotoni

19/10

Siamo dopo la ricostruzione, quindi abbiamo le tracce

GENIE mi dice l evento di neutrino cosa produce

EDEPSIM mi dice i depositi di energia

GEANT sim ottica

• DATI: STIMA AMPIEZZA FOTONI DI OGNI VOXEL

I valori di ampiezza di ogni voxel sono in formato pickle 🡪 funzione np.load(file name, allow\_pickle) per caricare il file

Dentro pickle c’è un dict che ha una sola chiave uuuid.

Poi dentro uuuid ci sono le chiavi degli eventi, scelgo la chiave

Poi dopo che metto l’ultima chiave, che rappresenta il numero di iterazione ottengo un array numpy 🡪 a me interessano le ultime iterazioni perché la traccia sarà quasi ricostruita alla fine

Alla fine ho un array numpy 3d con i valori di fotoni in ogni voxel. La dimensione del voxel è un parametro esterno che posso scegliere, mentre le dimensioni del volume le vedo dall array

I file sono nel path classico su hpc

Grain\_2row\_mup 🡪 pkl. Dentro input files ho il file root con la verità, e l’altro è il file di testo in cui ho il numero degli eventi.

• VERITA’ MONTECARLO DI ROOT DELL’AMPIEZZA DI OGNI VOXEL

SENSORSFILE ho la verità MC che mi dice le tracce vere delle particelle

Questi dati sono nel file sensors.root, stesso path dei dati simulati

• VERITA’ DI EDEPSIM DELLA DIREZIONE DI PRODUZIONE DELLA PARTICELLA

PRIMARIESFILE mi dice le direzione delle particelle

Edepsim.root…

Funzione è loadprimariesedepsim in mctruth.py per leggere il file primaries, questa funzione mi da un oggetto di tipo evento. Poi posso accedere agli attributi dell’evento che sono i VERTEX (quelli che mi interessano). Vertex ha altri attributi:

* Position che mi da il punto di intereazione
* Particles mi dice l’elenco delle particelle

Ogni particella poi avrà due attributi: uno mi dice che particella è e l’altro mi dice il suo momento.

Account DUNE (nu\_at\_fnal) computing resources

Ssh neutrino01

nomeutente@131.154.161.32

produzioni sono all’indirizzo:

storage/gpfs\_data/neutrino/SAND-LAr/SAND-Lar-OPTICALSIM-PROD/GRAIN/grain\_2row\_mup/input/edep-sim.root

numu\_LBNF\_GV2\_prod\_1.genie.edep-sim.root

SCRIPT SH PER PLOTTARE I VARI FILES

Nel file bash posso variare i parametri come il cut e altri parametri che spiego di seguito:

N.B. la ricostruzione ha problemi a ricostruire vicino alle camere, -m è l opzione che taglia 5 voxel vicino alle pareti che hanno le camere -> anche io devo fare questo taglio. Il problema è dovuto al modo con cui si stima il numero di fotoni in ogni voxel. Questo metodo fallisce quando si è vicino alla camera mentre funziona bene lontano.

- v: per dare la dimensione del voxel. La default è 10 mm ora è 12

-eq: fai l’equalizzazione dell’istogramma dei voxel. Vuol dire che tutti i voxel hanno ampiezza compresa tra 0 e 1. Questa opzione rende l’algoritmo più robusto al taglio che decidiamo di applicare sull’ampiezza dei voxel. Proviamo con questa opzione con un taglio 0.5 o 0.6. ovviamente l’analisi deve essere fatta con i valori di ampiezza giusti non quelli equalizzati.

-s salva l’immagine invece di mostrarla

DA INSTALLARE:

lxml pacchetto che gestisce i file xml •

pytraversal

pickle •

argparse •

scikit\_image •

Mi servono le geometrie, devo includerle quando eseguo il file: le posso prendere da baltig

GRAIN\_box31\_2row 🡪 geometria di GRAIN

4x4cam questo all’inizio 🡪 geometria di un cubo, nel caso venga fatta una simulazione iniziale su una geometria diversa da GRAIN