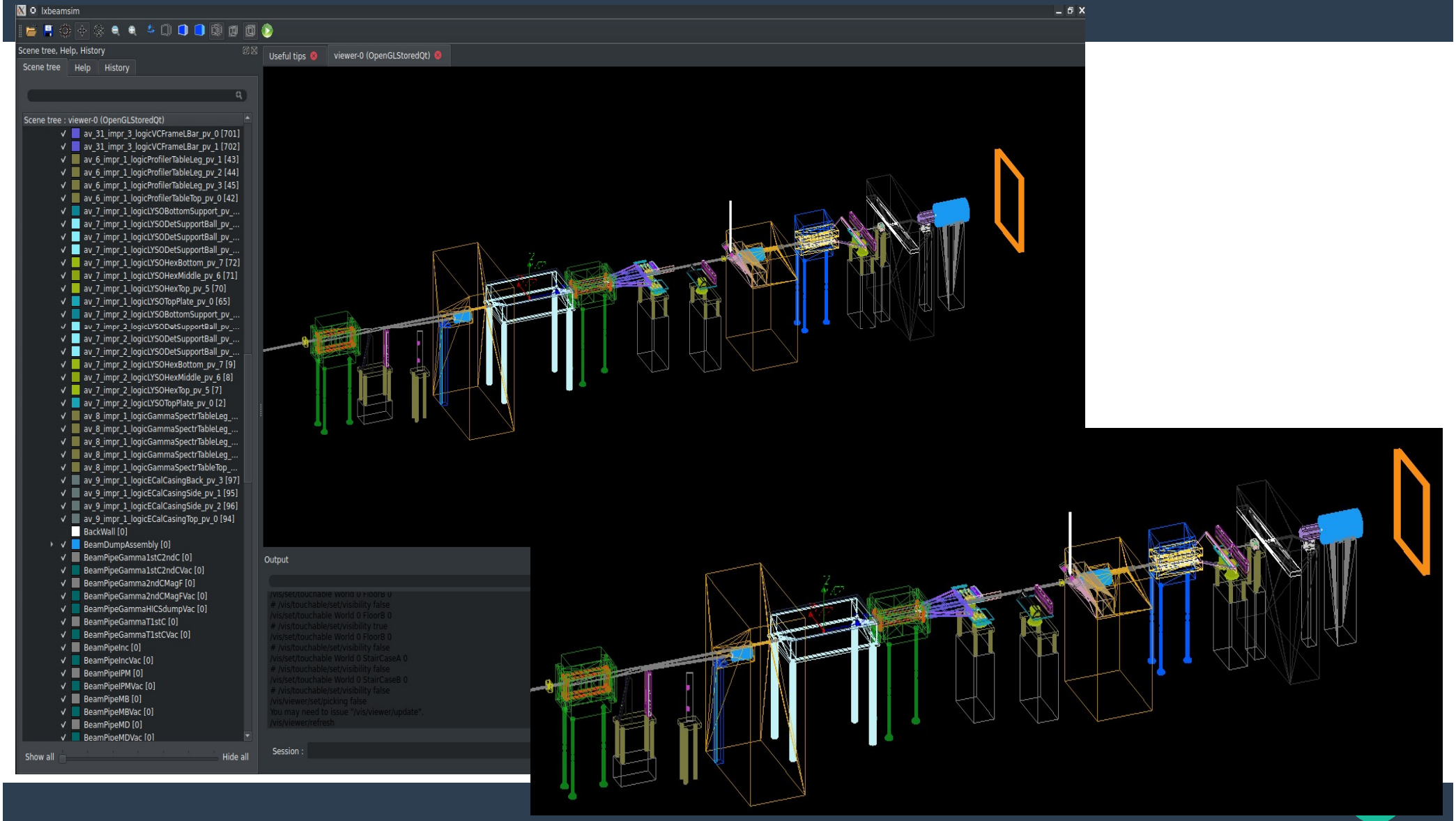
First glimpse on data from simulations

Mihai Potlog
Institute of Space Science, Bucharest

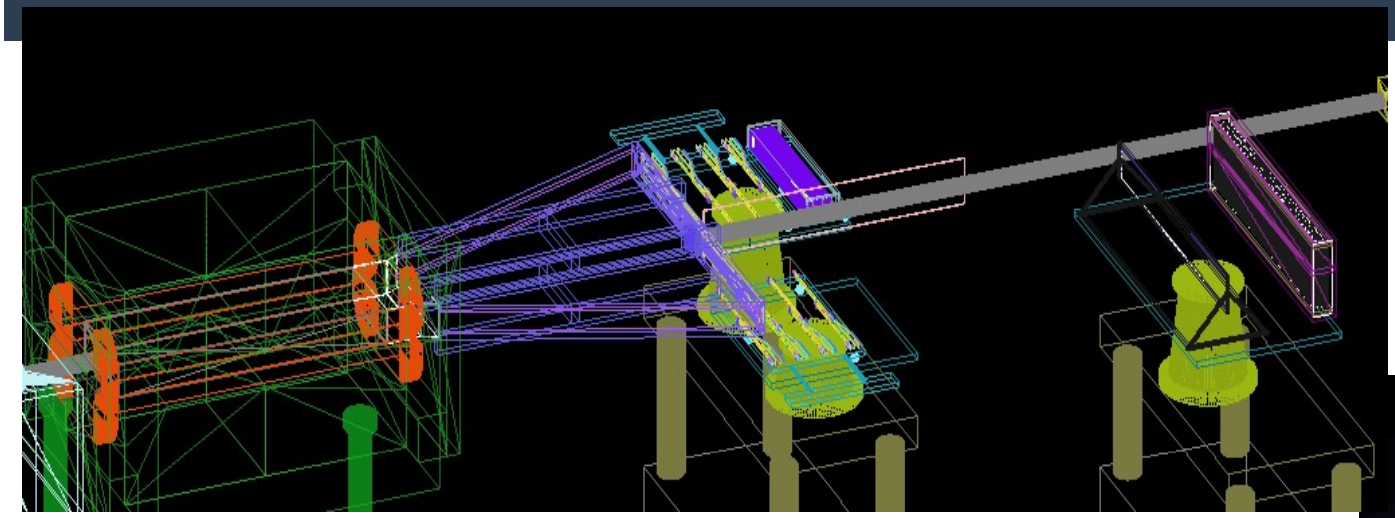
First steps ...

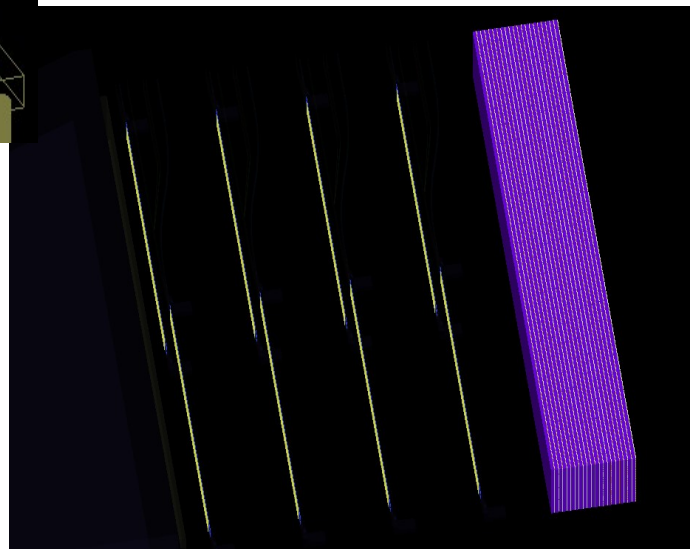
- Get acquainted with the LUXE experimental setup & understand the simulation implementation
- Check the .root format of simulation output
- Investigate the analysis chain
- Plot first graphs to verify the compliance of simulated data

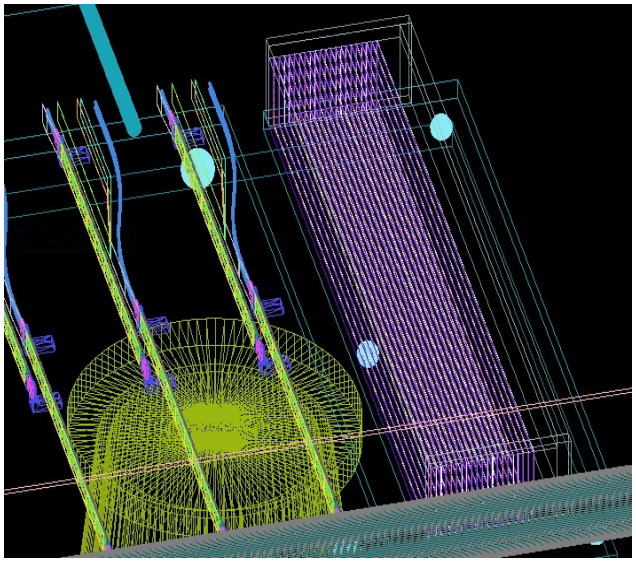
Get acquainted with the LUXE experimental setup



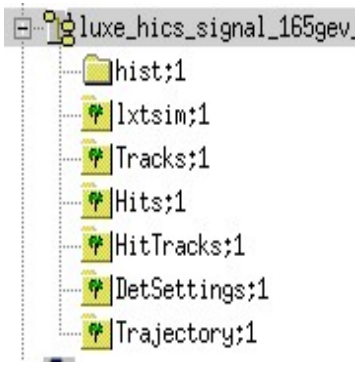
Understand the simulation implementation





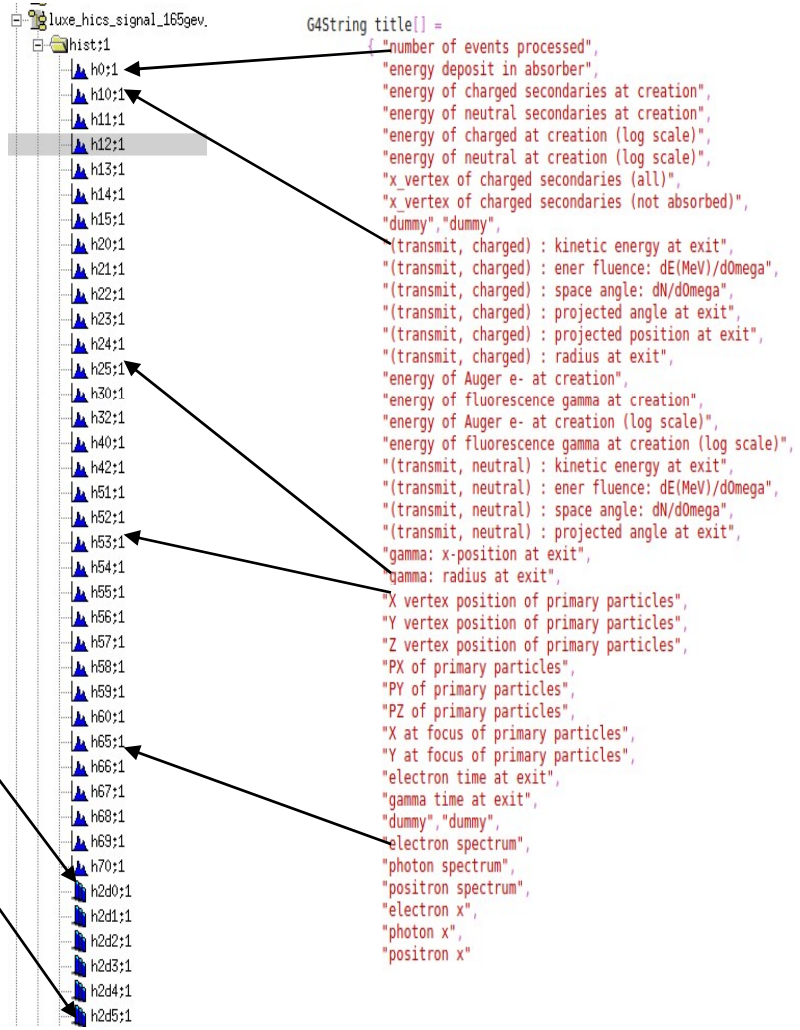


Check the .root format of simulation output



2-D histograms

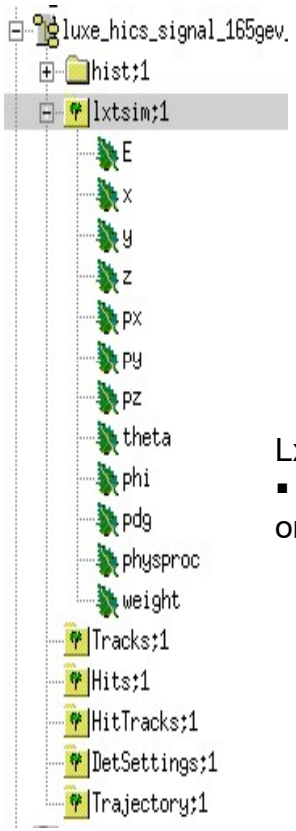
```
G4String title2D[] =
        { "(transmit, charged) : projected position at exit vs energy",
          "(transmit, charged vs neutral): kinetic energy e-gamma at exit",
          "(transmit, charged): x, y position at exit",
          "(transmit, neutral): x, y position at exit",
          "Z vertex position vs E of charged produced after the magnet",
          "Z vertex position vs E of neutral produced after the magnet",
          "electron polar anvgle vs E at exit",
          "photons polar anvgle vs E at exit",
          "photons polar anvgle vs E at exit, urad scale",
          "photons position x, y at exit",
          "primary electron x, vs dx/dz",
          "electron x, vs dx/dz at IP",
          "positron polar anvgle vs E at exit"
        };
```




```
G4String title[] =
        { "number of events processed",
          "energy deposit in absorber",
          "energy of charged secondaries at creation",
          "energy of neutral secondaries at creation",
          "energy of charged at creation (log scale)",
          "energy of neutral at creation (log scale)",
          "x_vertex of charged secondaries (all)",
          "x_vertex of charged secondaries (not absorbed)",
          "dummy","dummy",
          "(transmit, charged) : kinetic energy at exit",
          "(transmit, charged) : ener fluence: dE(MeV)/dOmega",
          "(transmit, charged) : space angle: dN/dOmega",
          "(transmit, charged) : projected angle at exit",
          "(transmit, charged) : projected position at exit",
          "(transmit, charged) : radius at exit",
          "energy of Auger e- at creation",
          "energy of fluorescence gamma at creation",
          "energy of Auger e- at creation (log scale)",
          "energy of fluorescence gamma at creation (log scale)",
          "(transmit, neutral) : kinetic energy at exit",
          "(transmit, neutral) : ener fluence: dE(MeV)/dOmega",
          "(transmit, neutral) : space angle: dN/dOmega",
          "(transmit, neutral) : projected angle at exit",
          "gamma: x-position at exit",
          "gamma: radius at exit",
          "X vertex position of primary particles",
          "Y vertex position of primary particles",
          "Z vertex position of primary particles",
          "PX of primary particles",
          "PY of primary particles",
          "PZ of primary particles",
          "X at focus of primary particles",
          "Y at focus of primary particles",
          "electron time at exit",
          "gamma time at exit",
          "dummy","dummy",
          "electron spectrum",
          "photon spectrum",
          "positron spectrum",
          "electron x",
          "photon x",
          "positron x"
```

1-D histograms

Check the .root format of simulation output




```
analysisManager->CreateNtuple("lxtsim", "Bremsstrahlung photons at IP");
analysisManager->CreateNtupleDColumn(0, "E");
analysisManager->CreateNtupleDColumn(0, "x");
analysisManager->CreateNtupleDColumn(0, "y");
analysisManager->CreateNtupleDColumn(0, "z");
analysisManager->CreateNtupleDColumn(0, "px");
analysisManager->CreateNtupleDColumn(0, "py");
analysisManager->CreateNtupleDColumn(0, "pz");
analysisManager->CreateNtupleDColumn(0, "theta");
analysisManager->CreateNtupleDColumn(0, "phi");
analysisManager->CreateNtupleIColumn(0, "pdg");
analysisManager->CreateNtupleIColumn(0, "physproc");
analysisManager->CreateNtupleDColumn(0, "weight");
analysisManager->FinishNtuple(0);
```

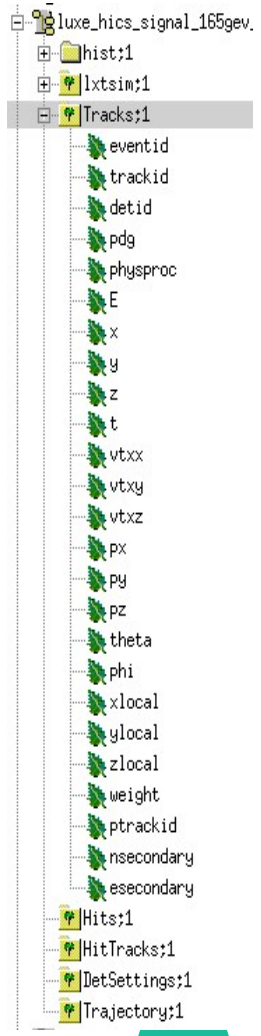
Lxtsim tree:

- didn't find any information on what is stored here

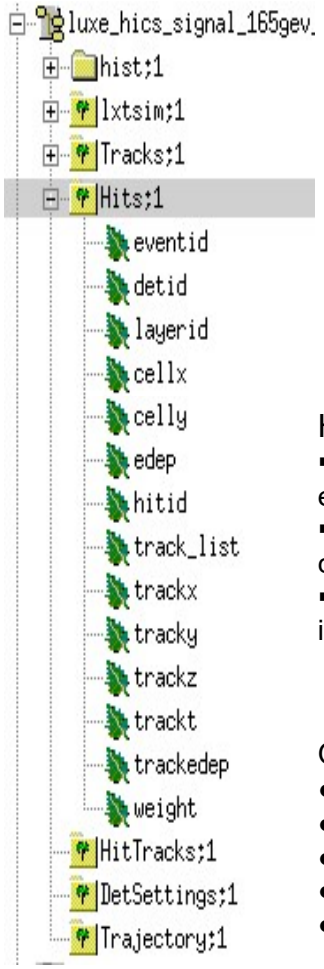
```
analysisManager->CreateNtuple("Tracks", "Tracks hitting volumes marked for track interception");
analysisManager->CreateNtupleIColumn(1, "eventid");                          //0
analysisManager->CreateNtupleIColumn(1, "trackid",  fvolTrackIMap[16]);      //1
analysisManager->CreateNtupleIColumn(1, "detid",    fvolTrackIMap[17]);      //2
analysisManager->CreateNtupleIColumn(1, "pdg",      fvolTrackIMap[18]);      //3
analysisManager->CreateNtupleIColumn(1, "physproc", fvolTrackIMap[19]);      //4
analysisManager->CreateNtupleDColumn(1, "E", fvolTrackDMap[0]);              //5
analysisManager->CreateNtupleDColumn(1, "x", fvolTrackDMap[1]);              //6
analysisManager->CreateNtupleDColumn(1, "y", fvolTrackDMap[2]);              //7
analysisManager->CreateNtupleDColumn(1, "z", fvolTrackDMap[3]);              //8
analysisManager->CreateNtupleDColumn(1, "t", fvolTrackDMap[4]);              //9
analysisManager->CreateNtupleDColumn(1, "vtxx", fvolTrackDMap[5]);           //10
analysisManager->CreateNtupleDColumn(1, "vtxy", fvolTrackDMap[6]);           //11
analysisManager->CreateNtupleDColumn(1, "vtxz", fvolTrackDMap[7]);           //12
analysisManager->CreateNtupleDColumn(1, "px",   fvolTrackDMap[8]);           //13
analysisManager->CreateNtupleDColumn(1, "py",   fvolTrackDMap[9]);           //14
analysisManager->CreateNtupleDColumn(1, "pz",   fvolTrackDMap[10]);          //15
analysisManager->CreateNtupleDColumn(1, "theta",  fvolTrackDMap[11]);        //16
analysisManager->CreateNtupleDColumn(1, "phi",    fvolTrackDMap[12]);        //17
analysisManager->CreateNtupleDColumn(1, "xlocal", fvolTrackDMap[13]);        //18
analysisManager->CreateNtupleDColumn(1, "ylocal", fvolTrackDMap[14]);        //19
analysisManager->CreateNtupleDColumn(1, "zlocal", fvolTrackDMap[15]);        //20
analysisManager->CreateNtupleDColumn(1, "weight");                           //21
analysisManager->CreateNtupleIColumn(1, "ptrackid", fvolTrackIMap[20]);      //22
analysisManager->CreateNtupleIColumn(1, "nsecondary", fvolTrackIMap[21]);    //23
analysisManager->CreateNtupleIColumn(1, "esecondary", fvolTrackDMap[22]);    //24
analysisManager->FinishNtuple(1);
```

Tracks tree:

- Information about tracks which entered in detectors from its outside
- collected in EventAction



Check the .root format of simulation output




```
analysisManager->CreateNtuple("Hits", "Hits in sensitive detectors");
analysisManager->CreateNtupleIColumn(2, "eventid");
analysisManager->CreateNtupleIColumn(2, "detid");
analysisManager->CreateNtupleIColumn(2, "layerid");
analysisManager->CreateNtupleIColumn(2, "cellx");
analysisManager->CreateNtupleIColumn(2, "celly");
analysisManager->CreateNtupleDColumn(2, "edep");
analysisManager->CreateNtupleIColumn(2, "hitid");
analysisManager->CreateNtupleIColumn(2, "track_list", fvHitTrackList);
analysisManager->CreateNtupleDColumn(2, "trackx", ftrackx);
analysisManager->CreateNtupleDColumn(2, "tracky", ftracky);
analysisManager->CreateNtupleDColumn(2, "trackz", ftrackz);
analysisManager->CreateNtupleDColumn(2, "trackt", ftrackt);
analysisManager->CreateNtupleDColumn(2, "trackedep", ftrackedep);
analysisManager->CreateNtupleDColumn(2, "weight");
analysisManager->FinishNtuple(2);
```

Hits tree contains:

- Energy deposited in a sensitive element of the detector;
- Associated Sensor ID, cell_x, cell_y, layer;
- List of tracks ID which contributed in energy deposition

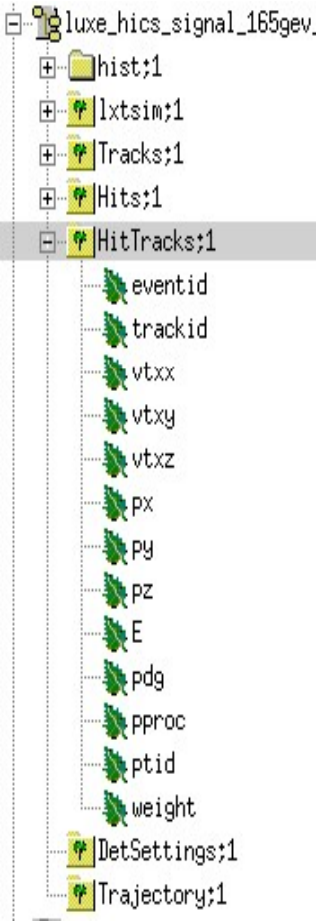
One can study:

- Deposited energy in a cell;
- Apply noise (or other) cuts to deposited energy;
- Detector occupancy at a hits level;
- Cluster size and energy;
- Detector occupancy at a clusters level;

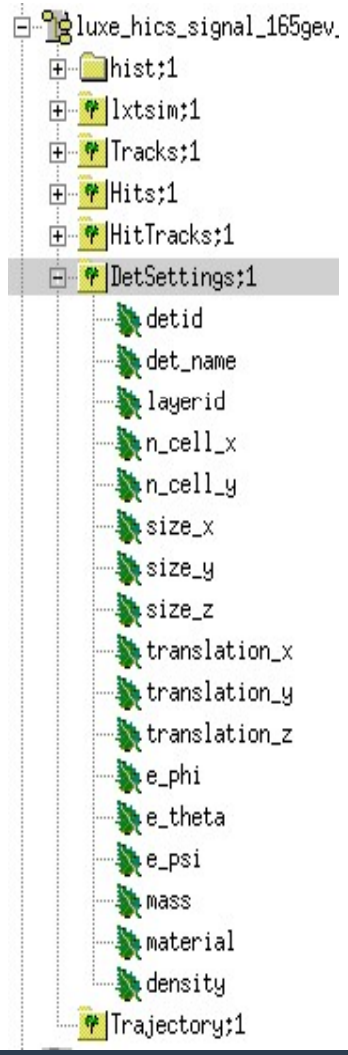
HitTracks tree contains:

- Information about tracks which produced hits in sensitive detector

```
analysisManager->CreateNtuple("HitTracks", "Tracks which produced hits
analysisManager->CreateNtupleIColumn(3, "eventid");
analysisManager->CreateNtupleIColumn(3, "trackid", fvTracks);
analysisManager->CreateNtupleDColumn(3, "vtxx", fVtxx);
analysisManager->CreateNtupleDColumn(3, "vtxy", fVtxy);
analysisManager->CreateNtupleDColumn(3, "vtxz", fVtxz);
analysisManager->CreateNtupleDColumn(3, "px", fPx);
analysisManager->CreateNtupleDColumn(3, "py", fPy);
analysisManager->CreateNtupleDColumn(3, "pz", fPz);
analysisManager->CreateNtupleDColumn(3, "E", fE);
analysisManager->CreateNtupleIColumn(3, "pdg", fPDG);
analysisManager->CreateNtupleIColumn(3, "pproc", fPhysProc);
analysisManager->CreateNtupleIColumn(3, "ptid", fPTId);
analysisManager->CreateNtupleDColumn(3, "weight");
analysisManager->FinishNtuple(3);
```



Check the .root format of simulation output

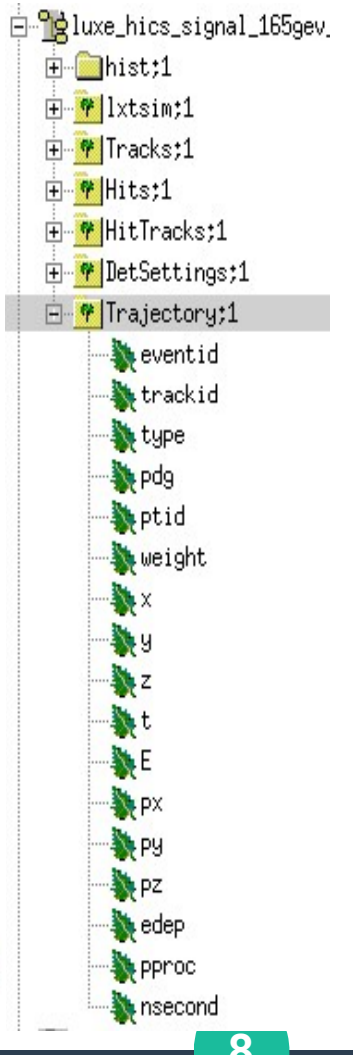



```
analysisManager->CreateNtuple("DetSettings", "Sensitive detector settings");
analysisManager->CreateNtupleIColumn(4, "detid");
analysisManager->CreateNtupleSColumn(4, "det_name");
analysisManager->CreateNtupleIColumn(4, "layerid");
analysisManager->CreateNtupleIColumn(4, "n_cell_x");
analysisManager->CreateNtupleIColumn(4, "n_cell_y");
analysisManager->CreateNtupleDColumn(4, "size_x");
analysisManager->CreateNtupleDColumn(4, "size_y");
analysisManager->CreateNtupleDColumn(4, "size_z");
analysisManager->CreateNtupleDColumn(4, "translation_x");
analysisManager->CreateNtupleDColumn(4, "translation_y");
analysisManager->CreateNtupleDColumn(4, "translation_z");
analysisManager->CreateNtupleDColumn(4, "e_phi");
analysisManager->CreateNtupleDColumn(4, "e_theta");
analysisManager->CreateNtupleDColumn(4, "e_psi");
analysisManager->CreateNtupleDColumn(4, "mass");
analysisManager->CreateNtupleSColumn(4, "material");
analysisManager->CreateNtupleDColumn(4, "density");
analysisManager->FinishNtuple(4);
```

DetSettings tree*:*
- sensitive detector information
- allow track/shower reconstruction and particles reconstruction
- collected in RunAction

Trajectory tree:
- didn't find any information on what is stored here

```
analysisManager->CreateNtuple("Trajectory", "Track trajectories");
analysisManager->CreateNtupleIColumn(5, "eventid");
analysisManager->CreateNtupleIColumn(5, "trackid");
analysisManager->CreateNtupleIColumn(5, "type");
analysisManager->CreateNtupleIColumn(5, "pdg");
analysisManager->CreateNtupleIColumn(5, "ptid");
analysisManager->CreateNtupleDColumn(5, "weight");
analysisManager->CreateNtupleDColumn(5, "x", fTrajectoryD[0]);
analysisManager->CreateNtupleDColumn(5, "y", fTrajectoryD[1]);
analysisManager->CreateNtupleDColumn(5, "z", fTrajectoryD[2]);
analysisManager->CreateNtupleDColumn(5, "t", fTrajectoryD[3]);
analysisManager->CreateNtupleDColumn(5, "E", fTrajectoryD[4]);
analysisManager->CreateNtupleDColumn(5, "px", fTrajectoryD[5]);
analysisManager->CreateNtupleDColumn(5, "py", fTrajectoryD[6]);
analysisManager->CreateNtupleDColumn(5, "pz", fTrajectoryD[7]);
analysisManager->CreateNtupleDColumn(5, "edep", fTrajectoryD[8]);
analysisManager->CreateNtupleIColumn(5, "pproc",   fTrajectoryI[0]);
analysisManager->CreateNtupleIColumn(5, "nsecond", fTrajectoryI[1]);
analysisManager->FinishNtuple(5);
```



Check the .root format of simulation output

"Tracks"	eventid	trackid	detid	pdg	phys-proc	E/px/py/pz	t/x/y/z	vtxx/y/z	theta/phi	x/y/zlocal	p-trackid	nsec-ondary	**weight**
Data type	integer	vector	vector	vector	vector	vector	vector	vector	vector	vector	vector	vector	**double**
"Hits"	eventid	**track_list**	detid	**hitid**		**edep**	layerid	cellx/y	trackx/y/z				**weight**
Data type	integer	**vector**	vector	**integer**		**double**	integer	integer	vector				**double**
"HitTracks"	eventid	trackid		pdg	pproc	E/px/py/pz		vtxx/y/z			ptid		**weight**
Data type	integer	vector		vector	vector	vector		vector			vector		**double**

```
#include <vector>

void read_vector()
{
   TFile *f = new TFile("luxe_hics_signal_165gev_3000nm_jeti40_cv12_em0_alw_1mu_cut_tv4_hv1_633.root");
   TTree *t1 = (TTree*)f->Get("Tracks");

   TH2F *hist_xy = new TH2F("hist_xy","Cluster energy", 100, 0, 500, 100, 0, 30);
   TH1F *h_vtxz = new TH1F("h_vtxz", "vtxz", 100, 0, 500);

   std::vector<Float_t> *vtxx = new std::vector<Float_t>();
   std::vector<Float_t> *vtxy = new std::vector<Float_t>();
   std::vector<Float_t> *vtxz = new std::vector<Float_t>();

   t1->SetBranchAddress("vtxx", &vtxx);
   t1->SetBranchAddress("vtxy", &vtxy);
   t1->SetBranchAddress("vtxz", &vtxz);

   Int_t nentries = (Int_t)t1->GetEntries();
   //cout << "Entries: " << nentries << endl;

   for (Int_t i=0; i<nentries;i++)
        {
        t1->GetEntry(i);

        Float_t hVtxz = 0;
        Int_t h_index = -1;

        if (vtxz->size() > 0)  {
            for (Int_t j = 0; j < vtxz->size(); j++){
                if ((*vtxz)[j] > hVtxz){
                    hVtxz = (*vtxz)[j];
                h_index = j;
                }
            }
            hist_xy->Fill((*vtxx)[h_index], (*vtxy)[h_index]);
            h_vtxz->Fill((*vtxz)[h_index]);
        }
    }

}
```

Investigate the analysis chain

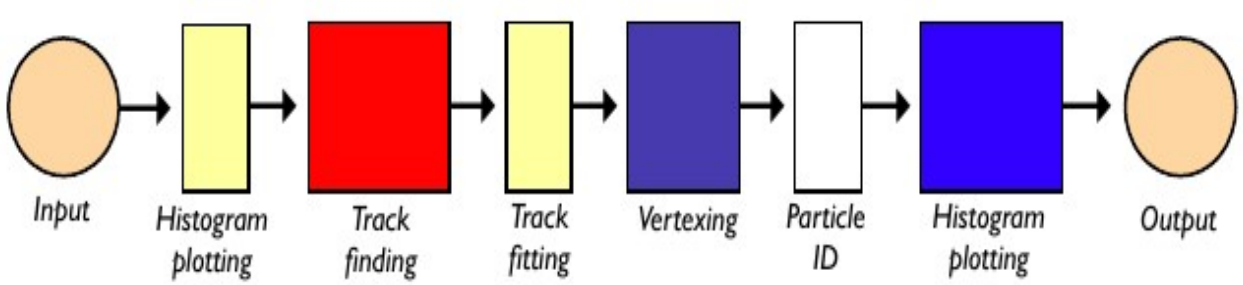
Analysis chain

Energy deposition in sensitive element of the detector – Hit:
- It comes as Chip ID, channel ID, Stave ID, ... signal => (Sensor ID, cell_x, cell_y, layer)

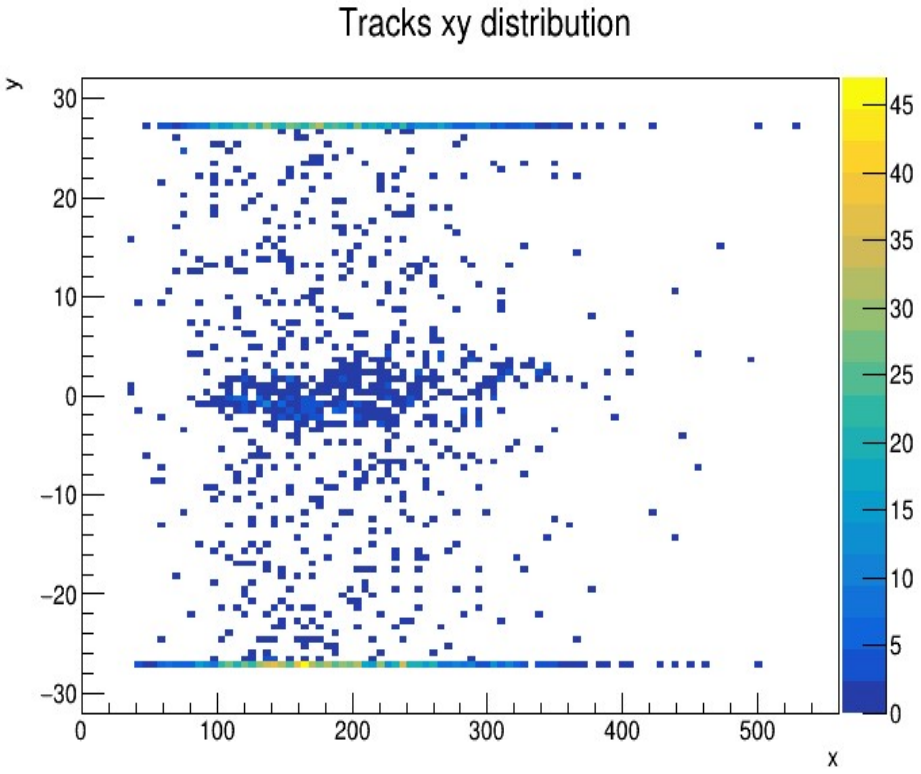
Collection of Hits in a sensor assigned to a single particle – Cluster:
- {(Sensor ID, cell_x, cell_y, layer)} => local: (x,y,z)

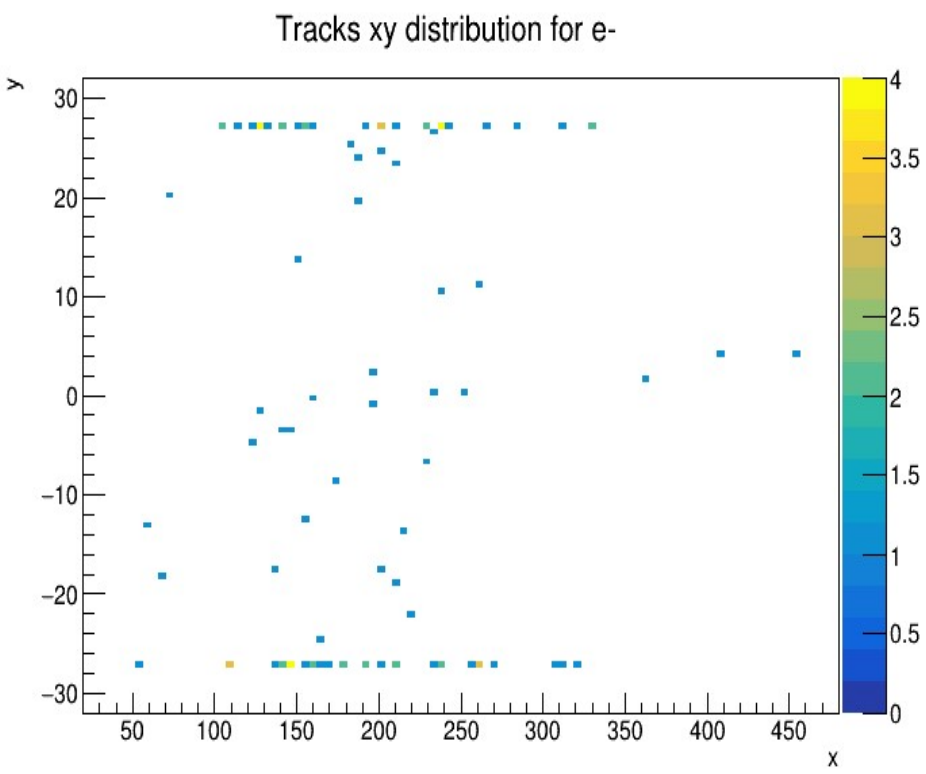
Assign clusters to particles track:
- local: (x,y,z) => global (x,y,z) + Track/Shower reconstruction (Fit)

Reconstruct particles:
- Particle type, vertex, momentum, ...

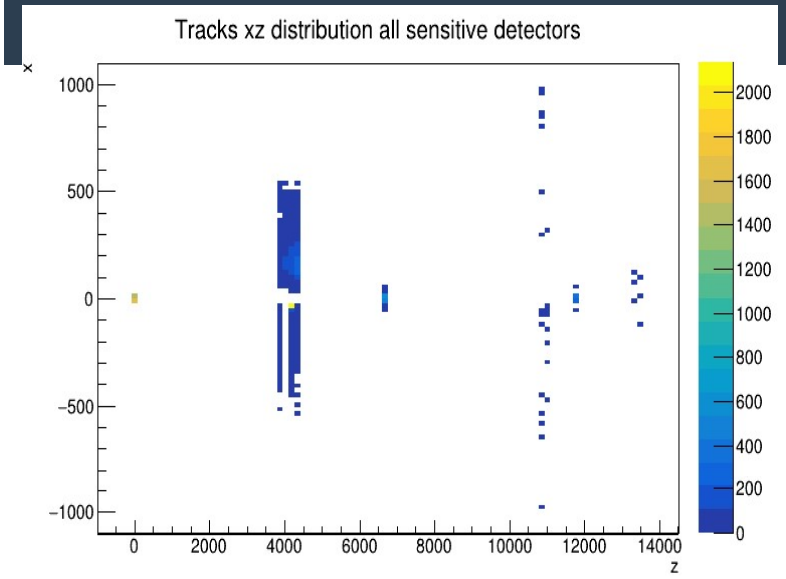


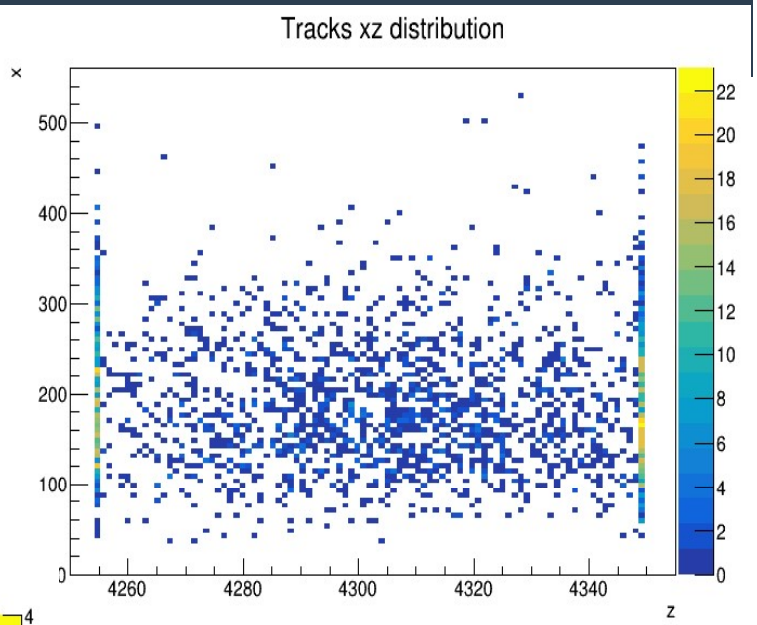
Plot first graphs to verify the compliance of simulated data

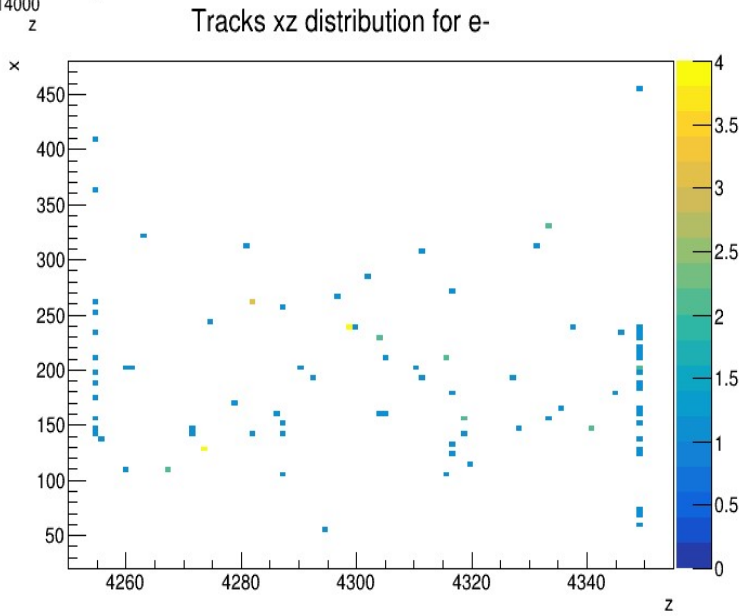




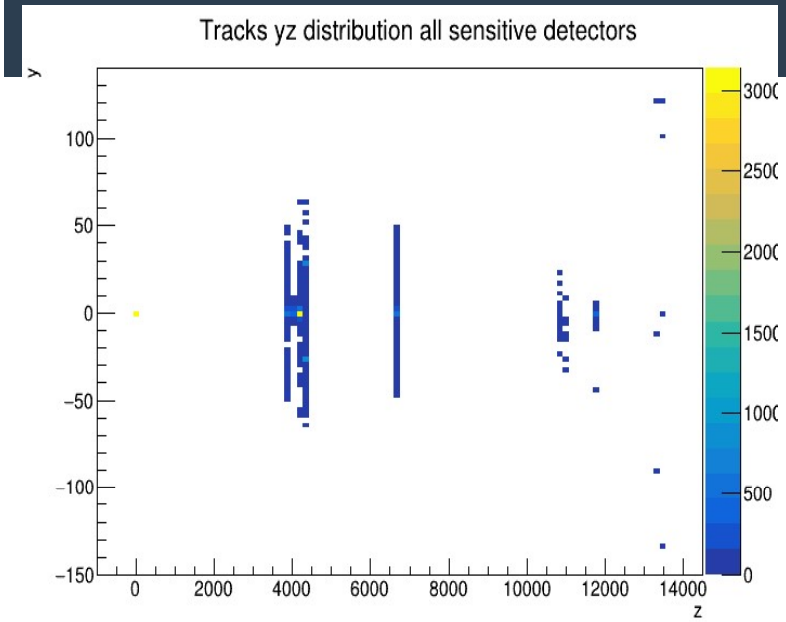
Plot first graphs to verify the compliance of simulated data

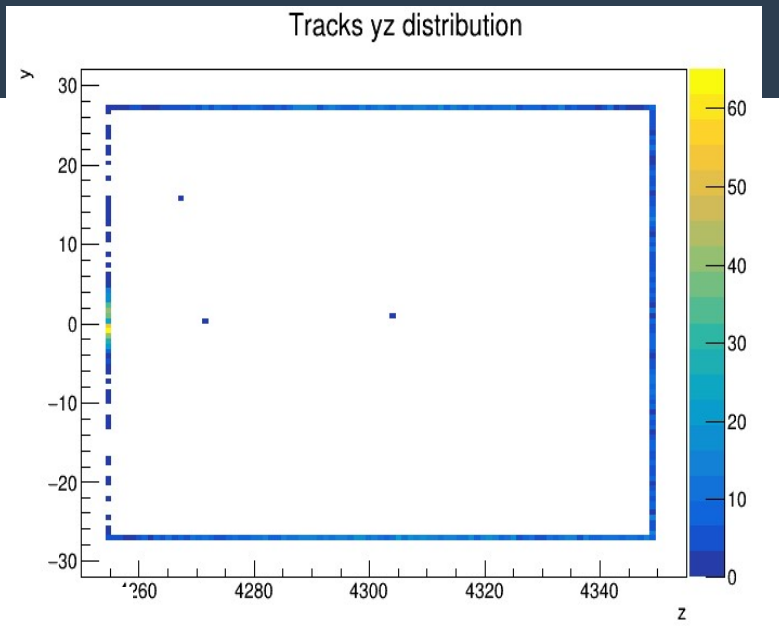


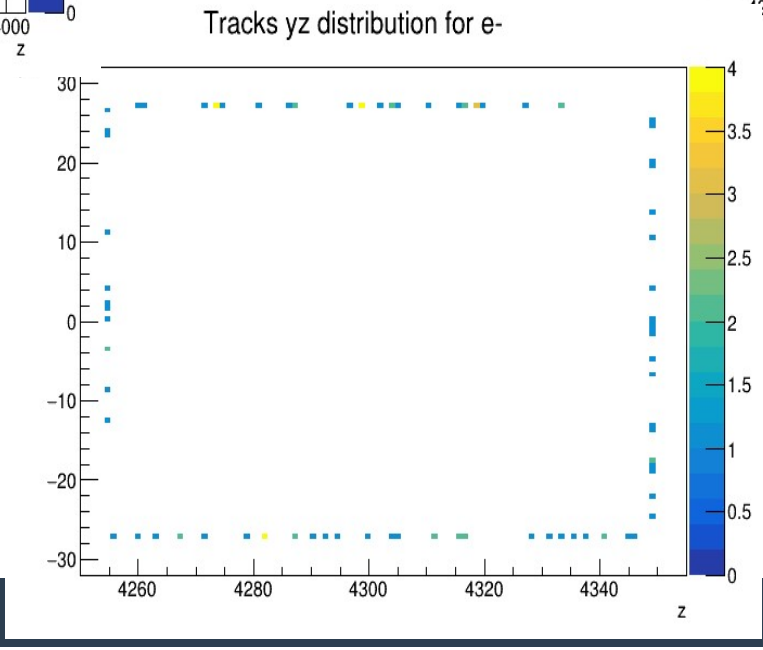




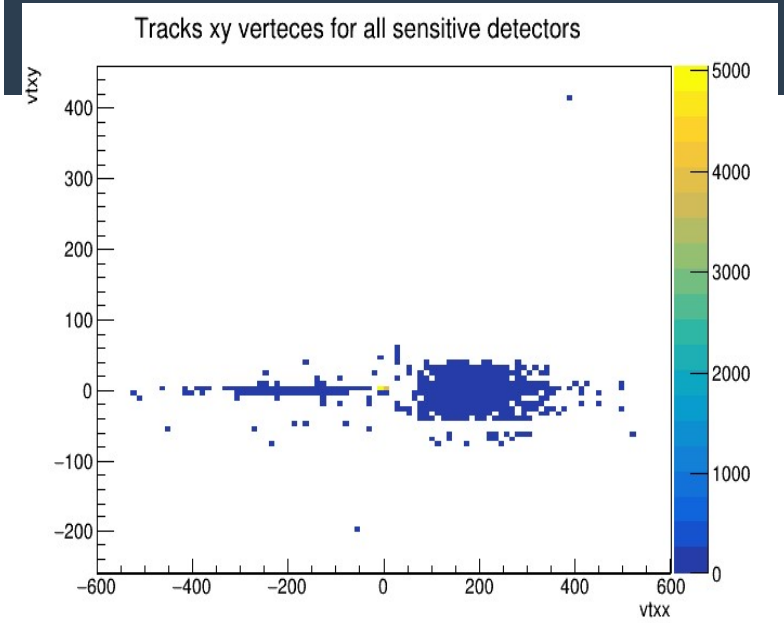
Plot first graphs to verify the compliance of simulated data

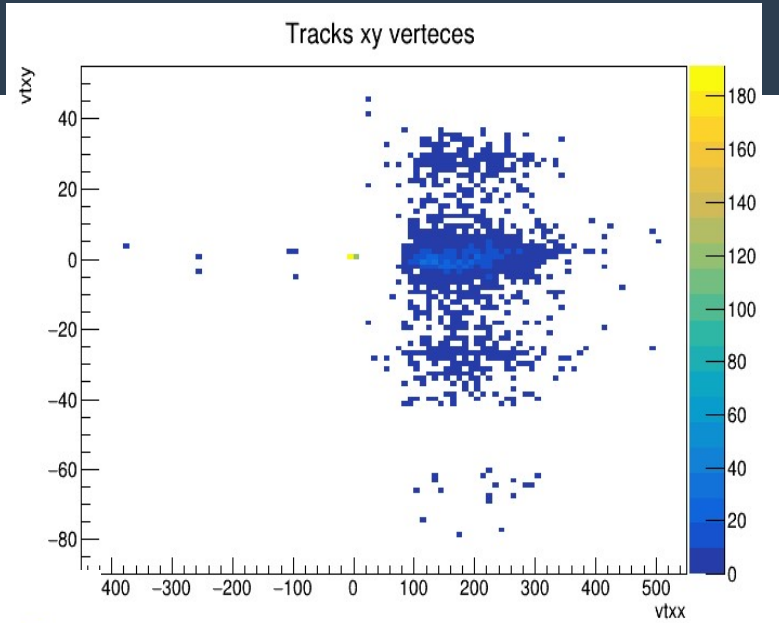


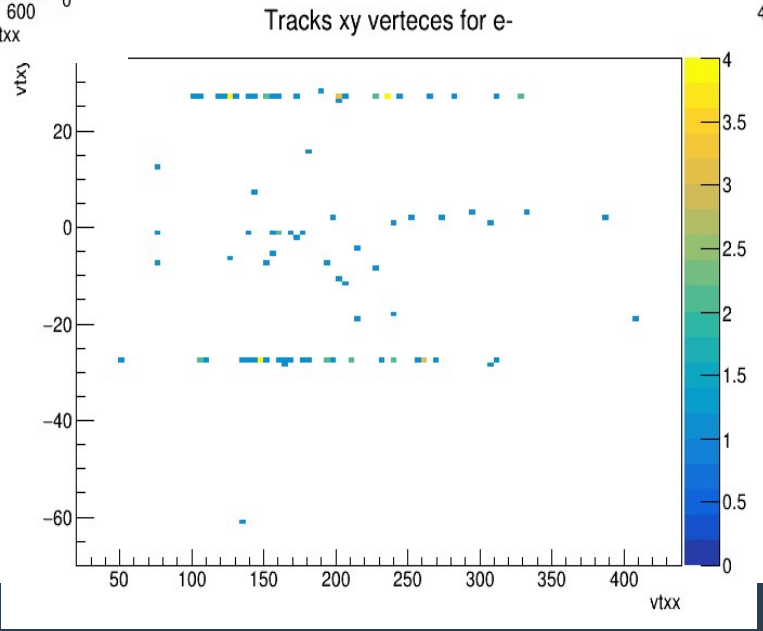




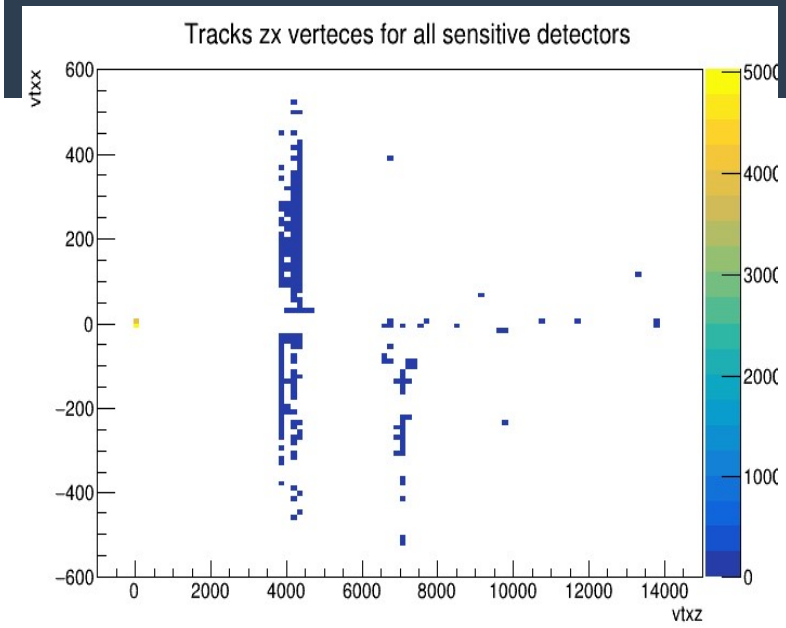
Plot first graphs to verify the compliance of simulated data

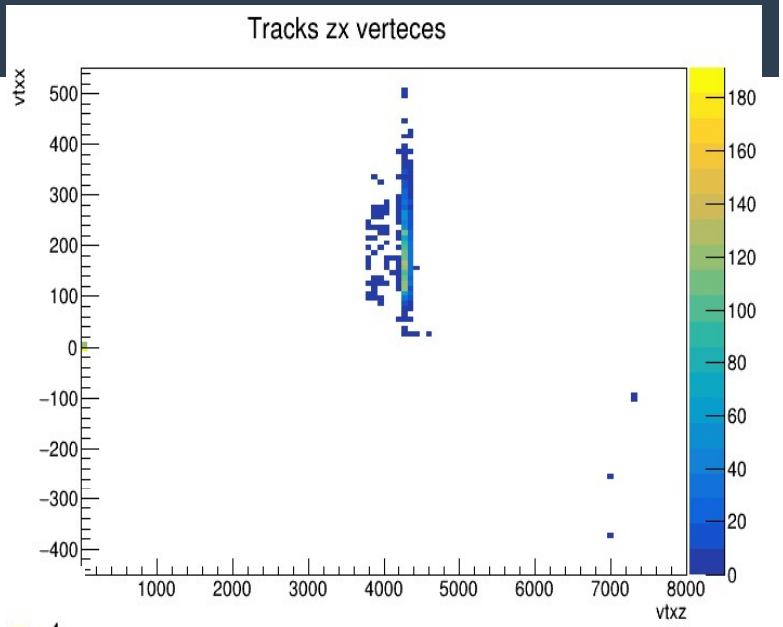


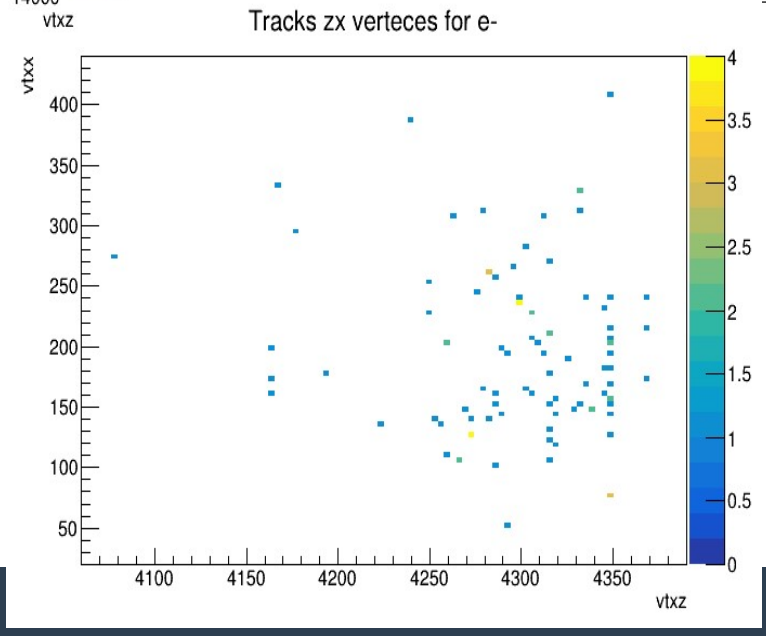




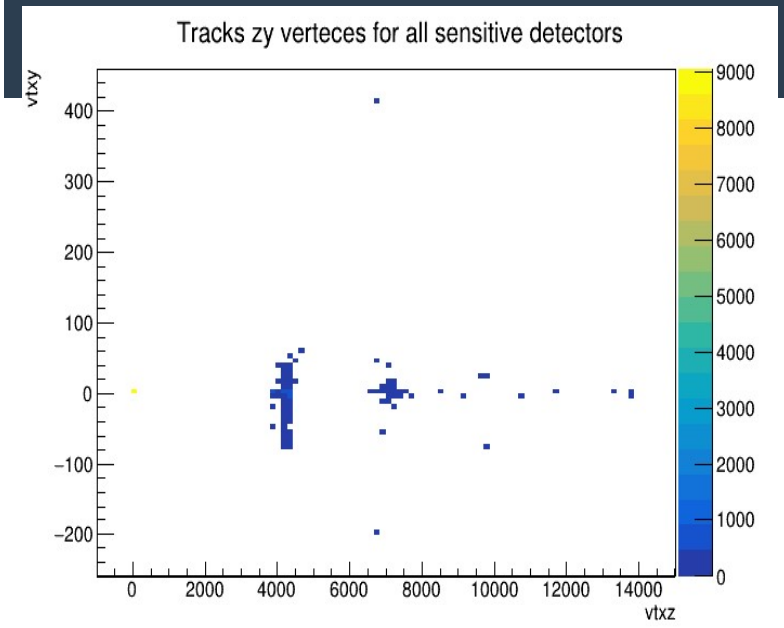
Plot first graphs to verify the compliance of simulated data

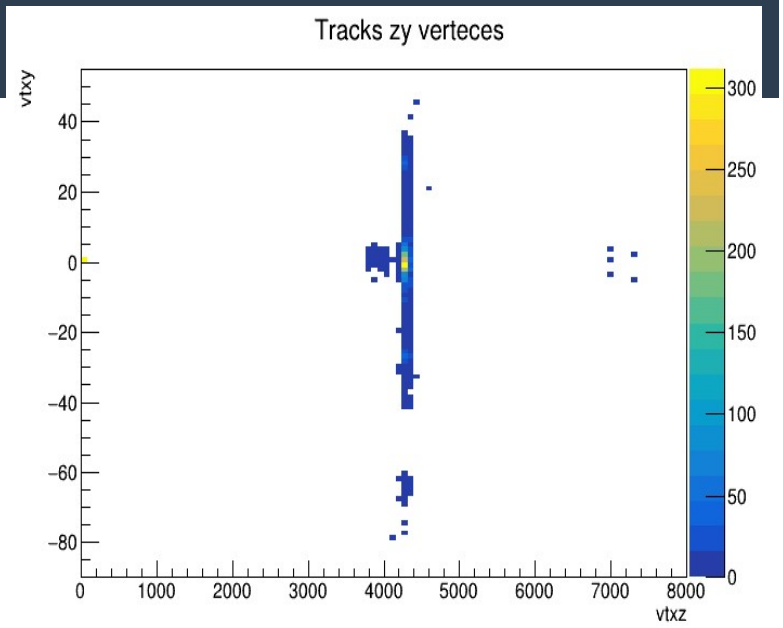


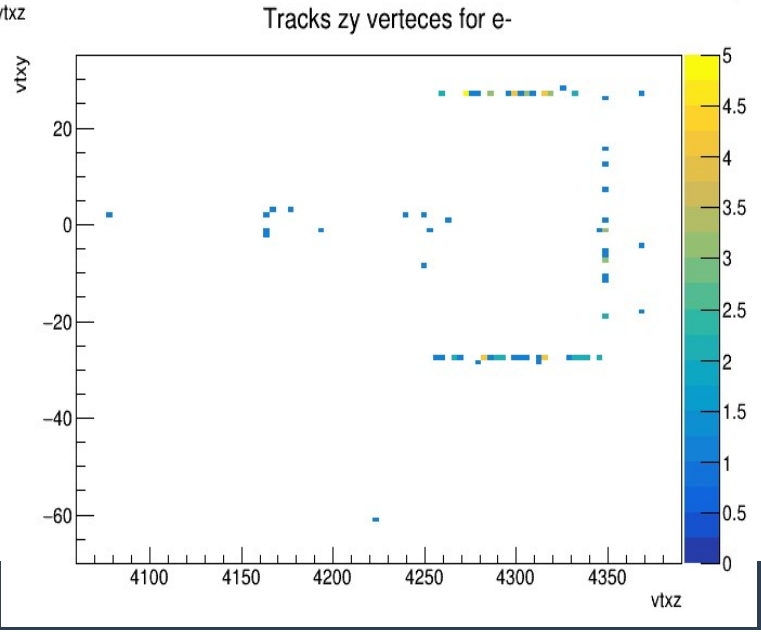




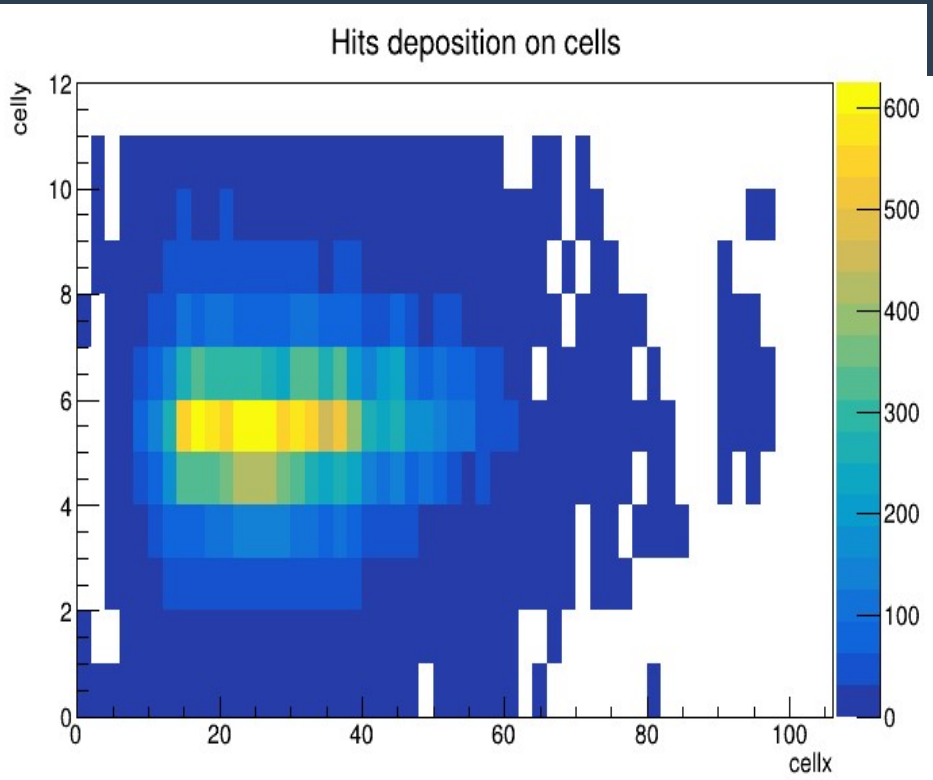
Plot first graphs to verify the compliance of simulated data

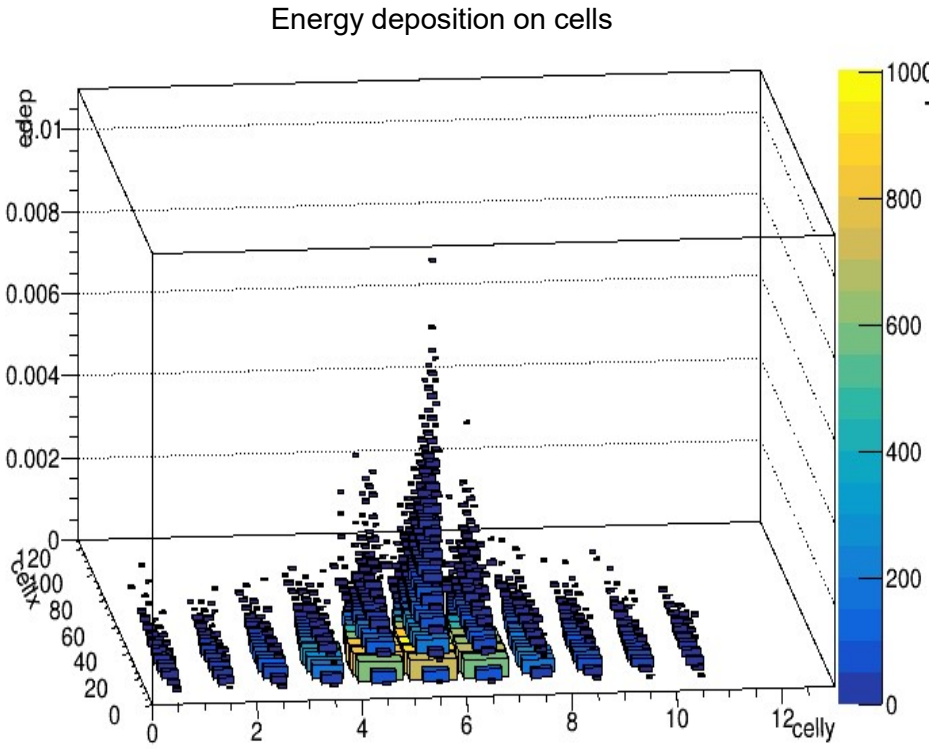






Plot first graphs to verify the compliance of simulated data





Thank you!