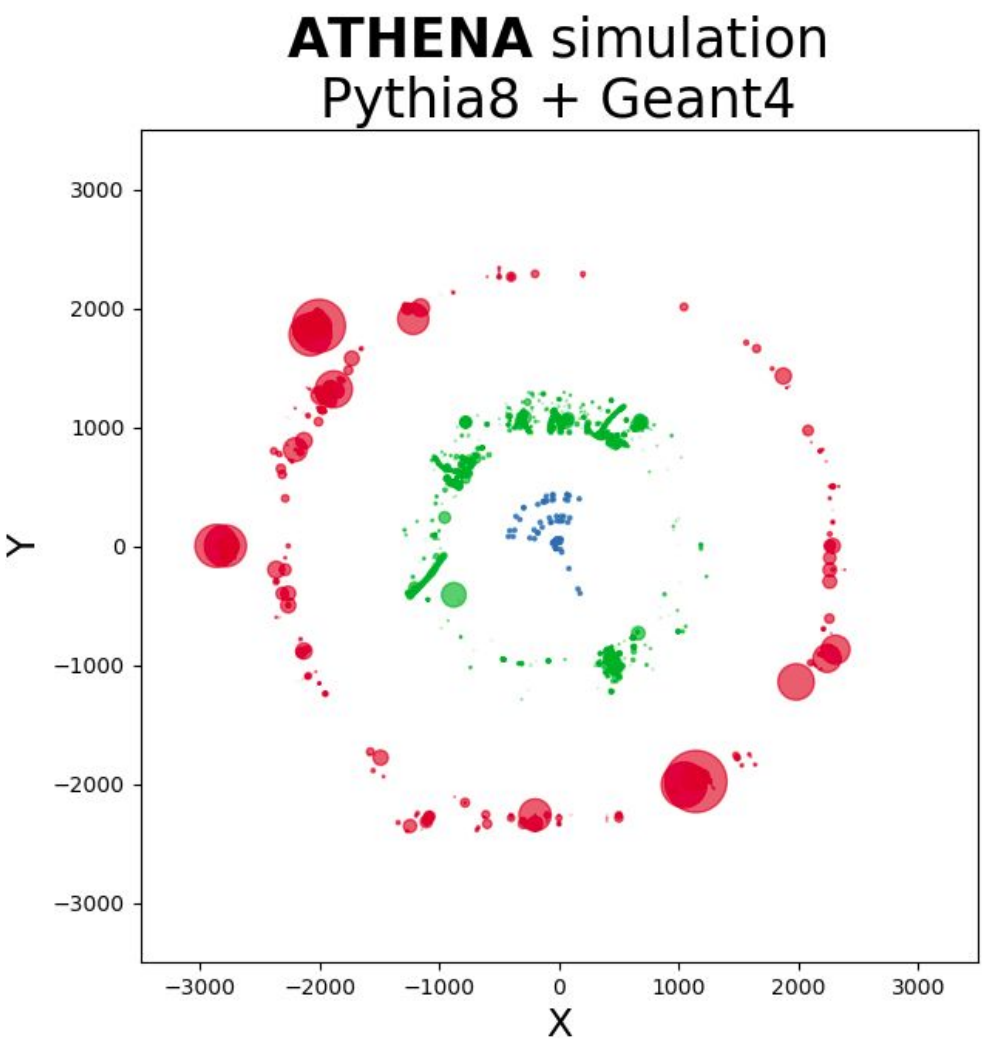
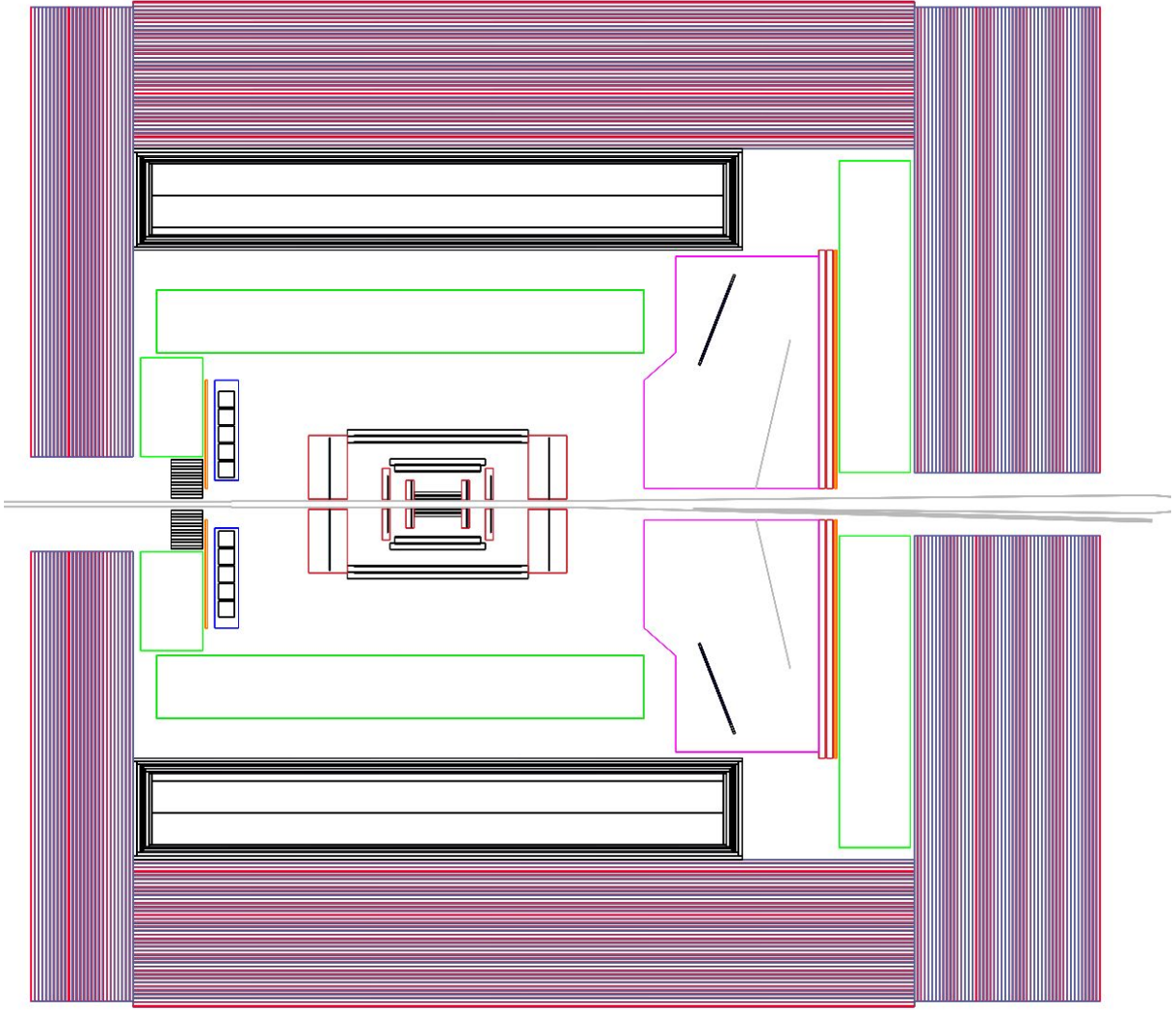
Update: towards full reconstruction and physics benchmarks

Miguel Arratia (UCR),
June 15th 2021
(T-169 days)



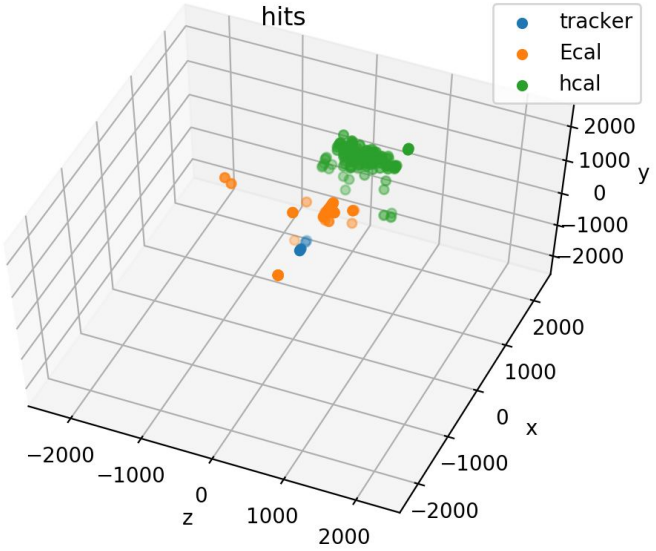


Reminder:

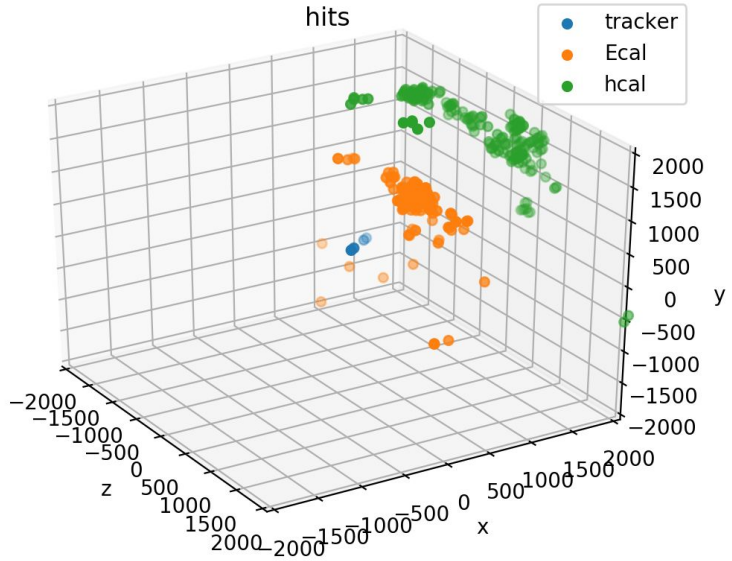
- We have “overspec” granularity in simulation.
- We will “regroup” layers/cells in the reconstruction step to study tradeoff granularity

Simulation status: single-particle G4 simulations

Pion 20 GeV

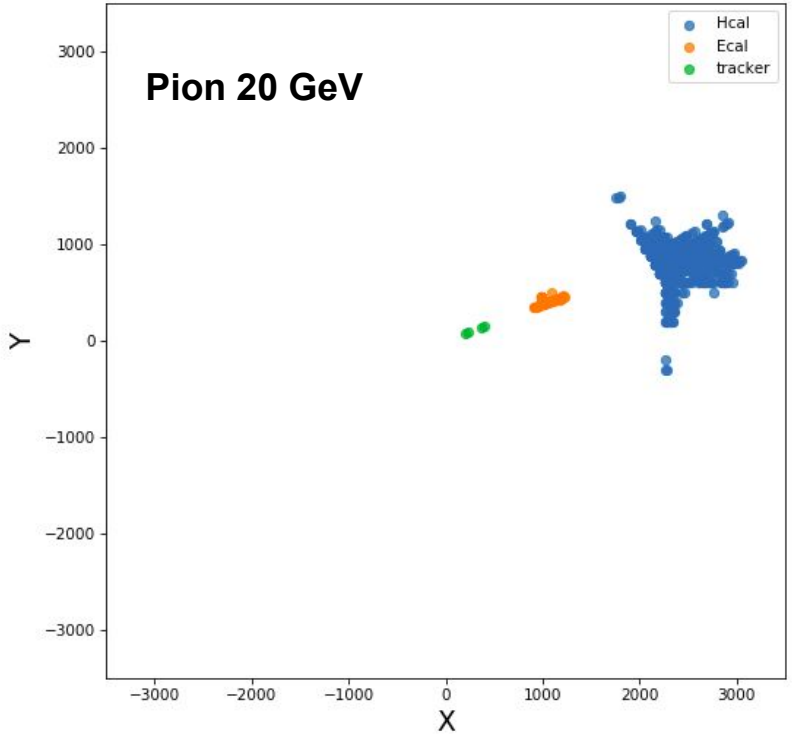


Pion 20 GeV, (showers in ECAL)

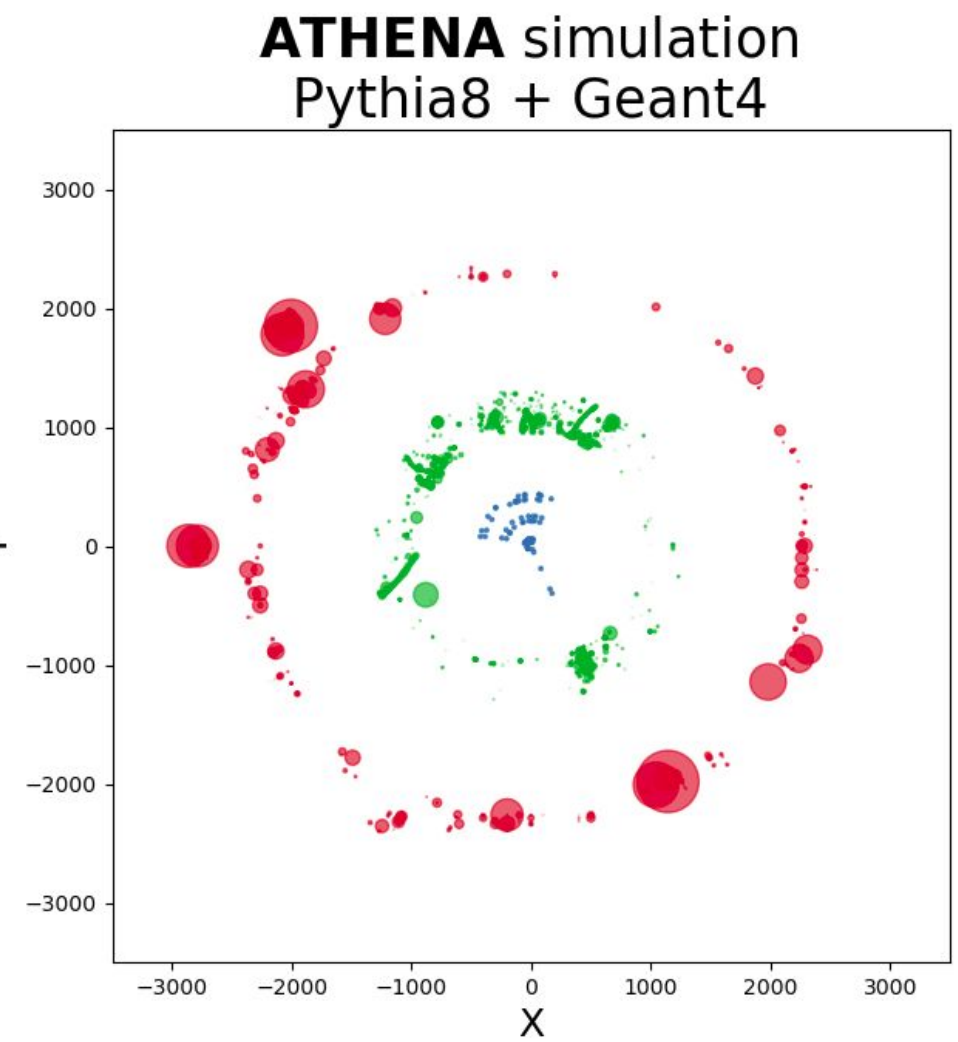


“Hits” from tracker, ECAL, and HCAL are shown here

Current status: debugging HCal reconstruction code



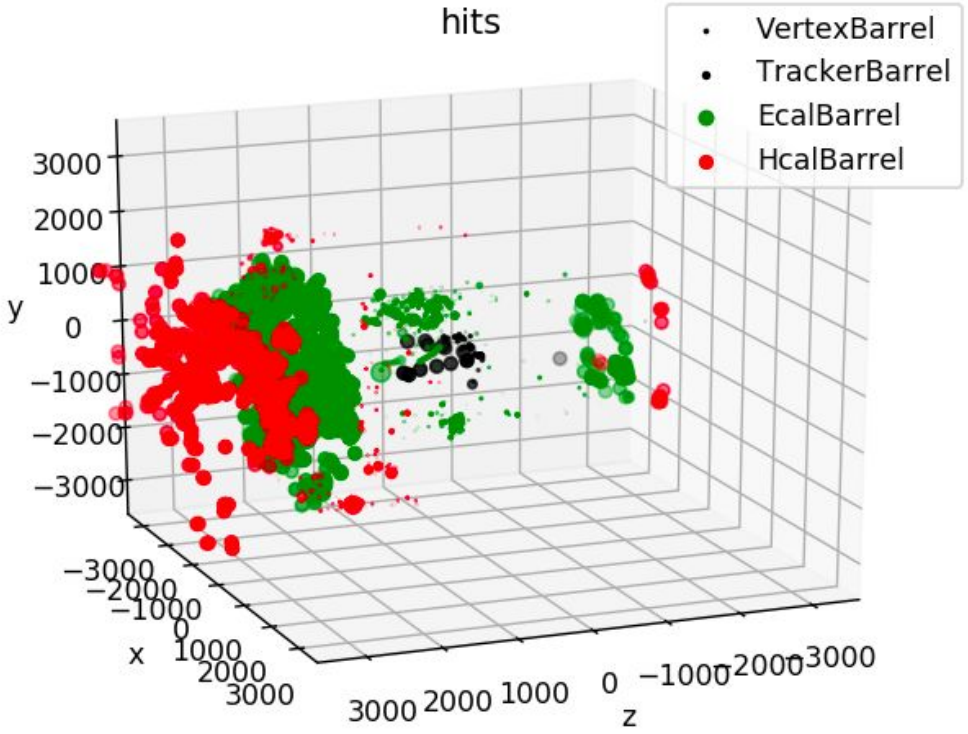
- We are are still testing output of HCal reconstruction code.
- Updates were required given that it is not uncommon to have HCAL clusters that encompass more than one sector. Chao made improvements, I am about to test this.
- Working in flexible codes for the “merging” of HCAL layers for granularity studies



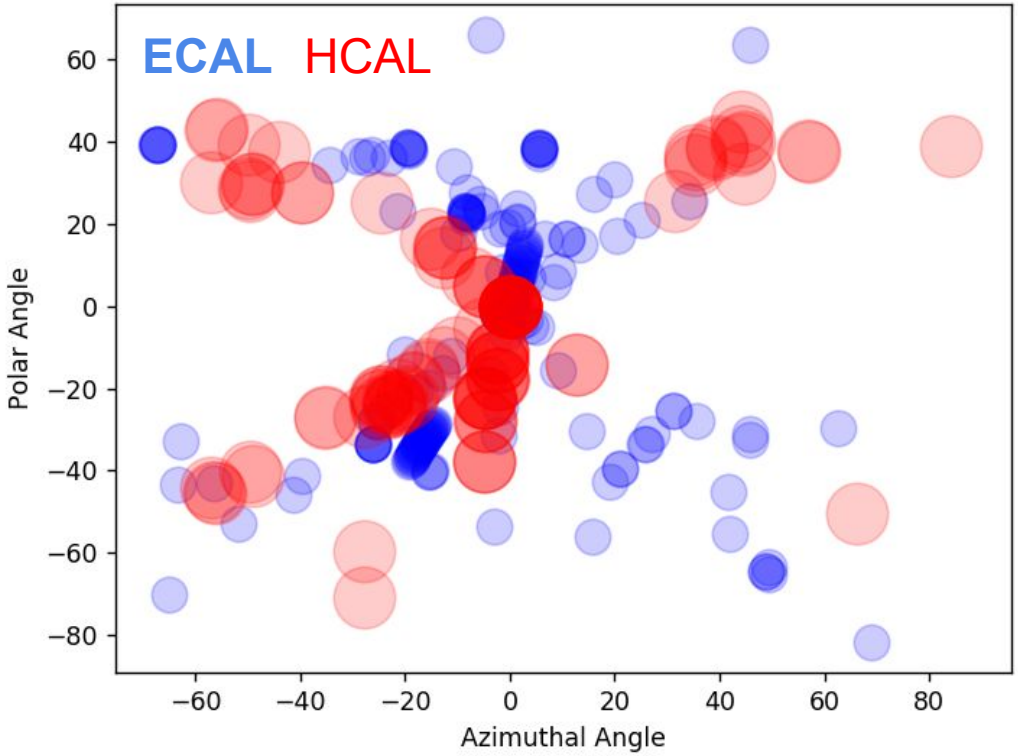
- First full-simulation campaign successfully ran last week (Wouter) using Pythia8 DIS events (Brian)
- Useful test to stress computer resources, estimate data sample sizes, and start physics analysis!

A milestone!

Full simulation files: G4 sim done, reconstruction ongoing

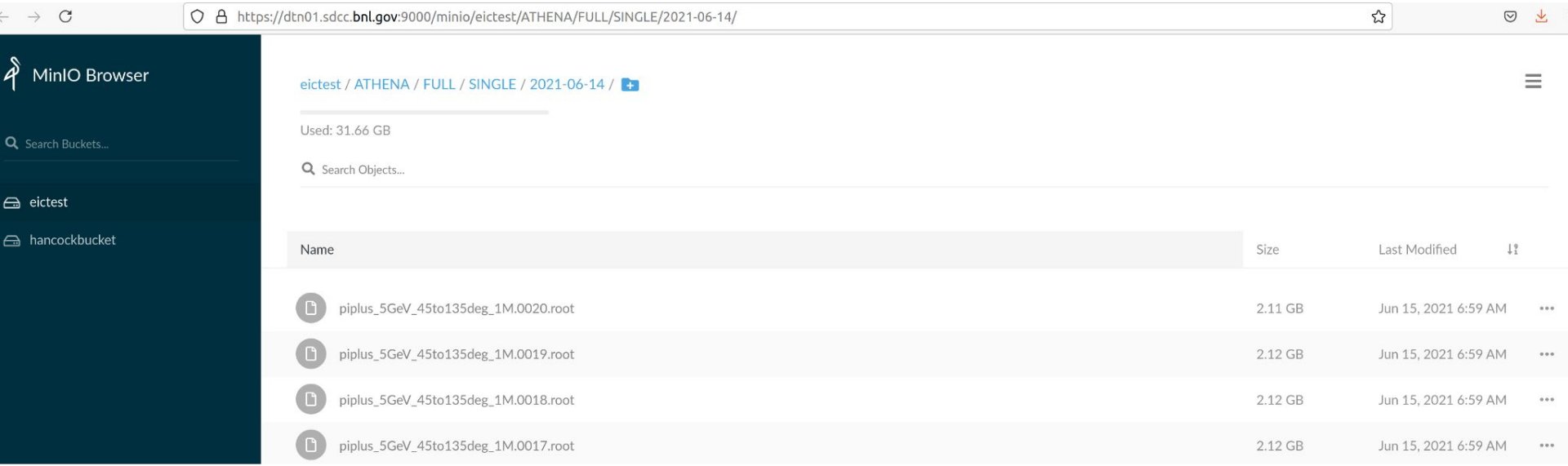


Pythia8 DIS event (Q2>100 GeV2), in barrel



Reminder:
We want to combine tracker+ECAL + HCAL energy (“energy-flow” algorithm)

Getting some simulation files to get started is easy:



How you could explore the simulation:
(for example: https://github.com/miguelignacio/calostudies)

In [1]:
```
%matplotlib notebook
import uproot as ur
import matplotlib.pyplot as plt
import k3d
import numpy as np
import awkward as ak
```

Get file and TTree, print branches, convert to array

In [2]:
```
#file = ur.open('sim_highq2.root')
file = ur.open('klong_10GeV_45to135deg_1M.0001.root')
tree = file['events']
print(tree.keys())
ak_arrays = tree.arrays()
```

```
['mcparticles', 'mcparticles/mcparticles.ID', 'mcparticles/mcparticles.g4Parent', 'mcparticles/mcparticles.reason
', 'mcparticles/mcparticles.mask', 'mcparticles/mcparticles.steps', 'mcparticles/mcparticles.secondaries', 'mcpart
icles/mcparticles.pdgID', 'mcparticles/mcparticles.status', 'mcparticles/mcparticles.colorFlow[2]', 'mcparticles/m
cparticles.genStatus', 'mcparticles/mcparticles.charge', 'mcparticles/mcparticles.spare[1]', 'mcparticles/mcpartic
les.spin[3]', 'mcparticles/mcparticles.vsx', 'mcparticles/mcparticles.vsy', 'mcparticles/mcparticles.vsz', 'mcpart
icles/mcparticles.vex', 'mcparticles/mcparticles.vey', 'mcparticles/mcparticles.vez', 'mcparticles/mcparticles.psx
', 'mcparticles/mcparticles.psy', 'mcparticles/mcparticles.psz', 'mcparticles/mcparticles.pex', 'mcparticles/mcpar
ticles.pey', 'mcparticles/mcparticles.pez', 'mcparticles/mcparticles.mass', 'mcparticles/mcparticles.time', 'mcpar
ticles/mcparticles.properTime', 'mcparticles/mcparticles.parents_begin', 'mcparticles/mcparticles.parents_end', 'm
cparticles/mcparticles.daughters_begin', 'mcparticles/mcparticles.daughters_end', 'mcparticles_0', 'mcparticles_1
', 'DIRCHits', 'DIRCHits/DIRCHits.cellID', 'DIRCHits/DIRCHits.flag', 'DIRCHits/DIRCHits.g4ID', 'DIRCHits/DIRCHits.
position.x', 'DIRCHits/DIRCHits.position.y', 'DIRCHits/DIRCHits.position.z', 'DIRCHits/DIRCHits.position.t', 'DIRC
Hits/DIRCHits.momentum.x', 'DIRCHits/DIRCHits.momentum.y', 'DIRCHits/DIRCHits.momentum.z', 'DIRCHits/DIRCHits.mome
ntum.t', 'DIRCHits/DIRCHits.length', 'DIRCHits/DIRCHits.truth.trackID', 'DIRCHits/DIRCHits.truth.pdgID', 'DIRCHits
/DIRCHits.truth.deposit', 'DIRCHits/DIRCHits.truth.time', 'DIRCHits/DIRCHits.truth.length', 'DIRCHits/DIRCHits.tru
```

```
rgyDeposit']
```

In [3]:
```
def get_vector(varname='HcalBarrelHits',energy='energyDeposit'):
    E = np.array(ak.to_list(ak_arrays["%s.%s"%(varname,energy)]), dtype="O")
    x = np.array(ak.to_list(ak_arrays["%s.position.x"%varname]), dtype="O")
    y = np.array(ak.to_list(ak_arrays["%s.position.y"%varname]), dtype="O")
    z = np.array(ak.to_list(ak_arrays["%s.position.z"%varname]), dtype="O")


    return E,x, y, z
```

In [4]:
```
E = {}
x = {}
y = {}
z  = {}
r={}
```

Get data

In [5]:
```
for i in ['HcalBarrel','EcalBarrel','TrackerBarrel','VertexBarrel']:
    E[i], x[i], y[i],z[i] = get_vector("%sHits"%i)
```

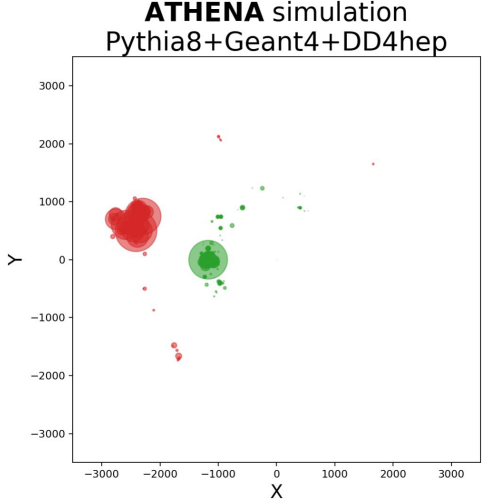
In [6]:
```
for i in ['HcalEndcap','EcalEndcap','TrackerEndcap','VertexEndcap']:
    E[i], x[i], y[i],z[i] = get_vector("%sHits"%i)
```

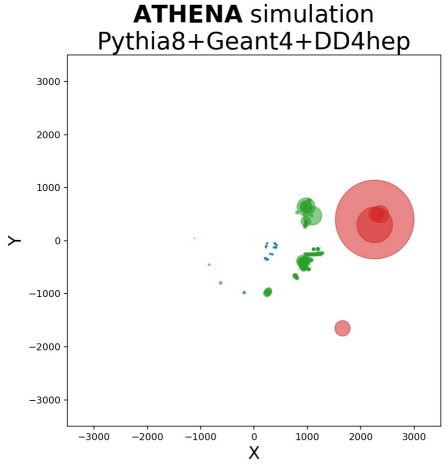
In [7]:
```
for i in ['DIRC']:
    E[i], x[i], y[i],z[i] = get_vector("%sHits"%i,'energy')
```

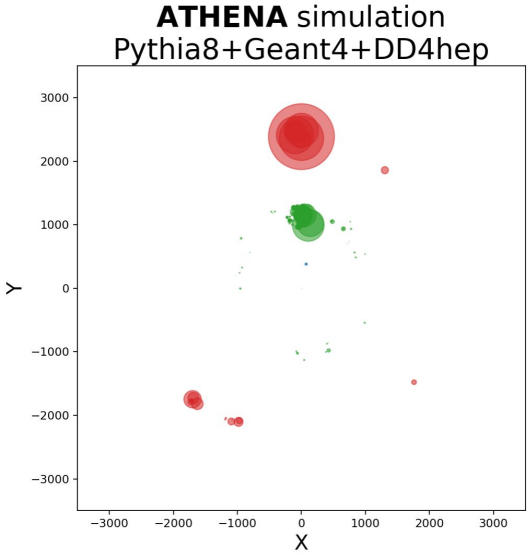
Plot 2D transverse view for hits in barrel

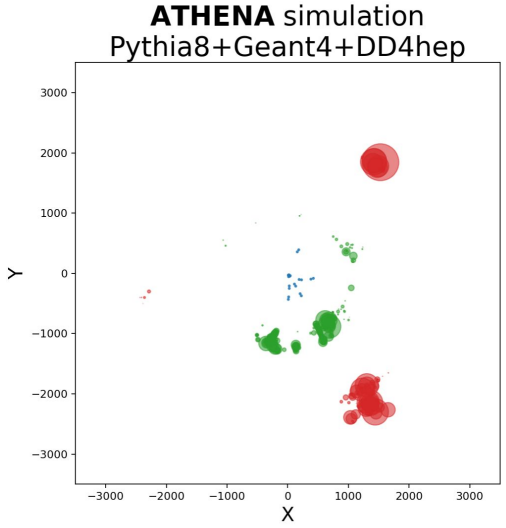
In [14]:
```
#loop over events
for ievt in range(0,30):
    #HERE FILTER YOUR EVENTS IF YOU WANT
    #if(len(x['VertexBarrel'][ievt])<3): continue
    #if(len(x['TrackerBarrel'][ievt])<20): continue
    #if(len(x['HcalBarrel'][ievt])<50): continue

    x_tracker = np.concatenate((x['VertexBarrel'][ievt], x['TrackerBarrel'][ievt]), axis=0)
    y_tracker = np.concatenate((y['VertexBarrel'][ievt], y['TrackerBarrel'][ievt]), axis=0)
    E_tracker = np.concatenate((E['VertexBarrel'][ievt], E['TrackerBarrel'][ievt]), axis=0)
```

Klong events

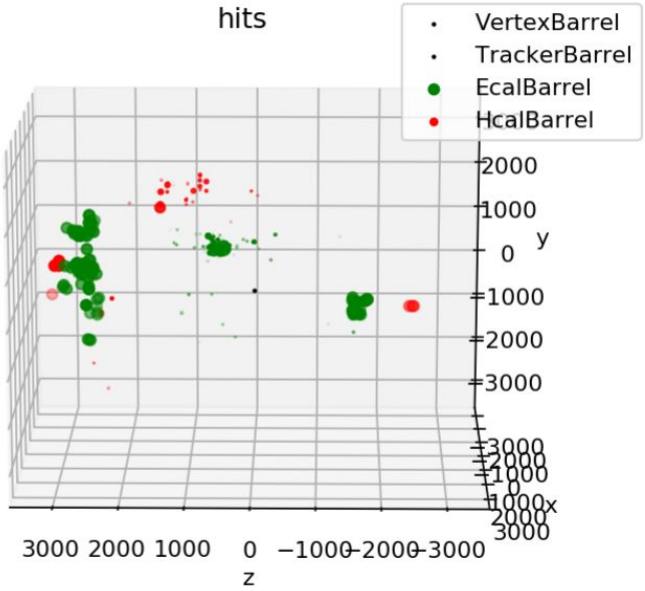


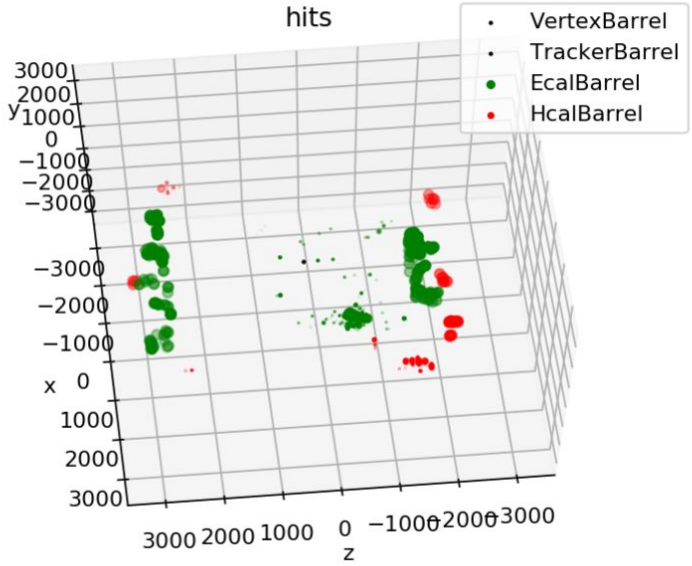






“Particle-gun” KL. 2 events shown below





- Beware, “single particle” events contain electron and proton as well.
- Should we change this behaviour for simplicity?

Summary

- HCAL fully integrated in DD4HEP simulations.
- Both single-particle and Pythia8 DIS events are already available. G4 simulation step done, reconstruction ongoing.
- Once we fix clustering code, we will proceed to perform physics benchmark analyzes.