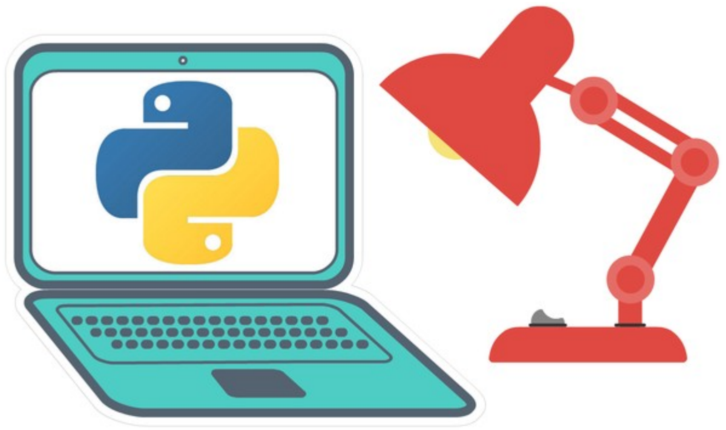
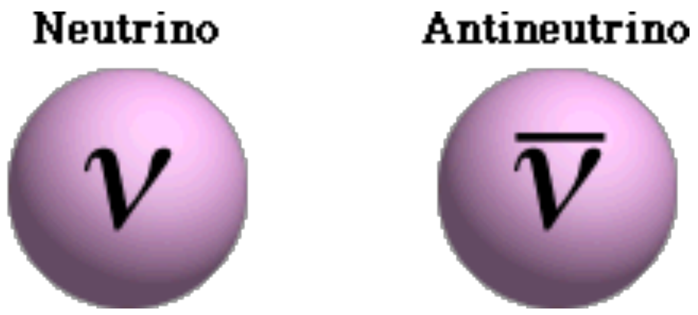


Python in NEXT Experiment

MAYA CHATTORAJ

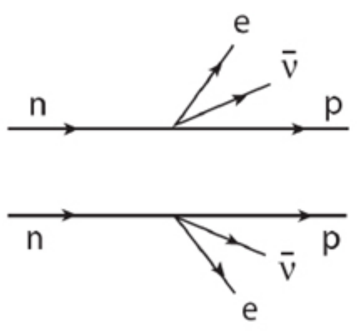
NEXT Experiment

- Are neutrinos their own anti-matter?
- If yes- two neutrinos cancel each other out
- How to see: double beta decay

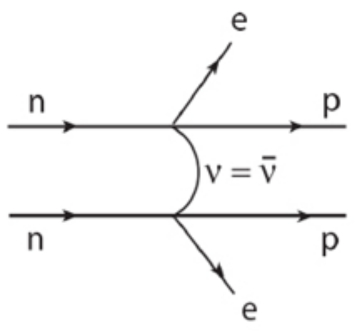


Double Beta Decay

- Two neutrons $\Longrightarrow$ two protons
- Ordinary: two electrons, two electron antineutrinos emitted
- Neutrinoless: only two electrons emitted
- Neutrinos annihilate each other



ordinary



neutrinoless

NEXT Experiment

- Detector filled with pressurized Xenon gas
- Located underground in Spain: Laboratorio Subterráneo de Canfranc (LSC)
- Tracks path & energy of electron(s)

Detector Performance Simulation

- Experiment has not run yet
- Simulations of performance are created
- Used to study predicted results

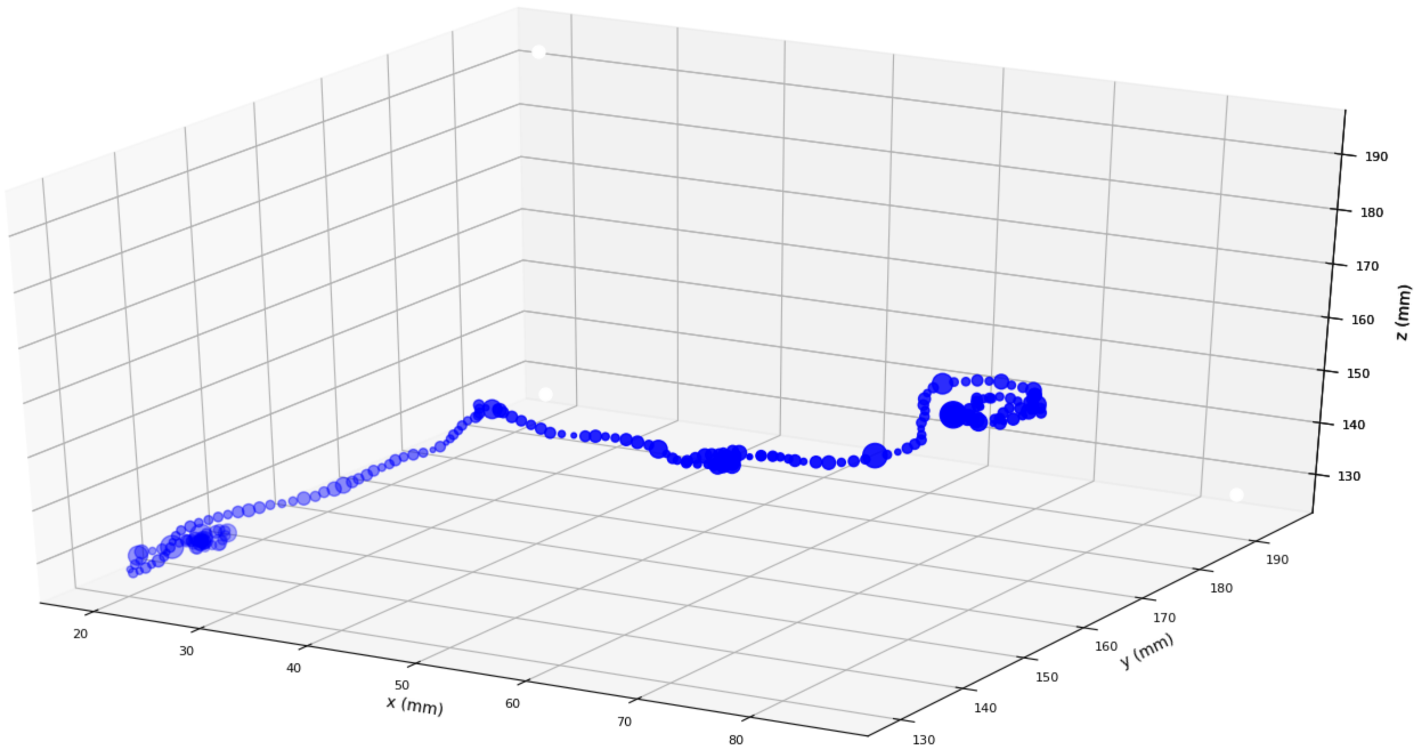
Signal & Background Events

- Two types of events detected
- Signal: double beta decay
- Background: some other reaction
- Sort between events to find signals

Using Python to Analyze Data

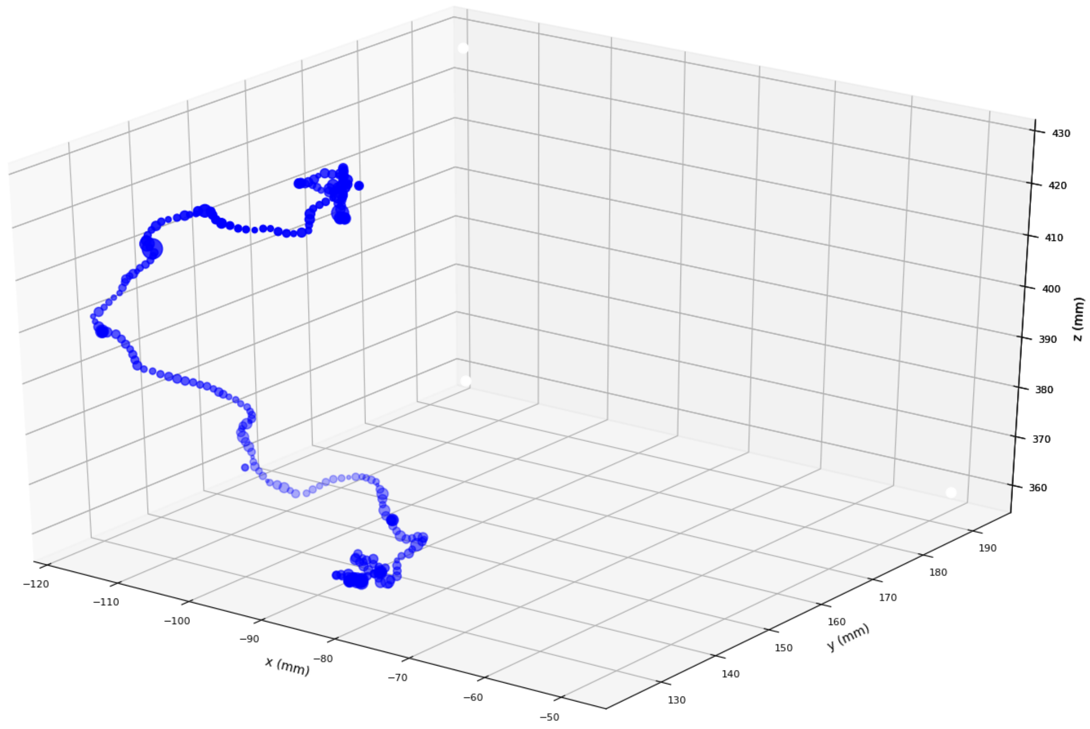
- Easy to understand
- Powerful tools
- Many libraries available
- Online help

3D Plot of Events



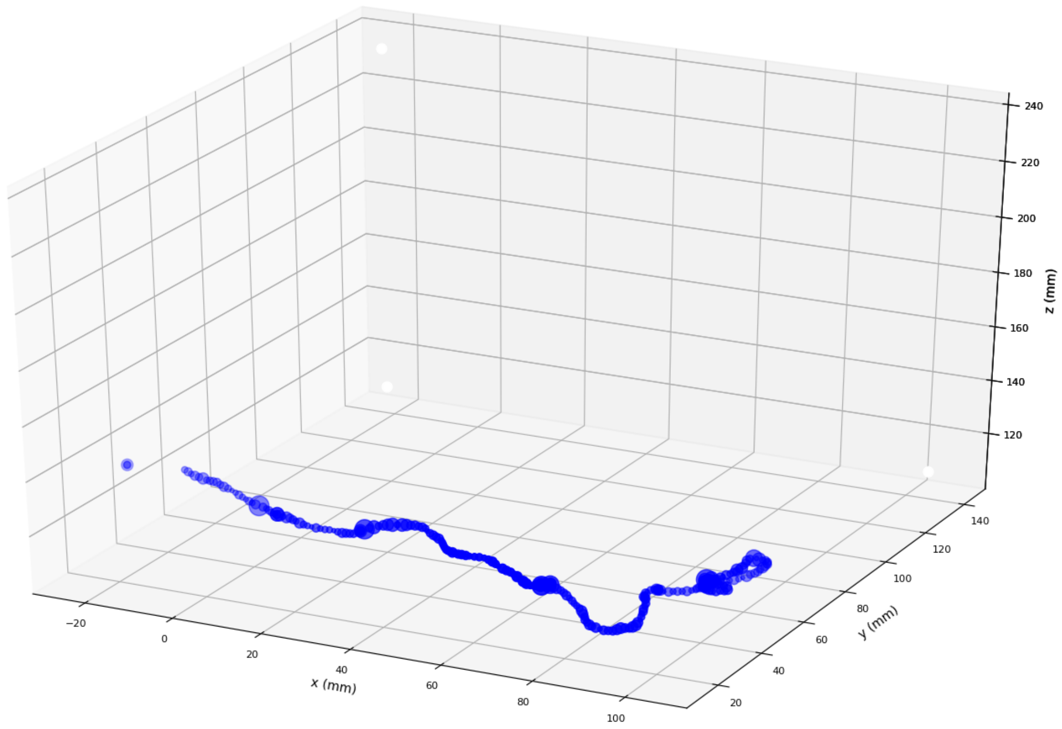
Signal Event

- Lose energy $\longrightarrow$ path curls
- Lose more energy at ends
- End of tracks are blob-like
- Two electrons, two blobs



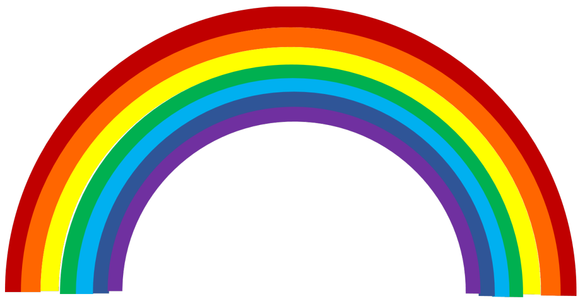
Background Event

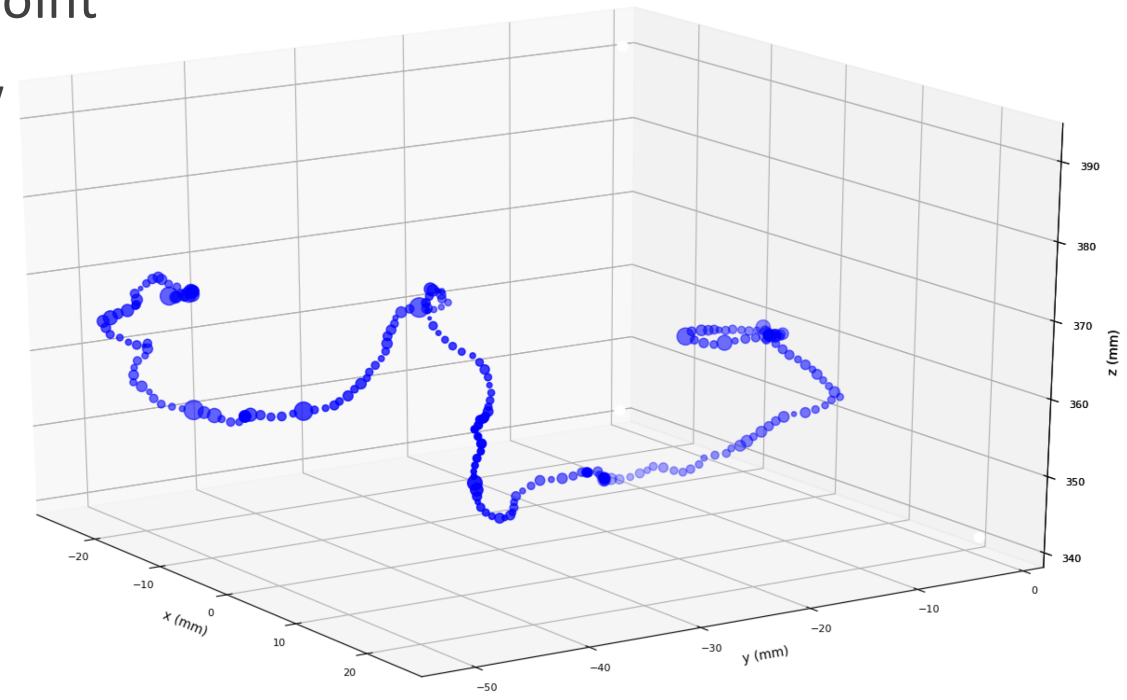
- Only one electron (not what we're looking for)
- One 'blob' at end
- Sort out from signal events



Display of Electrons' Energy

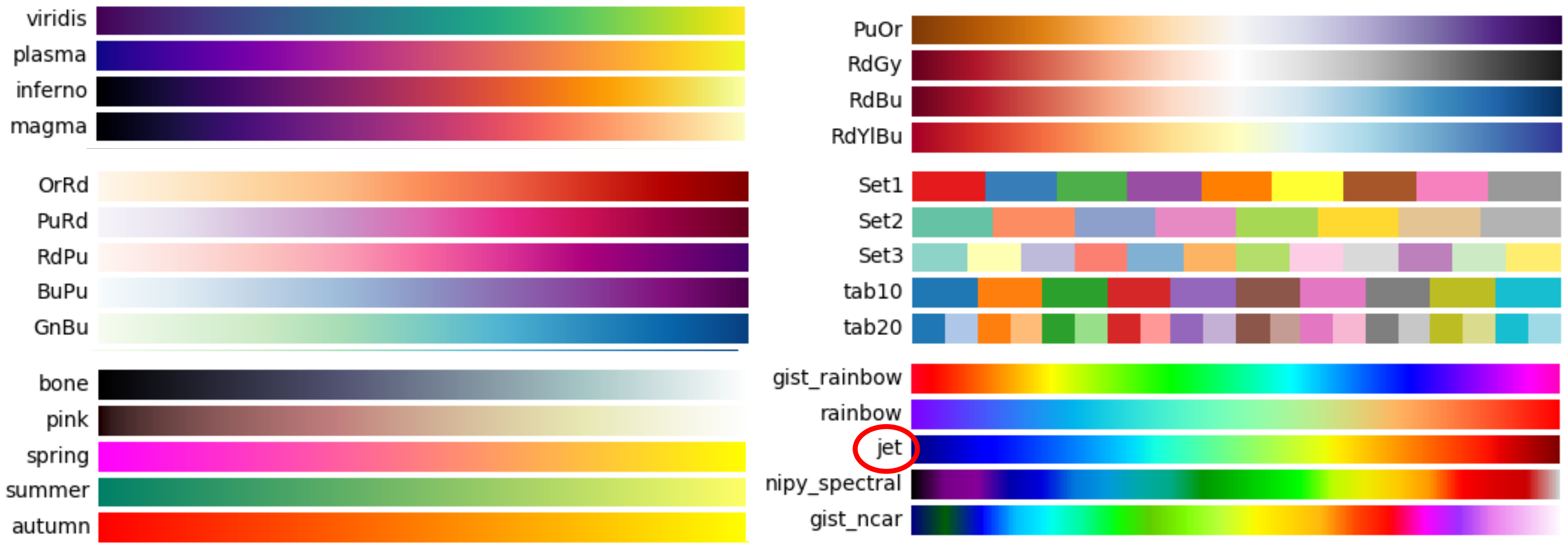
- Originally shown by size of point
- Distorts track, hard to follow
- Use color instead



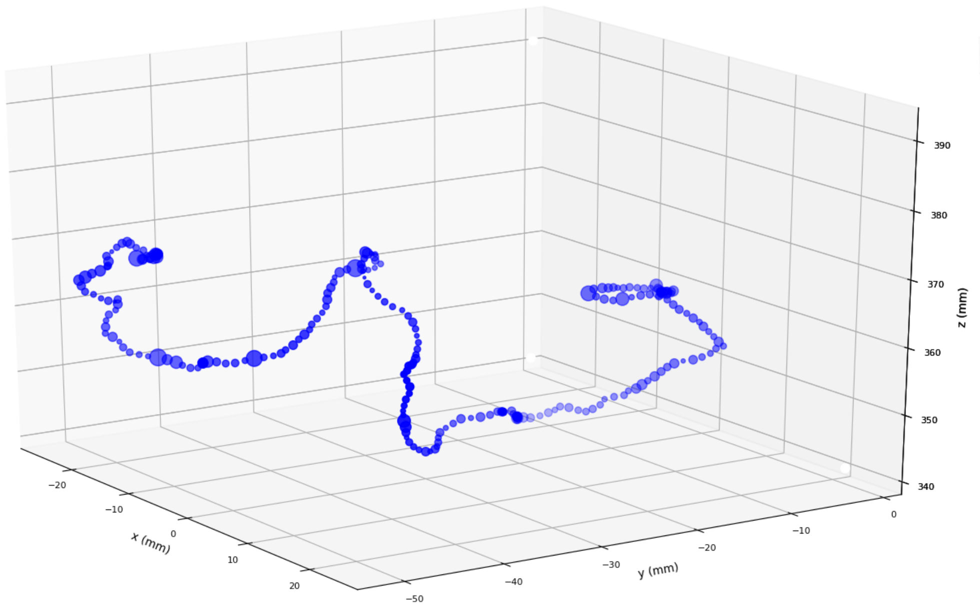


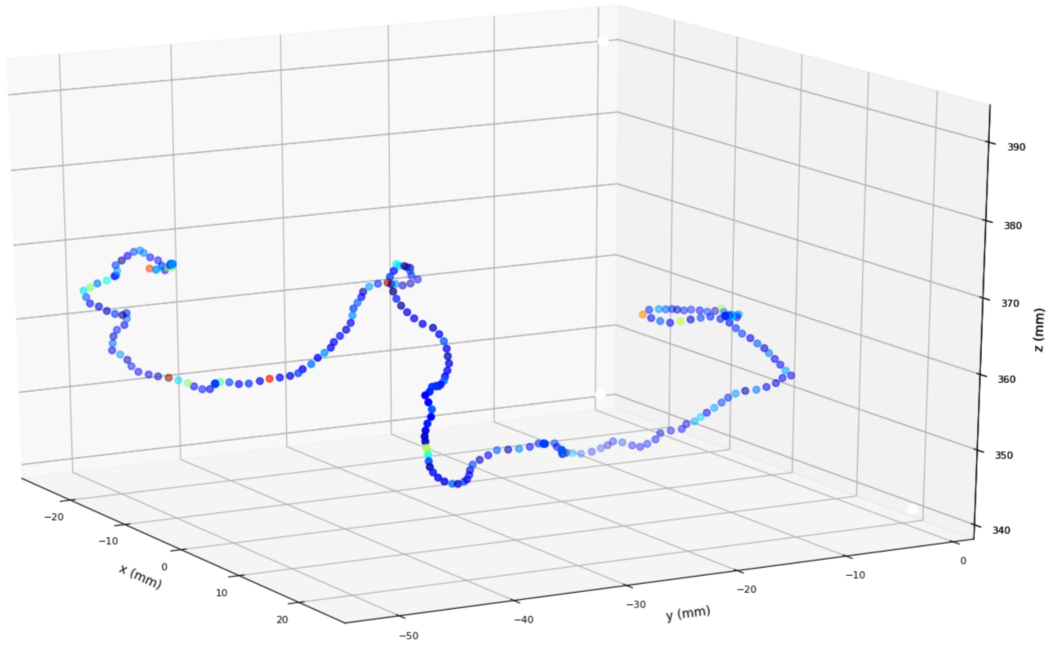
Colormapping

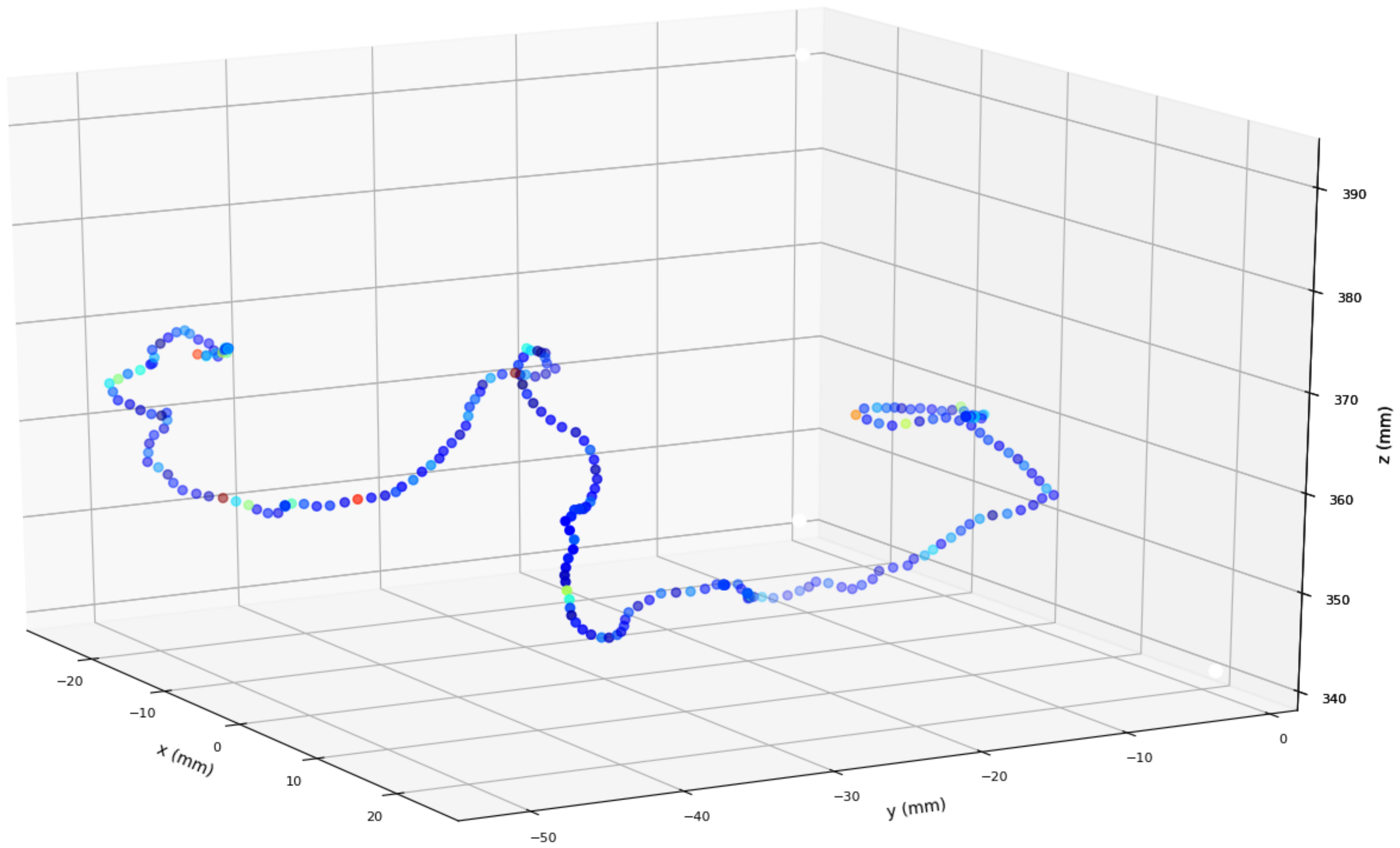
- Scale energy values with color



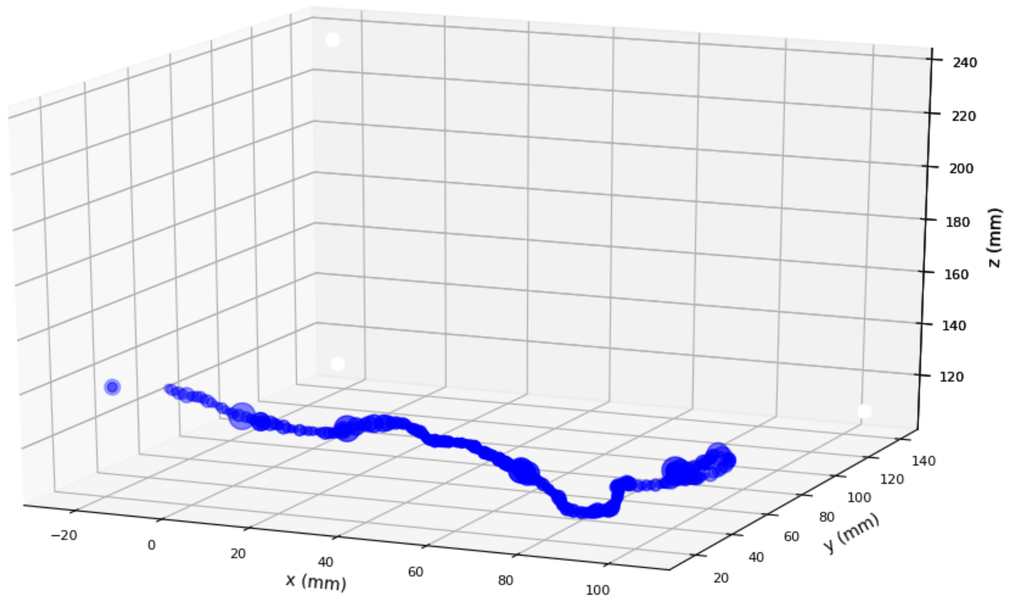
Signal Event with Colorscale

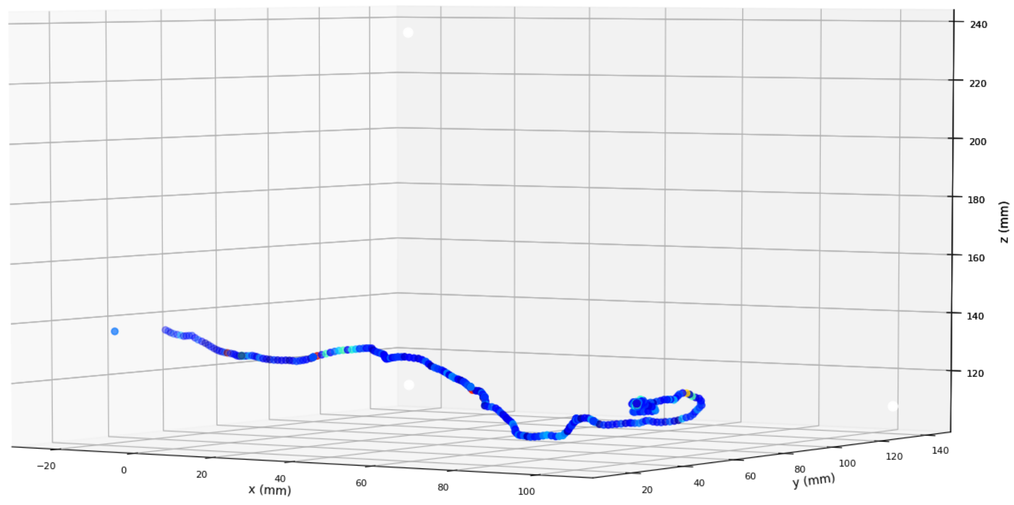


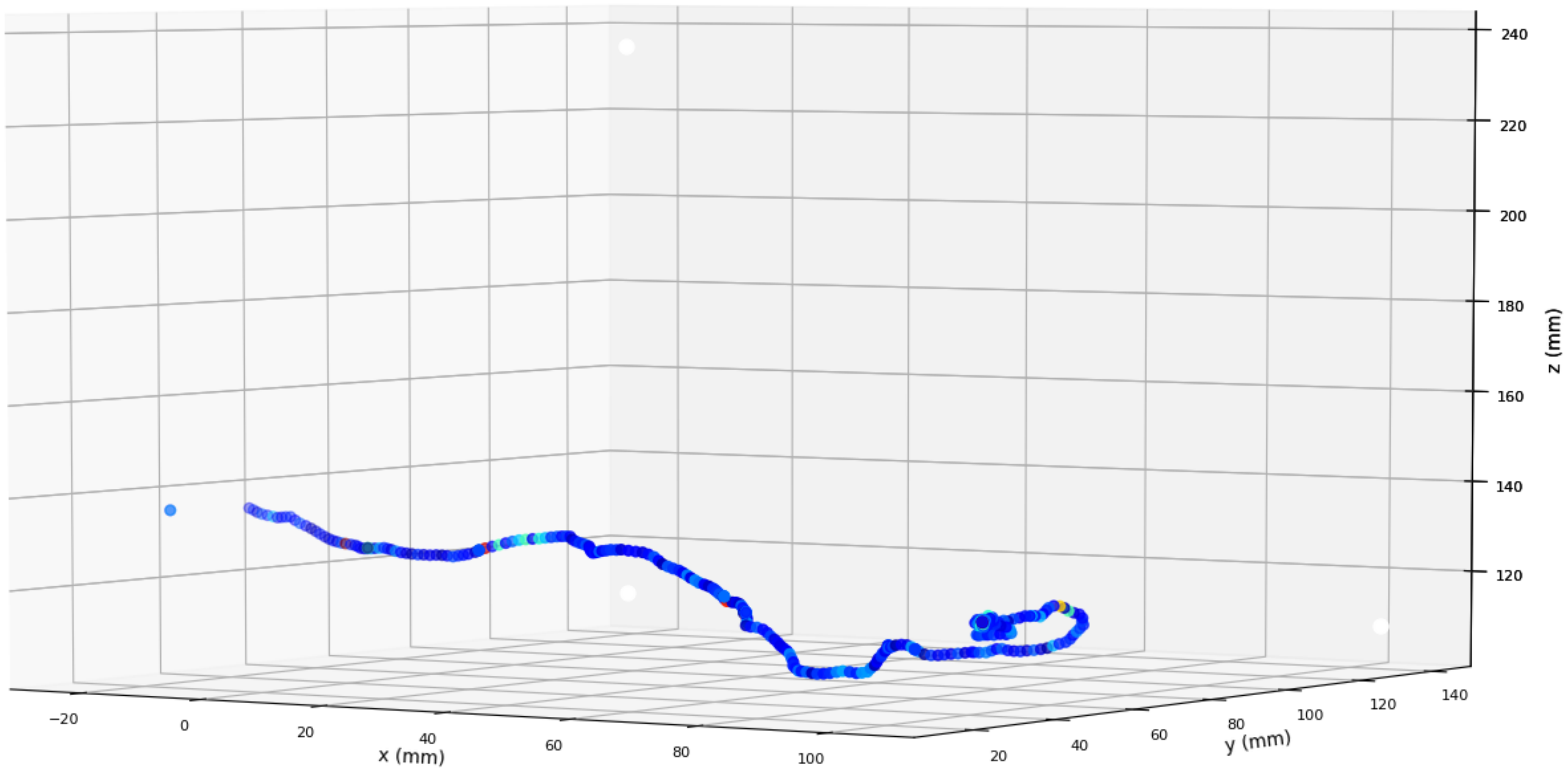



x (mm)
−20
−10
0
10
20
y (mm)
−50
−40
−30
−20
−10
0
z (mm)
340
350
360
370
380
390

Background Event with Colorscale

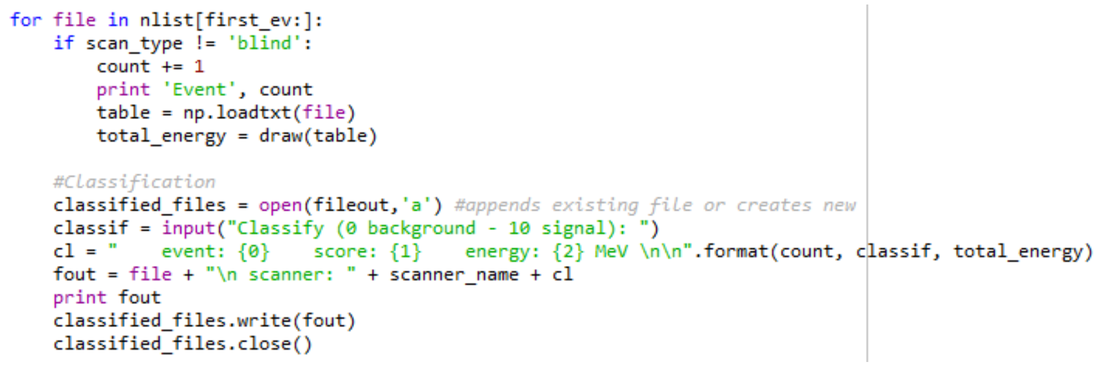




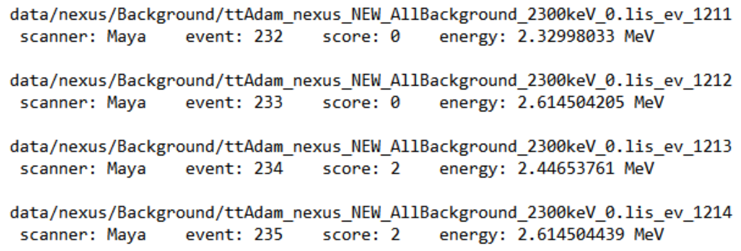

x (mm)
y (mm)
z (mm)
−20
0
20
40
60
80
100
20
40
60
80
100
120
140
120
140
160
180
200
220
240

Initial Scan & Classification

- Go through signal and background events
- Give a score 0-10
- 0: background
- 10: signal
- Write into text file



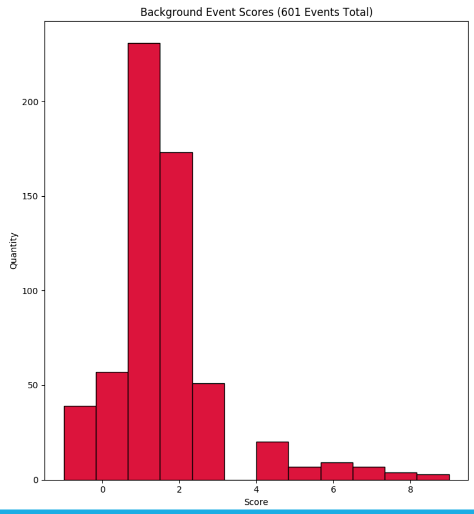
```
for file in nlist[first_ev:]:
    if scan_type != 'blind':
        count += 1
        print 'Event', count
        table = np.loadtxt(file)
        total_energy = draw(table)

    #Classification
    classified_files = open(fileout,'a') #appends existing file or creates new
    classif = input("Classify (0 background - 10 signal): ")
    cl = "    event: {0}     score: {1}     energy: {2} MeV \n\n".format(count, classif, total_energy)
    fout = file + "\n scanner: " + scanner_name + cl
    print fout
    classified_files.write(fout)
    classified_files.close()
```

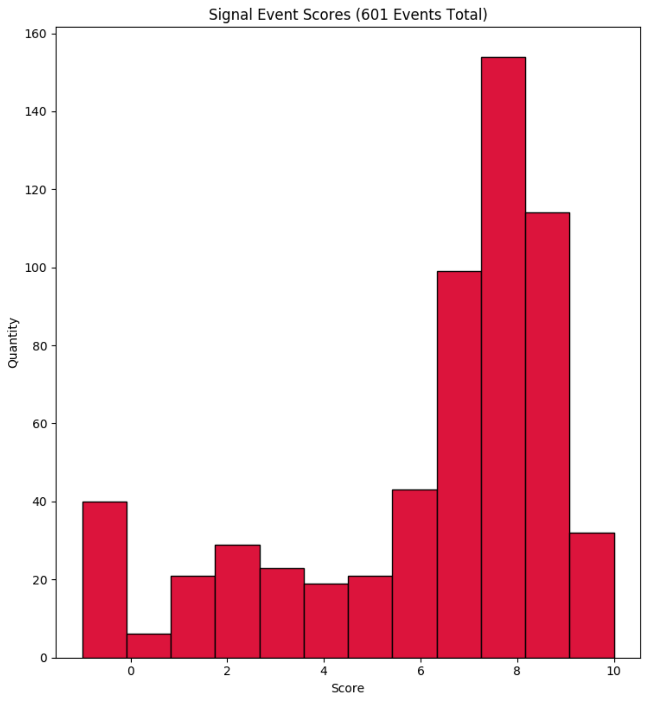


```
data/nexus/Background/ttAdam_nexus_NEW_AllBackground_2300keV_0.lis_ev_1211
 scanner: Maya     event: 232     score: 0     energy: 2.32998033 MeV

data/nexus/Background/ttAdam_nexus_NEW_AllBackground_2300keV_0.lis_ev_1212
 scanner: Maya     event: 233     score: 0     energy: 2.614504205 MeV

data/nexus/Background/ttAdam_nexus_NEW_AllBackground_2300keV_0.lis_ev_1213
 scanner: Maya     event: 234     score: 2     energy: 2.44653761 MeV

data/nexus/Background/ttAdam_nexus_NEW_AllBackground_2300keV_0.lis_ev_1214
 scanner: Maya     event: 235     score: 2     energy: 2.614504439 MeV
```

Plot Results of Scan

- Read text file with scores
- Parse scores and event types
- Plot results of score accuracies



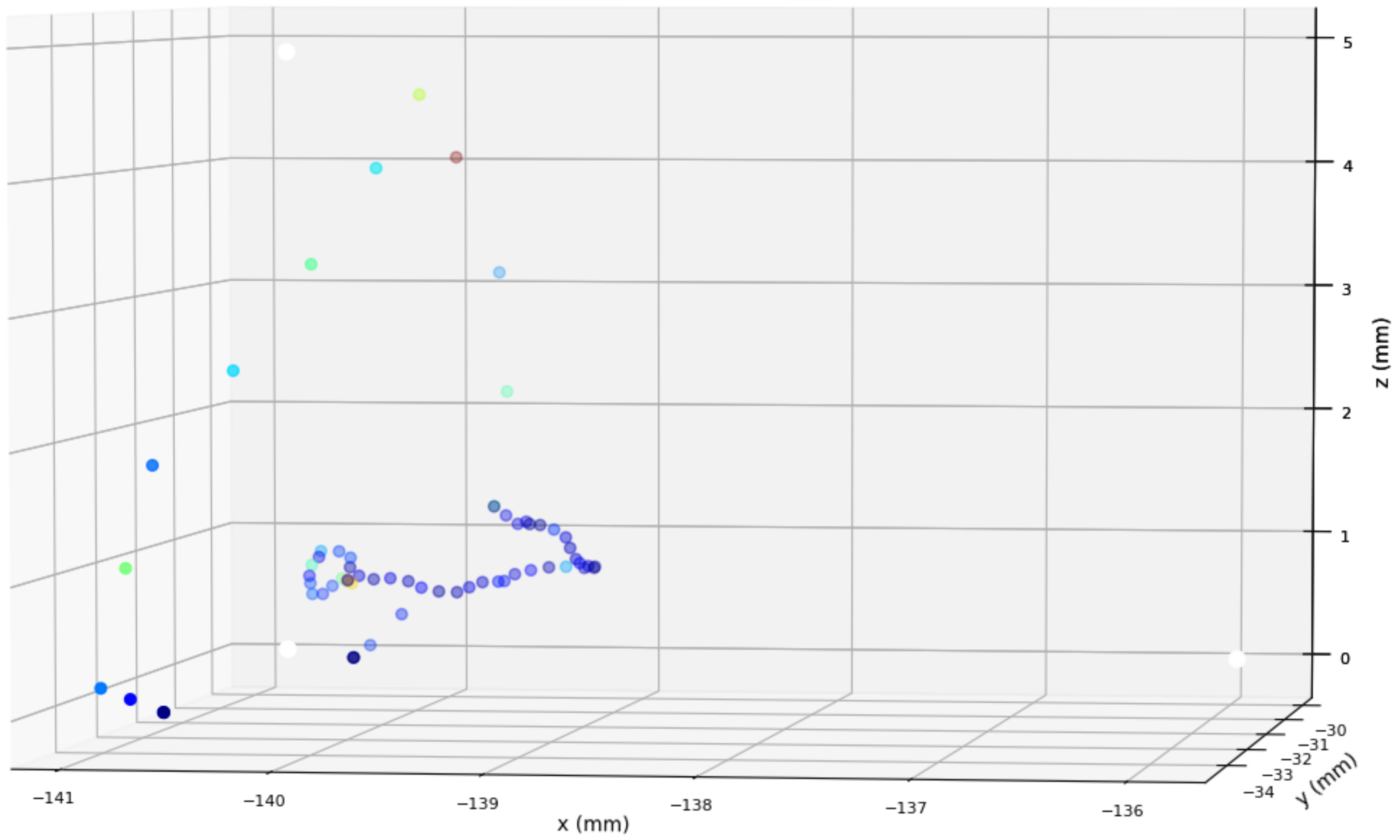


View Misclassified Events

- Go through score text files
- If score doesn't match event type, view event
- Identify misleading characteristics of events

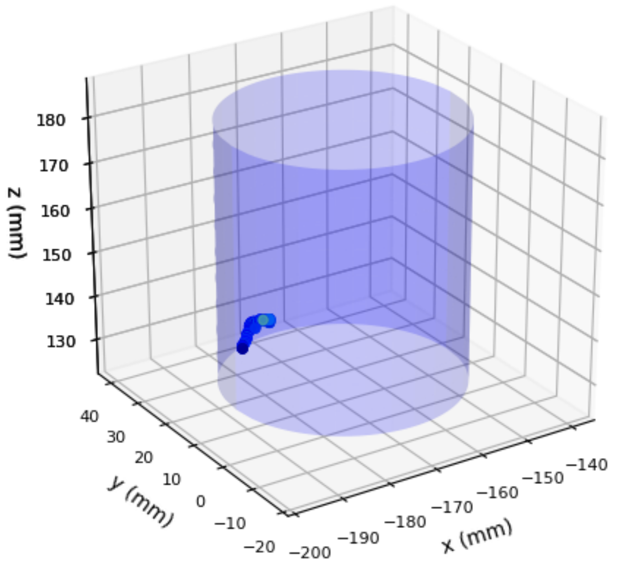
Cut-off Events

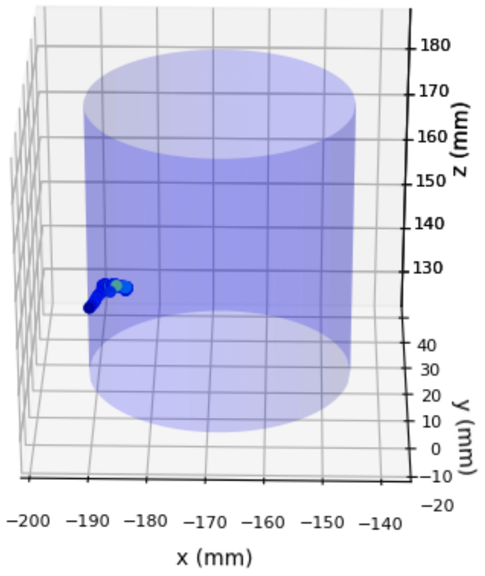
- Sections of event are missing
- Too close to sides of detector
- Alters classification

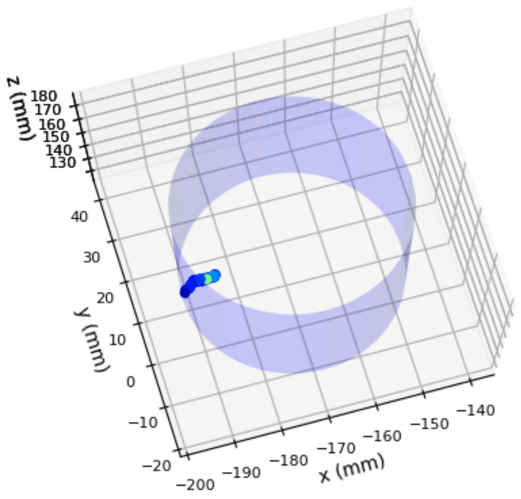

x (mm)
y (mm)
z (mm)
−141
−140
−139
−138
−137
−136
−30
−31
−32
−33
−34
0
1
2
3
4
5

Identify Cut-off Events

- Where in the detector is the event?
- Create image of detector with event inside

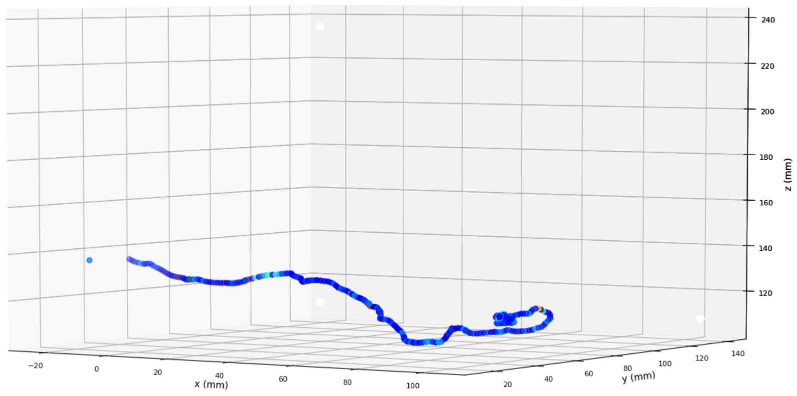


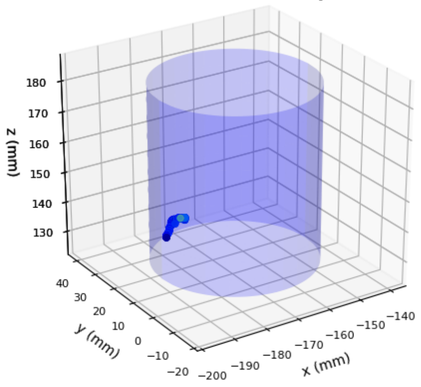


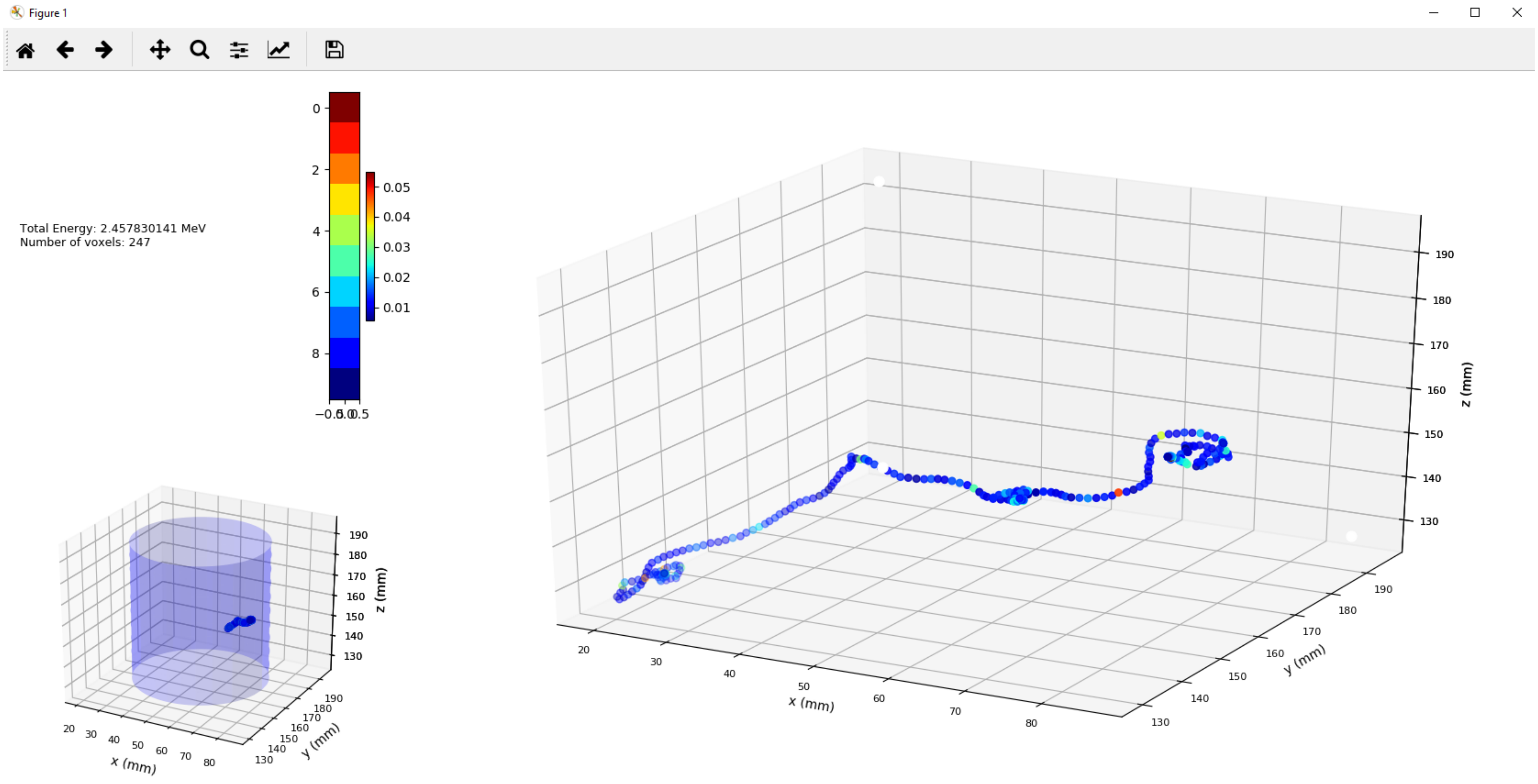


Full Display of Event Info

- 3D event plot
- Energy-color scale
- Location of event within detector
- Other information (text)



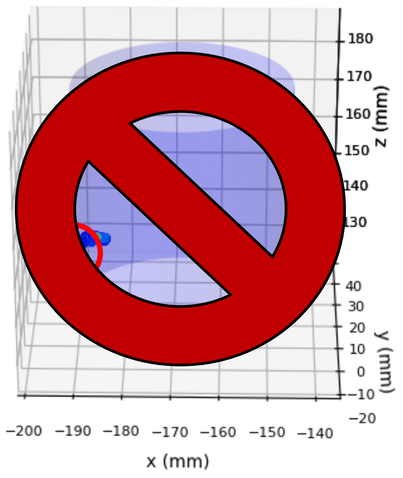


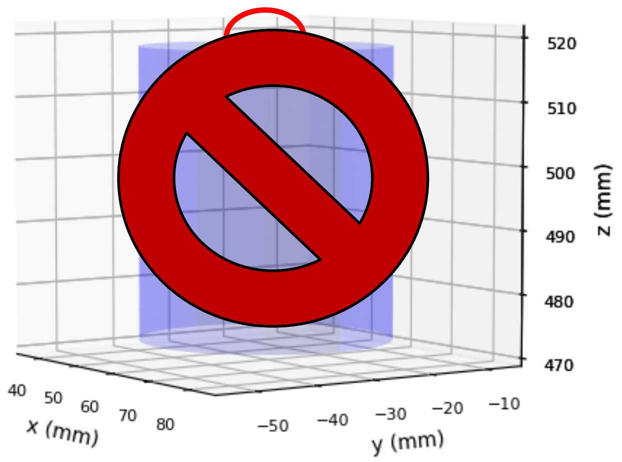


Total Energy: 2.457830141 MeV
Number of voxels: 247

Eliminate Cut-off Events

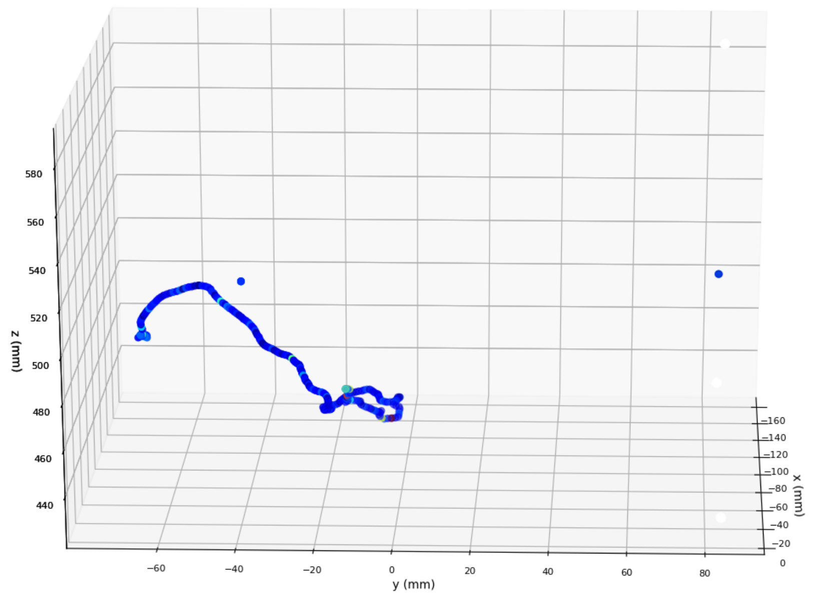
- Impossible to tell $\longrightarrow$ don't look at the event
- Go through coordinates
- Points too close to sides of detector- skip event

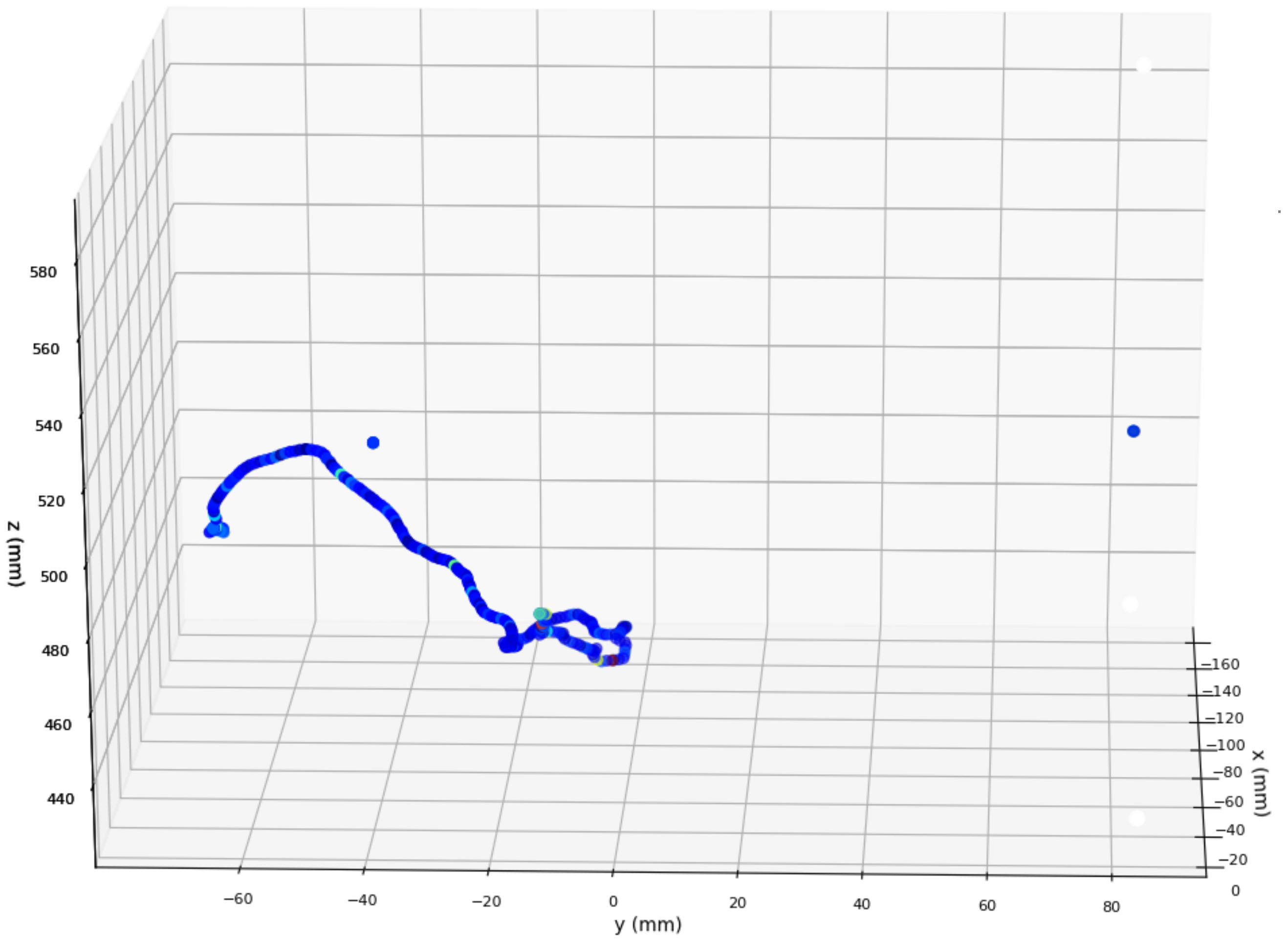


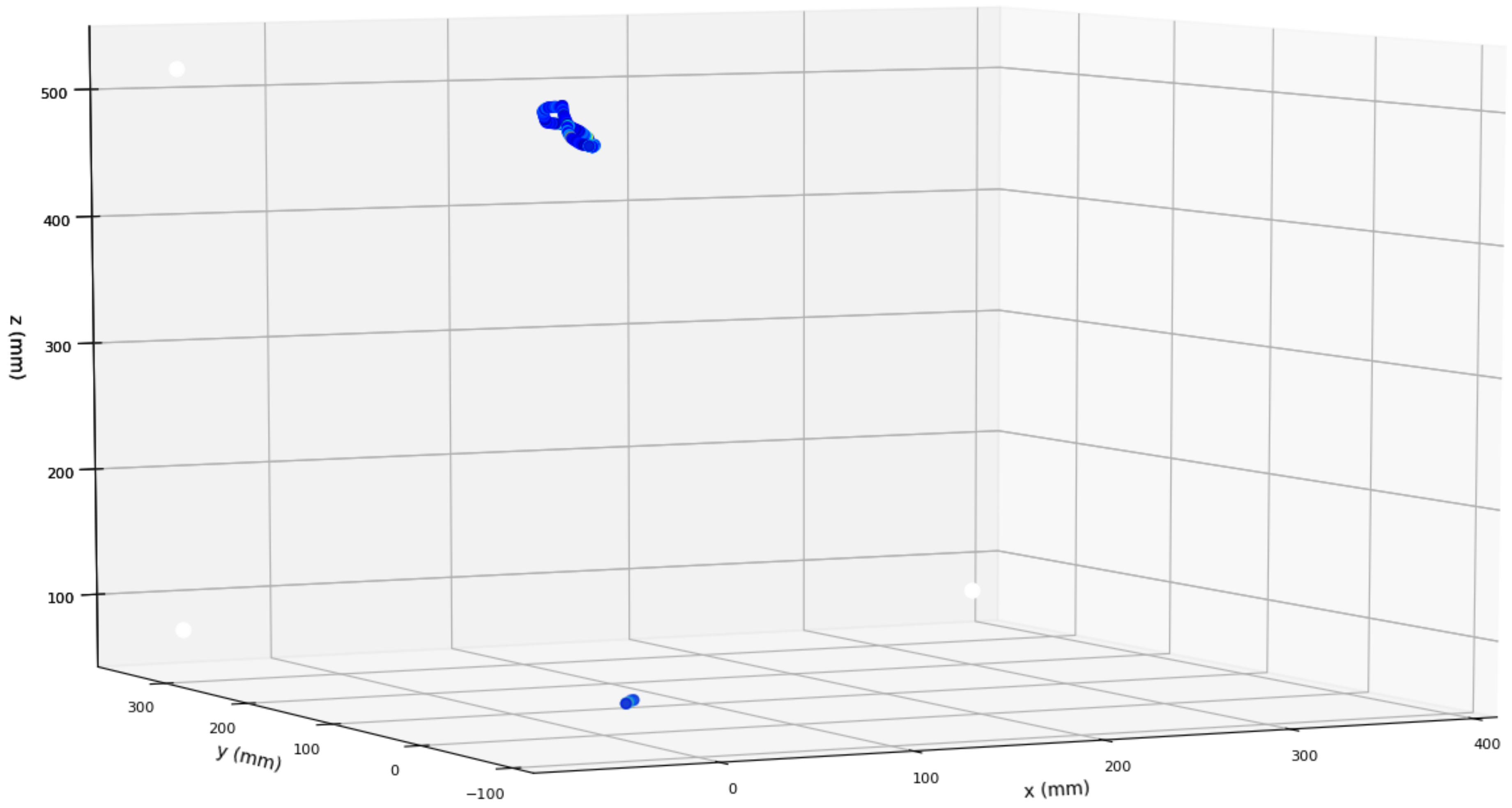


Scattered Events

- Random points far from actual event
- Hard to see the event
- Can’t accurately give a classification

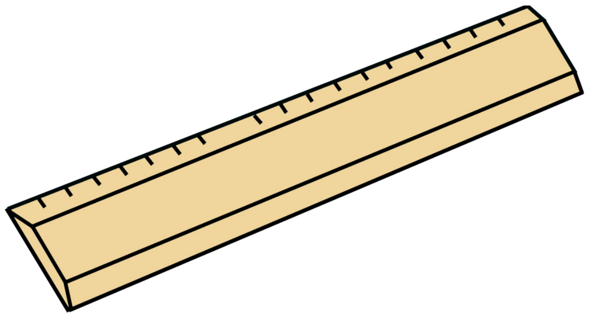


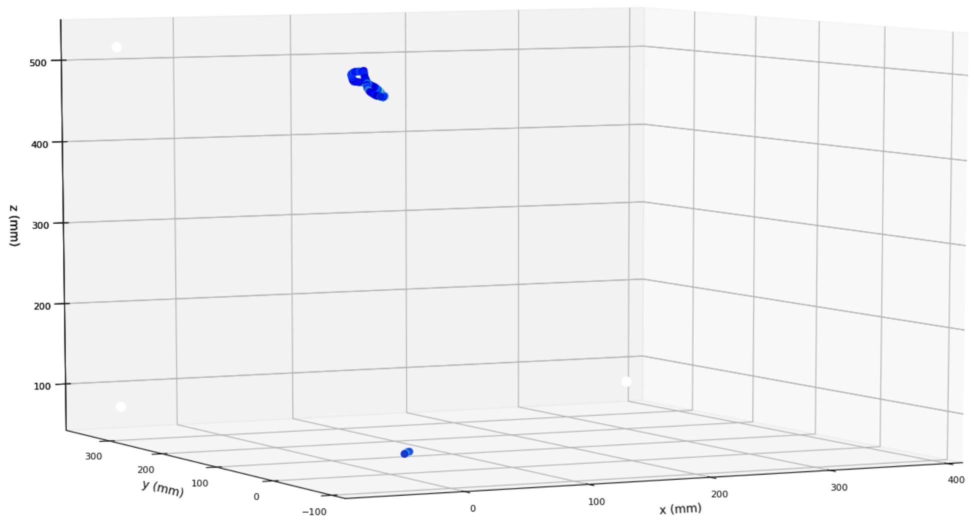

580
560
540
520
500
480
460
440
z (mm)
−60
−40
−20
0
20
40
60
80
y (mm)
−160
−140
−120
−100
−80
−60
−40
−20
0
x (mm)


z (mm)
500
400
300
200
100
y (mm)
300
200
100
0
−100
x (mm)
0
100
200
300
400

Eliminate Scattered Events

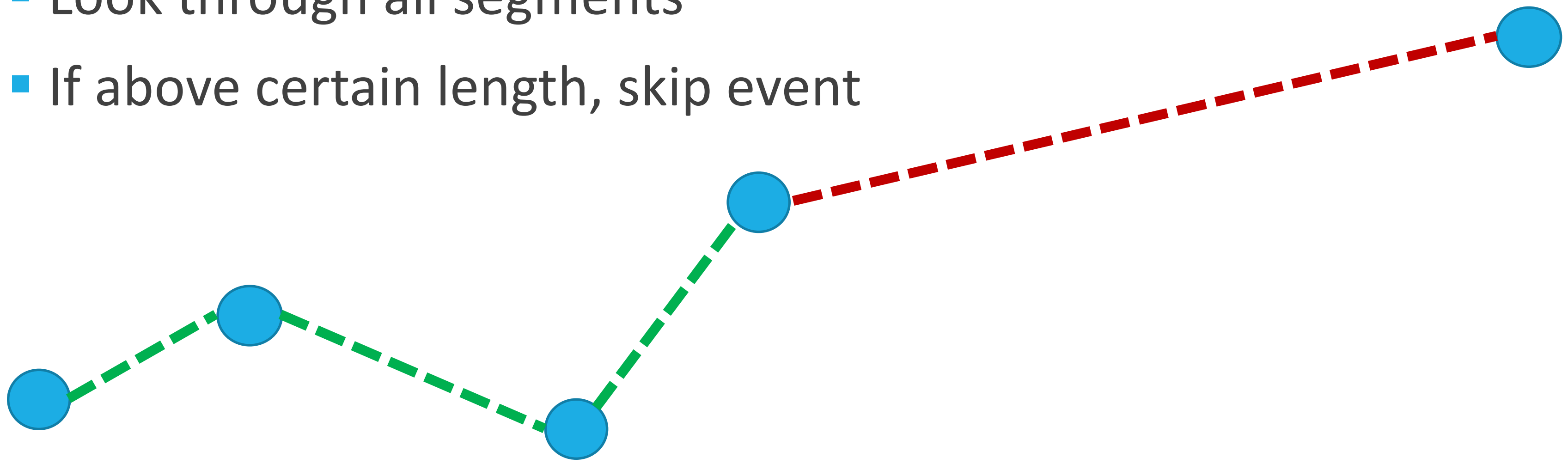
- Calculate distances between points
- Find points far from main event
- If present, skip event

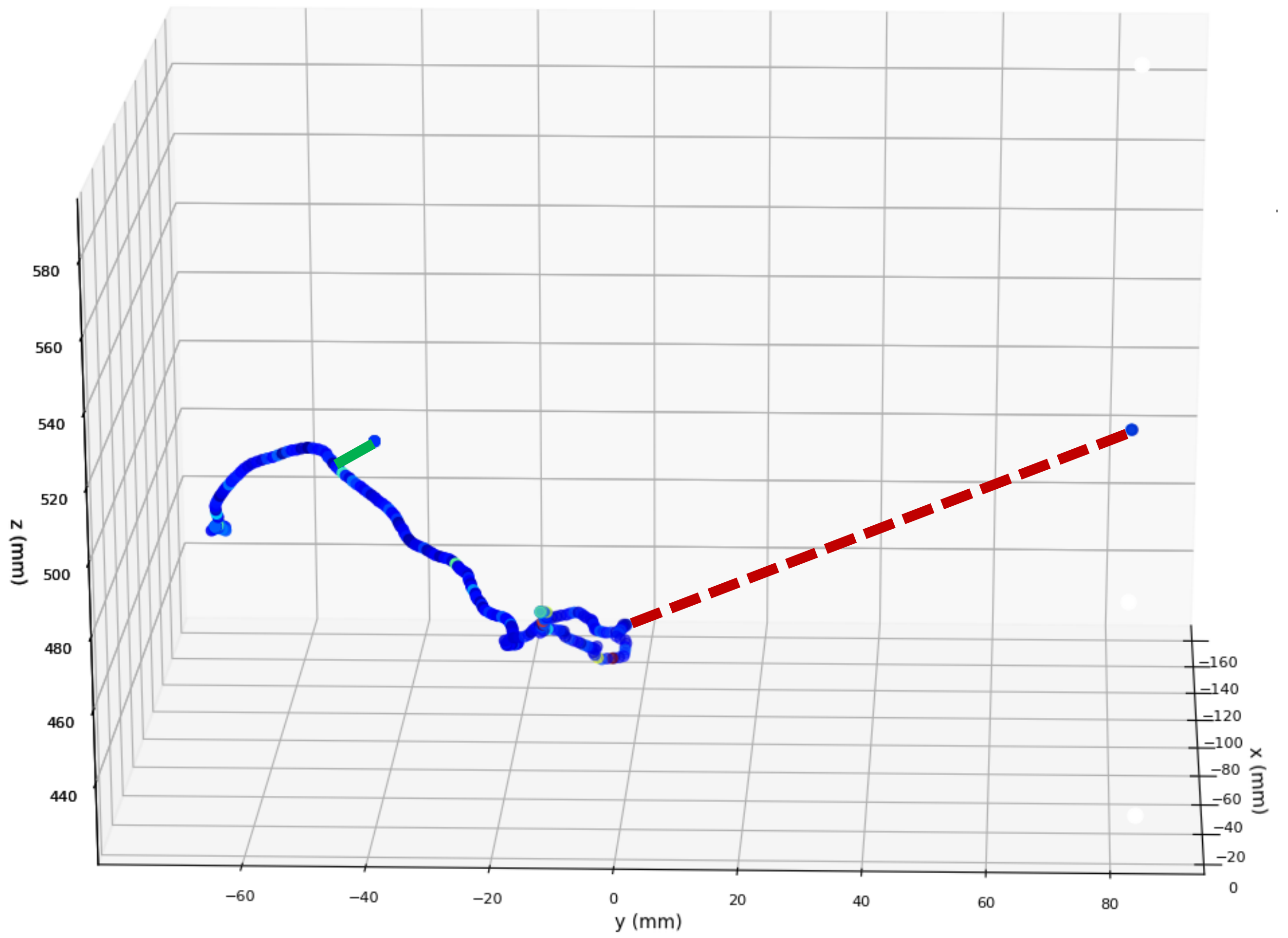


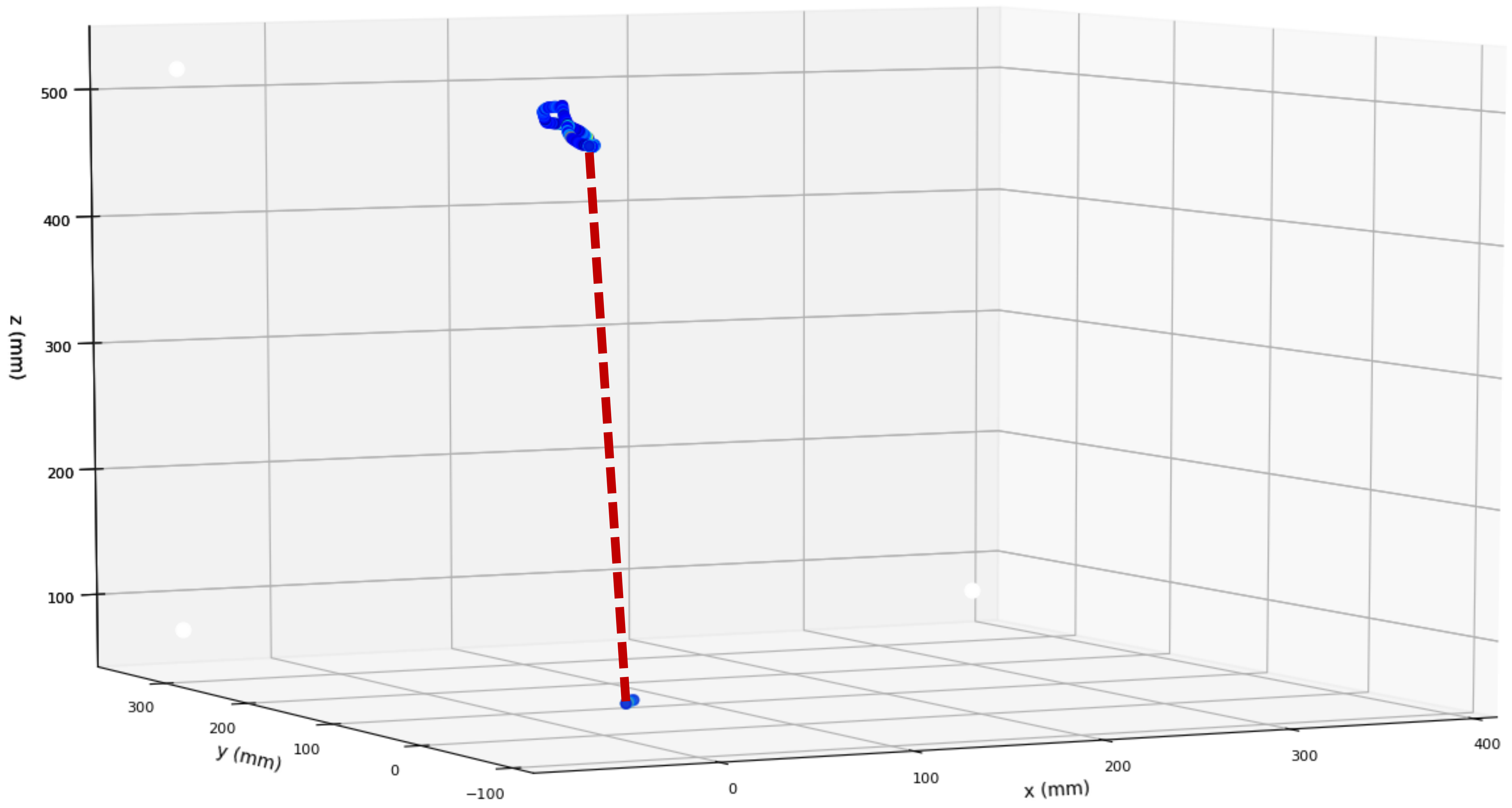


Minimum Spanning Tree

- Maps shortest path between all points
- Look through all segments
- If above certain length, skip event



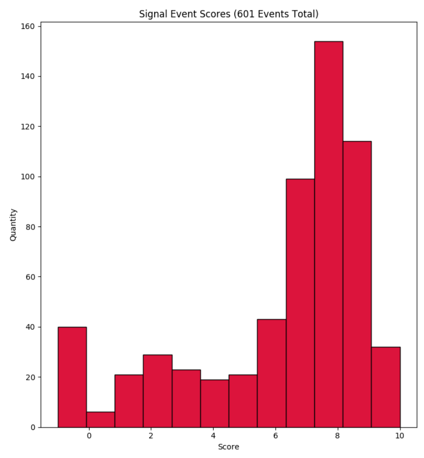

580
560
540
520
500
480
460
440
z (mm)
−60
−40
−20
0
20
40
60
80
y (mm)
−160
−140
−120
−100
−80
−60
−40
−20
0
x (mm)

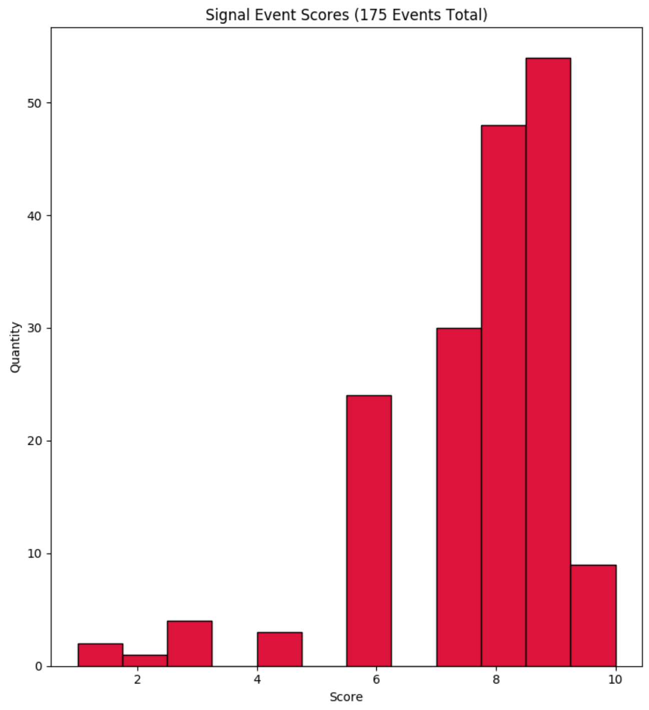


500
400
300
200
100
z (mm)
300
200
100
0
−100
y (mm)
0
100
200
300
400
x (mm)

Second Scan & Scoring of Events

- “Bad” events are now eliminated
- Able to more accurately classify events
- Results show high improvement





Future Ideas/Plans

- Zoom in on event track
- Simulate efficiency of detector
- Automatically sort events based on features
- Who knows??