Updates: August 31

- Modifying calorimeter digitization using layer dependent energy threshold
- Implemented a simple cut on first layer and study occupancy distributions with and without BIB
- Looking into further optimization by tweaking more parameters

Modifying calorimeter digitization

```
// create new collection
LCCollectionVec *ecalcol = new LCCollectionVec(LCIO::CALORIMETERHIT);
ecalcol->setFlag(_flag.getFlag());

// if making gap corrections clear the vectors holding pointers to calhits
if(_ecalGapCorrection!=0){
  for(int is=0;is<MAX_STAVES;is++){
    for(int il=0;il<MAX_LAYERS;il++){
      _calHitsByStaveLayer[is][il].clear();
      _calHitsByStaveLayerModule[is][il].clear();
    }
  }
}

// deal with strips split into virtual cells
//  if this collection is a strip which has been split into virtual cells, they need to be recombined
int orientation = getStripOrientationFromColName( colName );
if ( orientation!=SQUARE && getNumberOfVirtualCells()>1 ) {
  col = combineVirtualStripCells(col, caloLayout == CHT::barrel , orientation );
}


int numElements = col->getNumberOfElements();
streamlog_out ( DEBUG ) << colName << " number of elements = " << numElements << endl;

for (int j(0); j < numElements; ++j) {
  SimCalorimeterHit * hit = dynamic_cast<SimCalorimeterHit*>( col->getElementAt( j ) ) ;
  float energy = hit->getEnergy();

  // redefine energy threshold for first layer to be 5 MeV (by S.Lomte)
  if (idDecoder(hit)["layer"] < 2) {
    _thresholdEcal = 5e-03 ; //5MeV
  }

  // apply threshold cut
  if (energy > _thresholdEcal) {
    int cellid = hit->getCellID0();
    int cellid1 = hit->getCellID1();
    int layer = idDecoder(hit)["layer"];
    int stave = idDecoder(hit)["stave"];
    int module= idDecoder(hit)["module"];

    // check that layer and assumed layer type are compatible
    checkConsistency(colName, layer);

    // save hits by module/stave/layer if required later
    float calibr_coeff(1.);
    float x = hit->getPosition()[0];
    float y = hit->getPosition()[1];
    float z = hit->getPosition()[2];
    float r = sqrt(x*x+y*y+z*z);
    float rxy = sqrt(x*x+y*y);
    float cost = fabs(z)/r;
```

Simple cut: for layer1 in ECAL, the threshold E_{hit} is increased from 2MeV to 5 (10) MeV.

Default: 2MeV

On varying this E_{th}, study occupancy distribution of calo hits in ECal Barrel :

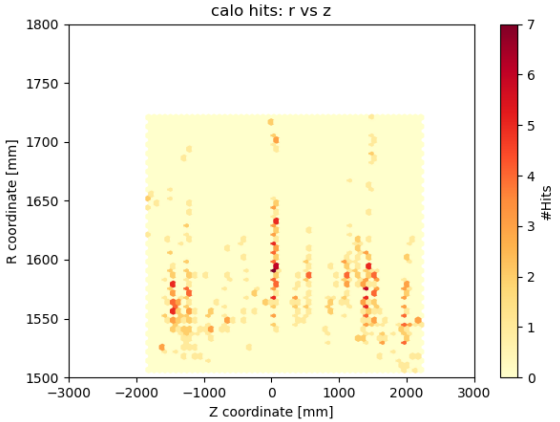
- X, Y, Z coordinates
- $R_{xy} = \sqrt{X^2 + Y^2}$
- Energy
- Timing
- Layer number using CellID of hits (ongoing)

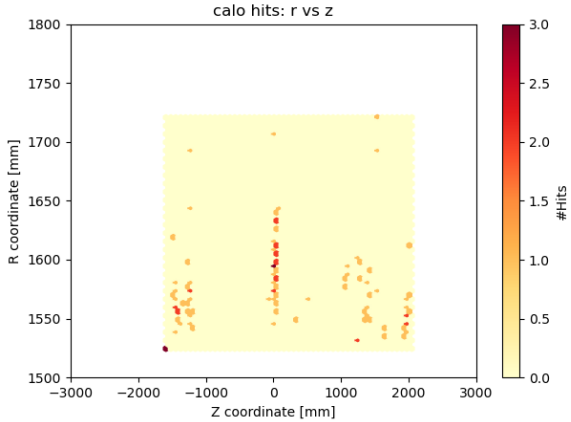
Signal process:
$\mu^+\mu^- \rightarrow H\nu\bar{\nu}, H \rightarrow b\bar{b}$ at 3 TeV
10 events each overlaid with 1 BIB event (full bunch crossing)

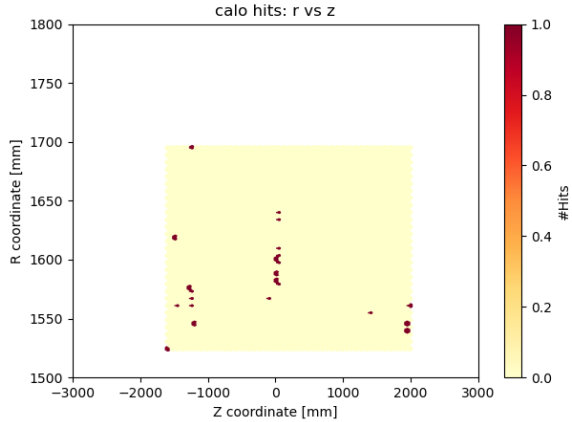
R_{xy} vs Z coordinate

2MeV	5MeV	10MeV

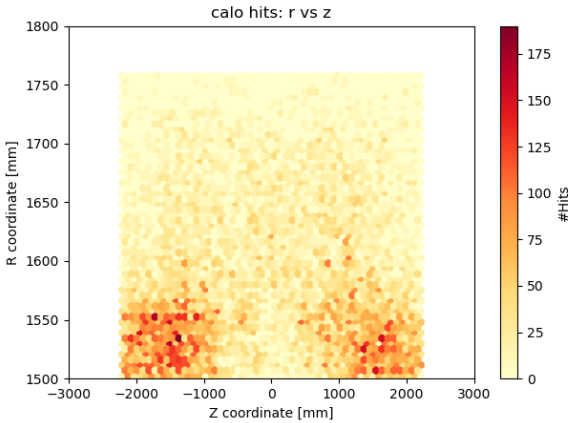
Signal only

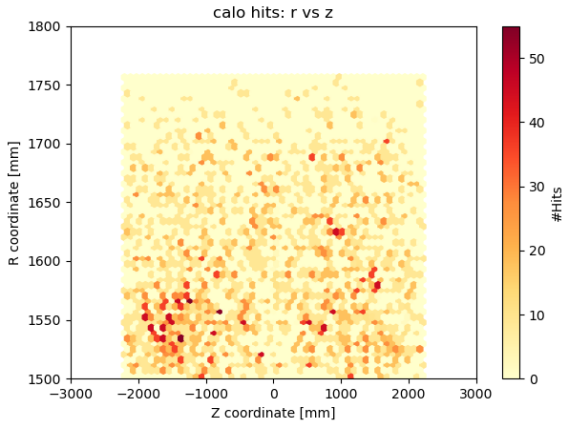


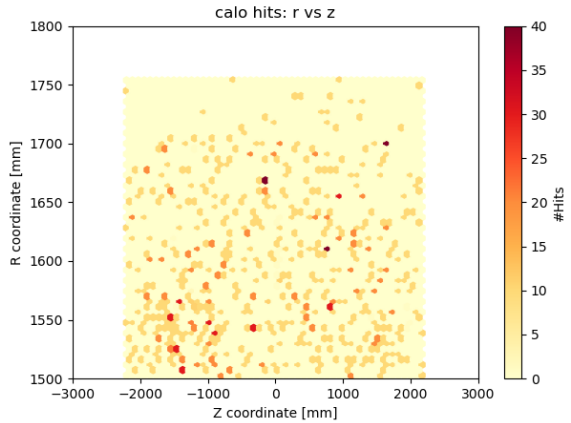




Signal + BIB



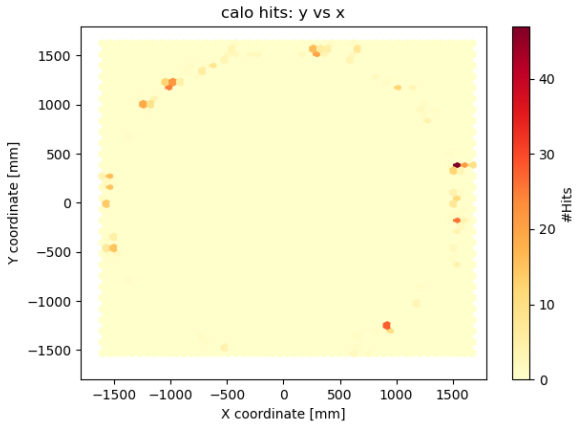


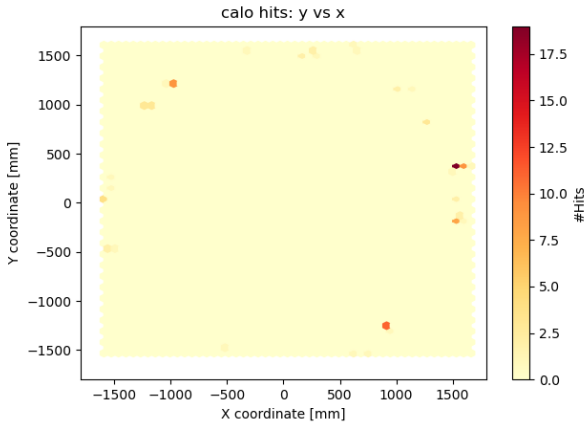


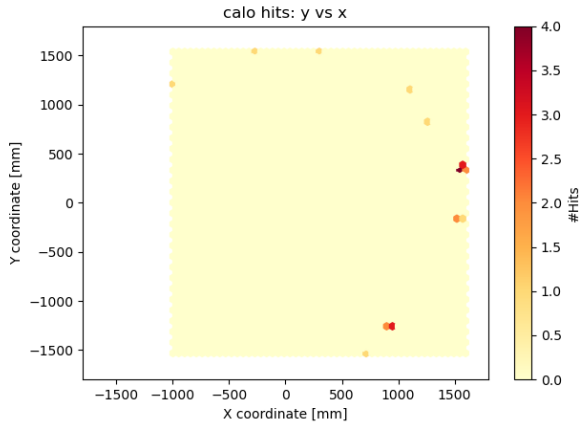
Y vs X coordinate

2MeV 5MeV 10MeV

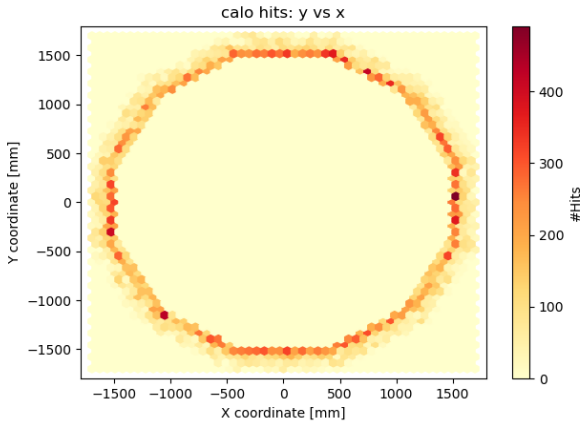
Signal only

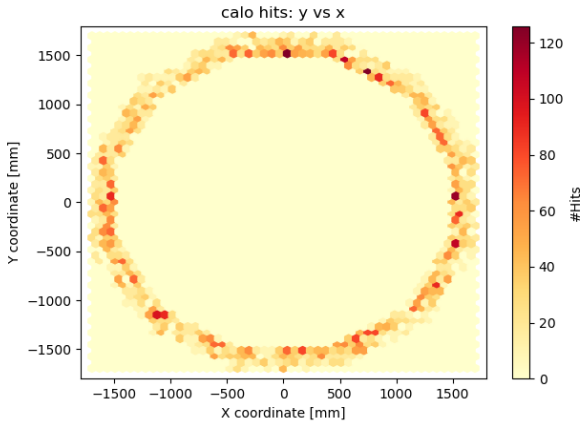


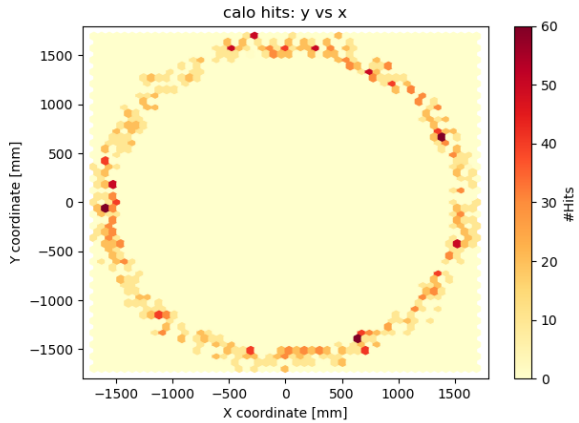




Signal + BIB







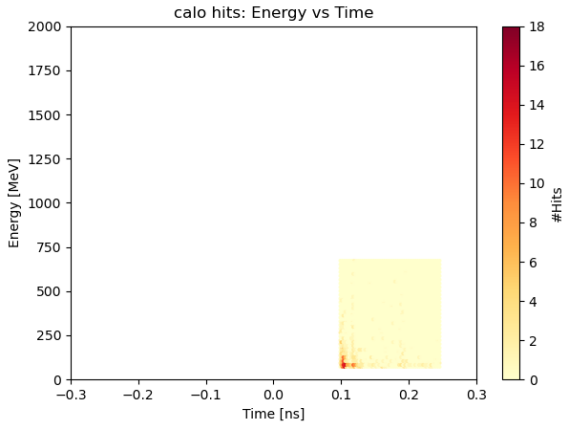
Energy vs Time

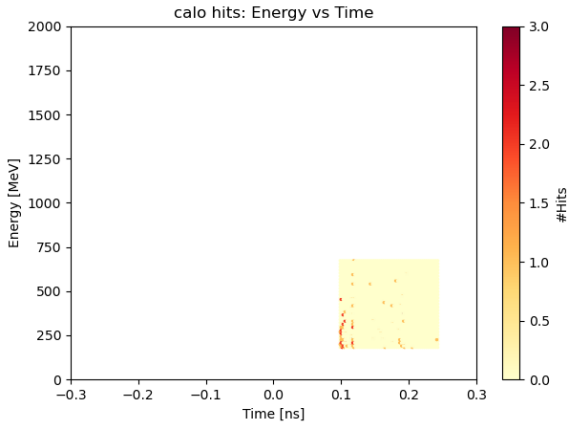
2MeV

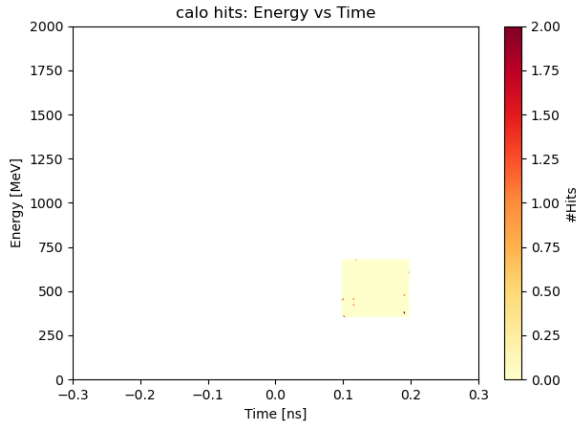
5MeV

10MeV

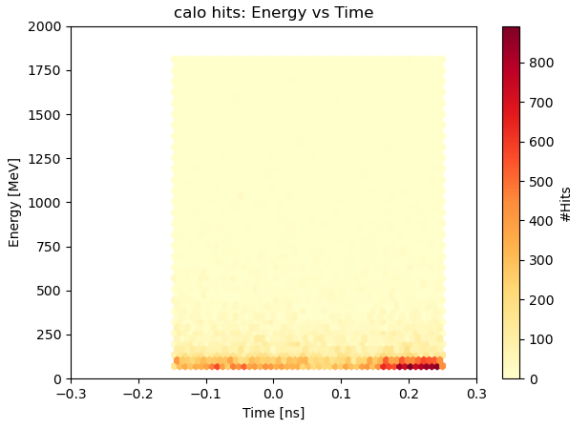
Signal only

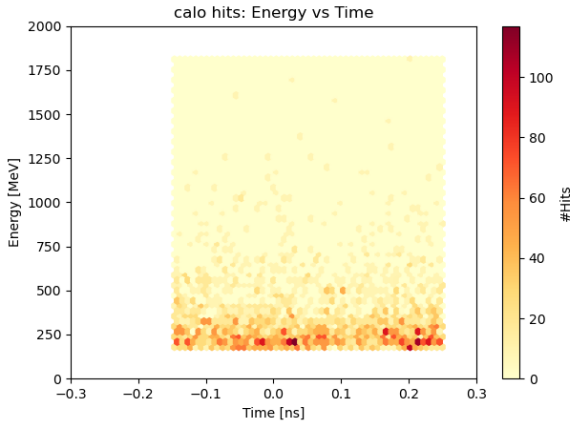


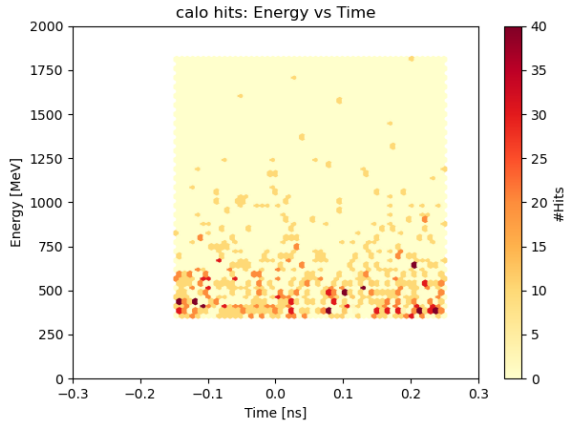




Signal + BIB



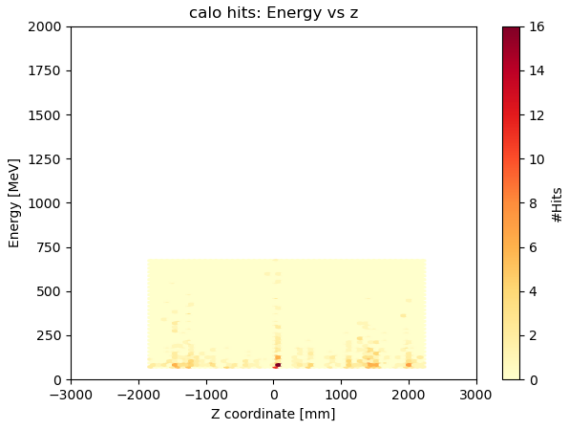


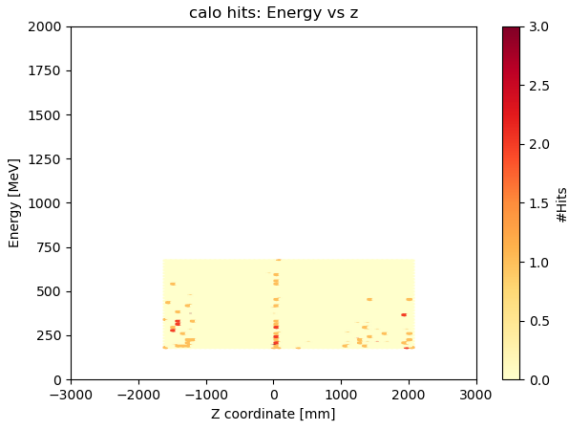


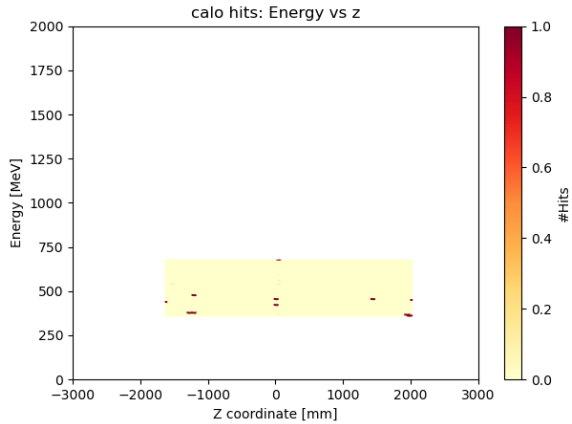
Energy vs Z coordinate

2MeV 5MeV 10MeV

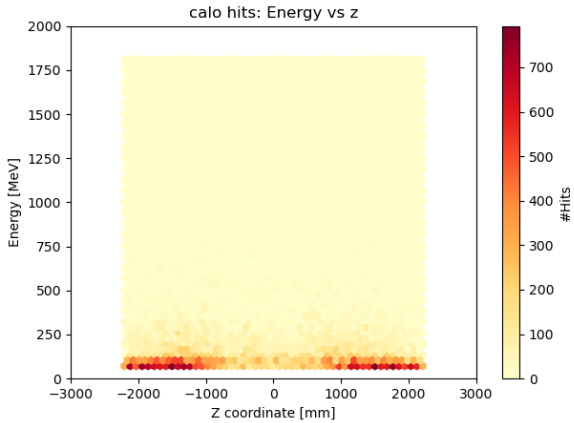
Signal only

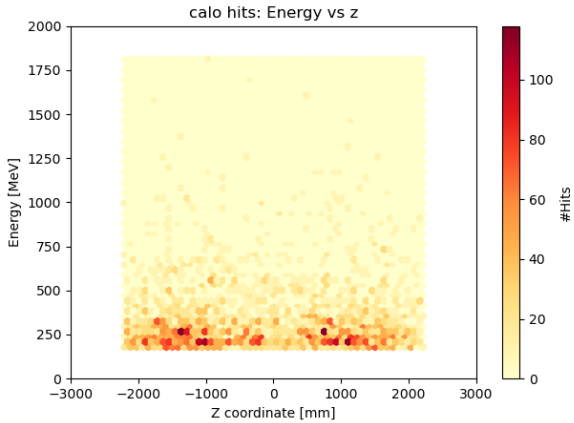


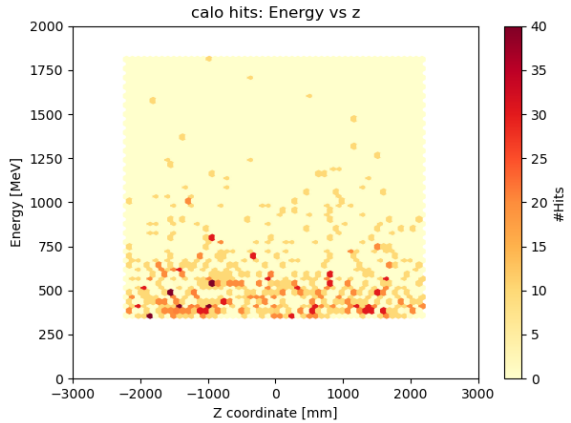




Signal + BIB



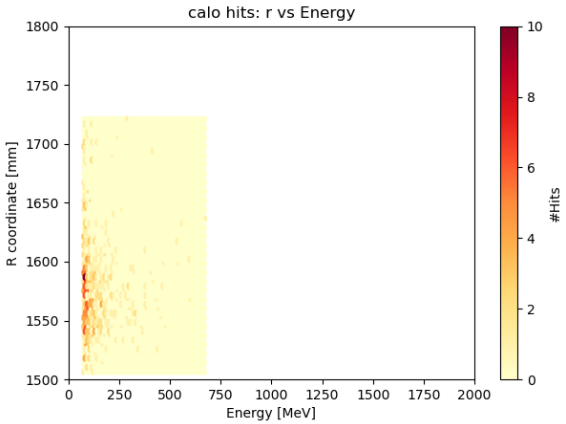


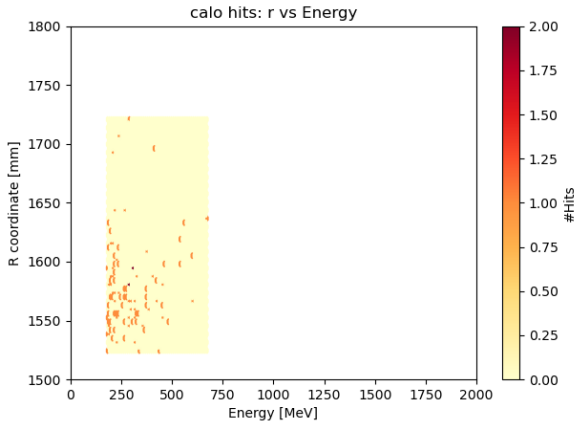


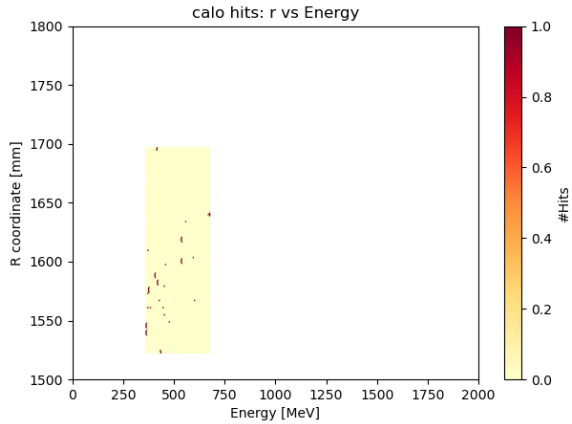
R_{xy} coordinate vs Energy

2MeV 5MeV 10MeV

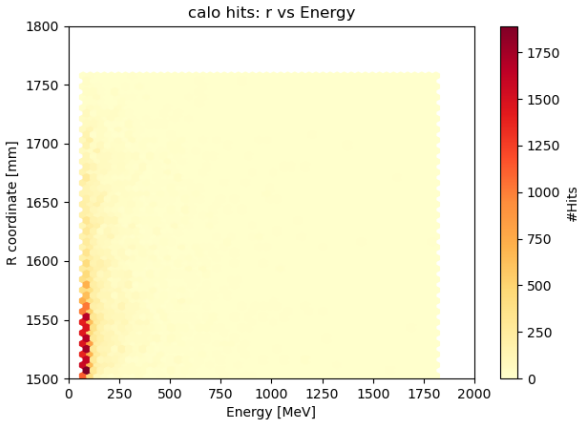
Signal only

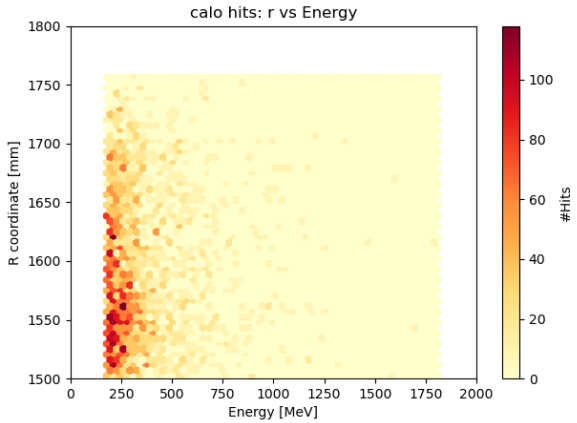


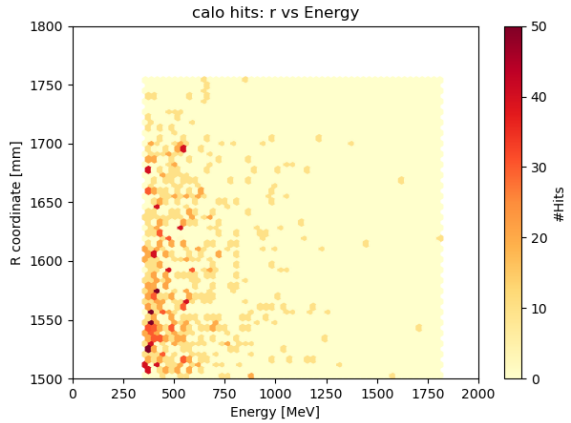




Signal + BIB







Average number of ECal Barrel hits in 10events:

E cut	**Signal only**	**Signal+BIB**
2 MeV	52.9	6632.2
5 MeV	9.7 (18%)	1976.8 (29.8%)
10 MeV	2.3 (4.3%)	571.3 (8.6%)

This was a simple cut, can tweak other parameters to optimize signal vs signal+BIB hits

Parameters in CaloDigitiser: ECAL

```
[ MESSAGE "CaloDigitiser"]
[ MESSAGE "CaloDigitiser"] ---- CaloDigitiser -  parameters:
[ MESSAGE "CaloDigitiser"]      CalibECALMIP:  0.0001
[ MESSAGE "CaloDigitiser"]      CalibHCALMIP:  0.0001
[ MESSAGE "CaloDigitiser"]      CalibrECAL:  35.8411 35.8411
[ MESSAGE "CaloDigitiser"]      CalibrHCALBarrel:  49.2031
[ MESSAGE "CaloDigitiser"]      CalibrHCALEndcap:  53.6263
[ MESSAGE "CaloDigitiser"]      CalibrHCALOther:  62.2126
[ MESSAGE "CaloDigitiser"]      ECALBarrelTimeWindowMax:  0.25
[ MESSAGE "CaloDigitiser"]      ECALCollections:  ECalBarrelCollection ECalEndcapCollection ECalPlugCollection
[ MESSAGE "CaloDigitiser"]      ECALCorrectTimesForPropagation:  1
[ MESSAGE "CaloDigitiser"]      ECALDeltaTimeHitResolution:  10
[ MESSAGE "CaloDigitiser"]      ECALEndcapCorrectionFactor:  1.06721
[ MESSAGE "CaloDigitiser"]      ECALEndcapTimeWindowMax:  0.25
[ MESSAGE "CaloDigitiser"]      ECALGapCorrection:  1
[ MESSAGE "CaloDigitiser"]      ECALGapCorrectionFactor:  1
[ MESSAGE "CaloDigitiser"]      ECALLayers:  41 100
[ MESSAGE "CaloDigitiser"]      ECALModuleGapCorrectionFactor:  0
[ MESSAGE "CaloDigitiser"]      ECALOutputCollection0:  ECALBarrelHits
[ MESSAGE "CaloDigitiser"]      ECALOutputCollection1:  ECALEndcapHits
[ MESSAGE "CaloDigitiser"]      ECALOutputCollection2:  ECALOtherHits
[ MESSAGE "CaloDigitiser"]      ECALSimpleTimingCut:  true
[ MESSAGE "CaloDigitiser"]      ECALThreshold:  0.002
[ MESSAGE "CaloDigitiser"]      ECALThresholdUnit:  GeV
[ MESSAGE "CaloDigitiser"]      ECALTimeResolution:  0.25
[ MESSAGE "CaloDigitiser"]      ECALTimeWindowMin:  -0.25
[ MESSAGE "CaloDigitiser"]      ECAL_PPD_N_Pixels:  10000
[ MESSAGE "CaloDigitiser"]      ECAL_PPD_N_Pixels_uncertainty:  0.05
[ MESSAGE "CaloDigitiser"]      ECAL_PPD_PE_per_MIP:  7
[ MESSAGE "CaloDigitiser"]      ECAL_apply_realistic_digi:  0
[ MESSAGE "CaloDigitiser"]      ECAL_deadCellRate:  0
[ MESSAGE "CaloDigitiser"]      ECAL_deadCell_memorise:  false
[ MESSAGE "CaloDigitiser"]      ECAL_default_layerConfig:  0000000000000000
```

Next steps:

- Modify Timing window and Resolution
- Check what is 'ECAL_apply_realistic_digi' ??

CaloDigitiser.xml ECAL Timing section

```
<!--Use ECAL hit times-->
<parameter name="UseEcalTiming" type="int">1 </parameter>
<!--Correct ECAL hit times for propagation: radial distance/c-->
<parameter name="ECALCorrectTimesForPropagation" type="int">1 </parameter>
<!--ECAL Time Window minimum time in ns-->
<parameter name="ECALTimeWindowMin" type="float">-0.250 </parameter>
<!--ECAL BarrelTime Window maximum time in ns-->
<parameter name="ECALBarrelTimeWindowMax" type="float">0.250 </parameter>
<!--ECAL Endcap Time Window maximum time in ns-->
<parameter name="ECALEndcapTimeWindowMax" type="float">0.250 </parameter>
<!--Use simple time window cut on hit times? If false: use original hit-time clustering algorithm.
    If true: use time window defined by ECALBarrelTimeWindowMin and ECALBarrelTimeWindowMax-->
<parameter name="ECALSimpleTimingCut" type="bool">true </parameter>
<!--ECAL Minimum Delta Time in ns for resolving two hits-->
<!-- used only if ECALSimpleTimingCut is false -->
<parameter name="ECALDeltaTimeHitResolution" type="float">10 </parameter>
<!--ECAL Time Resolution used to smear hit times-->
<parameter name="ECALTimeResolution" type="float">0.250 </parameter>
```