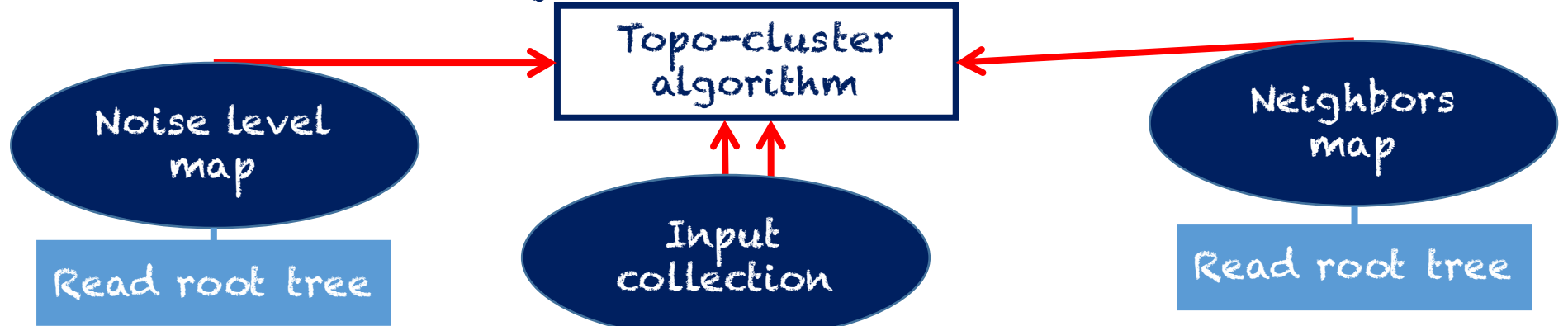


# PR #293

- Topo-Clustering

Runs on digitised Calo cells (CaloHitCollection)



- Interface: ICaloReadCellNoiseMap
- Tool: TopoCaloNeighbours (expects input file path to root file)

- Interface: ITopoClusterInputTool
- Tool: CaloTopoClusterInputTool

- Interface: ICaloReadCellNoiseMap
- Tool: TopoCaloNoisyCells (expects input file path to root file)

Geometry independent  
Reconstruction/RecCalorimeter

Geometry dependent  
Reconstruction/RecFCChhCalorimeter

Create noise  
root tree

Cell  
positions  
tool

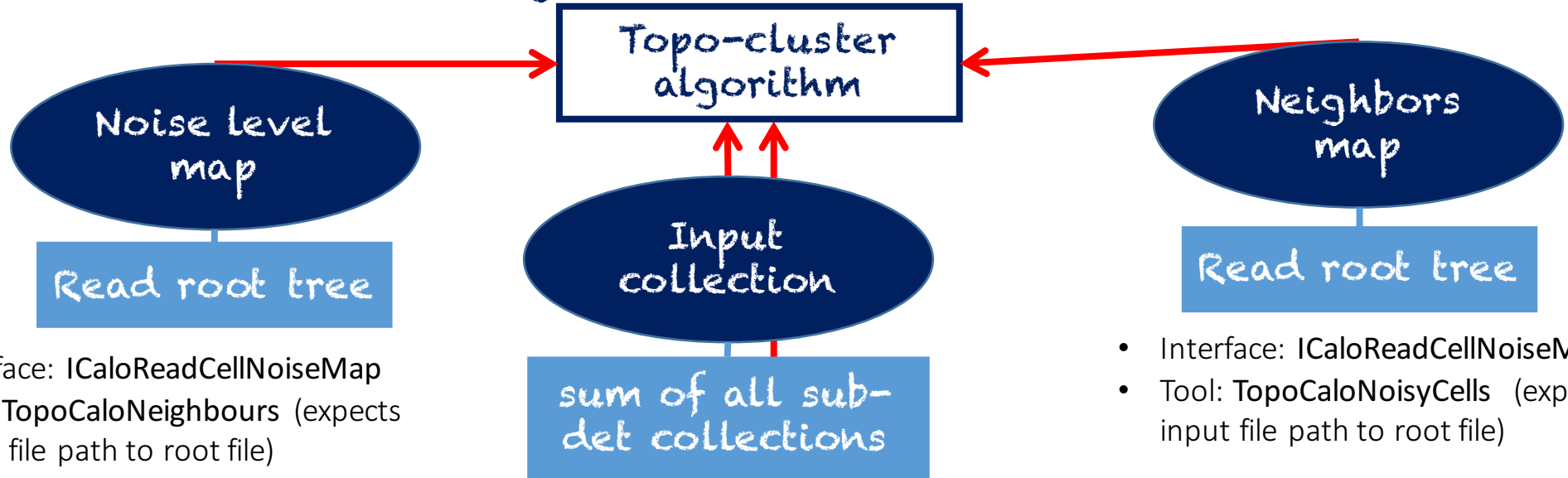
Create neighbours  
root tree

- Interface: INoiseConstTool
  1. ConstNoiseTool (same noise level for all)
  2. ReadNoiseFromFileTool (input file, hist name ect.)

- CreateFCChhCaloNeighbours (only Barrel)
  - Includes direct/diagonal neighboursAdditions in:

Detector/DetCommon/src/DetUtils.cpp

Runs on digitised Calo cells (CaloHitCollection)



- Interface: ICaloReadCellNoiseMap
- Tool: TopoCaloNeighbours (expects input file path to root file)

- Interface: ICaloReadCellNoiseMap
- Tool: TopoCaloNoisyCells (expects input file path to root file)

- Interface: ITopoClusterInputTool
- Tool: CaloTopoClusterInputTool

Geometry independent  
Reconstruction/RecCalorimeter

Geometry dependent  
Reconstruction/RecFCChhCalorimeter

- Interface: ICellPositionsTool
  1. CellPositionsECalBarrelTool
  2. CellPositionsCaloDiscsTool
  3. CellPositionsHCalBarrelNoSegTool
  4. CellPositionsTailCatcherTool-> input: detector readout name



Also needed/used for conversion in [FCC-hh-framework/FCCSimJobs](#) from edm to ntuple, which is input for jet reconstruction

[.../src/components/CreateCellPositions.cpp](#)  
Tested in:  
[.../tests/options/recoPositions\\_fullCaloSystem.py](#)

# Geometry dependent Reconstruction/RecFCChCalorimeter

Create noise  
root tree

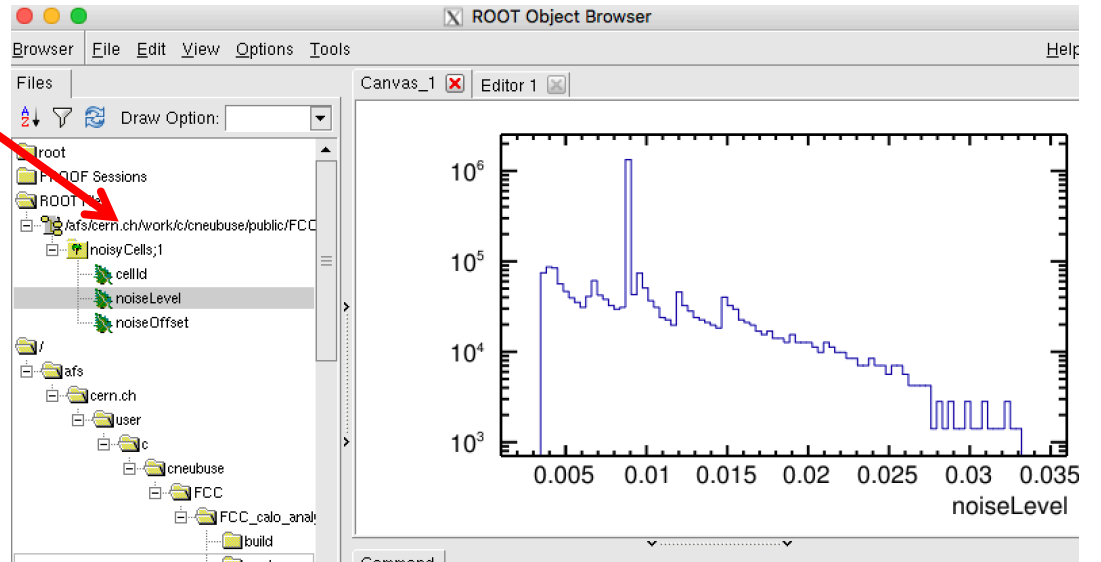
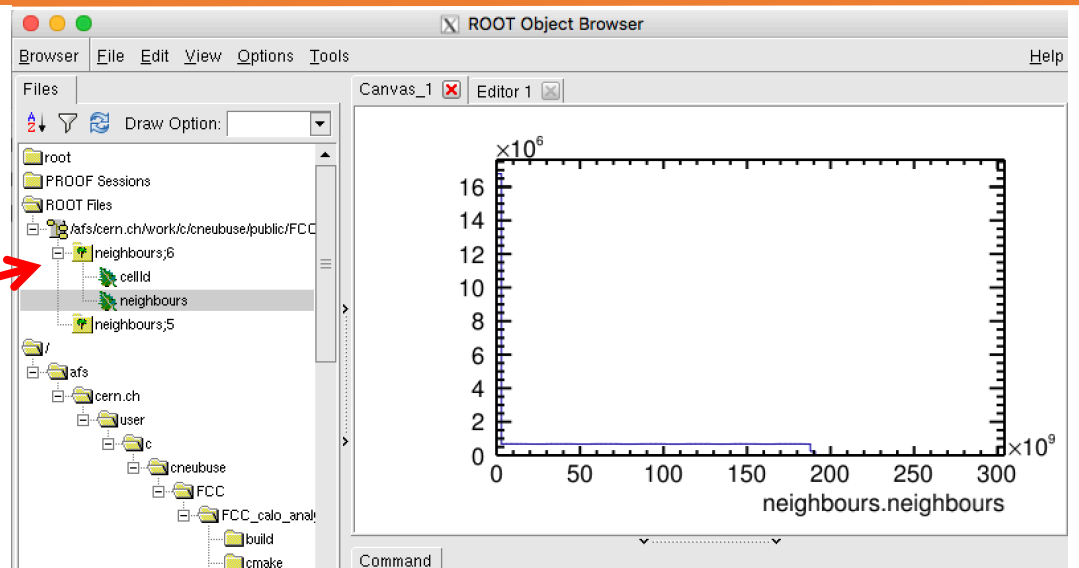
.../tests/options/neighbours.py

Create neighbours  
root tree

.../tests/options/noiseLevelPerCell.py

Cell  
positions  
tool - TEST

.../tests/options/recoPositions\_fullCaloSystem.py



## Geometry independent Reconstruction/RecCalorimeter

### CaloTopoClusterInput

- returns `std::map<cellID, energy>`

### TopoCaloNoisyCells

```
readNoisyCellsMap = TopoCaloNoisyCells("ReadNoisyCellsMap",  
                                       fileName = "/afs/cern.ch/work/c/cneubuse/public/FCChh/cellNoise_map_segHcal_constNoiseLevel",  
                                       OutputLevel = DEBUG)
```

- returns double noise RMS/Mean by cellID

### TopoCaloNeighbours

```
readNeighboursMap = TopoCaloNeighbours("ReadNeighboursMap",  
                                       fileName = "/afs/cern.ch/work/c/cneubuse/public/FCChh/neighbours_map_segHcal.root",  
                                       OutputLevel = DEBUG)
```

- returns `std::vec<cellID>` by cellID

.../[src/components/CaloTopoCluster.cpp](#)  
Tested in :

.../[tests/options/runBarrelCaloSystem\\_ReconstructionTopoClusters\\_noNoise.py](#)

```
from Configurables import CaloTopoClusterInputTool, CaloTopoCluster, TopoCaloNeighbours, TopoCaloNoisyCells  
createTopoInput = CaloTopoClusterInputTool("CreateTopoInput",  
                                           ecalBarrelReadoutName = ecalBarrelReadoutName,  
                                           ecalEndcapReadoutName = "",  
                                           ecalFwdReadoutName = "",  
                                           hcalBarrelReadoutName = hcalBarrelReadoutName,  
                                           hcalExtBarrelReadoutName = "",  
                                           hcalEndcapReadoutName = "",  
                                           hcalFwdReadoutName = "",  
                                           OutputLevel = DEBUG)  
  
createTopoInput.ecalBarrelCells.Path = "ECalBarrelCells"  
createTopoInput.ecalEndcapCells.Path = "emptyCaloCells"  
createTopoInput.ecalFwdCells.Path = "emptyCaloCells"  
createTopoInput.hcalBarrelCells.Path = "HCalBarrelCells"  
createTopoInput.hcalExtBarrelCells.Path = "emptyCaloCells"  
createTopoInput.hcalEndcapCells.Path = "emptyCaloCells"  
createTopoInput.hcalFwdCells.Path = "emptyCaloCells"
```

### CaloTopoCluster Algorithm

```
createTopoClusters = CaloTopoCluster("CreateTopoClusters",  
                                     TopoClusterInput = createTopoInput,  
                                     # expects neighbours map from cellid->vec<neighbourIds>  
                                     neighboursTool = readNeighboursMap,  
                                     # tool to get noise level per cellid  
                                     noiseTool = readNoisyCellsMap,  
                                     # cell positions tools for all sub-systems  
                                     positionsECalBarrelTool = ECalBcells,  
                                     positionsHCalBarrelTool = HCalBcellVols,  
                                     positionsHCalExtBarrelTool = HCalExtBcellVols,  
                                     positionsEMECTool = EMECTcells,  
                                     positionsHECTool = HECTcells,  
                                     positionsEMFwdTool = ECalFwdcells,  
                                     positionsHFwdTool = HCalFwdcells,  
                                     seedSigma = 4,  
                                     neighbourSigma = 0,  
                                     lastNeighbourSigma = 0,  
                                     OutputLevel = INFO)  
  
createTopoClusters.clusters.Path = "caloClustersBarrel"  
createTopoClusters.clusterCells.Path = "caloClusterBarrelCells"
```