Updates to the Directed Tree clustering algorithm

- Goals:
  - Optimization within the LCSim reconstruction framework
  - Make the DTree clustering algorithm available to others
  - Optimize the physics performance

- Key concepts behind the LCSim-related changes:
  - Maximize the use of standard utilities from the framework
  - Use of modified hitmaps and cluster interfaces to create "plugin"-like algorithms

Recent changes (1)

- MCReconstruction vs. CheatClusterer
  - Almost perfect agreement between MCReconstruction and CheatClusterer
  - CheatClusterer: look at MCinfo to define the clusters
    "perfect", impossible to be achieved by real reconstruction
  - Discrepancies mostly (if not always) due to differences between LCDG4 and SLIC on MCParticle table components
    backscatters and, in some cases, decay products (when backscatters?)
  - Still needed: look at energy cut vs. use of cheat clusters
- MCReconstruction, MCUtil and MCClusters: deprecated

Recent changes (2)

- MyCalorimeterHit vs. CalorimeterHit
  - Keep track of hit densities in LoadMyCalorimeterHit
    (LoadMyCalorimeterHit is a singleton, available to any class)
    Then `double kdens = (MyCalorimeterHit)khit.getDensity(0);`
    becomes `double kdens = _loader.getDensity(khit);`
    where `_loader = LoadMyCalorimeterHit.getInstance();`
  - A minor but significant redefinition of hit density:
    - Isolated hit - density = 1
    - Discarded hit - density = 0
- MyCalorimeterHit class is now deprecated

Recent changes (3)

- Cluster extends BasicCluster
  - Easier for other algorithms to use DirectedTree clusters, as these now contain a collection of standard CalorimeterHits
  - Each BasicCluster object points to a ClusterPropertyCalculator object for position calculation (also shape, position error, theta, phi and direction)
  - A DensityWeightingCalculator has been implemented

    to be tested soon
- Cluster class is likely to become also deprecated

  pending on the results of the DensityWeightingCalculator tests

Summary of classes

- Classes currently used by the DirectedTree algorithm
  - Pflow (main driver)
  - DirectedTree (subdriver with clustering algorithm)
  - LoadMyCalorimeterHit (hit selection + hit density info)
  - ClusterBuilder (builds a List<Cluster> from a Vector<Vector<CalorimeterHit>>)
  - Cluster (to be removed?)
  - DensityWeightingCalculator (new, defines density weighting)
  - RunControlParameters (algorithm parameters)
  - CalculateDistance (between two points)
  - Kinem (theta,phi of a point)
  - MyTools (utility class for debugging)

Next steps

- Use/certify DensityWeightingCalculator
- Deprecate Cluster class
- Test appending DirectedTree Clusters to event for downstream analysis
- Split DirectedTree into an algorithm proper class (no MC handling) and an analysis class (histo+ntuple filling)
- Closer look at DirectedTree reconstruction algorithm
  Need to get analysis code (Vishnu or Rob)