PFA interface updates

Mat Charles, U. Iowa

Added Niels's photon-finder

In org.lcsim.recon.cluster.mst:

+ MSTPhotonFinderDriver extends Driver
+ MSTPhotonFinder implements Clusterer

Two-pass approach: Find cores, then add nearby fragments.
Photon shape cuts not strong (feed into H-matrix?)

Inside a PFA, have to be a bit careful of MIPs:

1) Find MIP clusters in ECAL near inner surface
2) Look which ones are matched to tracks & remove them
3) Apply photon-finder to all remaining ECAL hits
4) Remove photon clusters
5) Run MIP-finder again

Charged particle finder

Refactored the interface. In org.lcsim.recon.pfa.identifier:

```
public interface TrackClusterMatcher {
    public Cluster matchTrackToCluster(Track, List<Cluster>);
}
```

Could be one cluster from the input list
Could be a BasicCluster composed of several (Guilherme)

Match 1 track to 0-1 cluster (nearest cluster within range):
SimpleTrackClusterMatcher extends Driver implements TrackClusterMatcher

Match 1 track to 0-1 MIP-like cluster (proximity and direction at intercept):
SimpleTrackMIPClusterMatcher extends SimpleTrackClusterMatcher

Match List<Track> to List<Cluster>
SimpleChargedParticleMaker extends Driver

Match List<Track> to MIP-like clusters, optionally working back to parent cluster
MIPChargedParticleMaker extends Driver

Calibration again

I'm thinking of an interface for calibration to feed to SimpleNeutralParticleMaker that looks like:

```
public interface Calibration {
    public double correctedEnergy(Cluster);
}
```

with default implementation based on Cluster.getEnergy();

But I remember this was controversial last time...