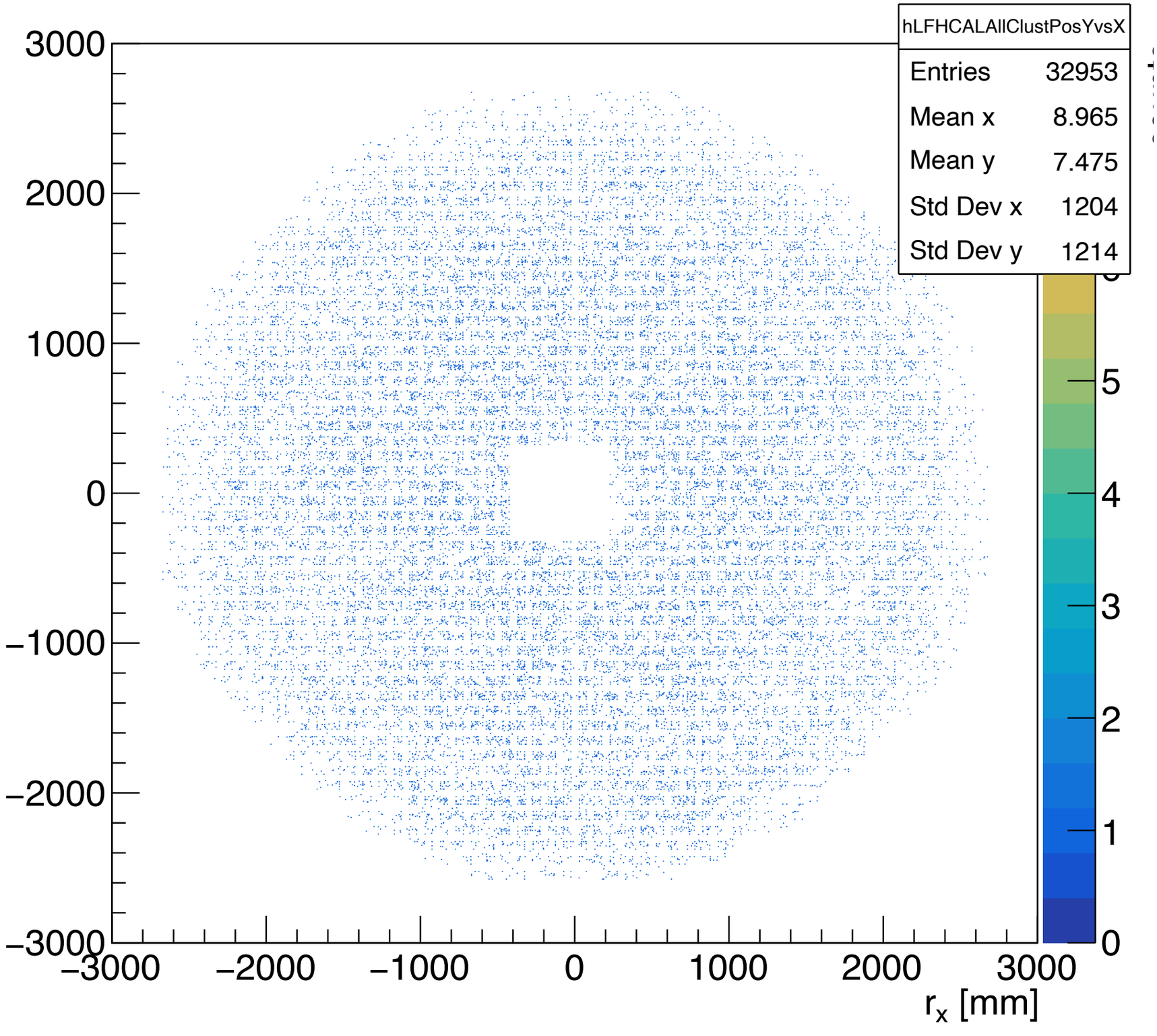
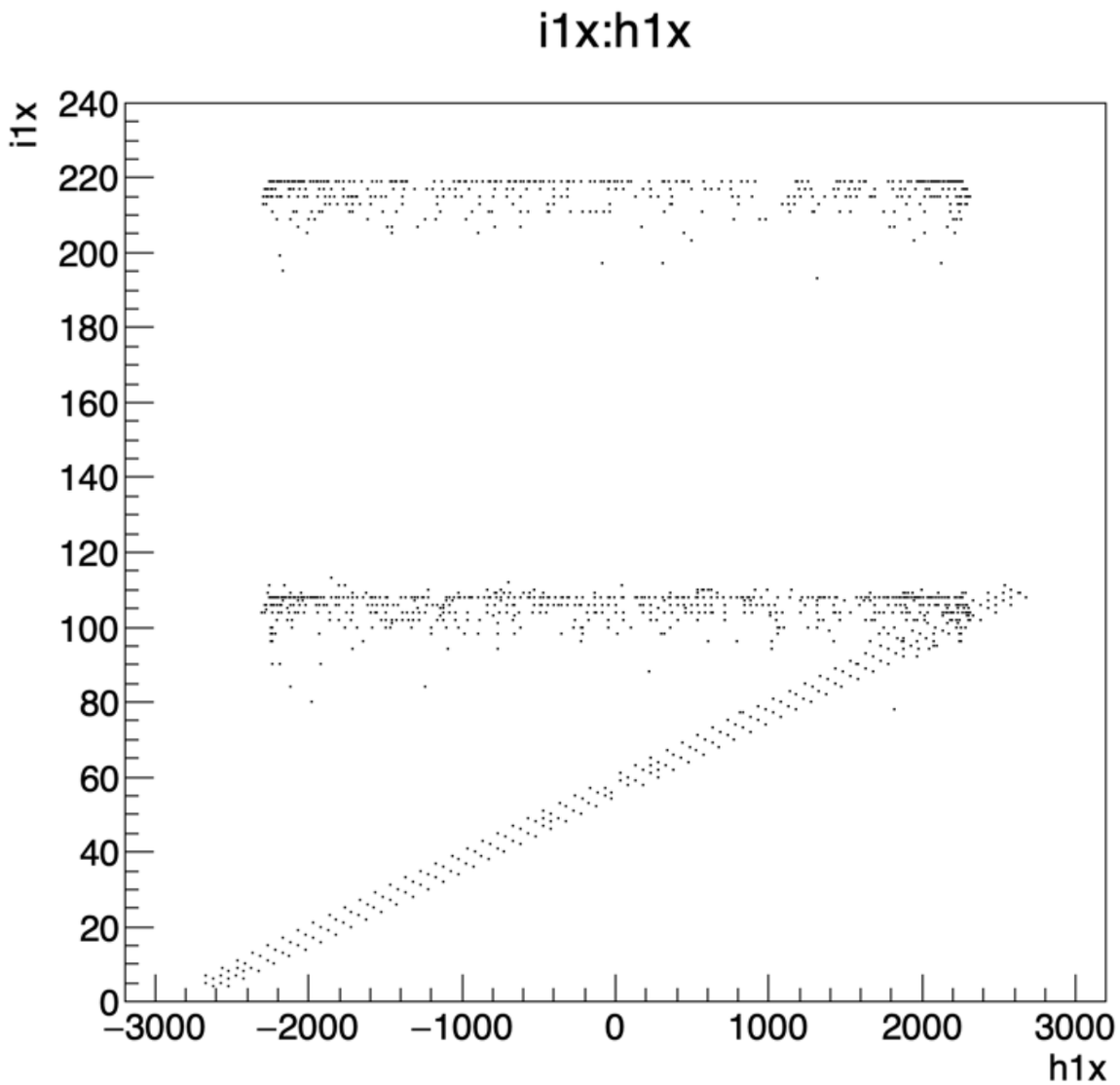
LFHCAL clustering update

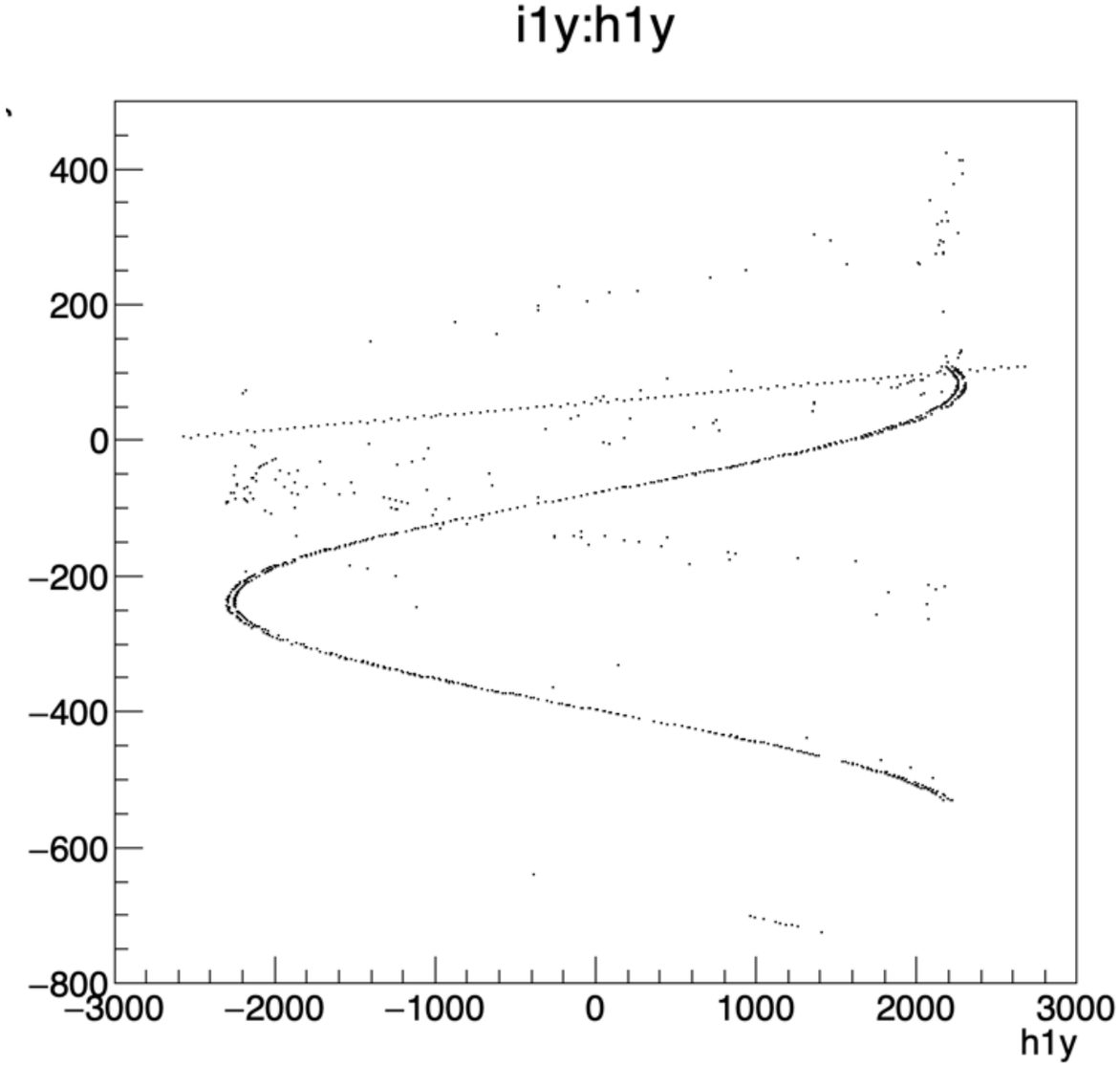
Peter Steinberg, BNL, 5 June 2024, LFHCAL weekly



Finally managed to run down this issue

Clustering bug





Cell coordinate (y) vs. global (x)

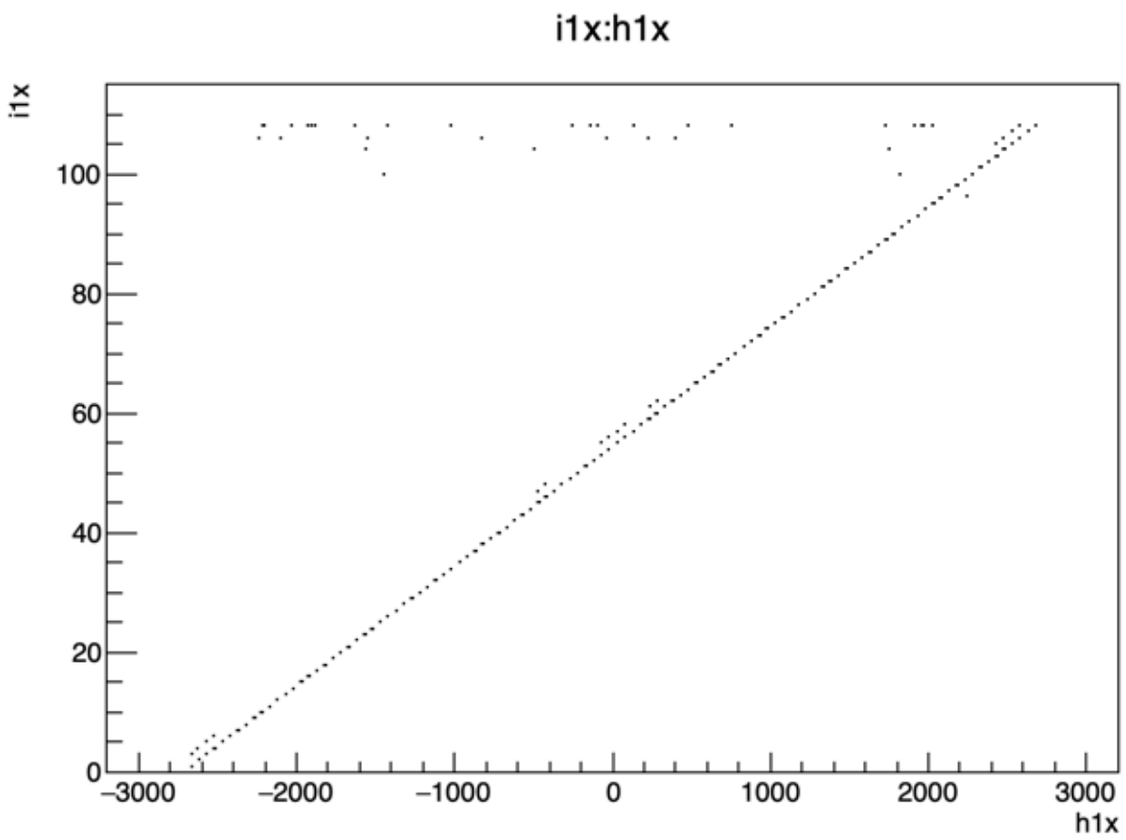
Problem appears to be that the combined coordinate has the wrong sign when combining with moduleID

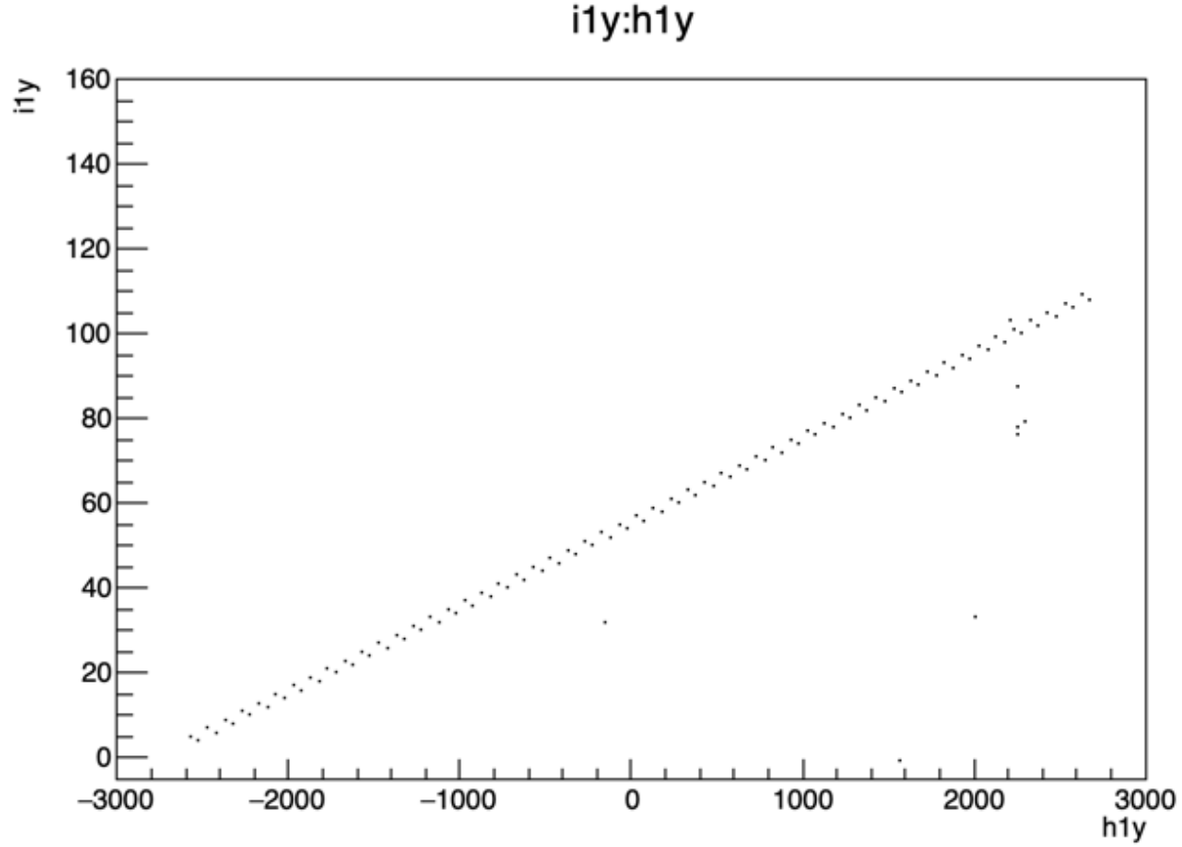
9 src/detectors/FHCAL/FHCAL.cc

```
@@ -186,10 +186,11 @@ extern "C" {
186  186          // Magic constants:
187  187          //  54 - number of modules in a row/column
188  188          //  2  - number of towers in a module
189       -       std::string cellIdx_1  = "(54*2-moduleIDx_1*2+towerx_1)";
190       -       std::string cellIdx_2  = "(54*2-moduleIDx_2*2+towerx_2)";
191       -       std::string cellIdy_1  = "(54*2-moduleIDy_1*2+towery_1)";
192       -       std::string cellIdy_2  = "(54*2-moduleIDy_2*2+towery_2)";
          189  +  // sign for towerx and towery are *negative* to maintain linearity with global X and Y
          190  +  std::string cellIdx_1  = "(54*2-moduleIDx_1*2-towerx_1)";
          191  +  std::string cellIdx_2  = "(54*2-moduleIDx_2*2-towerx_2)";
          192  +  std::string cellIdy_1  = "(54*2-moduleIDy_1*2-towery_1)";
          193  +  std::string cellIdy_2  = "(54*2-moduleIDy_2*2-towery_2)";
193  194          std::string cellIdz_1  = "rlayerz_1";
194  195          std::string cellIdz_2  = "rlayerz_2";
195  196          std::string deltaX     = Form("abs(%s-%s)", cellIdx_2.data(), cellIdx_1.data());
```

PR 1493 merged yesterday

Clustering bug



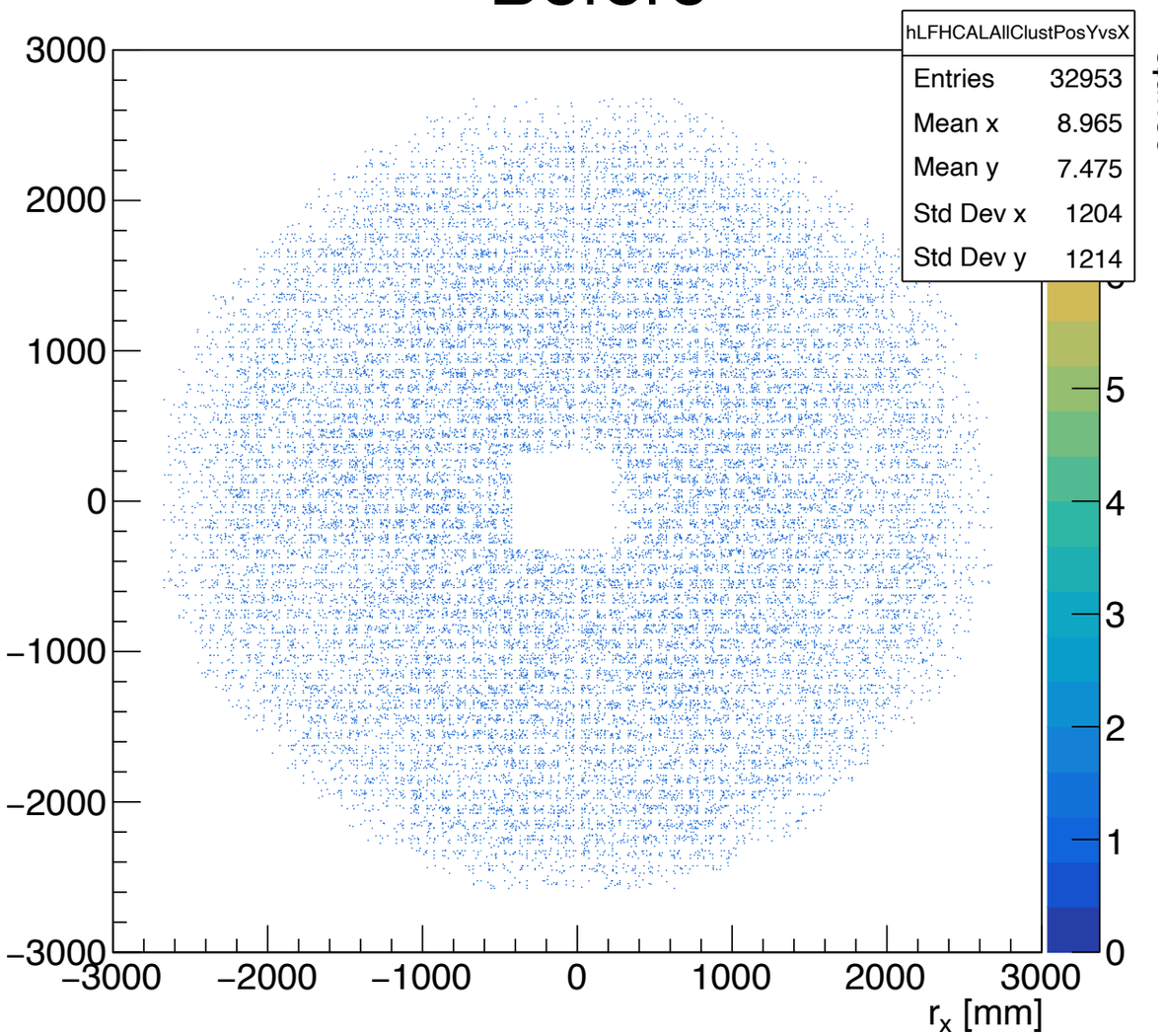


Cell coordinate (y) vs. global (x)

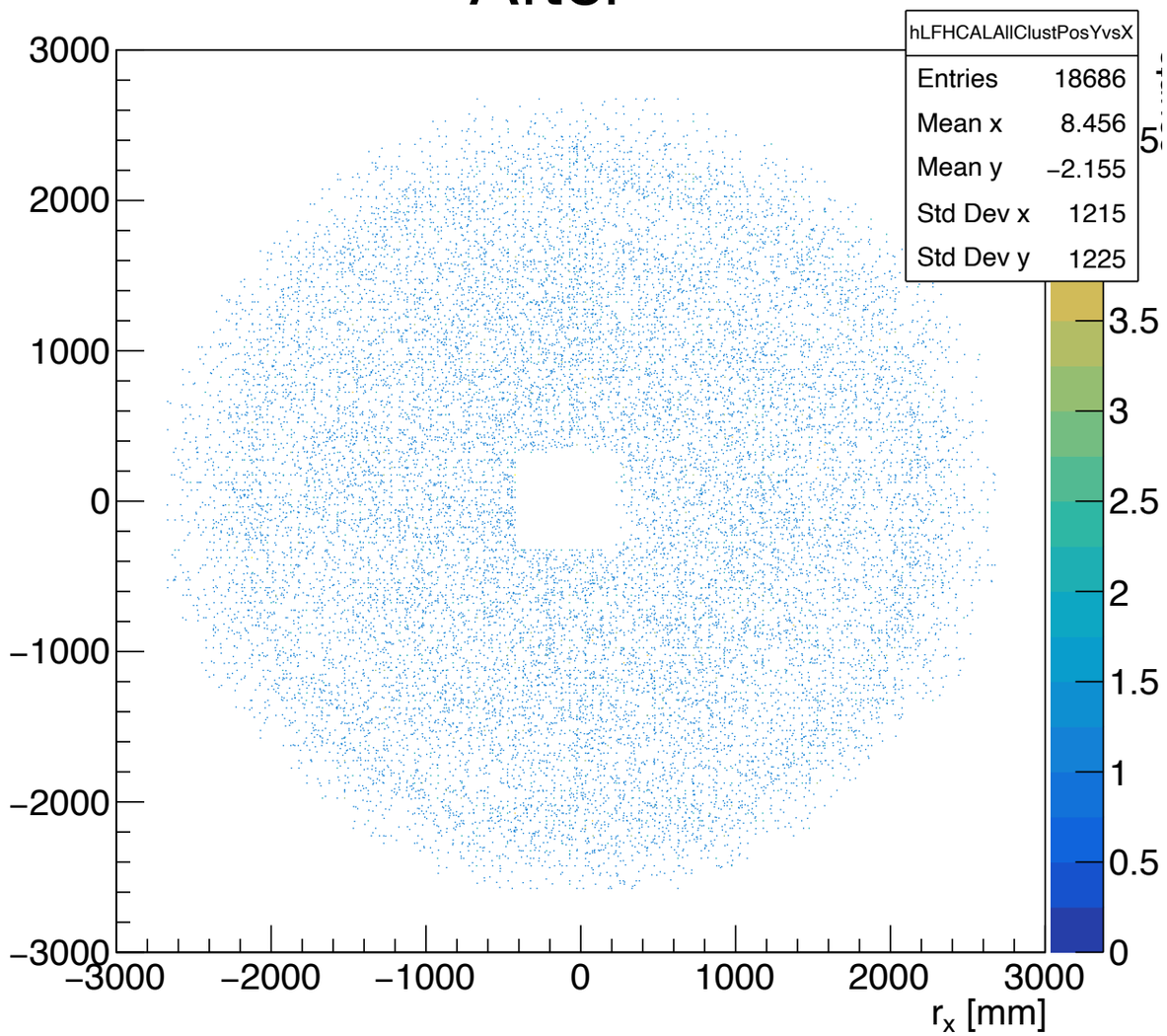
Less garbage and straighter in x, but still seeing “zig zag” in y: investigating further

Result

Before



After



No more “local reversal” in x, so fewer split clusters and no gaps, but persistence of mismatch in local and global y needs a fix