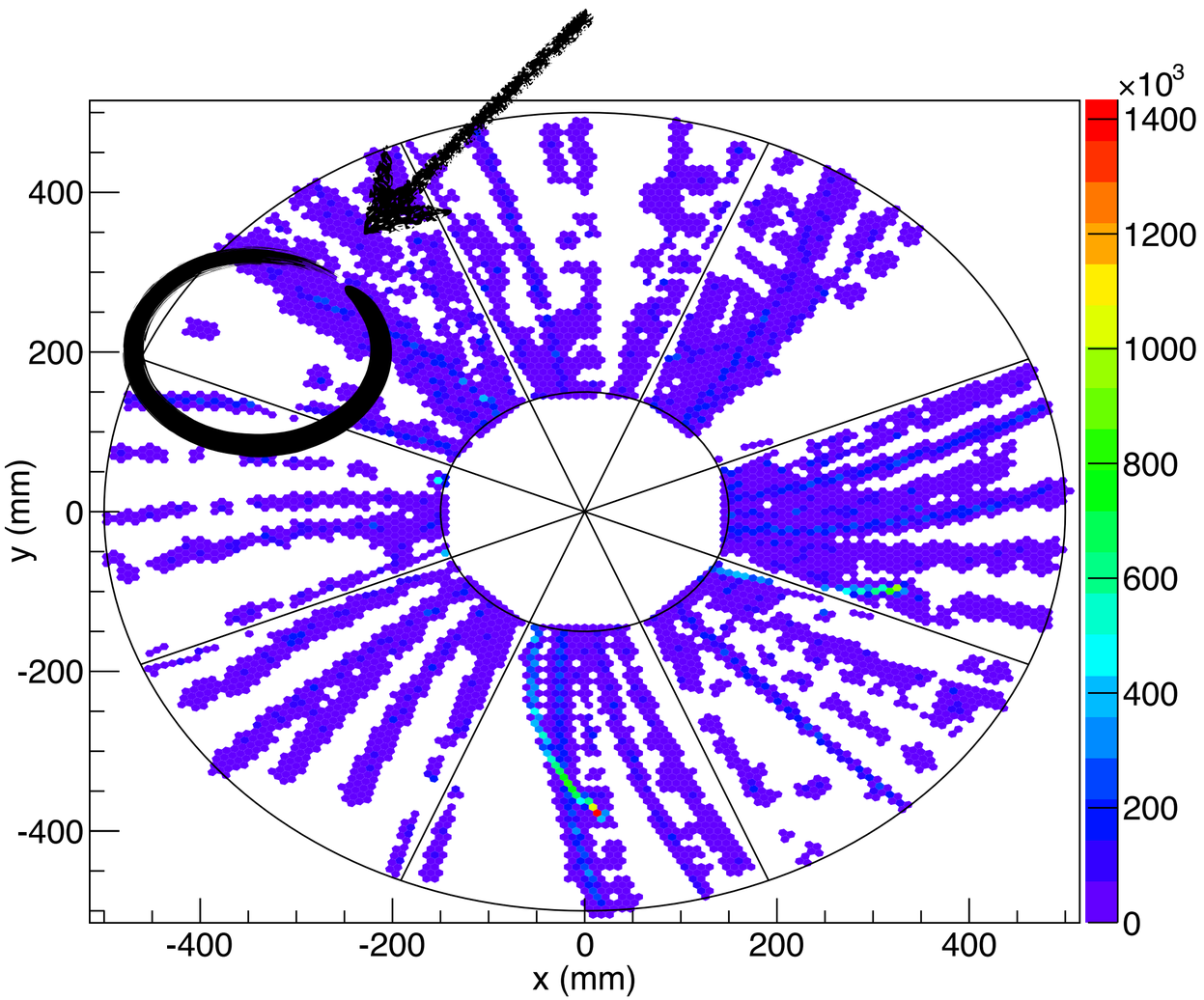
TPC

2013 06 21

Last results

- Tracks were **<u>discontinuous</u>!**



When changing gas for digitization, I didn't change the gas inside the TPC chamber(Geant4 simulation).

TPC Chamber Gas

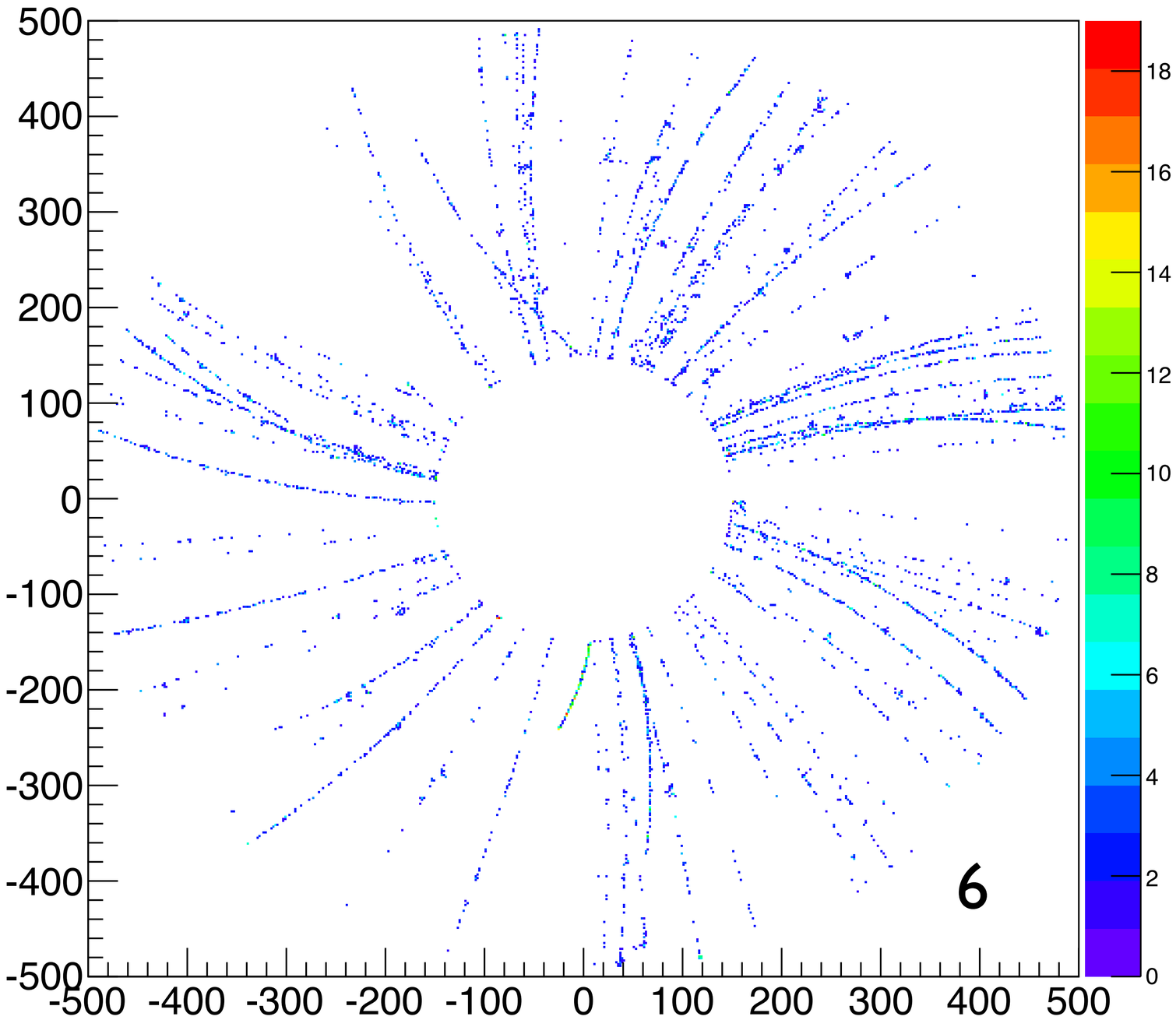
- LAMPSDetectorConstruction.cc :

```
G4VSolid* TPC_Solid_IN = new G4Tubs("TPC Inside", RIN_TPC_IN, ROUT_TPC_IN, DZ_TPC, 0., DPHI_TPC_IN);
G4LogicalVolume* TPC_LV_IN = new G4LogicalVolume(TPC_Solid_IN, C5, "TPC_LV_IN");
G4PVPlacement* TPC_PV_IN = new G4PVPlacement(0, G4ThreeVector(0, 0, DZ_TPC_OFFSET), "TPC_PV_IN", TPC_LV_IN, expHall, FALSE, 0);
G4VisAttributes* TPC_VisAtt_IN= new G4VisAttributes(true, G4Colour(1.,1.,0.));
TPC_LV_IN -> SetVisAttributes(TPC_VisAtt_IN);
```

- Gas inside the TPC chamber was not changing.
- Past results : only dispersion and gain properties of gas changed
- Gas was **Ne(100%)** .

Conditions

- IQMD
- 250 MeV
- Soft model
- No digitization ➜ Just raw data
- TPC plane is plotted



Ne

Can hardly see continuous lines.

Particles are rotating Clockwise!
(Corrections in B-field)

Ar and CO_2 mixtures

C10 ➜ 90% Ar + 10% CO_2

C20 ➜ 80% Ar + 20% CO_2

C5 ➜ 95% Ar + 5% CO_2

LAMPSDetectorConstruction.cc

- CO_2 and C10 gases were not even defined!

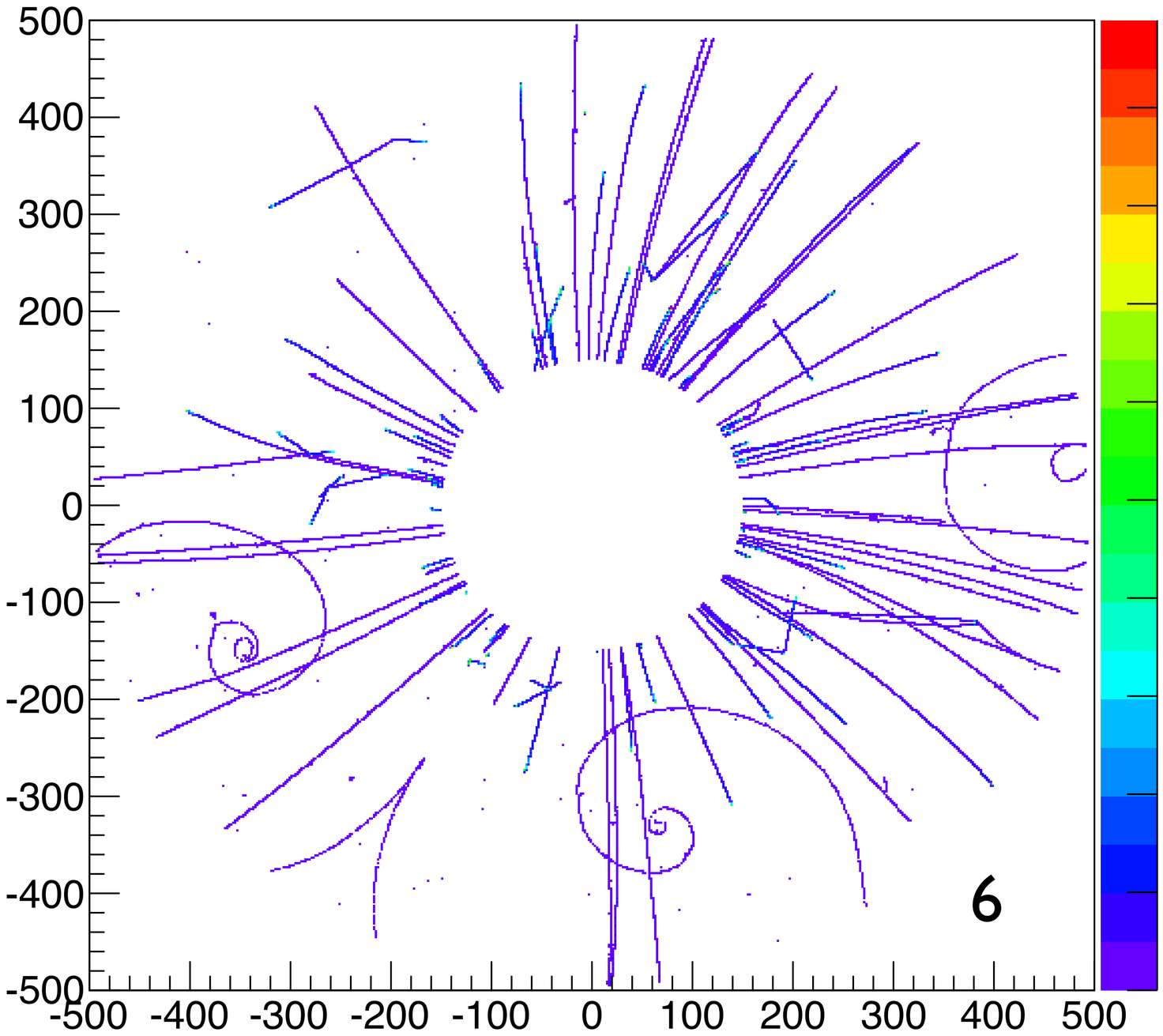
```
// CarbonDioxide (CO2) - JungWoo
const G4double denCarbonDioxide = 1.839 *g/cm3; //at 20'C, wolframalpha.com
G4Material* CarbonDioxide = new G4Material(name="CarbonDioxide", denCarbonDioxide,
                                           nel=2, kStateGas, expTemp);
CarbonDioxide -> AddElement(elC, 1);
CarbonDioxide -> AddElement(elO, 2);

// C5 gas : Ar(90%) + CO2(10%) mixture - JungWoo
density = 0.95*denAr+0.05*denCarbonDioxide;
G4Material* C5 = new G4Material(name="C5", density, nel=2, kStateGas, expTemp);
C5 -> AddMaterial(Ar, massfraction = 0.95*denAr/density);
C5 -> AddMaterial(CarbonDioxide, massfraction = 0.05*denCarbonDioxide/density);

// C10 gas : Ar(90%) + CO2(10%) mixture - JungWoo
density = 0.9*denAr+0.1*denCarbonDioxide;
G4Material* C10 = new G4Material(name="C10", density, nel=2, kStateGas, expTemp);
C10 -> AddMaterial(Ar, massfraction = 0.9*denAr/density);
C10 -> AddMaterial(CarbonDioxide, massfraction = 0.1*denCarbonDioxide/density);

// C20 gas : Ar(90%) + CO2(10%) mixture - JungWoo
density = 0.8*denAr+0.2*denCarbonDioxide;
G4Material* C20 = new G4Material(name="C20", density, nel=2, kStateGas, expTemp);
C20 -> AddMaterial(Ar, massfraction = 0.8*denAr/density);
C20 -> AddMaterial(CarbonDioxide, massfraction = 0.2*denCarbonDioxide/density);
```

C10

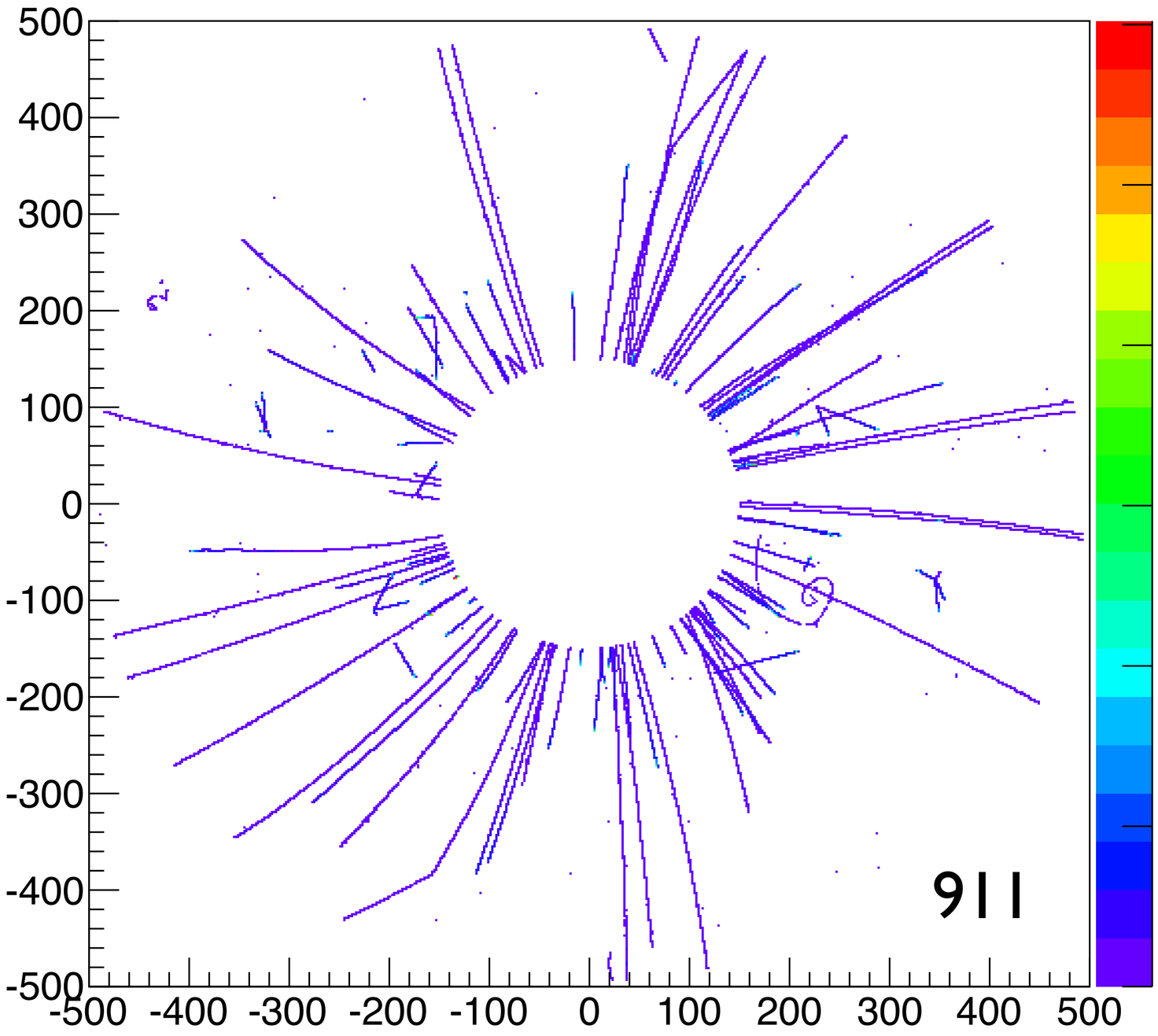


Continuous!

But

Too interactive!

C10

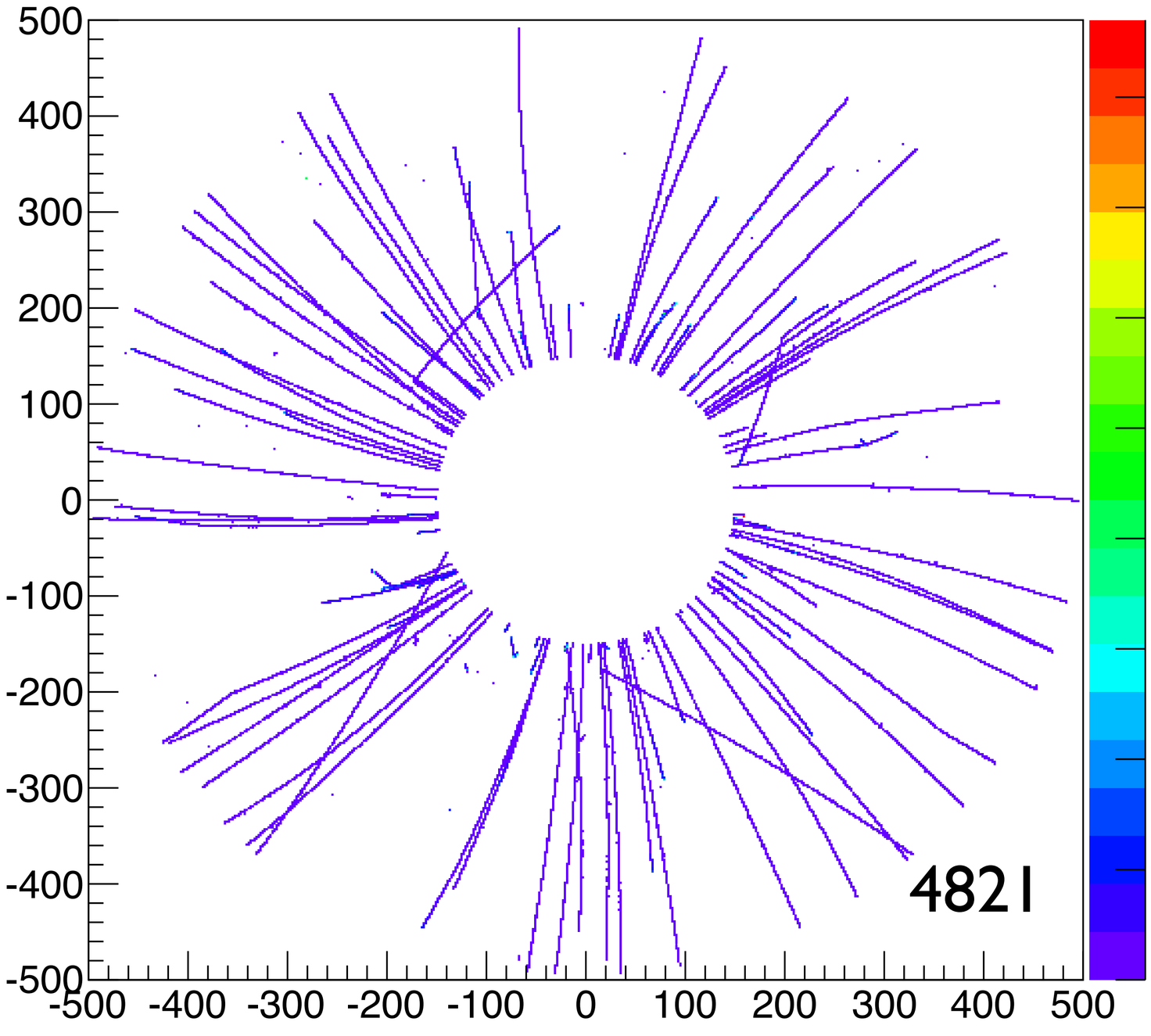


Continuous!

But

Too interactive!

C10

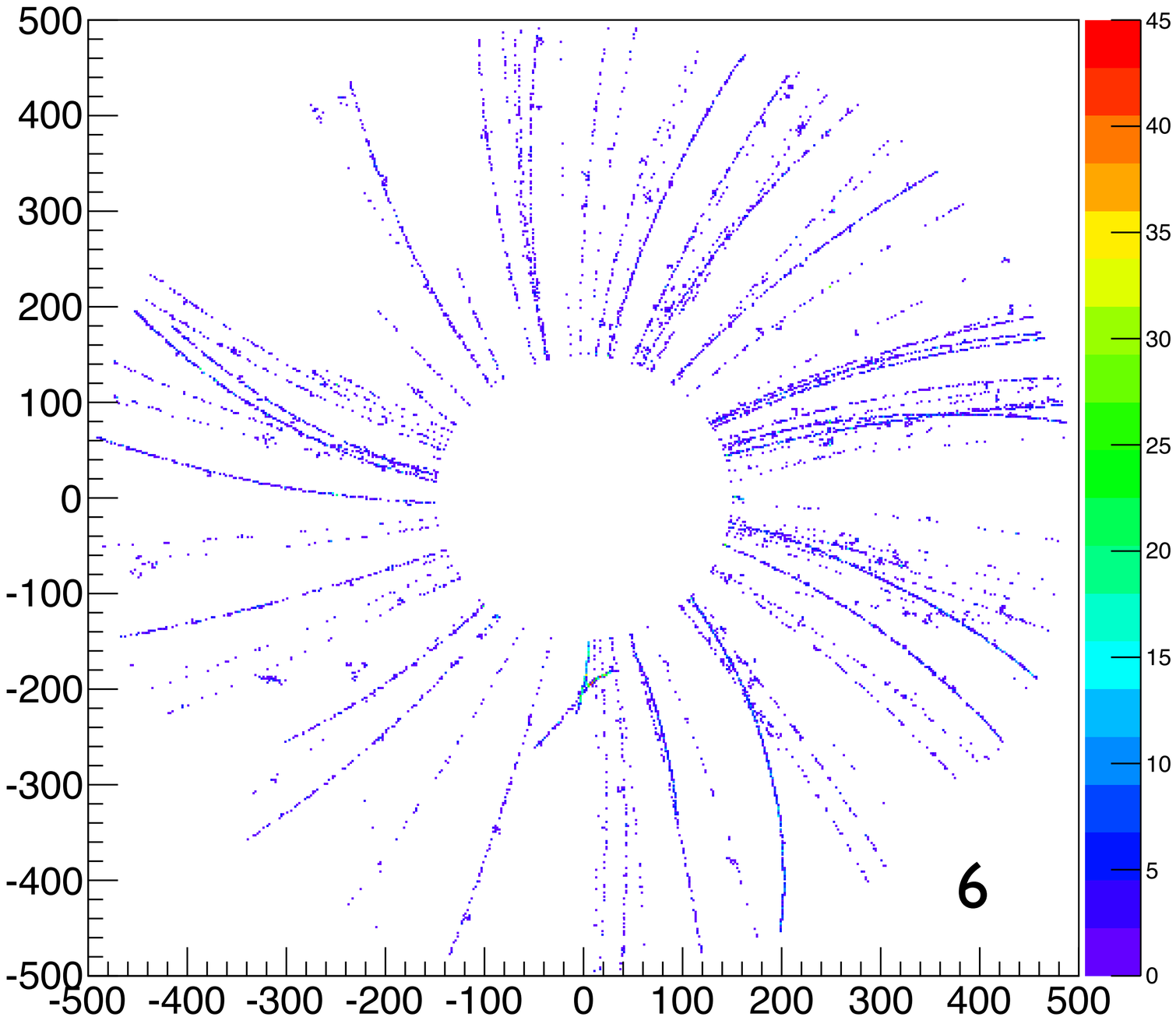


Continuous!

But

Too interactive!

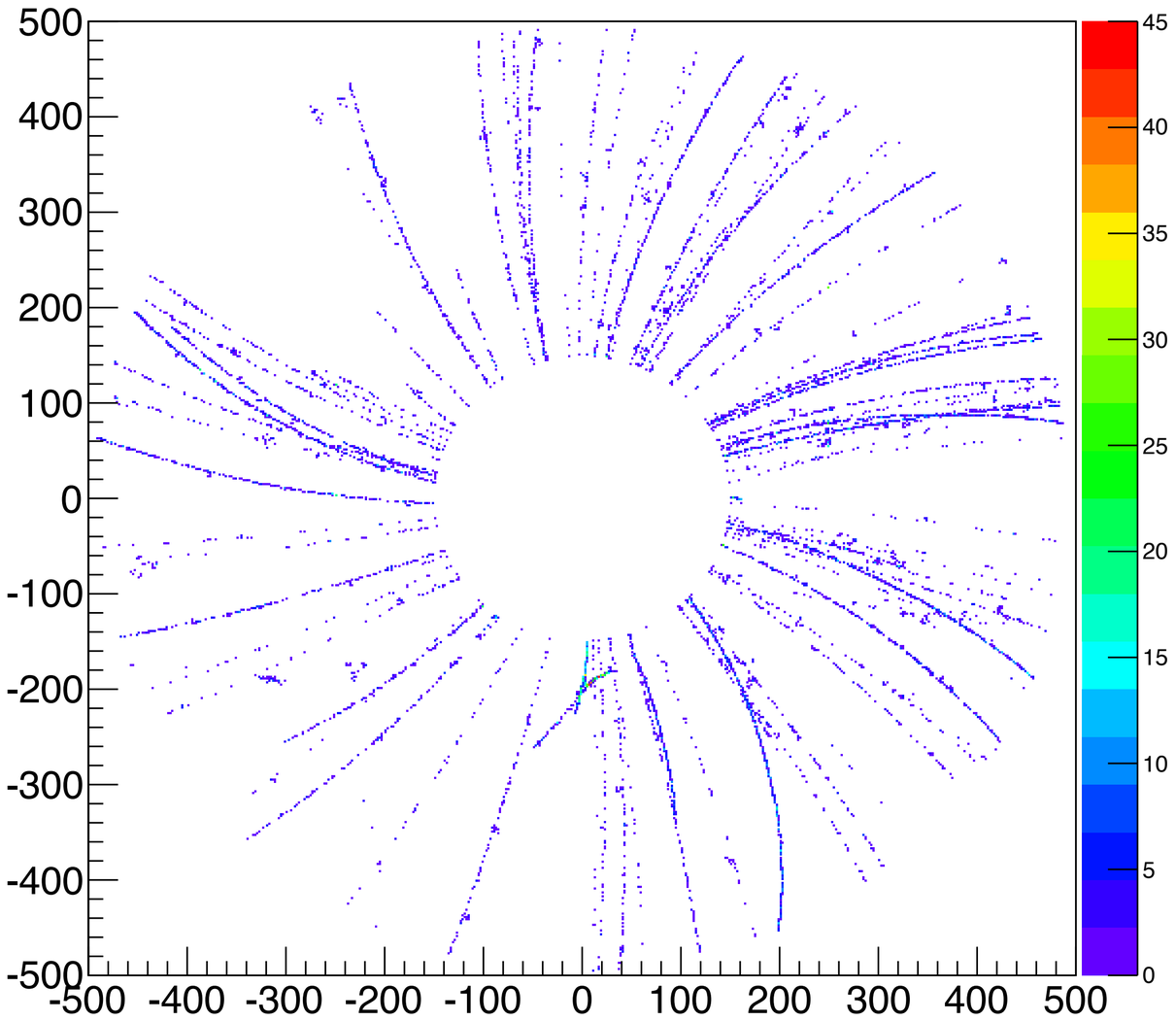
P10

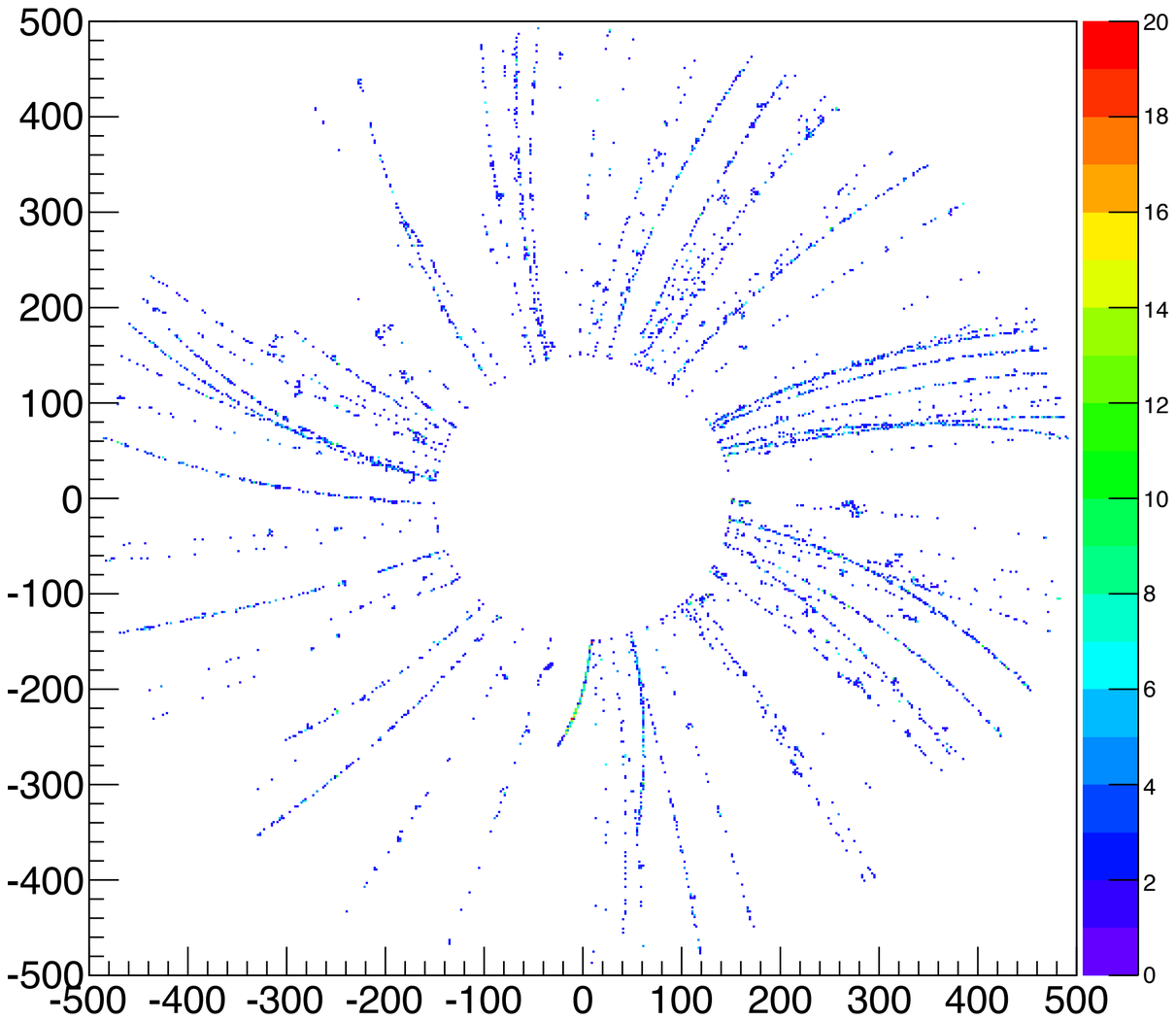


Better than “Ne”

P10

Ne


500
400
300
200
100
0
-100
-200
-300
-400
-500
-500 -400 -300 -200 -100 0 100 200 300 400 500
45
40
35
30
25
20
15
10
5
0


500
400
300
200
100
0
-100
-200
-300
-400
-500
-500 -400 -300 -200 -100 0 100 200 300 400 500
20
18
16
14
12
10
8
6
4
2
0

?

Density of CO_2 is three times that of CH_4

A01PhysicsList.cc

```
  // default cut value  (1.0mm)
  defaultCutValue = 1.0*um;
  SetVerboseLevel(1);

  // General Physics ( Create ALL Particle and apply Decay )
  RegisterPhysics( new A01GeneralPhysics("general") );

  // EM Physics ( Apply related Processes to gamma and e-/+)
  RegisterPhysics( new A01EMPhysics("standard EM"));

  // Muon Physics ( Apply related processes to mu and tau
  RegisterPhysics(  new A01MuonPhysics("muon"));

   // Hadron Physics ( Apply related processes to hadrons )
  RegisterPhysics(  new A01HadronPhysics("hadron"));
// We do not use hadronic lists since v7.
  //RegisterPhysics(  new HadronPhysicsQGSP_BERT("hadron"));
  //RegisterPhysics(  new HadronPhysicsQGSP_BIC("hadron"));

  // Ion Physics ( Apply related processes to ions )
  RegisterPhysics( new A01IonPhysics("ion"));

}

A01PhysicsList::~A01PhysicsList()
{
}

void A01PhysicsList::SetCuts()
{
  //  " G4VUserPhysicsList::SetCutsWithDefault" method sets
  //   the default cut value for all particle types
//  SetCutsWithDefault();
//  G4ProductionCutsTable::GetProductionCutsTable() -> SetEnergyRange(10.*eV, 100.*GeV);

  SetCutValue( 10.0*micrometer, "e-");
  SetCutValue( 1.0*micrometer, "e+");
  SetCutValue( 1.0*micrometer, "gamma");
}
```

“defaultCutValue”

“SetCutValue”

“CutValue”

A01PhysicsList.cc

2.4.2. Range Cuts

To avoid infrared divergence, some electromagnetic processes require a threshold below which no secondary will be generated. Because of this requirement, gammas, electrons and positrons require production thresholds which the user should define. This threshold should be defined as a distance, or range cut-off, which is internally converted to an energy for individual materials. The range threshold should be defined in the initialization phase using the `SetCuts()` method of `G4VUserPhysicsList`. Section 5.5 discusses threshold and tracking cuts in detail.

2.4.2.1. Setting the cuts

Production threshold values should be defined in `SetCuts()` which is a pure virtual method of the `G4VUserPhysicsList` class. Construction of particles, materials, and processes should precede the invocation of `SetCuts()`. `G4RunManager` takes care of this sequence in usual applications.

This range cut value is converted threshold energies for each material and for each particle type (i.e. electron, positron and gamma) so that the particle with threshold energy stops (or is absorbed) after traveling the range cut distance. In addition, from the 9.3 release ,this range cut value is applied to the proton as production thresholds of nuclei for hadron elastic processes. In this case, the range cut value does not means the distance of traveling. Threshold energies are calculated by a simple formula from the cut in range.

Note that the upper limit of the threshold energy is defined as 10 GeV. If you want to set higher threshold energy, you can change the limit by using "/cuts/setMaxCutEnergy" command before setting the range cut.

The idea of a "unique cut value in range" is one of the important features of Geant4 and is used to handle cut values in a coherent manner. For most applications, users need to determine only one cut value in range, and apply this value to gammas, electrons and positrons alike. (and proton too)

In such case, the `SetCutsWithDefault()` method may be used. It is provided by the `G4VuserPhysicsList` base class, which has a `defaultCutValue` member as the default range cut-off value. `SetCutsWithDefault()` uses this value.

It is possible to set different range cut values for gammas, electrons and positrons, and also to set different range cut values for each geometrical region. In such cases however, one must be careful with physics outputs because Geant4 processes (especially energy loss) are designed to conform to the "unique cut value in range" scheme.

“Geant4 User’s Guide for Application Developer”

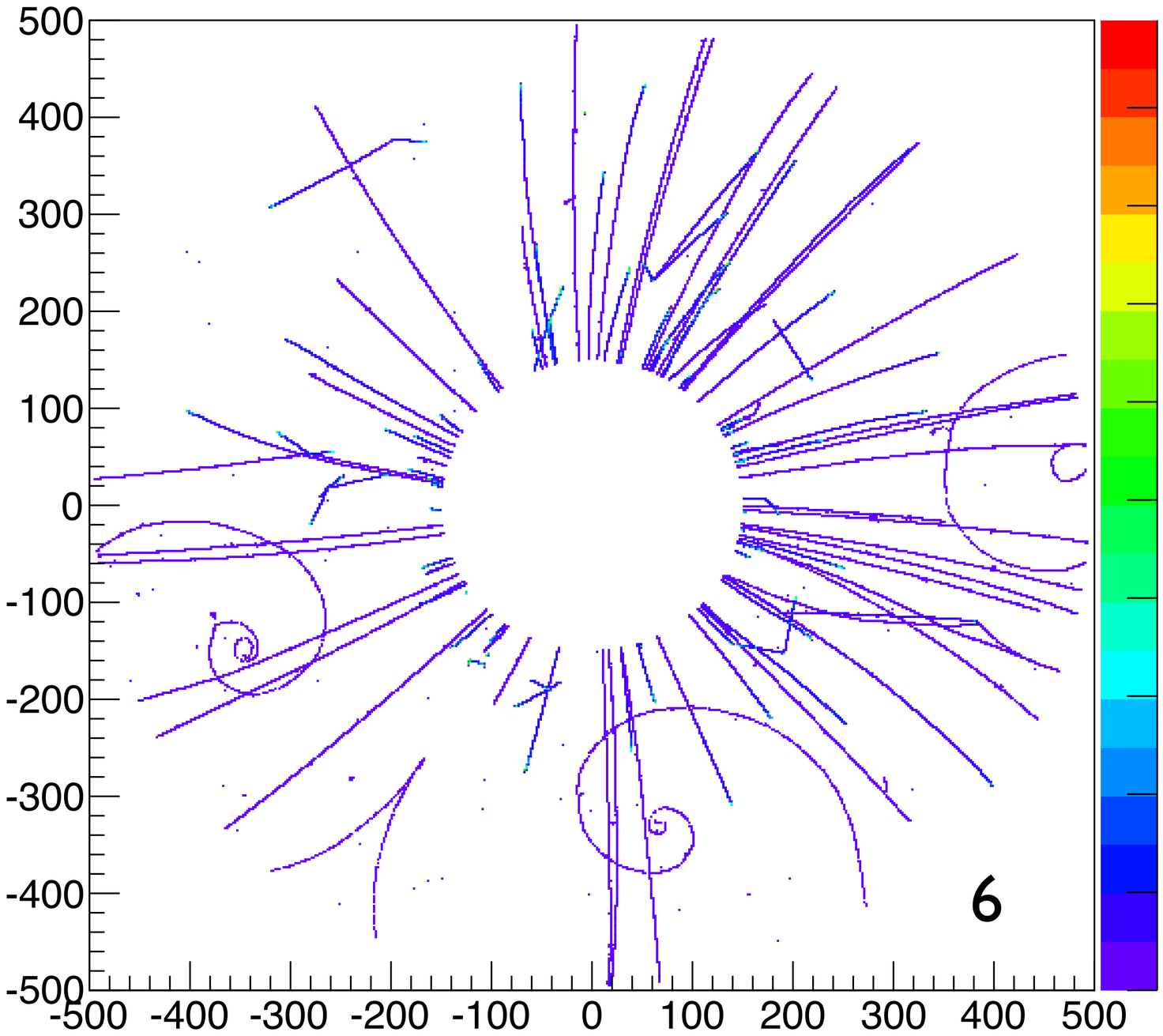
“Cuts”

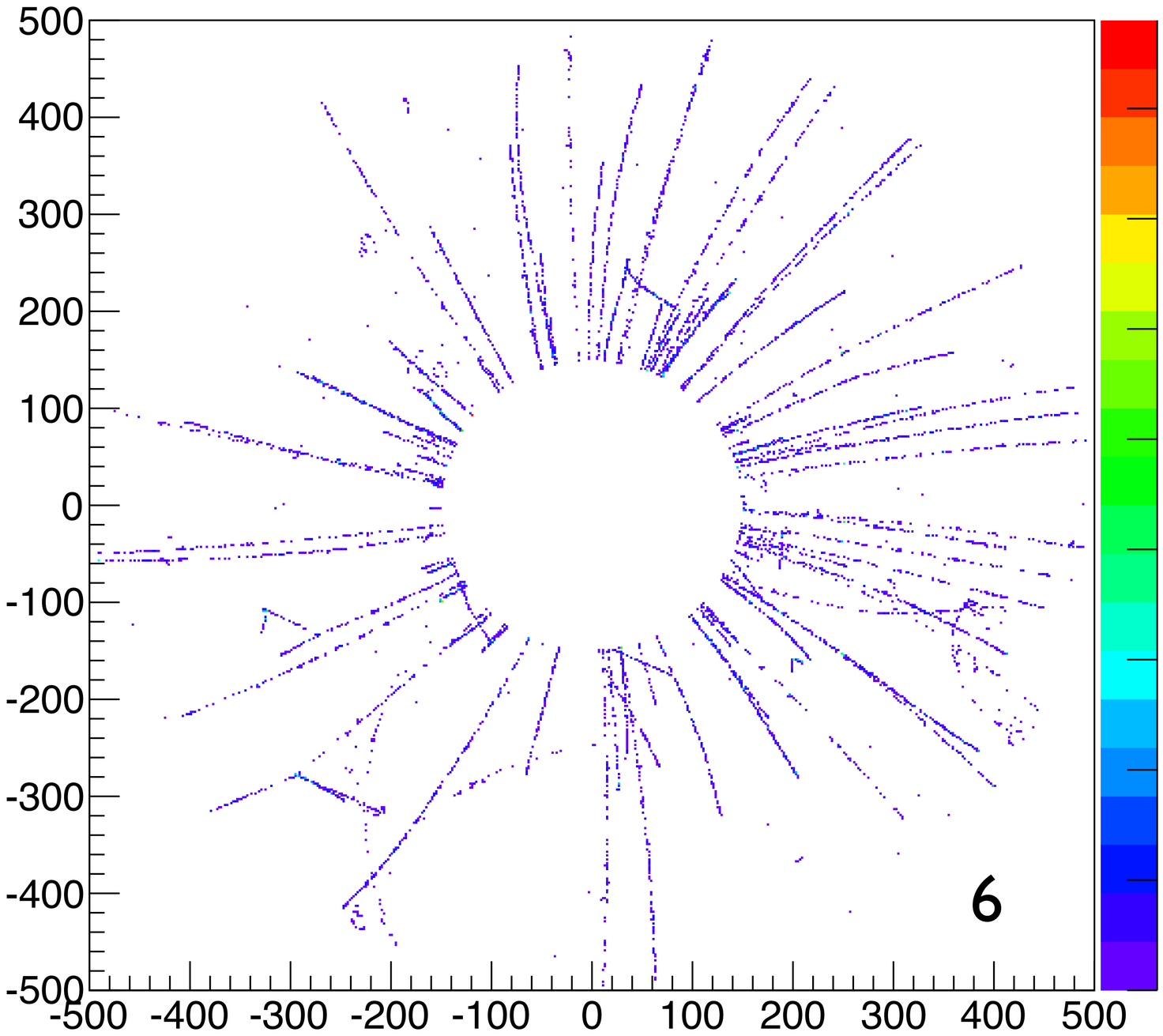
A01PhysicsList.cc

Default value	1 mm
LAMPS simulation	1 μm
Test	10 μm
Test	0.1 μm

C10

1 μm


500
400
300
200
100
0
-100
-200
-300
-400
-500
-500 -400 -300 -200 -100 0 100 200 300 400 500
6



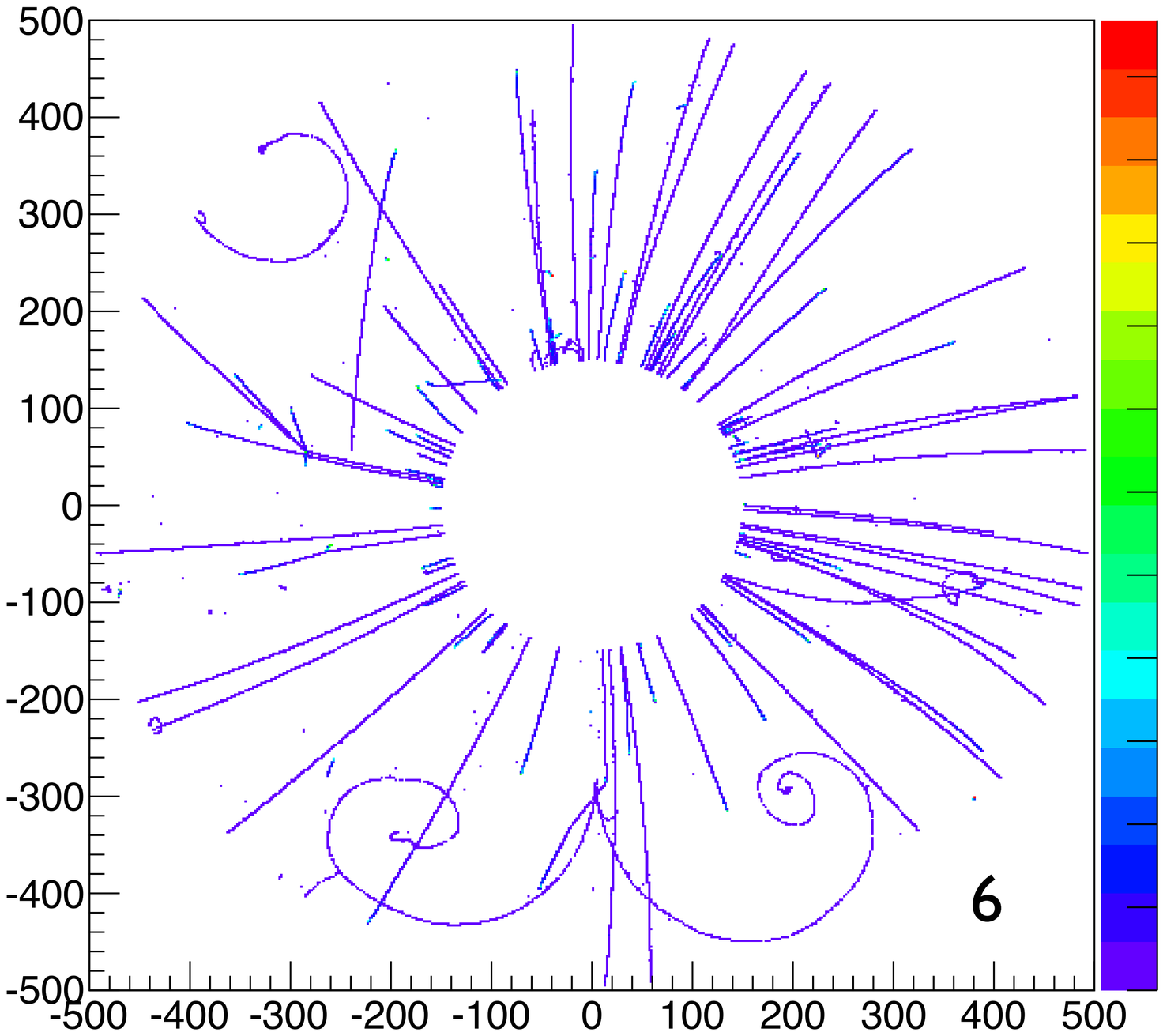
C10

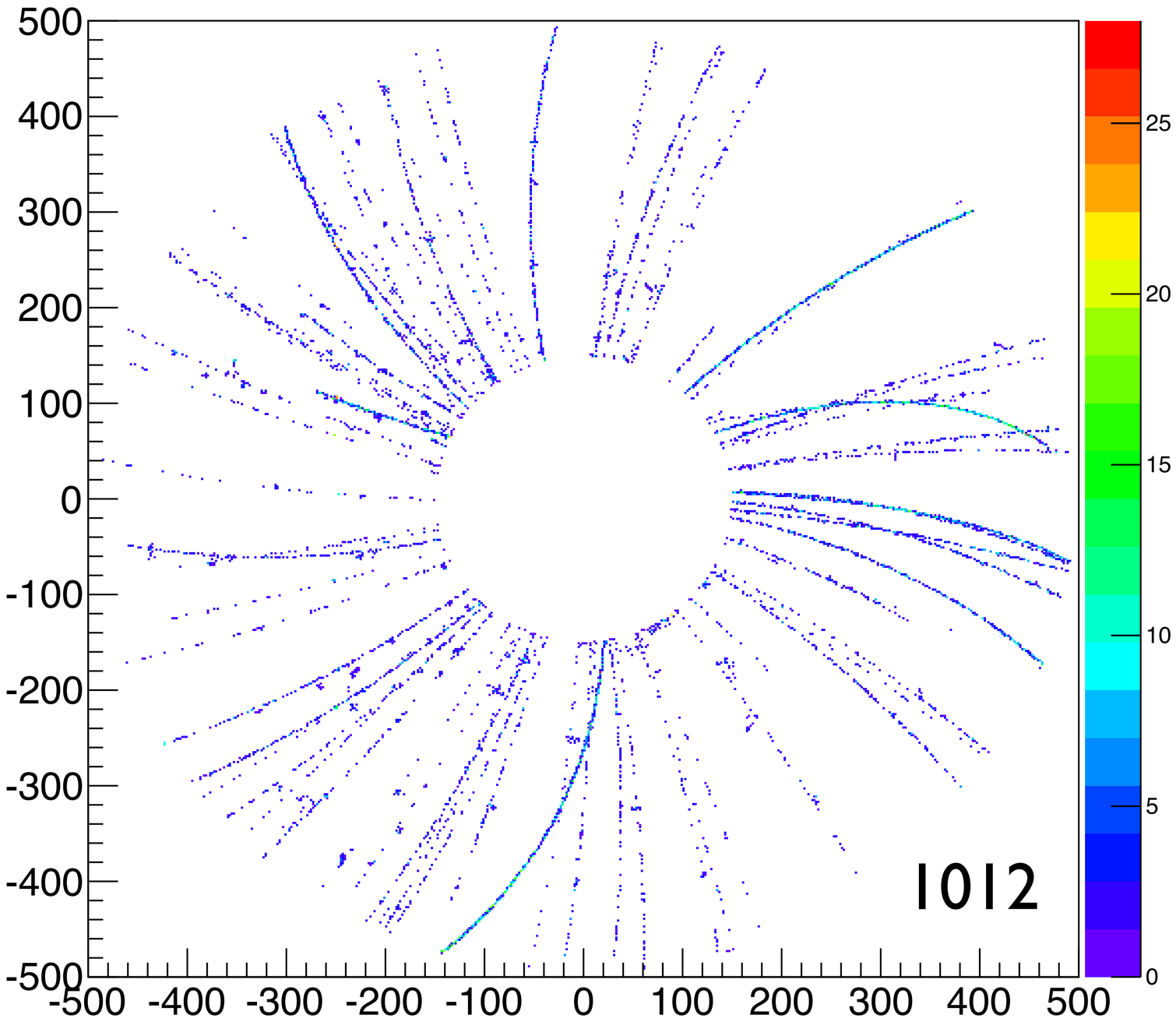
10 μm

Discontinuous!

C10

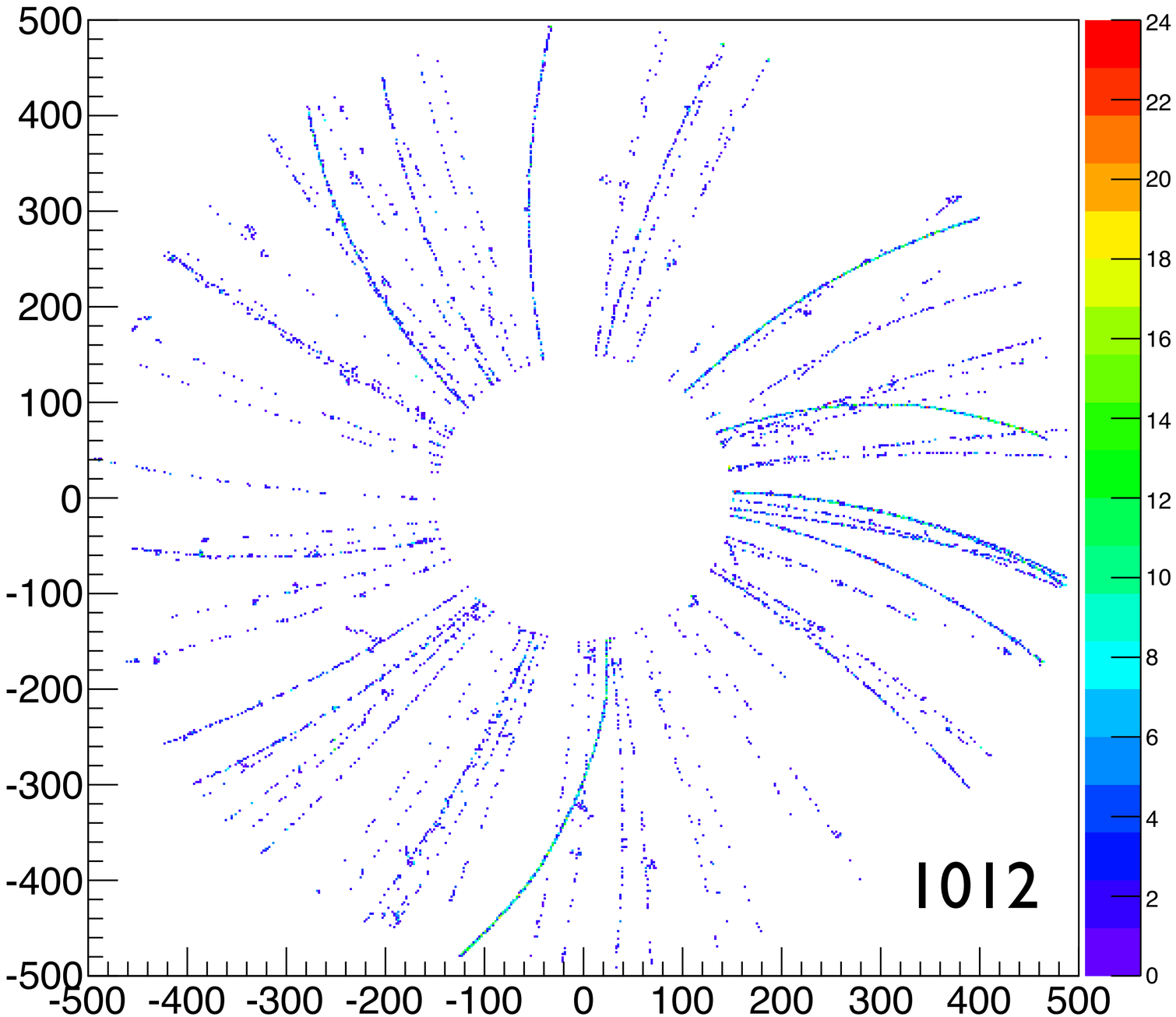
0.1 μm


500
400
300
200
100
0
-100
-200
-300
-400
-500
-500 -400 -300 -200 -100 0 100 200 300 400 500
6


500
400
300
200
100
0
-100
-200
-300
-400
-500
-500 -400 -300 -200 -100 0 100 200 300 400 500
25
20
15
10
5
0
1012

P10

1 μm



P10

0.1 μm

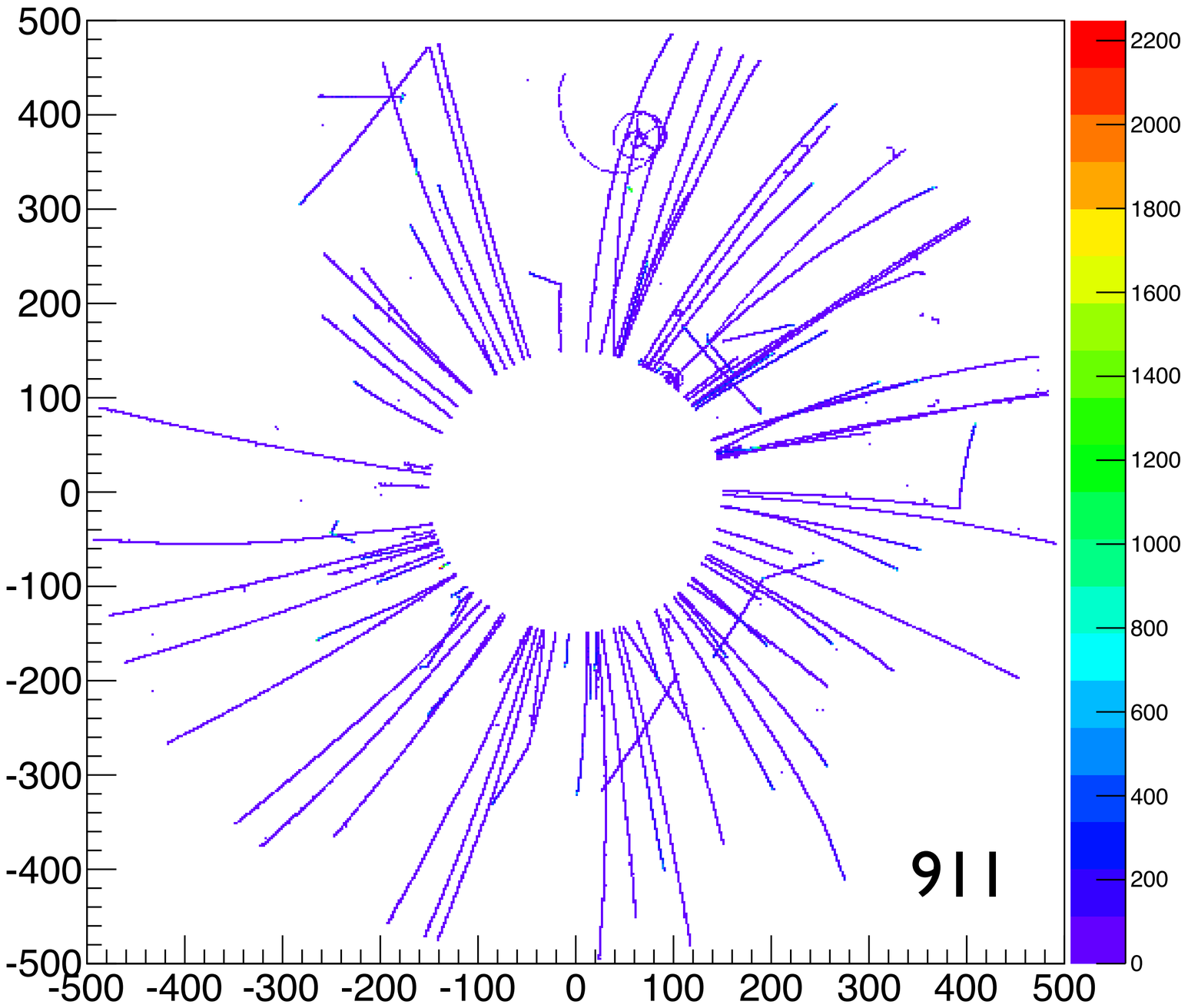
No change

defaultCutValue

1 μm is reasonable!

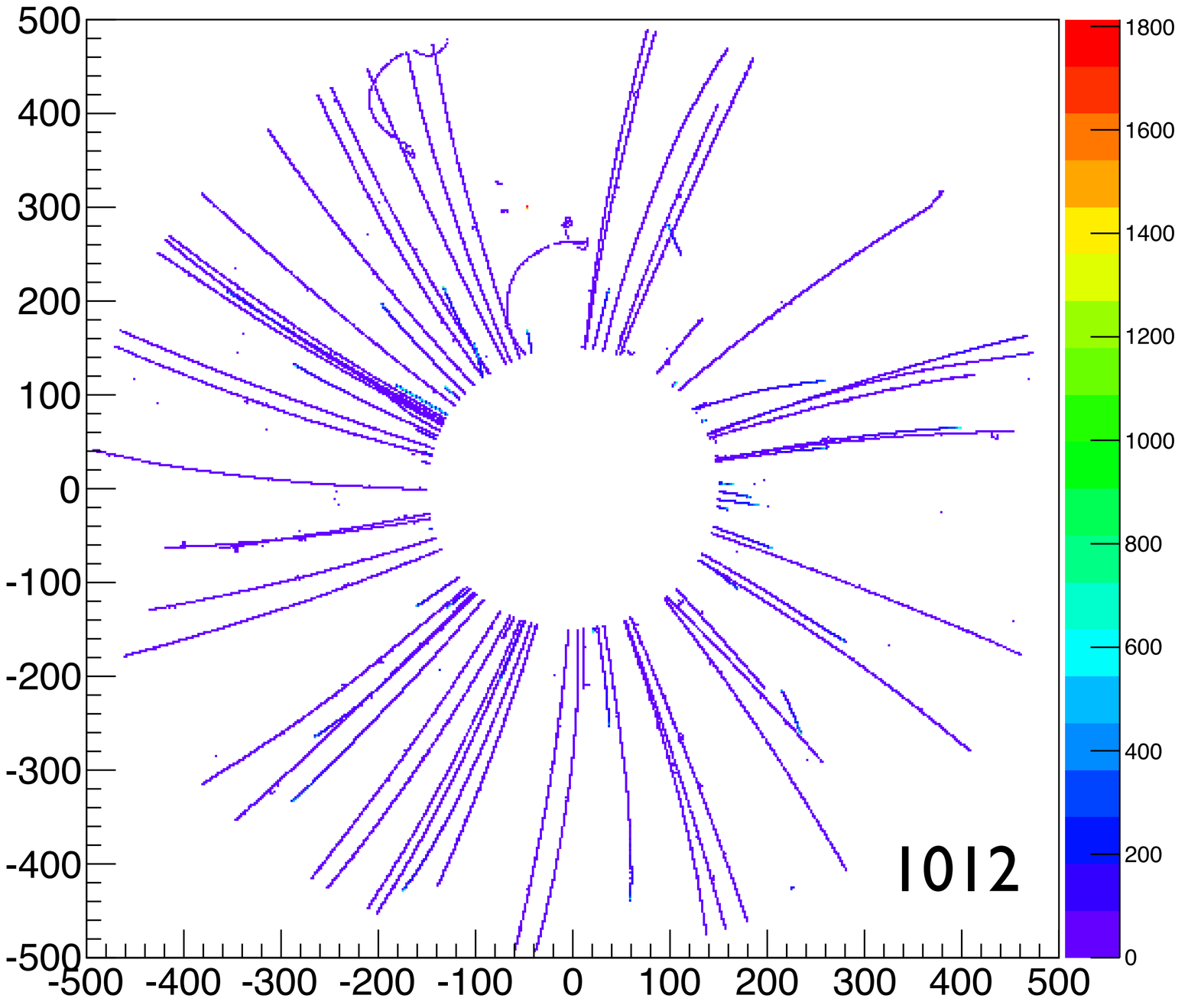
C5

1 µm


500
400
300
200
100
0
-100
-200
-300
-400
-500
-500 -400 -300 -200 -100 0 100 200 300 400 500
2200
2000
1800
1600
1400
1200
1000
800
600
400
200
0
911

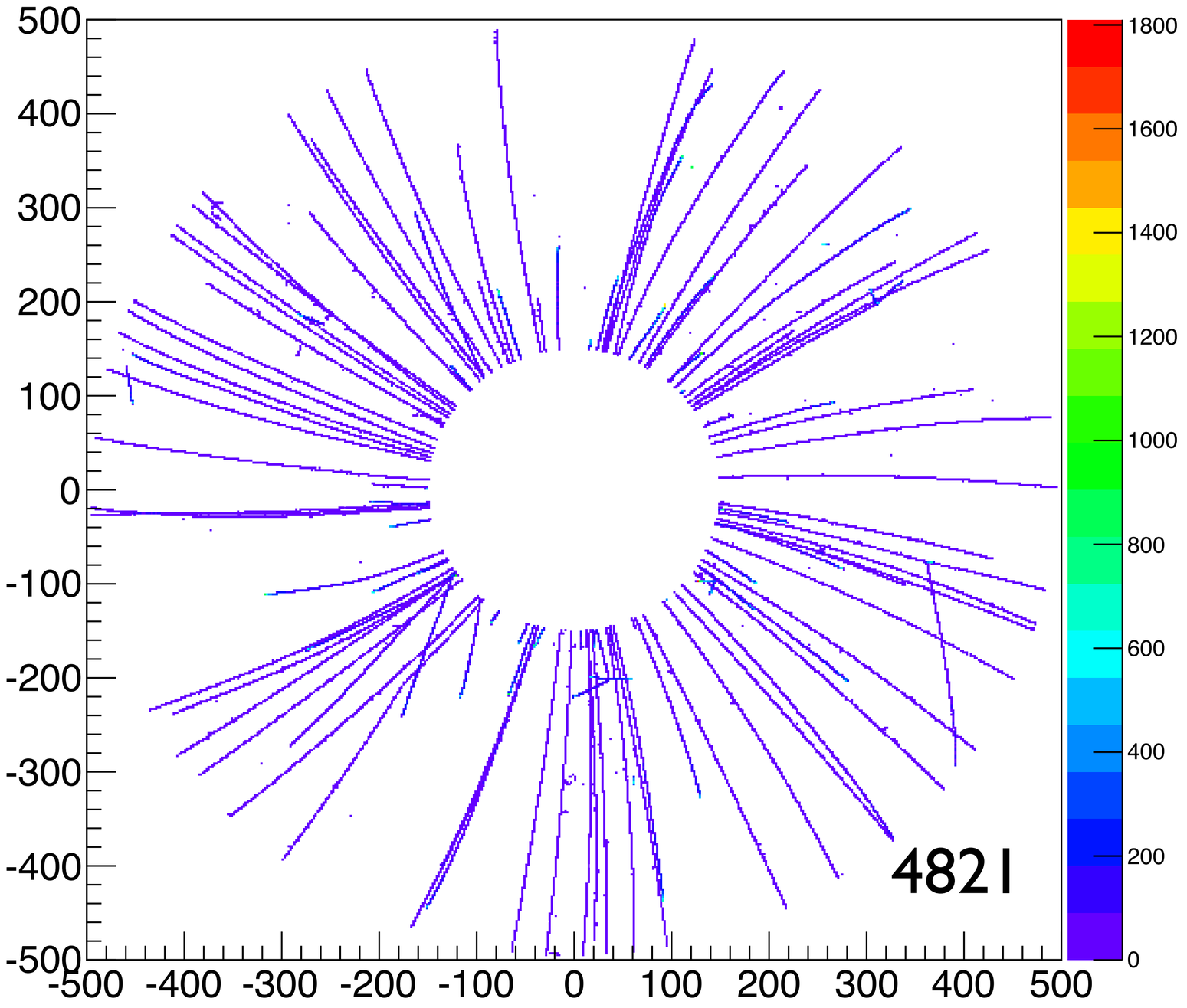
C5

1 µm


500
400
300
200
100
0
-100
-200
-300
-400
-500
-500 -400 -300 -200 -100 0 100 200 300 400 500
1012
1800
1600
1400
1200
1000
800
600
400
200
0

C5

1 µm


500
400
300
200
100
0
-100
-200
-300
-400
-500
-500 -400 -300 -200 -100 0 100 200 300 400 500
1800
1600
1400
1200
1000
800
600
400
200
0
4821

Summary

- Default cut value of 1 μm is fine.
- Need to study more about the gasses.
- Small errors were found in using Charles's law when defining density of the materials(LAMPSDetectorConstruction.cc).