

```

void makeHFTPlots(const Char_t *tag = ""){/"dEdx") {
    TString FileN(gDirectory->GetName());
    gInterpreter->ProcessLine(".L Chain.C");
    TChain *theChain = Chain();
    TString macro(".L ");
    macro += gSystem->Which(gROOT->GetMacroPath(),"T.C");
    macro += "+";
    gInterpreter->ProcessLine(macro);
    // gInterpreter->ProcessLine(".L T.C+");
    TT t(theChain);
// #if 1
    TString Out = gSystem->DirName(FileN); cout << Out;
    TString Dir = gSystem->DirName(Out);
    // Out.ReplaceAll(Dir,"");
    if (Out.BeginsWith("/")) Out = Out.Data()+1;
// #else
//     TString Out = gSystem->BaseName(FileN);
//     Out.ReplaceAll(".root","");
// #endif
    Out.ReplaceAll("/","_");
    Out += "Plots"; Out += tag;
    TString Tag(tag);
    if (Tag.Contains("G14G",TString::kIgnoreCase) ||
        Tag.Contains("G44G",TString::kIgnoreCase)) {
        t.SetuMinMax(2.5, 2.9);
        Out += "_u_2.5-2.9";
    }
    if (Tag.Contains("V5.0",TString::kIgnoreCase)) {
        t.SetVertexZCut(5.0); // cm from z = 0
    }
    if (Tag.Contains("D0.9",TString::kIgnoreCase)) {
        t.SetDipCut(0.9); // pT < DipCut*p
    }
    if (Tag.Contains("NoW",TString::kIgnoreCase)) {
        t.SetNoWafers();
    }
    if (Tag.Contains("LG",TString::kIgnoreCase)) {
        t.SetLaddersInGlobal(kTRUE);
    }
    if (Tag.Contains("SSD",TString::kIgnoreCase)) {
        t.SetSsd(kTRUE);
    }
    if (Tag.Contains("IST",TString::kIgnoreCase)) {
        t.SetIst(kTRUE);
    }
    if (Tag.Contains("east",TString::kIgnoreCase)) {
        if (Tag.Contains("fareast",TString::kIgnoreCase)) t.SetEastWest(3);
        else t.SetEastWest(1);
    } else {
        if (Tag.Contains("west",TString::kIgnoreCase)) {
            if (Tag.Contains("farwest",TString::kIgnoreCase)) t.SetEastWest(4);
            else t.SetEastWest(2);
        }
    }
    if (Tag.Contains("global",TString::kIgnoreCase)) {
        t.UseGlobal();
    }
    if (Tag.Contains("local",TString::kIgnoreCase)) {
        t.UseLocal();
    }
}

```

```
if (Tag.Contains("dEdx", TString::kIgnoreCase)) {
    t.SetdEdxCut(4.e-6, 40.);
}
// t.SetRCut(1.0);
t.SetRCut(0.5);
t.SetMinNoFitPoints(25);
Out += "NFP25";
Out += Form("rCut%3.1fcm", t.GetRCut());
Out += ".root";
cout << " ==> " << Out << endl;
t.SetOutFileName(Out);
t.Loop(0);
}
```