

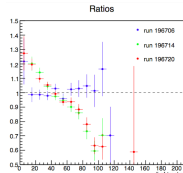
# QA et reprise de la distribution en $x_E$

Vauthier Astrid

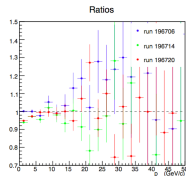
6 mai 2014

- Dernière réunion : mise en évidence de mauvais runs avec les distributions en  $p_T$  des hadrons et la multiplicité des traces
- Depuis : fit des déviations des ratios par rapport à 1
- Comparaison avec les listes de mauvais runs de Lucille et Marie
- Pour la suite on ne prendra pas les mauvais runs communs avec Marie

## 13f runs 6714 et 6720

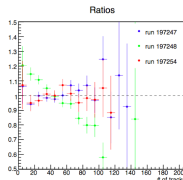


Multiplicité des traces

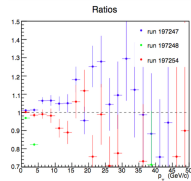


Distribution en  $p_T$  des hadrons

## 13f run 7248



Multiplicité des traces



Distribution en  $p_T$  des hadrons

Première distribution en  $x_E$  obtenue lors des premières semaines du stage  $\Rightarrow$  on ne pouvait rien en conclure.

Il fallait :

- Vérifier les données  $\rightarrow$  QA
- Vérifier l'UE (UE ou UE left + UE right)
- Prendre en compte le changement de la pureté avec les bins en  $p_T^{trigg}$

Pour les plots à venir, j'ai mergé les données suivantes :

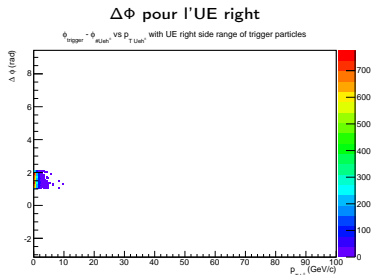
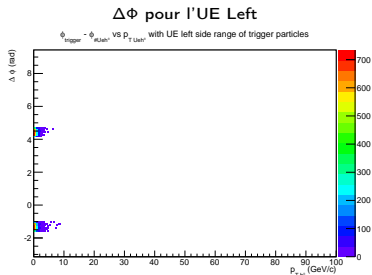
- LHC13d - bad runs → 66% de la statistique de la période
- LHC13e - bad runs → 87% de la statistique de la période
- LHC13f - bad runs → 88% de la statistique de la période

⇒ Il reste 84% de la statistique initiale après avoir enlevé les mauvais runs (la liste peut encore être modifiée).

# Comparaison de l'UE

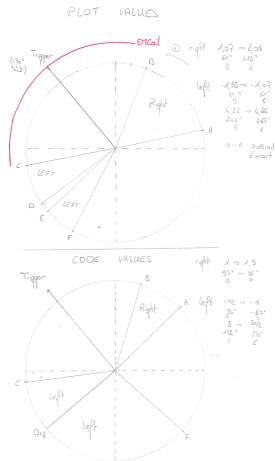
On a remarqué un problème dans le code pour les valeurs des coupures en  $\Delta\Phi$  pour l'estimation de l'UE.

Mais les valeurs sur le plot semblent correctes



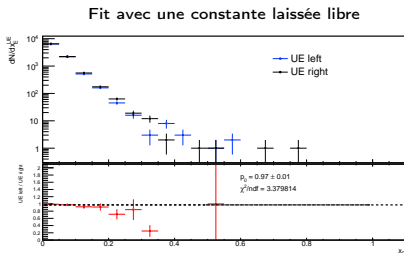
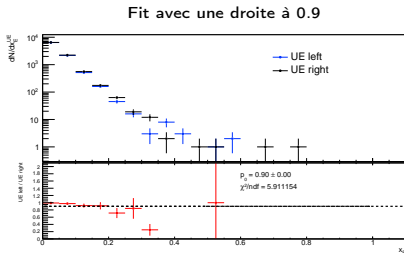
# Comparaison de l'UE : Valeurs de $\Delta\Phi$ pour le code

Si on regarde attentivement les valeurs il y a une asymétrie entre les côtés Left et Right  
→ on doit voir si l'asymétrie est significative



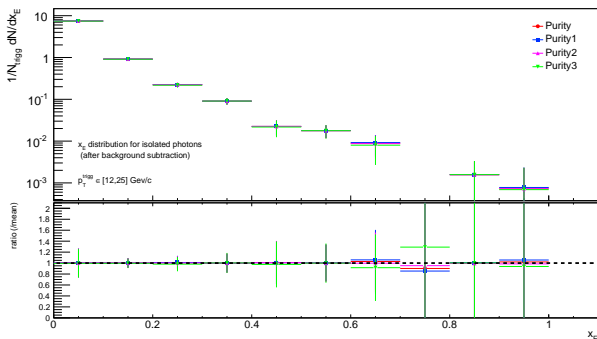
# Asymétrie UE Left et Right

En faisant le ratio des deux acceptances azimutales on s'attend à trouver un ratio Left/Right de 0.9 ( $\approx 10\%$ )



# Les différentes méthodes de pureté

Catherine m'a donné 4 datasets de pureté, chacun d'une méthode différente. Verification pour voir si tout est compatible  
Pour plus tard : trouver l'erreur systématique associée si elle existe



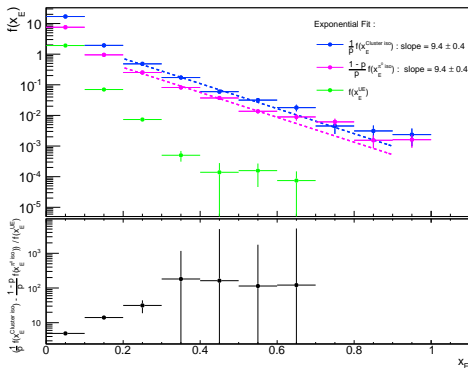


Plot pour les différentes distributions en  $x_E$  qui tient compte de la pureté.

On voit que l'UE (left + right) est négligeable face aux autres termes  $\Rightarrow$  moindre importance de ne pas avoir de statistique à haut  $x_E$  pour l'UE

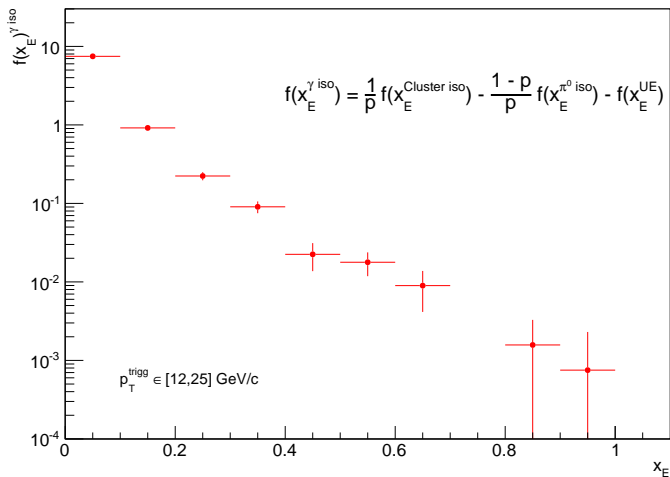
Les valeurs de fit sont semblables à celles de Nicolas.  
!!! Pas encore de correction en acceptance du détecteur!!!

12 bins en  $p_T^{trigg}$



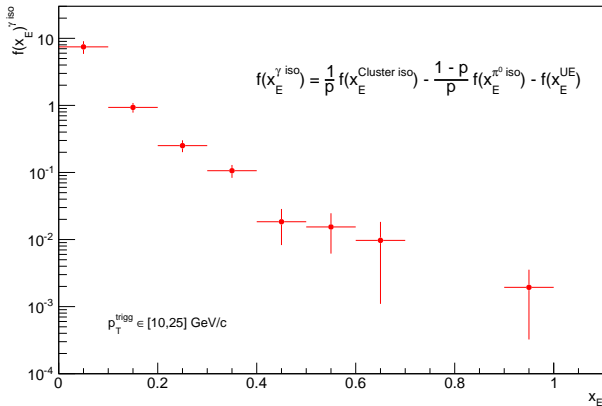
# Distribution en $x_E$ des photons isolés

Avec les distributions précédentes on obtient la distribution en  $x_E$  pour les photons isolés : (bin 0.7 négatif)



Données triggées L1 + L2 pour le plot

Fausse valeurs de la pureté pour de 10 à 12  $\text{GeV}/c$  : En dessous de 12  $\text{GeV}/c$  on estime mal le nombre de photons isolés  $\rightarrow$  on peut difficilement descendre plus bas en  $p_T^{trigg}$



- Prendre en compte la correction en acceptance du détecteur
- Tester le code pour les valeurs en  $\Delta\Phi$
- Incertitudes systématiques sur la pureté

# BACKUP

```
void AliAnaParticleHadronCorrelation::FillChargedUnderlyingEventSidesHistograms(Float_t ptTrig,
                                                                              Float_t ptAssoc,
                                                                              Float_t deltaPhi)
{
// Fill underlying event histograms to the left and right of trigger
if((deltaPhi<-fUeDeltaPhiMinCut) || (deltaPhi >2*fUeDeltaPhiMaxCut))
{
    fhDeltaPhiUeLeftCharged->Fill(ptAssoc,deltaPhi);
    Double_t randomphi = gRandom->Uniform(fDeltaPhiMinCut,fDeltaPhiMaxCut);
    Double_t uexE = -(ptAssoc/ptTrig)*TMath::Cos(randomphi);
    Double_t uezT = ptAssoc/ptTrig;

    if(uexE < 0.) uexE = -uexE;

    fhXEUELeftCharged->Fill(ptTrig,uexE);
    if(uexE > 0) fhPtHbpXEUELeftCharged->Fill(ptTrig,TMath::Log(1/uexE));

    fhZTUELeftCharged->Fill(ptTrig,uezT);
    if(uezT > 0) fhPtHbpZTUELeftCharged->Fill(ptTrig,TMath::Log(1/uezT));
    fhDeltaPhiUeLeftCharged->Fill(ptAssoc, deltaPhi);
}
}
```

```
if((deltaPhi > fUeDeltaPhiMinCut) && (deltaPhi < fUeDeltaPhiMaxCut))
{
  fhDeltaPhiUeRightCharged->Fill(ptAssoc,deltaPhi);
  Double_t randomphi = gRandom->Uniform(fDeltaPhiMinCut,fDeltaPhiMaxCut);
  Double_t uexE = -(ptAssoc/ptTrig)*TMath::Cos(randomphi);
  Double_t uezT = ptAssoc/ptTrig;

  if(uexE < 0.) uexE = -uexE;

  fhXEUeRightCharged->Fill(ptTrig,uexE);
  if(uexE > 0) fhPtHbpXEUeRightCharged->Fill(ptTrig,TMath::Log(1/uexE));

  fhZTUeRightCharged->Fill(ptTrig,uezT);
  if(uezT > 0) fhPtHbpZTUeRightCharged->Fill(ptTrig,TMath::Log(1/uezT));
  fhDeltaPhiUeRightCharged->Fill(ptAssoc, deltaPhi);
}
```

# Back Up : datasets de la pureté

```
Double_t purity[nbSliceBin] = {0.2837,0.3850,0.4544,0.5412,0.5645,0.6677,  
0.6763,0.7144,0.6988,0.7663,0.7549,0.8311};
```

```
Double_t purity_err[nbSliceBin] = {0.0193,0.0210,0.0227,0.0230,0.0257,  
0.0238,0.0269,0.0273,0.0328,0.0297,0.0266,0.0247};
```

```
Double_t purity1[nbSliceBin] = {0.2797,0.3816,0.4550,0.5466,0.5655,0.6796,  
0.6836,0.7201,0.7117,0.7856,0.7659,0.8366};
```

```
Double_t purity1_err[nbSliceBin] = {0.0193,0.0210,0.0227,0.0230,0.0257,0.0238,  
0.0269,0.0273,0.0328,0.0297,0.0266,0.0247};
```

```
Double_t purity2[nbSliceBin] = {0.2878,0.3885,0.4538,0.5358,0.5635,0.6558,  
0.6690,0.7087,0.6859,0.7470,0.7438,0.8257};
```

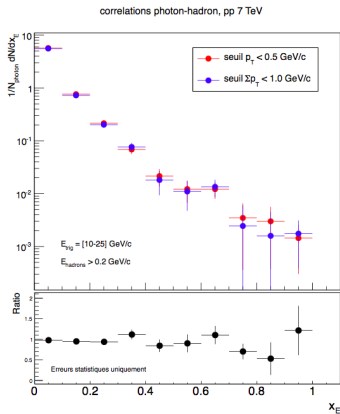
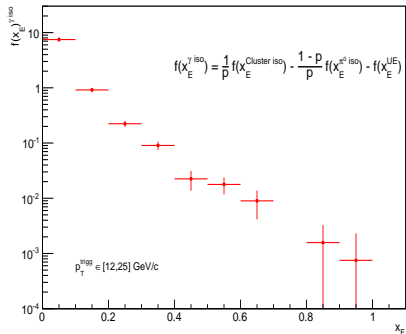
```
Double_t purity2_err[nbSliceBin] = {0.0194,0.0201,0.0214,0.0220,0.0238,0.0228,  
0.0251,0.0257,0.0326,0.0315,0.0267,0.0249};
```

```
Double_t purity3[nbSliceBin] = {0.2759,0.3760,0.4370,0.5097,0.5415,0.6161,  
0.6370,0.6756,0.6418,0.6850,0.6951,0.7843};
```

```
Double_t purity3_err[nbSliceBin] = {0.0169,0.0175,0.0187,0.0192,0.0204,0.0190,  
0.0212,0.0212,0.0275,0.0254,0.0214,0.0168};
```



# Back Up : Plot Nicolas Distribution en $x_E$ des photons isolés



# Back Up : Plot Nicolas Fit des distribution en $x_E$

