



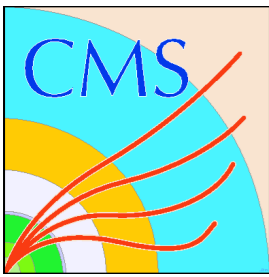
CMS Experiment at LHC, CERN
Data recorded: Tue Oct 26 19:50:37 2010 CEST
Run/Event: 149058 / 76943429
Lumi section: 64



Jet Yapılandırma ve FastJet Programı

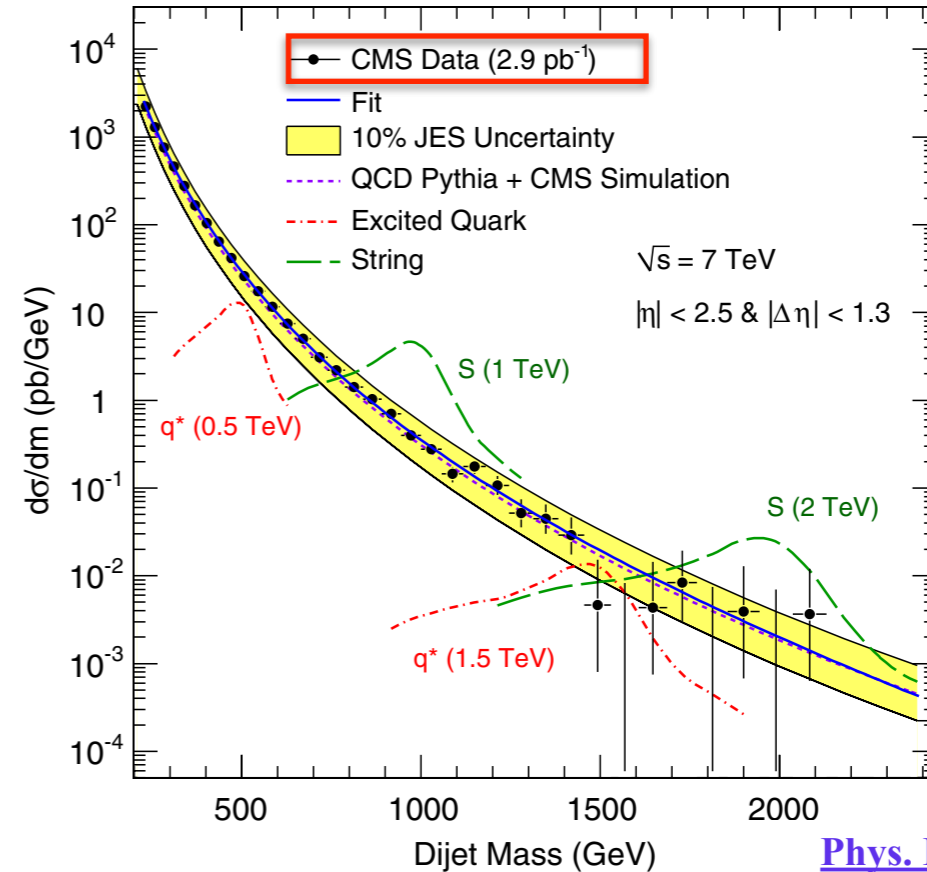
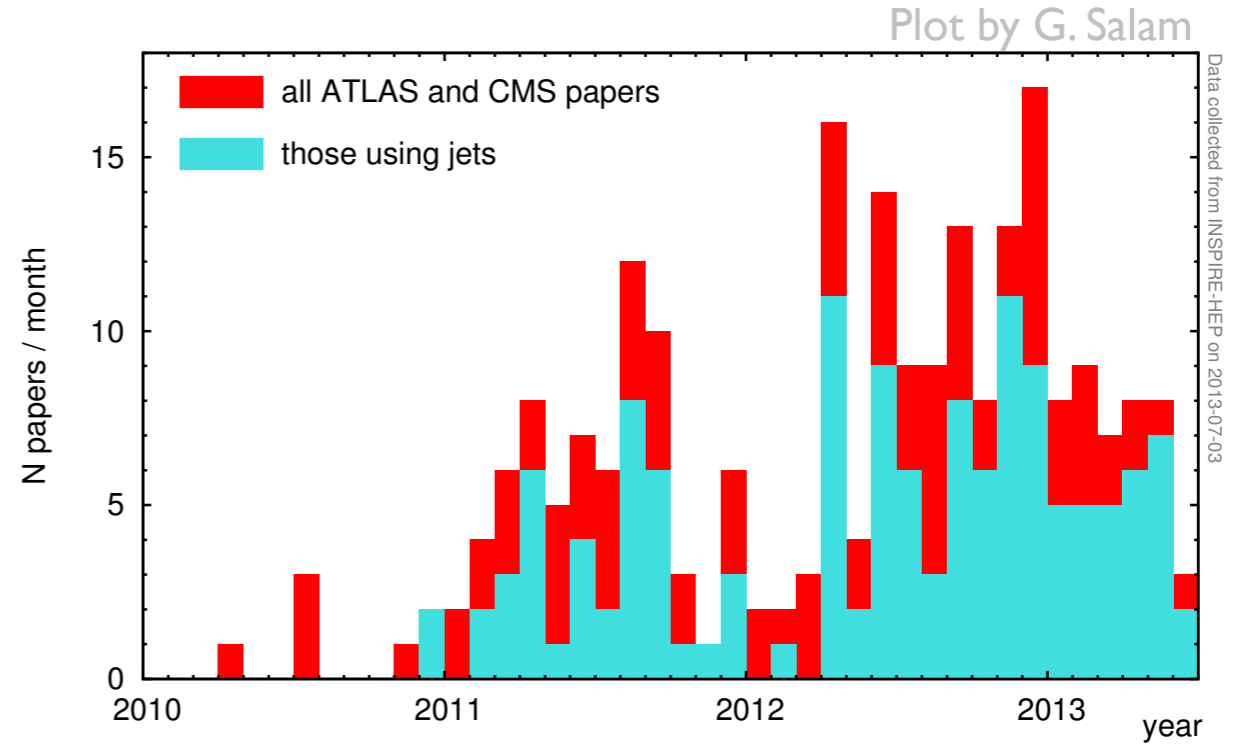
Sertaç Öztürk

Gaziosmanpaşa Üniversitesi
ve
University of Iowa



Dersin Amacı

- ☑ CMS+ATLAS araştırma makalelerinin %60 ı jetleri içermektedir.
- ☑ CMS in ilk araştırma makalesi jetler ile ilgilidir.
- ☑ Dersin amacı
 - ✓ Güçlü Etkileşimi Anlamak
 - ✓ Hadronik çarpışmaların yapısını anlamak
 - ✓ Jetleri anlamak (Bir jet makalesini anlamak)
 - ✓ FastJet programını öğrenmek
 - ✓ Jet içeren bir makaleyi anlayıp yorumlayabilmek.
- ☑ Kaynaklar
 - ✓ Sunumlar ve dersler (Gavin Salam, Matthew Schwartz, Matteo Cacciari, Gregory Soyez)





Standart Model

✓ Standart model atomaltı parçacıkları ve birbirleri arasındaki etkileşimleri açıklar.

✓ Standart Modele göre,

✓ 6 kuark & 6 lepton

→ u ve d kuark and elektron maddeyi oluşturur

✓ 4 kuvvet taşıyıcı parçacık (γ , W, Z and g)

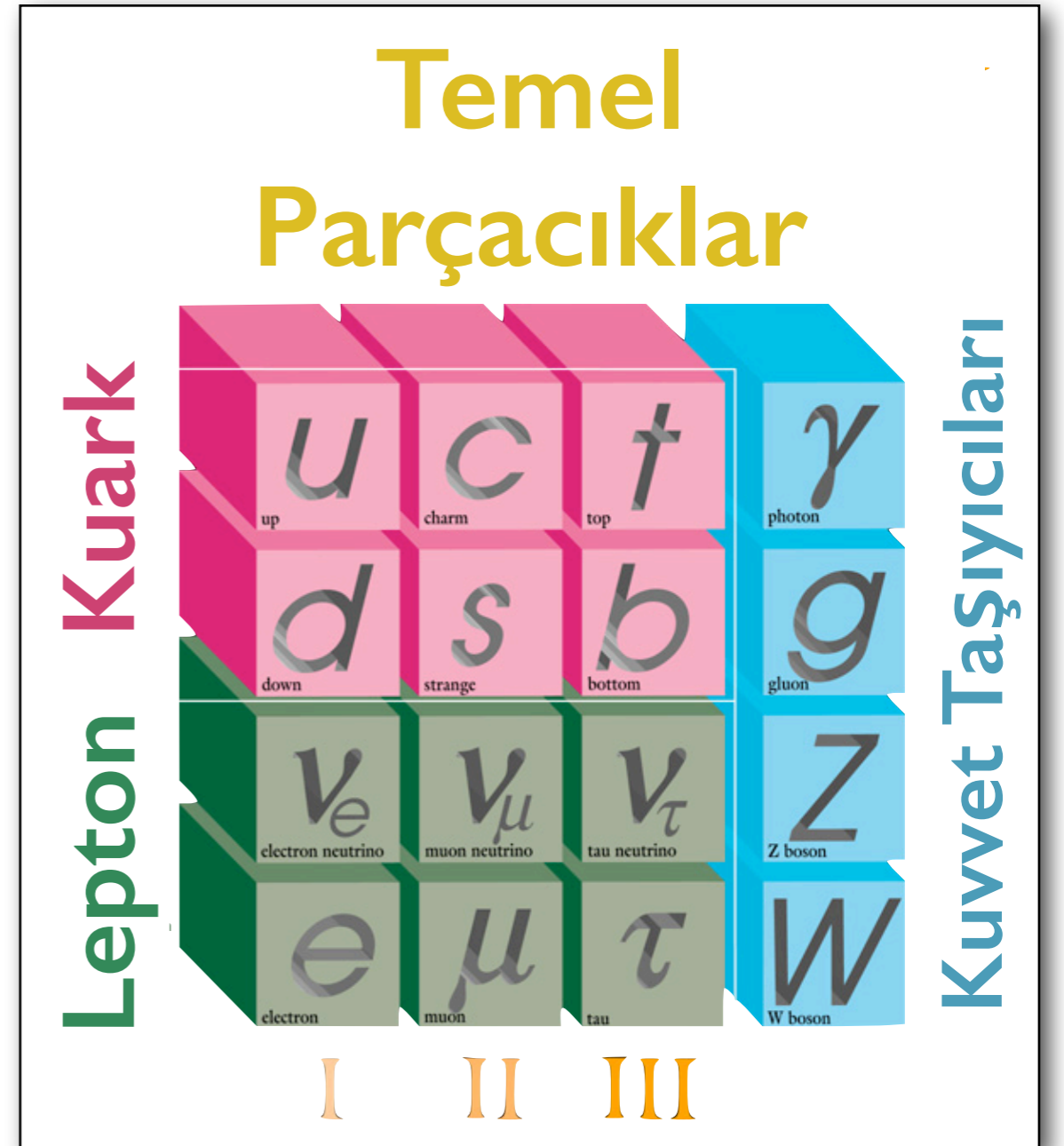
→ γ : Elektromagnetizma

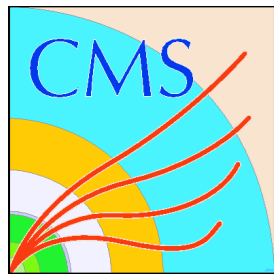
→ W & Z : Zayıf Etkileşim

→ g : Kuvvetli Etkileşim

✓ Kütle kazandırmak için Higgs bozonu

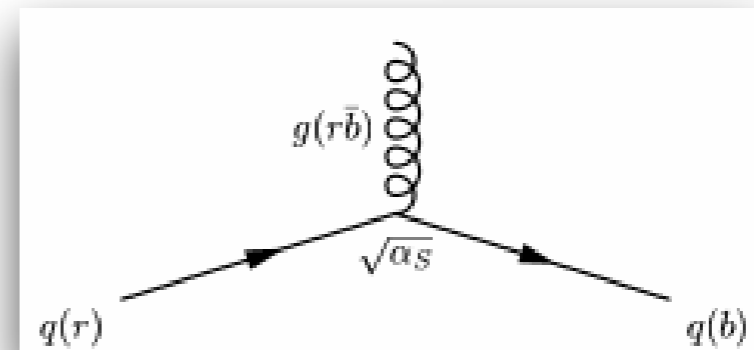
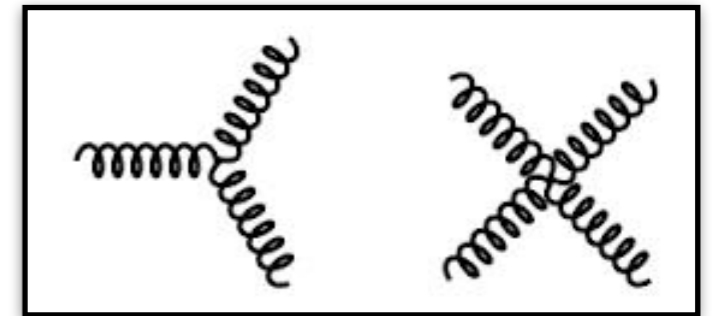
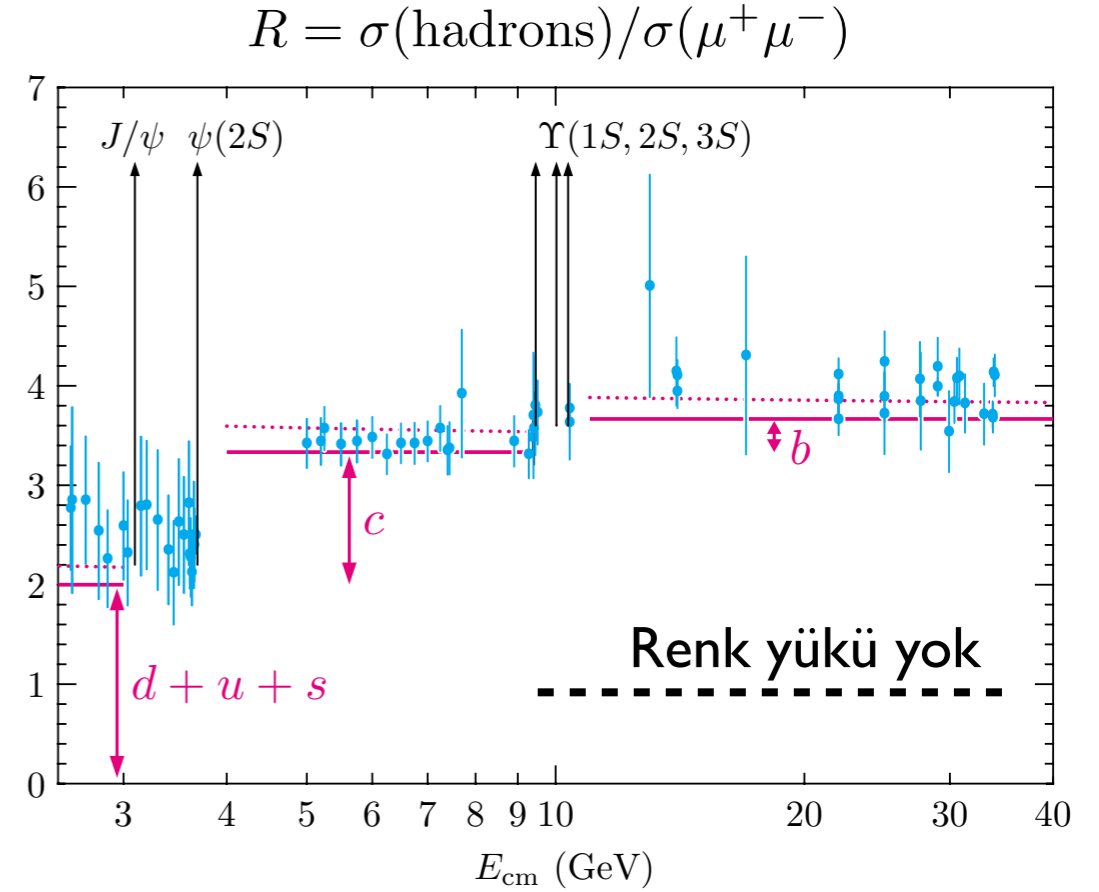
→ 4 Temmuz 2012

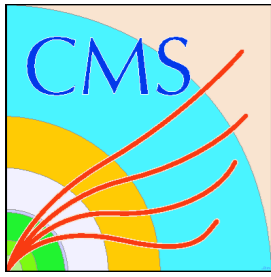




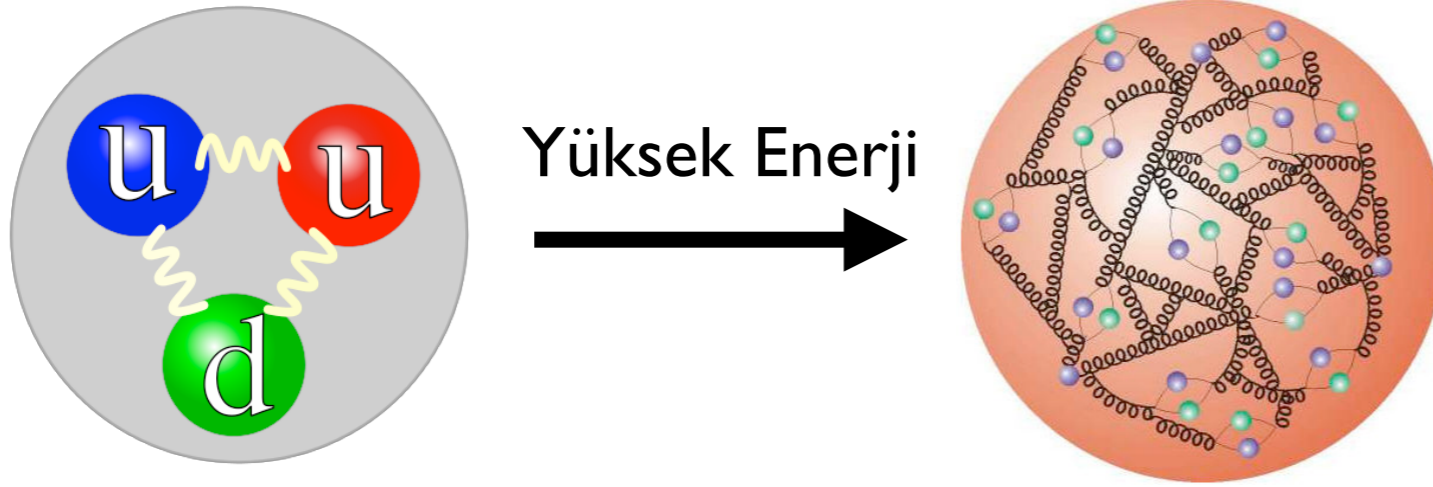
Kuantum Renk Dinamiği

- ✓ Kuantum Renk Dinamiği renkli kuarkların ve gluonların etkileşmelerini açıklamayan kuvvetli etkileşmenin teorisidir.
- ✓ Kuarklar ve gluonlar renk yüküne sahiptir (**r**, **g**, **b**).
- ✓ Serbest parçacıkların renk yükü nötrdür. (**r** + **g** + **b** → Beyaz)
- ✓ Gluonlar birbiri ile etkileşir.
- ✓ Kuarklar tek renk yüküne sahipken, gluonlar çift renk yüküne sahiptir.
- ✓ 8 tane gluon vardır.

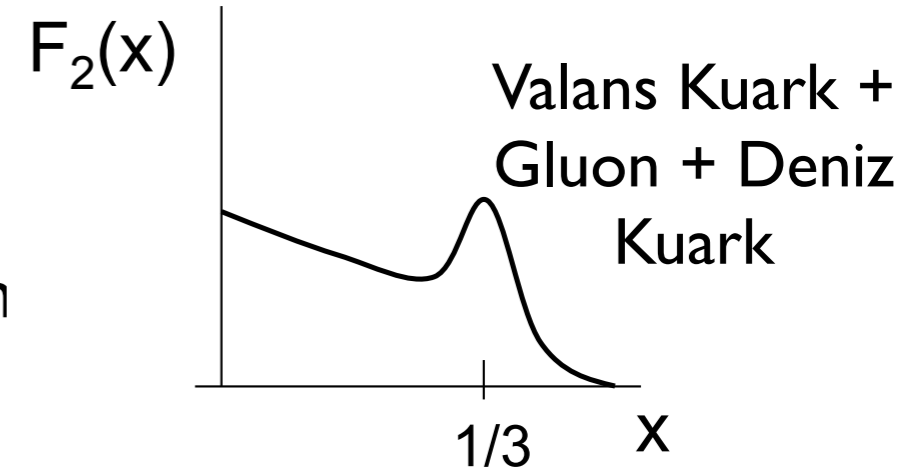
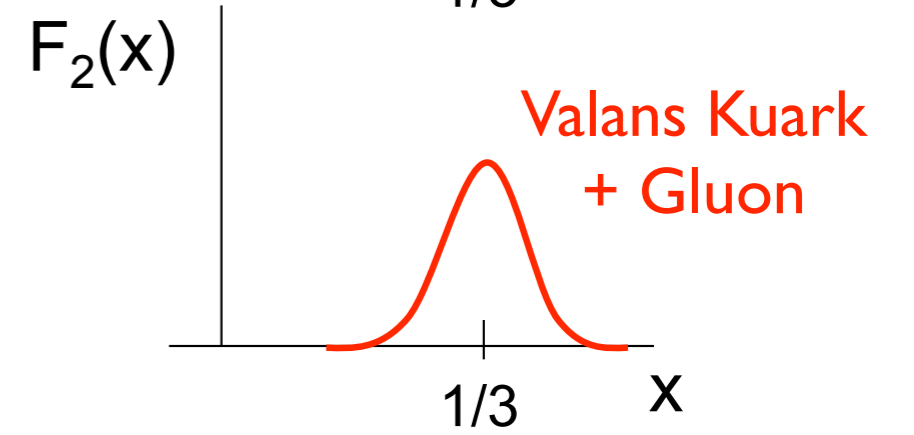
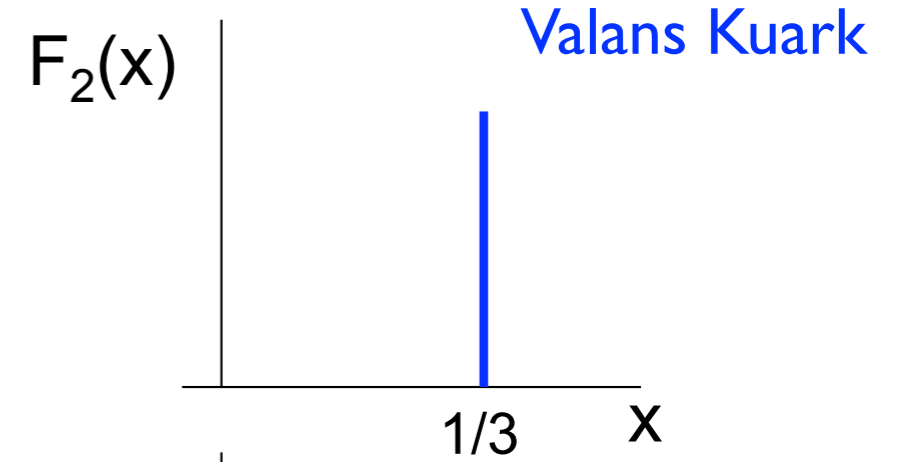


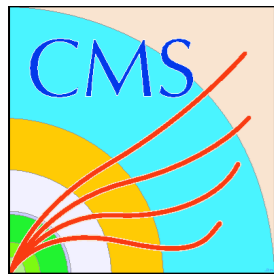


Proton ve Parton Dağılım Fonksiyonu

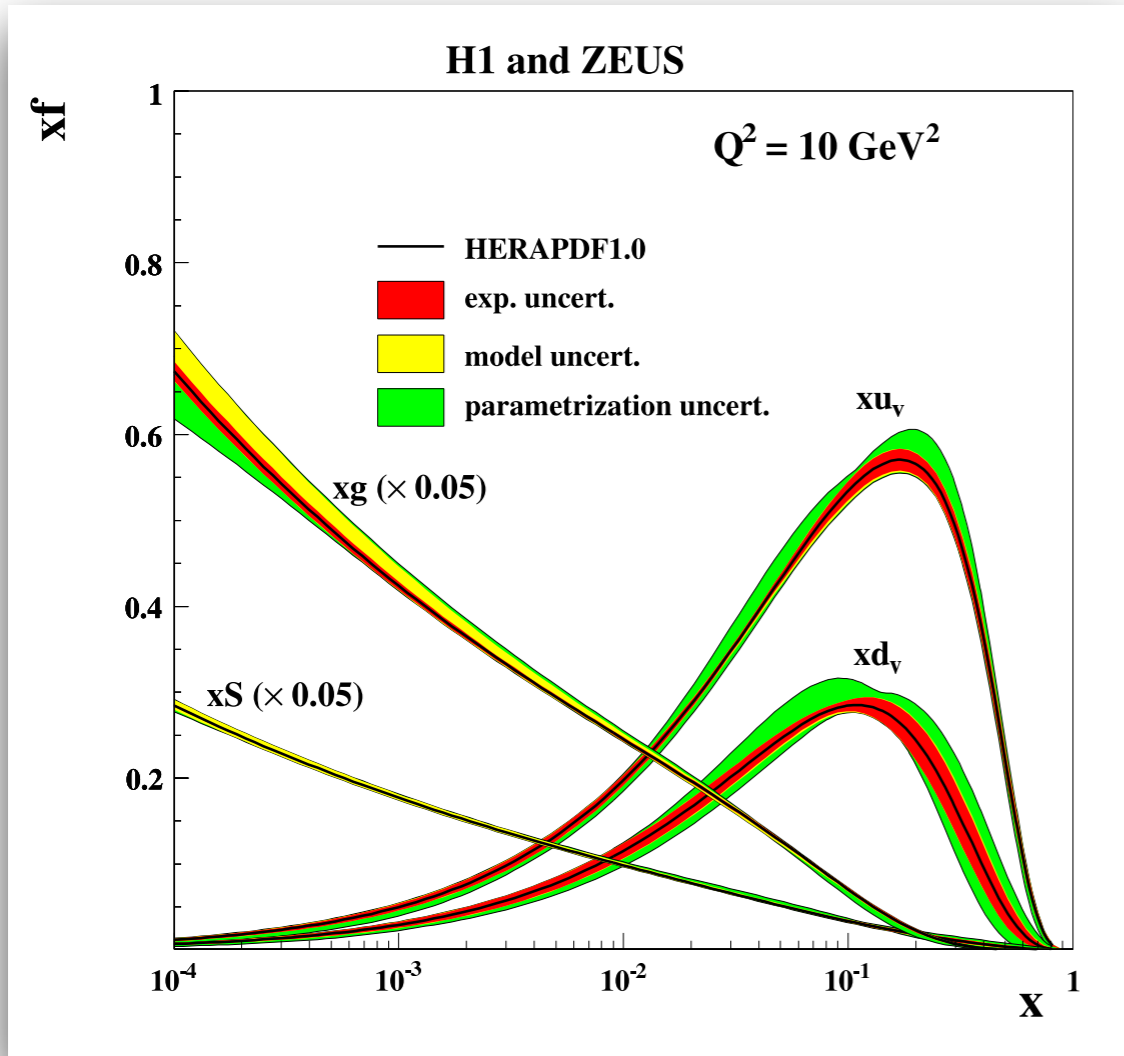


- ✓ Proton uud kuark birleşiminden oluşur.
 - ✓ Valans kuark
 - ✓ Gluonlar
- ✓ Yüksek enerjilere çıkıldığında proton içerisinde kuark-anti kuark çiftleri oluşup yok olur.
 - ✓ Deniz kuarklar
- ✓ Proton momentumu içindeki partonlar tarafından taşınır.
 - ✓ $x = P_{\text{parton}} / P_{\text{proton}}$





Parton Dağılım Fonksiyonu

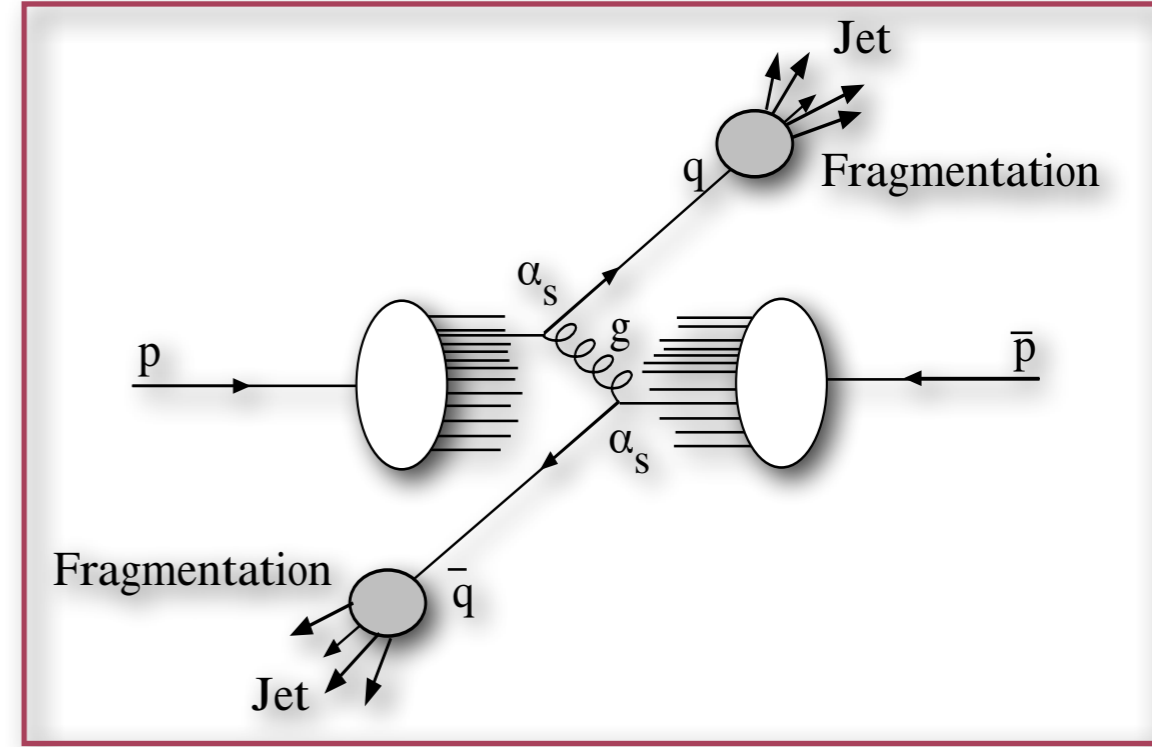


- ✓ Hadronu meydana getiren bileşenlerin toplam hadron momentumu taşıma kesri parton dağılım fonksiyonu (PDF) ile belirtilir.
- ✓ $f_i(x_i)$: i. partonun toplam hadron momentumunun x_i kesrini taşıma olasılığı
 - ✓ Eğer proton 3.5 TeV lik bir protonda bir kuark için $x=0.1$ ise 350 GeV lik bir momentumu tek başına taşır.
- ✓ Tesir kesiti PDF ile yakından ilişkilidir.

Çevrimiçi PDF çizdirme ve hesaplama
<http://hepdata.cedar.ac.uk/pdf/pdf3.html>

Sert Saçılma

- ☑ LHC deki iki hadron çarpıştığında aslında çarpışan iki partondur.
 - qq , qg, gg
- ☑ Çarpışma sonucu partonlar birbirinden saçılabilir veya kuark ve gluonlara bozunabilen parçacıklar oluşabilir.
- ☑ Kuark ve gluonlar renk yükü taşıdıkları için serbest halde bulunmazlar.
 - gluonlar yayımlar ve bu gluonlar kuark ve anti-kuark çiftlerine bozunurlar.
 - Bütün renkli objeler hadronizasyon sonucu renksiz parçacıklar oluşturur.



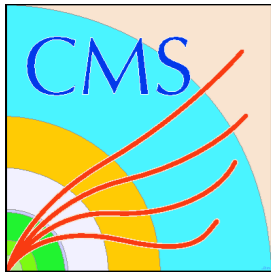
Sert saçılma tesir kesiti

$$\sigma(P_1, P_2) = \sum_{i,j} \int dx_1 dx_2 f_i(x_1, \mu_F^2) f_j(x_2, \mu_F^2) \hat{\sigma}_{ij}(p_1, p_2, \alpha_s(\mu^2), Q^2/\mu^2)$$

KRD etkileşim sabiti

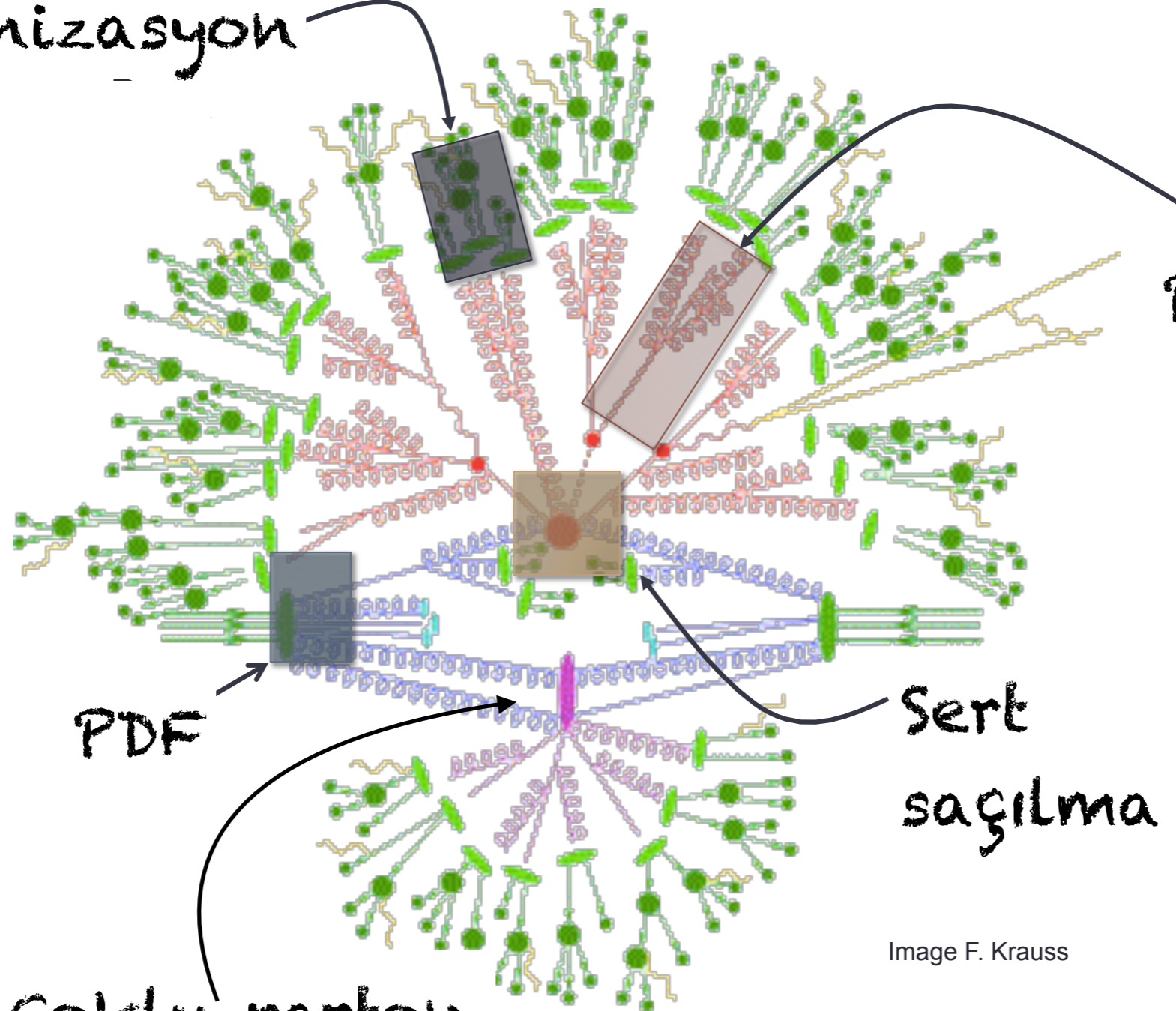
$$\alpha_s(Q^2) = \frac{12\pi}{(33 - 2n_f) \ln(\frac{Q^2}{\Lambda^2})}$$

$Q^2 \rightarrow \infty$ veya $\Lambda \rightarrow 0$ için kuarklar serbest parçacık gibi davranır, buna “asimtotik serbestlik” denir.



Hadron Çarpışması

Hadronizasyon



Parton duşu

PDF

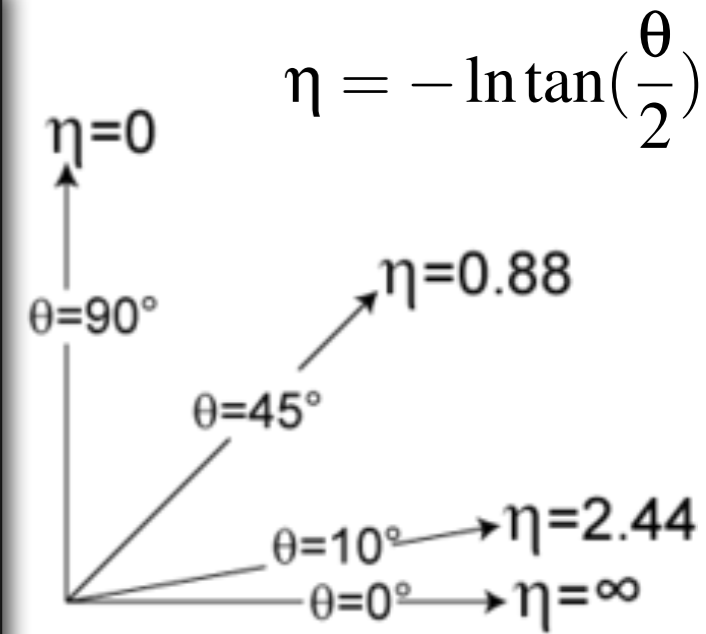
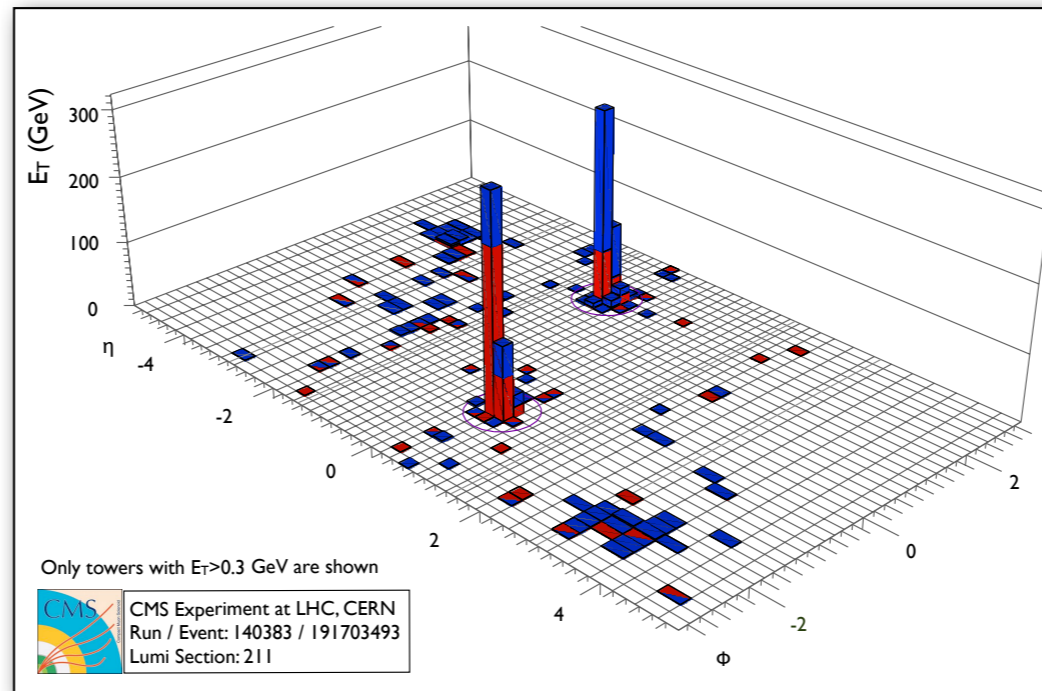
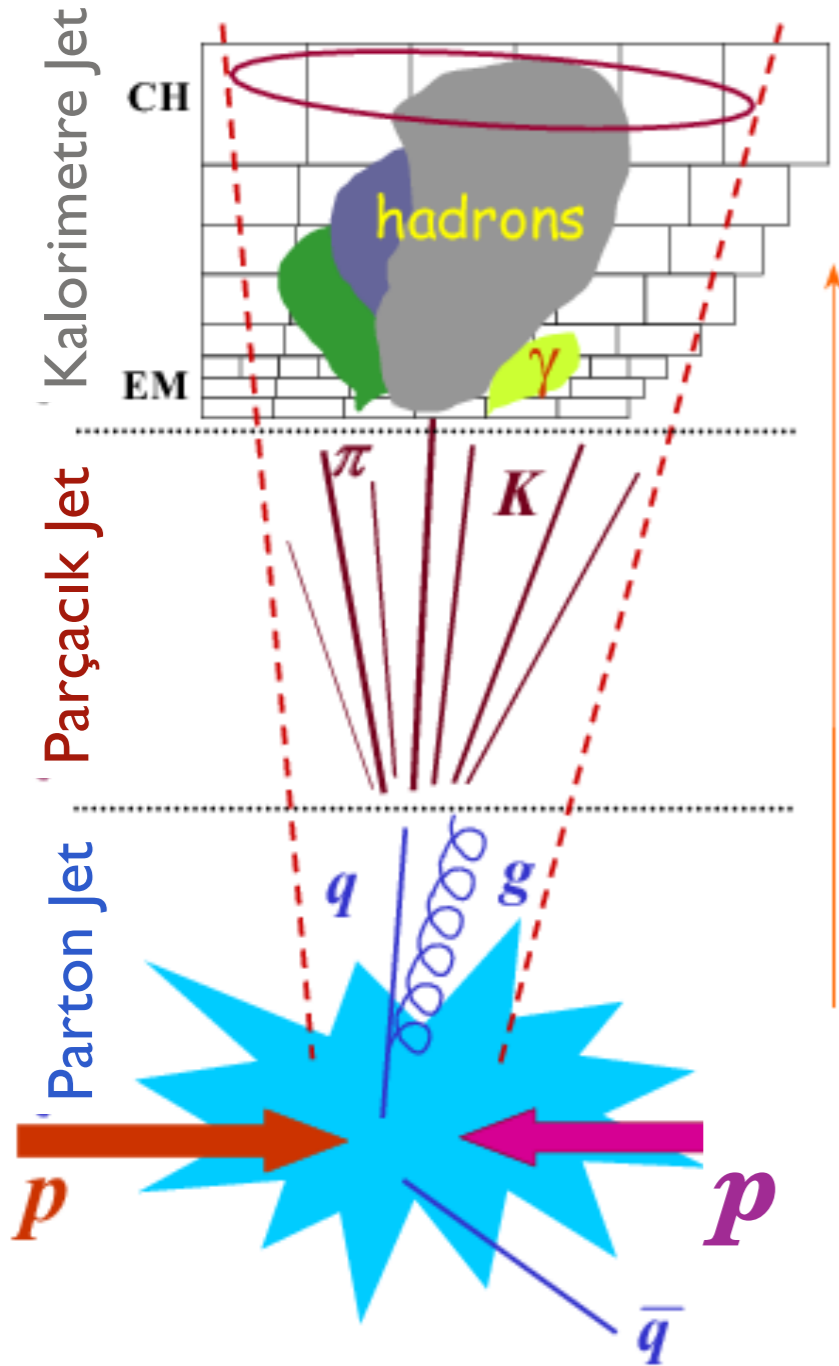
Sert saçılma

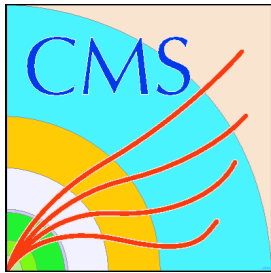
Çoklu parton etkileşimi

Image F. Krauss

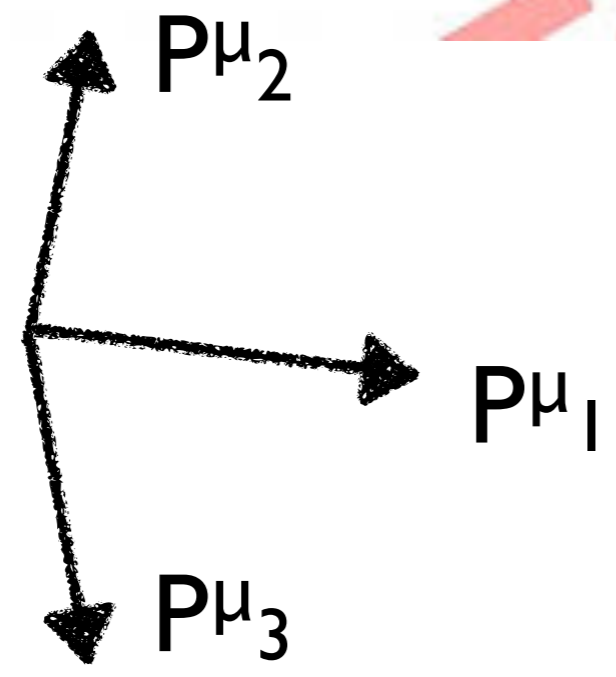
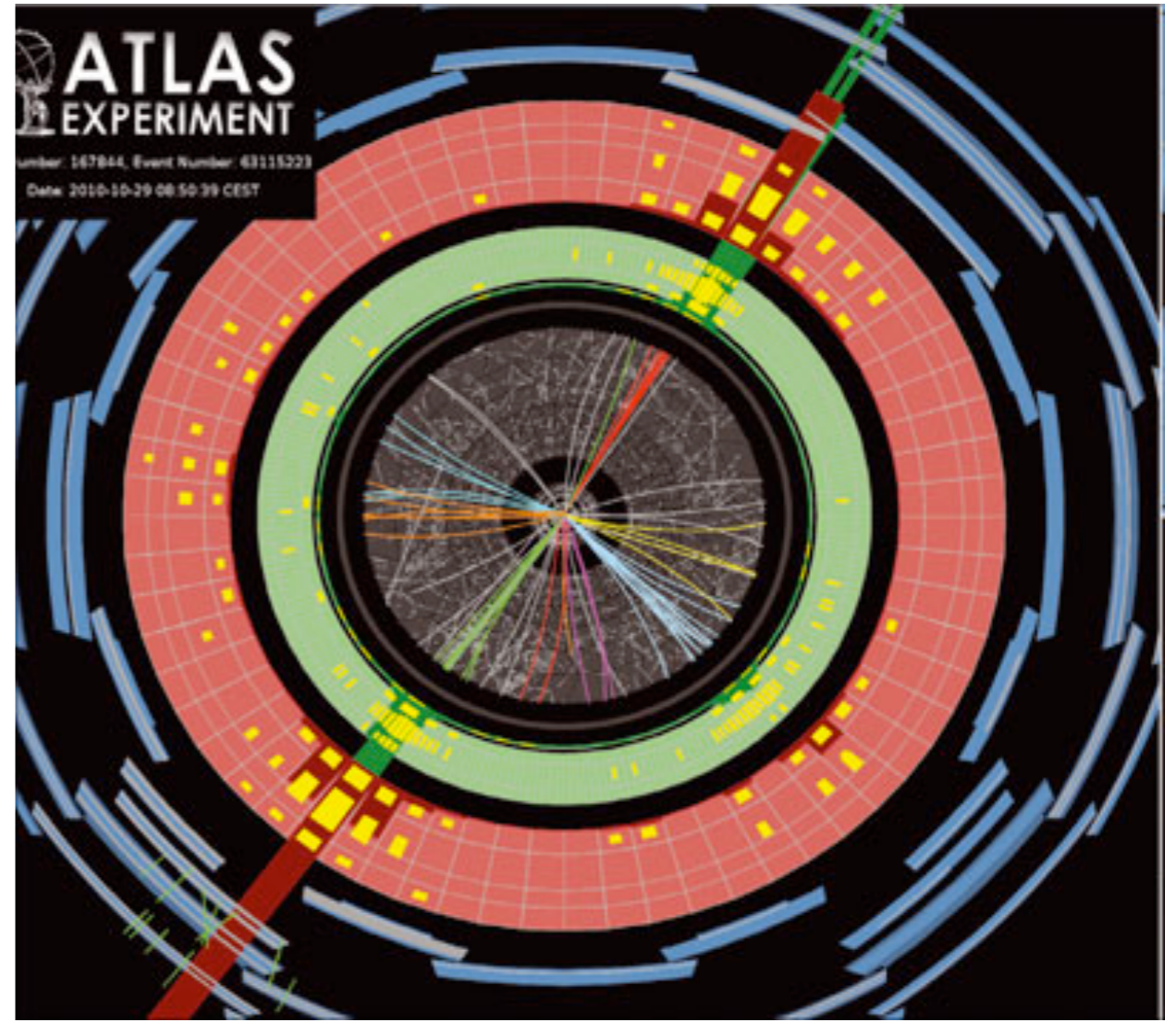
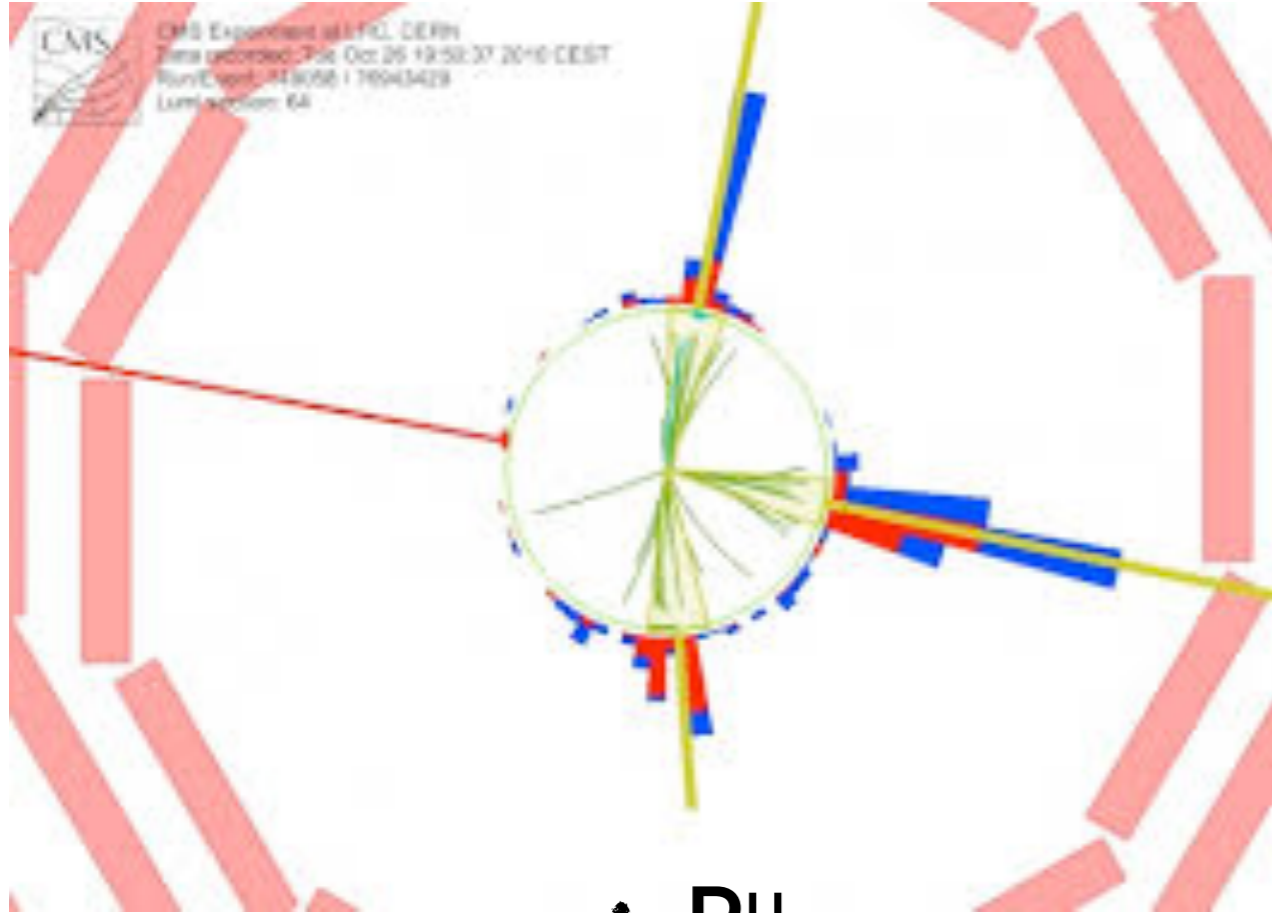
Jet Nedir

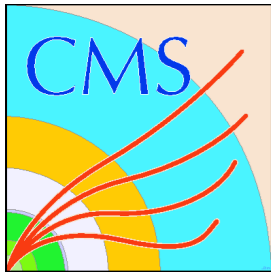
☑ Jetler yüksek enerjili çarpışmalarda açığa çıkan parçacık fıskiyesi olarak bilinen parçacık püskürtüleridir ve kuark ve glonların deneysel olarak detektörlerde gözlemlenmesidir. Detektörlerde kalorimetrelerde bıraktıkları enerjiler sayesinde tespit edilir.





Algıçlarda Jet

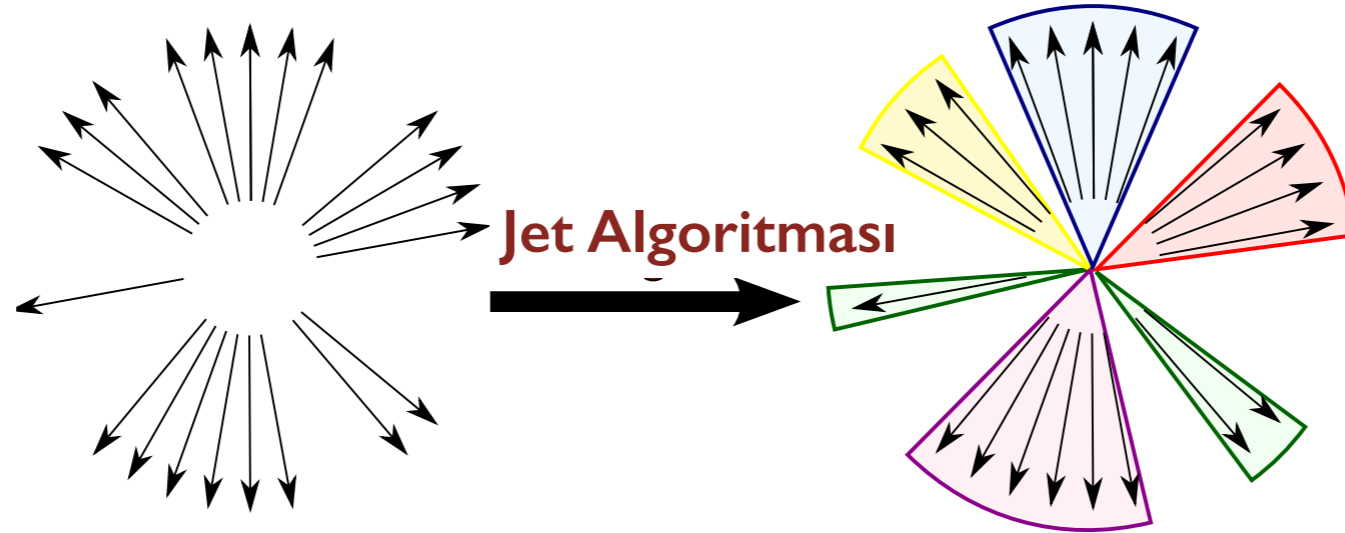




Jet Yapılandırma Algoritması

Parçacıklar:

- Partonlar
- Hadronlar
- Kalorimetre kümeleri ..



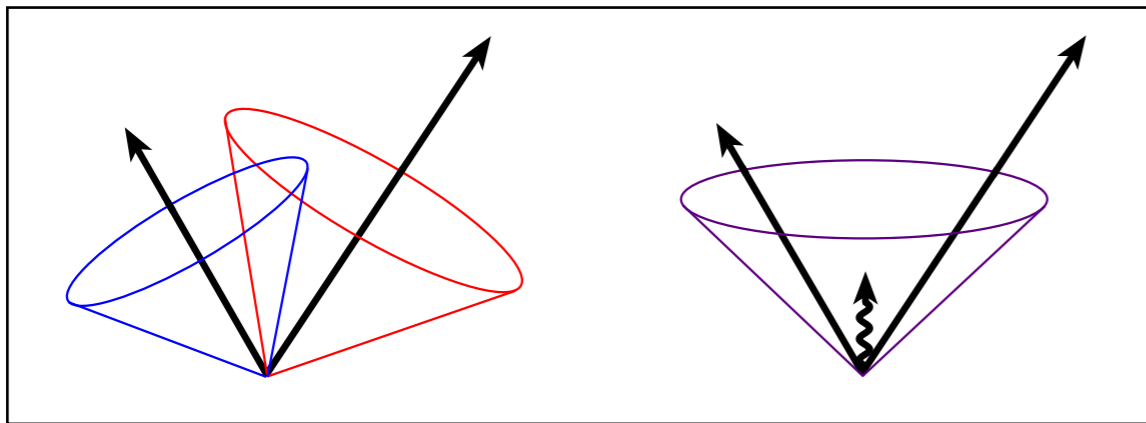
Parçacıklar

Grupları:

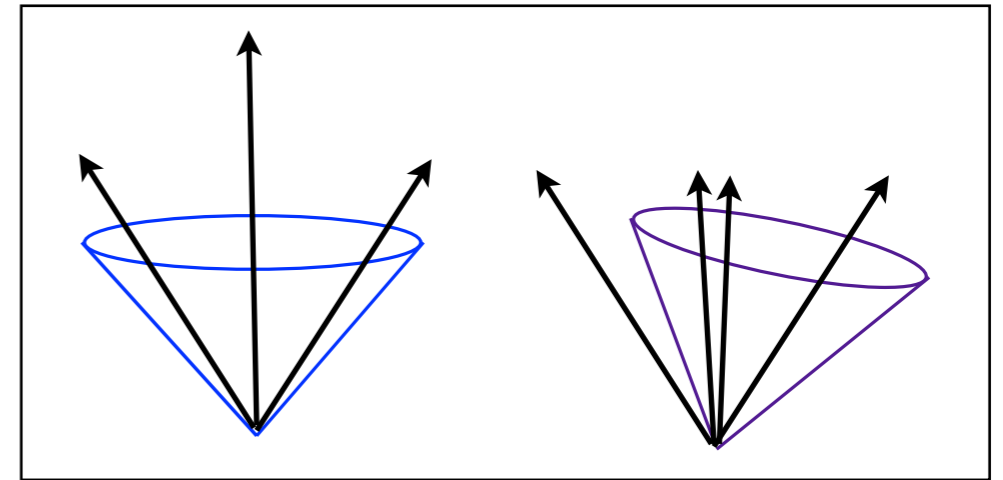
- Jetler

Teorik gereksinimler:

- ✓ infrared (kızılötesi) ve collinear (aynı yönlülük) güvenilirlilik
- ✓ seviyeye bağımsızlık (parton, hadron, kalorimetre)



Infrared güvenilirlilik: Düşük enerjili bir gluon eklendiğinde jet yapılandırılması değişmez.



Collinear güvenilirlilik: Bir partonun iki aynı yönlü parton olarak bölünmesi jet yapılandırmasını değiştirmez.



Koni Algoritması

☑ Temelde iki tip jet algoritma türü vardır.

- Koni tipi ve kümeleme tipi

☑ Koni algoritması basittir.

- En yüksek enerjili parçacığı bul ve kon yarıçapı içindeki bütün parçacıkları al.

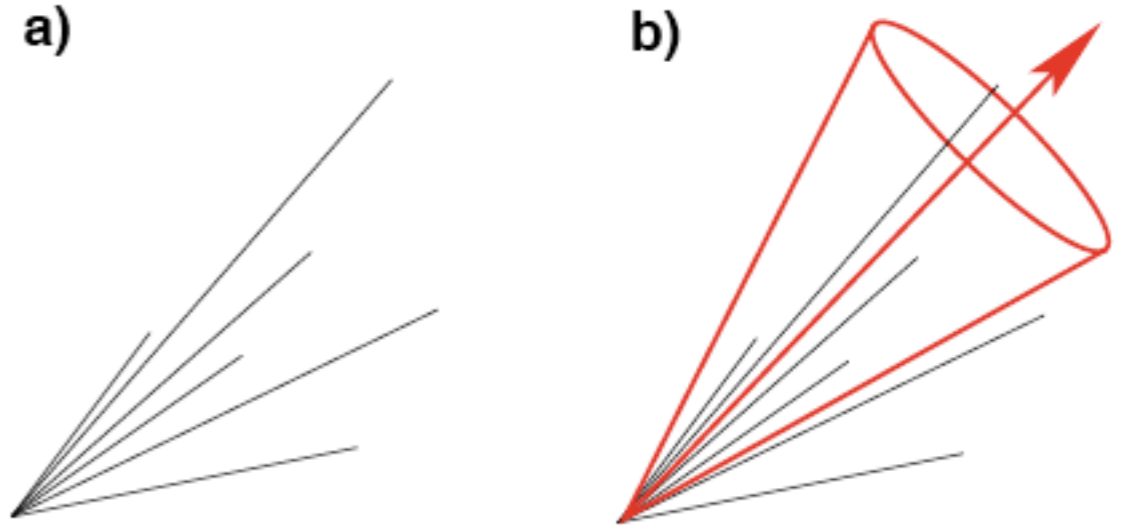
$$\Delta R_{ij} = \sqrt{(y_i - y_j)^2 + (\phi_i - \phi_j)^2} < R$$

☑ Oldukça hızlıdır.

- Tekrarlanan koni algoritması CMS de tetiklemede kullanılır.

☑ Collinear ve infrared güvenilir değildir.

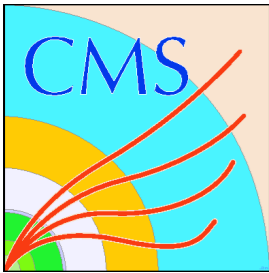
- Fizik analizlerinde artık kullanılmıyor.



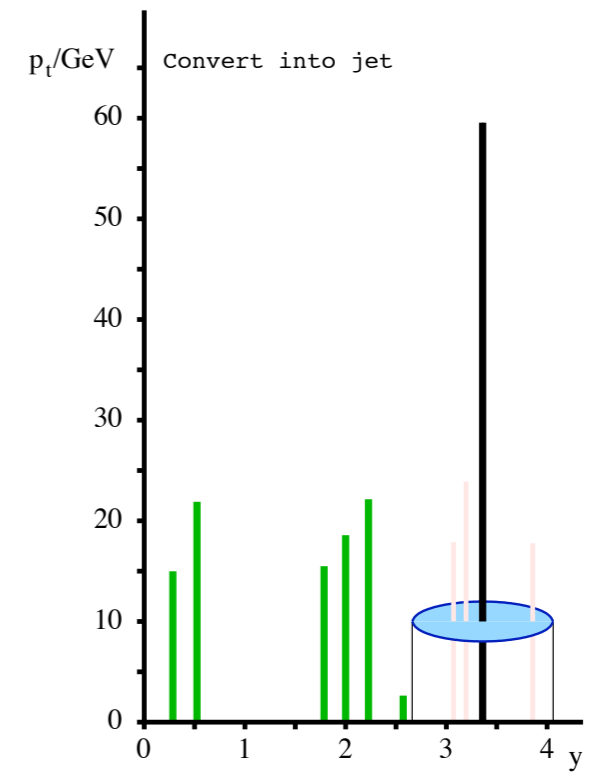
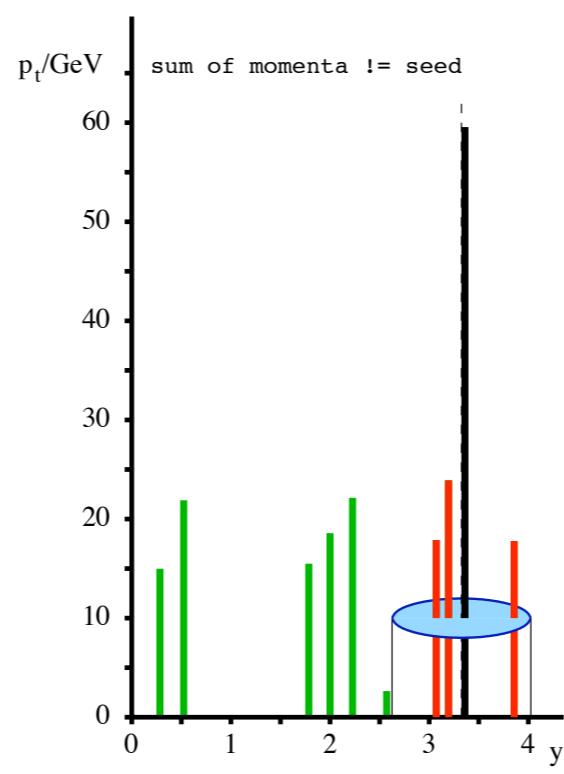
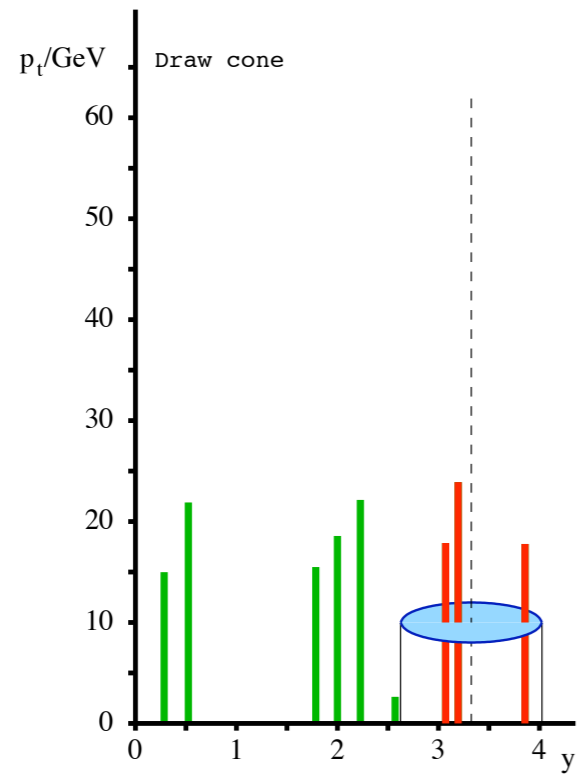
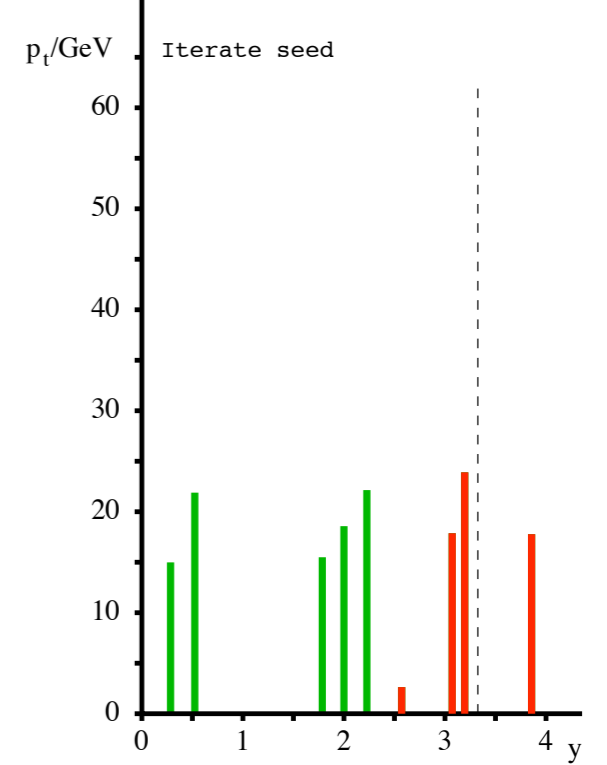
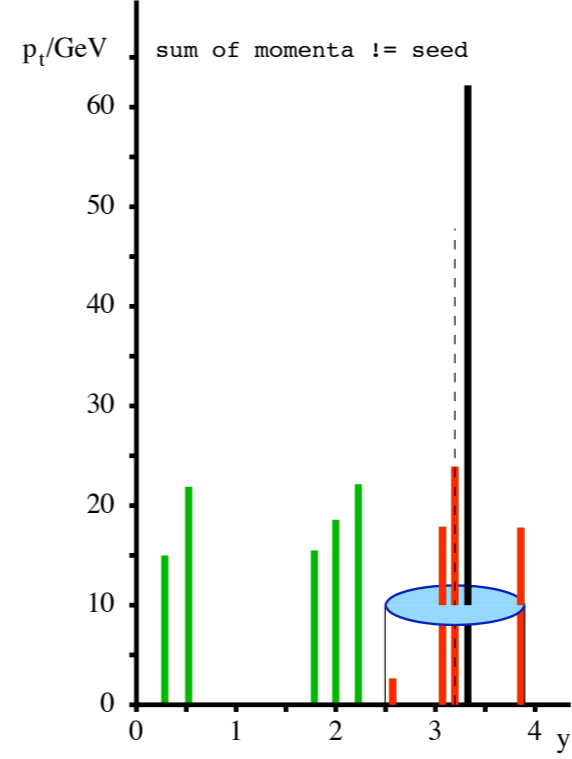
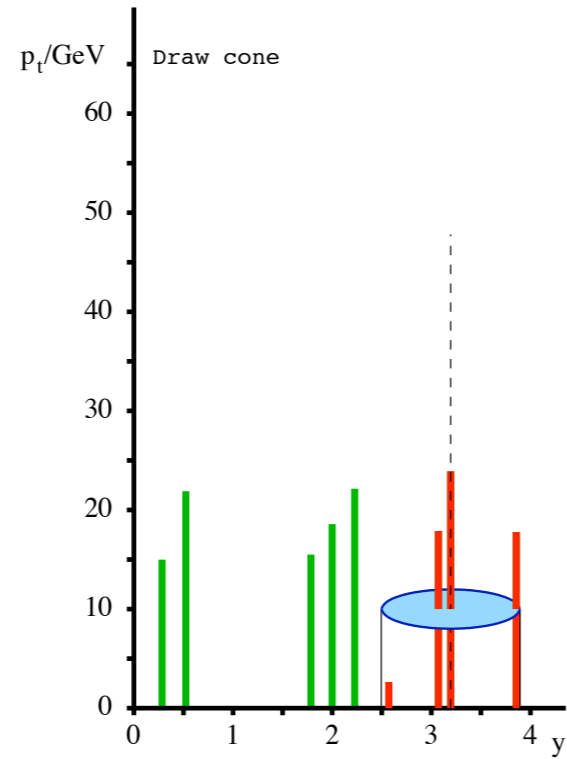
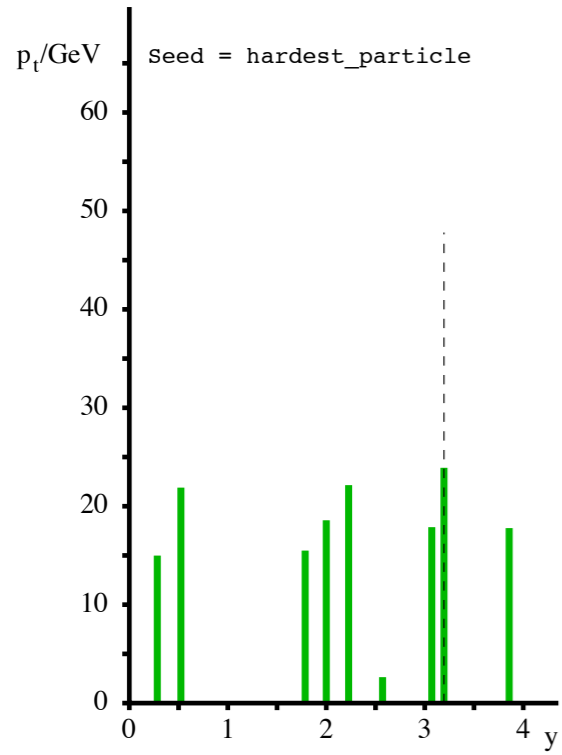
$$E_{T_{jet}} = \sum_{i \in jet} E_{T_i}$$

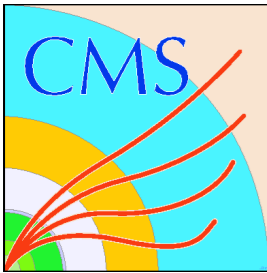
$$\eta_{jet} = \frac{1}{E_{T_{jet}}} \sum_{i \in jet} E_{T_i} \eta_i$$

$$\phi_{jet} = \frac{1}{E_{T_{jet}}} \sum_{i \in jet} E_{T_i} \phi_i$$



Tekrarlanan Koni Algoritması





Genel kt Algoritması

Düşük veya Yüksek

$$d_{ij} = \min(p_{Ti}^{2p}, p_{Tj}^{2p}) \frac{\Delta R_{ij}^2}{R^2}$$

$$d_{iB} = p_{Ti}^{2p}$$

Doğrultu

$$\Delta R_{ij}^2 = (y_i - y_j)^2 + (\phi_i - \phi_j)^2$$

p=1: “**k_T**” (düşük p_T parçacıklar)

p=0: “**Cambridge-Aachen**” (mesafe)

p=-1: “**anti-k_T**” (yüksek p_T parçacıklar)

✓ Infrared and collinear güvenir.

✓ Küme algoritması.

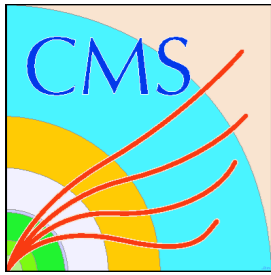
✓ Tüm parçacıkları listele.

✓ d_{ij} and d_{iB} mesafelerini hesapla,

✓ Eğer d_{ij} en küçük ise, i ve j parçacıklarını **birleştir** ve başa dön

✓ Eğer d_{iB} en küçük ise, son durumu jet olarak belirle

✓ Hiç bir parçacık kalmayıncaya kadar tekrarla.



Anti-kt Algoritması

Infrared and collinear güvenilir.

Küme algoritması.

✓ Tüm parçacıkları listele.

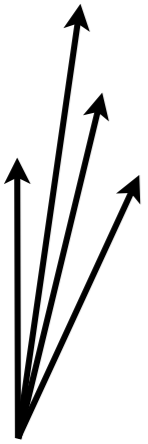
✓ d_{ij} and d_{iB} mesafelerini hesapla,

✓ Eğer d_{ij} en küçük ise, i ve j parçacıklarını birleştir ve başa dön

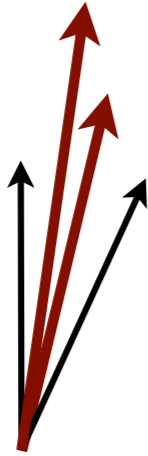
✓ Eğer d_{iB} en küçük ise, son durumu jet olarak belirle

✓ Hiç bir parçacık kalmayıncaya kadar tekrarla.

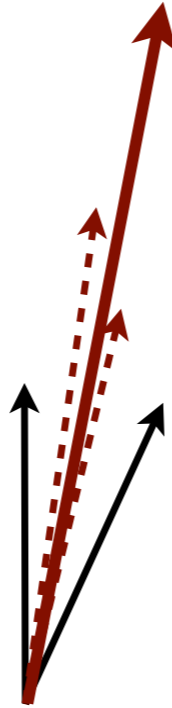
1



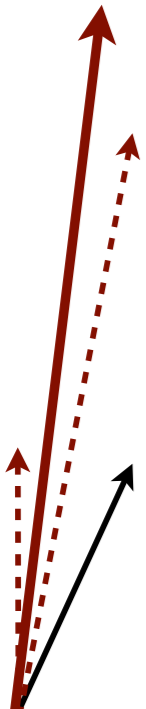
2



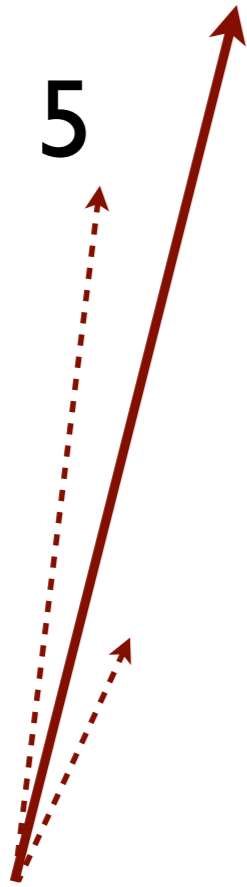
3



4

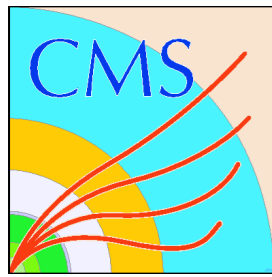


5

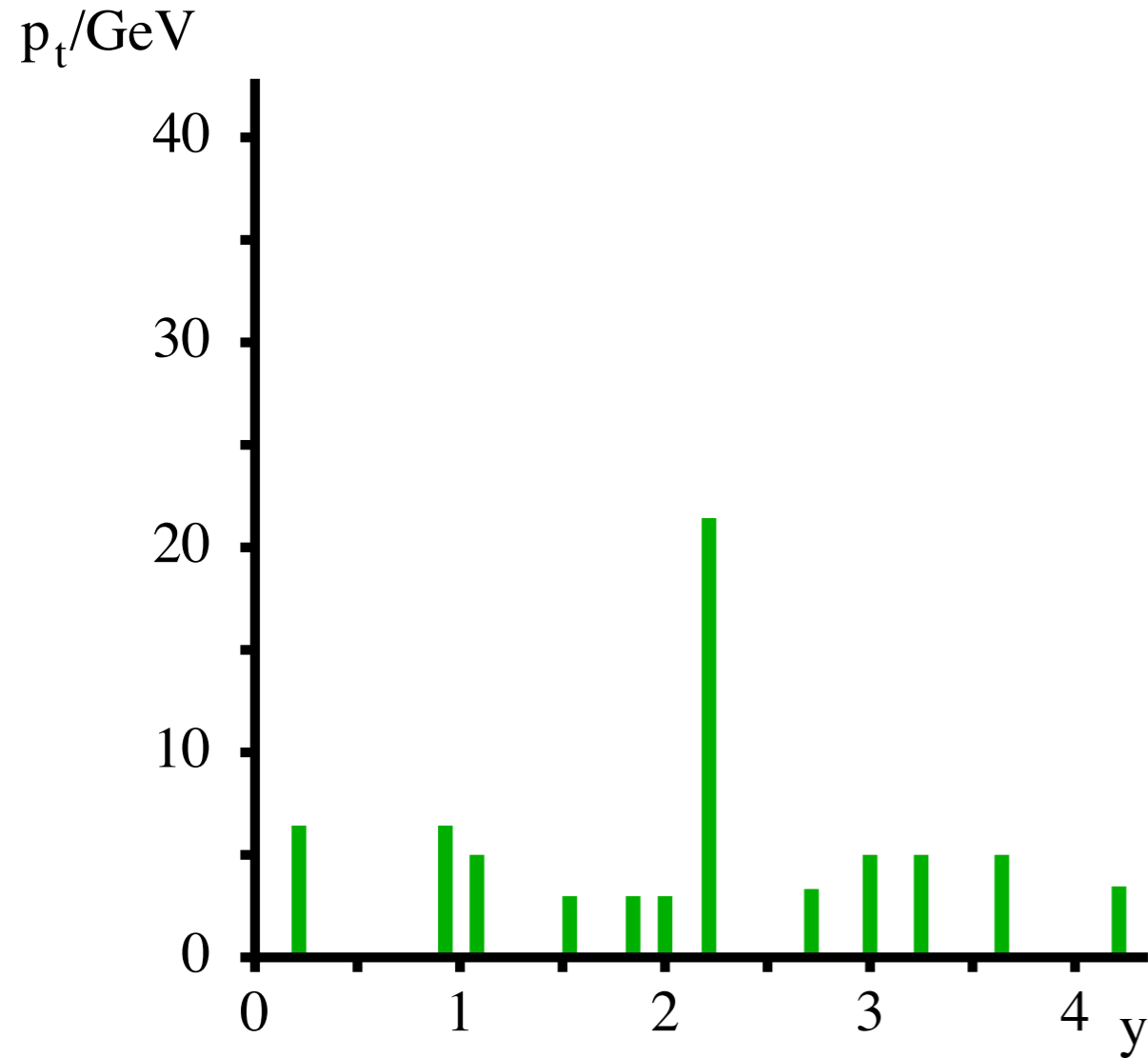


6



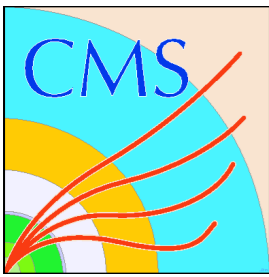


CA Algoritması Örnekleme

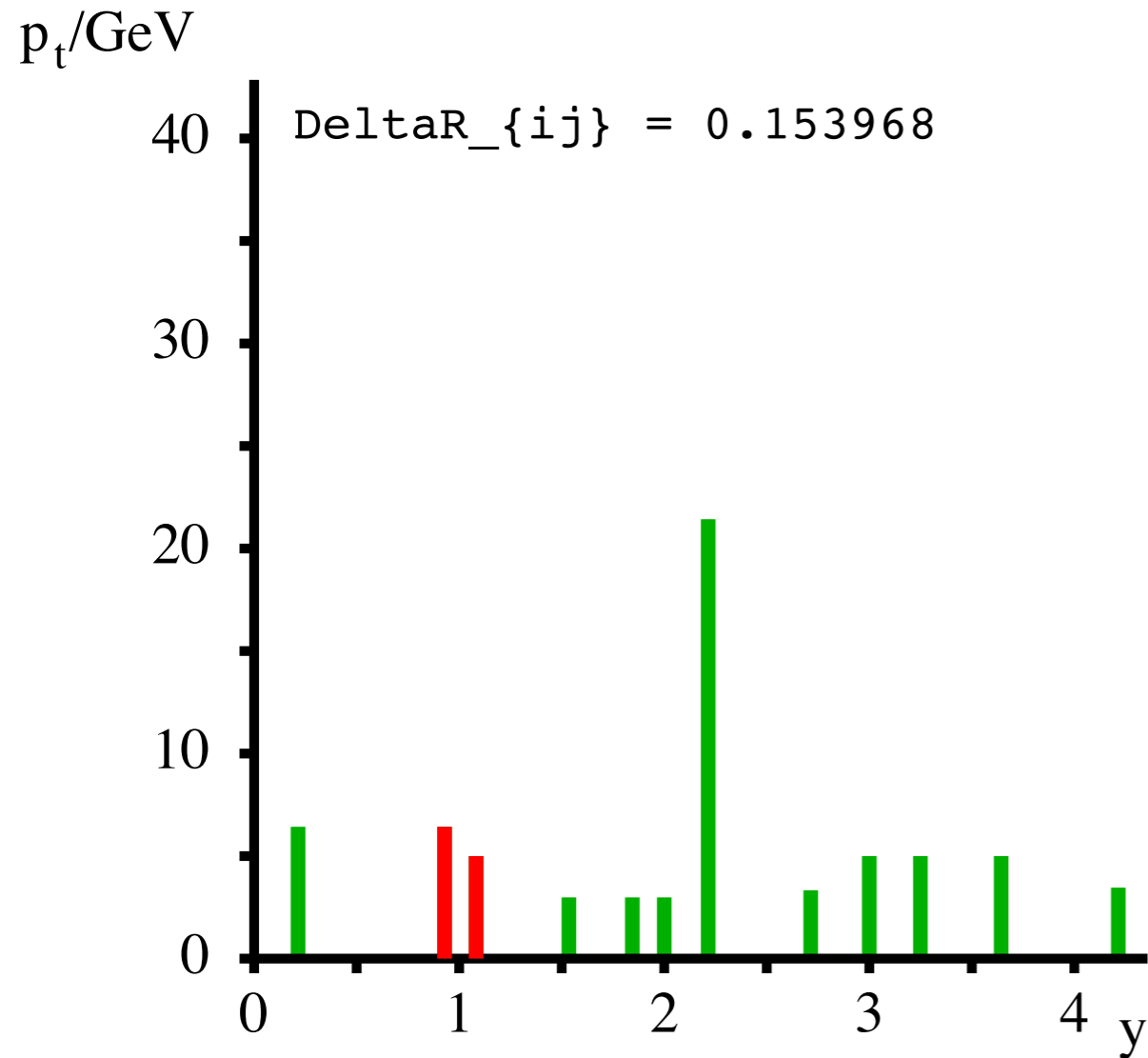


$$R = I$$

- ✓ Infrared and collinear güvenilir.
- ✓ Küme algoritması.
 - ✓ Tüm parçacıkları listele.
 - ✓ d_{ij} and d_{iB} mesafelerini hesapla,
 - ✓ Eğer d_{ij} en küçük ise, i ve j parçacıklarını birleştir ve başa dön
 - ✓ Eğer d_{iB} en küçük ise, son durumu jet olarak belirle
 - ✓ Hiç bir parçacık kalmayıncaya kadar tekrarla.



CA Algoritması



✓ Infrared and collinear güvenilir.

✓ Küme algoritması.

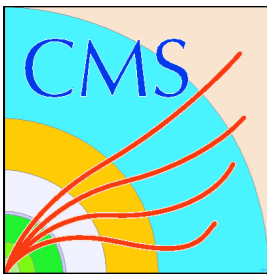
✓ Tüm parçacıkları listele.

✓ d_{ij} and d_{iB} mesafelerini hesapla,

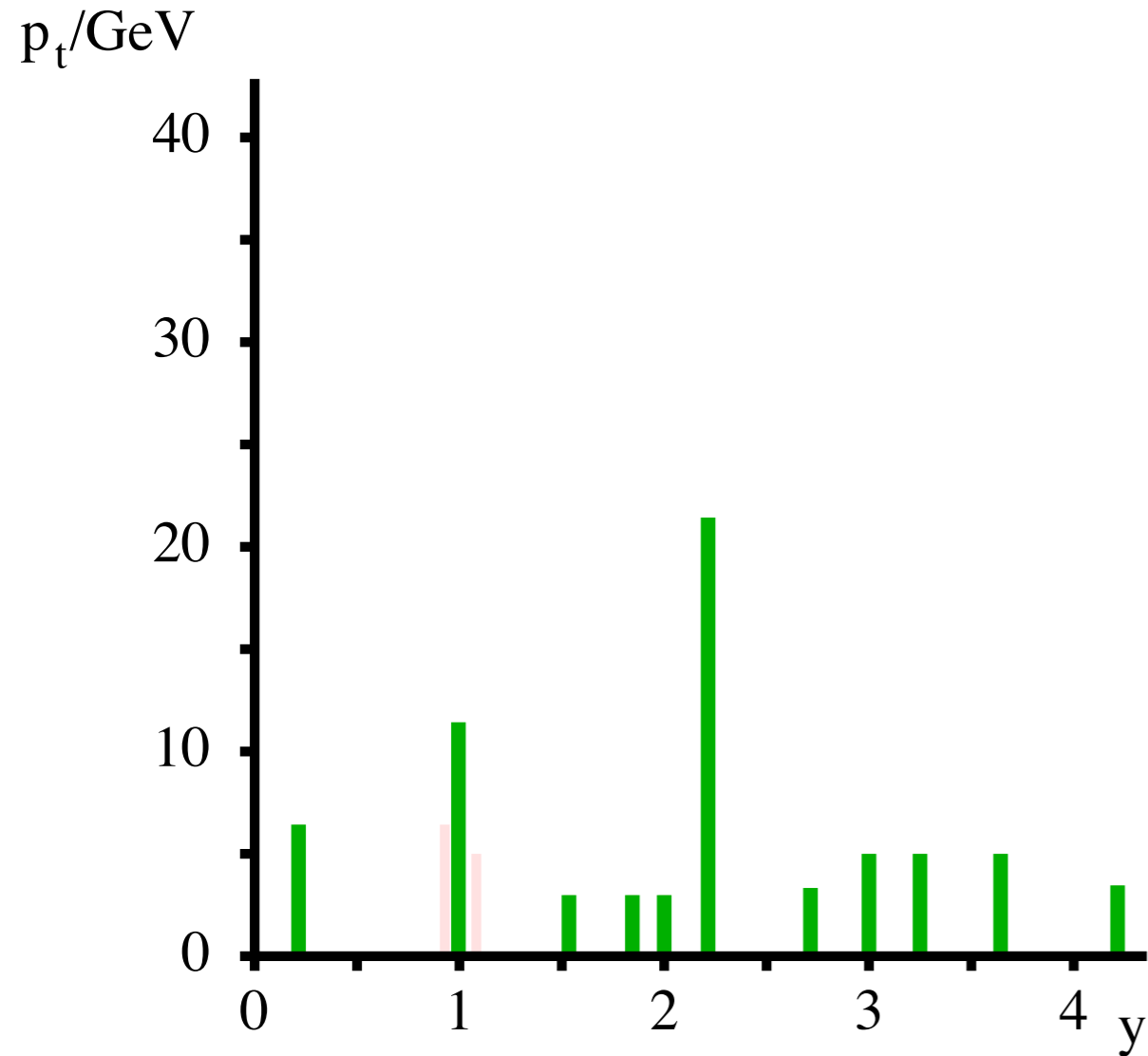
✓ Eğer d_{ij} en küçük ise, i ve j parçacıklarını birleştir ve başa dön

✓ Eğer d_{iB} en küçük ise, son durumu jet olarak belirle

✓ Hiç bir parçacık kalmayıncaya kadar tekrarla.



CA Algoritması



✓ Infrared and collinear güvenilir.

✓ Küme algoritması.

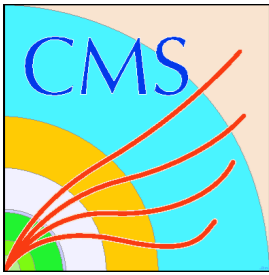
✓ Tüm parçacıkları listele.

✓ d_{ij} and d_{iB} mesafelerini hesapla,

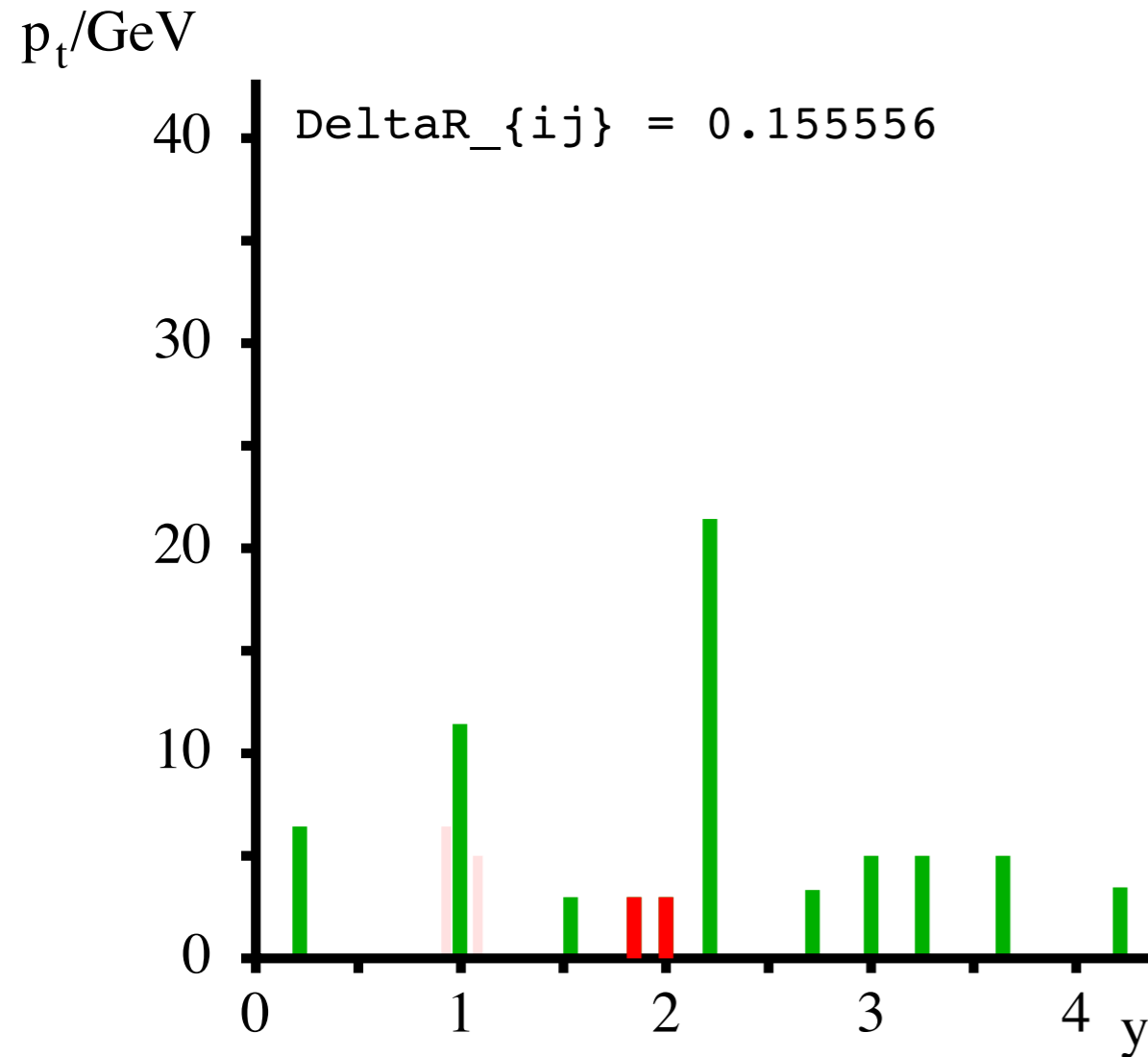
✓ Eğer d_{ij} en küçük ise, i ve j parçacıklarını birleştir ve başa dön

✓ Eğer d_{iB} en küçük ise, son durumu jet olarak belirle

✓ Hiç bir parçacık kalmayıncaya kadar tekrarla.



CA Algoritması



✓ Infrared and collinear güvenilir.

✓ Küme algoritması.

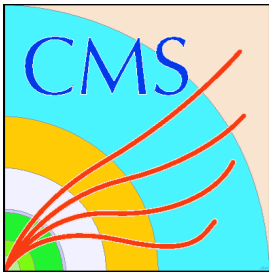
✓ Tüm parçacıkları listele.

✓ d_{ij} and d_{iB} mesafelerini hesapla,

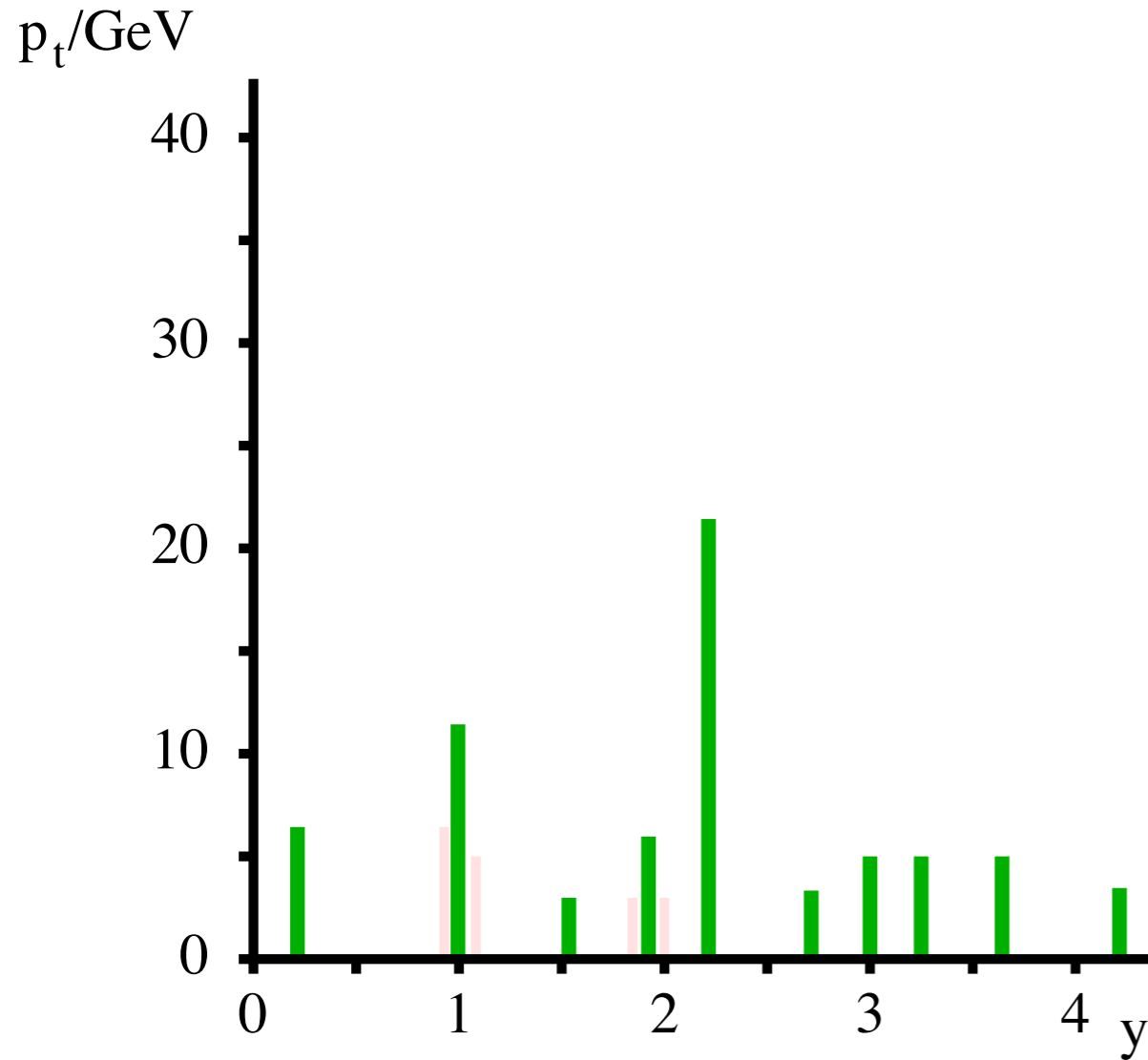
✓ Eğer d_{ij} en küçük ise, i ve j parçacıklarını birleştir ve başa dön

✓ Eğer d_{iB} en küçük ise, son durumu jet olarak belirle

✓ Hiç bir parçacık kalmayıncaya kadar tekrarla.



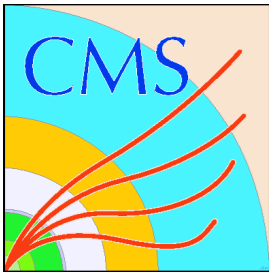
CA Algoritması



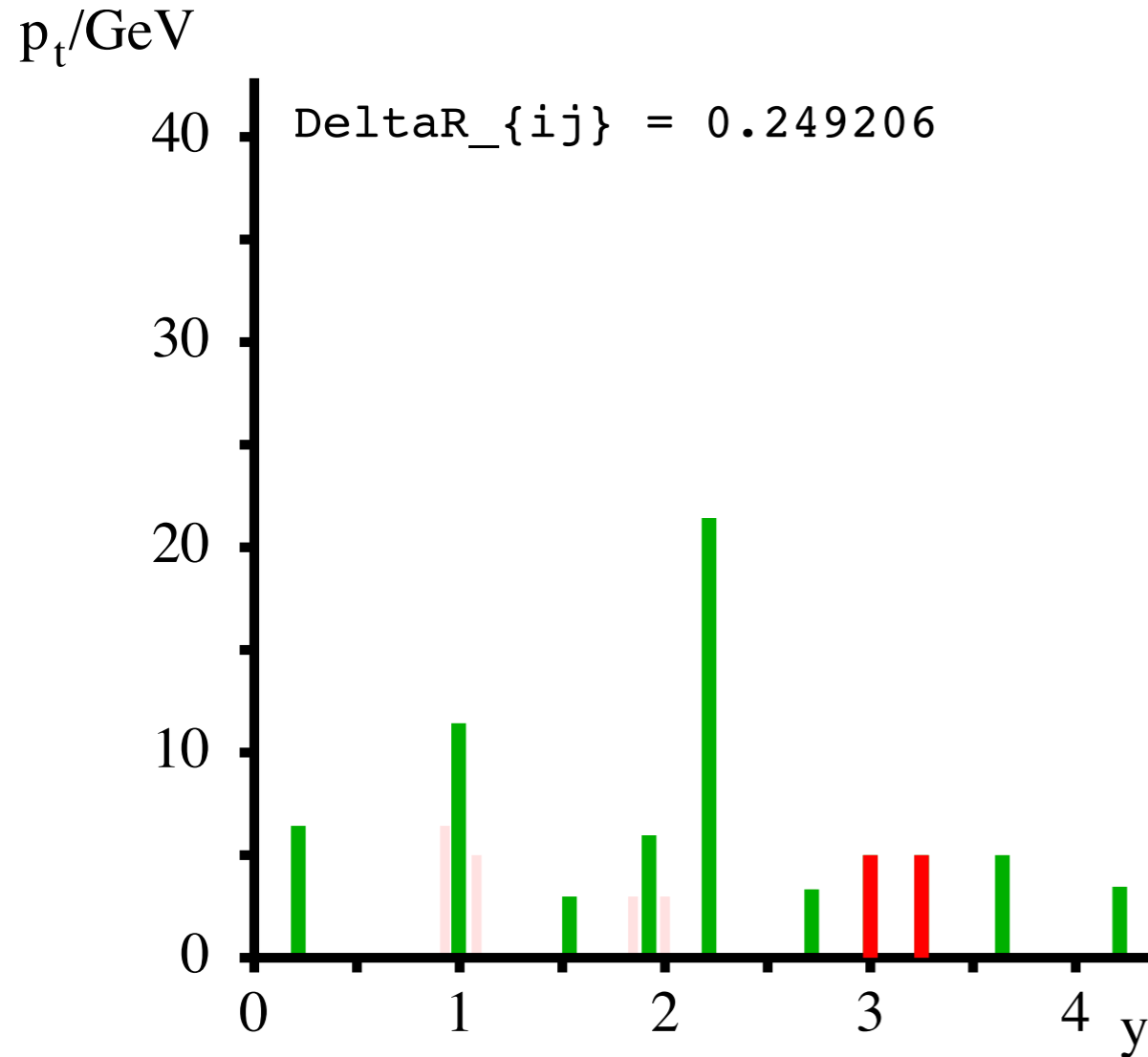
☑ Infrared and collinear güvenilir.

☑ Küme algoritması.

- ✓ Tüm parçacıkları listele.
- ✓ d_{ij} and d_{iB} mesafelerini hesapla,
- ✓ Eğer d_{ij} en küçük ise, i ve j parçacıklarını birleştir ve başa dön
- ✓ Eğer d_{iB} en küçük ise, son durumu jet olarak belirle
- ✓ Hiç bir parçacık kalmayıncaya kadar tekrarla.



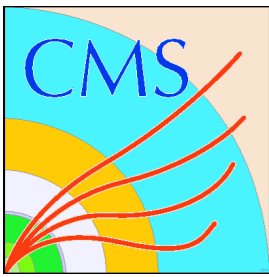
CA Algoritması



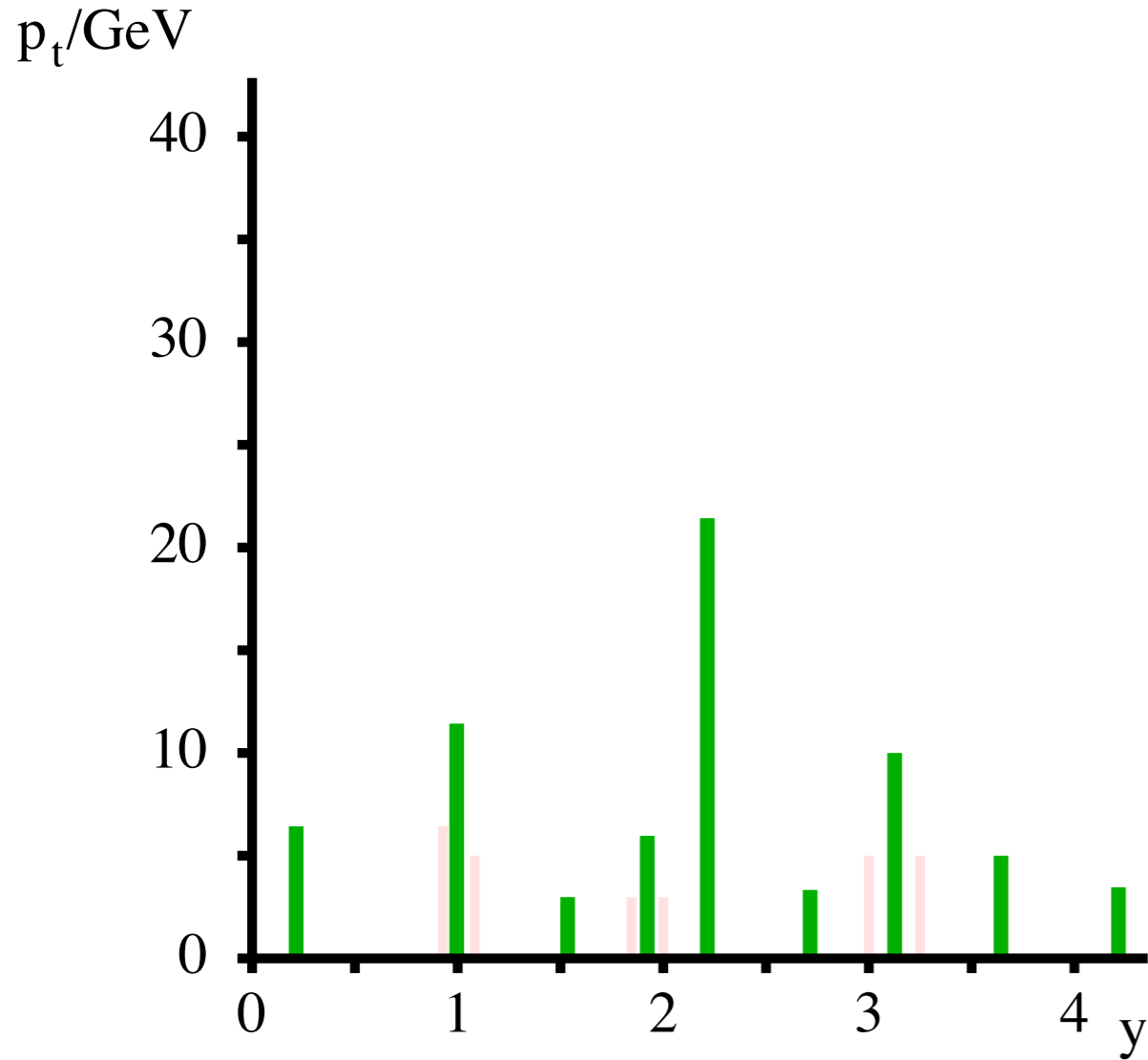
☑ Infrared and collinear güvenilir.

☑ Küme algoritması.

- ✓ Tüm parçacıkları listele.
- ✓ d_{ij} and d_{iB} mesafelerini hesapla,
- ✓ Eğer d_{ij} en küçük ise, i ve j parçacıklarını birleştir ve başa dön
- ✓ Eğer d_{iB} en küçük ise, son durumu jet olarak belirle
- ✓ Hiç bir parçacık kalmayıncaya kadar tekrarla.



CA Algoritması



✓ Infrared and collinear güvenilir.

✓ Küme algoritması.

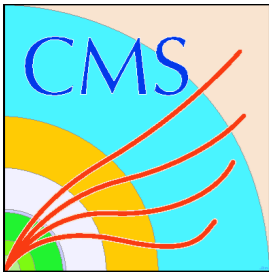
✓ Tüm parçacıkları listele.

✓ d_{ij} and d_{iB} mesafelerini hesapla,

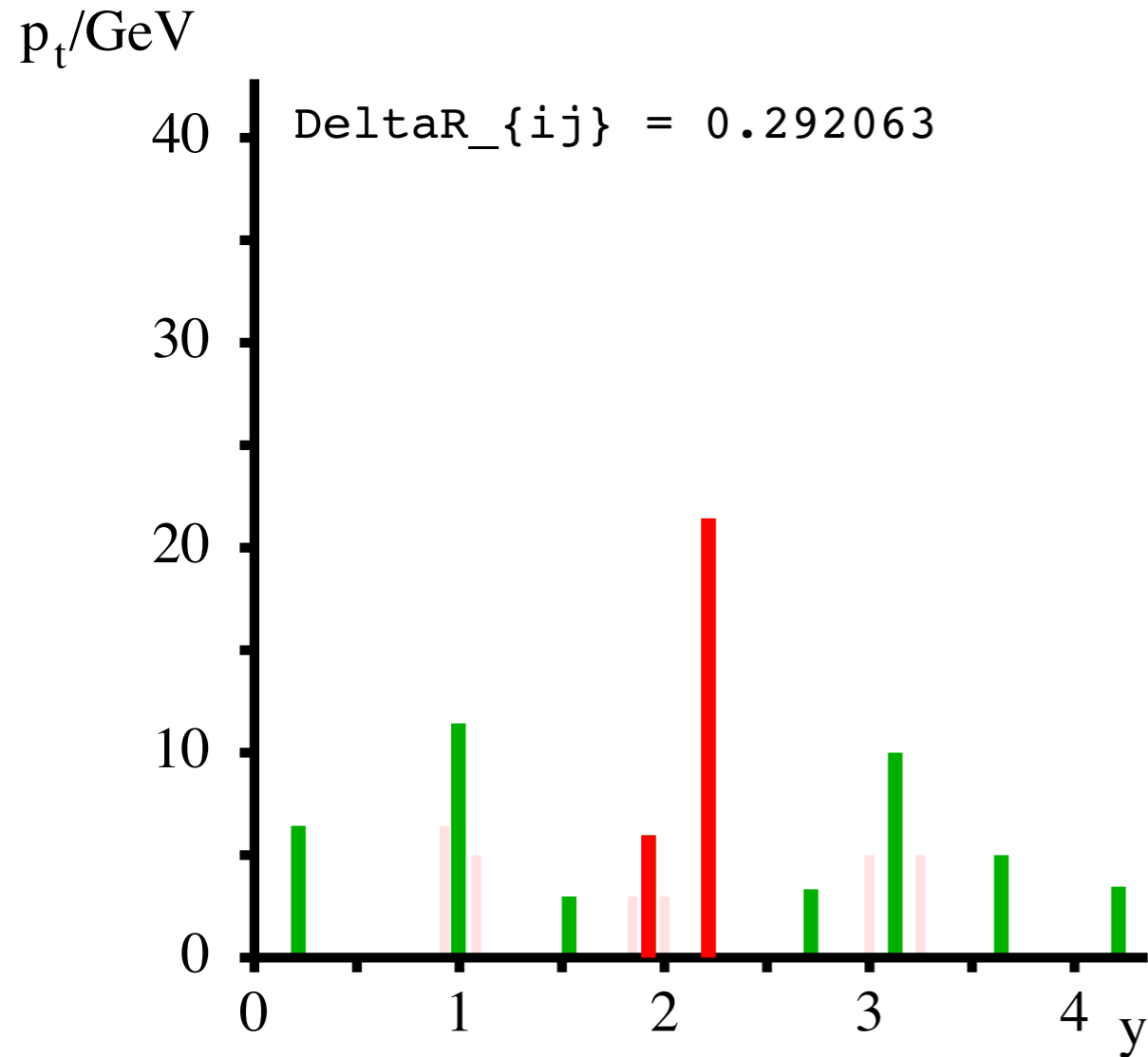
✓ Eğer d_{ij} en küçük ise, i ve j parçacıklarını birleştir ve başa dön

✓ Eğer d_{iB} en küçük ise, son durumu jet olarak belirle

✓ Hiç bir parçacık kalmayıncaya kadar tekrarla.



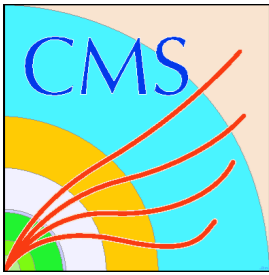
CA Algoritması



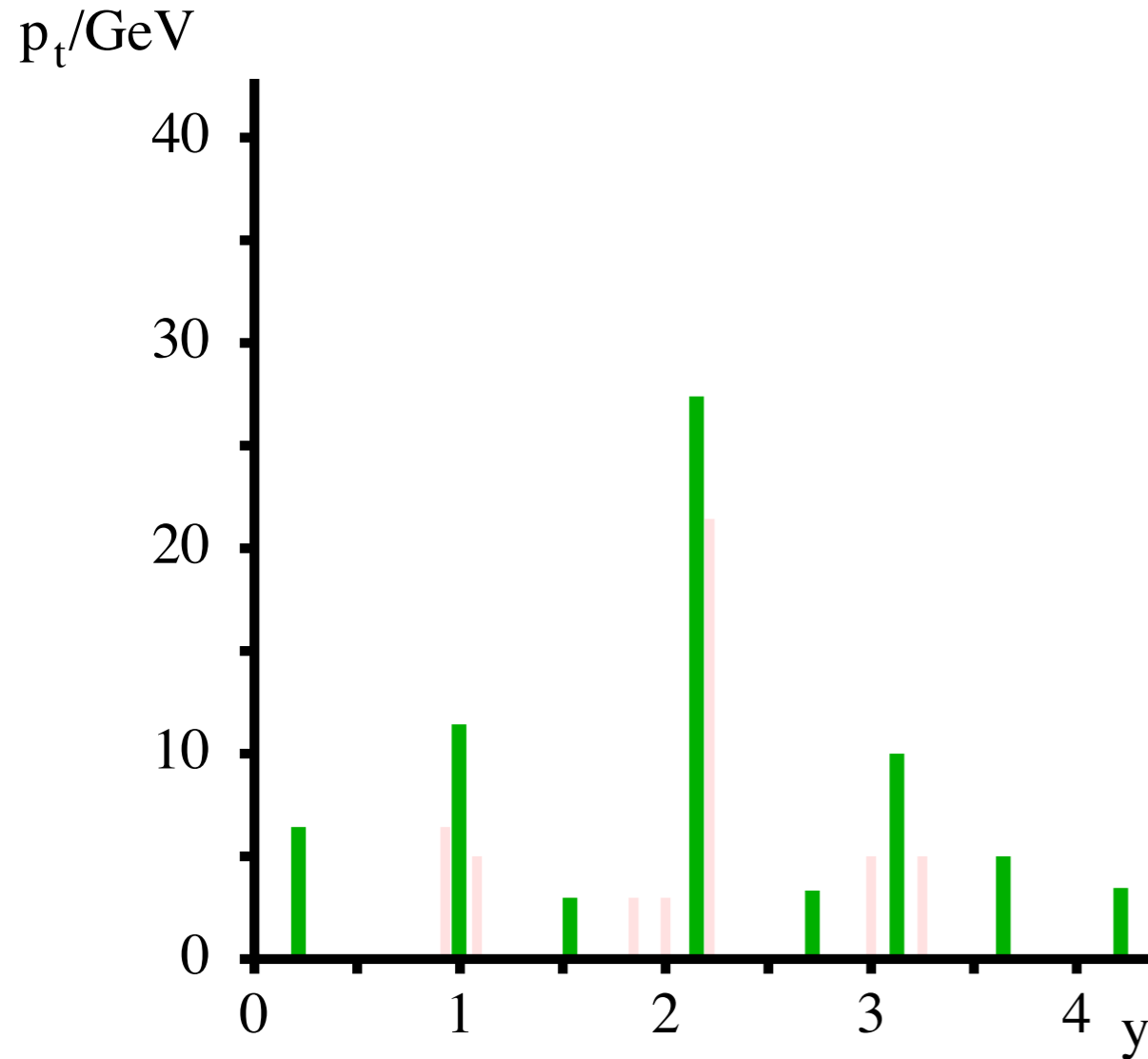
☑ Infrared and collinear güvenilir.

☑ Küme algoritması.

- ✓ Tüm parçacıkları listele.
- ✓ d_{ij} and d_{iB} mesafelerini hesapla,
- ✓ Eğer d_{ij} en küçük ise, i ve j parçacıklarını birleştir ve başa dön
- ✓ Eğer d_{iB} en küçük ise, son durumu jet olarak belirle
- ✓ Hiç bir parçacık kalmayıncaya kadar tekrarla.



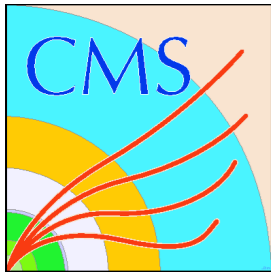
CA Algoritması



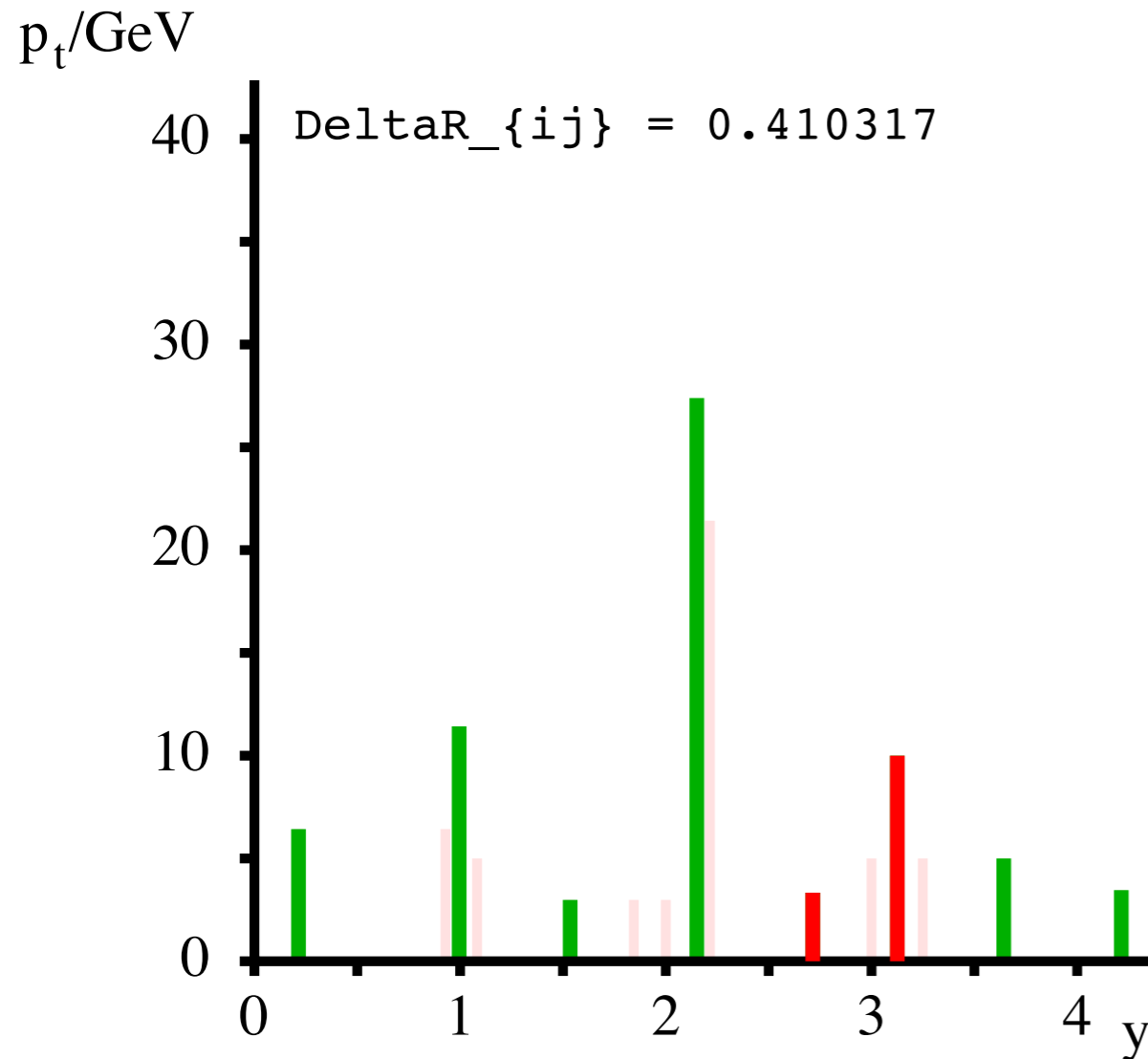
☑ Infrared and collinear güvenilir.

☑ Küme algoritması.

- ✓ Tüm parçacıkları listele.
- ✓ d_{ij} and d_{iB} mesafelerini hesapla,
- ✓ Eğer d_{ij} en küçük ise, i ve j parçacıklarını birleştir ve başa dön
- ✓ Eğer d_{iB} en küçük ise, son durumu jet olarak belirle
- ✓ Hiç bir parçacık kalmayıncaya kadar tekrarla.



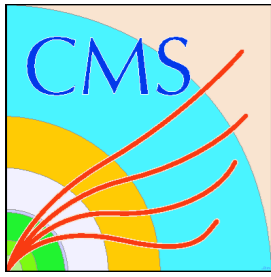
CA Algoritması



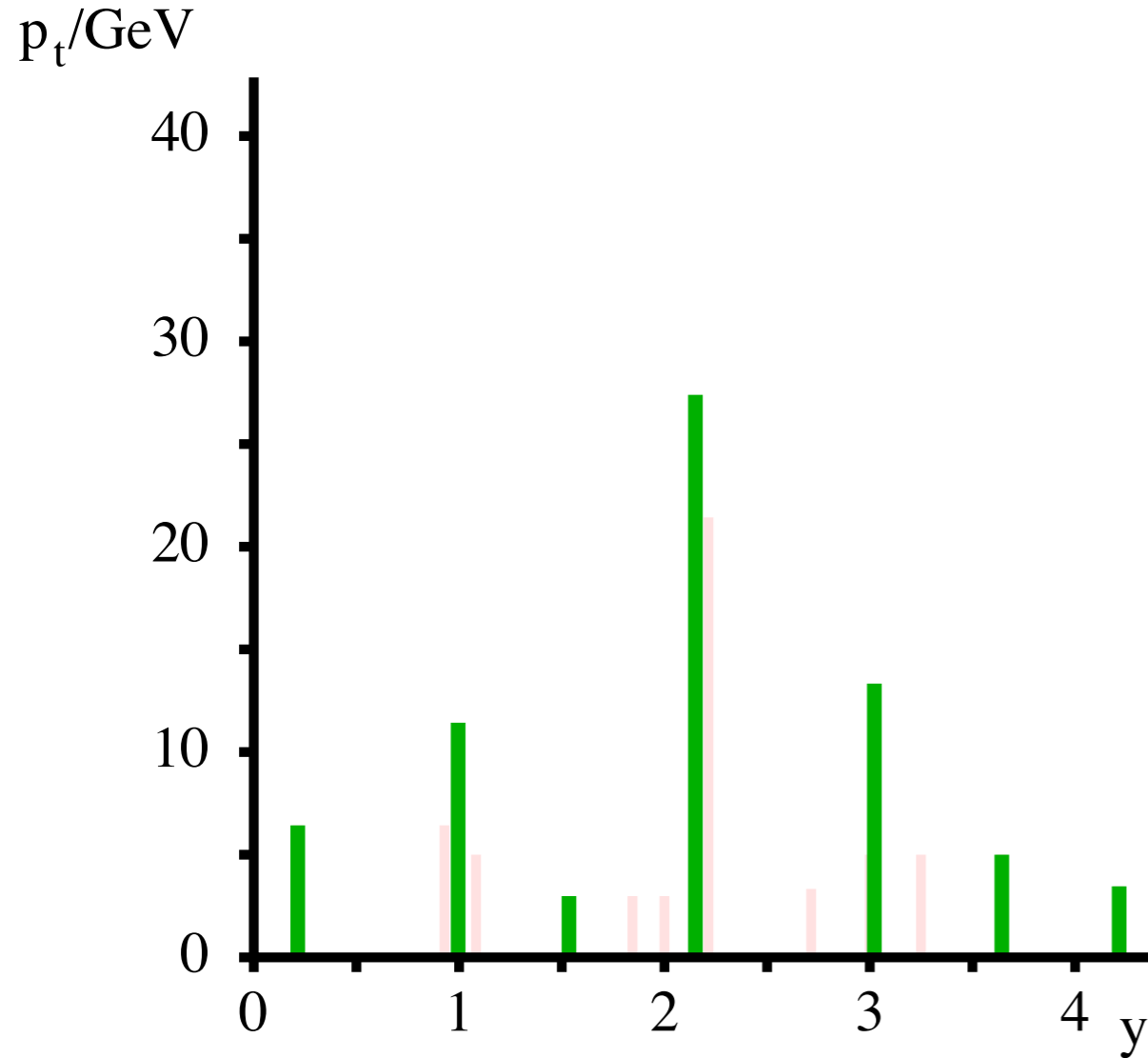
Infrared and collinear güvenilir.

Küme algoritması.

- ✓ Tüm parçacıkları listele.
- ✓ d_{ij} and d_{iB} mesafelerini hesapla,
- ✓ Eğer d_{ij} en küçük ise, i ve j parçacıklarını birleştir ve başa dön
- ✓ Eğer d_{iB} en küçük ise, son durumu jet olarak belirle
- ✓ Hiç bir parçacık kalmayıncaya kadar tekrarla.



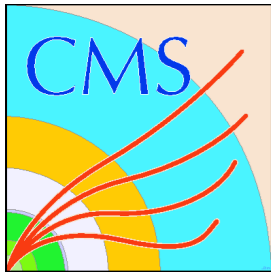
CA Algoritması



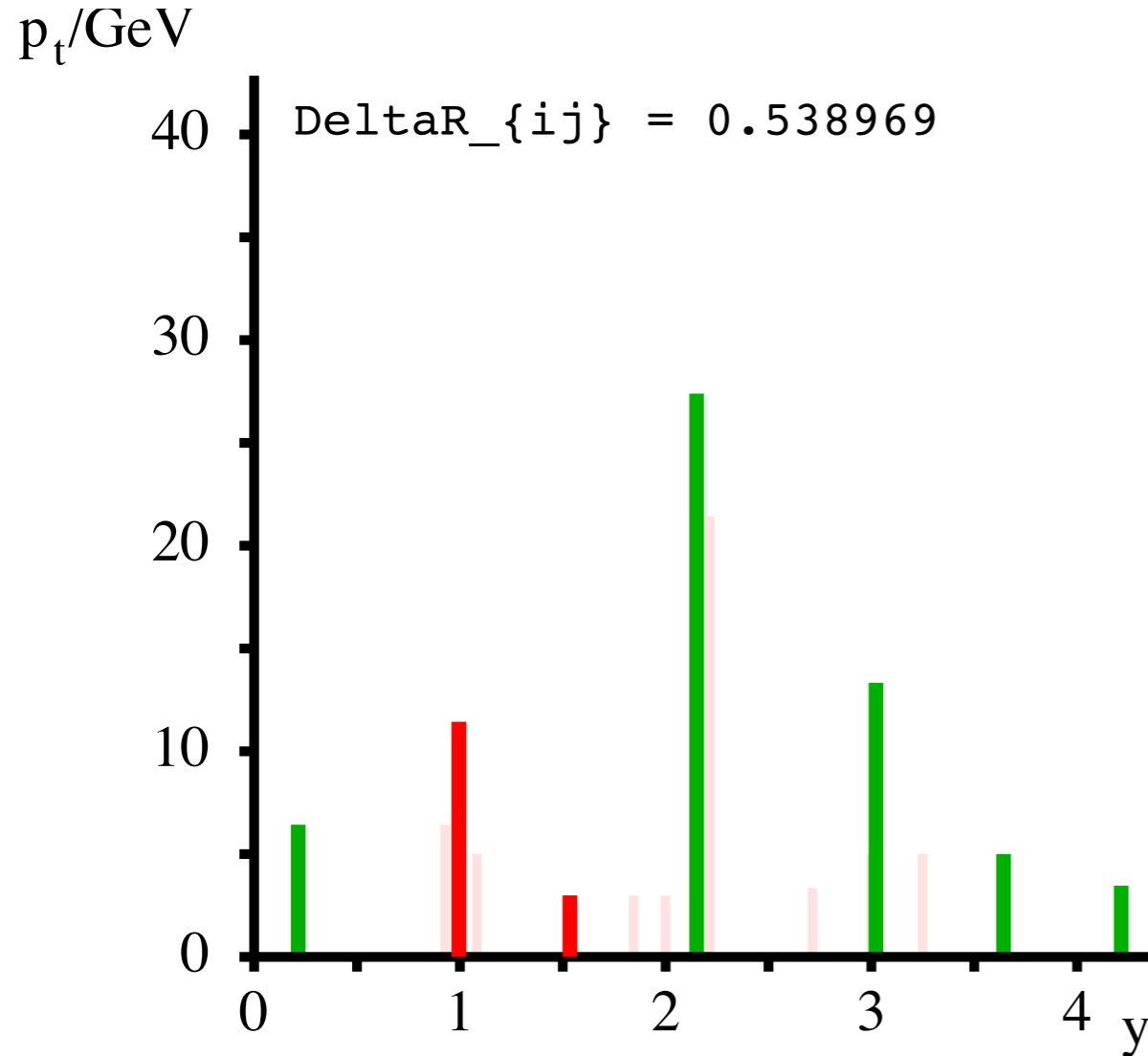
☑ Infrared and collinear güvenilir.

☑ Küme algoritması.

- ✓ Tüm parçacıkları listele.
- ✓ d_{ij} and d_{iB} mesafelerini hesapla,
- ✓ Eğer d_{ij} en küçük ise, i ve j parçacıklarını birleştir ve başa dön
- ✓ Eğer d_{iB} en küçük ise, son durumu jet olarak belirle
- ✓ Hiç bir parçacık kalmayıncaya kadar tekrarla.



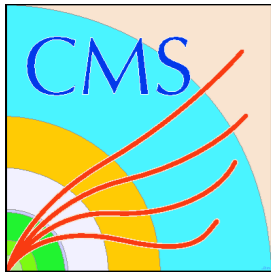
CA Algoritması



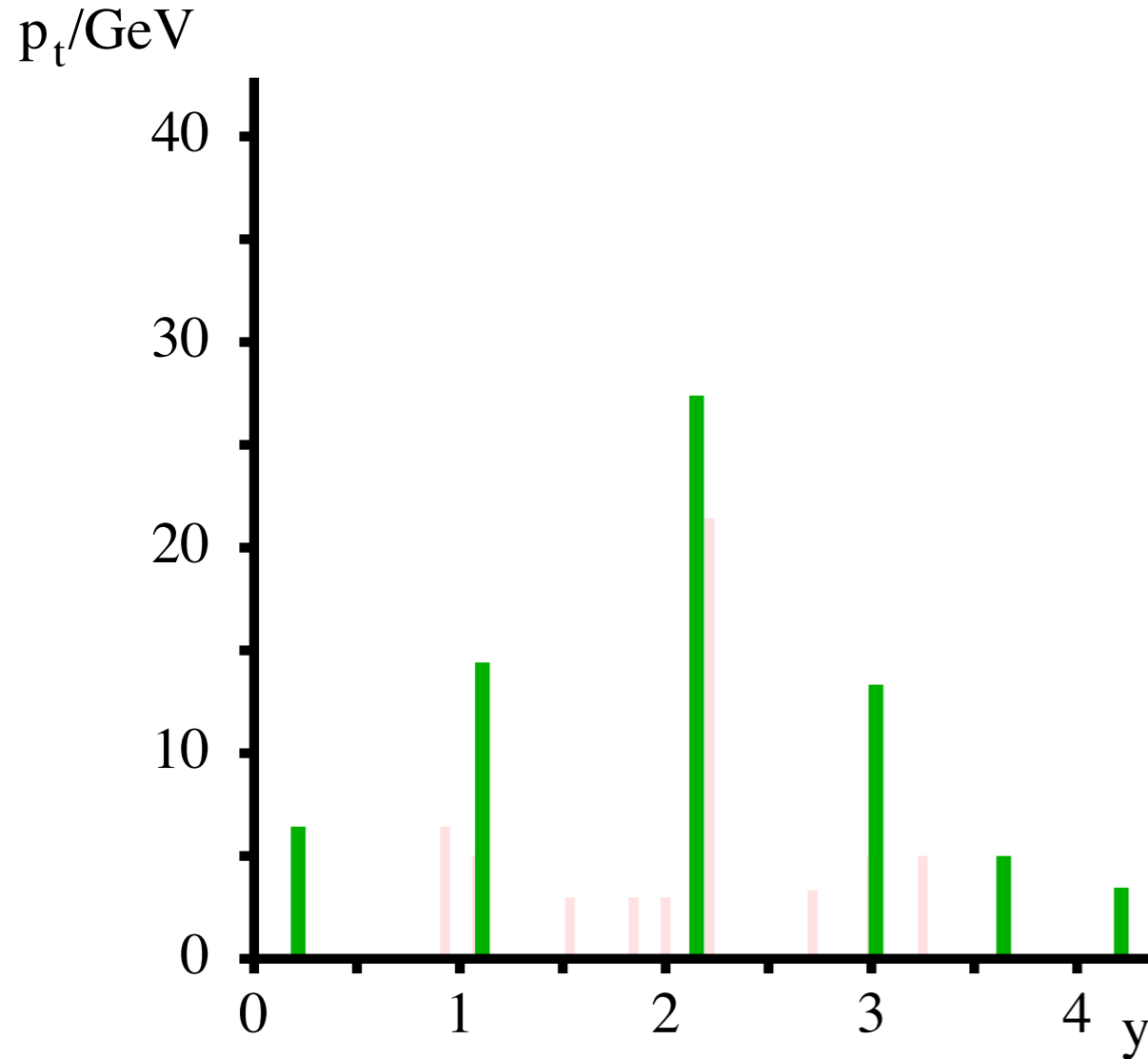
☑ Infrared and collinear güvenilir.

☑ Küme algoritması.

- ✓ Tüm parçacıkları listele.
- ✓ d_{ij} and d_{iB} mesafelerini hesapla,
- ✓ Eğer d_{ij} en küçük ise, i ve j parçacıklarını birleştir ve başa dön
- ✓ Eğer d_{iB} en küçük ise, son durumu jet olarak belirle
- ✓ Hiç bir parçacık kalmayıncaya kadar tekrarla.



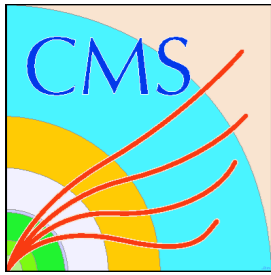
CA Algoritması



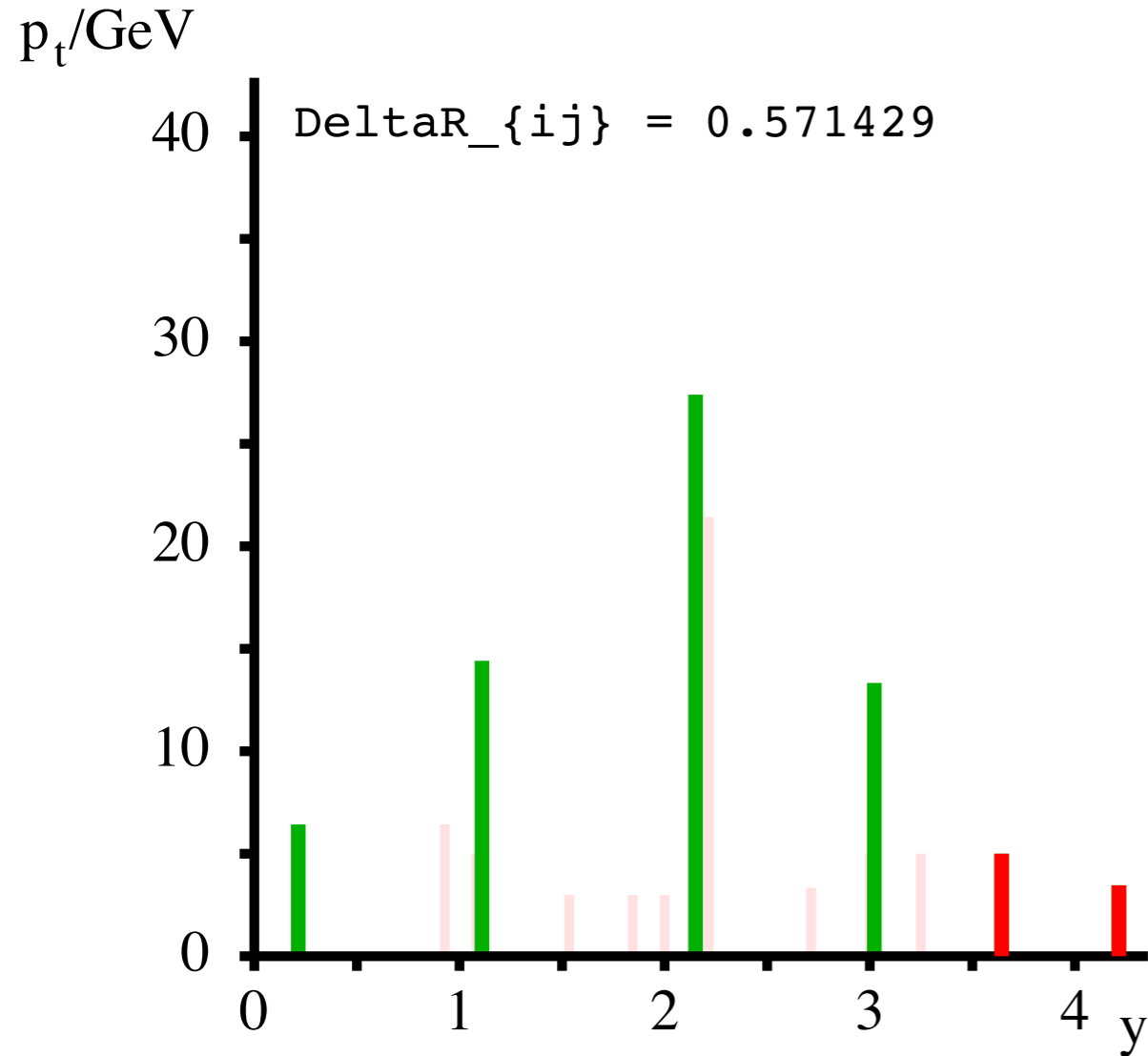
☑ Infrared and collinear güvenilir.

☑ Küme algoritması.

- ✓ Tüm parçacıkları listele.
- ✓ d_{ij} and d_{iB} mesafelerini hesapla,
- ✓ Eğer d_{ij} en küçük ise, i ve j parçacıklarını birleştir ve başa dön
- ✓ Eğer d_{iB} en küçük ise, son durumu jet olarak belirle
- ✓ Hiç bir parçacık kalmayıncaya kadar tekrarla.



CA Algoritması



✓ Infrared and collinear güvenilir.

✓ Küme algoritması.

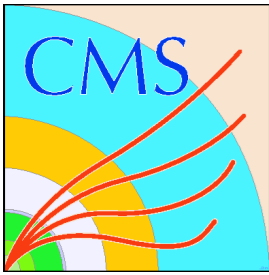
✓ Tüm parçacıkları listele.

✓ d_{ij} and d_{iB} mesafelerini hesapla,

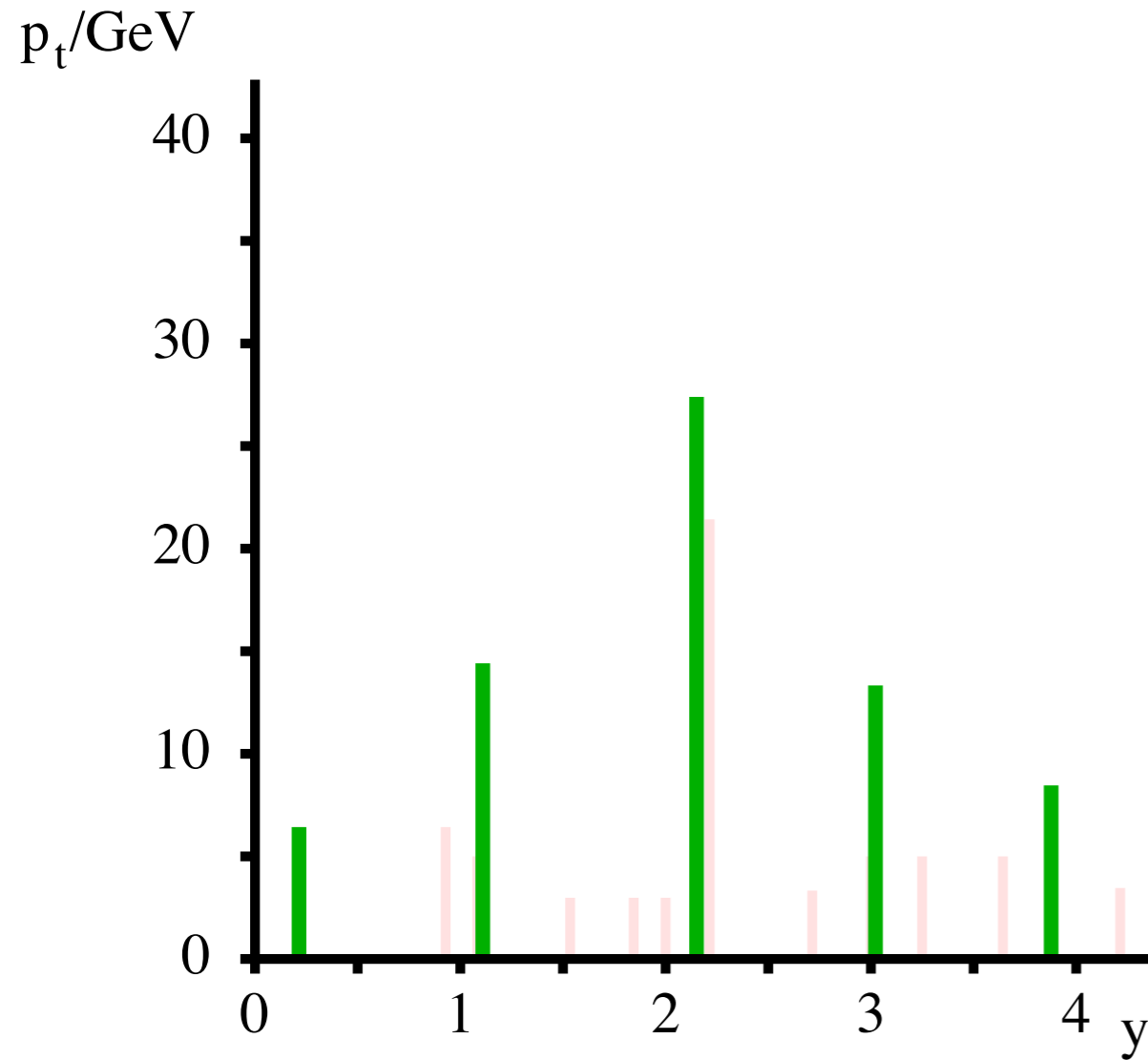
✓ Eğer d_{ij} en küçük ise, i ve j parçacıklarını birleştir ve başa dön

✓ Eğer d_{iB} en küçük ise, son durumu jet olarak belirle

✓ Hiç bir parçacık kalmayıncaya kadar tekrarla.



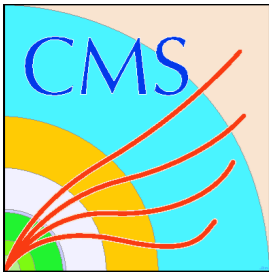
CA Algoritması



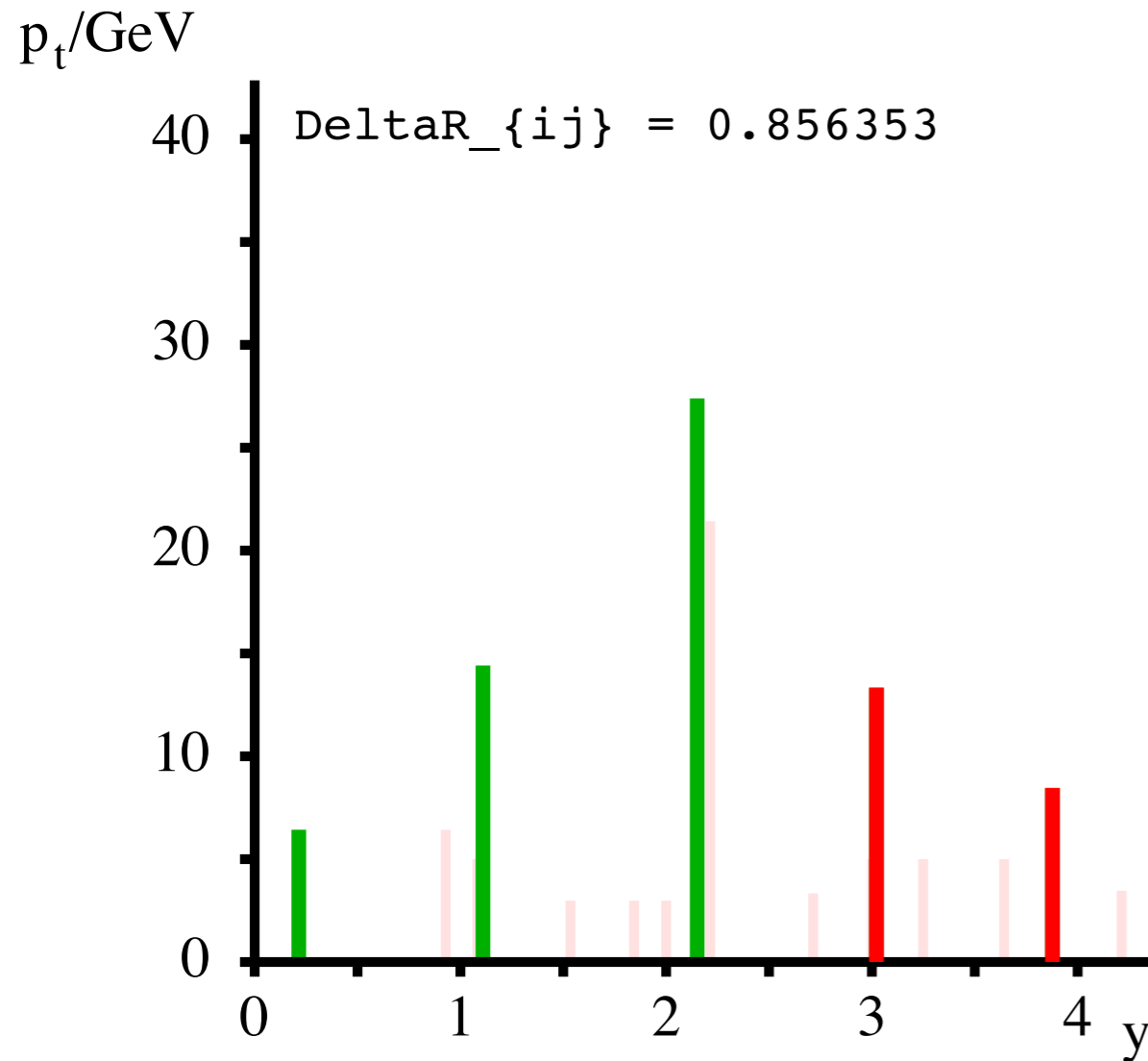
☑ Infrared and collinear güvenilir.

☑ Küme algoritması.

- ✓ Tüm parçacıkları listele.
- ✓ d_{ij} and d_{iB} mesafelerini hesapla,
- ✓ Eğer d_{ij} en küçük ise, i ve j parçacıklarını birleştir ve başa dön
- ✓ Eğer d_{iB} en küçük ise, son durumu jet olarak belirle
- ✓ Hiç bir parçacık kalmayıncaya kadar tekrarla.



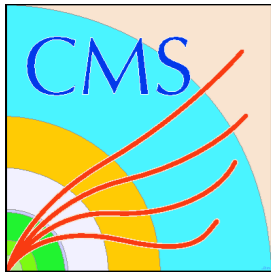
CA Algoritması



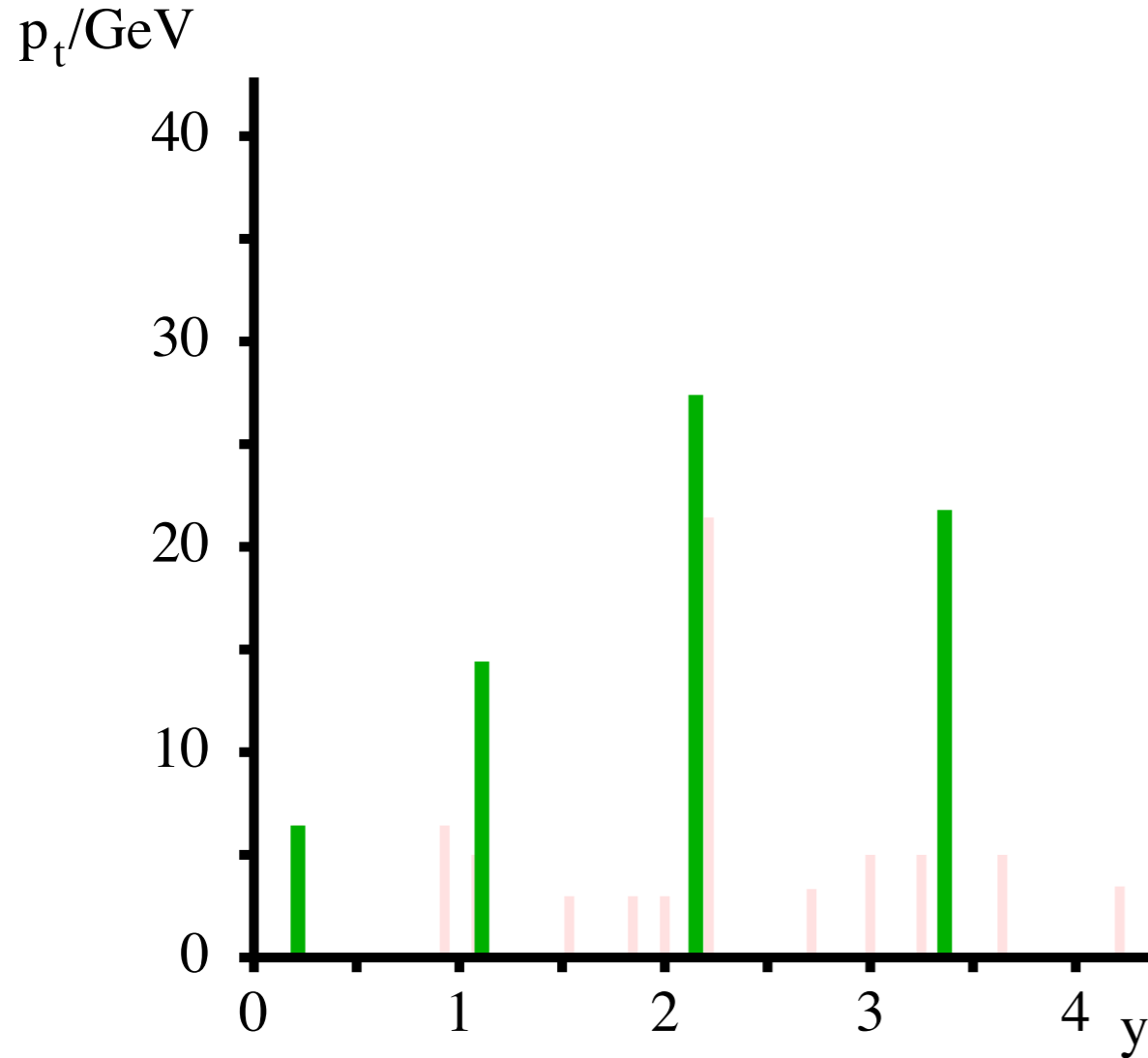
☑ Infrared and collinear güvenilir.

☑ Küme algoritması.

- ✓ Tüm parçacıkları listele.
- ✓ d_{ij} and d_{iB} mesafelerini hesapla,
- ✓ Eğer d_{ij} en küçük ise, i ve j parçacıklarını birleştir ve başa dön
- ✓ Eğer d_{iB} en küçük ise, son durumu jet olarak belirle
- ✓ Hiç bir parçacık kalmayıncaya kadar tekrarla.



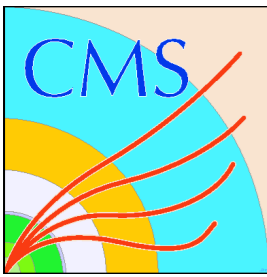
CA Algoritması



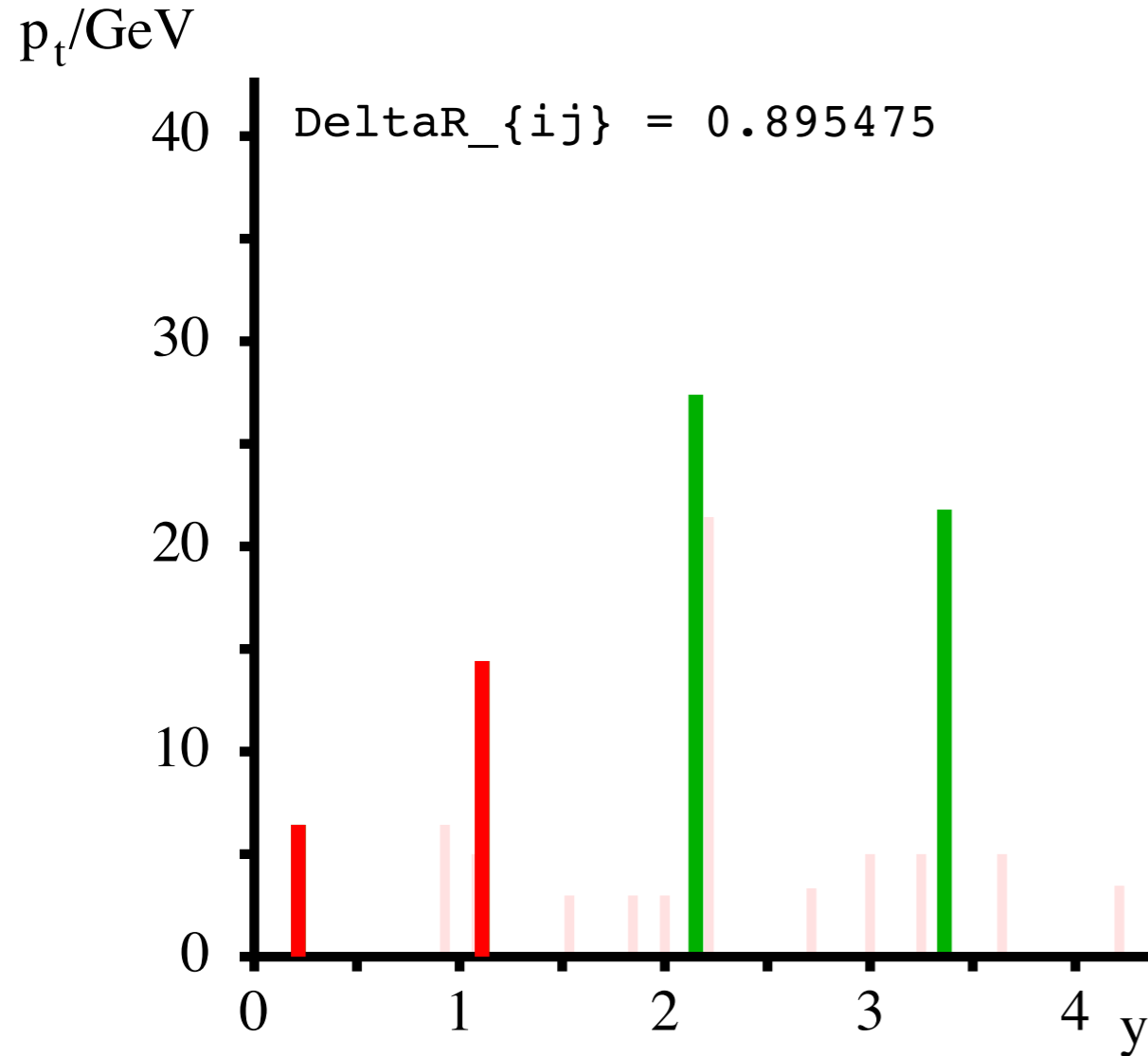
☑ Infrared and collinear güvenilir.

☑ Küme algoritması.

- ✓ Tüm parçacıkları listele.
- ✓ d_{ij} and d_{iB} mesafelerini hesapla,
- ✓ Eğer d_{ij} en küçük ise, i ve j parçacıklarını birleştir ve başa dön
- ✓ Eğer d_{iB} en küçük ise, son durumu jet olarak belirle
- ✓ Hiç bir parçacık kalmayıncaya kadar tekrarla.



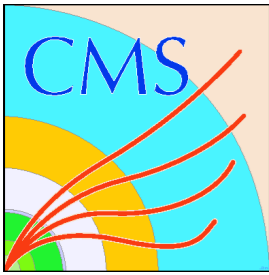
CA Algoritması



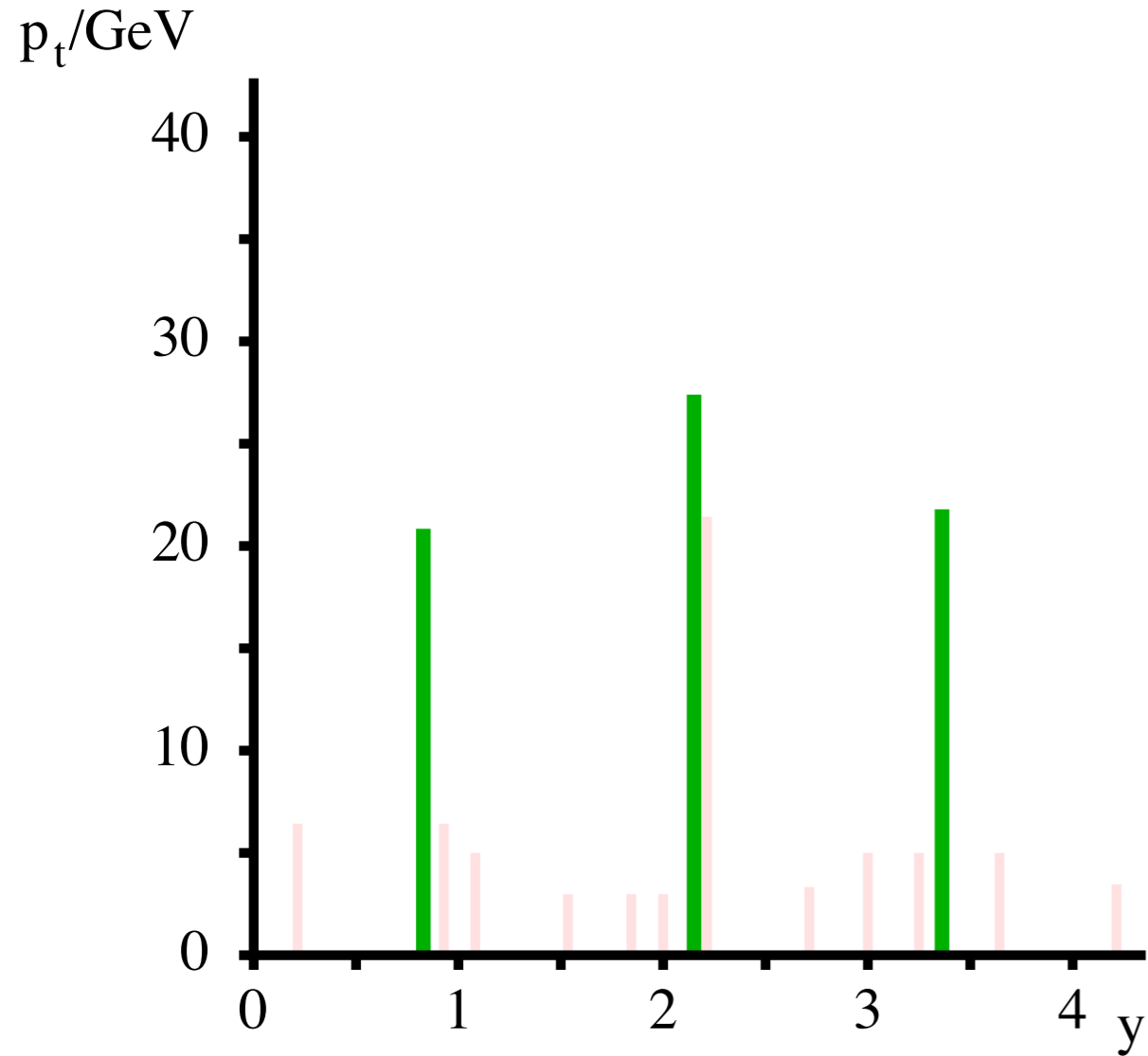
Infrared and collinear güvenilir.

Küme algoritması.

- ✓ Tüm parçacıkları listele.
- ✓ d_{ij} and d_{iB} mesafelerini hesapla,
- ✓ Eğer d_{ij} en küçük ise, i ve j parçacıklarını birleştir ve başa dön
- ✓ Eğer d_{iB} en küçük ise, son durumu jet olarak belirle
- ✓ Hiç bir parçacık kalmayınca kadar tekrarla.



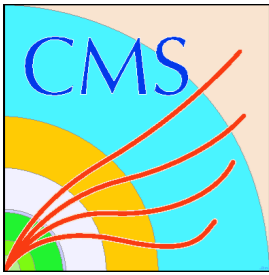
CA Algoritması



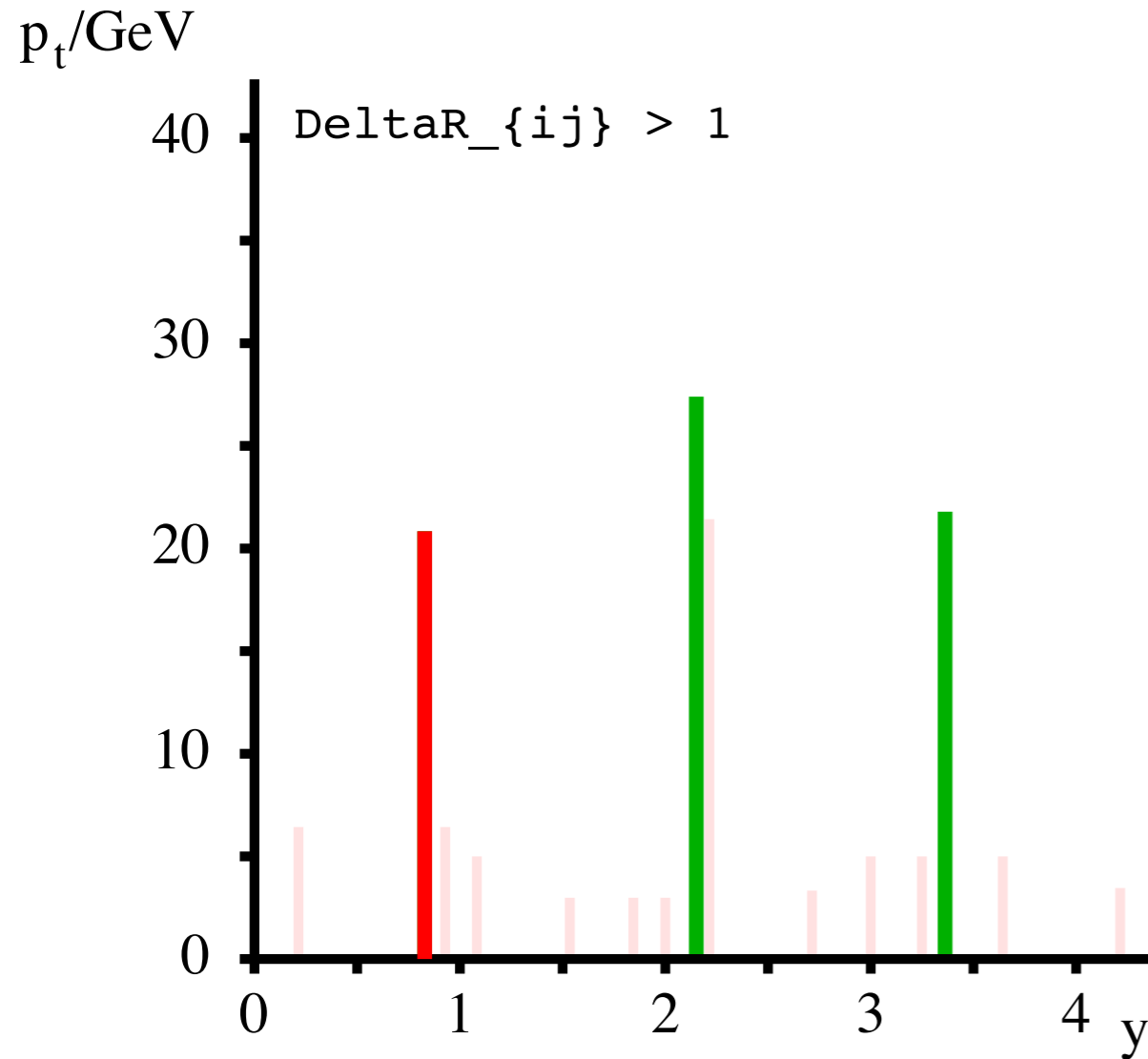
☑ Infrared and collinear güvenilir.

☑ Küme algoritması.

- ✓ Tüm parçacıkları listele.
- ✓ d_{ij} and d_{iB} mesafelerini hesapla,
- ✓ Eğer d_{ij} en küçük ise, i ve j parçacıklarını birleştir ve başa dön
- ✓ Eğer d_{iB} en küçük ise, son durumu jet olarak belirle
- ✓ Hiç bir parçacık kalmayıncaya kadar tekrarla.



CA Algoritması



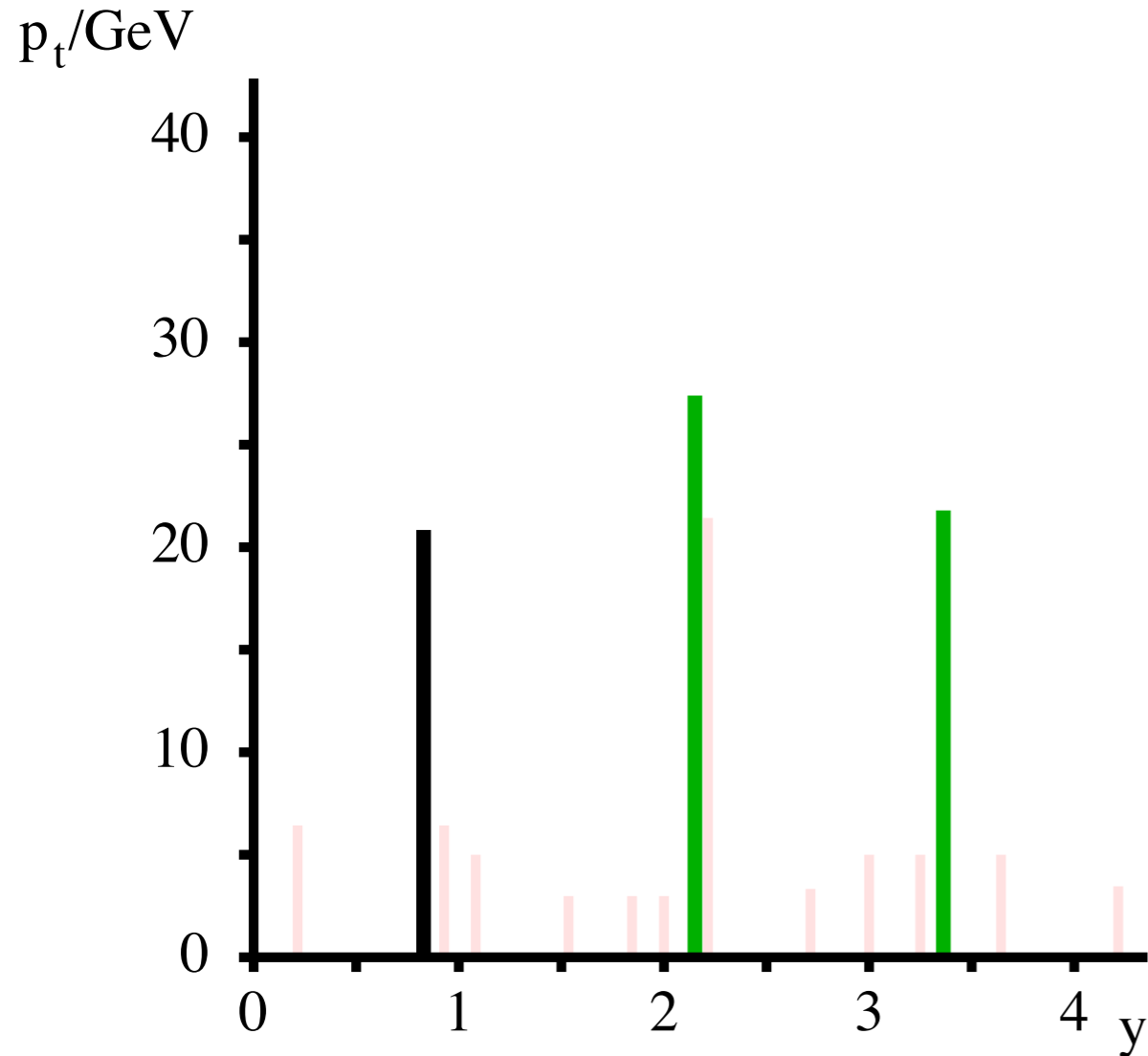
☑ Infrared and collinear güvenilir.

☑ Küme algoritması.

- ✓ Tüm parçacıkları listele.
- ✓ d_{ij} and d_{iB} mesafelerini hesapla,
- ✓ Eğer d_{ij} en küçük ise, i ve j parçacıklarını birleştir ve başa dön
- ✓ Eğer d_{iB} en küçük ise, son durumu jet olarak belirle
- ✓ Hiç bir parçacık kalmayınca kadar tekrarla.



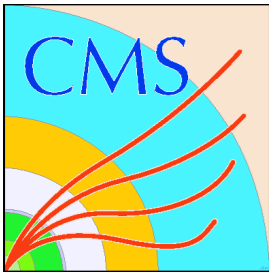
CA Algoritması



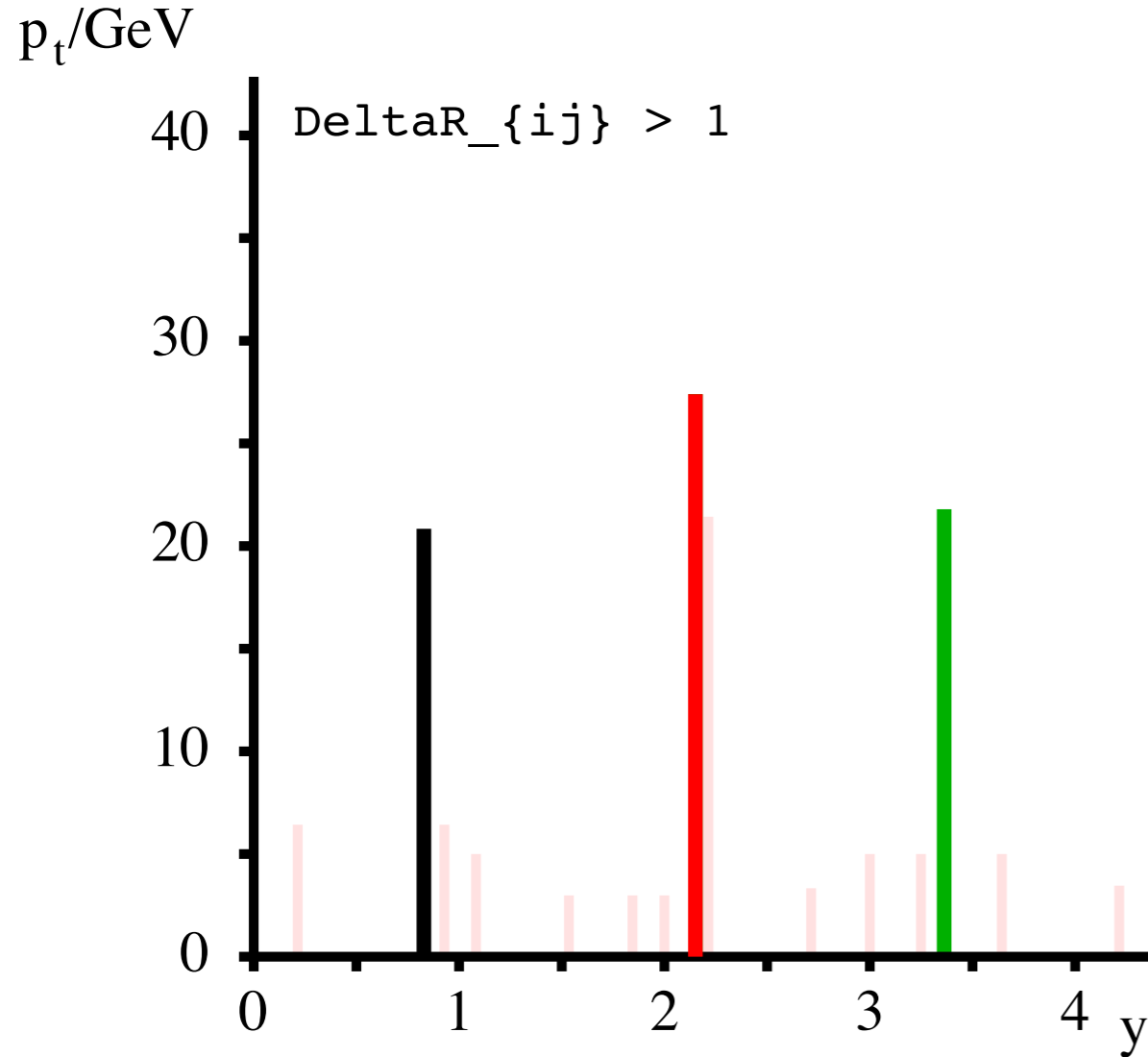
☑ Infrared and collinear güvenilir.

☑ Küme algoritması.

- ✓ Tüm parçacıkları listele.
- ✓ d_{ij} and d_{iB} mesafelerini hesapla,
- ✓ Eğer d_{ij} en küçük ise, i ve j parçacıklarını birleştir ve başa dön
- ✓ Eğer d_{iB} en küçük ise, son durumu jet olarak belirle
- ✓ Hiç bir parçacık kalmayıncaya kadar tekrarla.



CA Algoritması



✓ Infrared and collinear güvenilir.

✓ Küme algoritması.

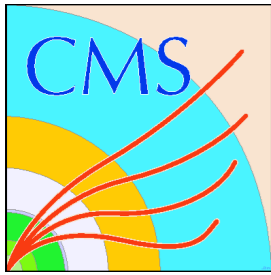
✓ Tüm parçacıkları listele.

✓ d_{ij} and d_{iB} mesafelerini hesapla,

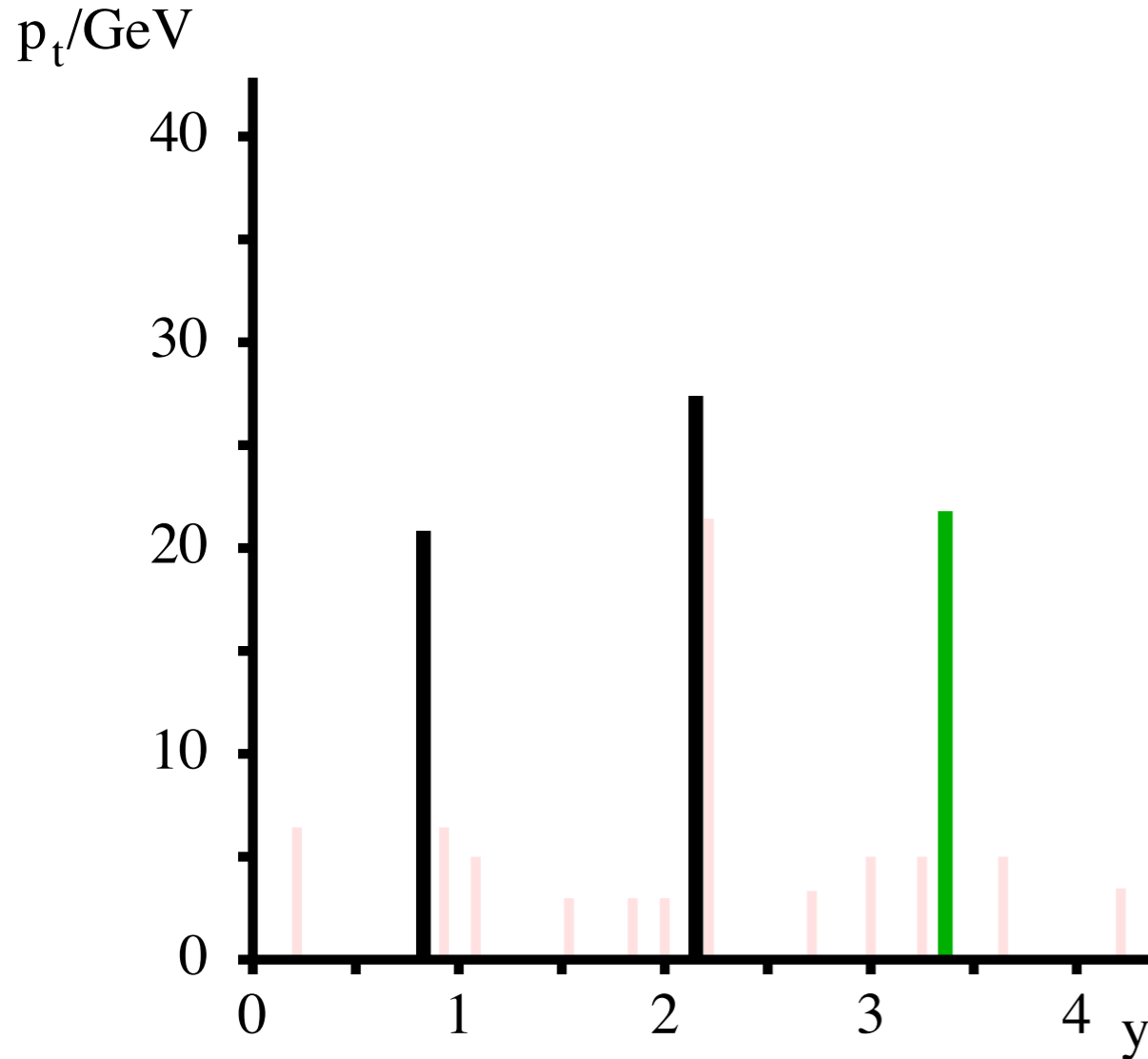
✓ Eğer d_{ij} en küçük ise, i ve j parçacıklarını birleştir ve başa dön

✓ Eğer d_{iB} en küçük ise, son durumu jet olarak belirle

✓ Hiç bir parçacık kalmayıncaya kadar tekrarla.



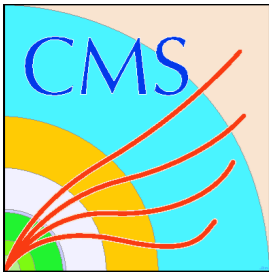
CA Algoritması



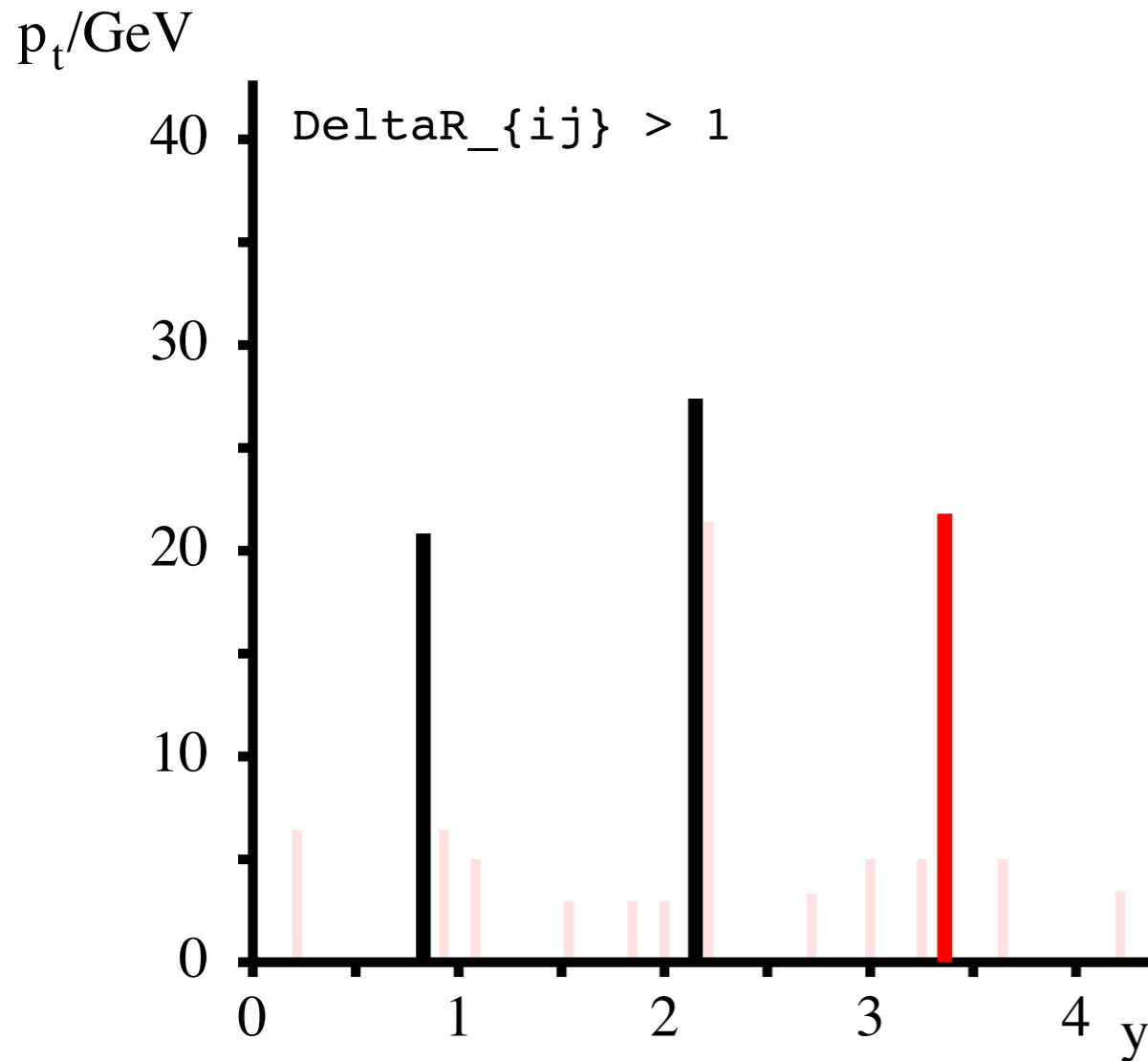
☑ Infrared and collinear güvenilir.

☑ Küme algoritması.

- ✓ Tüm parçacıkları listele.
- ✓ d_{ij} and d_{iB} mesafelerini hesapla,
- ✓ Eğer d_{ij} en küçük ise, i ve j parçacıklarını birleştir ve başa dön
- ✓ Eğer d_{iB} en küçük ise, son durumu jet olarak belirle
- ✓ Hiç bir parçacık kalmayıncaya kadar tekrarla.



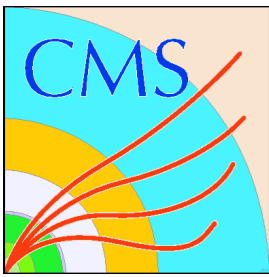
CA Algoritması



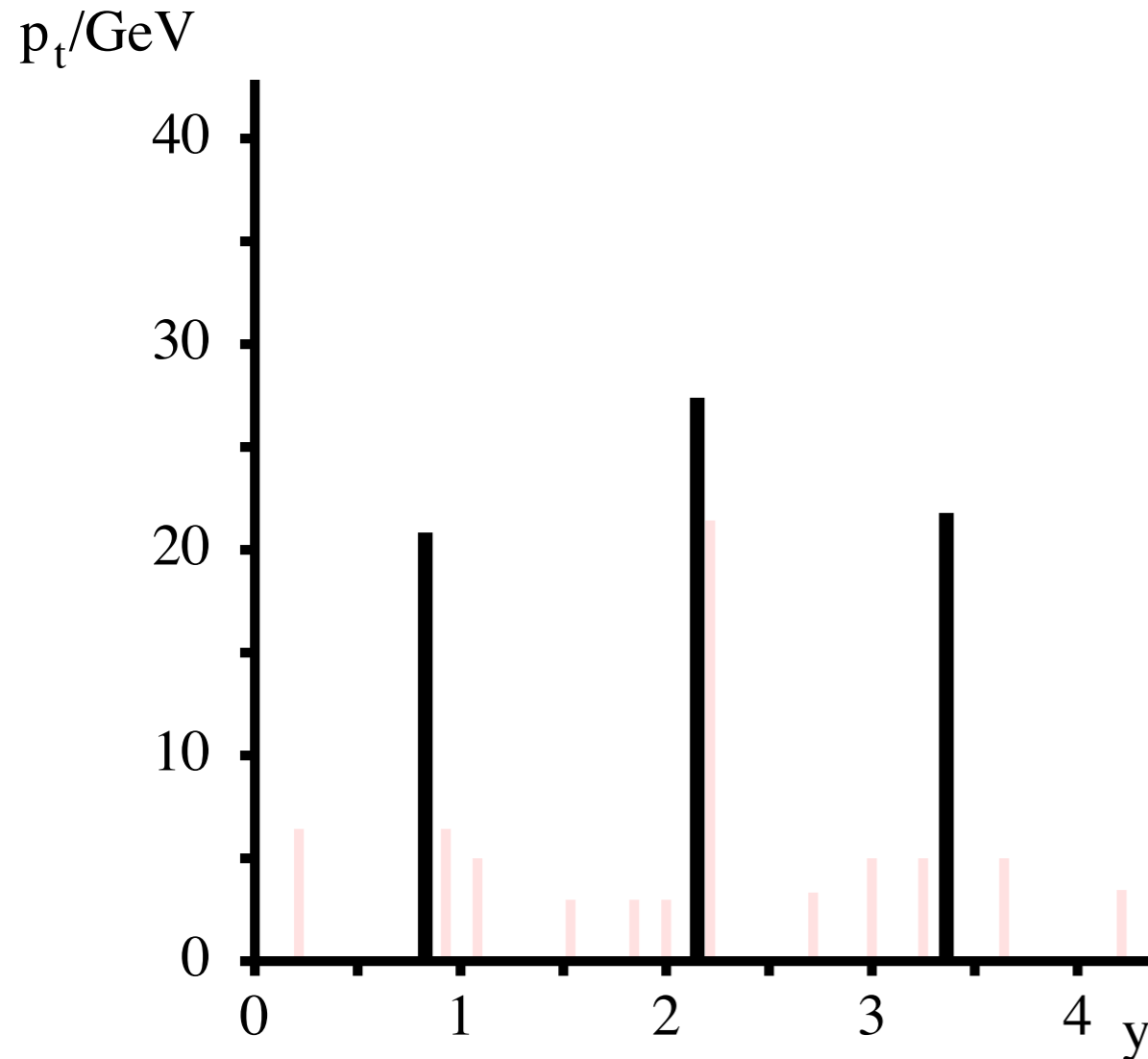
☑ Infrared and collinear güvenilir.

☑ Küme algoritması.

- ✓ Tüm parçacıkları listele.
- ✓ d_{ij} and d_{iB} mesafelerini hesapla,
- ✓ Eğer d_{ij} en küçük ise, i ve j parçacıklarını birleştir ve başa dön
- ✓ Eğer d_{iB} en küçük ise, son durumu jet olarak belirle
- ✓ Hiç bir parçacık kalmayıncaya kadar tekrarla.



CA Algoritması



✓ Infrared and collinear güvenilir.

✓ Küme algoritması.

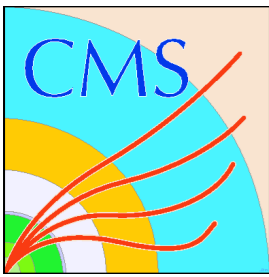
✓ Tüm parçacıkları listele.

✓ d_{ij} and d_{iB} mesafelerini hesapla,

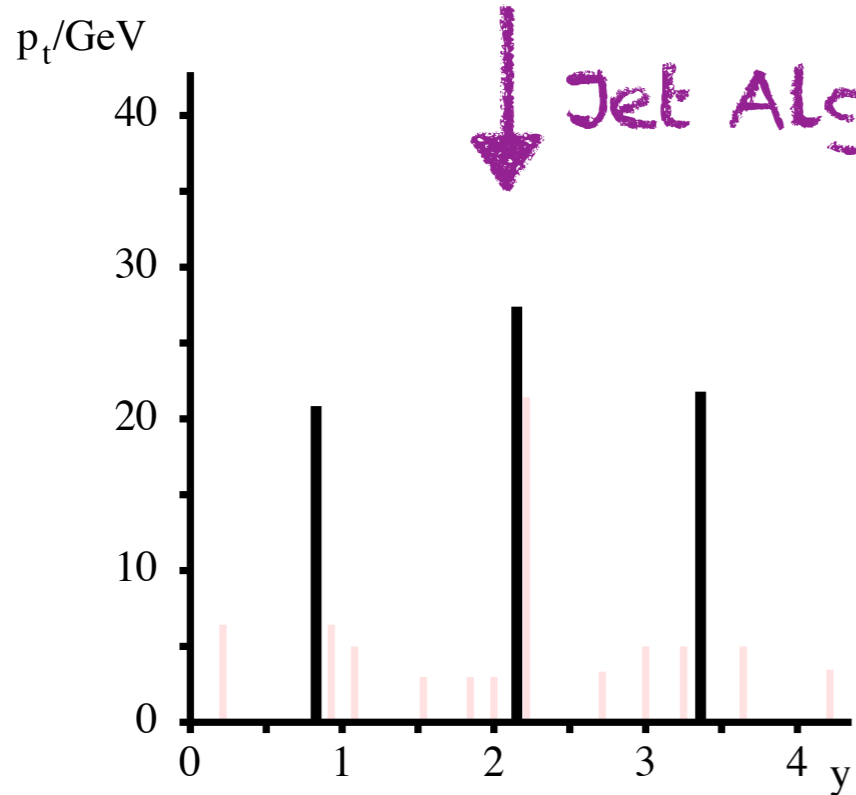
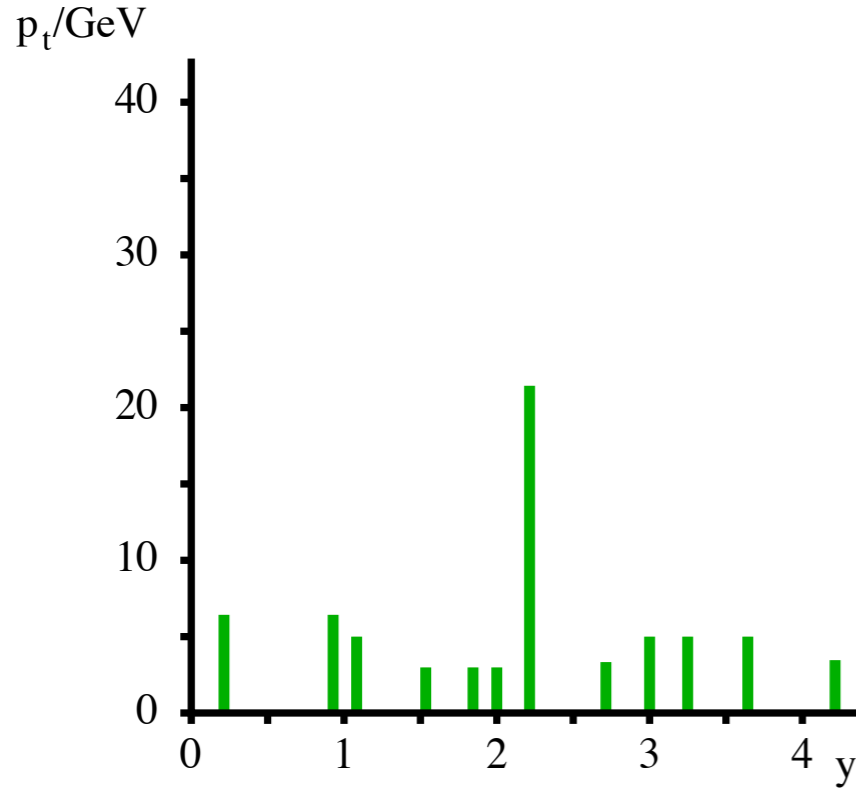
✓ Eğer d_{ij} en küçük ise, i ve j parçacıklarını birleştir ve başa dön

✓ Eğer d_{iB} en küçük ise, son durumu jet olarak belirle

✓ Hiç bir parçacık kalmayıncaya kadar tekrarla.



CA Algoritması



✓ Infrared and collinear güvenilir.

✓ Küme algoritması.

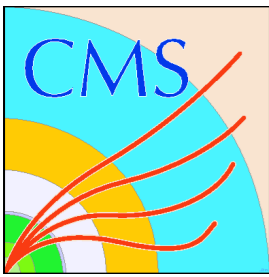
✓ Tüm parçacıkları listele.

✓ d_{ij} and d_{iB} mesafelerini hesapla,

✓ Eğer d_{ij} en küçük ise, i ve j parçacıklarını birleştir ve başa dön

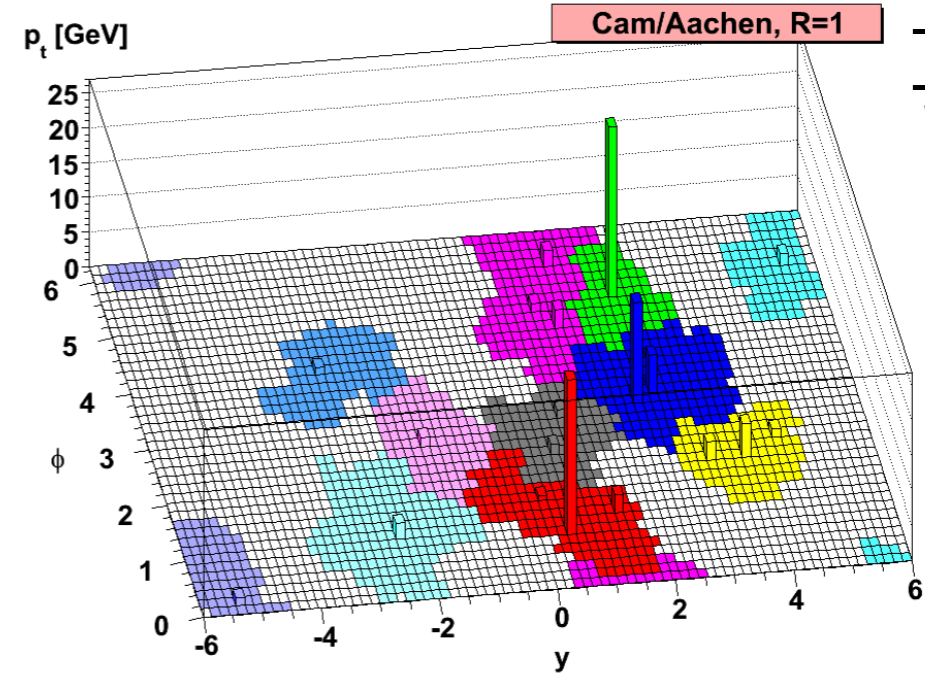
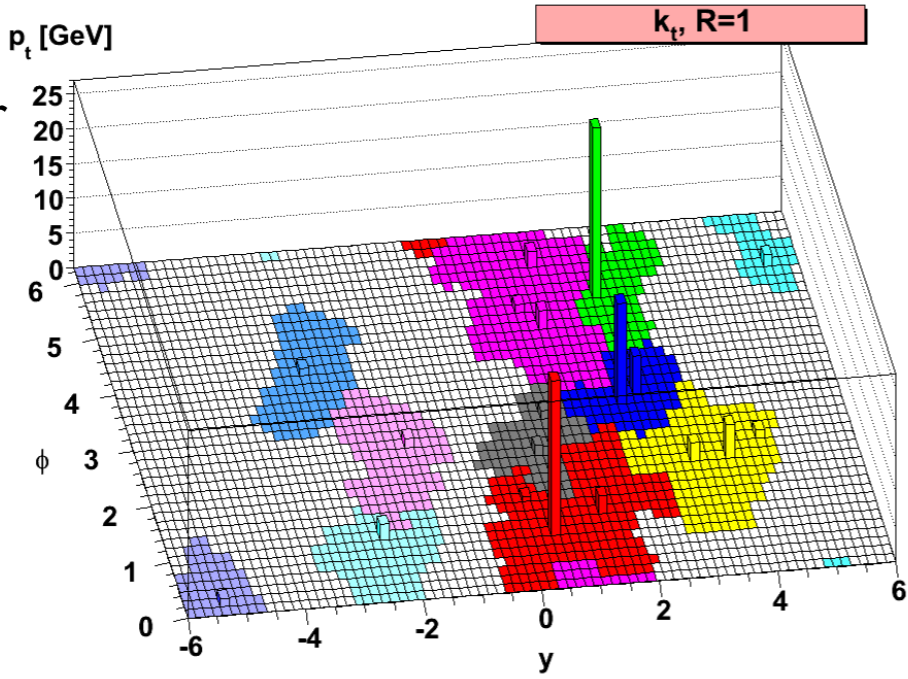
✓ Eğer d_{iB} en küçük ise, son durumu jet olarak belirle

✓ Hiç bir parçacık kalmayıncaya kadar tekrarla.



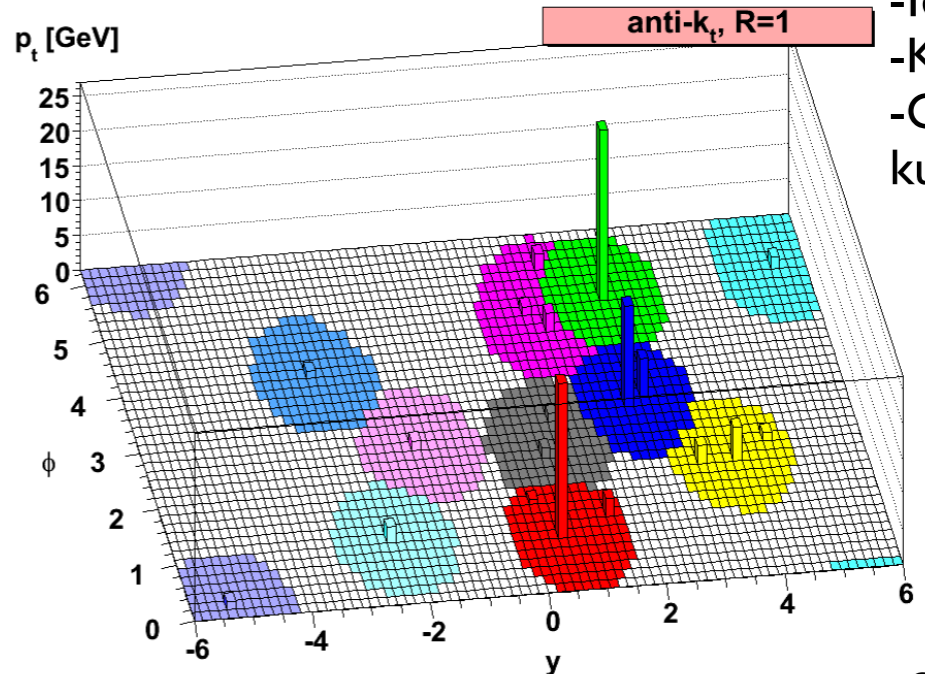
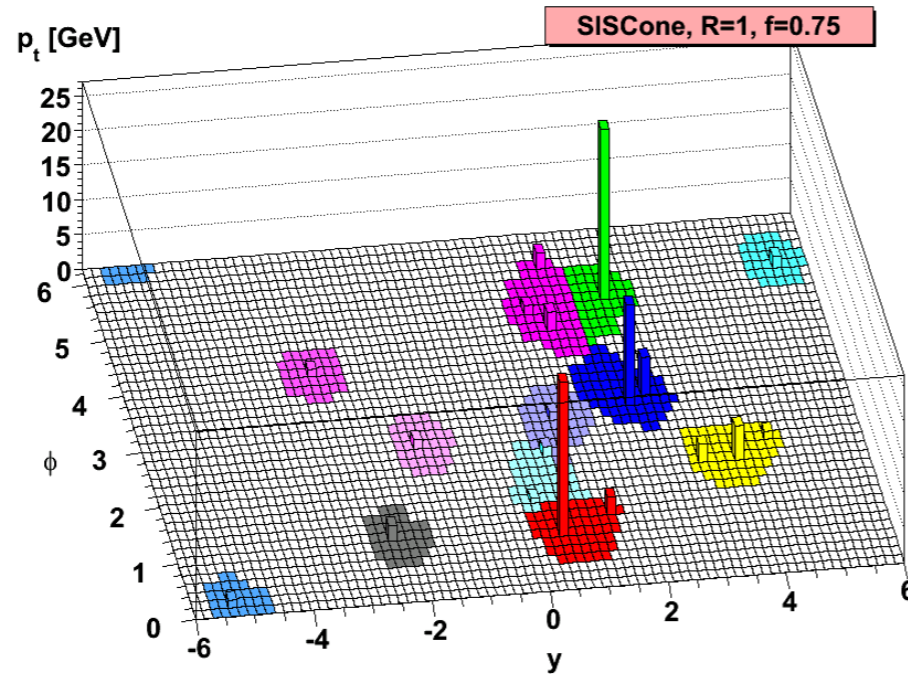
Jet Algoritmaları

- Düzensiz jetler
- Düşük p_t için iyi
- Kalibrasyonu zor



- Düzensiz jetler
- Jet alyapısı için iyi

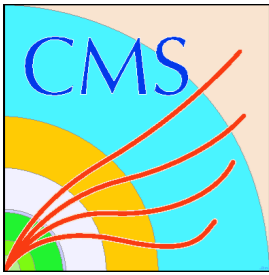
- Yüksek pile-up için kullanışsız



- İdeal kon şekli
- Kalibrasyonu kolay
- CMS ve ATLAS da kullanılır

arXiv-hep-ph:
0906.1833v2

Sertaç Öztürk

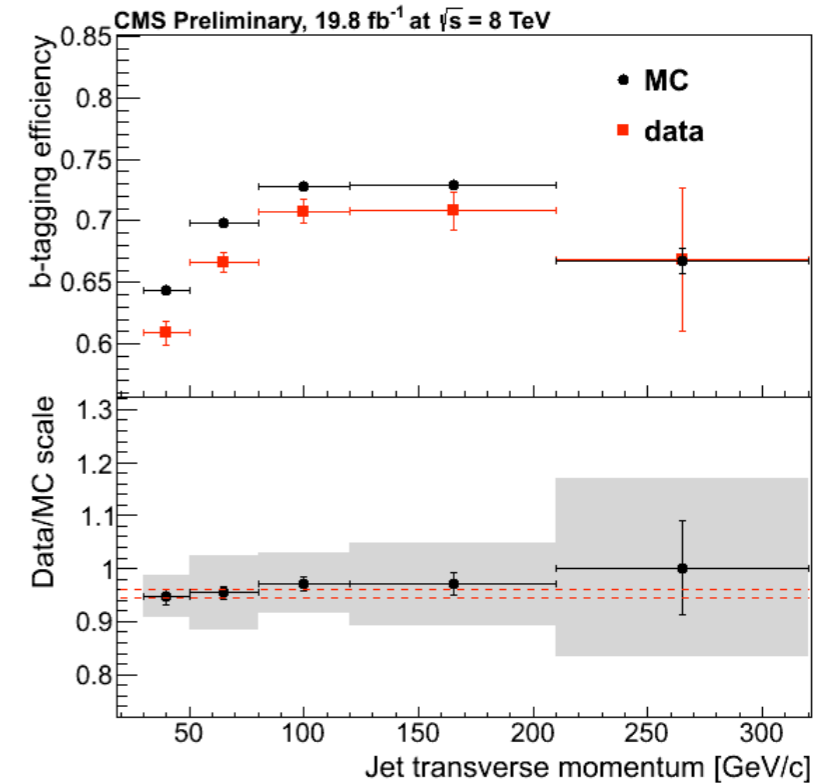
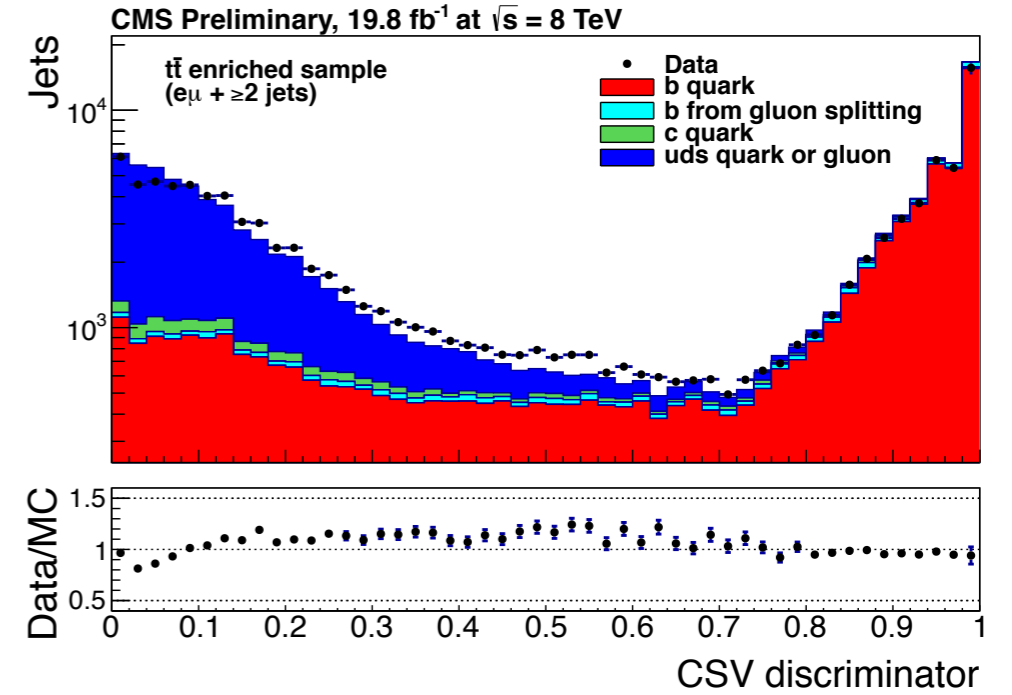
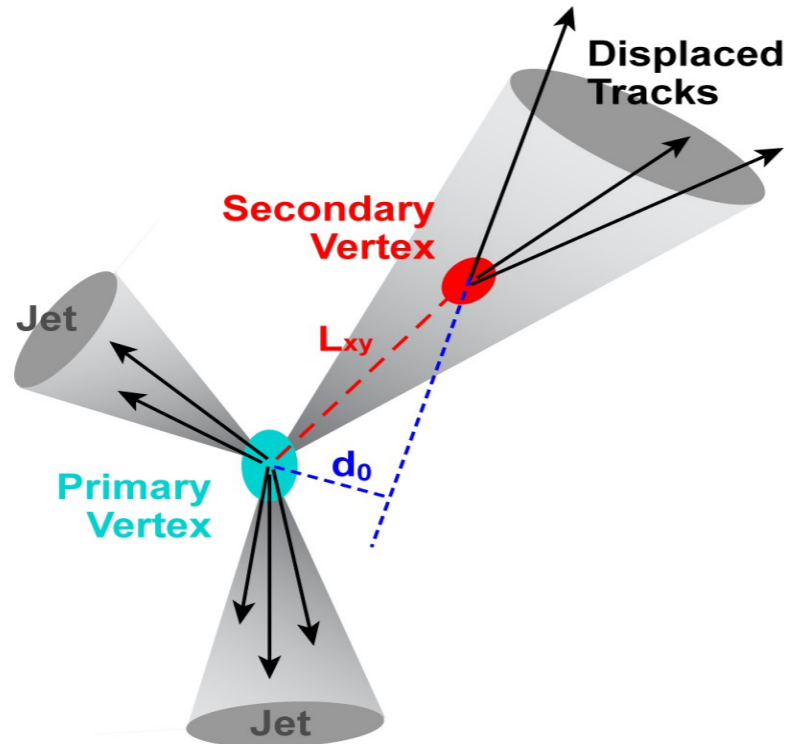


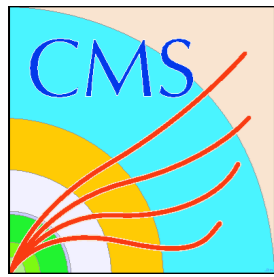
Jet Etiketleme

☑ b-jet etiketleme, kuark/gluon jet etiketleme, W-jet etiketleme

☑ b-jet etiketleme

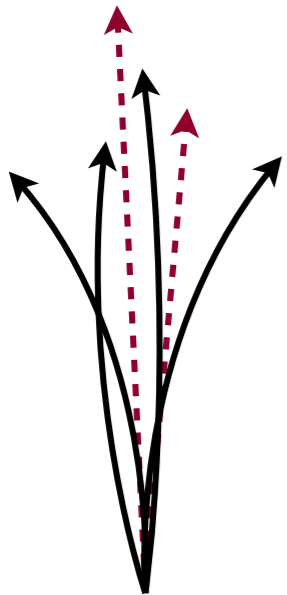
- ✓ b-kuarklar daha uzun yarı ömüre sahiptir.
- ✓ İkincil köşe
- ✓ Büyük etki parametrelili izler
- ✓ Jet içinde muon veya elektron



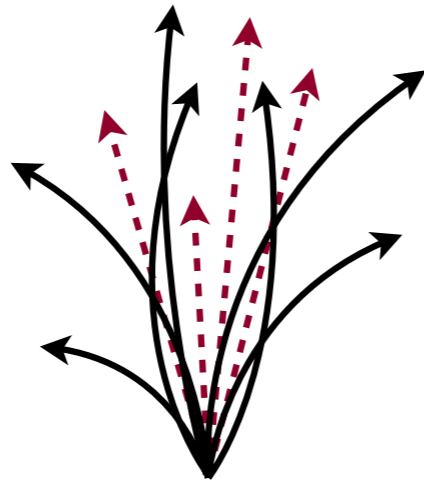


Kuark/Gluon Jet Ayrımı

- ☑ Kuark jet ve gluon jet farklı özelliklere sahiptir.
 - ✓ Renk yükü
 - ✓ Elektrik yükü
 - ✓ Spin, ...
- ☑ Gözlenebilir farklılıklara neden olurlar.
 - ✓ Kütle/pt, yüklü parçacık sayısı, jet şekli, enerji dağılım profili

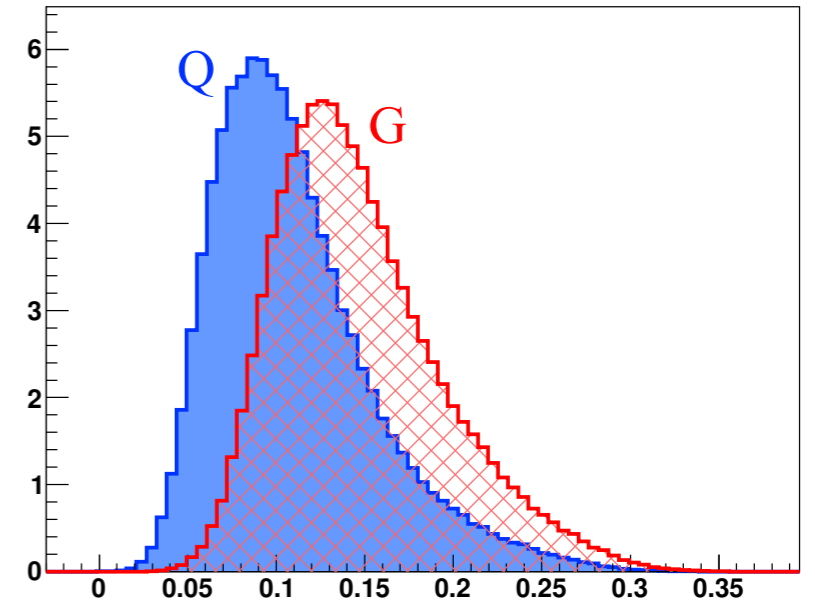


Kuark Jet
 $CF = 4/3$

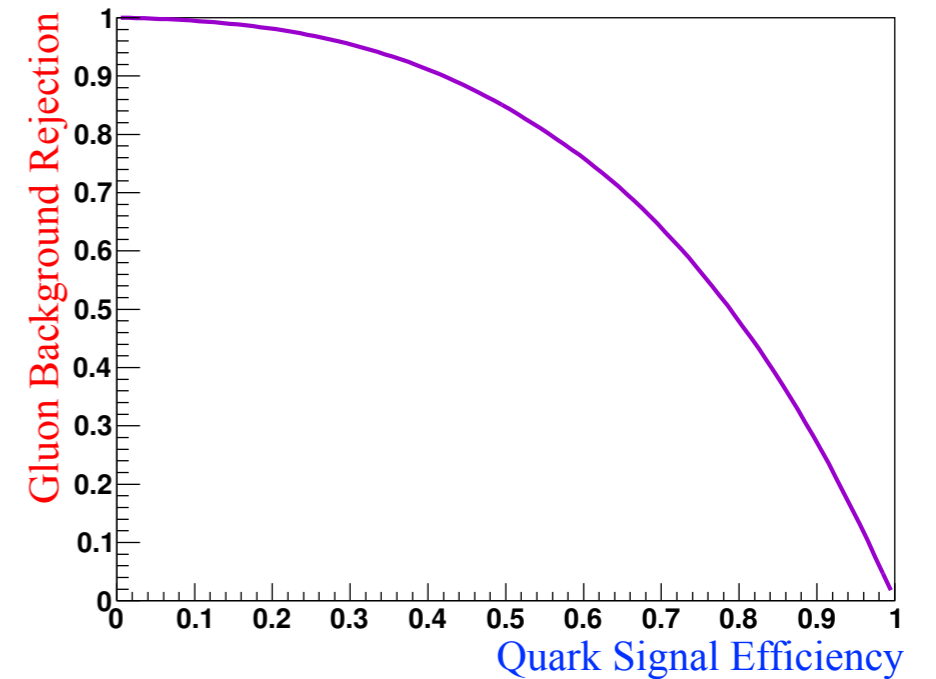


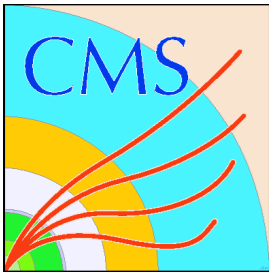
Gluon Jet
 $CA = 3$

mass/Pt



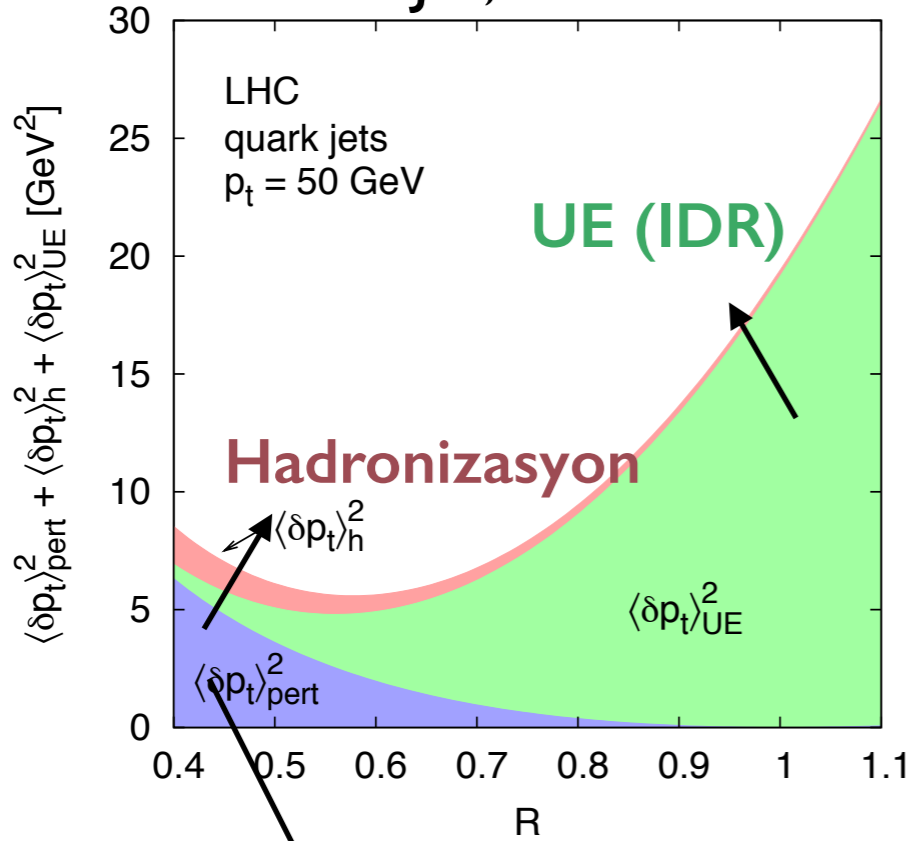
ROC Curve for mass/Pt



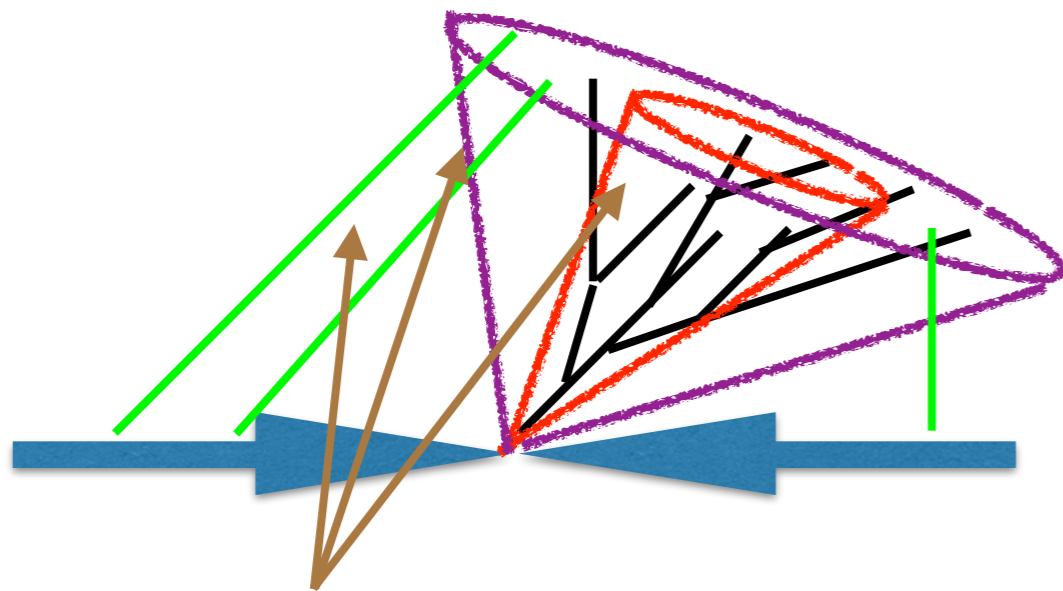


R Seçimi

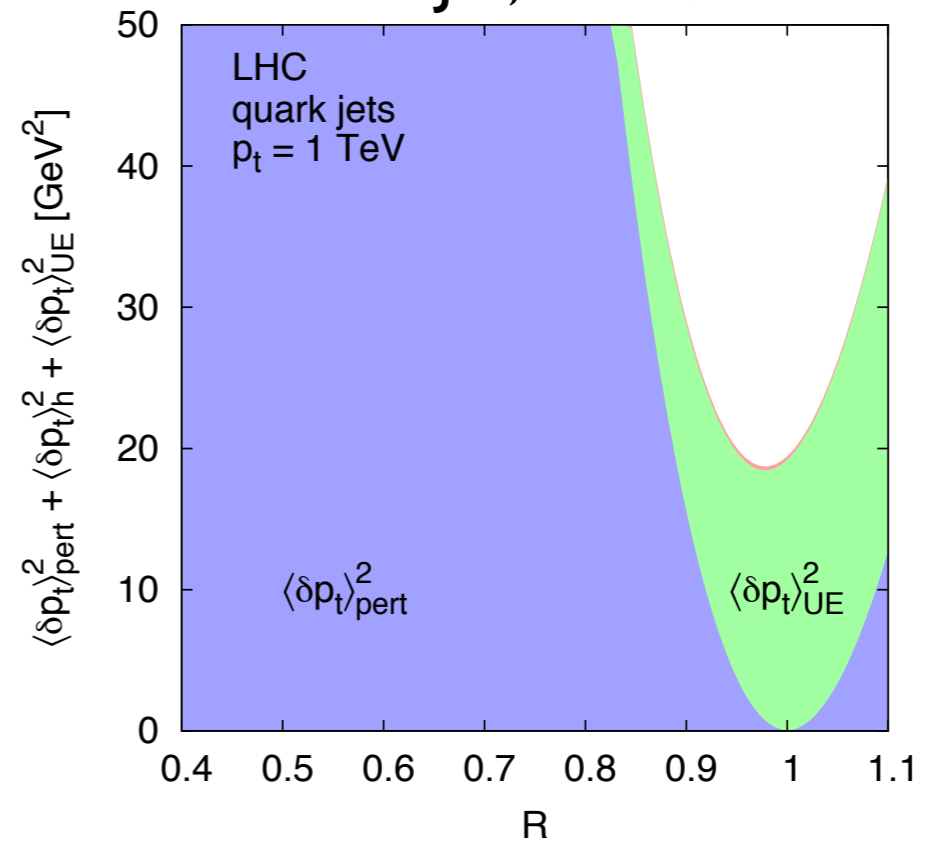
Kuark Jet, $p_t=50$ GeV



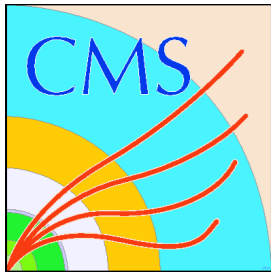
Pertürbative Kısım (SDR)



Kuark Jet, $p_t=1$ TeV

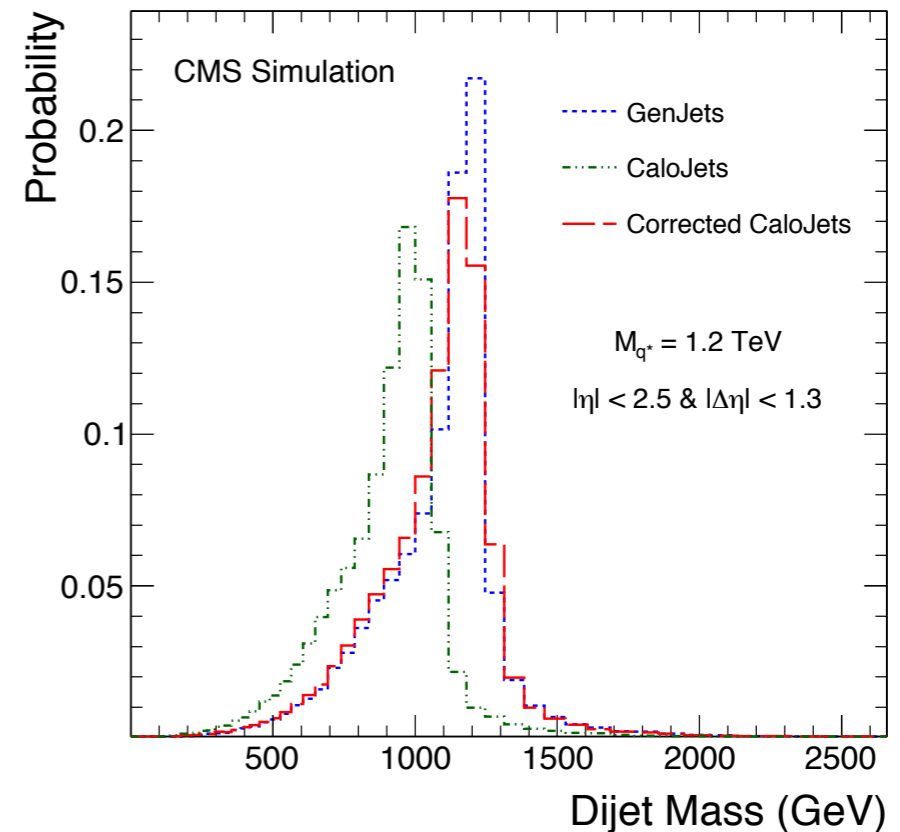
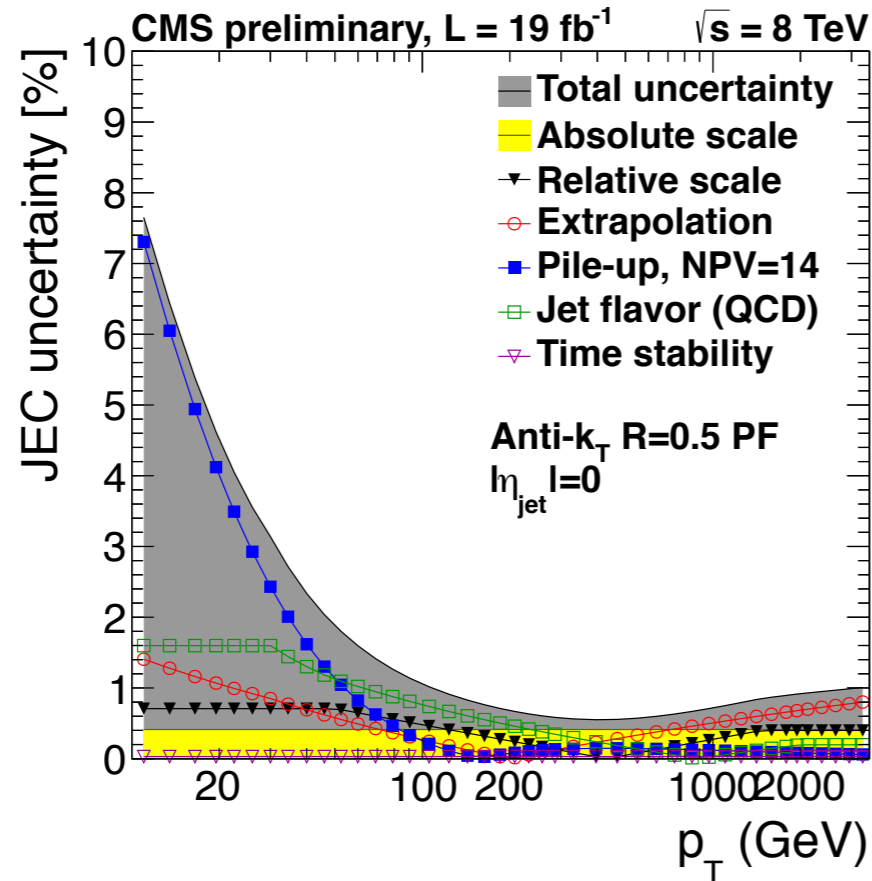


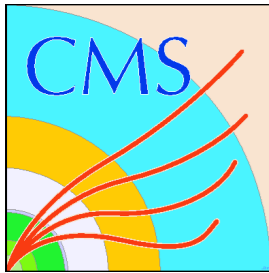
- ✓ Amaç: Parton momentumunu yapılandırmak
- ✓ Optimal R kon yarıçapı jet p_t değerine göre değişiklik gösterir.
 - ✓ Düşük p_t için $R=0.4-0.7$
 - ✓ Yüksek p_t için $R=1$
- ✓ CMS , $R = 0.5$ ve 0.7
- ✓ ATLAS , $R = 0.4$ ve 0.6



Jet Enerji Kalibrasyonu

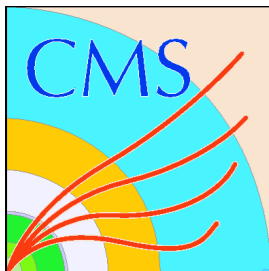
- ✓ Jetler kirlili fizik nesnelere ve kalibre edilmedir.
- ✓ Kalorimetre tepkisi ve dedektörün üniform olmaması.
- ✓ İki-jet olayları ve γ +jet olayları kalibrasyon katsayılarını belirlemek için kullanılır.





Jet Yapılandırma Programları

- Jetleri yapılandırmak için bazı özel programlar kullanılmalıdır.
- ✓ Slowjet (Pythia8 de kendiliğinden var)
- ✓ Fastjet
- ✓ Spartyjet
- ✓ ...



Fastjet Kurulumu

✓ <http://fastjet.fr/>

✓ Kurulum

✓ `wget http://fastjet.fr/repo/fastjet-3.0.6.tar.gz`

✓ `tar -xzf fastjet-3.0.6.tar.gz`

✓ `cd fastjet-3.0.6`

✓ `./configure --prefix=Kurulacak_yer --enable-allcxxplugins`

✓ `make`

✓ `make install`

✓ Pythia8 ile baglanti

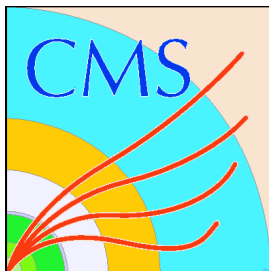
✓ Pythia8/examples

▶ `./configure --with-fastjet=Kurulacak_yer`

✓ Pythia8/rootexamples

▶ `Makefile` ⇒ `CXX += -I/Kurulacak_yer/include -L$(PYTHIA8)/lib -lpythia8 -llhapdfdummy -L/Kurulacak_yer/lib -lfastjet`

▶ `config.mk` ⇒ `FASTJETLOCATION = Kurulacak_yer`



Kullanma Klavuzu

CERN-PH-TH/2011-297

FastJet user manual (for version 3.0.3)

Matteo Cacciari,^{1,2} Gavin P. Salam^{3,4,1} and Gregory Soyez⁵

¹LPTHE, UPMC Univ. Paris 6 and CNRS UMR 7589, Paris, France

²Université Paris Diderot, Paris, France

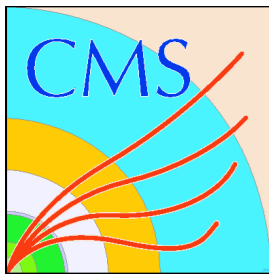
³CERN, Physics Department, Theory Unit, Geneva, Switzerland

⁴Department of Physics, Princeton University, Princeton, NJ 08544, USA

⁵Institut de Physique Théorique, CEA Saclay, France

Abstract

FastJet is a C++ package that provides a broad range of jet finding and analysis tools. It includes efficient native implementations of all widely used $2 \rightarrow 1$ sequential recombination jet algorithms for pp and e^+e^- collisions, as well as access to 3rd party jet algorithms through a plugin mechanism, including all currently used cone algorithms. **FastJet** also provides means to facilitate the manipulation of jet substructure, including some common boosted heavy-object taggers, as well as tools for estimation of pileup and underlying-event noise levels, determination of jet areas and subtraction or suppression of noise in jets.



Örnek Program (I)

```
//Kütüphane dosyaları
#include <iostream>
//Pythia8 ve Fastjet programlarına erişmek için kütüphane dosyaları
#include "Pythia8/Pythia.h"
#include "fastjet/PseudoJet.hh"

using namespace Pythia8;

int main(int argc, char* argv[]) {

    // Root uygulama ortamı oluşturma
    TApplication theApp("hist", &argc, argv);

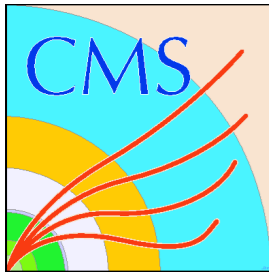
    // Sert KRD süreçleri için Pythia8 ayarları
    Pythia pythia;
    pythia.readString("HardQCD:all = on");
    pythia.readString("PhaseSpace:pTHatMin = 1000 ");
    pythia.readString("PhaseSpace:pTHatMax = 1200 ");

    // Çarpışma ayarları
    pythia.init( 2212, 2212, 14000 );

    //Fastjet analizi - Algoritma ve parametreleri seç
    double Rparam = 0.5;
    fastjet::Strategy          strategy          = fastjet::Best;
    fastjet::RecombinationScheme recombScheme = fastjet::E_scheme;

    fastjet::JetDefinition jetDef_ak(fastjet::antikt_algorithm, Rparam, recombScheme, strategy);

    // Fastjet girdisi
    std::vector <fastjet::PseudoJet> fjInputs;
```



Örnek Program (II)

```
// Olay döngüsüne başla. Olay üret, Olay üretimi başarısızsa atla
for (int iEvent = 0; iEvent < 100; ++iEvent) {
    if (!pythia.next()) continue;

    // Fastjet girdilerini temizle
    fjInputs.resize(0);

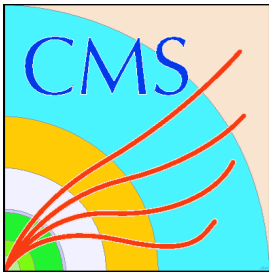
    for (int i = 0; i < pythia.event.size(); ++i){
        // Sadece son durum parçacıklar
        if (!pythia.event[i].isFinal()) continue;
        // pt < 1.0 GeV ve yüklü parçacıkları dışarla
        if (pythia.event[i].isCharged() && pythia.event[i].pT() < 1.0) continue;
        // Nötrinoları dışarla
        if (pythia.event[i].idAbs() == 12 || pythia.event[i].idAbs() == 14 ||
            pythia.event[i].idAbs() == 16) continue;

        // Fastjete girdi olarak kaydet
        fjInputs.push_back( fastjet::PseudoJet( pythia.event[i].px(), pythia.event[i].py(), pythia.event[i].pz(), pythia.event[i].e() ) );
    }

    // Fastjet algoritmasını çalıştır
    vector <fastjet::PseudoJet> inclusiveJets_ak, sortedJets_ak;
    fastjet::ClusterSequence clustSeq_ak(fjInputs, jetDef_ak);

    // Jetleri pT sıraları çıkart (en küçük pT değeri 5.0 GeV)
    inclusiveJets_ak = clustSeq_ak.inclusive_jets(5.0);
    sortedJets_ak = sorted_by_pt(inclusiveJets_ak);

    cout<<sortedJets_ak[0].eta()<<"\t"<<sortedJets_ak[0].phi_std()<<"\t"<<sortedJets_ak[0].perp()<<endl;
}
//Olay üretiminin istatistiğini yazdır
pythia.stat();
return 0;
```



Fastjet Kullanımı

```
#include "fastjet/ClusterSequence.hh"
```

FastJet kütüphane dosyası

```
// Fastjet analizi - Algoritma ve parametreleri seç
```

```
double Rparam = 0.5;
```

Jet yapılandırma yarıçapı

```
fastjet::Strategy strategy = fastjet::Best;
```

Pargacık sayısı ve jet yarıçapına göre en uygun pargacık kümelenme yöntemini belirler

```
fastjet::RecombinationScheme recombScheme = fastjet::E_scheme;
```

pargacık kümelenmesi boyunca jetin 4-momentumunun nasıl birleştirileceğini belirler.

```
fastjet::JetDefinition jetDef(fastjet::antikt_algorithm, Rparam, recombScheme, strategy);
```

```
enum JetAlgorithm {kt_algorithm, cambridge_algorithm,  
                  antikt_algorithm, genkt_algorithm,  
                  ee_kt_algorithm, ee_genkt_algorithm, ...};
```

E_scheme
pt_scheme
pt2_scheme
Et_scheme
Et2_scheme
BIpt_scheme
BIpt2_scheme

```
// Fastjet girdisi
```

```
std::vector <fastjet::PseudoJet> fjInputs;
```

Pargacık bilgilerini girmek için vektör

```
fjInputs.push_back( fastjet::PseudoJet( pythia.event[i].px(), pythia.event[i].py(), pythia.event[i].pz(), pythia.event[i].e() ) );
```



Fastjet Kullanımı

// Fastjet algoritmasını çalıştır

```
fastjet::ClusterSequence clustSeq(fjInputs, jetDef);
```

Verilen parametrelere göre jet yapılandırılması yapılır ve jetler belirlenir

// pT sıralı jetleri çıkar (minimum pT 5.0 GeV)

```
double pt_min = 0.5;
```

```
vector<fastjet::PseudoJet> inclusiveJets = clustSeq.inclusive_jets(pt_min);
```

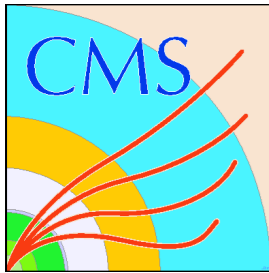
Belirlenen jetlerin dik momentumuna kısıtlama getiri

```
vector<fastjet::PseudoJet> sortedJets = sorted_by_pt(inclusiveJets);
```

Jetler pt büyüklüklerine göre sıralanır.

```
cout<<sortedJets[0].eta()<<"\t"<<sortedJets[0].phi_std()<<"\t"<<sortedJets_ak[0].perp()<<endl;
```

En yüksek dik momentumlu jete ait bazı değerleri yazdırır.



Erişilebilir Değişkenler

`double E(), e()` ⇒ Jet enerjisi

`double px(), py(), pz()` ⇒ Jet momentumunun x, y ve z bileşenleri

`double phi()` ⇒ Jet azimutal açısı ($0-2\pi$)

`double phi_std()` ⇒ Jet azimutal açısı ($-\pi - \pi$)

`double rap(), rapidity()` ⇒ Jet rapiditesi

`double pseudorapidity(), eta()` ⇒ Jet pseudo-rapiditesi

`double pt2(), perp2()` ⇒ Jet dik momentumun karesi

`double pt(), perp()` ⇒ Jet dik momentumunu

`double m2()` ⇒ Jet kütesinin karesi

`double m()` ⇒ Jet kütesini

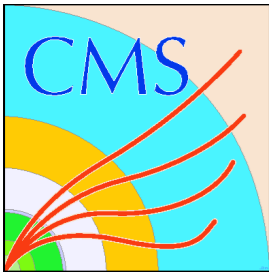
`double mt2(), mperp2()` ⇒ Jet dik kütesinin karesi

`double mt(), mperp()` ⇒ Jet dik kütesi

`vector<PseudoJet> sorted_by_pt(const vector<PseudoJet> & jets);` → Azalan pt sıralı

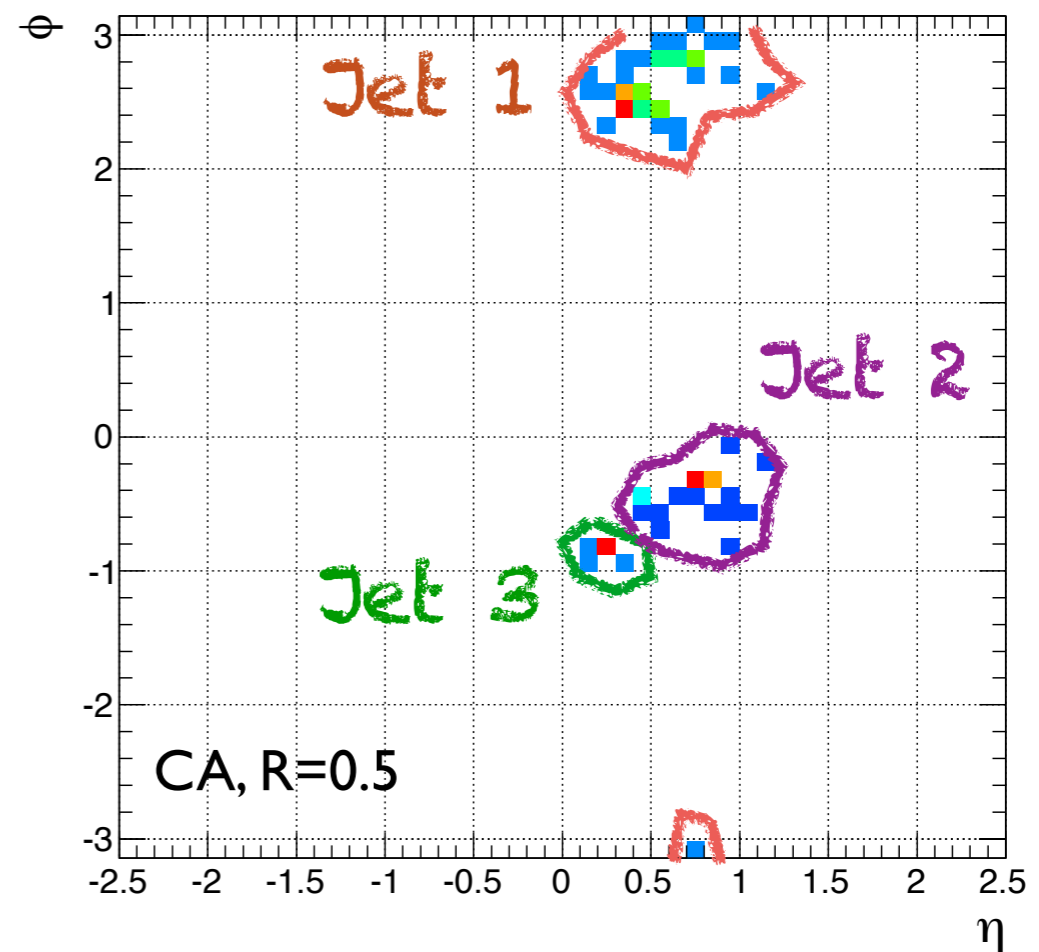
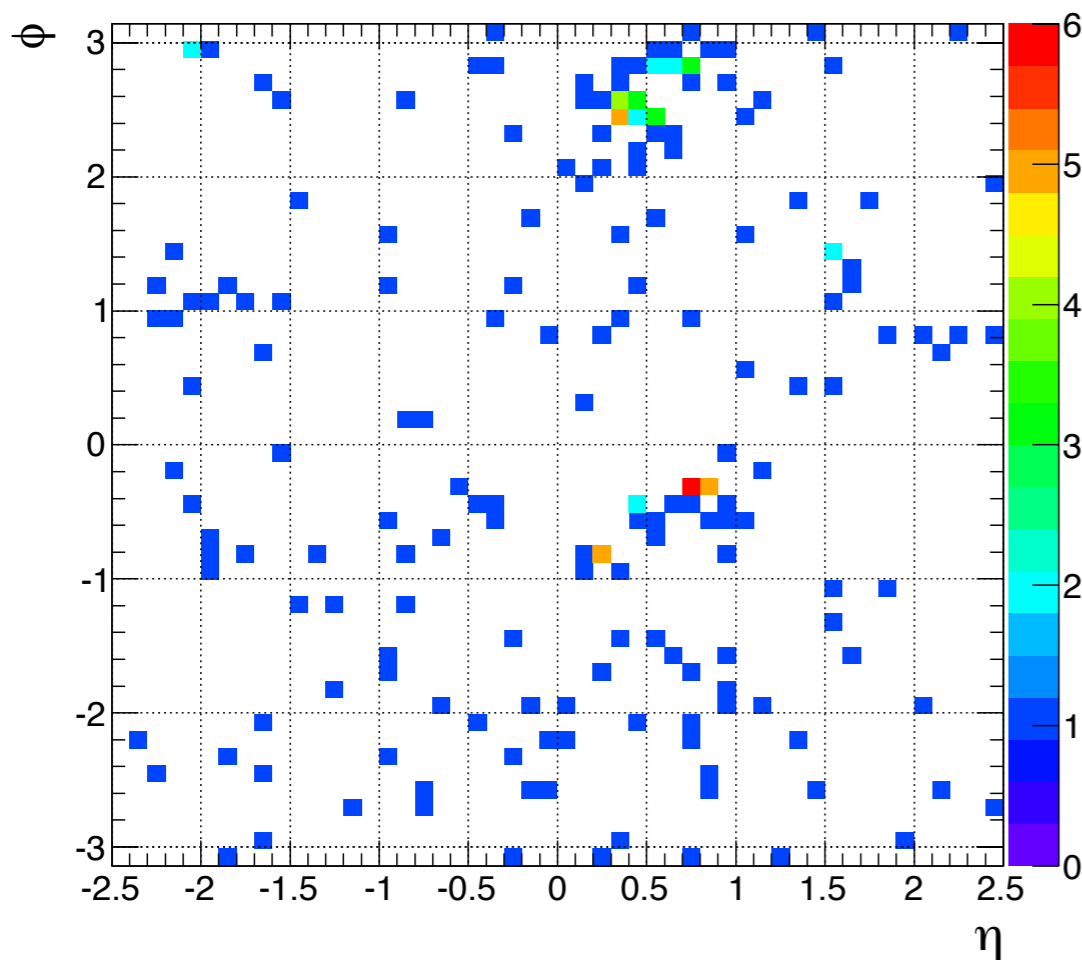
`vector<PseudoJet> sorted_by_rapidity(const vector<PseudoJet> & jets);` → Artan rapidite sıralı

`vector<PseudoJet> sorted_by_E(const vector<PseudoJet> & jets);` → Azalan enerji sıralı

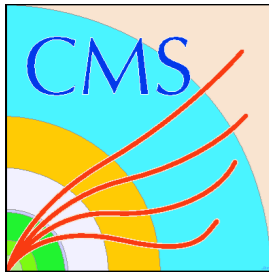


Jetin Bileşenleri

```
vector<fastjet::PseudoJet> constituents_jet1 = sortedJets[0].constituents();  
  
for (unsigned int j = 0; j < constituents_jet1.size(); j++) {  
  
cout<<j<<“\t”<<constituents_jet1[j].eta()<<“\t”<<constituents_jet1[j].phi_std()<<“\t”  
<<constituents_jet1[j].pt()<<endl;  
}
```



$pp \rightarrow W+g \rightarrow uug \rightarrow 3 \text{ jet}$



Seçici (Selector)

✓ `#include "fastjet/Selector.hh"`

✓ `Selector{Değişken}{Limit Tipi}`

✓ `{Değişken}` → Pt, Et, E, Mass, Rap, AbsRap, Eta, AbsEta

✓ `{Limit Tipi}` → Min, Max, Range

▶ `SelectorAbsRapMax(2.5)` → $|y| < 2.5$ rapidite aralığındaki jetleri seçer

▶ `SelectorRapRange(1.3, 2.5)` → $1.3 < y < 2.5$ rapidite aralığındaki jetleri seçer

▶ `SelectorPtMin(25.0)` → Dik momentumu 25 GeV den büyük jetleri seçer

▶ `SelectorNHardest(n)` → Dik momentumu en yüksek n sayıda jeti seçer

✓ `&&`, `||` ve `!` operatörleri kullanılabilir.

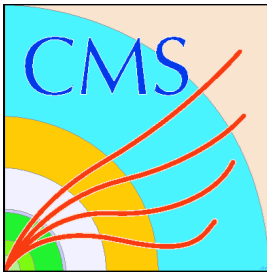
✓ `SelectorNHardest(2) && SelectorAbsRapMax(2.5)`

✓ `! SelectorNHardest(2)`

✓ Sıralamalı seçim yapmak için `*` kullanılır

✓ `SelectorAbsRapMax(2.5) * SelectorPtMin(25.0)`

▶ Önce rapidite seçimini sonra dik momentum seçimini uygular.



Seçicinin Kullanımı

Parçacık Seviyesinde

```
fjInputs.push_back( fastjet::PseudoJet( pythia.event[i].px(), pythia.event[i].py(), pythia.event[i].pz(),  
pythia.event[i].e() ) );
```

```
fastjet::Selector particle_selector = fastjet::SelectorAbsRapRange(1.0,2.5) ||  
(fastjet::SelectorAbsRapMax(1.0) && fastjet::SelectorPtMin(1.0));
```

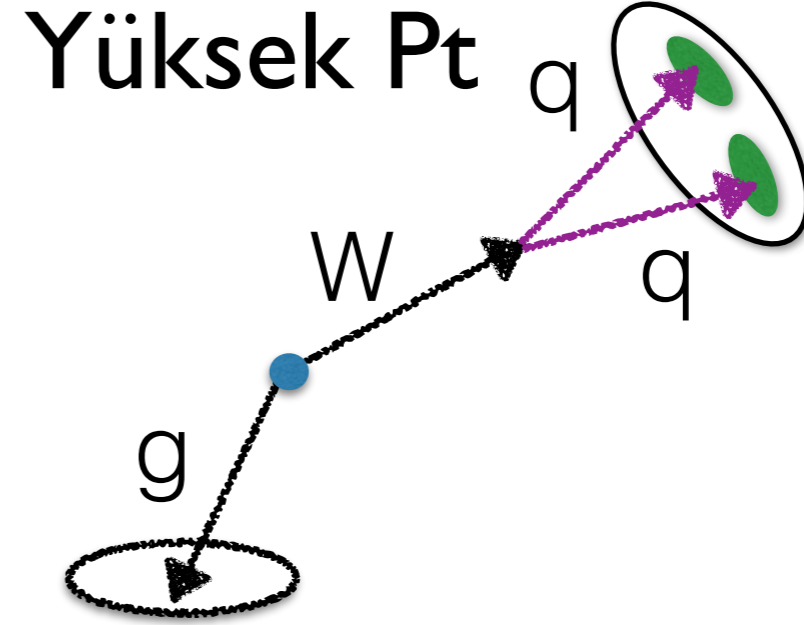
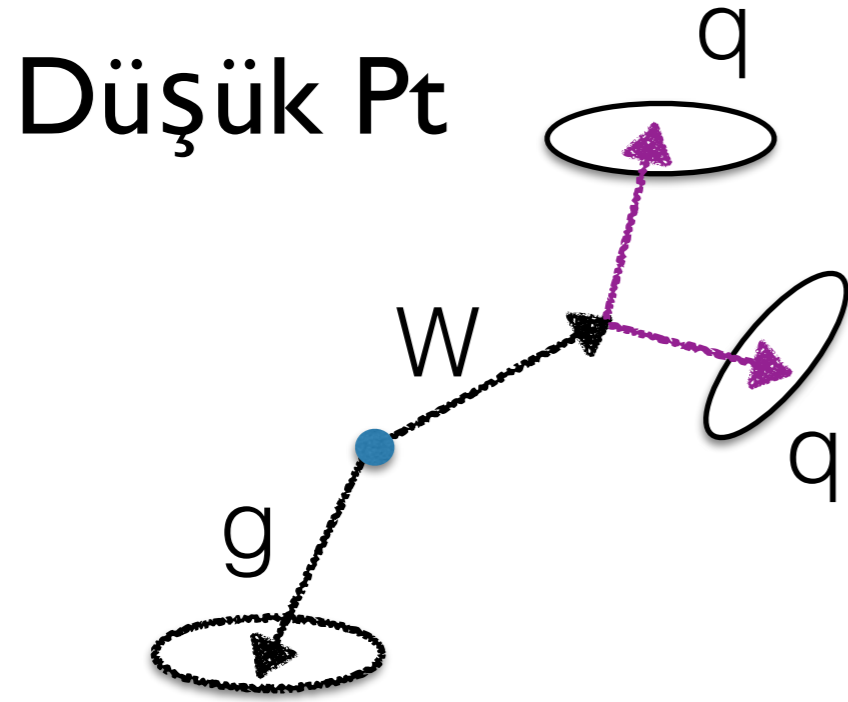
```
fjInputs = particle_selector(fjInputs);
```

```
fastjet::ClusterSequence clustSeq(fjInputs, jetDef)
```

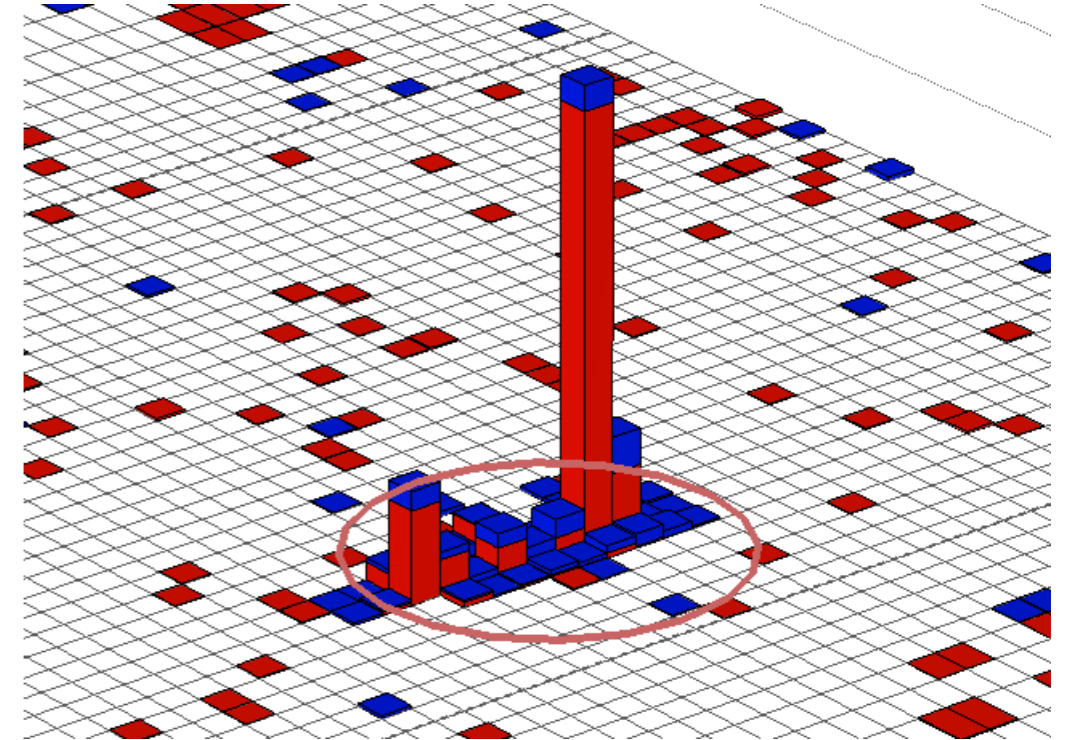
Jet Seviyesinde

```
fastjet::Selector jet_selector = fastjet::SelectorNHardest(2) * fastjet::SelectorAbsRapMax(2.5);
```

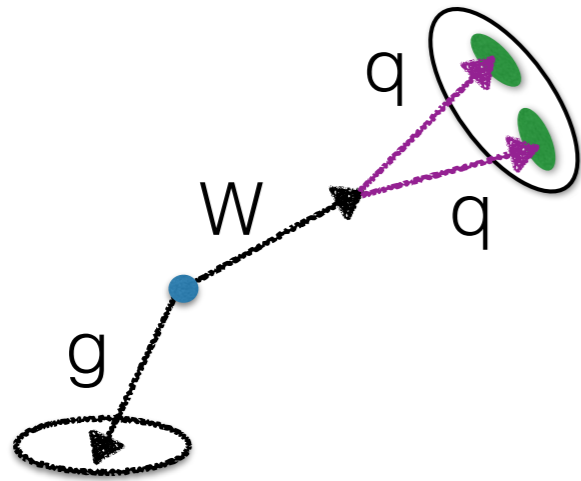
```
vector<fastjet::PseudoJet> inclusive_jets = sorted_by_pt(jet_selector(clust_seq.inclusive_jets()));
```



- ☑ Bazı yüksek enerjili durumlarda iki jete bozunan bir parçacık tek bir jet gibi gözlemlenebilir.
- ☑ Bir jet iki veya daha fazla alt jet içerebilir.
- ☑ Alt jetleri belirlemek için özel yöntemler kullanılmaktadır.



Alt-Jetler (II)

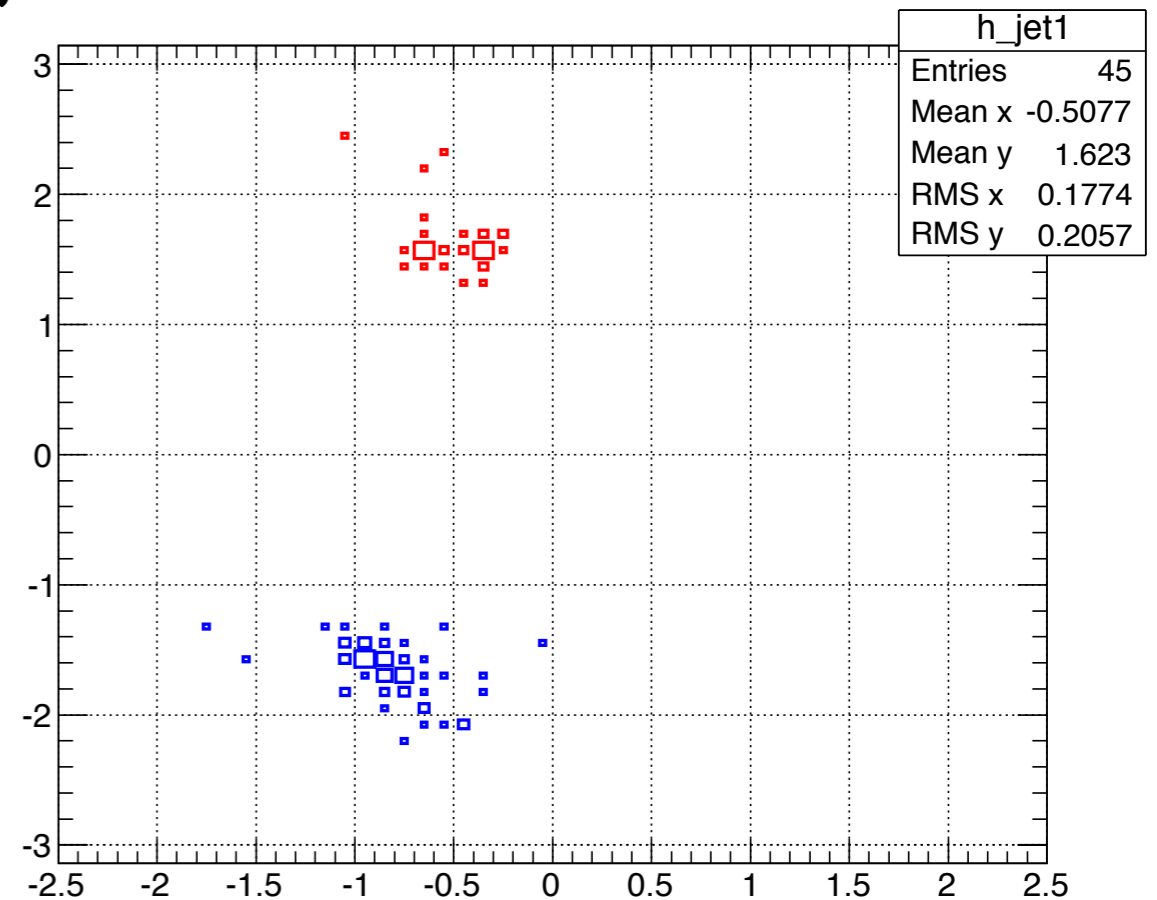
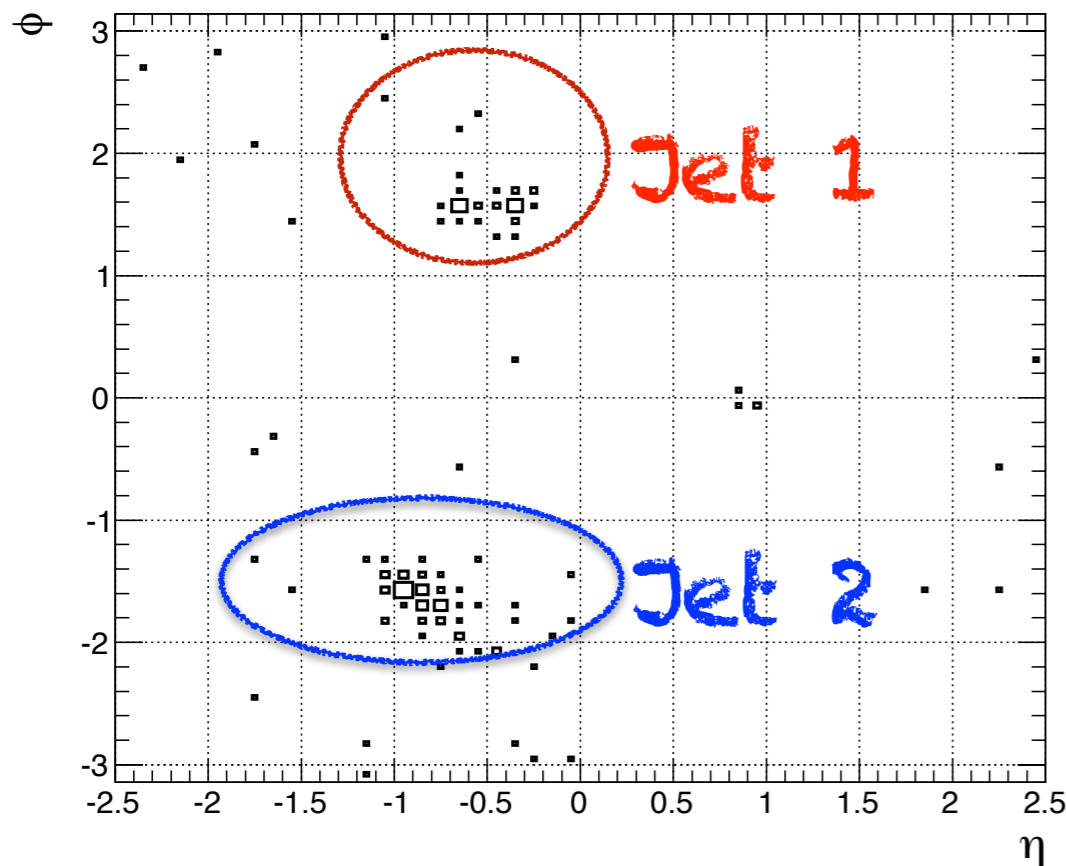


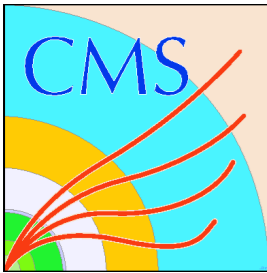
$pp \rightarrow W+g \rightarrow q+q+g$

✓ 3 jetli olay

Yüksek enerjide $W \rightarrow q+q$ ötelenmiş tek bir jet gibi davranabilir.

✓ 2 jetli





Alt-Jetleri Belirleme

```
double Rsub = 0.3;
double dcut = pow(Rsub/Rparam,2);

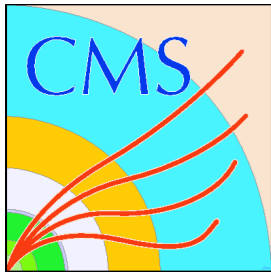
for (unsigned int i = 0; i <sortedJets.size(); i++) {

cout<<sortedJets[i].eta()<<"\t"<<sortedJets[i].phi_std()<<"\t"<<sortedJets[i].perp()<<"\t"<<sortedJets[i].constituents().size()<<endl;

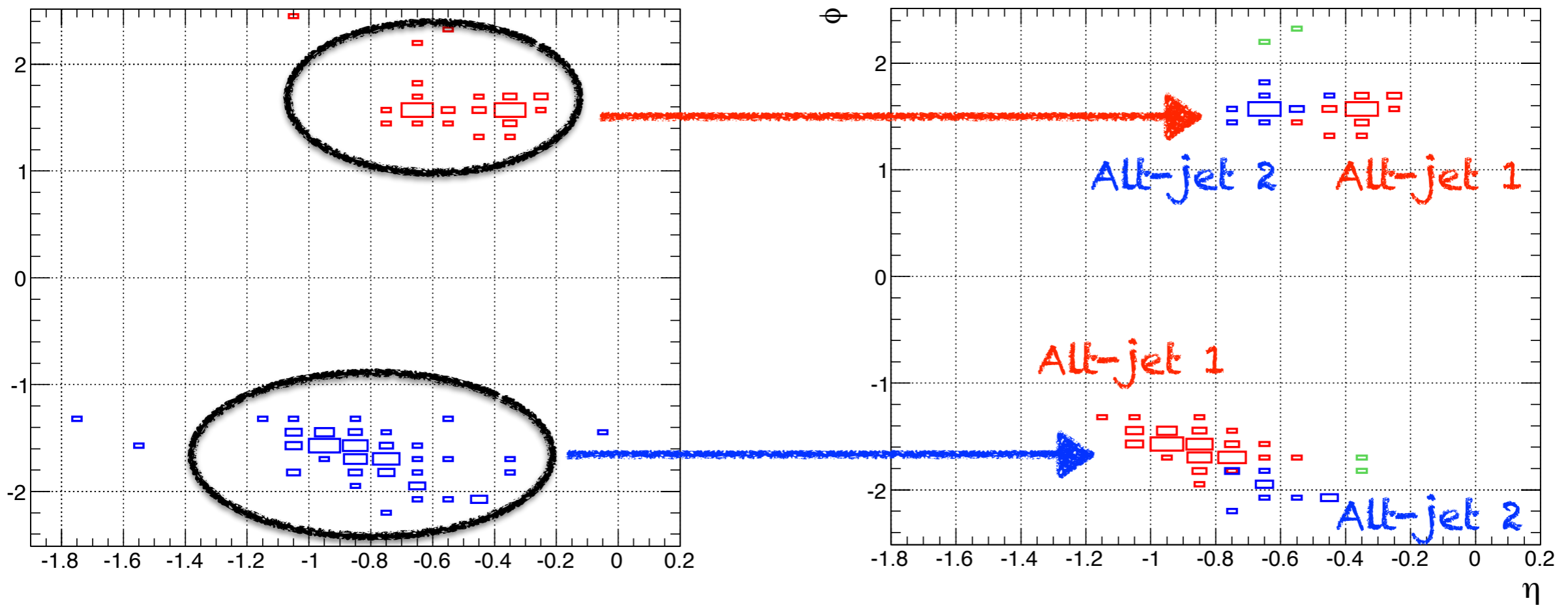
vector<fastjet::PseudoJet> subjets = sorted_by_pt(sortedJets[i].exclusive_subjets(dcut));

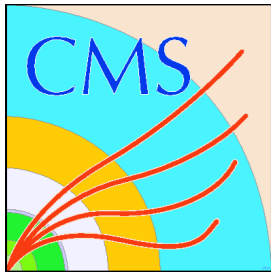
for (unsigned int j=0; j<subjets.size(); j++)

cout<<subjets[j].eta()<<"\t"<<subjets[j].phi_std()<<"\t"<<subjets[j].perp()<<"\t"<<subjets[j].constituents().size()<<endl;
```



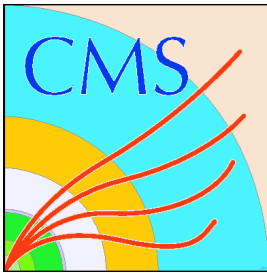
Alt-Jetleri Belirleme (II)





Akşam için çerezlik

- 14 TeV lik kütle merkezi enerjisinde iki proton çarpışsın. Kütle 2 TeV olan Randall-Sundrum Graviton parçacığı üretilip iki gluona bozulsun. Anti-kt algoritması kullanarak $R=0.3$, $R=0.7$ ve $R=1.1$ koni yarıçaplarını için yapılandırılan iki-jet kütle dağılımlarını elde edip karşılaştırınız (ve yorumlayınız).



Yol Gösterme

- `ptyhia8/rootexamples`
 - ✓ `hist.cc`
- Olay sürecini yeniden tanımla.
- Jet yapılandırma parametrelerini değiştir.
- İki-jet kütle dağılımını elde et.
 - ✓ `TLorentzVector`
- Karşılaştır.