



---

# VXDTF2:

## funzionamento, performance, selezione del training sample

---

BELLE II ITALIA

20 NOVEMBRE 2017, PISA

*Valerio Bertacchi,*

*Eugenio Paoloni,*

*Thomas Lück*



# Outline

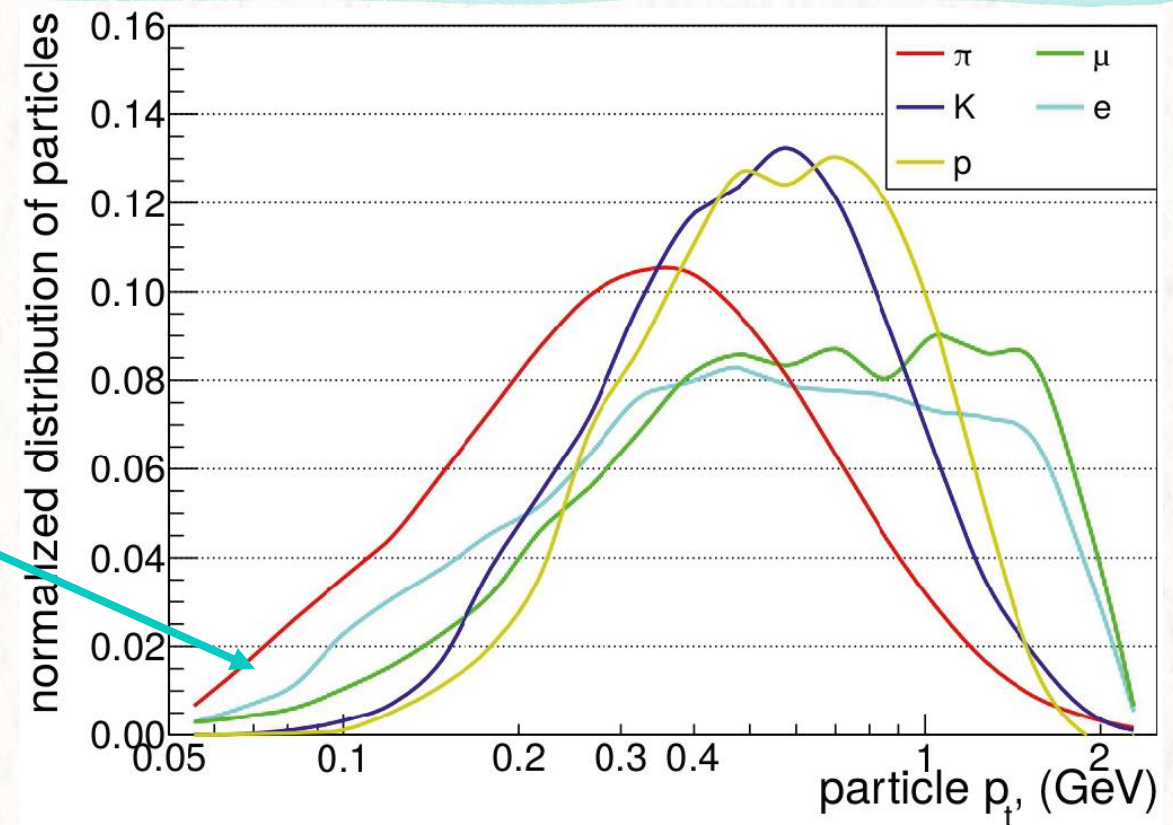
- il Vertex Detector Track Finder redesigned (VXDTF2)
  - Funzionamento
  - Performance release-09
- Una migliore definizione di Training Sample per il VXDTF2
  - Motivazioni per una migliore definizione e selezione
  - Procedura di selezione
  - Risultati della selezione sul Training Sample, sulla Sector Map e sulle performance di Tracking

# Il VXDTF2

- Dalla release 1 diventerà il software di default per la ricostruzione VXD-standalone
- **Fast Reconstruction:** ricostruzione SVD-only nell'HLT (300 ms di latenza) ed estrapolazione Regioni di Interesse (ROI) su PXD
- **Full Reonstrcution:** ricostruzione SVD+PXD offline

## Compiti

- Tracking a basso impulso (fino a 25 MeV/c)
- Produzione delle ROI
- Miglioramento della risoluzione in impulso e vertexing, combinando le tracce con la CDC



# Il concetto di Sector Map

- Raccogliere le informazioni geometriche delle tracce:
  - Distribuzione delle hit
  - Distribuzione dei segmenti di 2 hit
  - Distribuzione delle triplette di 3 hit
- Individuare insieme di variabili che discriminino le tracce dalle combinazioni casuali
- Filtri su tali variabili

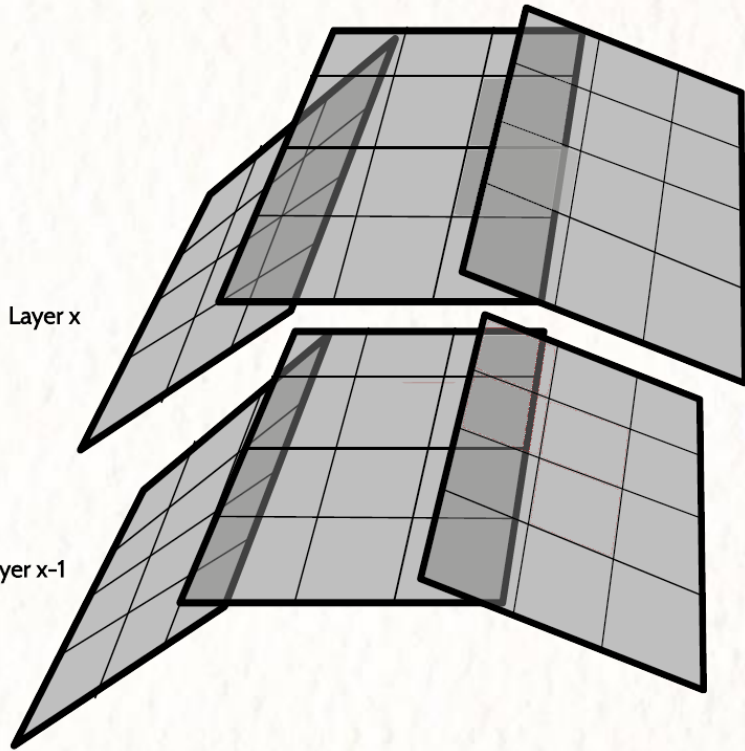


**Sector Map:** insieme di connessioni e valori di soglia dei filtri

# Il concetto di Sector Map

**Sector Map:** insieme di connessioni e valori di soglia dei filtri

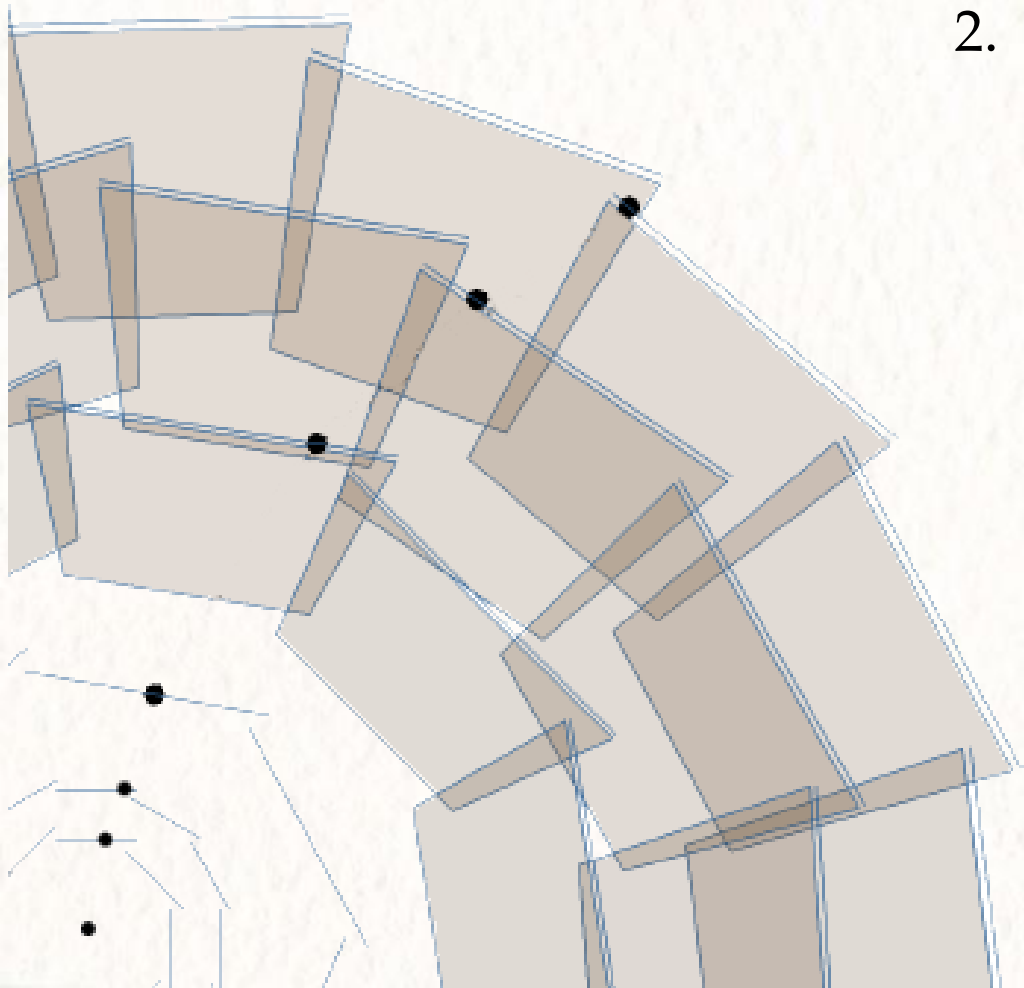
1. Suddiviso ogni sensore dell'VXD in *Settori*



# Il concetto di Sector Map

**Sector Map:** insieme di connessioni e valori di soglia dei filtri

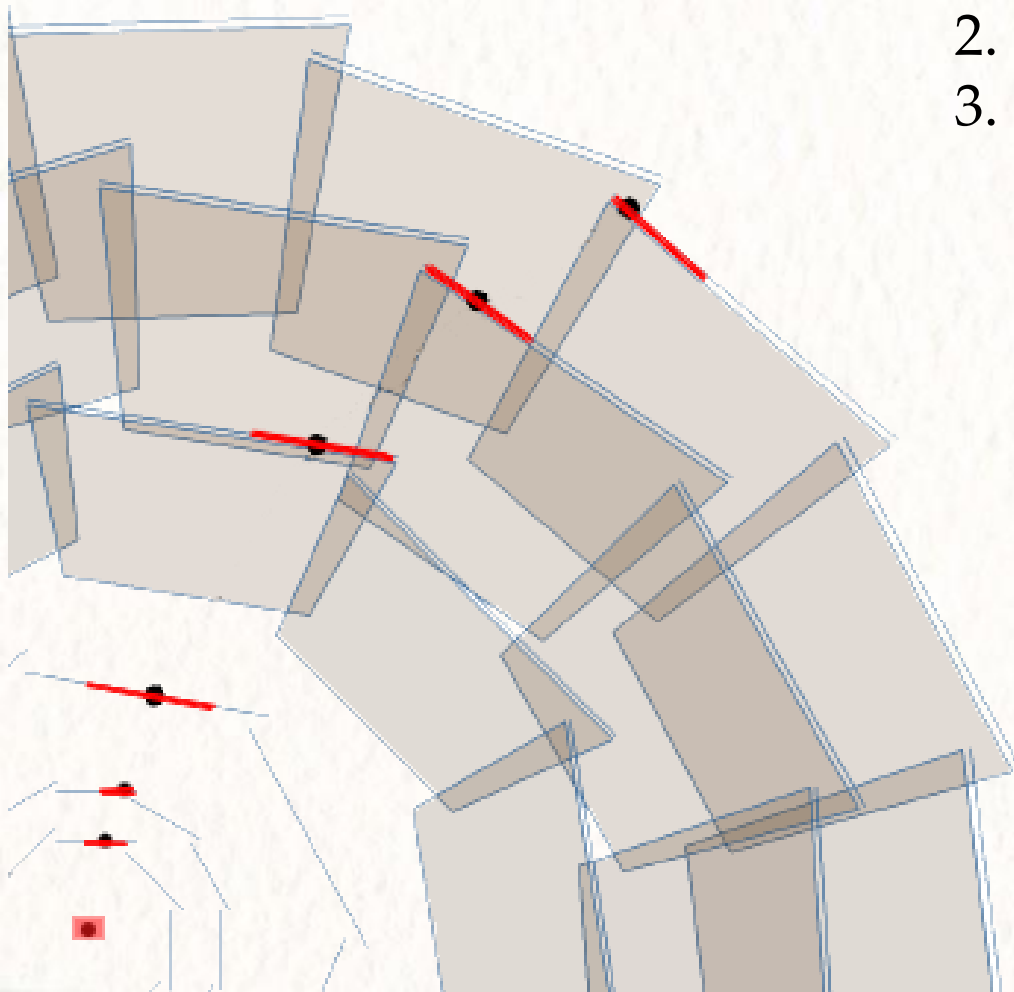
1. Suddiviso ogni sensore dell'VXD in *Settori*
2. Simulati eventi (*Training Sample*) nel VXD



# Il concetto di Sector Map

**Sector Map:** insieme di connessioni e valori di soglia dei filtri

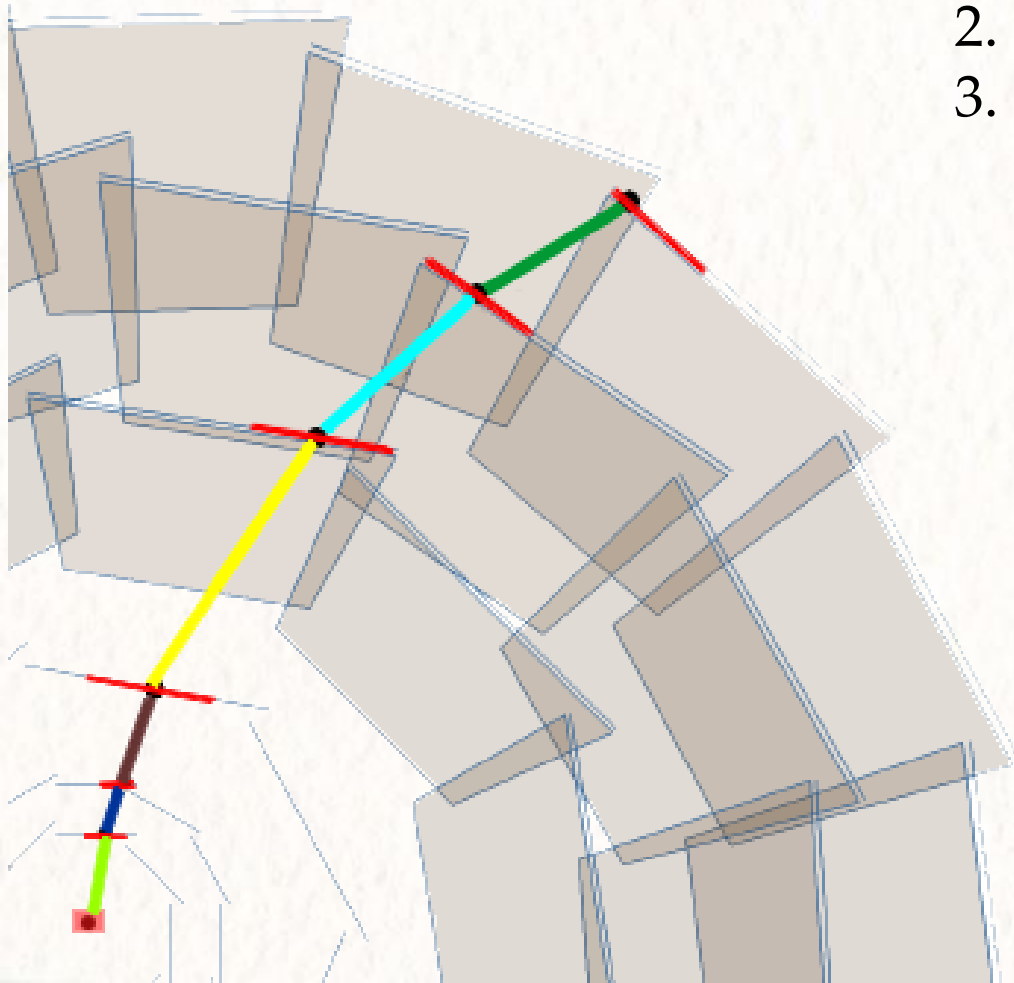
1. Suddiviso ogni sensore dell'VXD in *Settori*
2. Simulati eventi (*Training Sample*) nel VXD
3. Memorizzate nella Sector Map:
  - Settori su cui è presente una hit simulata (MC-attivi)



# Il concetto di Sector Map

**Sector Map:** insieme di connessioni e valori di soglia dei filtri

1. Suddiviso ogni sensore dell'VXD in *Settori*
2. Simulati eventi (*Training Sample*) nel VXD
3. Memorizzate nella Sector Map:
  - Settori su cui è presente una hit simulata (MC-attivi)
  - Connessioni tra coppie settori MC-attivi (*Settori Amici*)

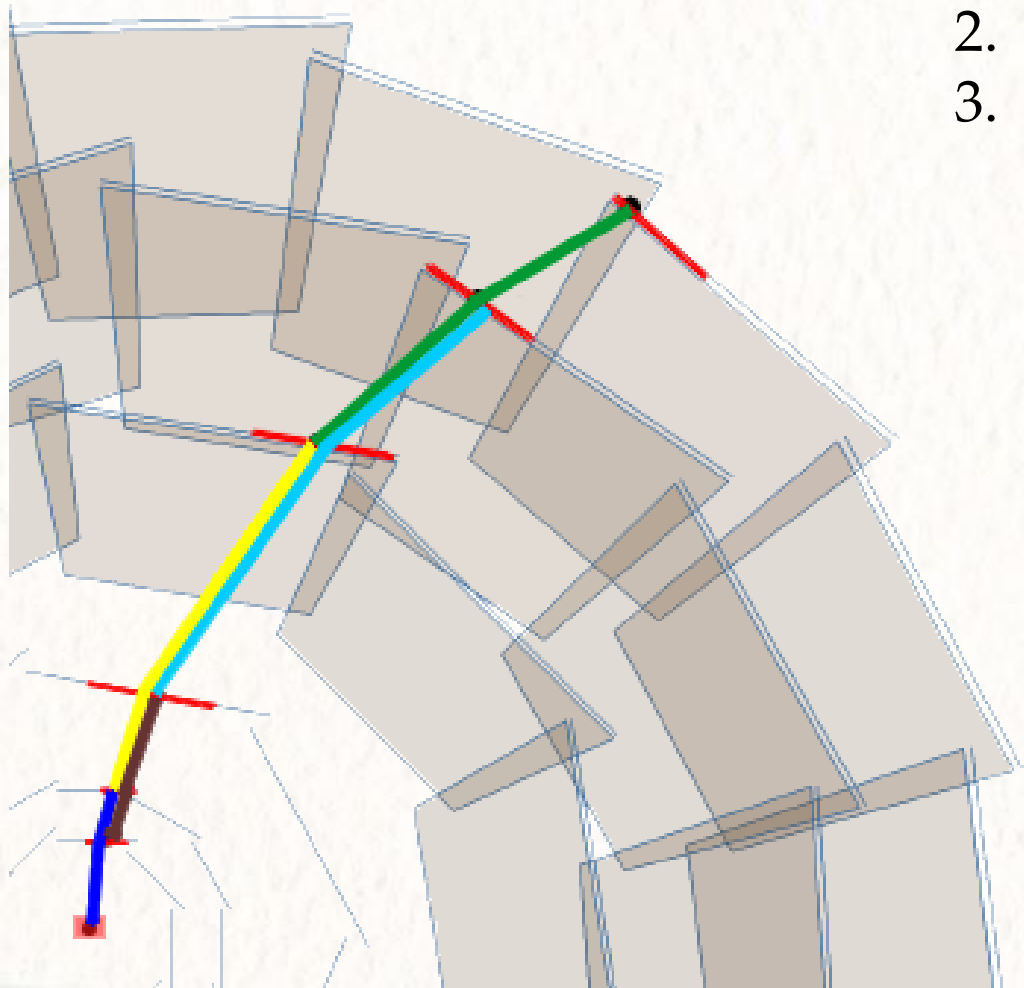




# Il concetto di Sector Map

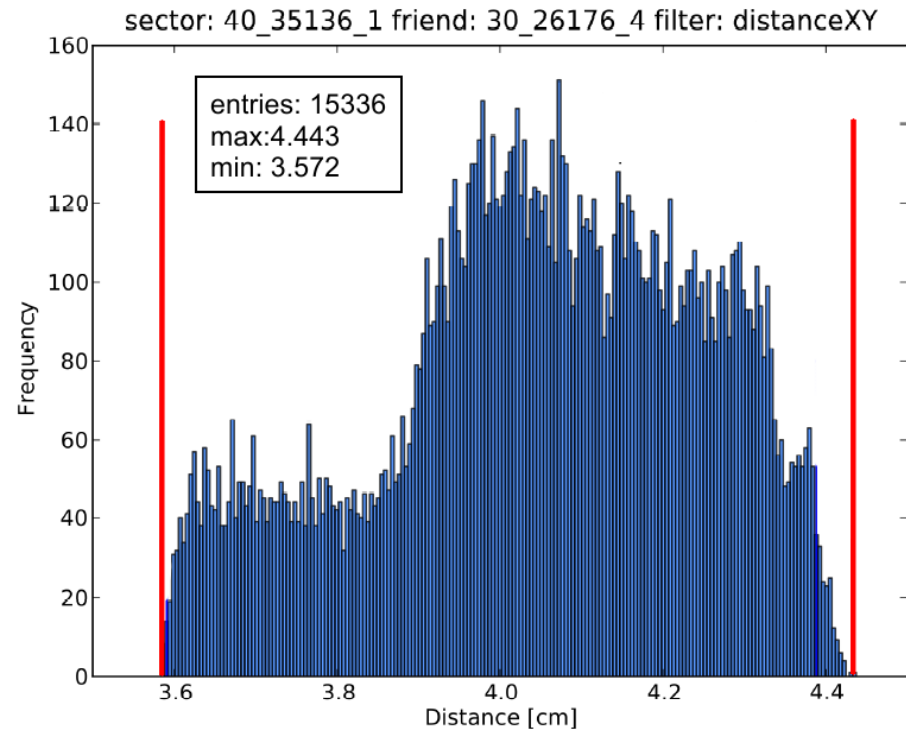
**Sector Map:** insieme di connessioni e valori di soglia dei filtri

1. Suddiviso ogni sensore dell'VXD in *Settori*
2. Simulati eventi (*Training Sample*) nel VXD
3. Memorizzate nella Sector Map:
  - Settori su cui è presente una hit simulata (MC-attivi)
  - Connessioni tra coppie settori MC-attivi (*Settori Amici*)
  - Connessioni tra triplette di settori MC-attivi (*Segmenti Vicini*)



# Il concetto di Sector Map

**Sector Map: insieme di connessioni e valori di soglia dei filtri**



1. Suddiviso ogni sensore dell'VXD in *Settori*
2. Simulati eventi (*Training Sample*) nel VXD
3. Memorizzate nella Sector Map:
  - Settori su cui è presente una hit simulata (MC-attivi)
  - Connessioni tra coppie settori MC-attivi (*Settori Amici*)
  - Connessioni tra triplette di settori MC-attivi (*Segmenti Vicini*)
4. Definiti 2 insiemi di variabili geometriche per Amici e Vicini
  - Distanze su piani XY, RZ, angoli, distanza 3D,
5. Create le distribuzioni di queste variabili con le tracce del training sample per ogni connessione memorizzata
6. Memorizzate nella Sector Map Max e Min delle distribuzioni → soglie dei filtri

# 0 – Fase di Training

- Fase preliminare, effettuata una volta sola
- Prodotta la Sector Map
- Non ripetuta durante la presa dati

## Training Sample (TS)

- Campione di eventi simulati che deve insegnare alla Sector Map le caratteristiche geometriche delle tracce attese in fase di presa dati:
  - momento (curvatura)
  - distribuzione spaziale
  - interazione con la materia (deviazione da elica)
- Scelta naturale:  $\sim 10^6 \Upsilon(4S)$

# 0 – Fase di Training

- Fase preliminare, effettuata una volta sola
- Prodotta la Sector Map
- Non ripetuta durante la presa dati

## Training Sample (TS)

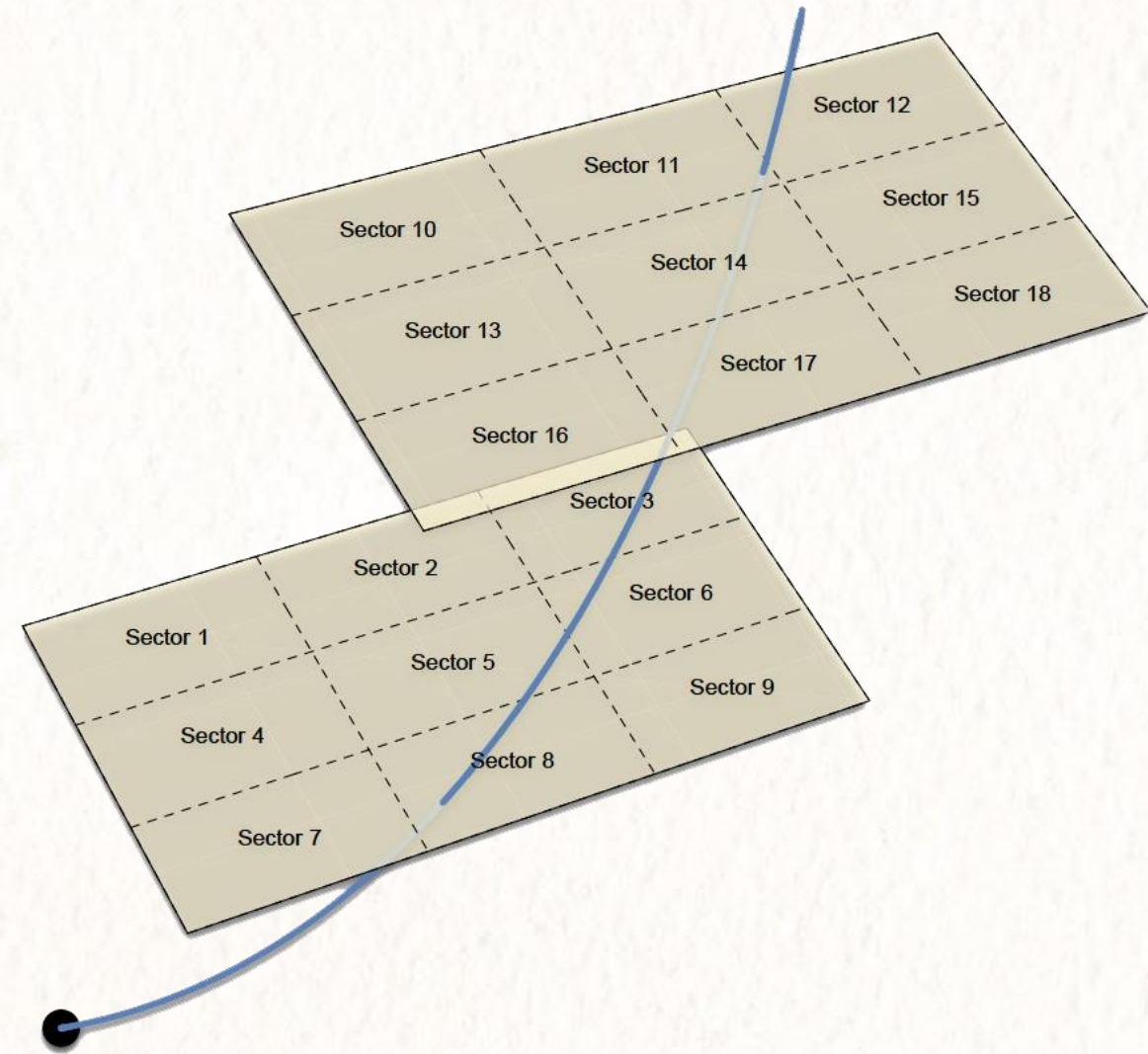
- Campione di eventi simulati che deve insegnare alla Sector Map le caratteristiche geometriche delle tracce attese in fase di presa dati:
  - momento (curvatura)
  - distribuzione spaziale
  - interazione con la materia (deviazione da elica)
- Scelta naturale:  $\sim 10^6 \Upsilon(4S)$

È sempre necessario fare ipotesi  
su modello di traccia

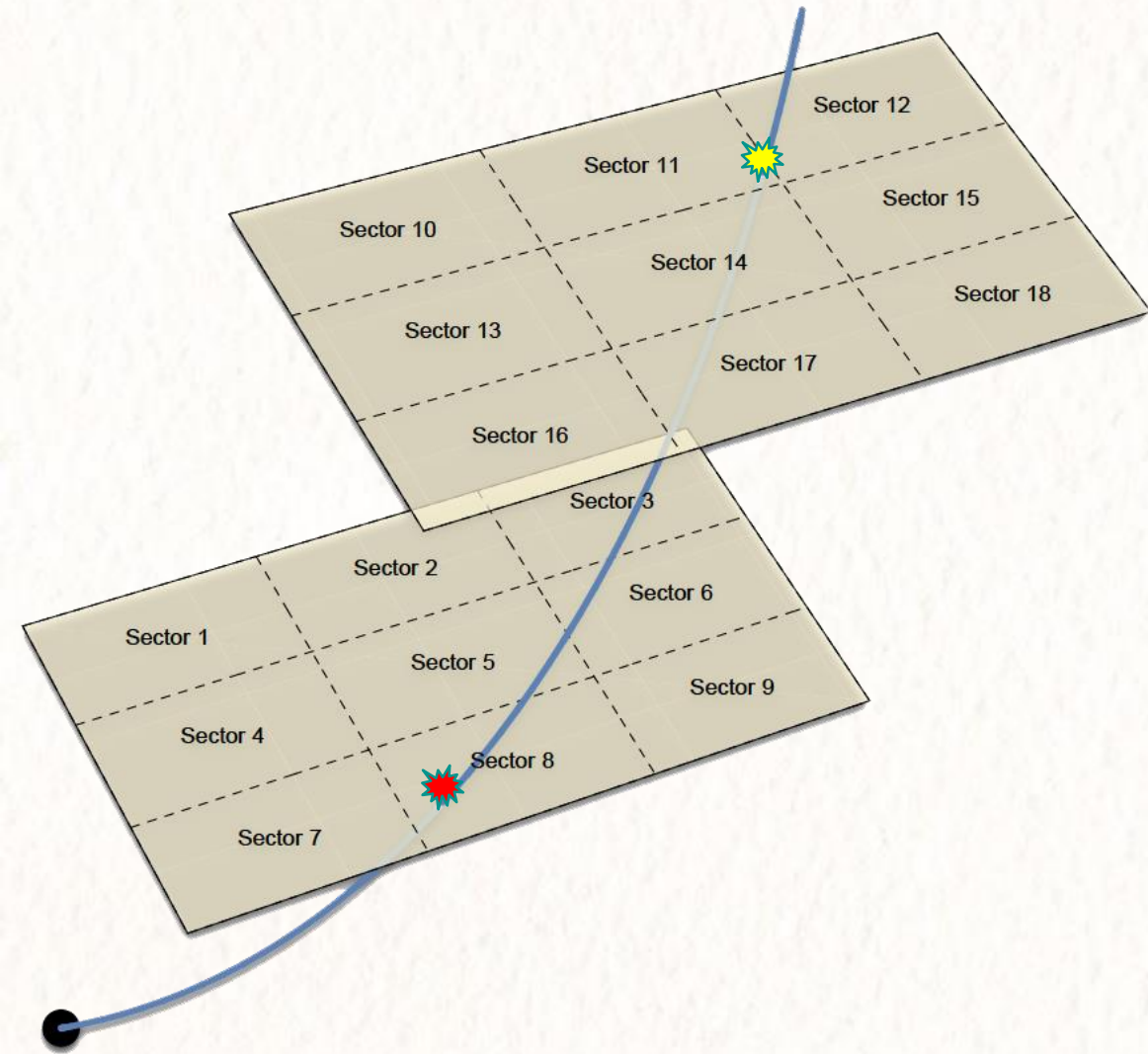


Usare le informazioni  
Monte Carlo

# 1 – Applicazione dei filtri



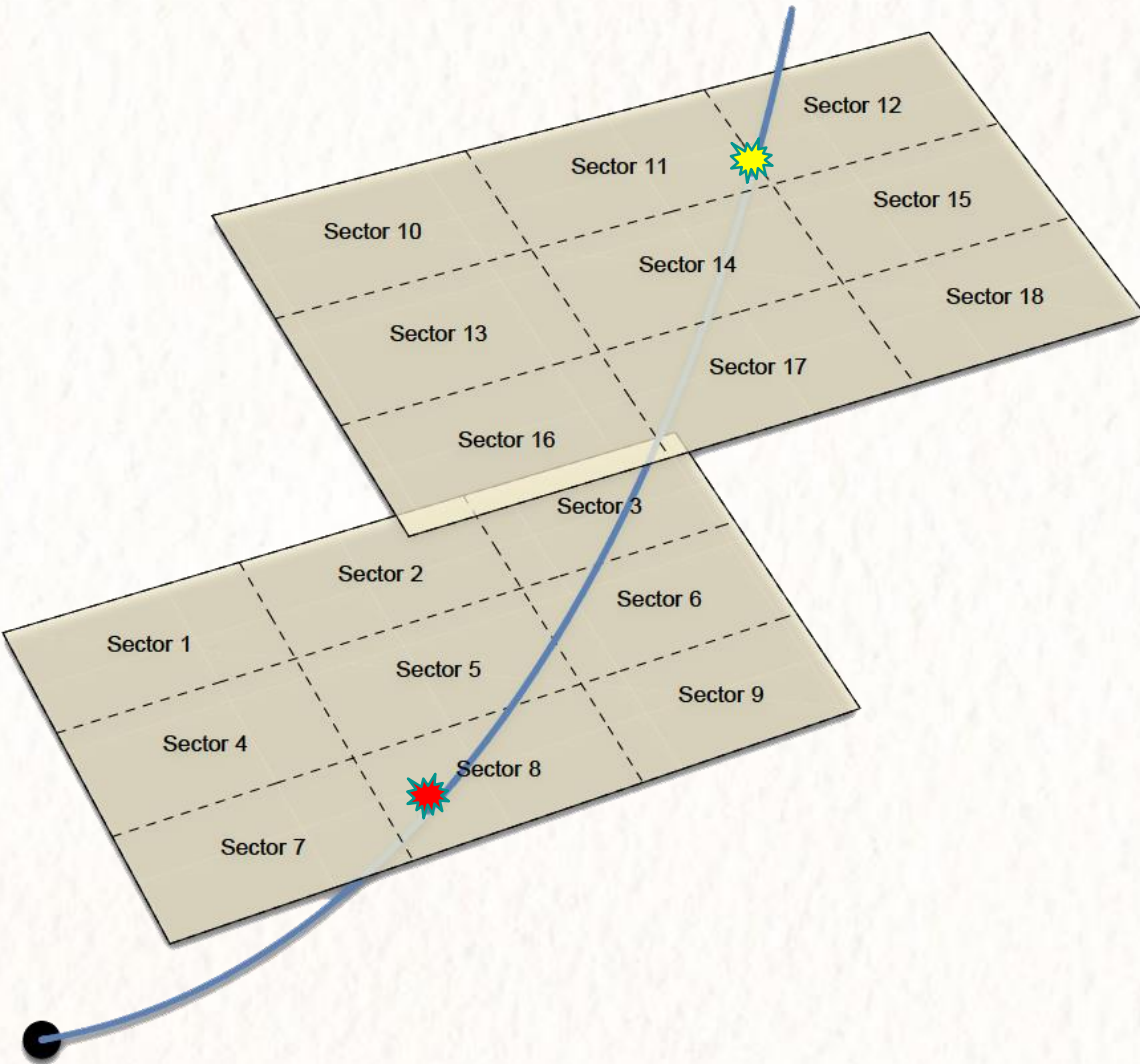
# 1 – Applicazione dei filtri



# 1 – Applicazione dei filtri

## 1. Individuazione settori

settore 8,  
settore 11



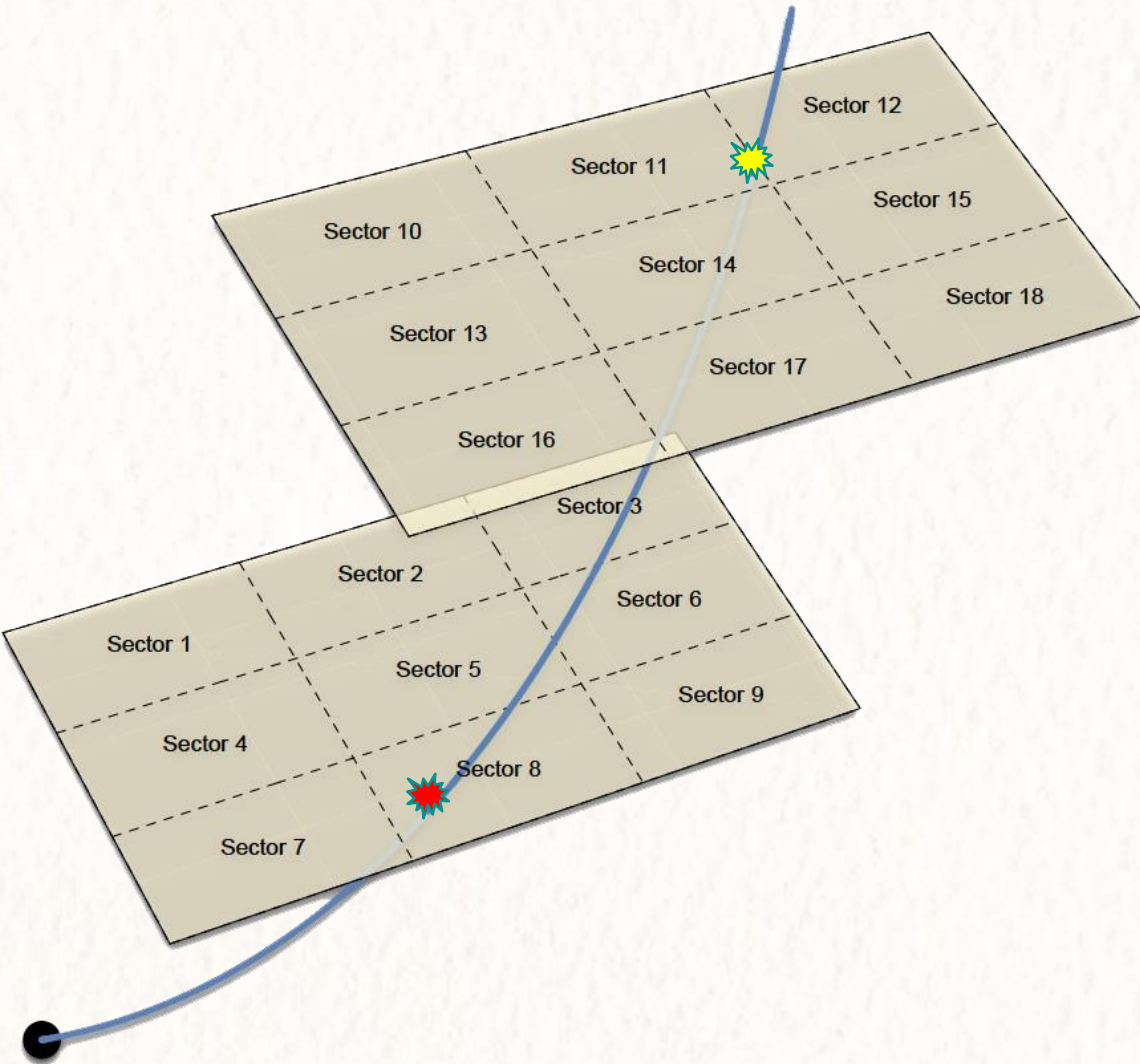
# 1 – Applicazione dei filtri

1. Individuazione settori

settore 8,  
settore 11

2. Individuazione amicizia

amicizia 8->11





# 1 – Applicazione dei filtri

1. Individuazione settori

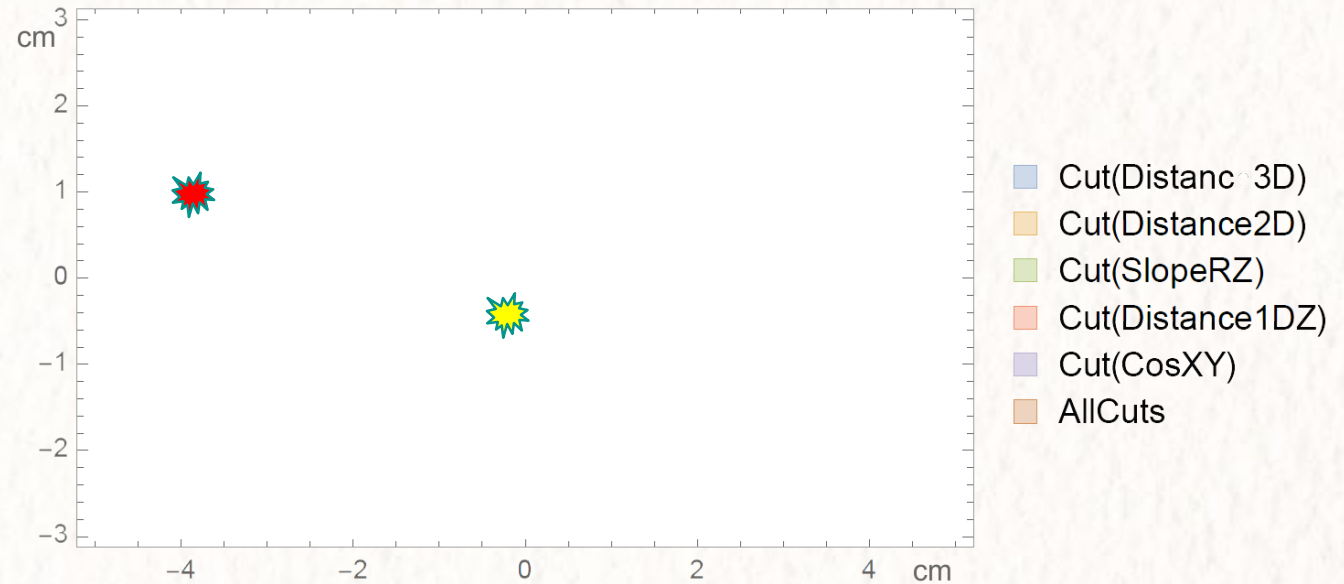
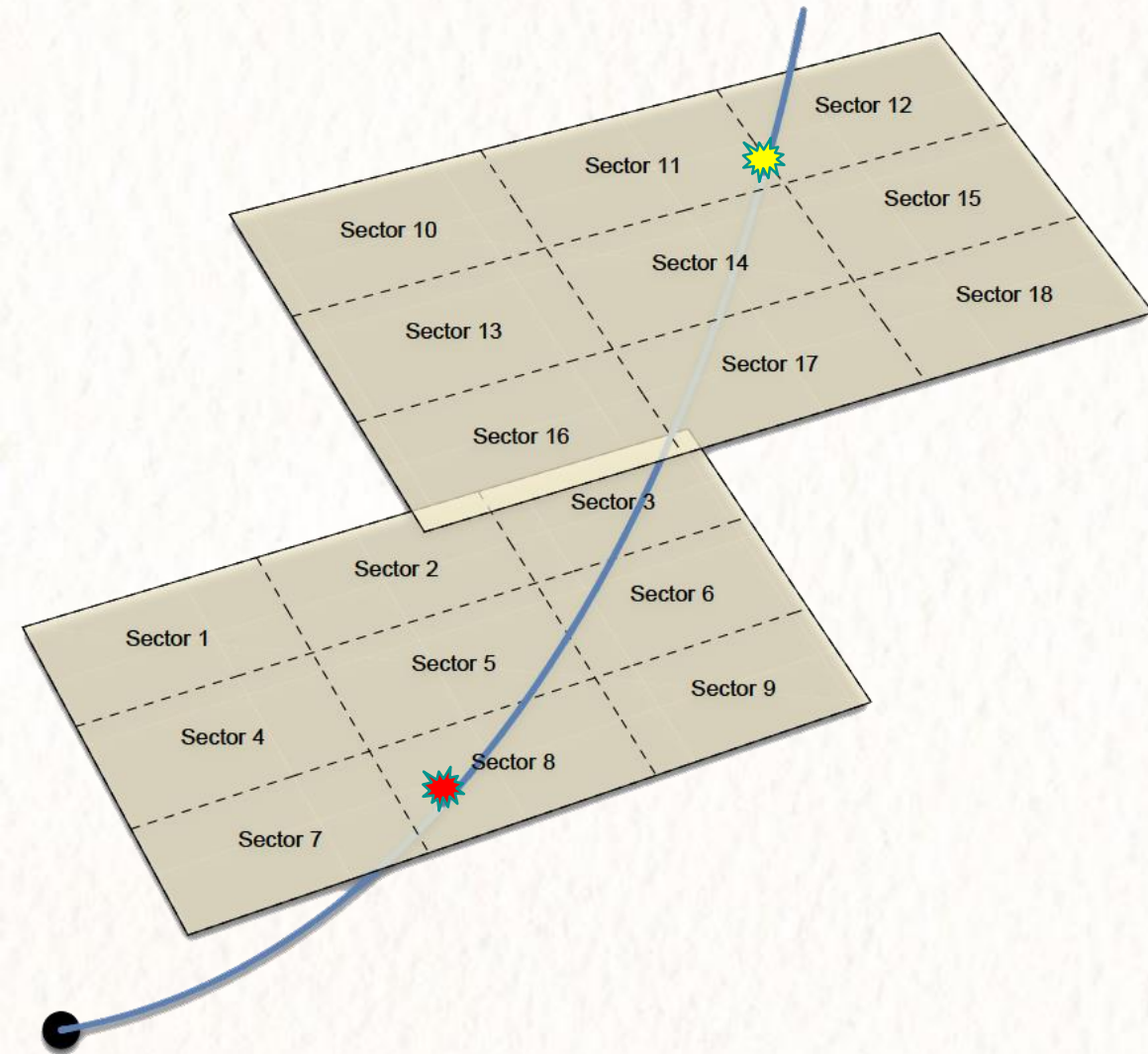
settore 8,  
settore 11

2. Individuazione amicizia

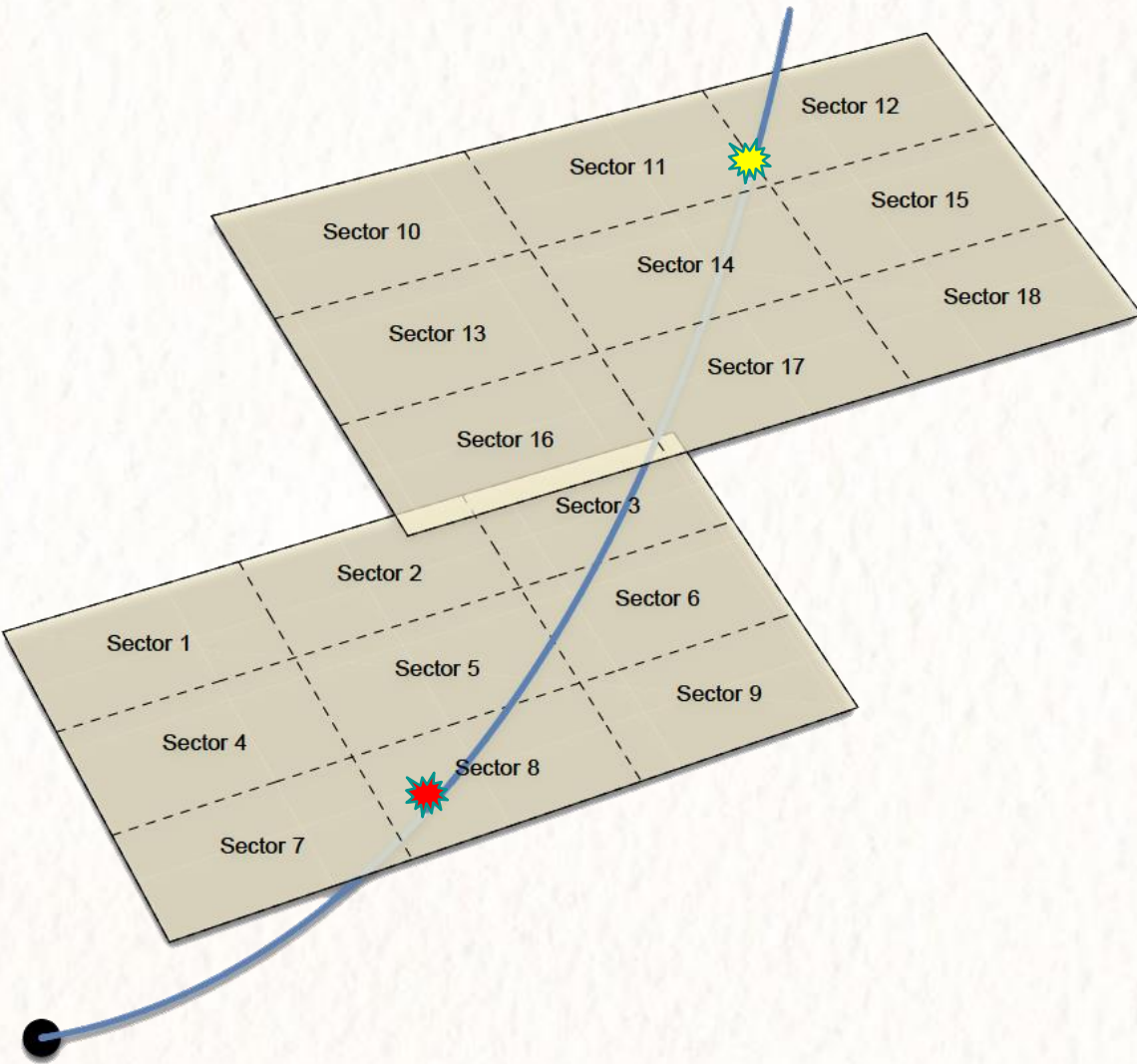
amicizia 8->11

3. Applicazione dei filtri a 2 hit

filtri8-11:



# 1 – Applicazione dei filtri



1. Individuazione settori

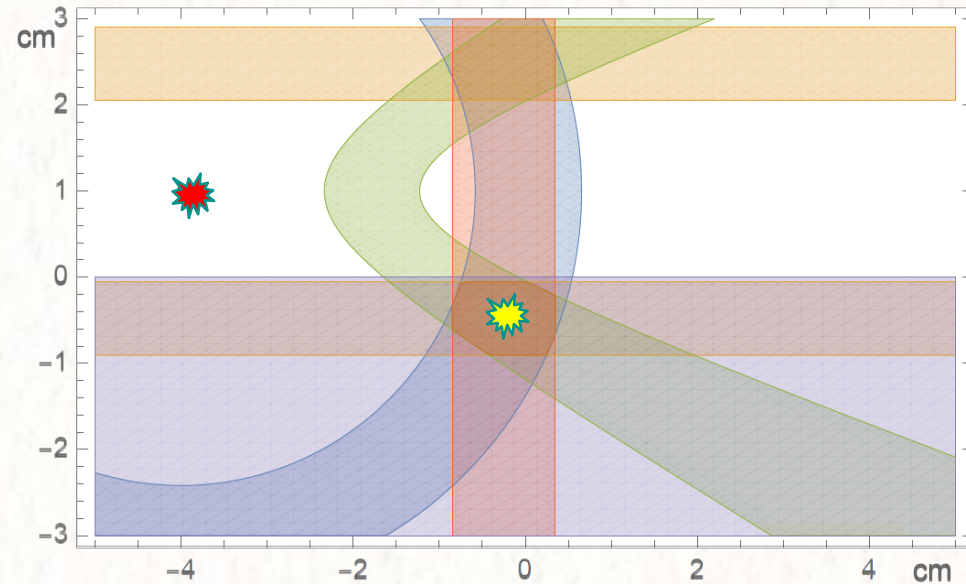
settore 8,  
settore 11

2. Individuazione amicizia

amicizia 8->11

3. Applicazione dei filtri a 2 hit

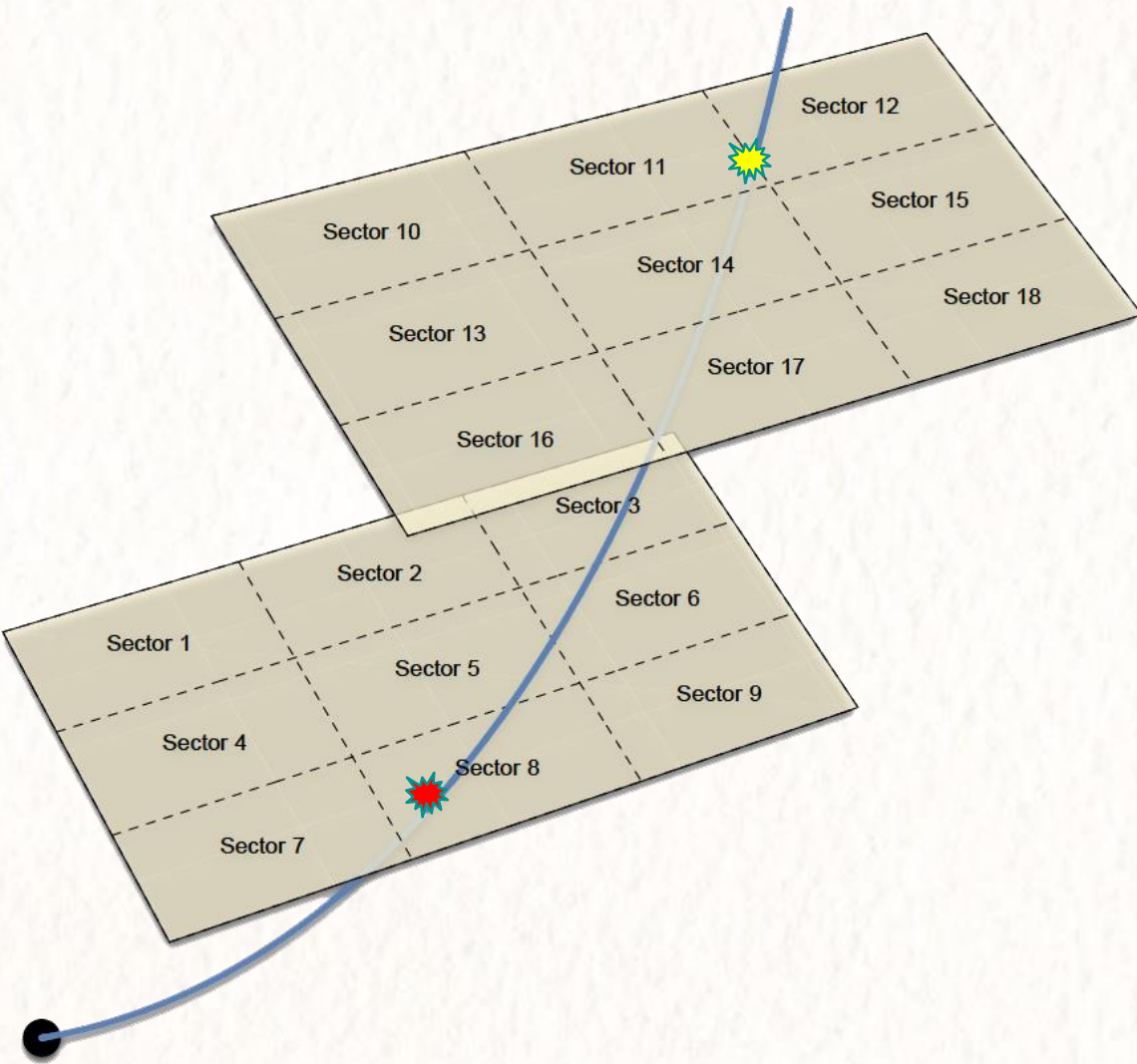
filtri8-11:



filtri  
superati!

- Cut(Distance3D)
- Cut(Distance2D)
- Cut(SlopeRZ)
- Cut(Distance1DZ)
- Cut(CosXY)
- AllCuts

# 1 – Applicazione dei filtri



1. Individuazione settori

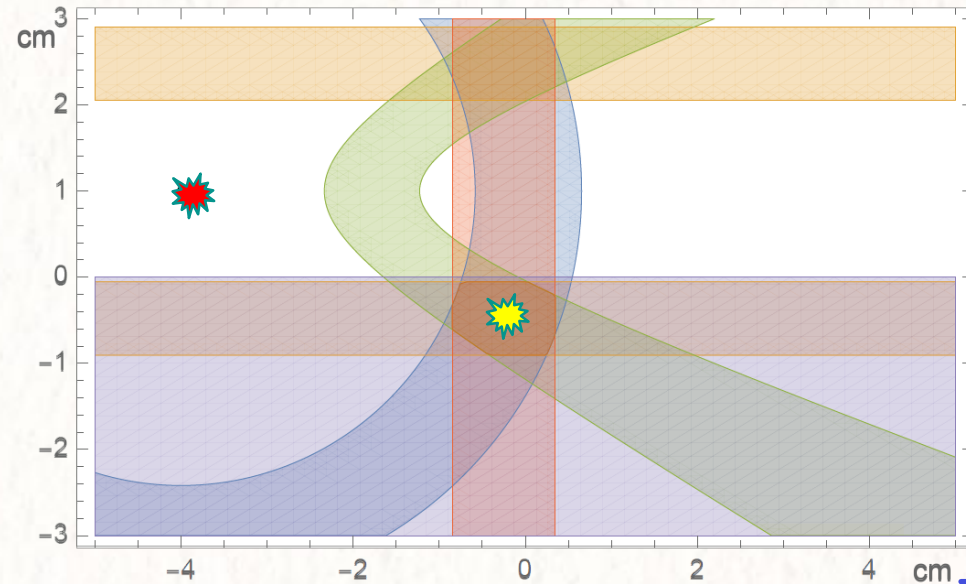
settore 8,  
settore 11

2. Individuazione amicizia

amicizia 8->11

3. Applicazione dei filtri a 2 hit

filtri8-11:



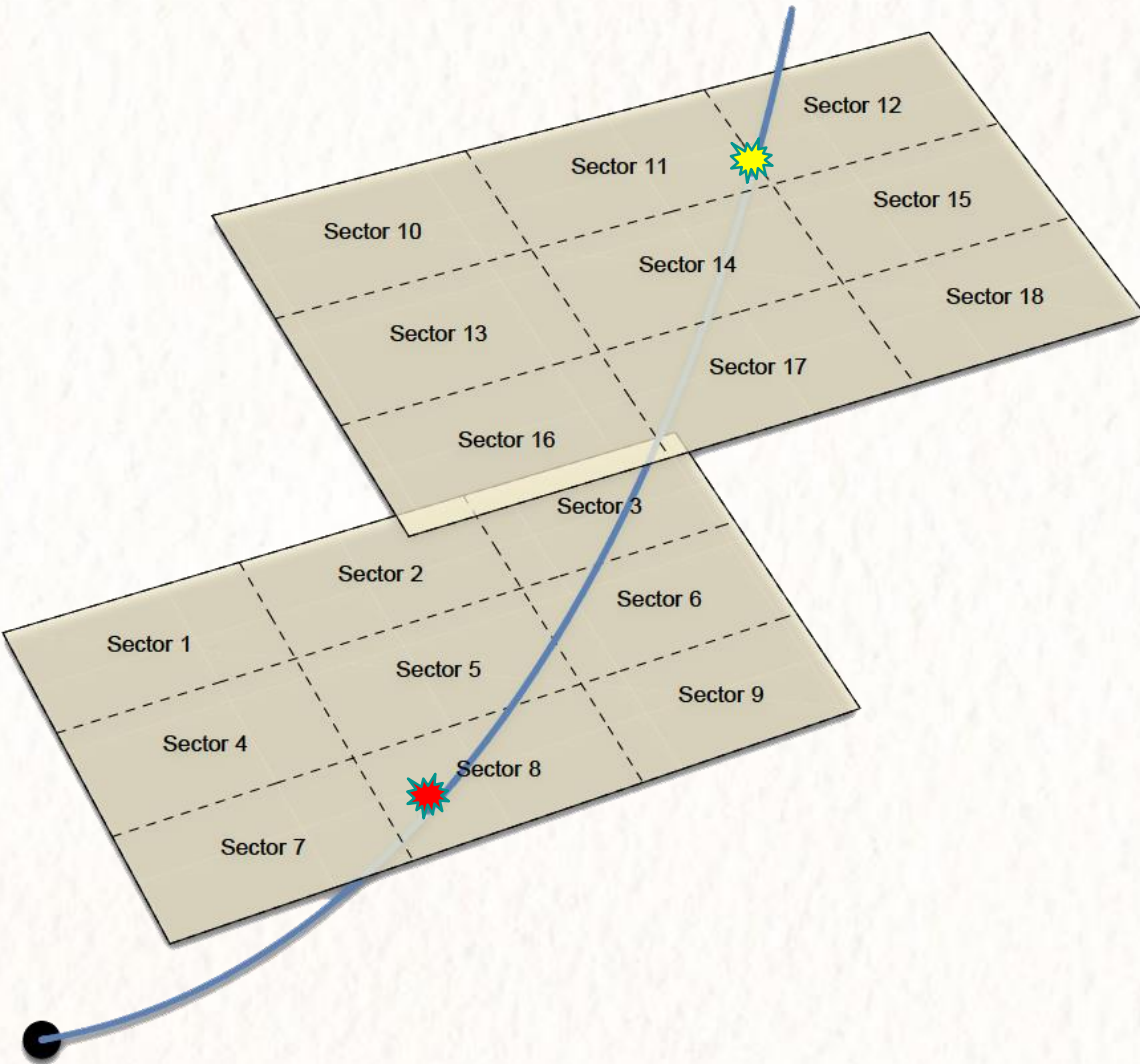
filtri  
superati!

- Cut(Distance3D)
- Cut(Distance2D)
- Cut(SlopeRZ)
- Cut(Distance1DZ)
- Cut(CosXY)
- AllCuts

4. Individuazione vicinanza

vicinanza  
8-11->11-24

# 1 – Applicazione dei filtri



1. Individuazione settori

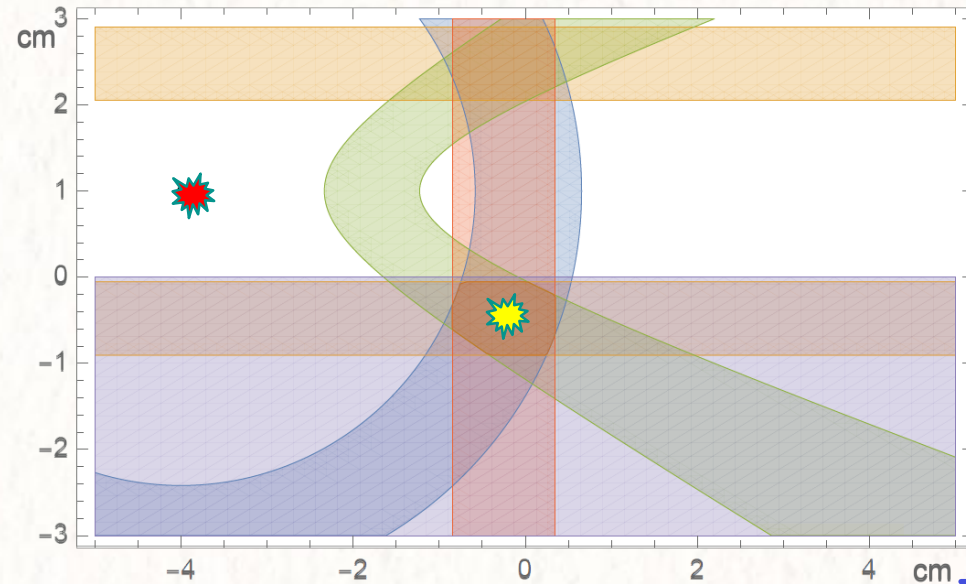
settore 8,  
settore 11

2. Individuazione amicizia

amicizia 8->11

3. Applicazione dei filtri a 2 hit

filtri8-11:



filtri  
superati!

- Cut(Distance 3D)
- Cut(Distance 2D)
- Cut(Slope RZ)
- Cut(Distance 1DZ)
- Cut(Cos XY)
- AllCuts

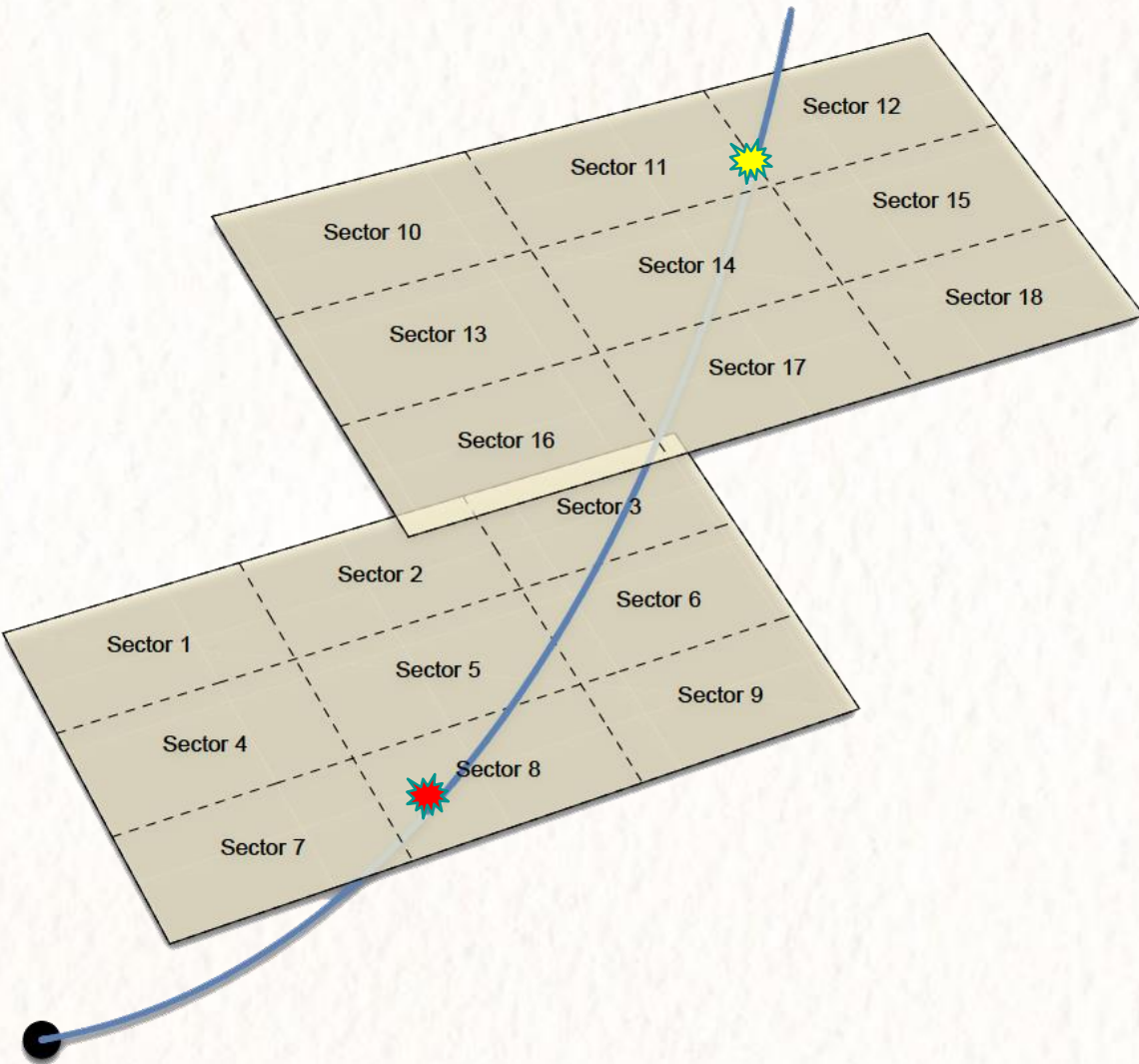
4. Individuazione vicinanza

vicinanza  
8-11->11-24

5. Applicazione filtri a 3 hit

filtri8-11-24:

# 1 – Applicazione dei filtri



1. Individuazione settori

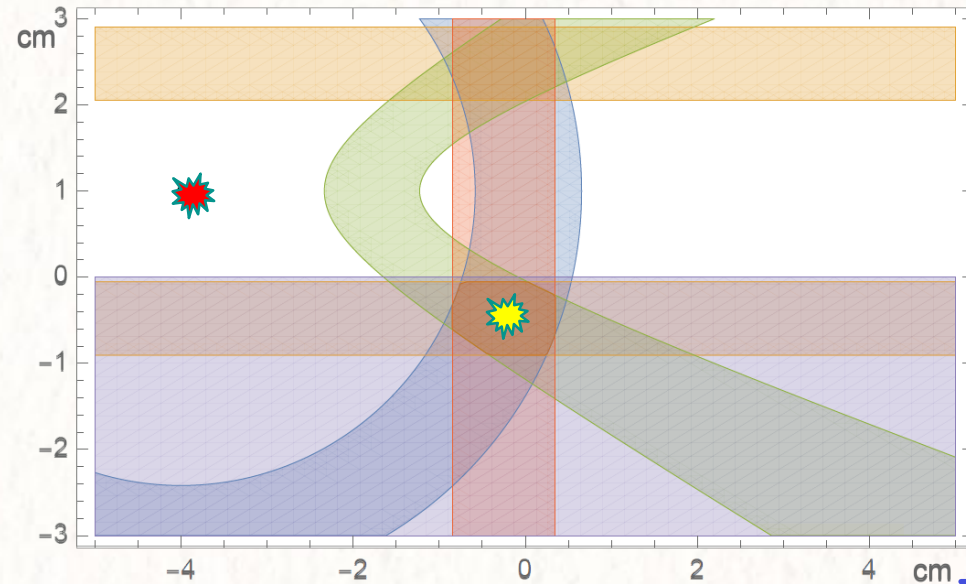
settore 8,  
settore 11

2. Individuazione amicizia

amicizia 8->11

3. Applicazione dei filtri a 2 hit

filtri8-11:



filtri  
superati!

- Cut(Distance 3D)
- Cut(Distance 2D)
- Cut(Slope RZ)
- Cut(Distance 1DZ)
- Cut(Cos XY)
- AllCuts

4. Individuazione vicinanza

vicinanza  
8-11->11-24

5. Applicazione filtri a 3 hit

filtri8-11-24:

6. Tripletta compatibile

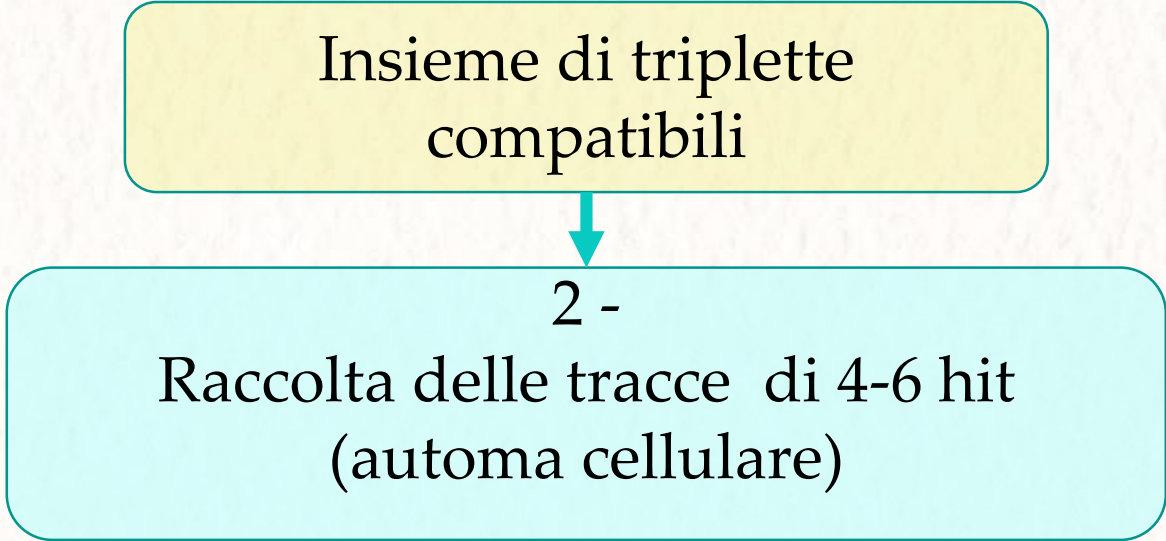
filtri  
superati! 21

# Passi successivi del VXDTF

Insieme di triplette  
compatibili

# Passi successivi del VXDTF

Insieme di triplette  
compatibili

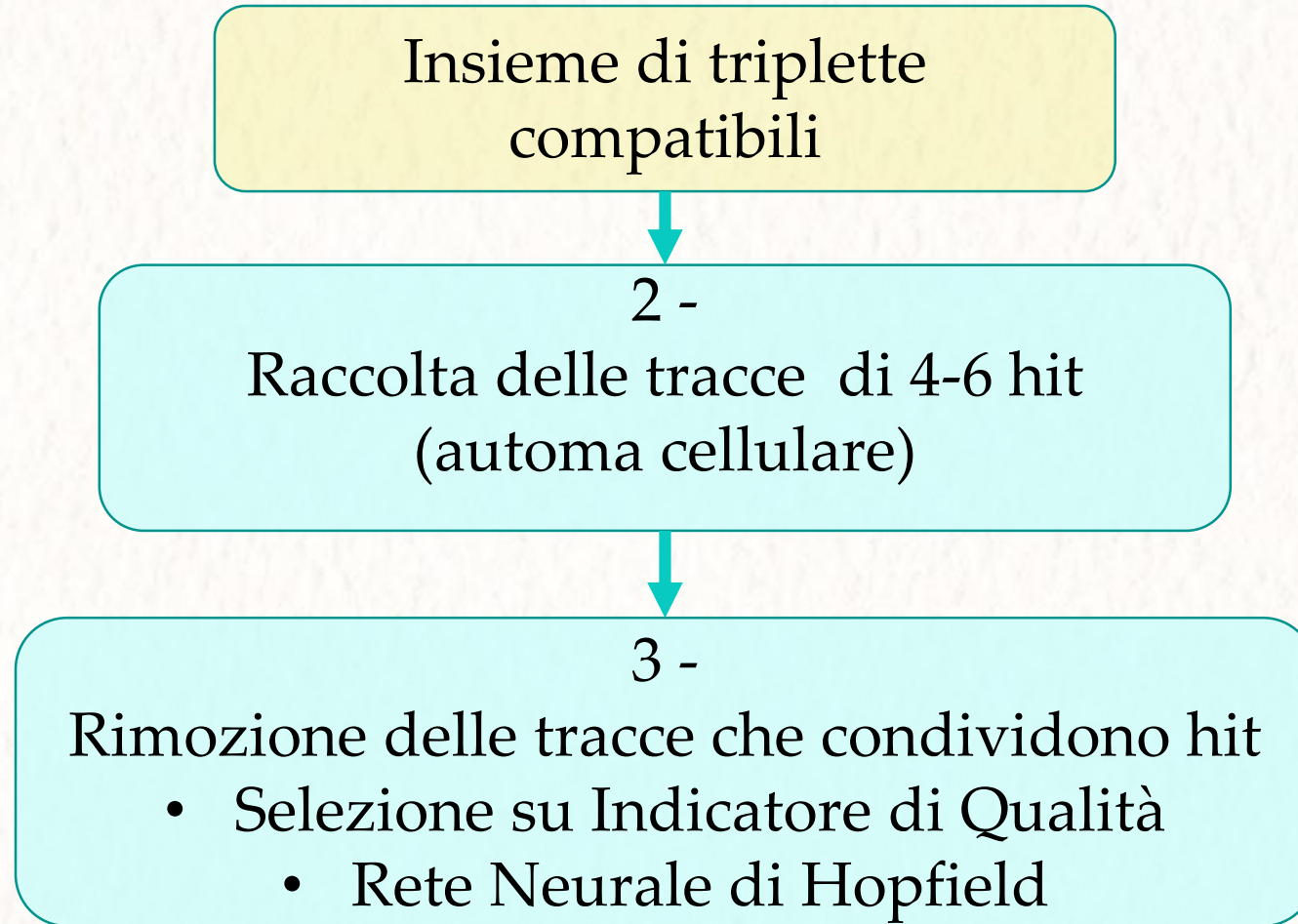


```
graph TD; A[Insieme di triplette compatibili] --> B[2 - Raccolta delle tracce di 4-6 hit (automa cellulare)];
```

2 -

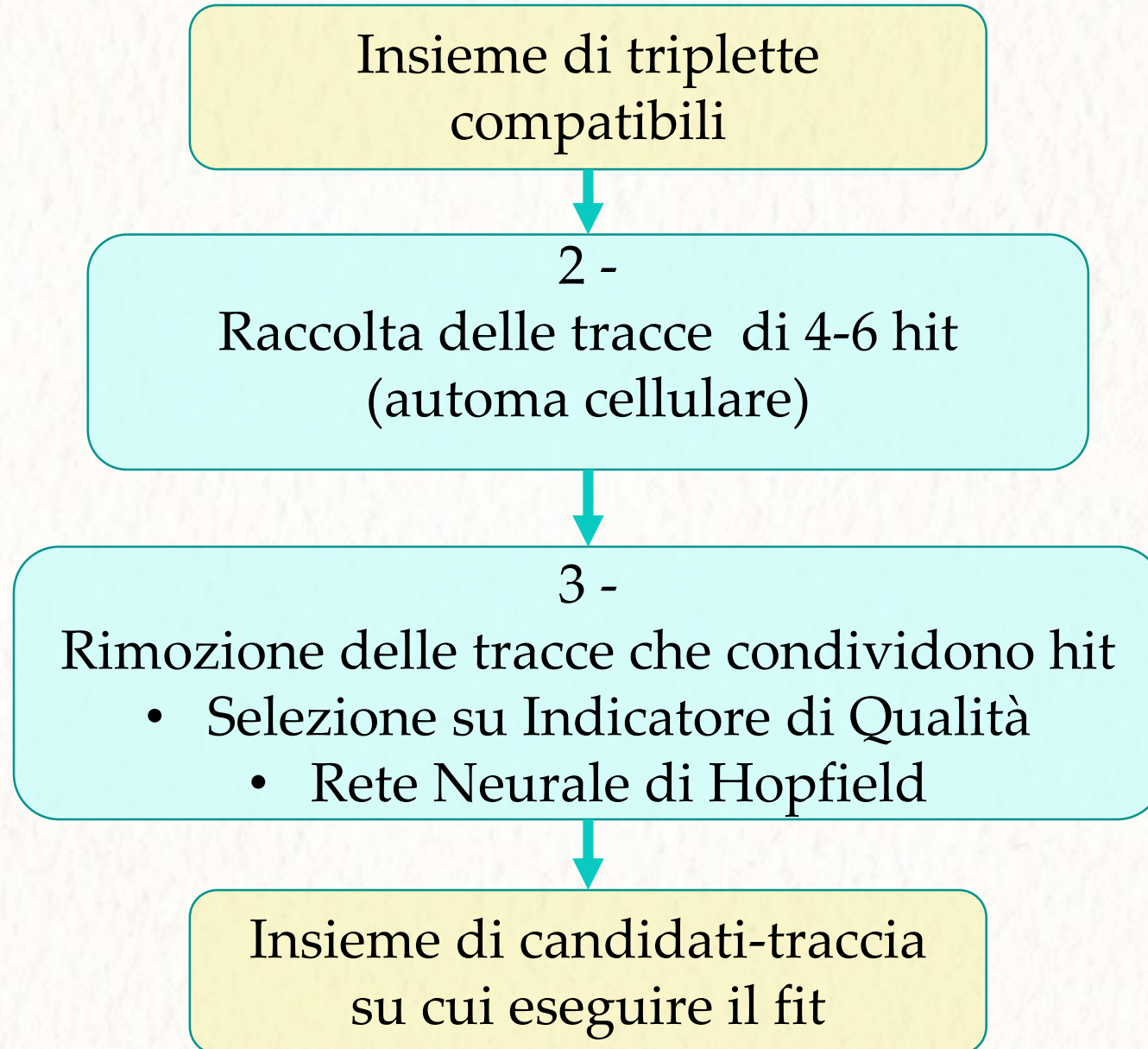
Raccolta delle tracce di 4-6 hit  
(automa cellulare)

# Passi successivi del VXDTF

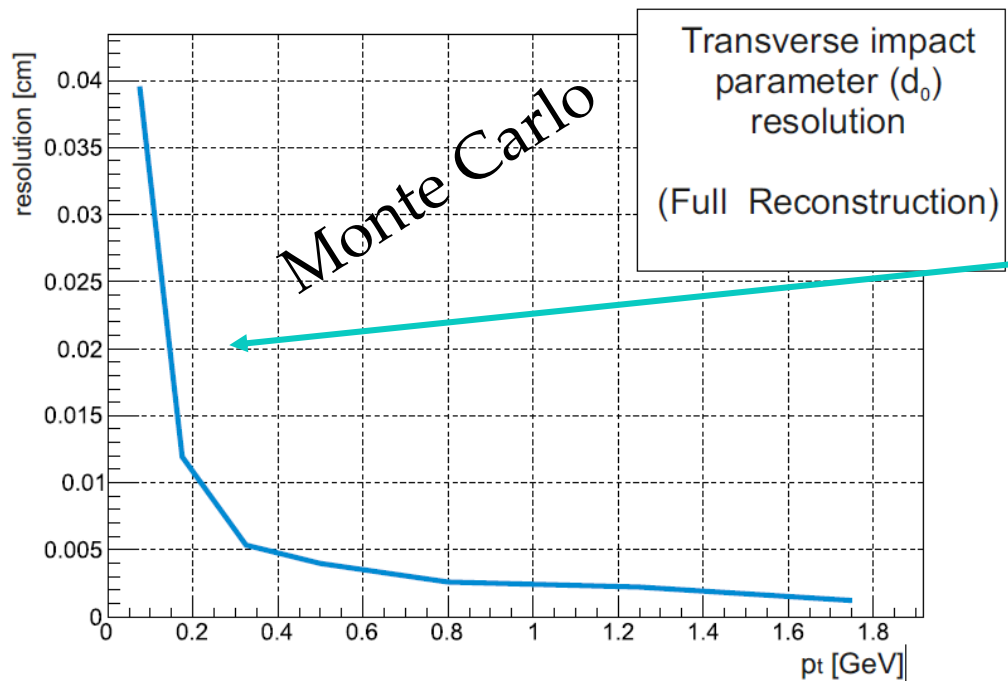
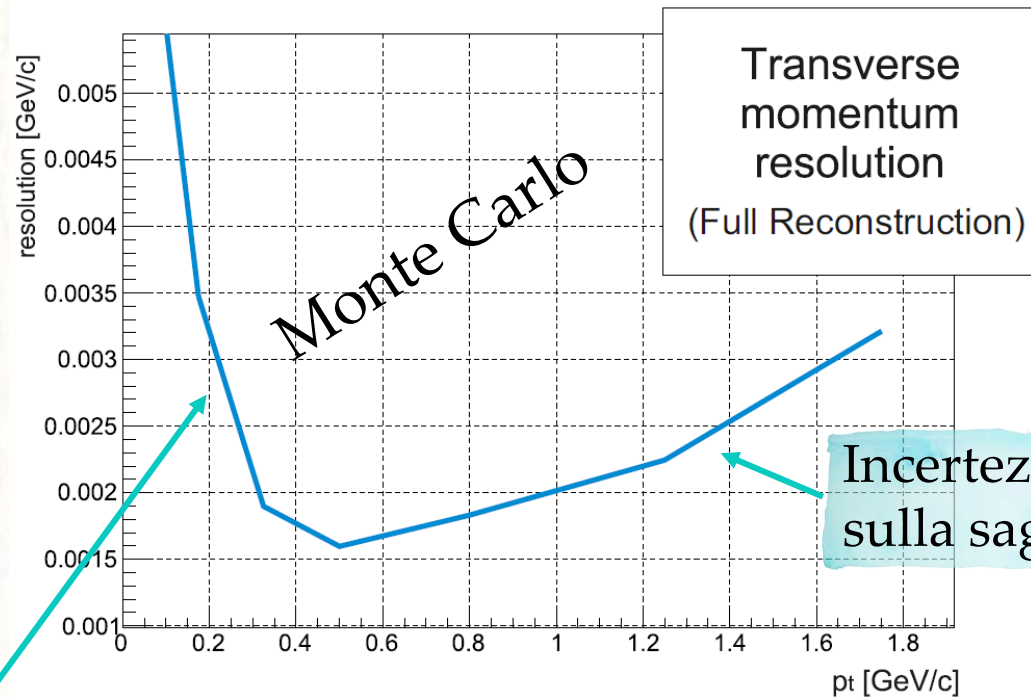
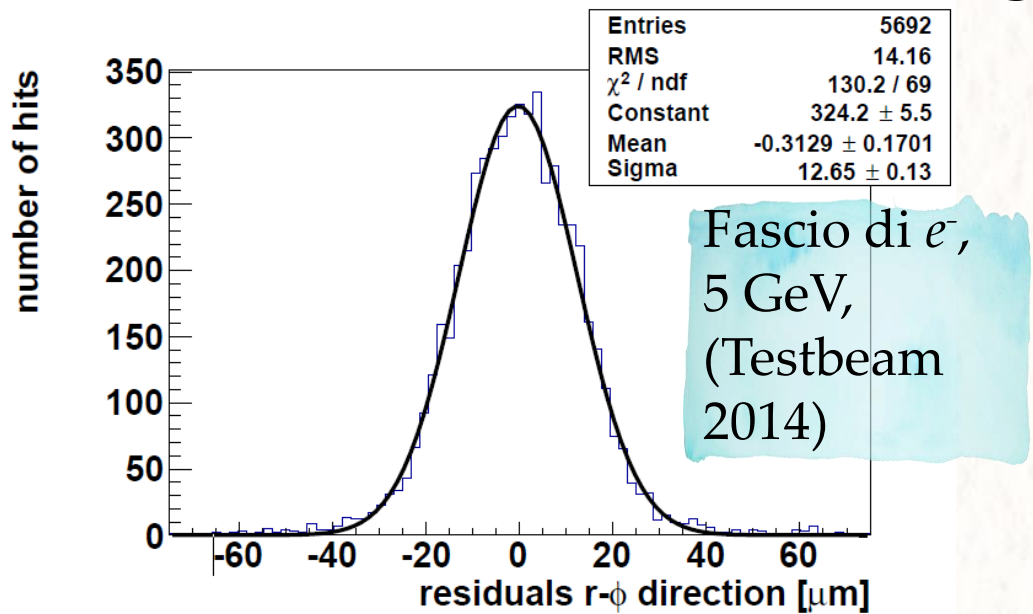




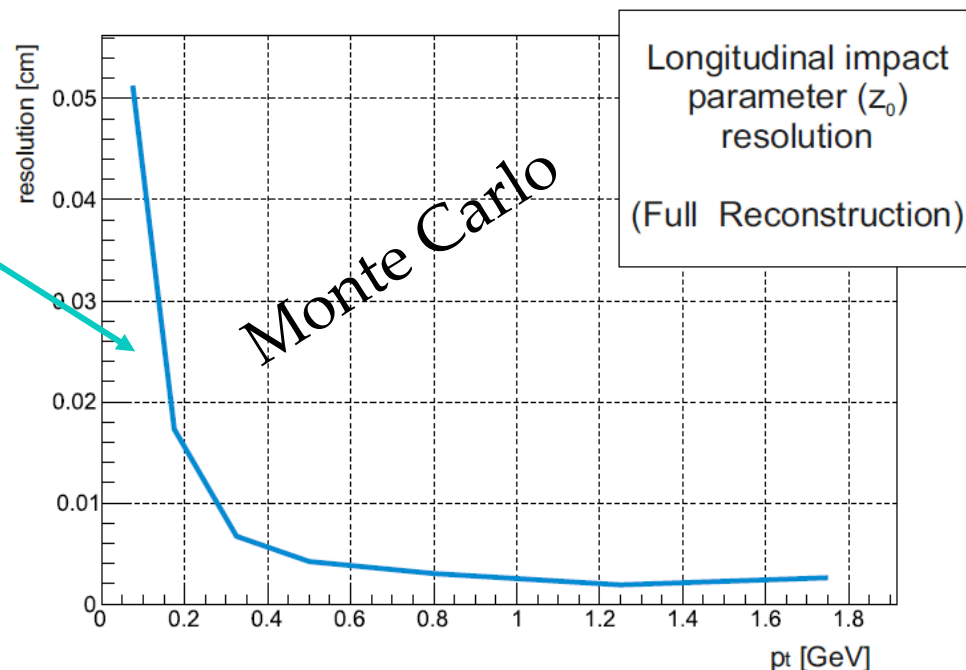
# Passi successivi del VXDTF



# Performance di tracking



Effetti del rivelatore e materiale



# Problemi del Training Sample e Strategia

## Precedente strategia

- riprodurre tutti i possibili eventi attesi
- Implementazione:  $\sim 10^6 \Upsilon(4S)$

Inclusi nel  
TS eventi  
molto rari

## Problemi

1. filtri meno stringenti → aumento delle false combinazione
2. Ricostruzione pattern non utilizzabili (causa effetti del materiale:  $\theta_0 \sim \frac{1}{p\sqrt{\sin \theta}}$ )  
→ efficienza di fit minore e rallentamento

# Problemi del Training Sample e Strategia

## Precedente strategia

- riprodurre tutti i possibili eventi attesi
- Implementazione:  $\sim 10^6 \Upsilon(4S)$



Inclusi nel TS eventi molto rari



## Problemi

1. filtri meno stringenti → aumento delle false combinazione
2. Ricostruzione pattern non utilizzabili (causa effetti del materiale:  $\theta_0 \sim \frac{1}{p\sqrt{\sin \theta}}$ )  
→ efficienza di fit minore e rallentamento

## Obiettivo

Rimuovere dal TS eventi rari da interazione con il materiale (scattering multiplo) con traiettorie non elicoidali (*tracce catastrofiche*)

## Strategia

- **Fase A:** definire lista di tagli da applicare al TS
  - Definizione automatizzata: insegnato al VXDTF2 a individuare le tracce catastrofiche usando eventi simulati
- **Fase B:** filtrare le tracce del TS applicando questi tagli

# Fase A, Passo 1 – scelta delle variabili

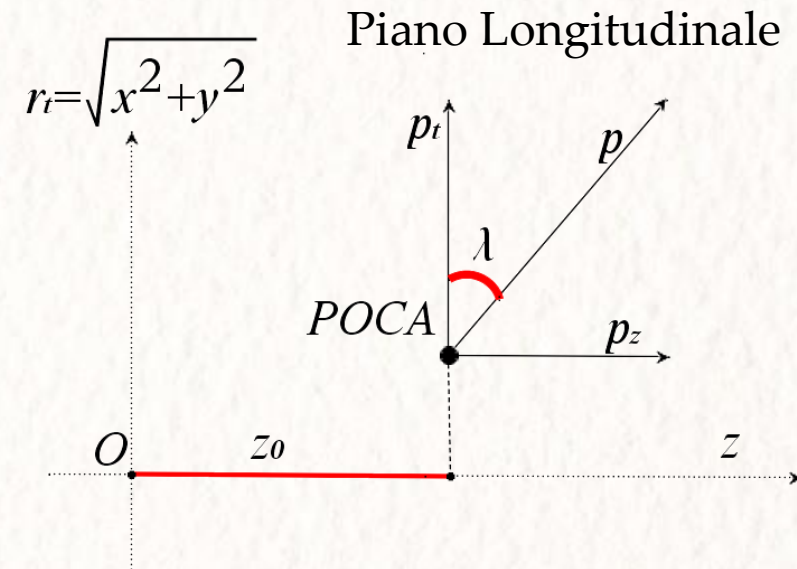
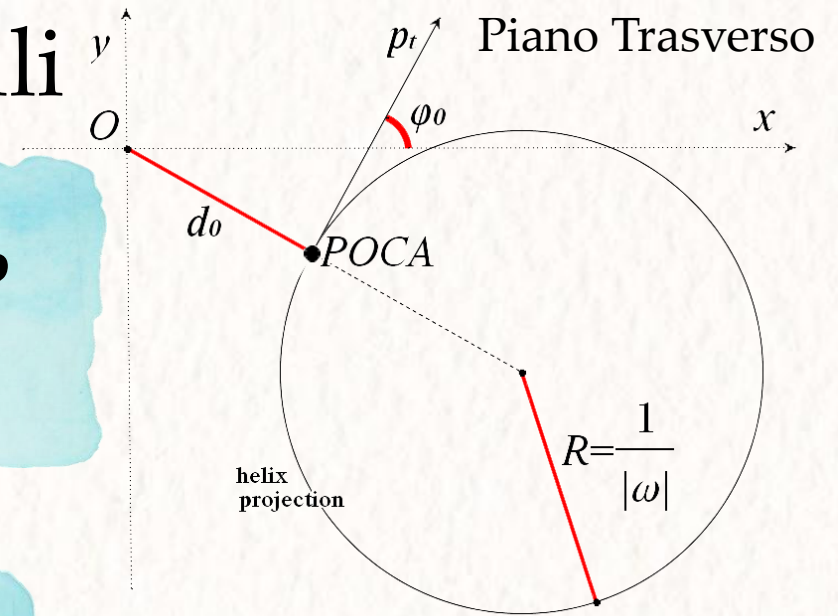
## Parametrizzazione delle eliche

- Parametri:  $\omega, d_0, \varphi_0, z_0 \tan \lambda$  esprimibili in funzione di  $\mathbf{x}, \mathbf{p}$
- Per ogni MC-hit  $\mathbf{x}, \mathbf{p}$  noti  $\rightarrow$  noti parametri per ogni hit lungo la traccia

## Identificare le tracce catastrofiche

- Parametri costanti lungo la traccia per eliche ideali
- Forti variazioni puntuali lungo la traccia indicano interazione anomala con il materiale
- $\Delta X$  = differenza di  $X$  ( $X$ =parametro di traccia) nell'attraversamento di un layer del VXD

  
 $\Delta X(|\mathbf{p}|, \theta, \text{primo layer}, \text{secondo layer})$



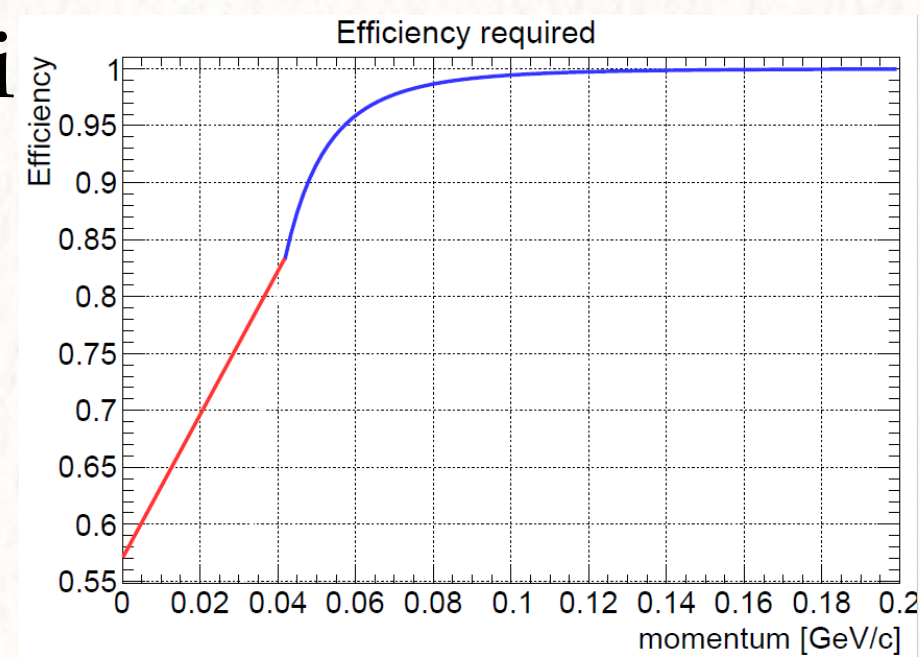
# Fase A, Passo 2 – definizione dei tagli

## Richieste globali

- Almeno 3 layer di SVD con almeno un hit
- $|d_0| < 1$  cm,  $|z_0| < 1$  cm per ogni hit

## *Cut-function*

- Richiesta di efficienza fissata sul singolo taglio  $\epsilon_{\text{req}}$



# Fase A, Passo 2 – definizione dei tagli

## Richieste globali

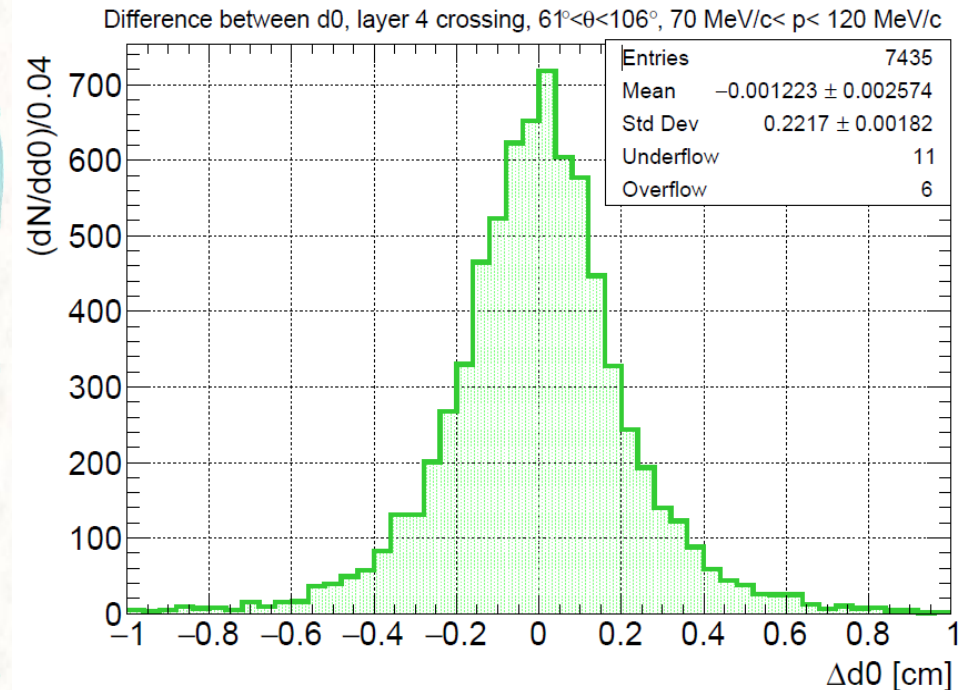
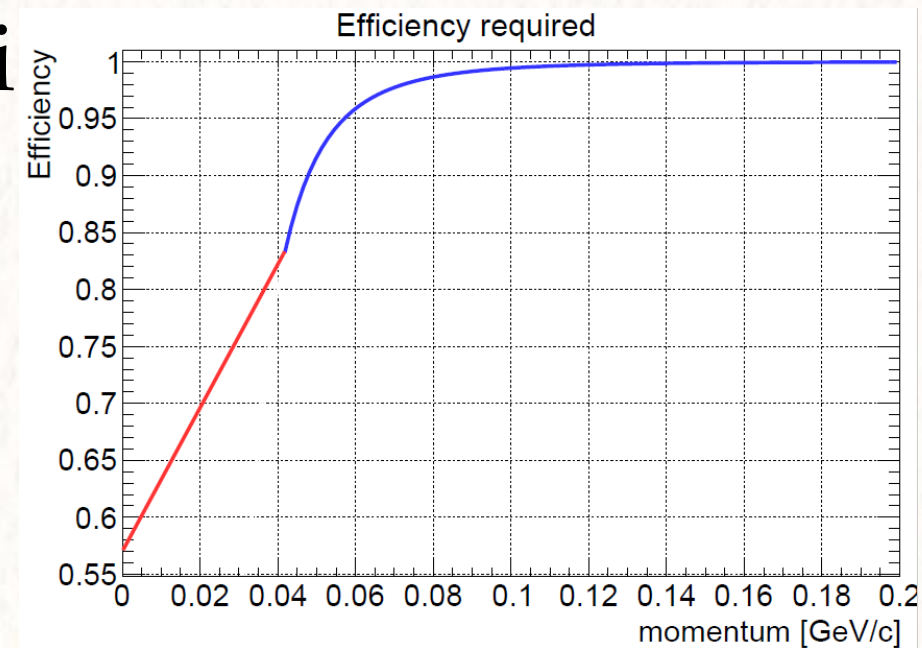
- Almeno 3 layer di SVD con almeno un hit
- $|d_0| < 1$  cm,  $|z_0| < 1$  cm per ogni hit

## Cut-function

- Richiesta di efficienza fissata sul singolo taglio  $\varepsilon_{\text{req}}$

## Insegnare a riconoscere le tracce catastrofiche

- Simulati  $10^6$  eventi  $\Upsilon(4S)$
- Valutate distribuzioni  $\Delta X(|\mathbf{p}|, \theta, l_1, l_2)$



# Fase A, Passo 2 – definizione dei tagli

## Richieste globali

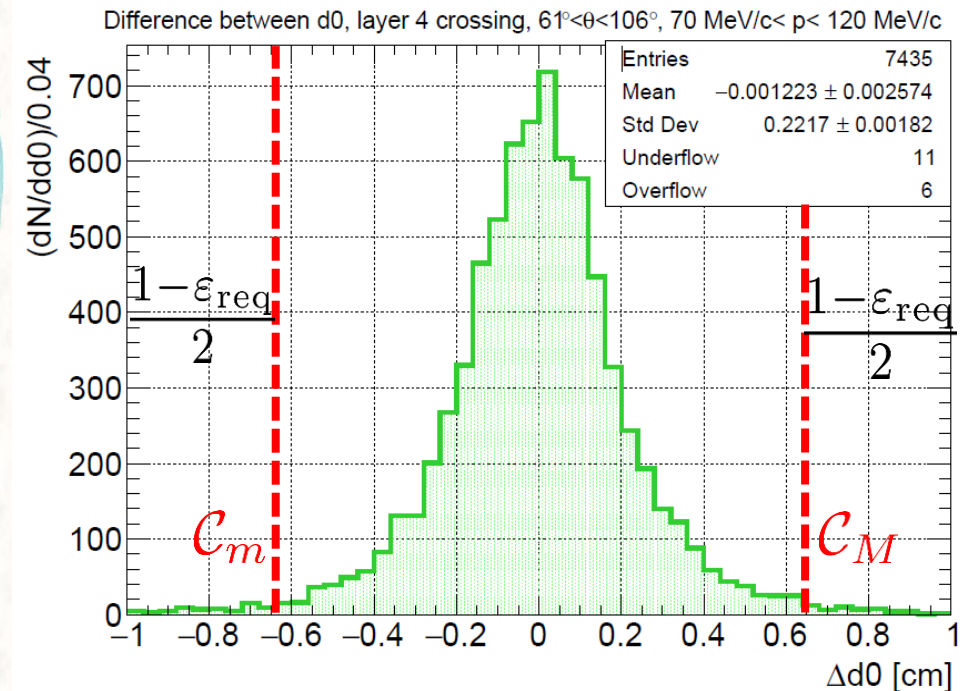
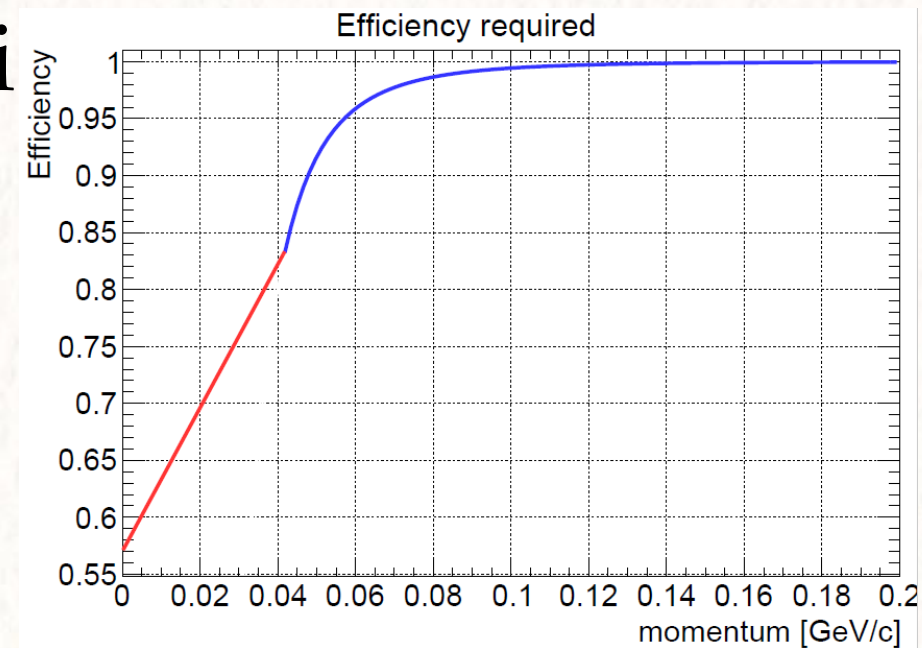
- Almeno 3 layer di SVD con almeno un hit
- $|d_0| < 1$  cm,  $|z_0| < 1$  cm per ogni hit

## Cut-function

- Richiesta di efficienza fissata sul singolo taglio  $\epsilon_{\text{req}}$

## Insegnare a riconoscere le tracce catastrofiche

- Simulati  $10^6$  eventi  $\Upsilon(4S)$
- Valutate distribuzioni  $\Delta X(|\mathbf{p}|, \theta, l_1, l_2)$
- Identificate code delle distribuzioni
- Definiti, per ogni  $\Delta X$ , le soglie Max e min:  $C_m, C_M$





# Fase A, Passo 2 – definizione dei tagli

## Richieste globali

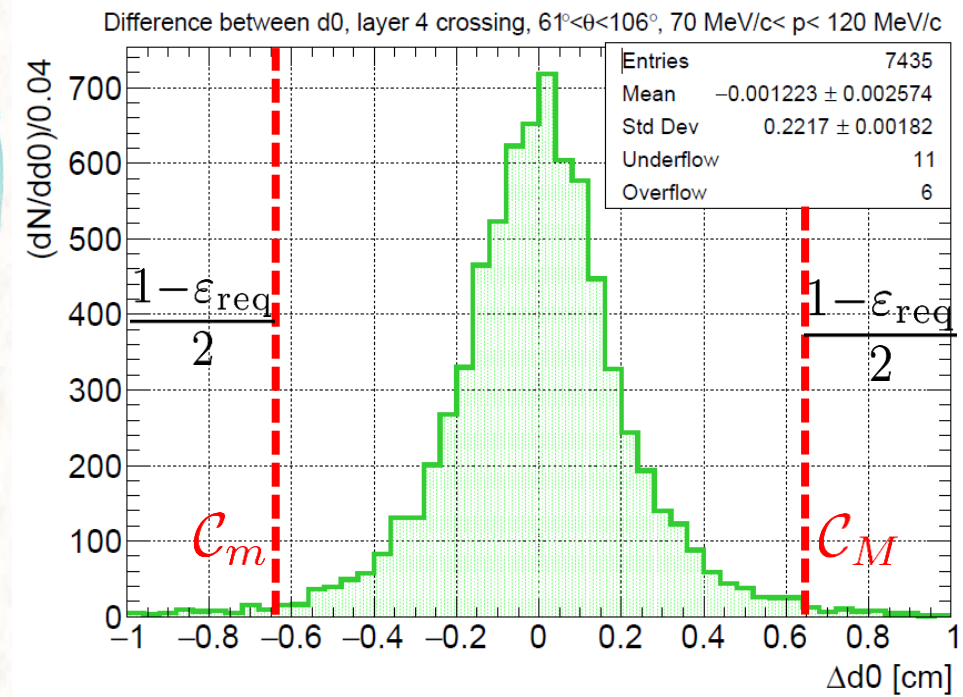
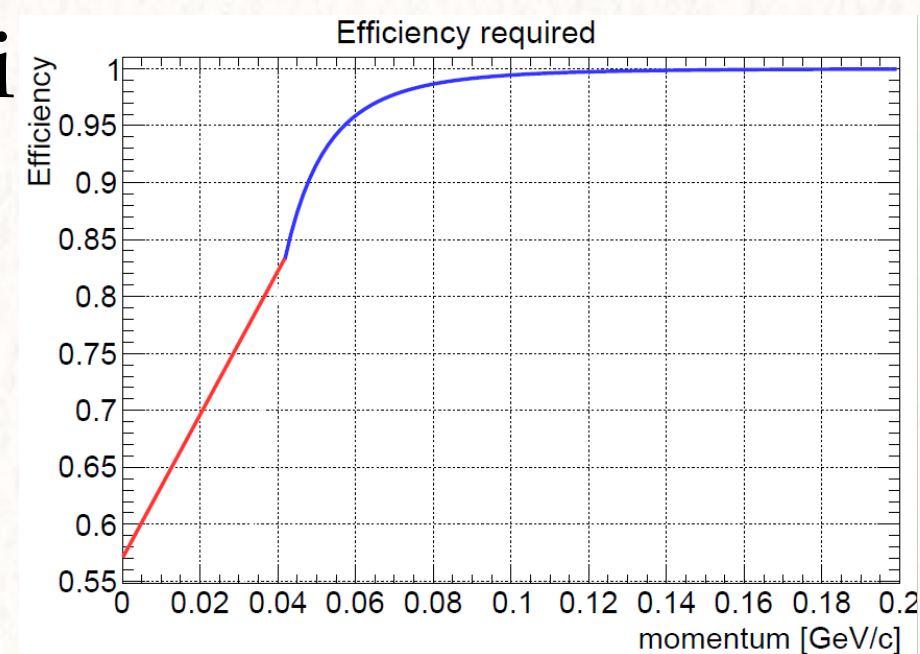
- Almeno 3 layer di SVD con almeno un hit
- $|d_0| < 1$  cm,  $|z_0| < 1$  cm per ogni hit

## Cut-function

- Richiesta di efficienza fissata sul singolo taglio  $\epsilon_{\text{req}}$

## Insegnare a riconoscere le tracce catastrofiche

- Simulati  $10^6$  eventi  $\Upsilon(4S)$
- Valutate distribuzioni  $\Delta X(|\mathbf{p}|, \theta, l_1, l_2)$
- Identificate code delle distribuzioni
- Definiti, per ogni  $\Delta X$ , le soglie Max e min:  $C_m, C_M$
- Costruiti istogrammi 2D  $|\mathbf{p}| \times \theta$  con  $C_m, C_M$
- fit con:  $\Delta X(p, \theta, l_1, l_2) \sim \frac{A(X, l_1, l_2)}{(p)^{B(X, l_1, l_2)} \sqrt{\sin \theta}} + C(X, l_1, l_2)$
- Memorizzati parametri A, B, C per ogni coppia di layer e parametro X



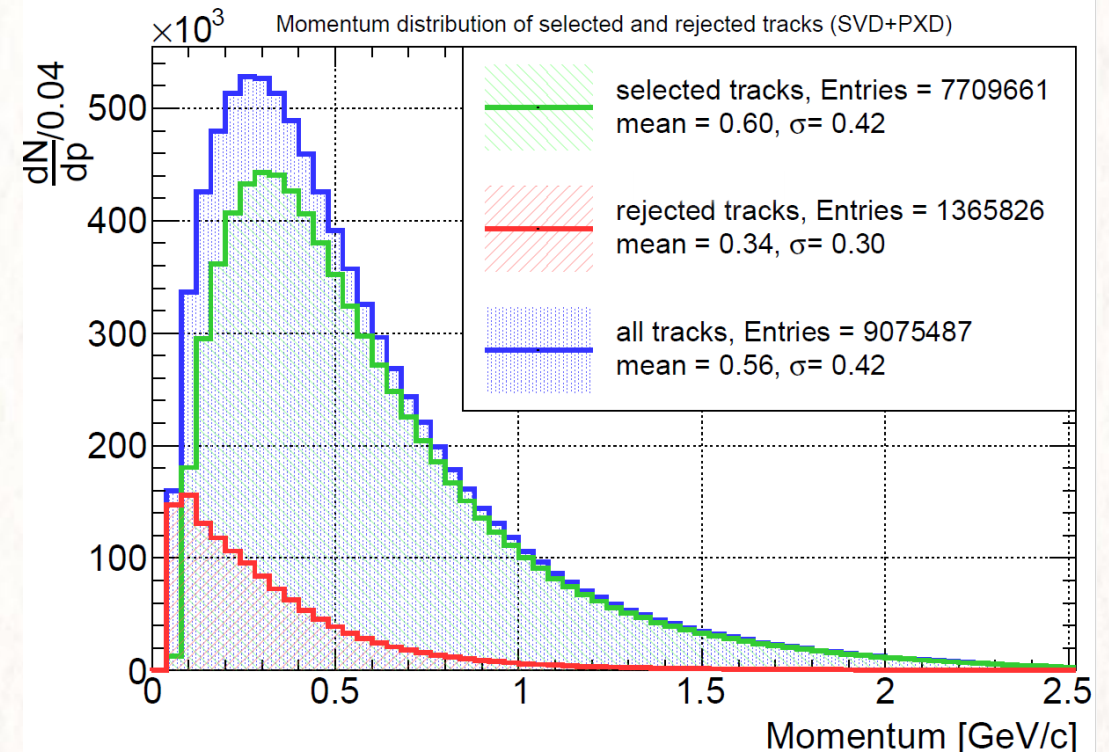
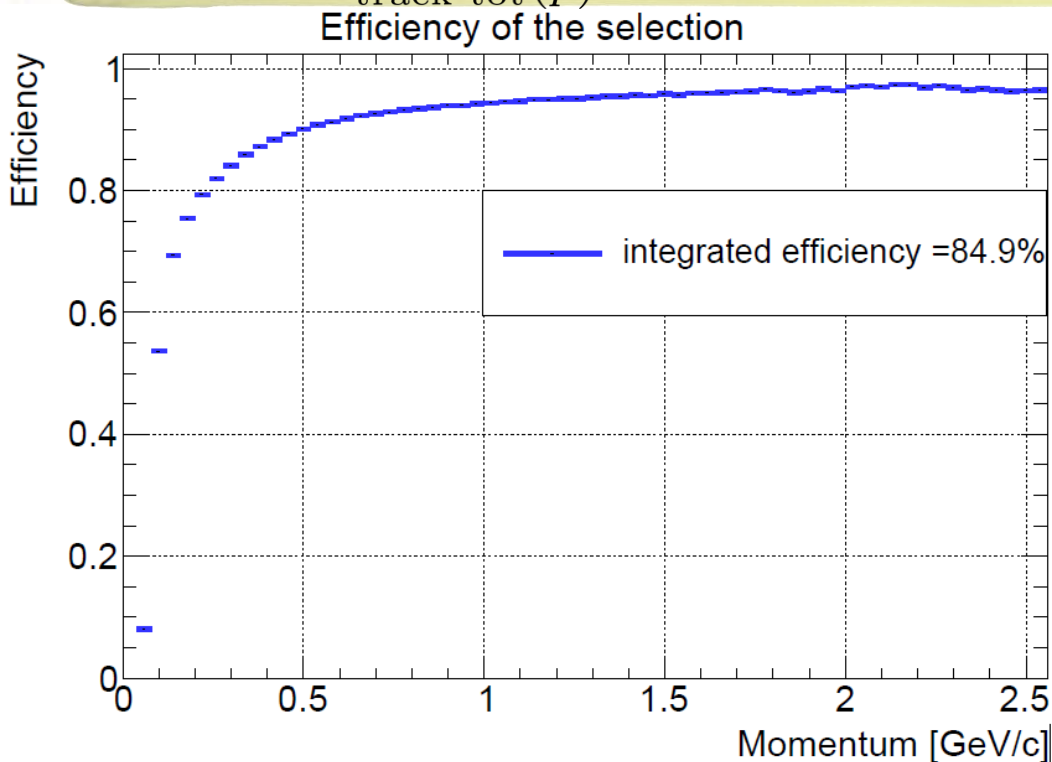
# Fase B – applicazione dei tagli

- Ogni traccia del TS passata a un filtro integrato nella fase di allenamento
- Applicate le richieste globali
- Valutato  $\Delta X(|\mathbf{p}|, \theta, l_1, l_2)$  per ogni coppia di hit (5 tagli a coppia)
- A partire dai parametri A,B,C memorizzati valutato il taglio:  $\Delta X_{cut} = \frac{A}{(p)^B \sqrt{\sin \theta}} + C$
- Per ogni coppia di hit richiesto:  $|\Delta X| < \Delta X_{cut}$

# Fase B – applicazione dei tagli

- Ogni traccia del TS passata a un filtro integrato nella fase di allenamento
- Applicate le richieste globali
- Valutato  $\Delta X(|\mathbf{p}|, \theta, l_1, l_2)$  per ogni coppia di hit (5 tagli a coppia)
- A partire dai parametri A,B,C memorizzati valutato il taglio:  $\Delta X_{cut} = \frac{A}{(p)^B \sqrt{\sin \theta}} + C$
- Per ogni coppia di hit richiesto:  $|\Delta X| < \Delta X_{cut}$

$$\varepsilon(p) = \frac{N_{\text{track pass}}(p)}{N_{\text{track tot}}(p)} \rightarrow 15.1\% \text{ delle tracce rimosse (10\% da richieste globali)}$$



# Risultati – Tracking (senza background)

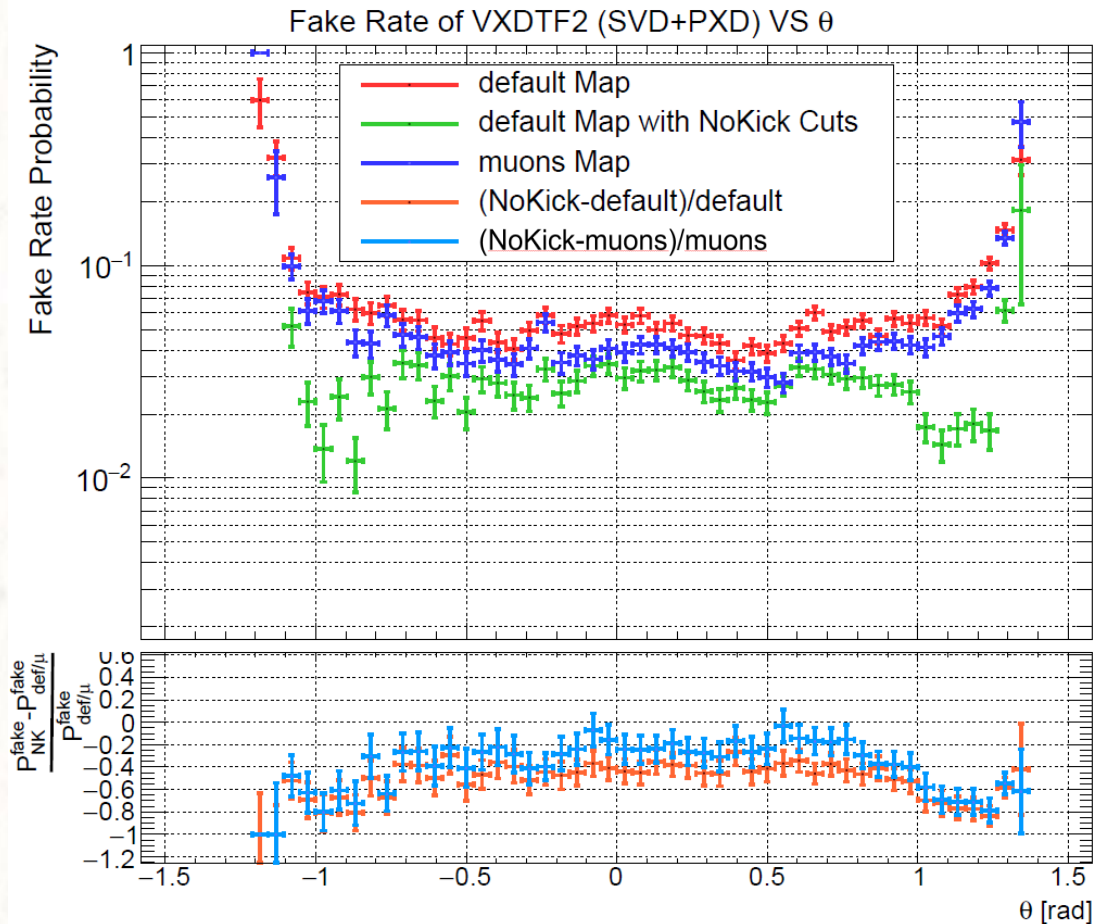
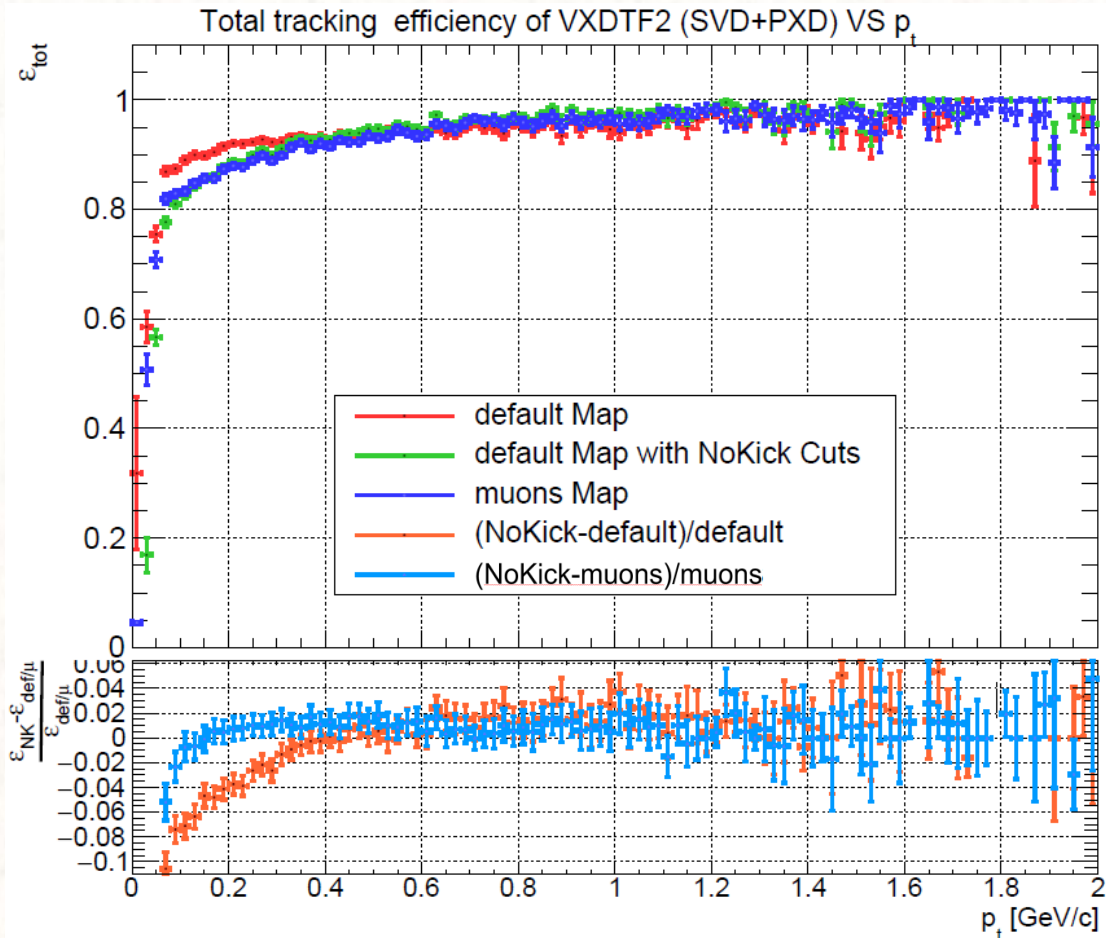
$$\varepsilon_{PR} = \frac{N_{\text{reco good}}}{N_{\text{tot}}} \quad \varepsilon_{fit} = \frac{N_{\text{reco good}} + \text{fit converged}}{N_{\text{reco good}}} \quad \text{Fake Rate} = \frac{N_{\text{reco non good}}}{N_{\text{tot}}}$$

$$\varepsilon_{tot} = \varepsilon_{PR} \cdot \varepsilon_{fit}$$

- *good* = reco. con  $>2/3$  hit su singola MC-particle
- *tot* = entro l'accettanza

## Training Sample di muoni

- Aggiunto nell'analisi un TS prodotto con 2 milioni di eventi da 10 muoni con spettro in impulso piatto
- Rappresentano eventi facili da ricostruire, ma possono introdurre bias



# Risultati – Tracking (senza background)

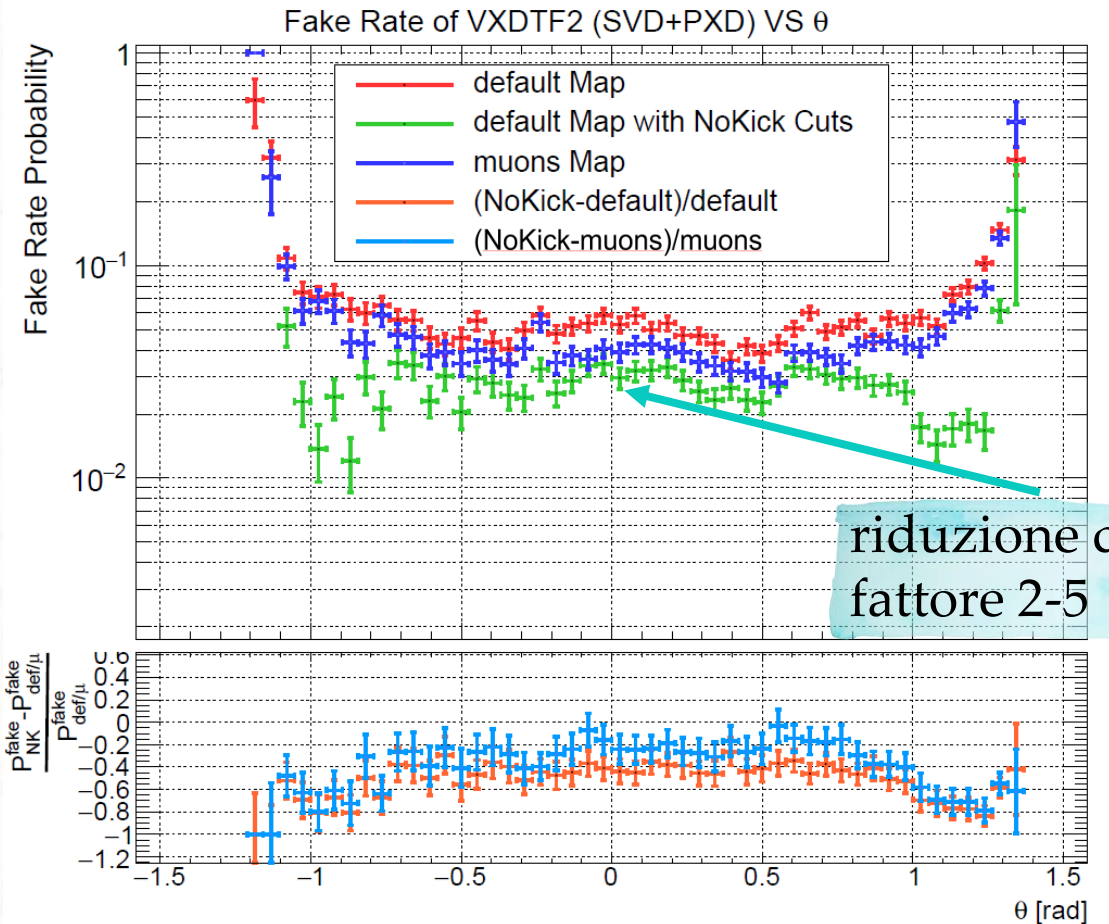
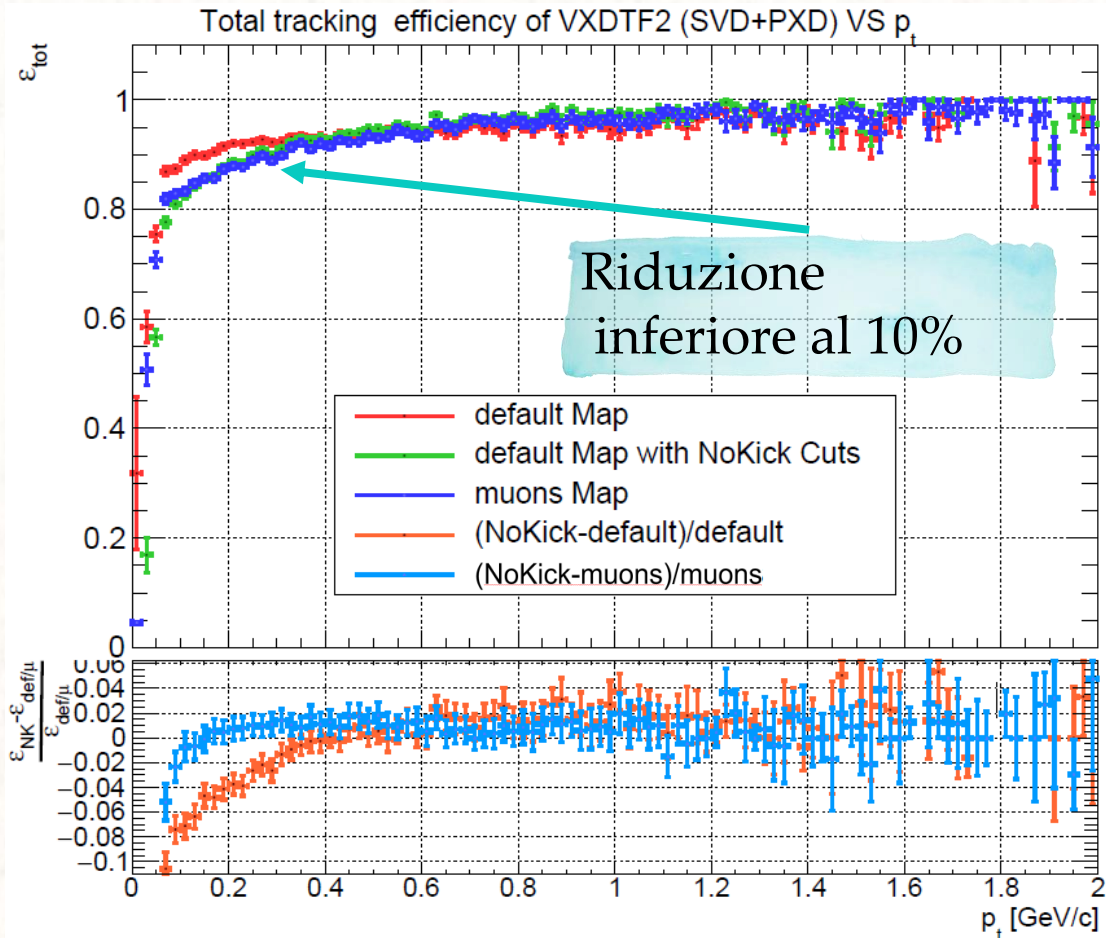
$$\varepsilon_{PR} = \frac{N_{\text{reco good}}}{N_{\text{tot}}} \quad \varepsilon_{fit} = \frac{N_{\text{reco good}} + \text{fit converged}}{N_{\text{reco good}}} \quad \text{Fake Rate} = \frac{N_{\text{reco non good}}}{N_{\text{tot}}}$$

$$\varepsilon_{tot} = \varepsilon_{PR} \cdot \varepsilon_{fit}$$

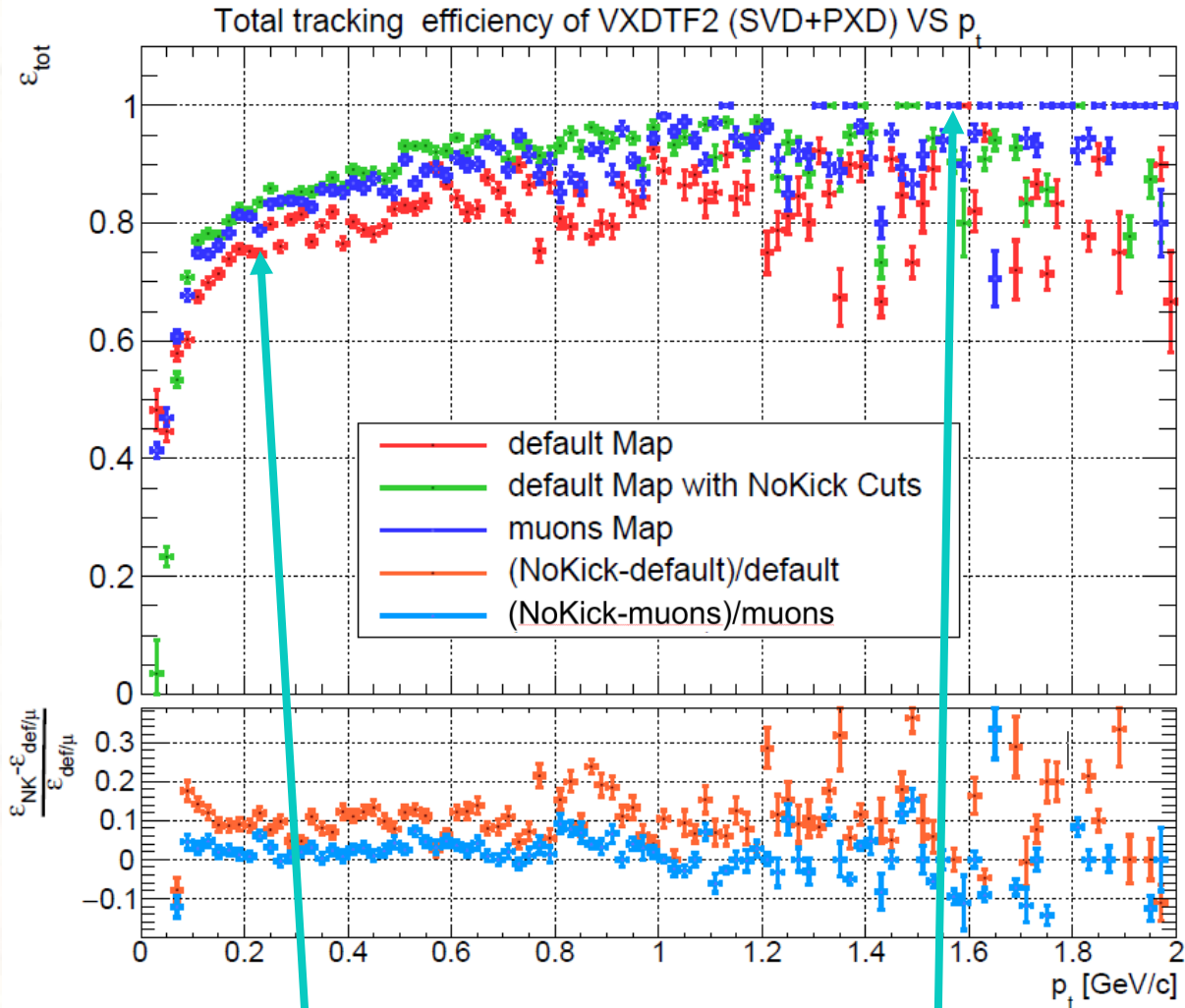
- *good* = reco. con  $>2/3$  hit su singola MC-particle
- *tot* = entro l'accettanza

## Training Sample di muoni

- Aggiunto nell'analisi un TS prodotto con 2 milioni di eventi da 10 muoni con spettro in impulso piatto
- Rappresentano eventi facili da ricostruire, ma possono introdurre bias

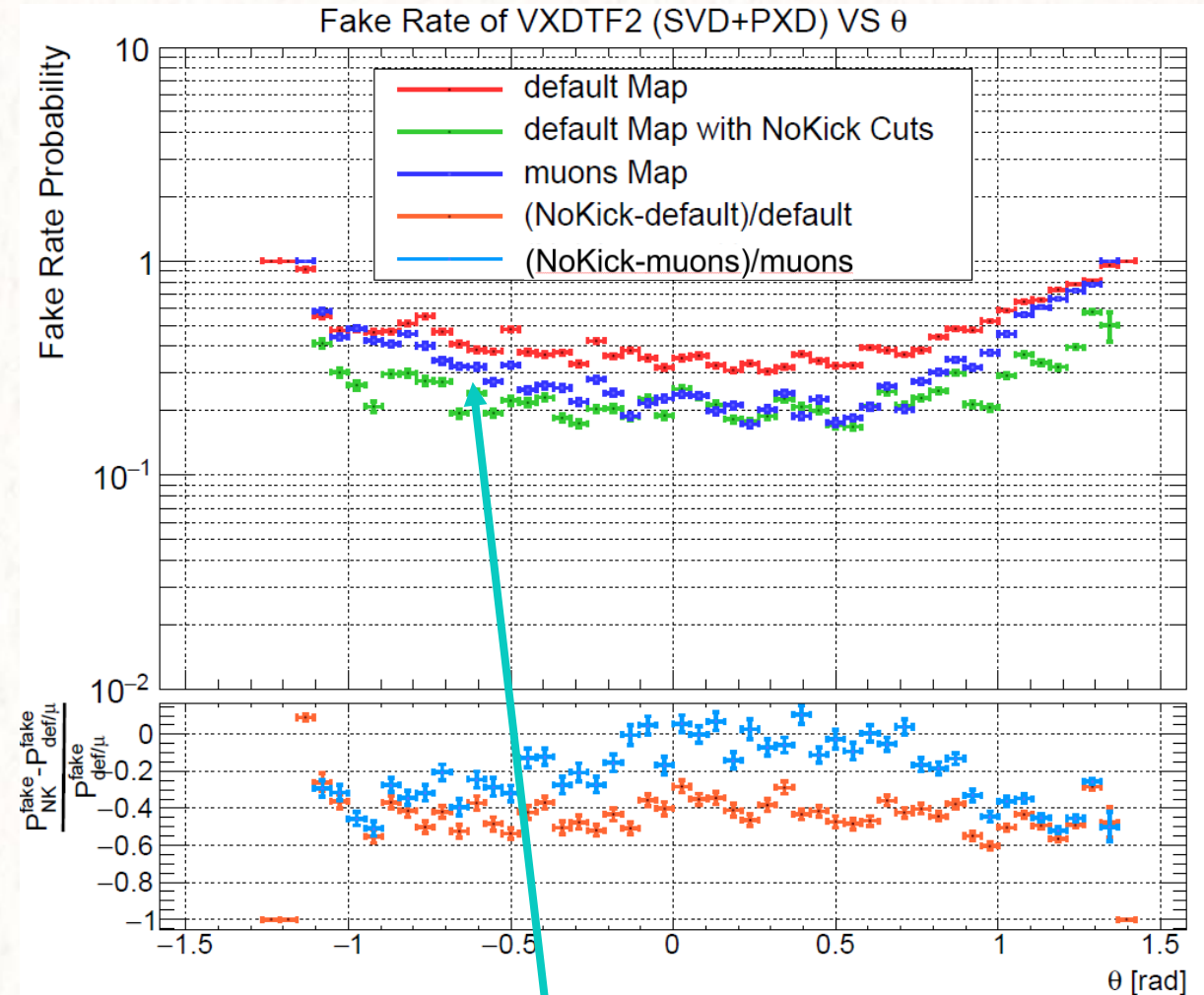


# Risultati – Tracking (con background atteso)



NoKick Map risente meno del bkg

Muon Map migliore ad alto  $p_t$



Riduzione relativa ridotta

# Risultati – Tracking (analisi dei tempi)

NB: grosse fluttuazioni (fino al 100%)

	Pattern Recognition [ms/ev]				Fitting [ms/ev]			
	senza Background		con Background		senza Background		con background	
	Fast	Full	Fast	Full	Fast	Full	Fast	Full
<b>Default Map</b>	10	32	48	125	270	425	280	394
<b>NoKick Map</b>	5	10	13	40	251	396	237	370
<b>Muon Map</b>	3	12	11	1610	267	361	272	378

Riduzione di un  
fattore 2-3

Muon Map anomala

Guadagno marginale

con background può  
essere rilevante

# Conclusioni

- Il VXDTF2 ha delle performance soddisfacenti in termini di:
  - Efficienza
  - Fake rate
  - Tempo di esecuzione
- La selezione del Training Sample migliora le performance del VXDTF2
  - il fake rate è dimezzato
  - Perdita di efficienza contenuta entro quanto previsto
  - Tempi di Pattern Recognition ridotti di un fattore 3
- In presenza di background le performance relative sono le migliori



# Sviluppi Futuri

- Ottimizzazione della selezione
- Automatizzazione dell'efficienza richiesta
- Integrazione della selezione nel Training Sample di muoni
- Upgrade alla release 1 (utilizzo dell'informazione temporale) e validazione
- Validazione delle performance sulla geometria di Fase II

Grazie  
dell'attenzione

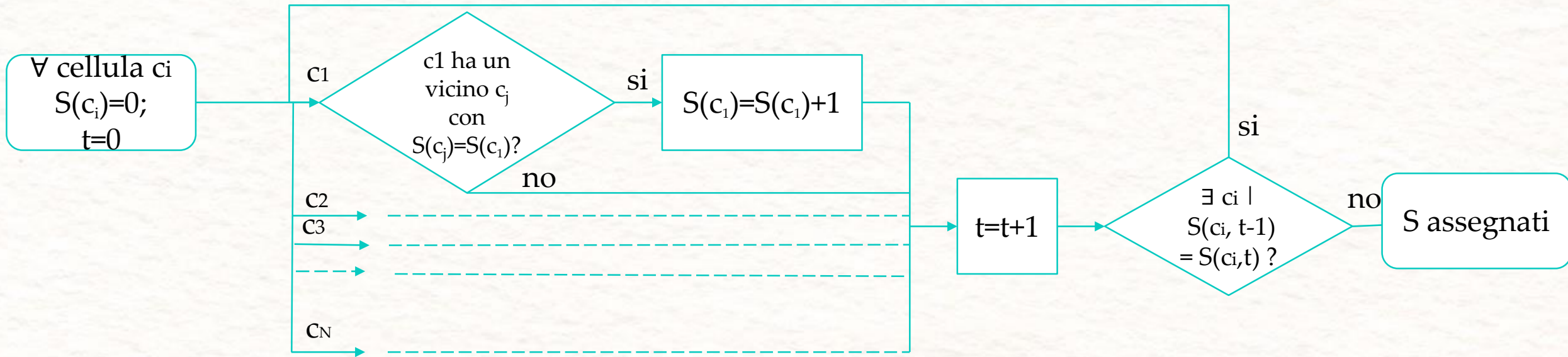


# SLIDES di BACKUP

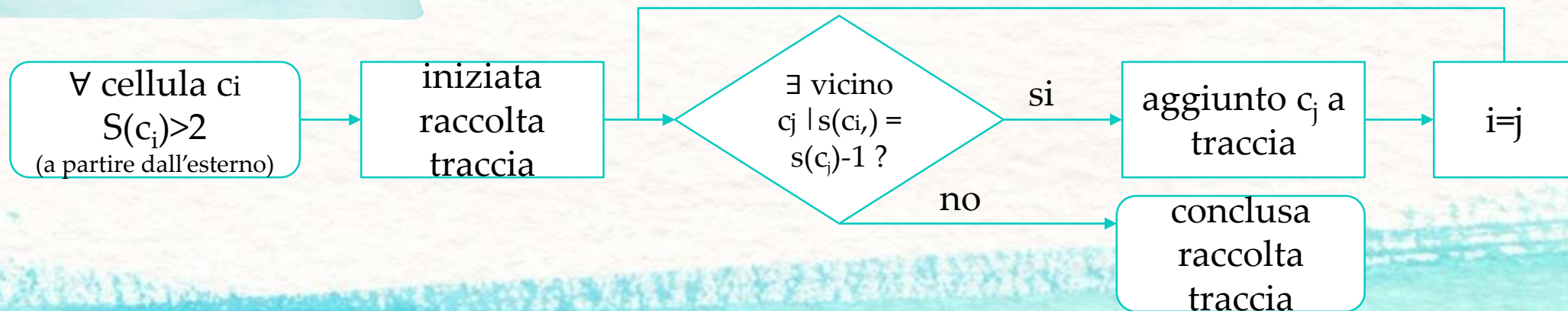
# 2 – Raccolta delle tracce: Cellular Automaton

- **Cellula:** segmento
- **Vicino:**  $x$  vicino di  $y$  se condividono una hit e  $x$  ha le hit su layer più interni

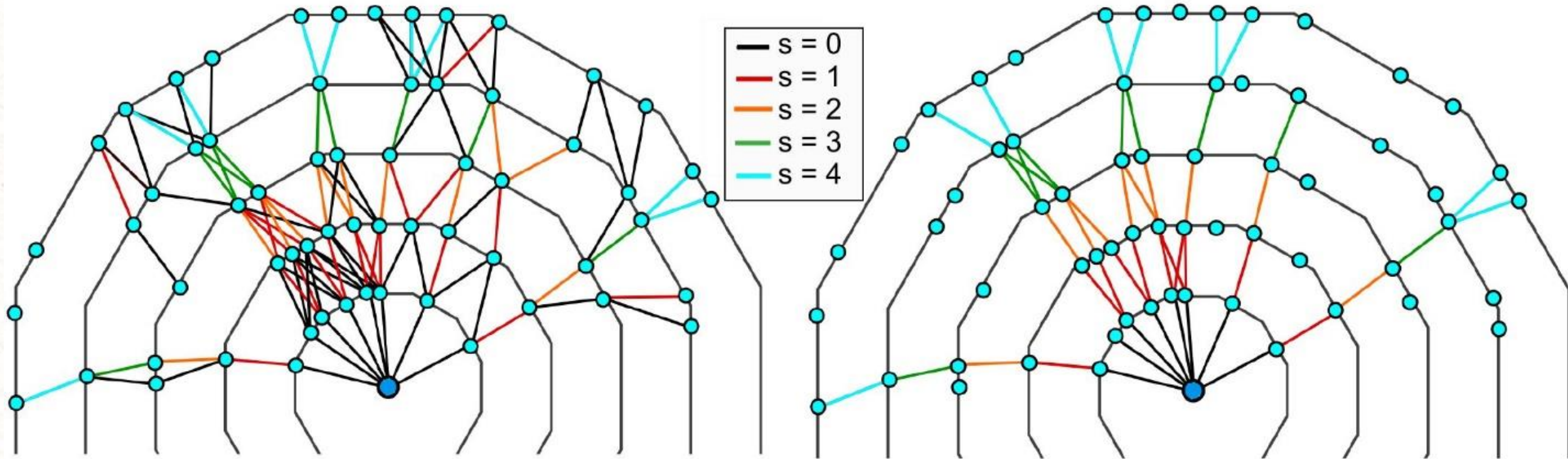
## Fase 1: assegnazione stati



## Fase 2: raccolta tracce



# Esempio applicazione Cellular Automaton



# 3 – Rimozione delle tracce sovrapposte

## Necessità

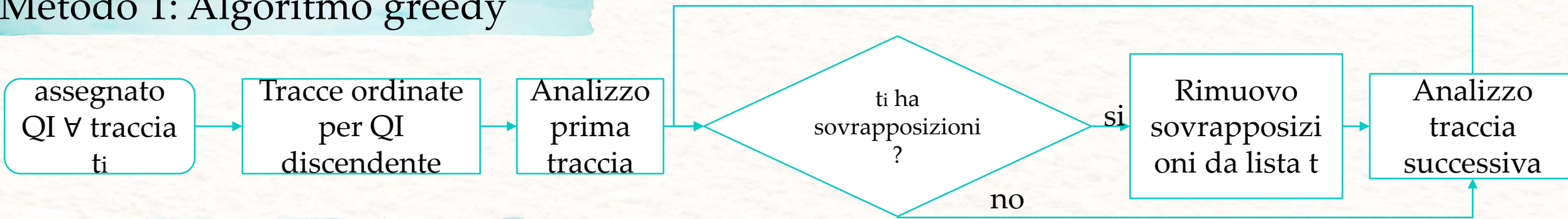
- Con Cellular Automaton due (o più) candidati traccia possono avere hit condivise
- In tal caso uno dei due è un falso → deve essere rimosso

## Indicatore di Qualità (QI)

- Assegnabile QI ad ogni candidato traccia (QI=0 → cattivo, QI=1 → buona)
  - P-value di fit parziali (cerchio sul piano trasverso, retta su quello longitudinale)
  - Kalman filter
  - Output Neural Network

# 3 – Rimozione delle tracce sovrapposte (2)

## Metodo 1: Algoritmo greedy



- PRO: veloce
- CONTRO: inefficiente

## Metodo 2: Rete Neurale di Hopfield

- Neuroni = candidati traccia
- Compatibili = non sovrapposti
- Dinamica degli stati  $S_i(S_j, w_{ij}, QI, T)$   $w_{ij}$  = pesi relativi (compatibilità) T=temperatura di annealing
- Energia da minimizzare  $E(s_i, s_j, w_{ij})$
- Raggiunto il minimo tenuti neuroni con stato sopra una certa soglia di compatibilità
- PRO: efficiente
- CONTRO: su eventi con alta occupancy è molto lento

## 4 – Passaggi multipli

- Procedura del VXDTF iterabile con Sector Map allenate su diversi range d'impulso
- Ogni volta riservate hit delle tracce individuate
- Partendo da range ad alto impulso si semplifica combinatorio per tracce più difficili (basso impulso)



# Redesign del software: VXDTF2

## Necessità

- VXDTF scritto e gestito da singola persona
- Singolo modulo (44k linee di codice) → manutenzione impossibile
- Autore ha lasciato la collaborazione

## Redesign: VXDTF2

- Struttura Modulare
- Diversi miglioramenti tecnici:
  - Gestione filtri
  - Training Sample
  - Nuovi QI
  - Gestione sovrapposizioni
  - Eventi ad alta occupancy
- *Belle II - Pisa tracking group*: ruolo rilevante nella riscrittura del codice

# Parametri di Traccia

$$\omega \rightarrow \frac{B_3 q}{\sqrt{P_1^2 + P_2^2}} \quad (1)$$

$$\tan \lambda \rightarrow \frac{P_3}{\sqrt{P_1^2 + P_2^2}} \quad (2)$$

$$d_0 \rightarrow \operatorname{sgn}(B_3 q) \left( \sqrt{\left( \frac{P_2}{B_3 q} + X_1 \right)^2 + \left( X_2 - \frac{P_1}{B_3 q} \right)^2} - \sqrt{\frac{P_1^2 + P_2^2}{B_3^2 q^2}} \right) \quad (3)$$

$$\chi \rightarrow \tan^{-1} \left( \operatorname{sgn}(B_3 q) \left( \frac{P_1^2 + P_2^2}{B_3 q} + P_2 X_1 - P_1 X_2 \right), (-P_1 X_1 - P_2 X_2) \operatorname{sgn}(B_3 q) \right)$$

$$\varphi_0 \rightarrow \tan^{-1}(P_1, P_2) - \chi \quad (4)$$

$$z_0 \rightarrow \frac{P_3 \chi}{B_3 q} + X_3 \quad (5)$$

$$s \rightarrow -\frac{\sqrt{P_1^2 + P_2^2} \chi}{B_3 q}$$

[Eugenio, Oliver, Tobi,  
*helices: the nitty-gritty of their Parametrization*,  
B2GM 2015 ]

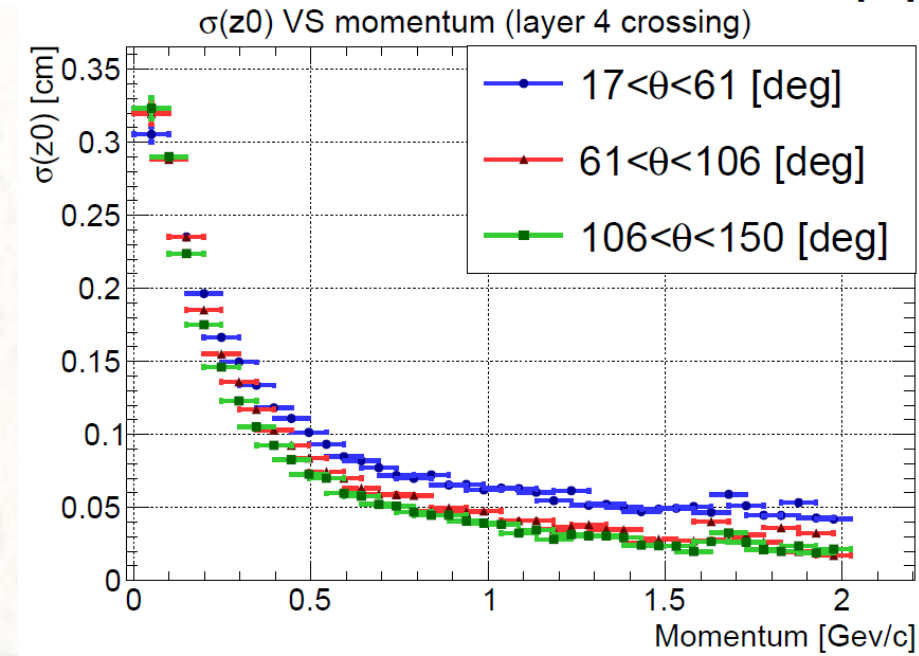
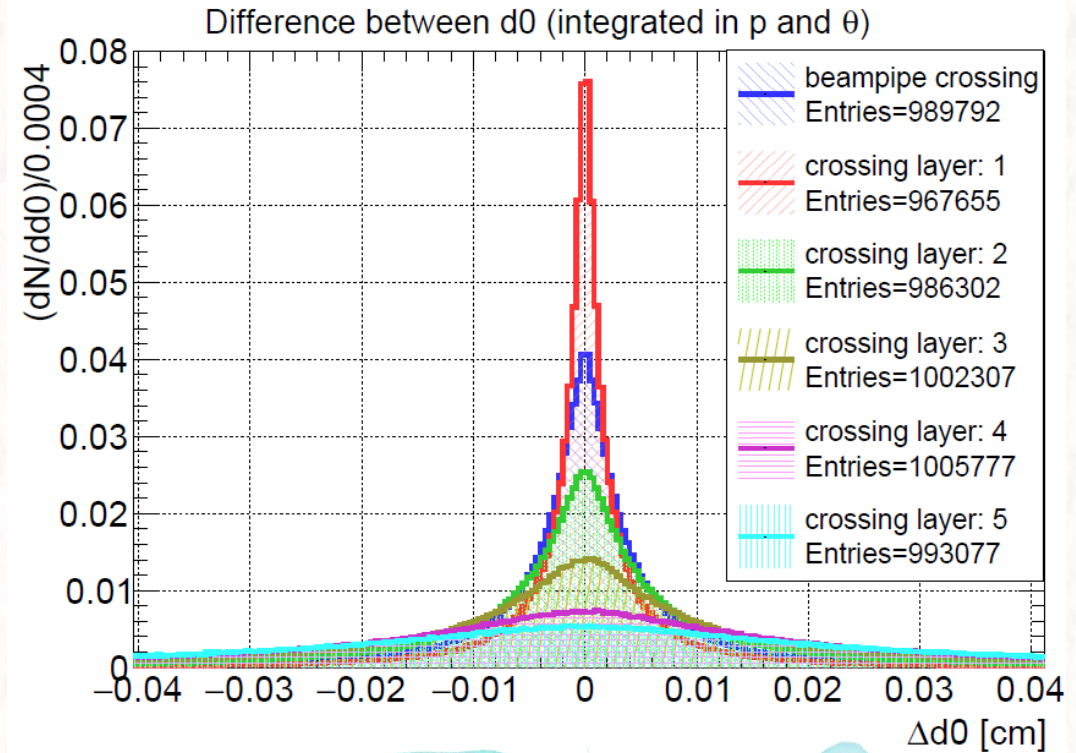
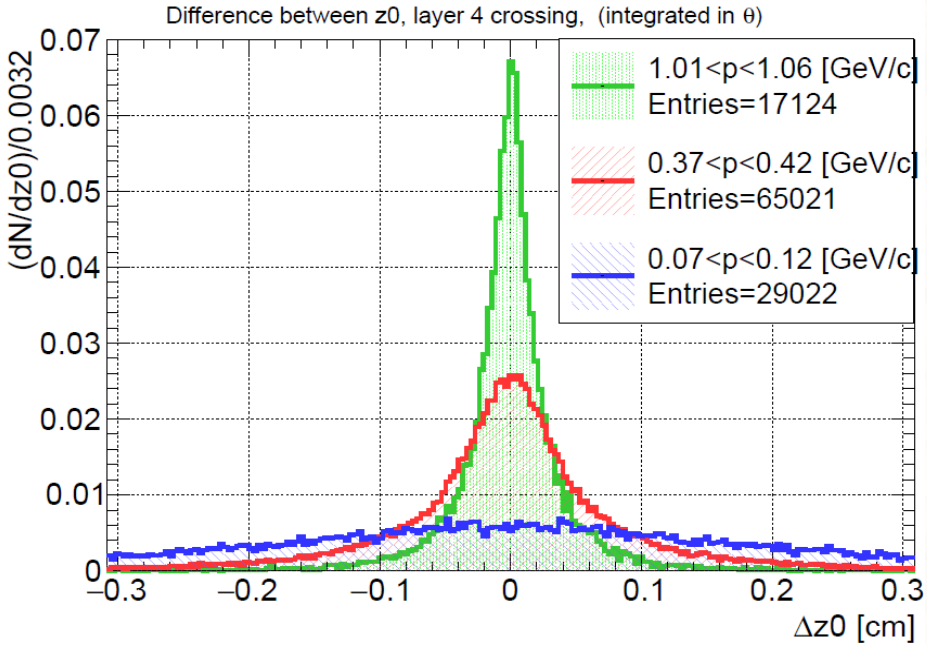
# Cut – function

$p$ [GeV/c]	$\varepsilon$	$N_\sigma$
0.2	0.999	3.3
0.1	0.995	2.8
0.07	0.98	2.3
0.05	0.9	1.6
0.04	0.84	1.4
0.03	0.75	1.1
0.02	0.7	1.0

$$\varepsilon_{\text{req}}(p) = -\frac{7.5 \cdot 10^{-7}}{p^{3.88}} + 1 \quad 0.041 < p < 1 \text{ GeV/c} \quad (1)$$

$$\varepsilon_{\text{req}}(p) = 6.3p + 0.57 \quad 0.02 < p < 0.041 \text{ GeV/c} \quad (2)$$

# Fase A, Passo 2 – Proprietà delle distribuzioni $\Delta X$



- $\Delta X(|\mathbf{p}|)$ : si allargano a basso momento
- $\Delta X(\theta)$ : circa costanti
- $\Delta X(l_1, l_2)$ : proporzionali al materiale a meno di effetti geometrici dei parametri



coerenza con allargamento da scattering multiplo

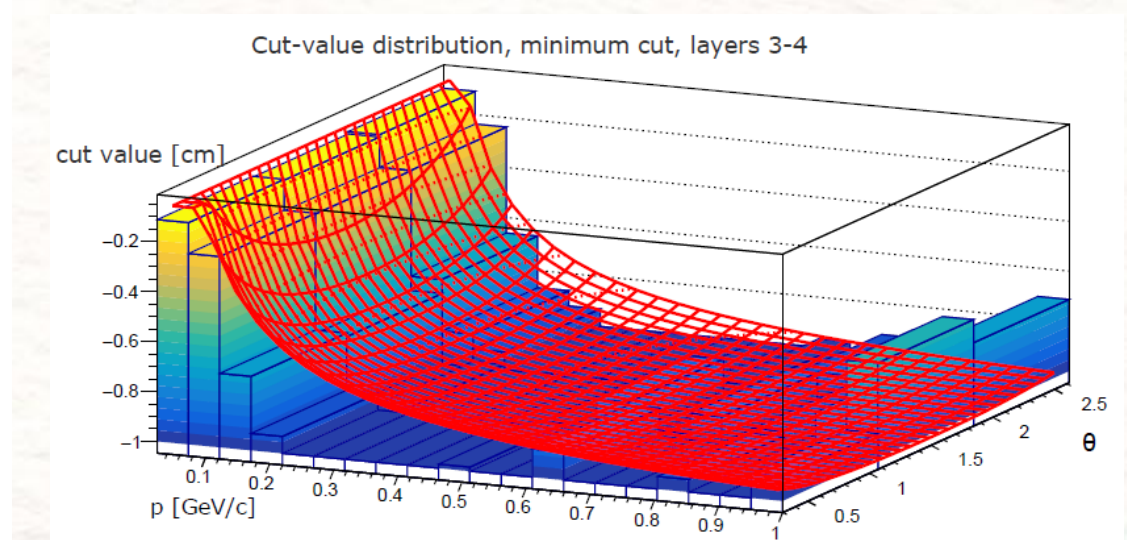
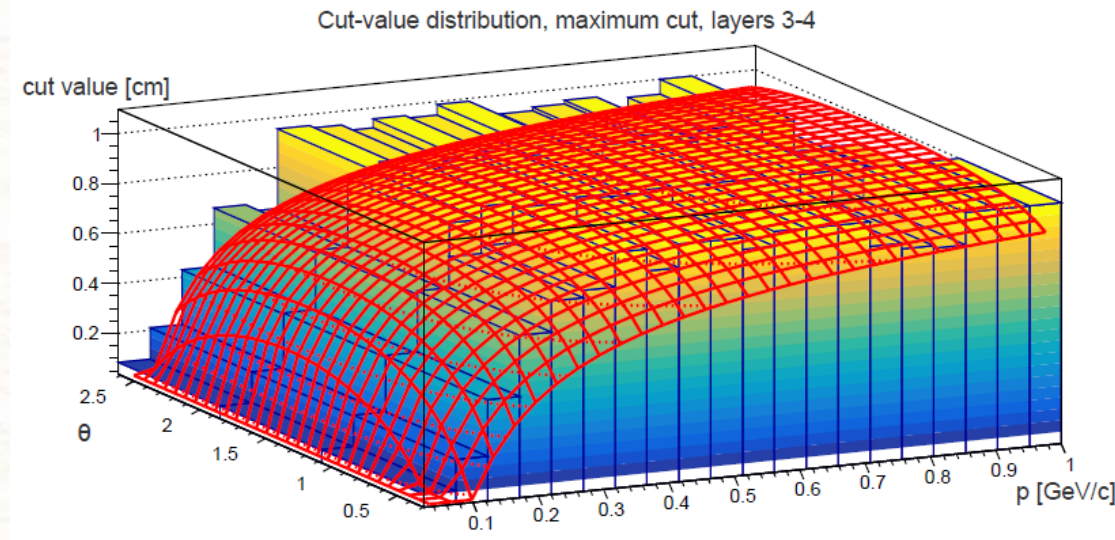
# Fase A – fit e memorizzazione dei tagli

- Effetti di binning in momento e angolo rilevanti
- Creati istogrammi 2D con  $C_m, C_M$  e fittati con:

$$\Delta X(p, \theta, l_1, l_2) \sim \frac{A(X, l_1, l_2)}{(p)^{B(X, l_1, l_2)} \sqrt{\sin \theta}} + C(X, l_1, l_2)$$

dove  $l_1, l_2$  sono i layer

- Memorizzati permanentemente i parametri liberi A,B,C per ogni coppia di layer e parametro X
- Limiti degli istogrammi  $\Delta X$  causano effetti (controllati) di saturazione
  - Limiti scelti:  $d_0, z_0 : 1 \text{ cm}, \phi_0, \tan \lambda : 0.3, \omega : 0.4 \text{ cm}^{-1}$



# Risultati della selezione del TS – Sector Map

## Complessità

- Numero di relazioni d'amicizia:

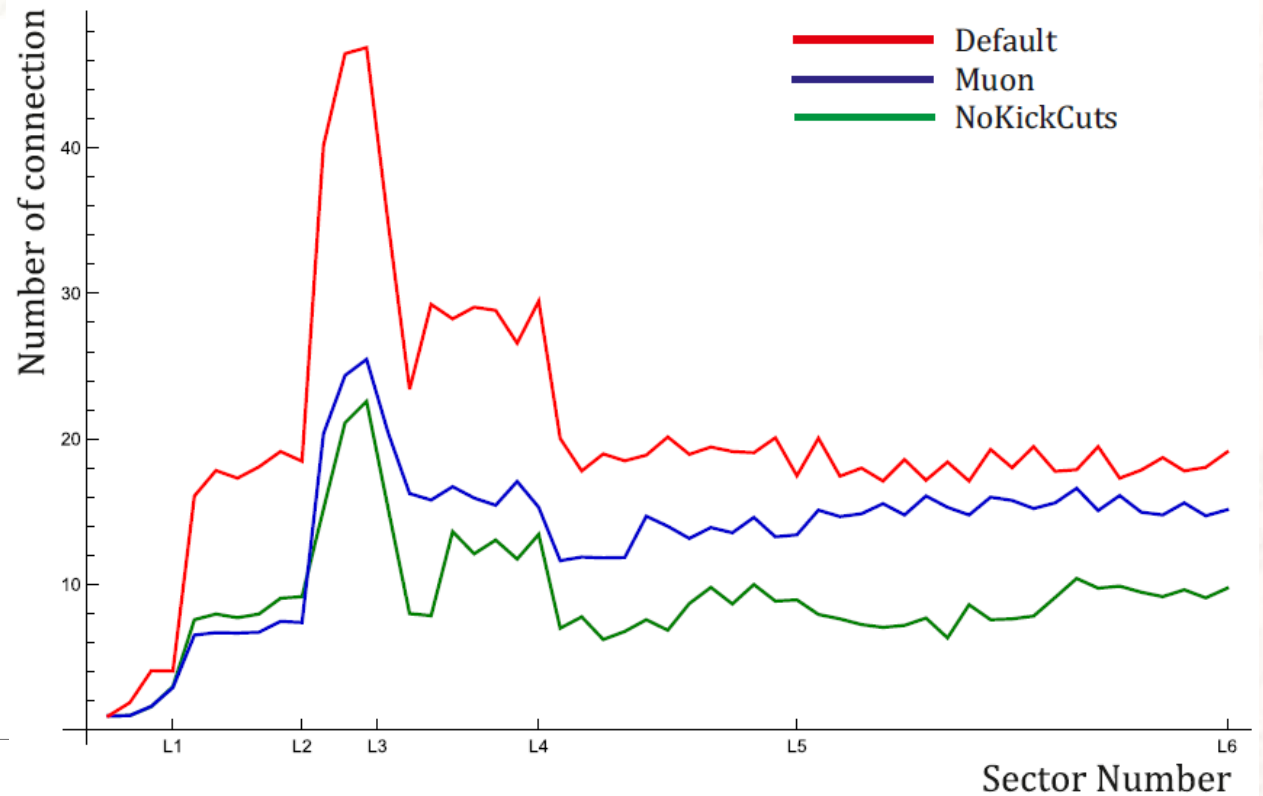
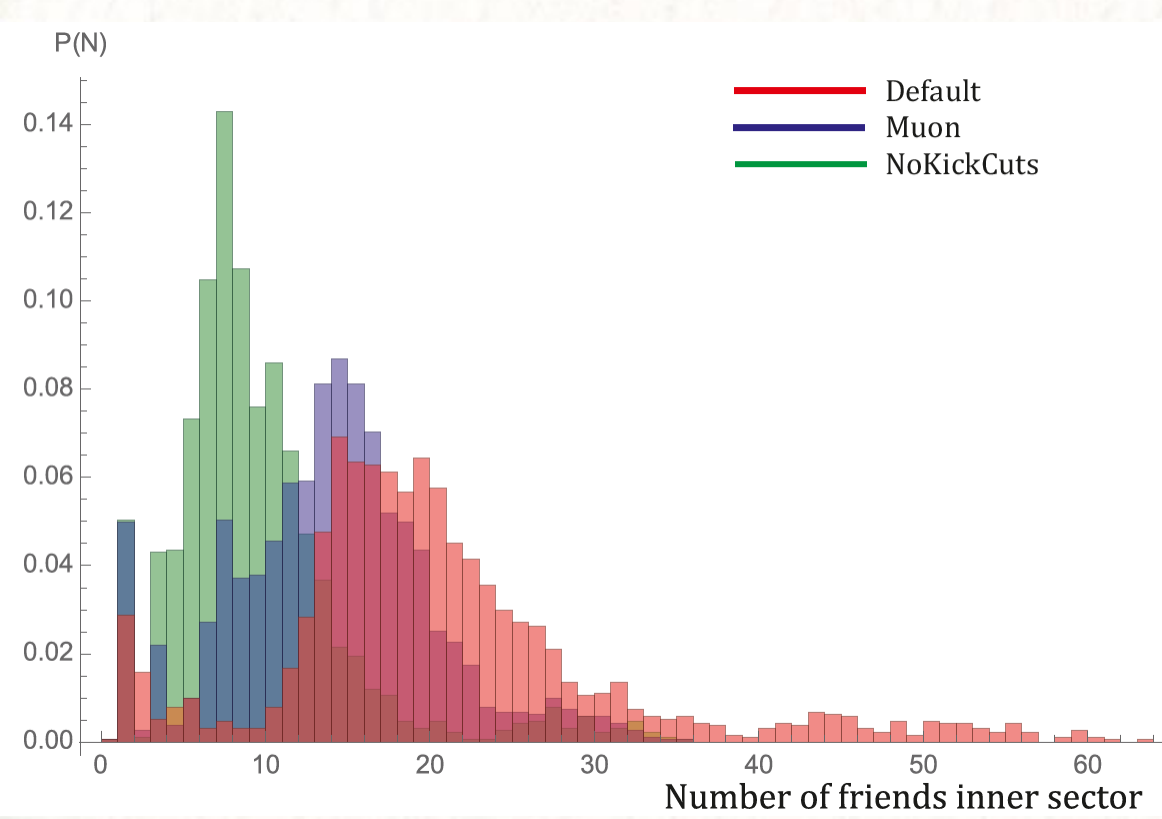
$$N_F^{\text{default}} = 38689,$$

$$N_F^{\text{NoKick}} = 17545,$$

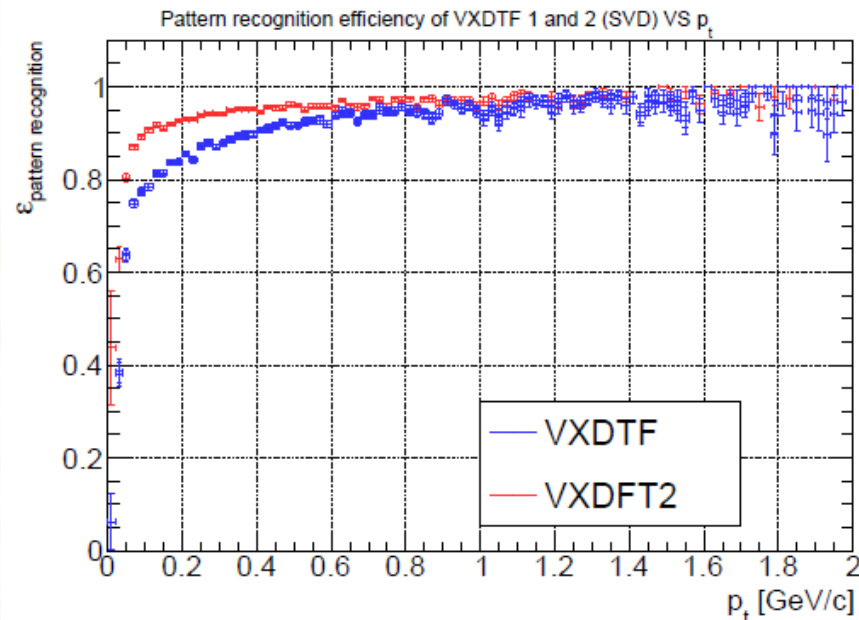
$$N_F^{\mu} = 25785.$$

## Dimensioni Sector Map

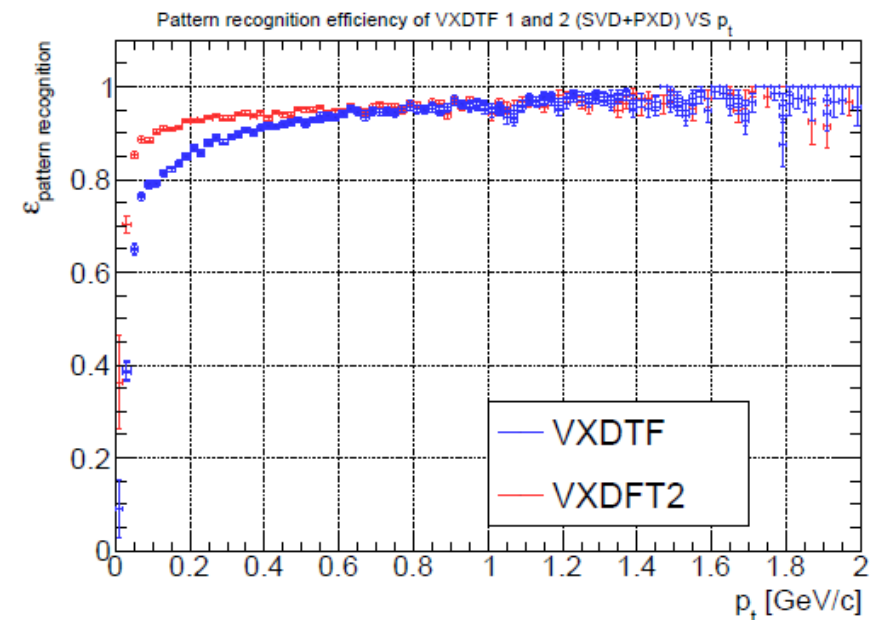
- Default (Fast/Full): 12.3 / 21.0 MB
- NoKick (Fast/Full): 5.1 / 8.3 MB
- Muon (Fast/Full): 6.8 / 14.0 MB



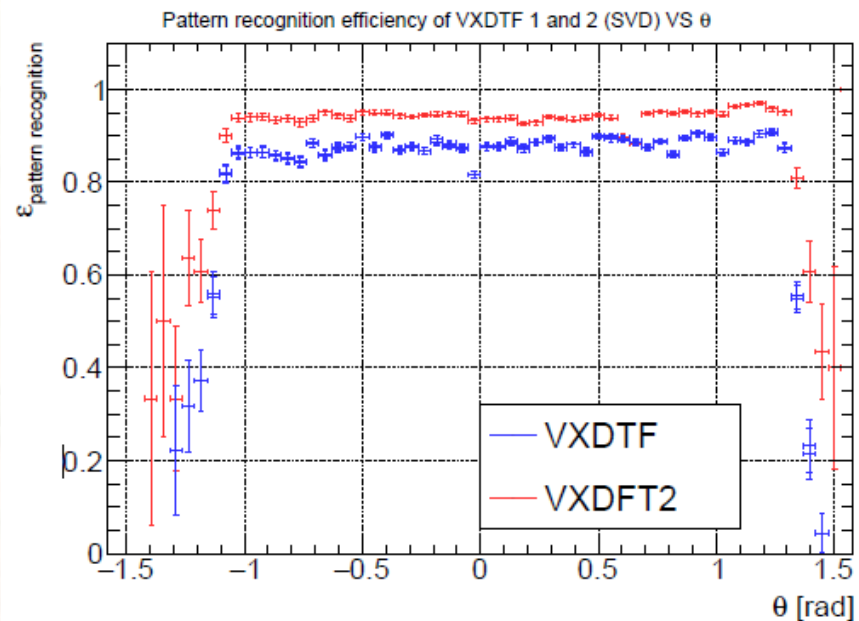
# Efficienza di Pattern Recognition – VXDTF VS VXDTF2



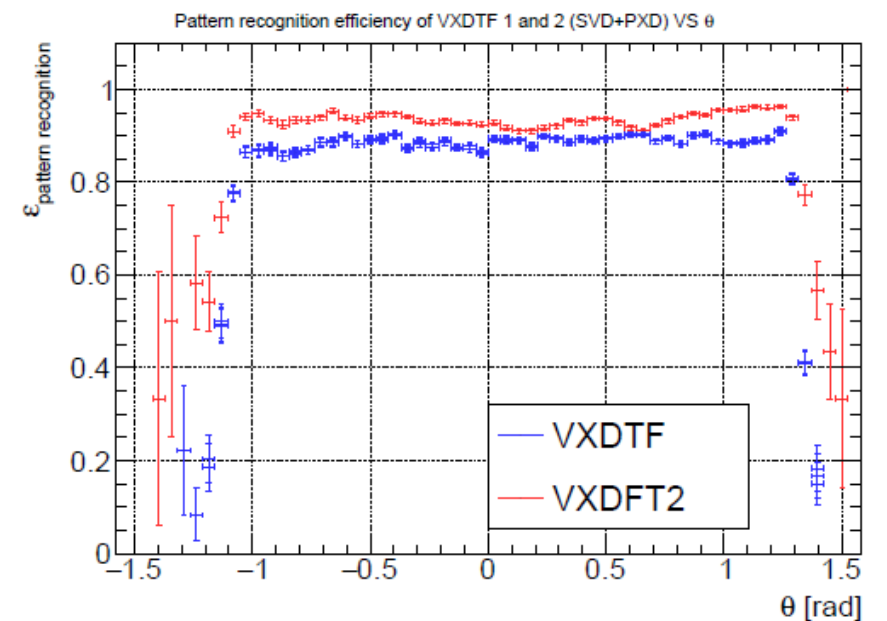
(a)



(b)

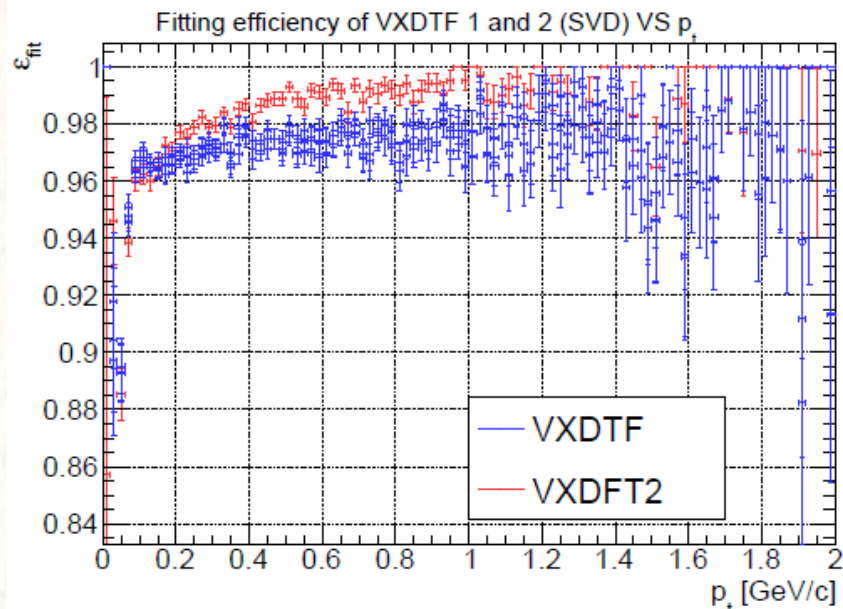


(c)

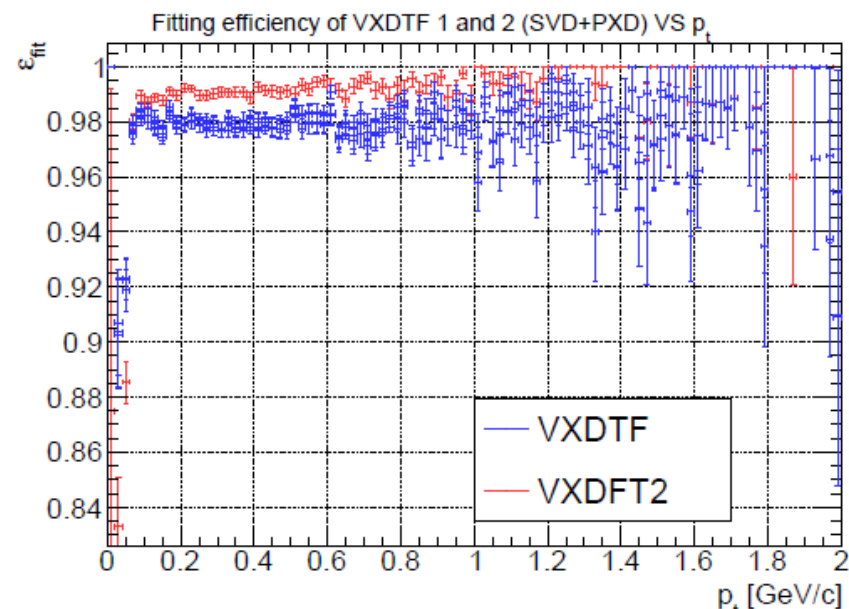


(d)

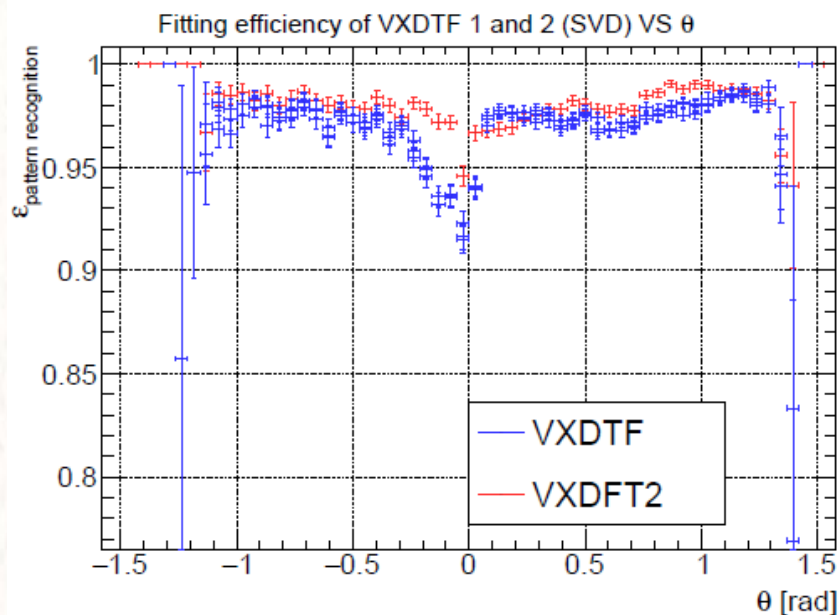
# Efficienza di fit – VXDTF VS VXDFT2



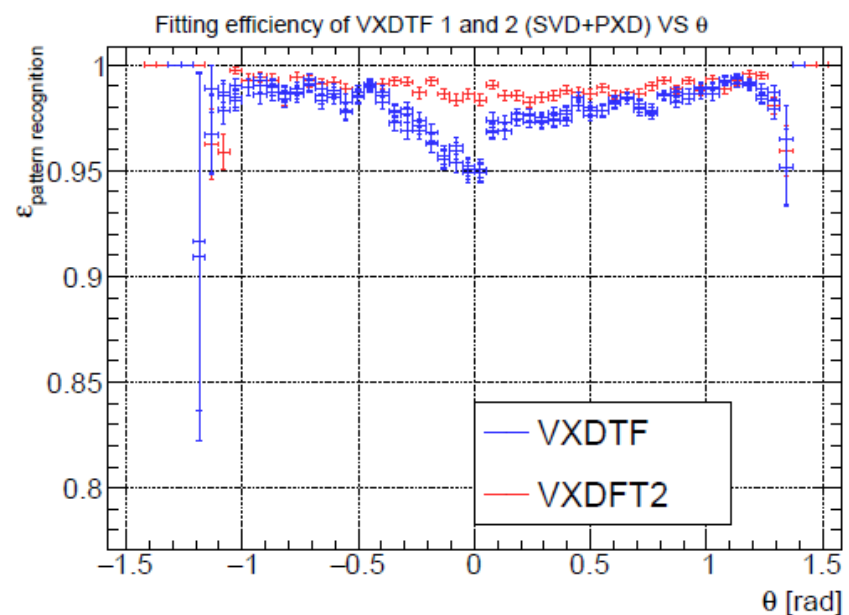
(a)



(b)



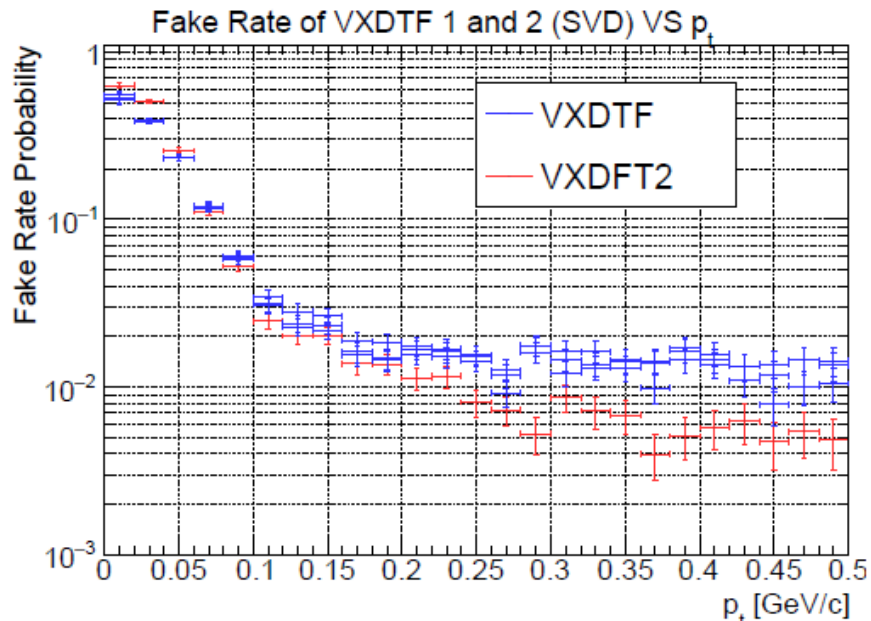
(c)



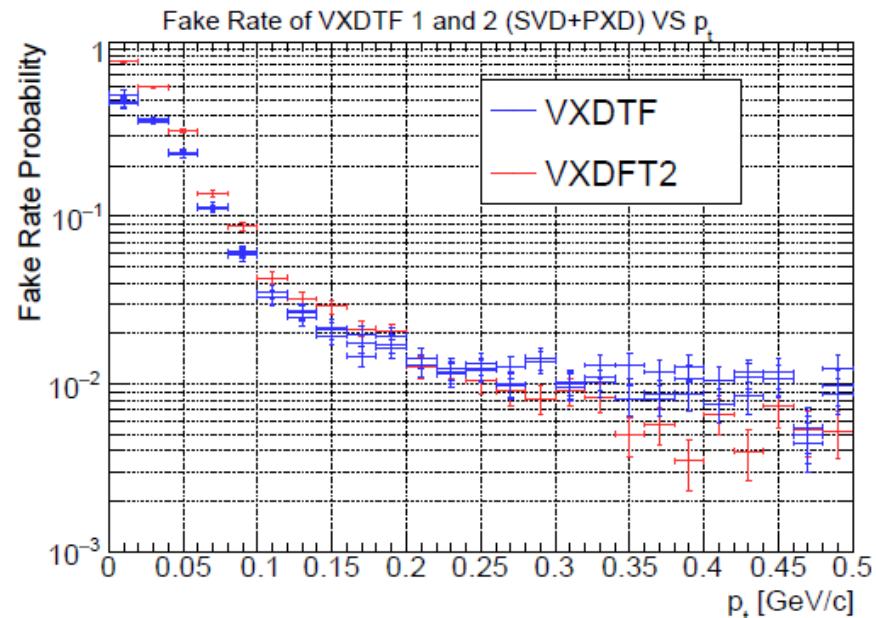
(d)



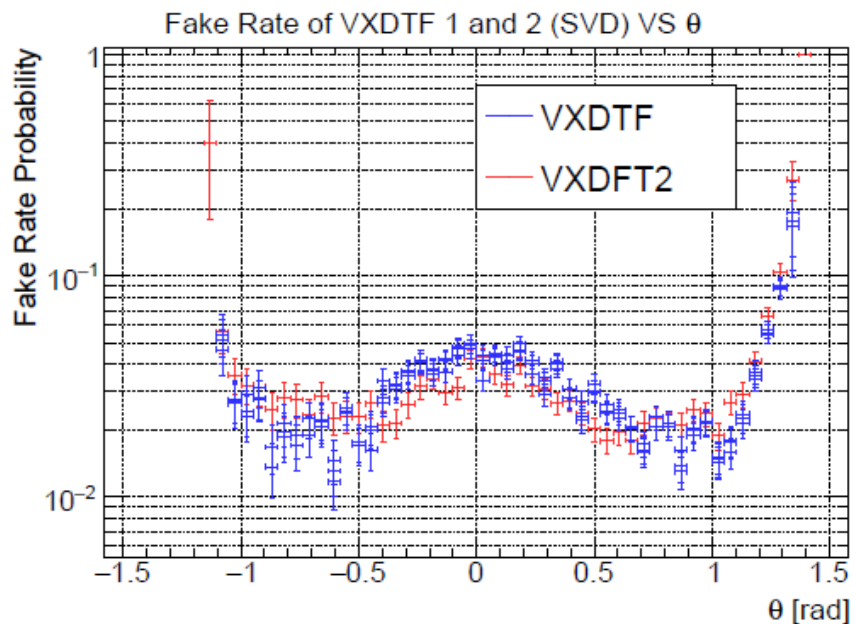
# Fake Rate— VXDTF VS VXDTF2



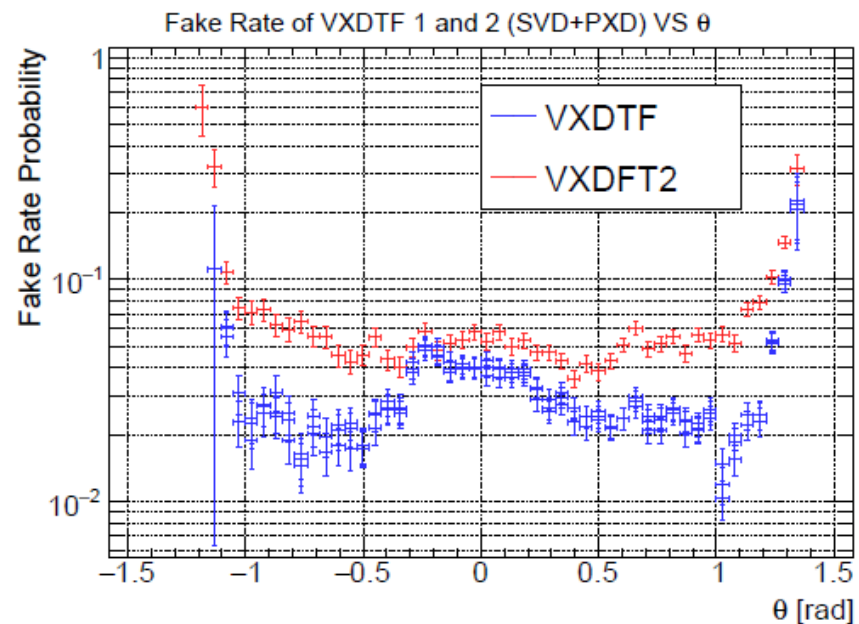
(a)



(b)

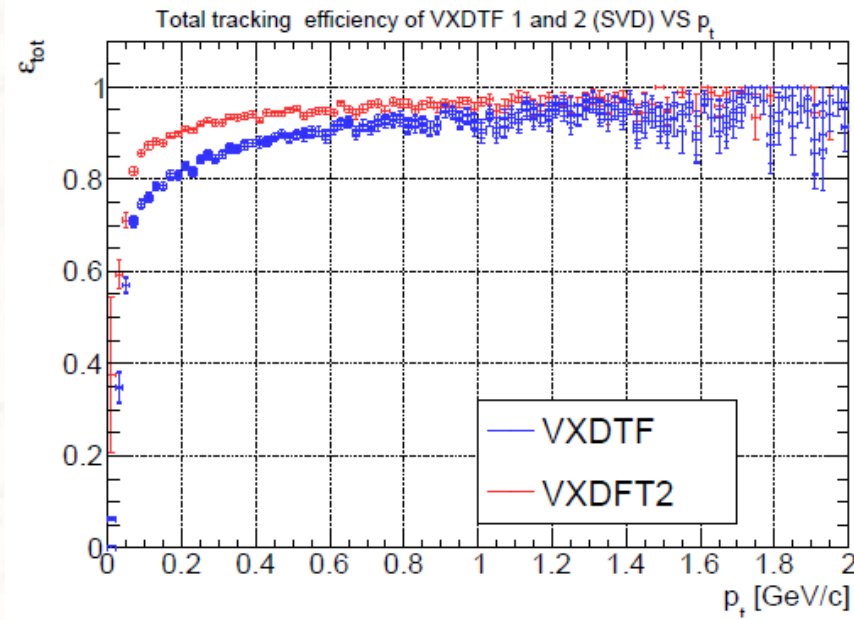


(c)

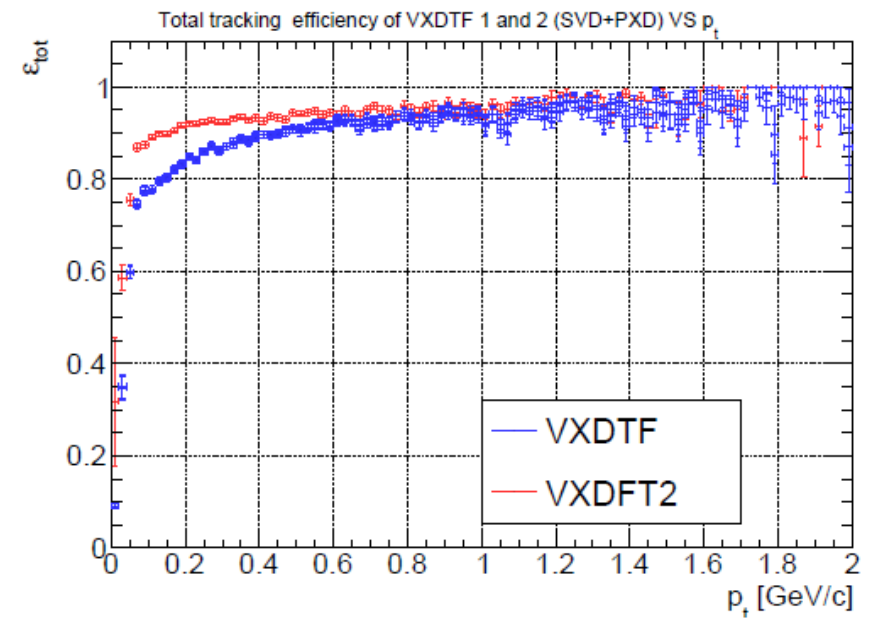


(d)

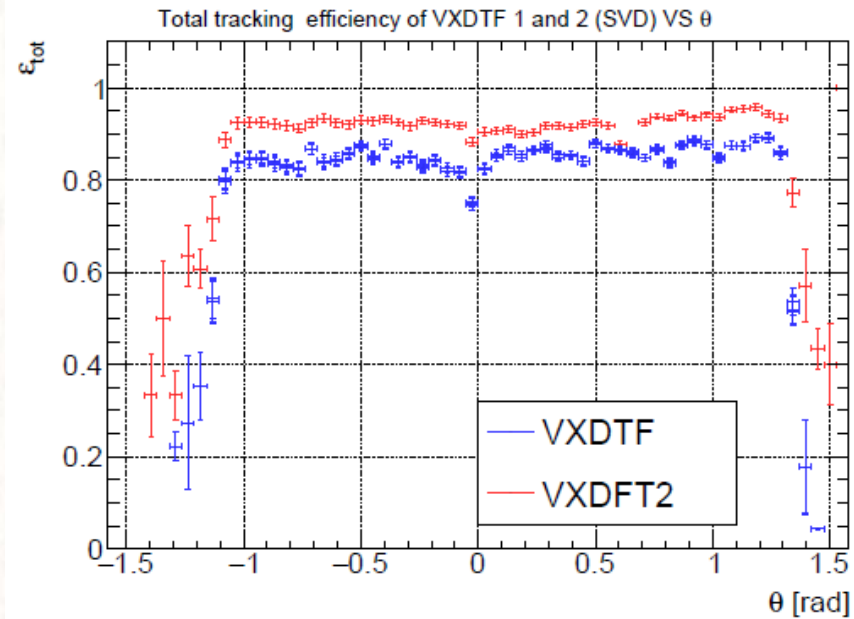
# Efficienza di tracking – VXDTF VS VXDTF2



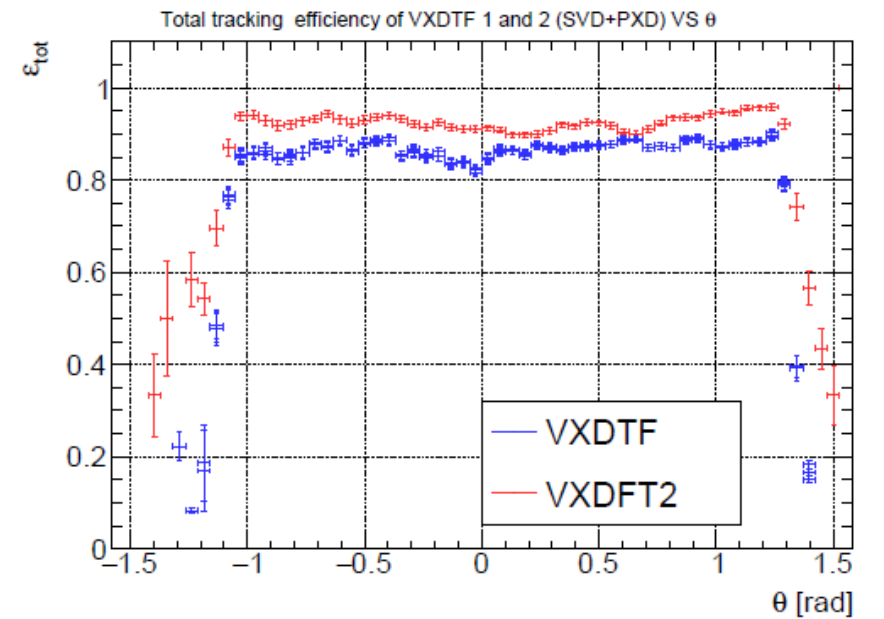
(a)



(b)

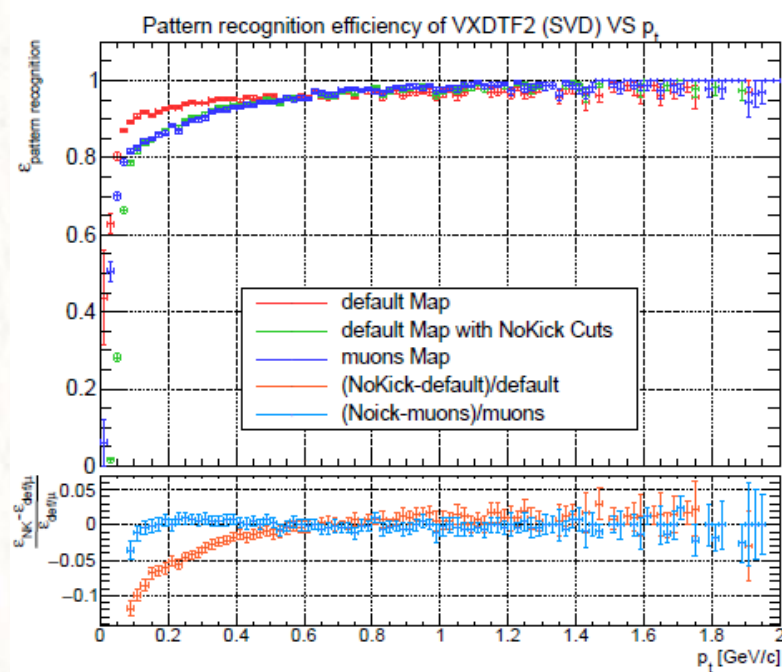


(c)

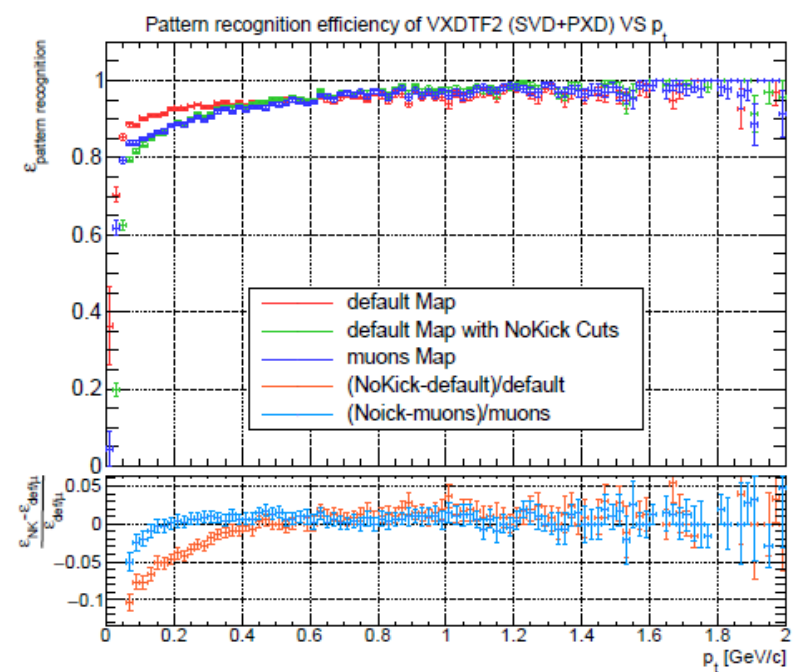


(d)

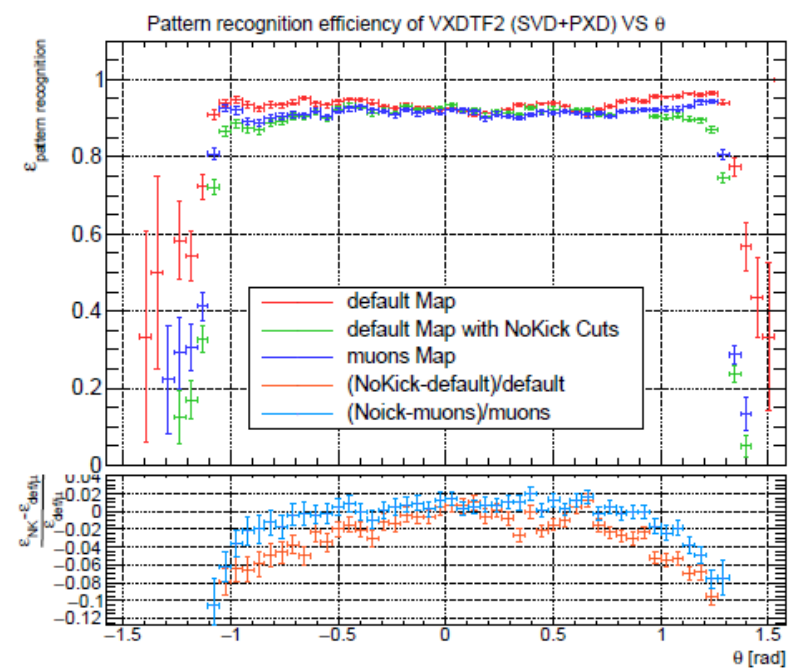
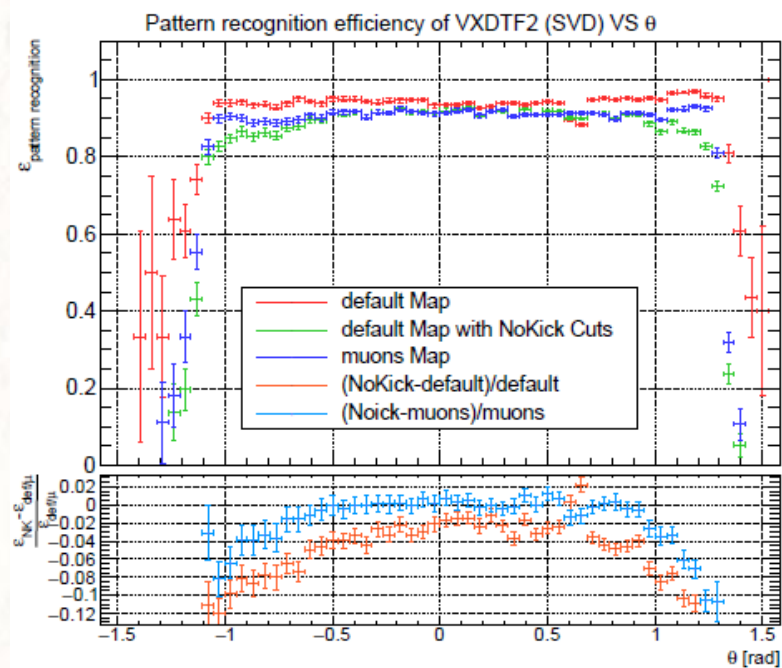
# Efficienza di Pattern Recognition – Confronto Sector Map



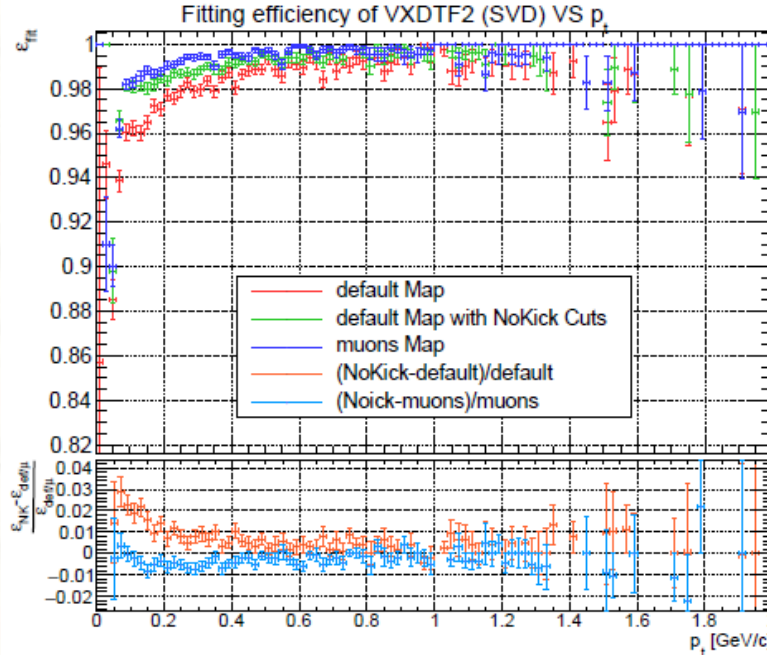
(a)



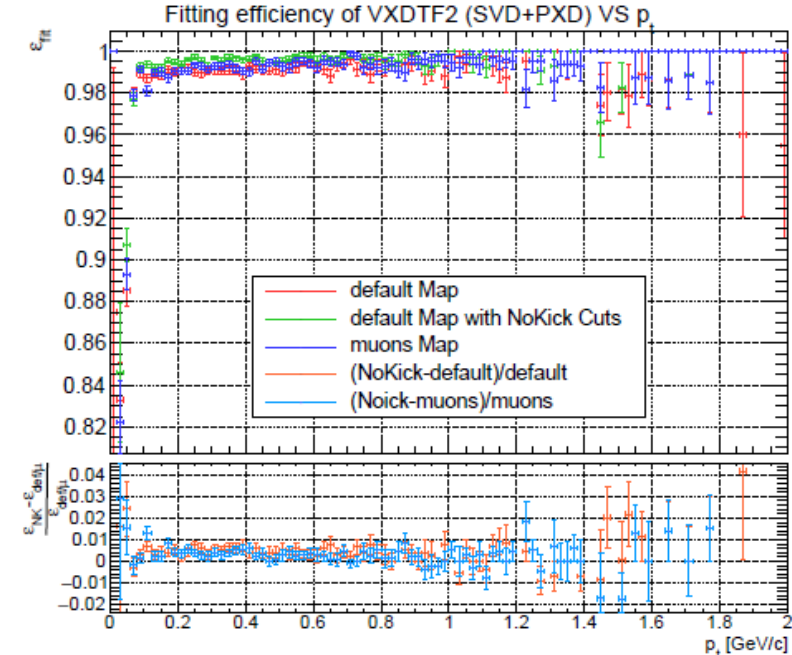
(b)



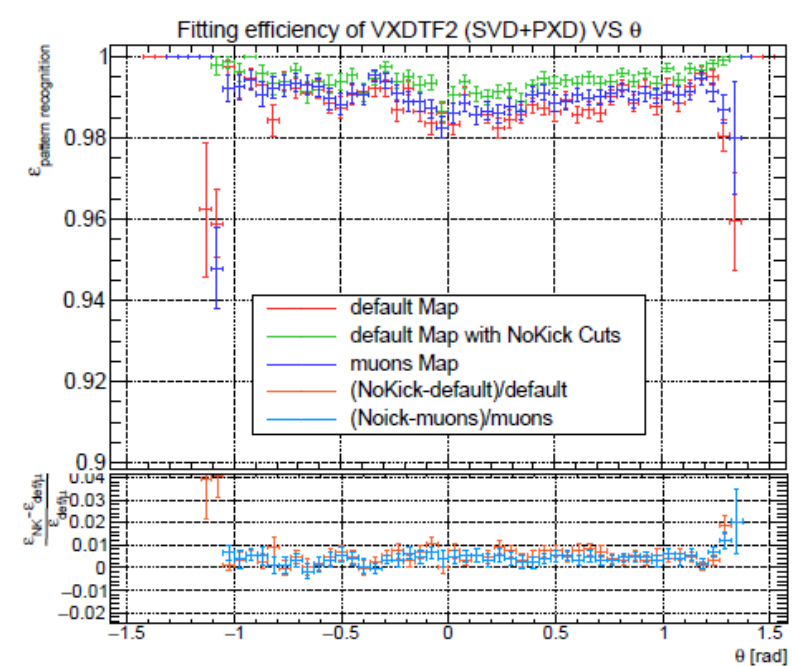
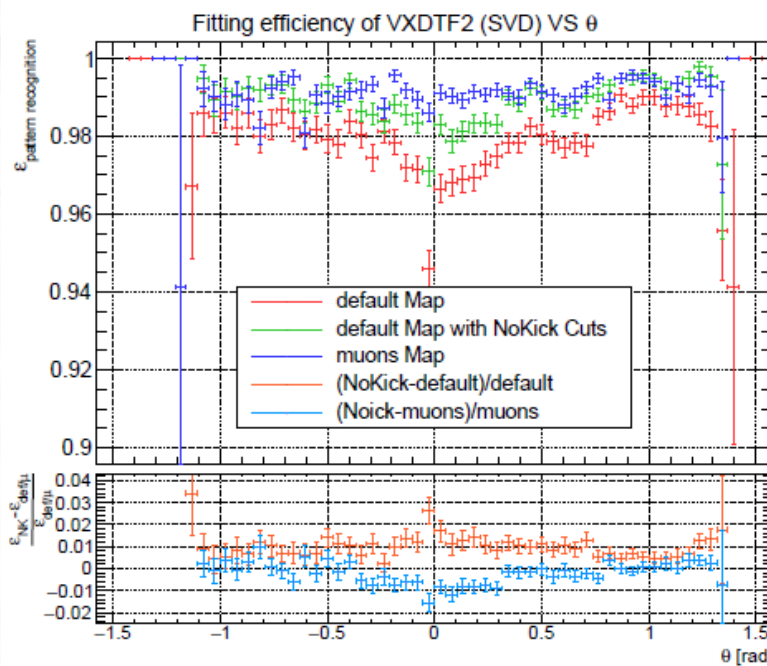
# Efficienza di fit – Confronto Sector Map



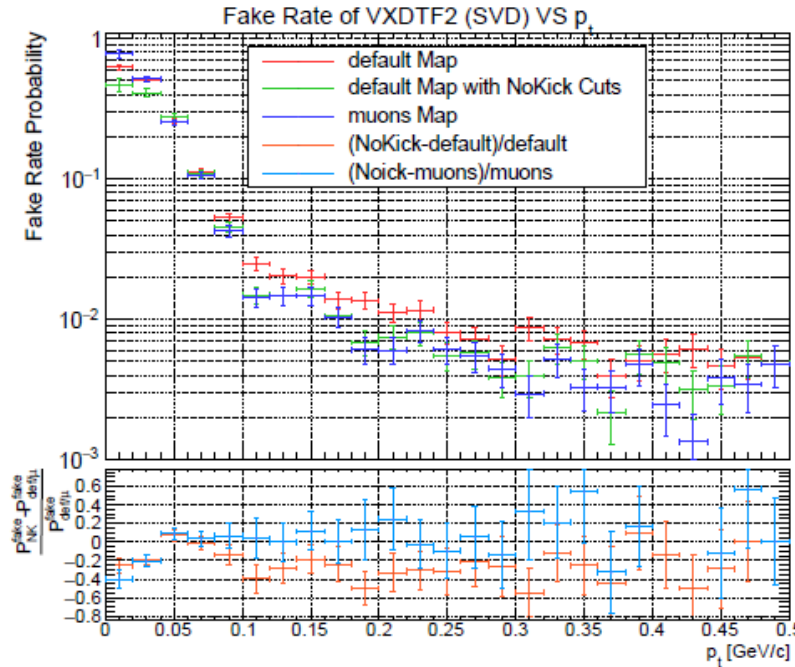
(a)



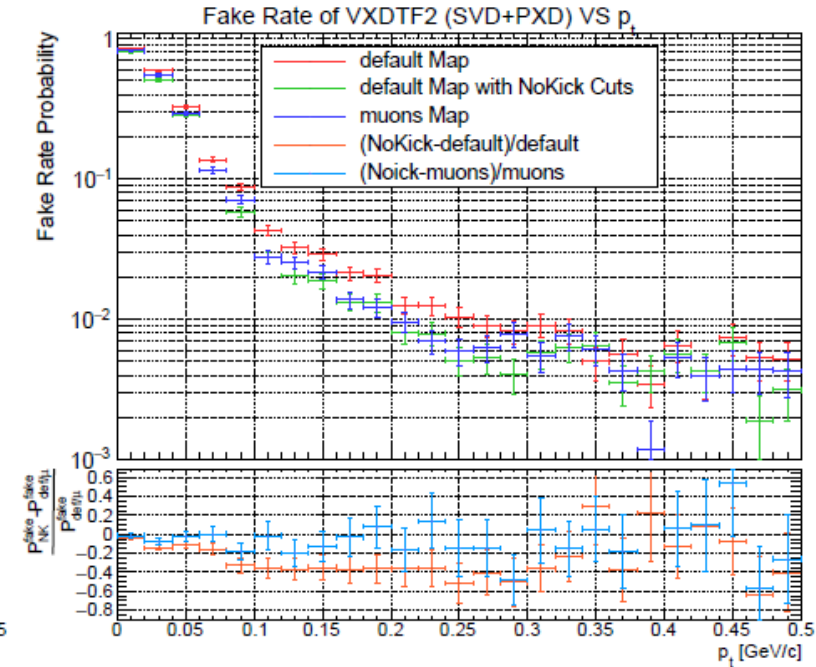
(b)



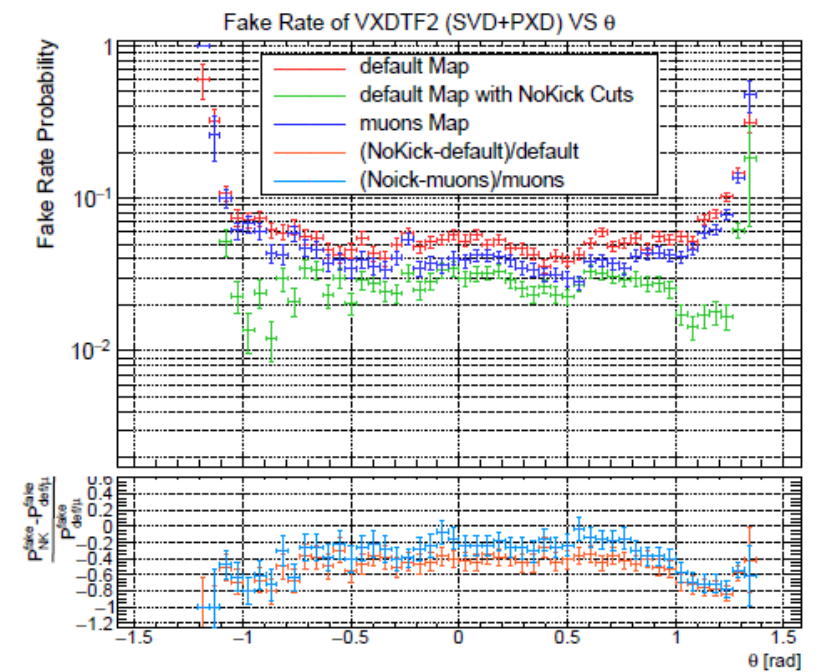
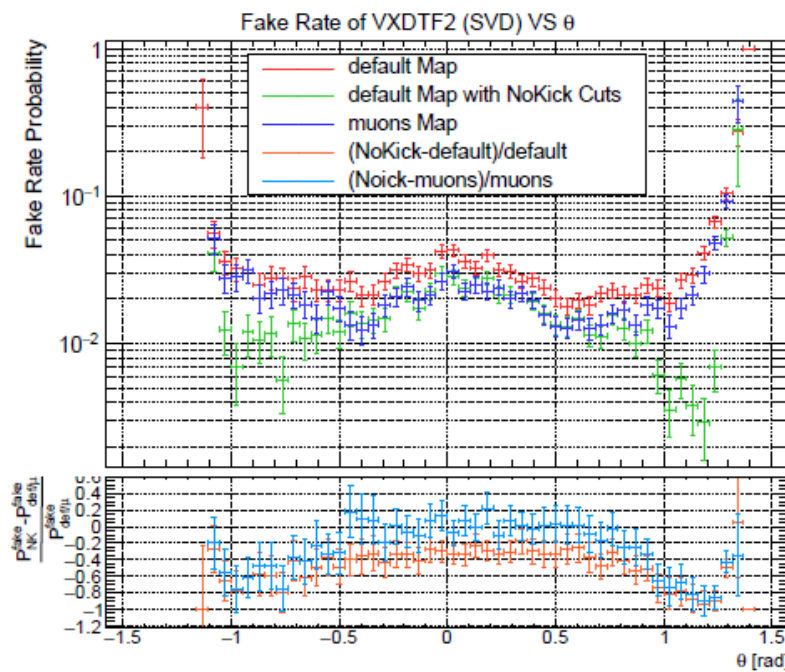
# Fake Rate— Confronto Sector Map



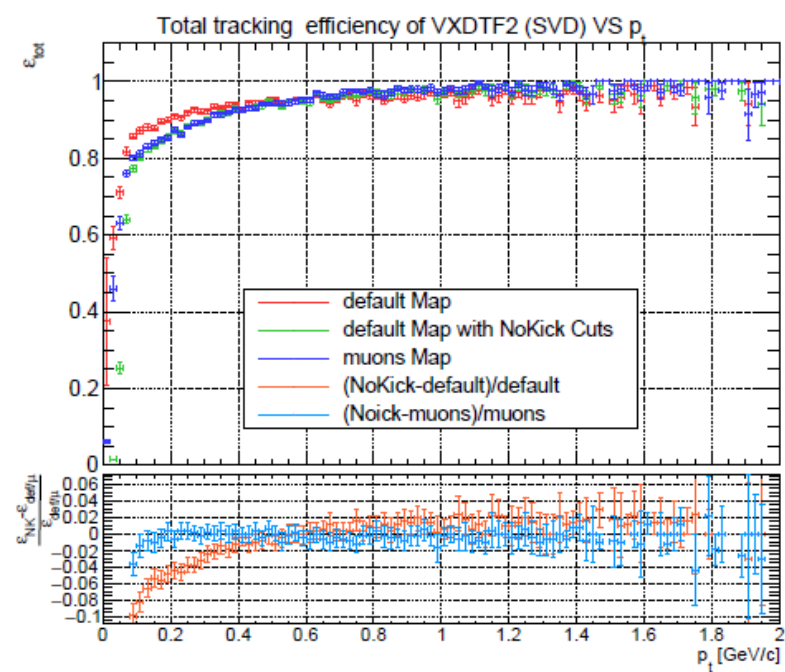
(a)



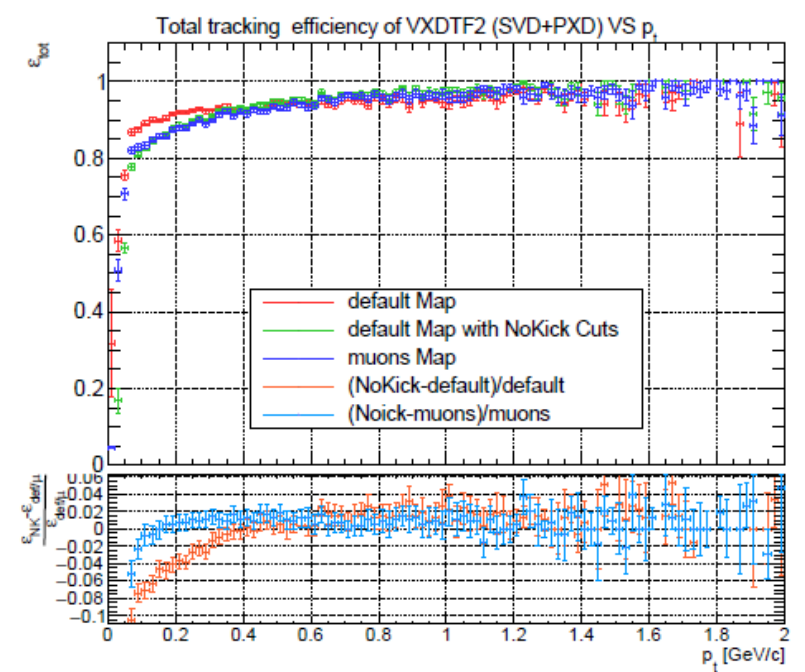
(b)



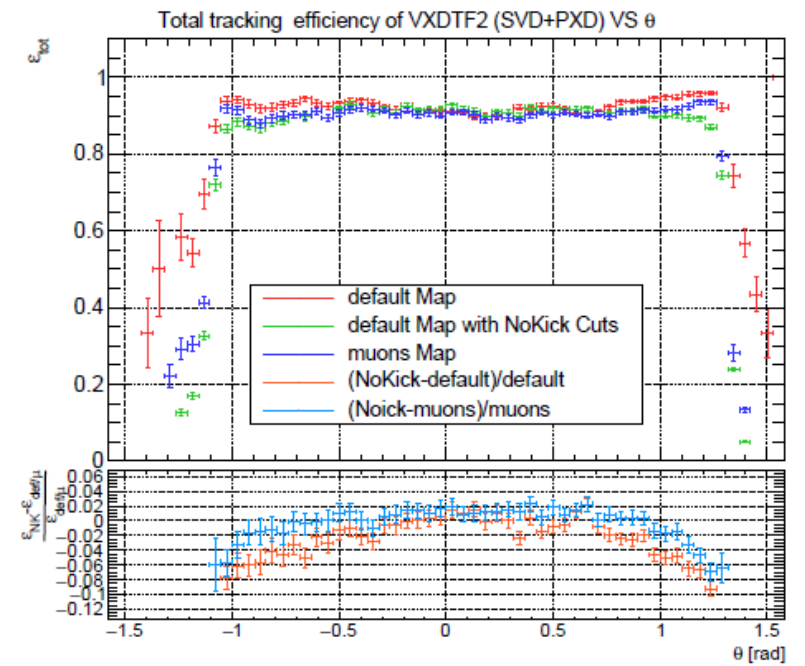
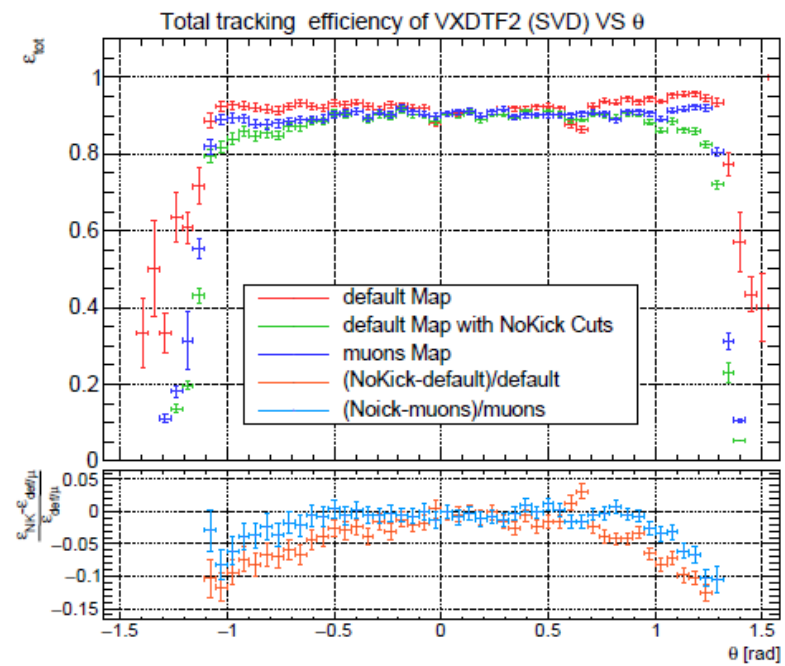
# Efficienza di tracking – Confronto Sector Map



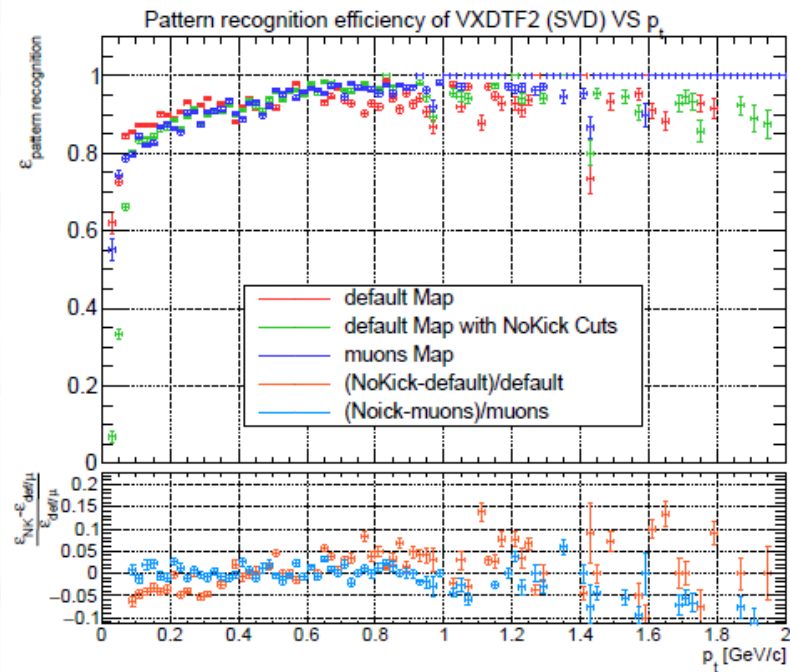
(a)



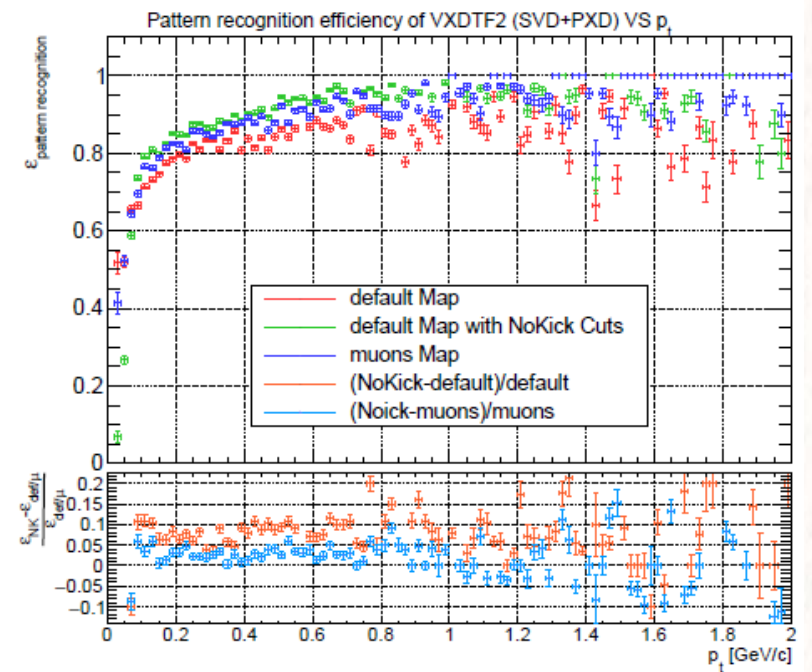
(b)



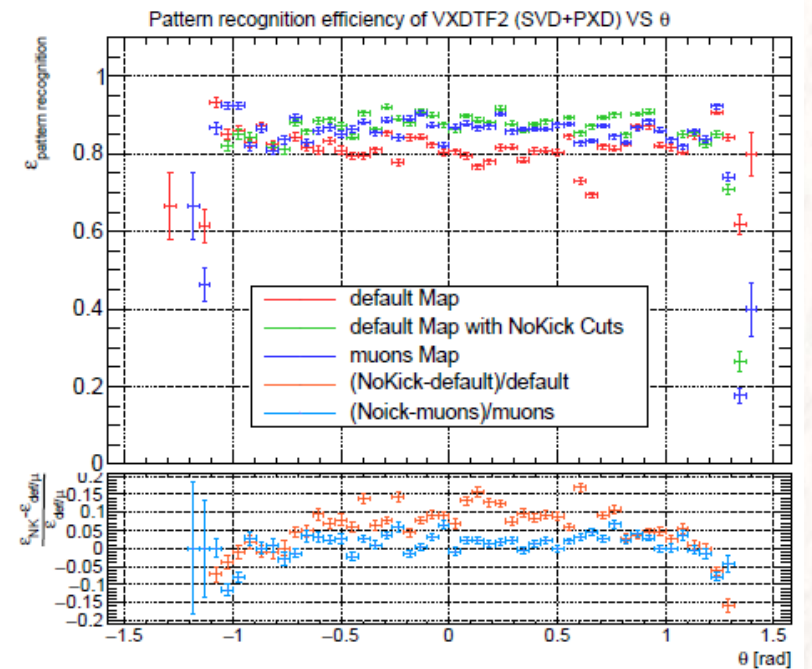
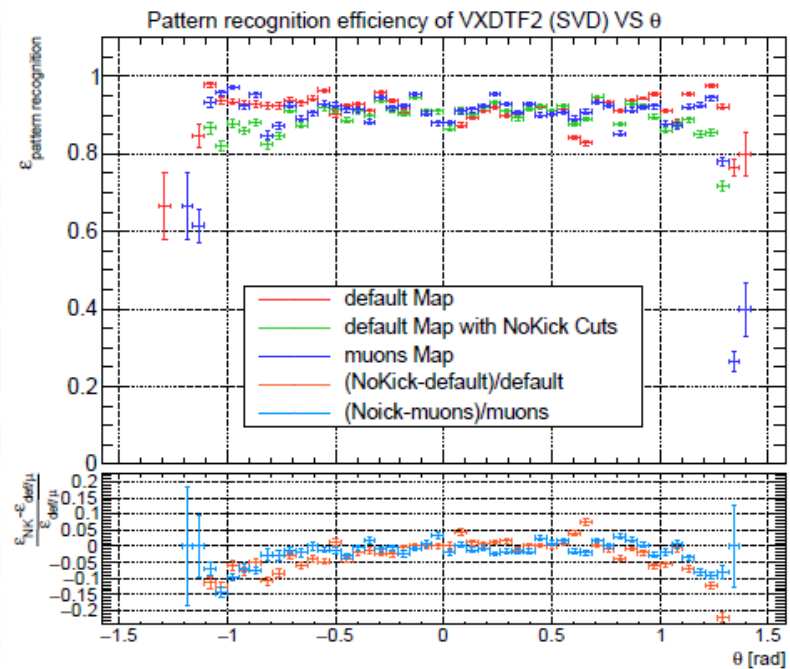
# Efficienza di Pattern Recognition – Confronto SecMap+bkg



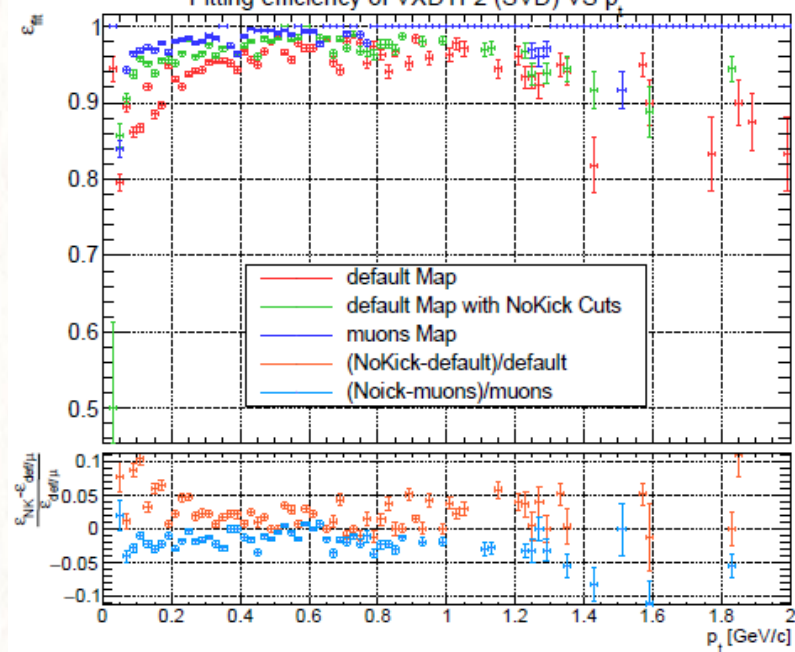
(a)



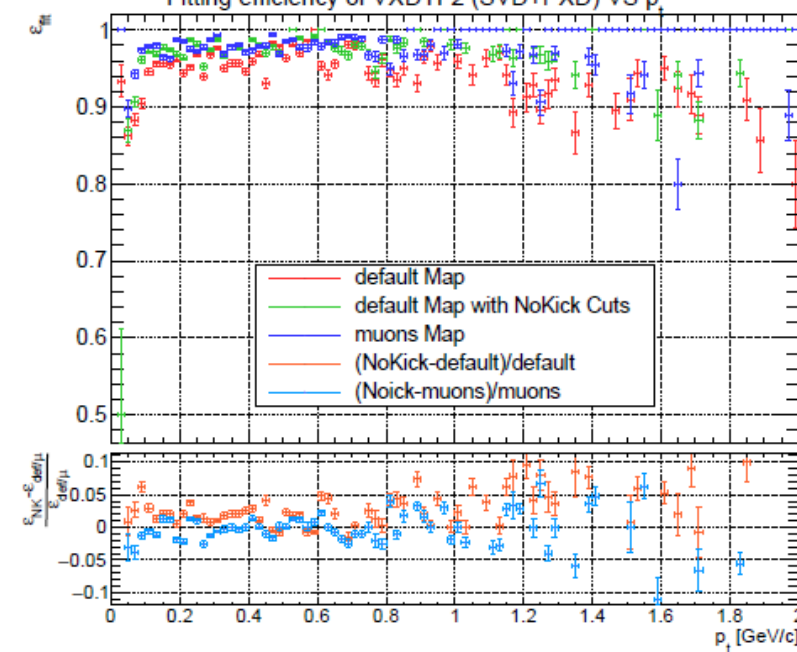
(b)



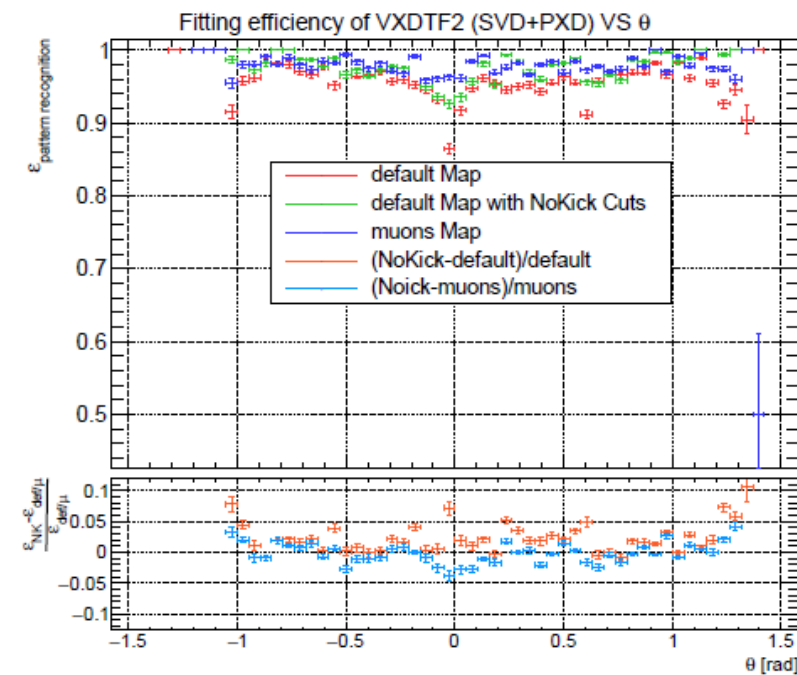
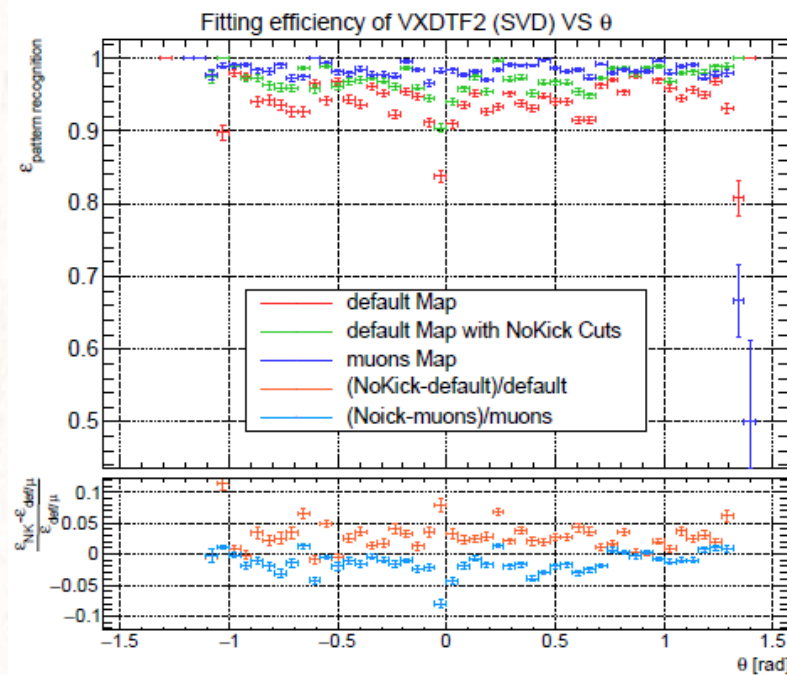
# Efficienza di fit – Confronto SecMap+bkg



(a)

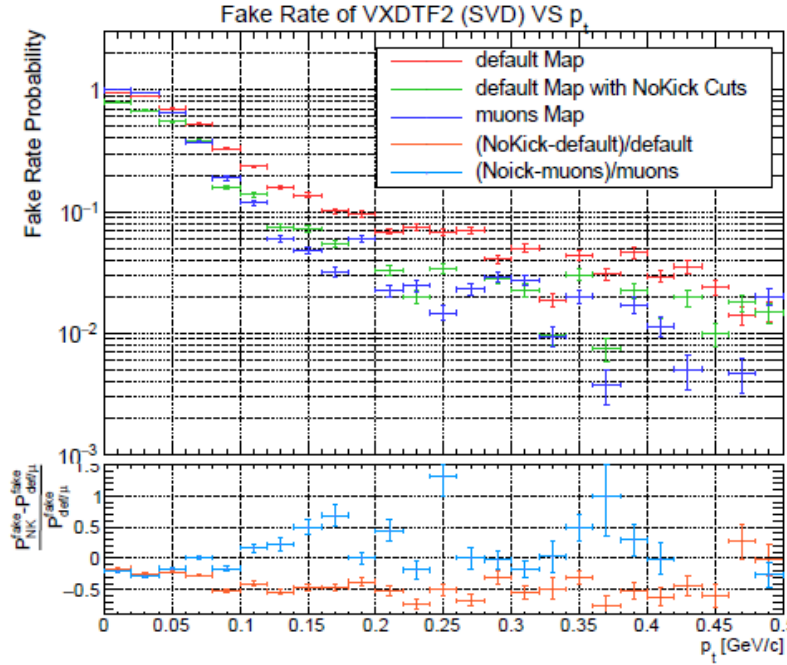


(b)

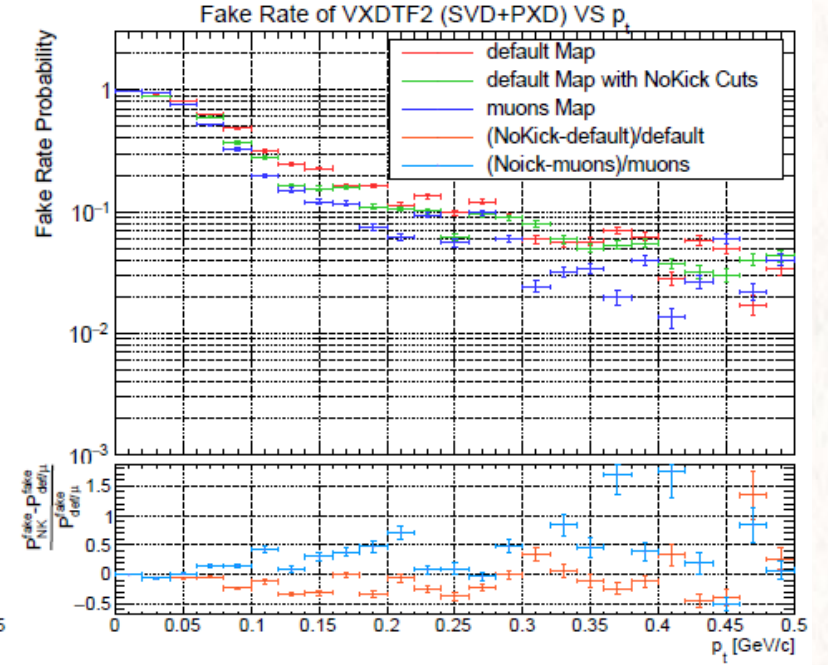




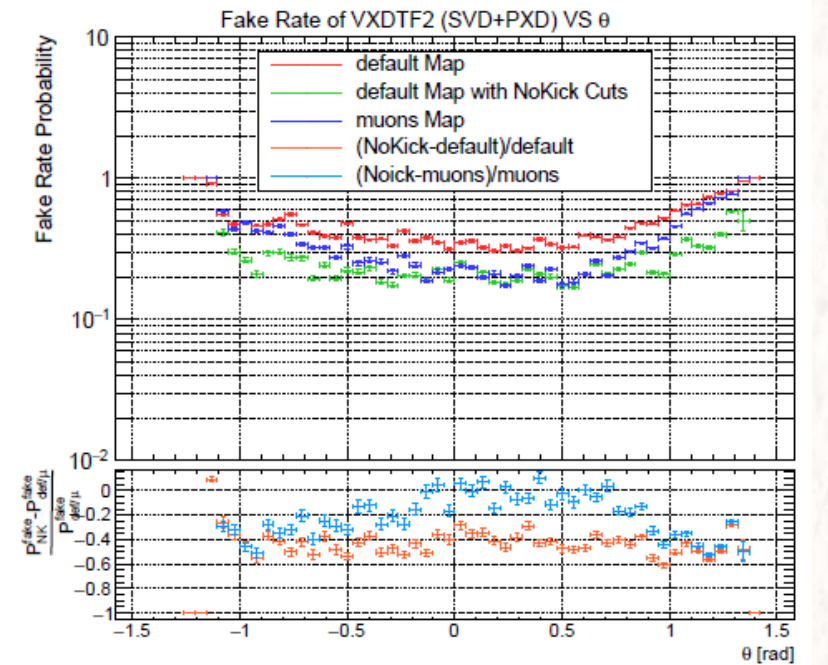
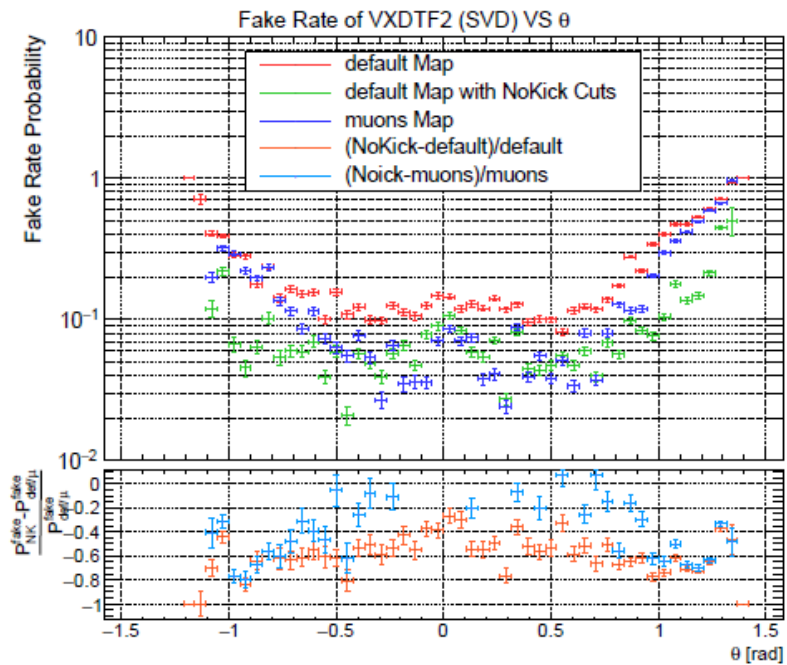
# Fake Rate – Confronto SecMap+bkg



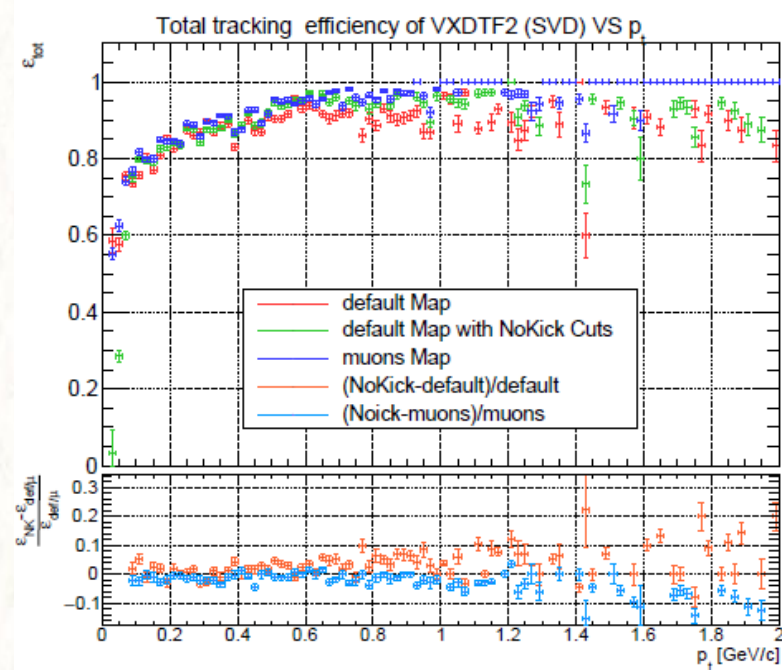
(a)



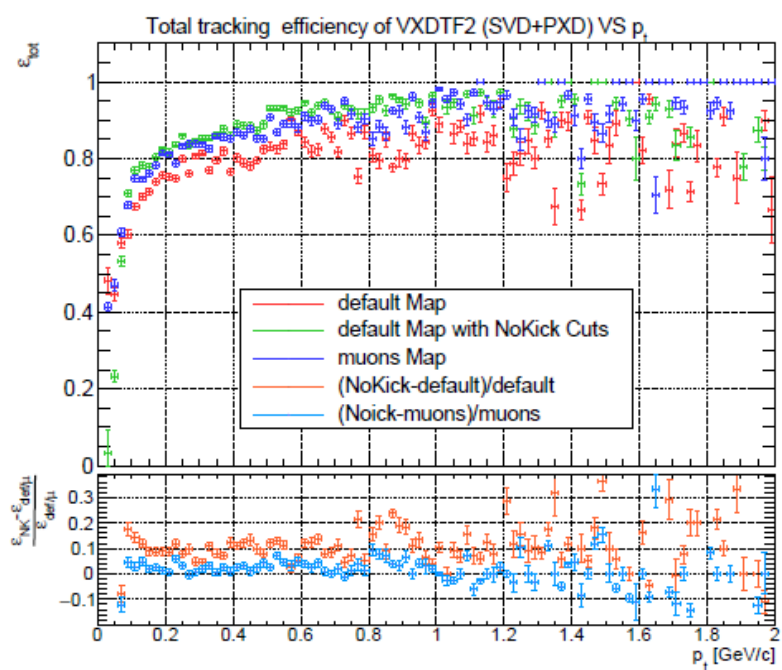
(b)



# Efficienza di tracking – Confronto SecMap+bkg



(a)



(b)

