UNIVERSITÀ DEGLI STUDI DI MILANO

Facoltà di Scienze e Tecnologie
Laurea triennale in Fisica

Studio della purezza del campione di fotone inclusivo del run a 8 TeV (2012) di Atlas con il metodo *Isolation Template Fit* e caratterizzazione del modello di accesso ai dati

Relatori:

Dott. Francesco Prelz
Dott. Leonardo Carminati

Elaborato finale di:
Giacomo Parolini
Matricola 776092

PACS: 13.85.Qk

Anno accademico 2012–2013

Indice

Capitolo 1

Introduzione

Lo scopo di questa tesi è duplice: da un lato, calcolare la purezza del campione di fotone inclusivo analizzato nel 2012 dall'esperimento ATLAS a LHC, e dall'altro cercare di ottimizzare il metodo di accesso ai dati stessi, messi a disposizione dalla rete di calcolo distribuito WLCG (*Worldwide LHC Computing Grid*) a numerosi centri di ricerca in tutto il mondo.

Questo secondo obiettivo si è dimostrato di importanza fondamentale per la buona riuscita del primo, dal momento che la dimensione dell'intero campione di dati raccolti da ATLAS nel 2012 è molto ingente (piú di 700 milioni di eventi), ed è necessario un approccio computazionalmente efficiente per poterlo analizzare anche molte volte in tempi contenuti. A questo fine, ho creato un framework di analisi consistente in vari programmi e script che, lavorando in sinergia con RootCore [1], DQ2 [2] e XRootD [3] (tre strumenti per l'analisi distribuita dei dati), consente di condurre un algoritmo di analisi su una grande mole di dati accedendovi in remoto, lavorando però su macchine locali. Questo approccio porta diversi vantaggi: il fatto che l'analisi venga eseguita localmente e non su macchine remote consente un maggiore controllo sui propri *job* (da tempi di latenza più controllabili alla possibilità di salvare log ed eseguire profiling in modo semplice), mentre l'accesso remoto ai dati distribuiti via WLCG evita di dover salvare su storage locali i circa 70 TB di dati analizzati e di generare un ingente traffico di rete durante il trasferimento degli stessi dagli altri centri di calcolo.

Il calcolo vero e proprio della purezza viene invece eseguito mediante il metodo dell'*isolation template fit* su un sotto-campione di eventi che hanno passato una serie di tagli di qualità e di selezione che consentono di ridurre la contaminazione di eventi di fondo nel campione selezionato: la purezza è definita proprio come il rapporto tra il numero di eventi di segnale e il numero di eventi totali selezionati. Il metodo del *template fit* consiste nel costruire una *probability density function* (PDF) modello come la somma di una PDF di segnale e di una PDF di background: $M = N_s \times S + N_b \times B$, dove N_s e N_b sono i pesi di segnale e background; un appropriato fit di M sul campione di dati reali permette di determinare tali coefficienti, e la purezza può quindi essere calcolata come $P = \frac{N_s}{N_s+N_b}$.

La struttura di questo elaborato sarà pertanto suddivisa in tre sezioni logiche: nella prima verrà descritto brevemente LHC e in particolare l'esperimento ATLAS, nella seconda verrà descritto il framework di analisi ed infine sarà descritto in maggiore dettaglio l'*isolation template fit* e verranno presentati i risultati ottenuti.

Capitolo 2

LHC

Il Large Hadron Collider, LHC, è un acceleratore di protoni (e ioni pesanti) costruito dal CERN nei pressi di Ginevra. Questo acceleratore è andato a sostituire il suo predecessore, il LEP (Large Electron-Positron), diventando effettivamente operativo nel 2008.

2.1 Caratteristiche

LHC è stato progettato per raggiungere energie molto più elevate di qualunque altro acceleratore esistente: fino a 14 TeV di energia di centro di massa e una luminosità istantanea di $10^{34}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$ a massimo regime[1]. Nel Run I in realtà è stata raggiunta un'energia di centro di massa pari a 7 TeV nel 2011 e 8 TeV nel 2012, con una luminosità effettiva di $8 \times 10^{33}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$.

L'acceleratore ha un diametro di circa 4.3 km per una circonferenza totale di approssimativamente 27 km (vedi fig. 1). [4]

I fasci protonici, che possono essere fatti viaggiare a velocità estremamente prossime a quelle della luce ($\sim 99,999999\%$), sono suddivisi in circa 2800 *bunch* contenenti $n \sim 10^{11}$ protoni l'uno, con una sezione trasversale di 16.6 µm e una lunghezza di 7.55 cm, che vengono fatti collidere circa ogni 50 ns. Questo intervallo di tempo, teoricamente riducibile fino a 25 ns, è stato mantenuto alto durante il Run I per ragioni tecniche.

Il campo magnetico necessario a mantenere gli adroni in orbita circolare è fornito da oltre 1600 magneti superconduttori in lega Ni−Ti raffreddati alla temperatura di 1.9 K, che producono un campo di circa 8 T.

2.2 Perché LHC?

Gli obiettivi di ricerca di LHC sono molteplici; tra quelli principali vi sono:

1) **comprendere l'origine della massa**: in particolare, verificare sperimentalmente l'esistenza del bosone di Higgs [5] [6] - traguardo che è stato raggiunto nel 2013, e che è valso il Premio Nobel per la Fisica a F. Englert e P. W. Higgs per averne previsto l'esistenza;

[1] La luminosità è definita come: $\mathrm{L} = f \frac{n_1 n_2}{4\pi\sigma_x\sigma_y}$, dove f è il rate di collisioni, $n_{1,2}$ il numero di particelle per *bunch* e $\sigma_{x,y}$ le dimensioni del fascio in direzione longitudinale e trasversale.

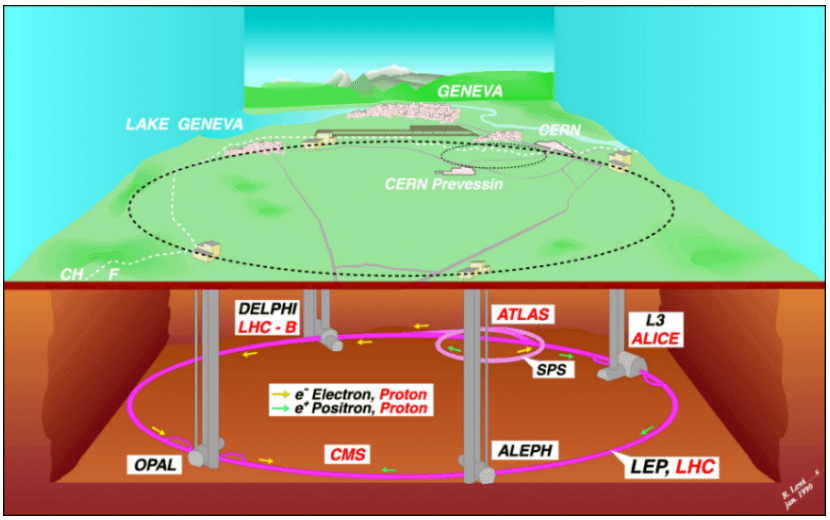


Figura 1: Panoramica di LHC

2) **rispondere a domande ancora aperte**: come la natura della materia ed energia oscura, che secondo alcune teorie costituiscono il 95% della materia dell'Universo e spiegare l'asimmetria materia-antimateria nell'Universo;

3) **studiare modelli alternativi al Modello Standard**: in particolare cercare le particelle supersimmetriche, che dovrebbero avere una massa elevata ma teoricamente raggiungibile da LHC, e cercare le dimensioni extra previste dalla teoria delle stringhe;

4) **studiare condizioni sperimentali ancora inesplorate**: le collisioni ad altissima energia che avvengono in LHC consentono di studiare in maggiore dettaglio stati esotici della materia, come il plasma di quark-gluoni, e forniscono una panoramica sulla condizione dell'Universo primordiale.

2.3 Interazione protone-protone

L'interazione di maggior interesse in LHC è naturalmente la collisione tra due protoni, che avviene ogni volta che i due fasci che viaggiano in direzioni opposte all'interno dell'acceleratore vengono fatti incrociare.

Il numero di collisioni al secondo (*rate*) si calcola come:

$$\mathrm{R} = \sigma \mathrm{L} \tag{2.1}$$

dove σ è la sezione d'urto dell'urto totalmente anelastico protone-protone e L la luminosità dell'acceleratore. Poiché la sezione d'urto anelastica p-p all'energia di centro di massa di 7 TeV è pari a circa 70 mb[2], si ottiene un rate pari a:

$$\mathrm{R} = 70\,\mathrm{mb} \times 10^{34}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1} \approx 3.5 \times 10^{8}\,\mathrm{s}^{-1} \tag{2.2}$$

il che significa un numero di collisioni per *bunch crossing* pari a:

$$\mathrm{N} = \mathrm{R} \times \tau_{bc} = 3.5 \times 10^{8}\,\mathrm{s}^{-1} \times 50\,\mathrm{ns} \approx 18 \tag{2.3}$$

dove $\tau_{bc} = 50\,\mathrm{ns}$ è la separazione temporale tra due *bunch*.

[2]La sezione d'urto è misurata in *barn*: $1\,\mathrm{b} = 10^{-28}\,\mathrm{m}^{-2} = 10^{-24}\,\mathrm{cm}^{-2}$

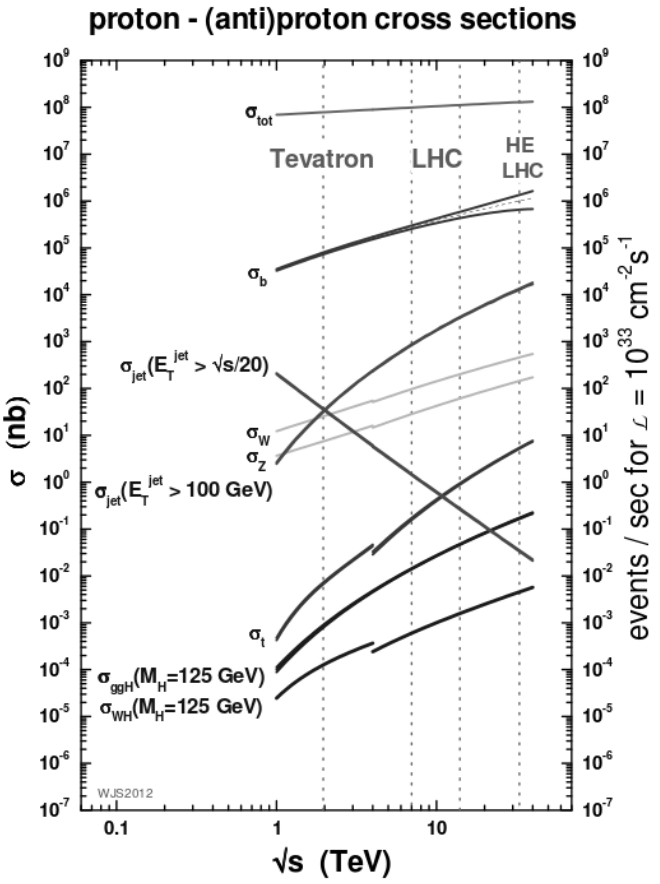


Figura 2: Sezioni d'urto in funzione di $\sqrt{s}$ per diversi processi ad LHC. La sezione d'urto di produzione di fotoni è di circa 10 nb a 14 TeV.

Queste collisioni possono essere suddivise in due tipologie: **soft** e **hard**.

- Nelle collisioni **soft** il parametro d'impatto[3] tra i due protoni incidenti è molto grande, quindi l'urto è poco centrale e il momento trasferito è piccolo. Questo implica che non si registrano scattering ad angoli elevati e il momento finale dei protoni è prevalentemente longitudinale (mentre il momento trasversale p_T è dell'ordine dei 500 MeV);

- Nelle collisioni **hard** al contrario l'urto è fortemente centrale e si ha un grande momento trasferito, prevalentemente della direzione trasversale (con un p_T dell'ordine di decine/centinaia di GeV): questo processo può essere visto come un urto tra due singoli partoni che compongono i protoni incidenti. In questo tipo di collisione, lo stato finale può essere popolato di nuove particelle altamente massive createsi durante l'urto stesso.

Il numero di eventi *hard* è molto piccolo, tipicamente di circa 5 ordini di grandezza inferiore rispetto a quelli *soft* (fig. 2). Il caso *hard* è quello che andremo a studiare, trattandosi del caso di interesse nell'ambito della fisica delle particelle.

[3] cioè la distanza tra le due particelle sul piano d'impatto.

Poiché l'hard scattering viene interpretato come collisione diretta tra due partoni, l'energia di centro di massa dell'urto sarà data da:

$$\sqrt{\hat{s}} = \sqrt{x_a x_b s} = x\sqrt{s} \tag{2.4}$$

dove $x_{a,b}$ sono le frazioni del momento del protone trasportate dai singoli partoni coinvolti nell'urto (supposte uguali per semplicità), s l'energia di centro di massa nominale della macchina e $\hat{s}$ l'energia di centro di massa effettiva della collisione. Produrre una particella massiva richiede che la frazione di energia trasportata dai due partoni sia sufficientemente elevata: ad esempio, produrre una particella da 1 TeV con un'energia di centro di massa della macchina di 8 TeV richiede che $x \gtrsim 0.125$.

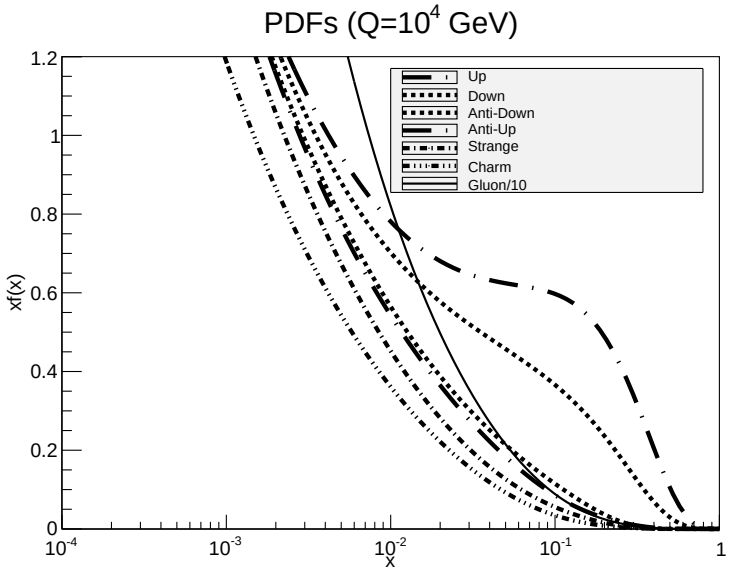


Figura 3: Funzioni di distribuzione dei partoni per $Q^2 = 10^4\,\mathrm{GeV}^2$

La sezione d'urto di un generico processo *hard* è:

$$\sigma = \sum_{a,b} \int \mathrm{d}x_a \mathrm{d}x_b f_a(x, Q^2) f_b(x, Q^2) \hat{s}_{ab}(x_a, x_b) \tag{2.5}$$

dove $\hat{s}_{ab}$ è la sezione d'urto dell'interazione partone-partone e $f_{a,b}(x, Q^2)$ le *partonic distribution function* (PDF) dei protoni[4] (fig. 3).

2.4 Condizioni e problemi sperimentali

Nello studio delle interazioni protone-protone a LHC si riscontrano diverse difficoltà a livello sperimentale; le due principali sono dovute al **pile-up** degli eventi e al **fondo QCD**.

[4] $Q^2 = -q^2 = 4EE' \sin\theta/2^2$ è proporzionale al momento trasferito (in pratica l'energia) del processo di scattering .

Il **pile-up** è la sovrapposizione di più eventi consecutivi dovuta alla brevità dell'intervallo temporale τ_{bc} che intercorre tra essi e all'elevata densità dei pacchetti di protoni. Come abbiamo visto in 2.2 e 2.3, i *bunch* collidono ogni 50 ns con una media di circa 18 eventi per *bunch crossing*: questo implica che ogni 50 ns si hanno circa 1000 particelle cariche che finiscono nel rivelatore[5]. Un evento di hard scattering verrà dunque a sovrapporsi a ~ 20 eventi soft prodotti dallo stesso *bunch crossing* più gli eventuali residui nei rivelatori dell'interazione precedente (rispettivamente, sovrapposizione *in-time* e *out-of-time*).

Le misure adottate da LHC per minimizzare il problema del *pile-up* sono le seguenti:

- il *tempo di risposta* dei rivelatori è estremamente breve (dell'ordine dei 20-50ns), in modo da eliminare buona parte dei residui delle interazioni precedenti;
- la *risoluzione spaziale* dei rivelatori è molto piccola, in modo da minimizzare la probabilità che due particelle provenienti da eventi diversi siano rilevate dal medesimo detector;
- le componenti del rivelatore sono resistenti al notevole flusso di radiazioni prodotte dalle frequenti collisioni protone-protone (dell'ordine dei 10^{17} neutroni cm^{-2} nell'arco di 10 anni).

Il **fondo QCD** è costituito prevalentemente da jet di particelle prodotti dalla frammentazione dei quark e gluoni nello stato finale. Dal momento che questo processo è guidato dall'interazione forte, la sua sezione d'urto è elevata, per cui la produzione di jet risulta essere il processo dominante. Molti processi interessanti per LHC sono interazioni a piccole sezioni d'urto e quindi rare e più difficili da osservare, spesso mascherate da eventi con jet nello stato finale.

2.5 Gli esperimenti di LHC

Esistono 4 esperimenti principali che utilizzano LHC:

ATLAS e CMS : sono gli esperimenti più completi (*general purpose detectors*) progettati ed ottimizzati principalmente per la ricerca del bosone di Higgs e i segnali di particelle supersimmetriche, oltre che per lo studio della fisica del quark top, della violazione di CP nel decadimento dei mesoni B e della natura fondamentale dei fermioni. Oltre alle diverse tecnologie usate per i rivelatori, la differenza fondamentale tra essi è che CMS (Compact Muon Solenoid) utilizza un unico campo magnetico per tutti i rivelatori, mentre in ATLAS (A Toroidal LHC ApparatuS) viene generato un campo solenoidale interno all'inner detector e un sistema di campi toroidali in cui sono immersi i calorimetri esterni.[6]

LHCb : è un rivelatore specializzato nello studio del mesone B e il suo scopo principale è la ricostruzione delle proprietà del quark b: infatti la copertura del rivelatore è concentrata solo verso la direzione del fascio dove la probabilità dei decadimenti B è massima.

ALICE (A Large Ion Collider Experiment): è un esperimento dedicato allo studio delle collisioni tra ioni pesanti e delle proprietà delle interazioni tra nuclei ad altissima energia.

[5] considerando solo le particelle con *pseudorapidità* $\eta < 2.5$.

[6] per una descrizione più dettagliata, si rimanda al capitolo 3

È di particolare interesse la produzione del plasma di quark-gluoni e lo studio delle collisioni pp, per cercare di risolvere alcuni problemi chiave della QCD.

Due esperimenti minori, TOTEM e LHCf, si occupano di studiare gli scattering a basso momento trasferito.

In totale, questi esperimenti producono una mole di dati pari a 15 PB[7] all'anno.

[7] 1 PB = 10^{15} byte

Capitolo 3

ATLAS

Come già introdotto nel capitolo 2.5, ATLAS[1] è uno dei quattro esperimenti principali di LHC, il più importante insieme a CMS. [7]

Uno degli obiettivi principali di ATLAS fino ad oggi è stato certamente la ricerca del bosone di Higgs, la cui esistenza è stata confermata sperimentalmente nel 2012, tuttavia le sue potenzialità vanno ben oltre questa ricerca, trattandosi a tutti gli effetti di un *general purpose detector* in grado di rivelare una gamma molto vasta di particelle, tra cui fotoni, elettroni, muoni e vari tipi di adroni.

3.1 Nomenclatura

Definiamo fin da subito la nomenclatura standard utilizzata per descrivere l'apparato:

- L'asse z è l'asse parallelo alla direzione dei fasci adronici;
- L'asse x è la direzione perpendicolare a z che punta verso il centro dell'anello di LHC (con verso positivo entrante);
- L'asse y è la direzione perpendicolare a x e z;
- L'angolo *azimutale* ϕ è misurato attorno all'asse z e l'angolo *polare* θ è misurato partendo dall'asse z.

Date queste definizioni, si definisce la **pseudorapidità** come (vedi fig. 4):

$$\eta = -\ln(\tan\frac{\theta}{2}) \tag{3.1}$$

Il momento trasverso p_T, e in generale qualunque variabile 'trasversa', è definito nel piano $\mathrm{x}-\mathrm{y}$, mentre la distanza nello spazio $\eta-\phi$ è data da:

$$\Delta\mathrm{R} = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2} \tag{3.2}$$

[1] Acronimo di *A Toroidal LHC ApparatuS*: un apparato toroidale di LHC

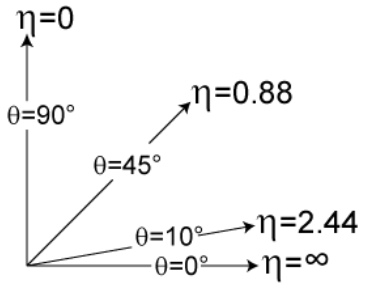


Figura 4: Valori di pseudorapidità per diversi valori di θ. La direzione orizzontale è parallela all'asse z.

3.2 Struttura

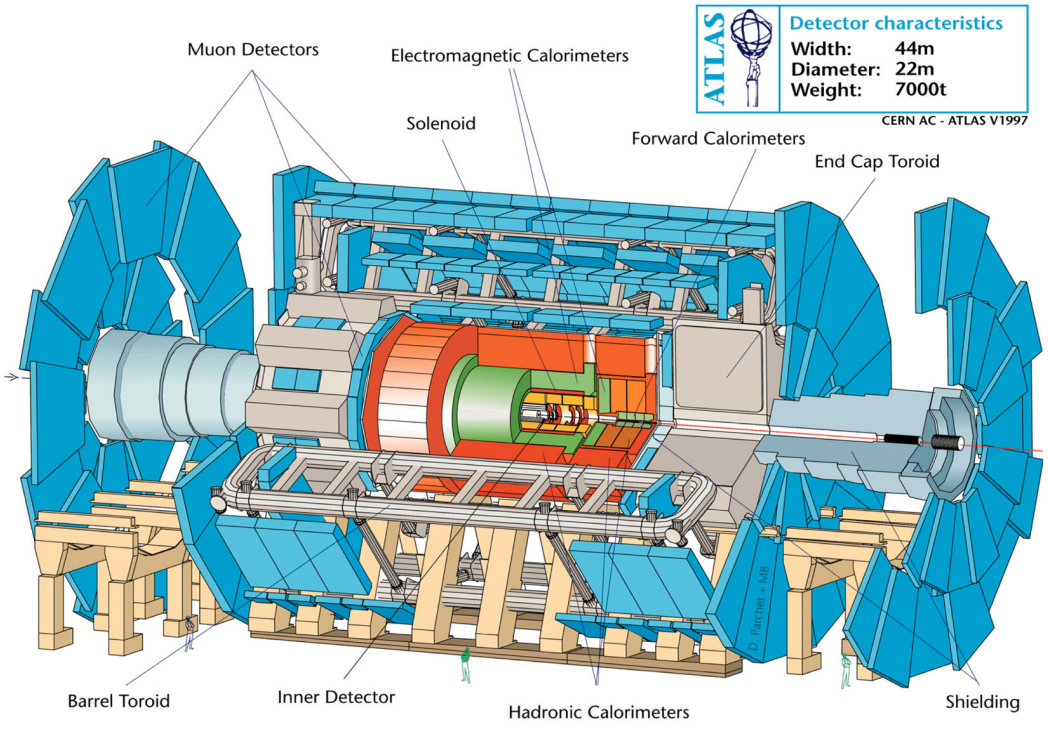


Figura 5: Struttura del rivelatore ATLAS

La struttura di ATLAS (figg. 5 e 6) è quella di un cilindro di circa 22 m di diametro per 44 m di lunghezza, costituito da una serie di sezioni concentriche che costituiscono le sue tre parti principali:

1) L'*inner detector*, dove vengono rilevate le tracce delle particelle cariche quali protoni ed elettroni;

2) i calorimetri, dove viene misurata l'energia totale di tutte le particelle emesse ad eccezione dei muoni (e dei neutrini);

3) il detector di muoni, dove viene misurato anche il momento dei muoni prodotti dall'interazione, e il sistema di magneti esterno.

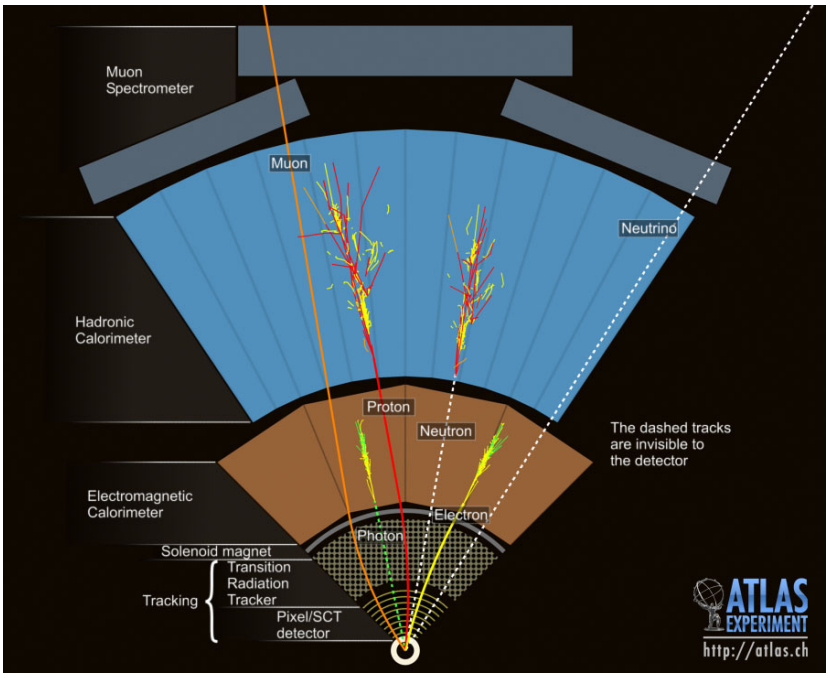


Figura 6: Particelle nel rivelatore ATLAS

Come vedremo, ognuna di queste sezioni è ulteriormente suddivisa in più strati. La massa totale del rivelatore è circa 7000 tonnellate.

3.2.1 L'inner detector

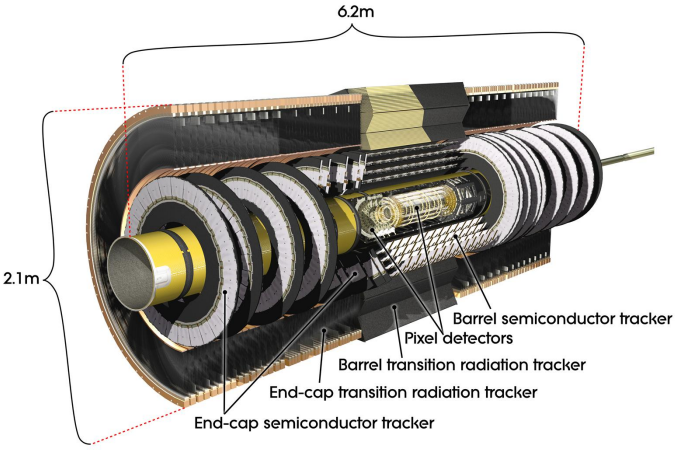
L'inner detector è la parte più interna del rivelatore, ed è idealmente contenuto in un cilindro lungo 7 m per 1.15 m di raggio. La sua funzione principale consiste nel rilevare le particelle cariche tramite l'interazione fra queste e materiali traccianti posti a distanza regolare ad alta granularità.

Il cilindro è costituito da una parte centrale (*barrel*) chiusa da entrambi i lati da due tappi (*end-caps*), sempre composti da rivelatori traccianti, che completano la copertura in pseudorapidità fino a $|\eta| < 2.5$.

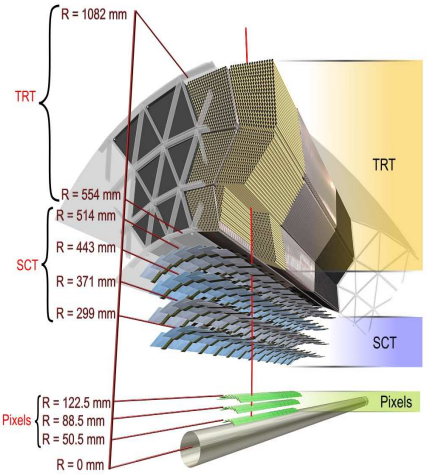
L'inner detector è suddiviso in 3 strati, la cui risoluzione è tanto maggiore quanto più interno è lo strato.

- Lo strato più vicino al fascio adronico (suddiviso in 3 layer posti a 50, 90 e 120 mm dal fascio) è il **detector a pixel**, composto da circa 1750 moduli di 2×6 cm, ognuno dei quali contiene circa 47 000 pixel delle dimensioni di 50×400 µm. Questo rivelatore possiede una risoluzione di circa 12 µm lungo la direzione ϕ e di circa 66 µm in direzione z.

- Lo strato intermedio (299 mm $<$ r $<$ 514 mm) è il **tracciatore a semiconduttori** (SCT), che ha un funzionamento simile al detector a pixel ma al posto di questi ultimi utilizza dei microstrip di silicio di 80 µm $\times$ 12 cm che rendono più pratico coprire un'area maggiore rispetto al detector a pixel. La risoluzione di questo rivelatore è circa 16 µm in direzione ϕ e 580 µm in direzione z.



(a) L'inner detector.



(b) Strati dell'inner detector.

– Lo strato esterno è il **tracciatore a radiazione di transizione** (TRT), composto da una serie di tubi (*straw tubes*) lunghi 144 cm e dal diametro di 4 mm. La tecnologia degli straw tubes permette di coprire spazi maggiori a costi contenuti e di ottenere informazioni non solo spaziali ma anche di identificazione delle particelle. Questi fattori compensano la minore risoluzione intrinseca del dispositivo.

3.2.2 I calorimetri

Il sistema di calorimetri è formato da due strati: all'interno si trova il calorimetro elettromagnetico (che ricopre la regione $|\eta| < 3.2$), mentre all'esterno si trova il calorimetro adronico (che ricopre la regione $|\eta| < 4.9$).

La loro funzione è di misurare l'energia totale delle particelle incidenti mediante assorbimento nel materiale del rivelatore.

Il calorimetro elettromagnetico

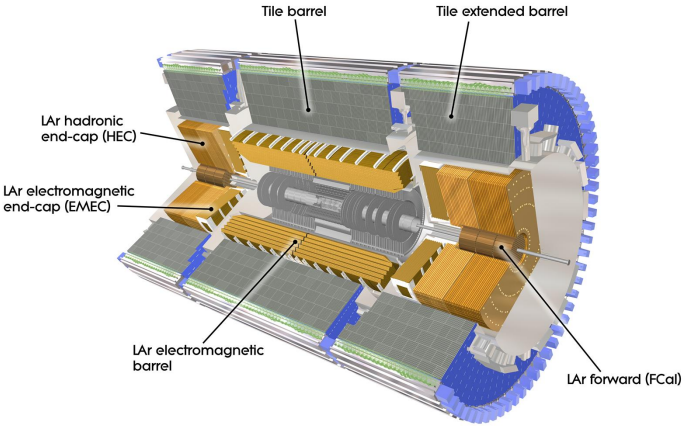
Il calorimetro elettromagnetico è racchiuso in un cilindro del diametro di 4.5 m e che si estende per 6.65 m di lunghezza in entrambi i versi del fascio; come l'inner detector, è formato da una parte centrale cilindrica racchiusa da due tappi alle estremità.

Si tratta di un calorimetro a sampling che utilizza l'argon liquido come materiale attivo ed il piombo come materiale passivo. All'interno dell'argon liquido sono immersi degli elettrodi e degli assorbitori disposti in una struttura a fisarmonica: questo consente di ricoprire l'intero angolo azimutale uniformemente senza lasciare zone cieche. Il calorimetro è segmentato sia lungo le direzioni angolari che lungo quella longitudinale in circa 190 000 celle per la misura della posizione degli sciami elettromagnetici prodotti dall'interazione delle particelle con il calorimetro.

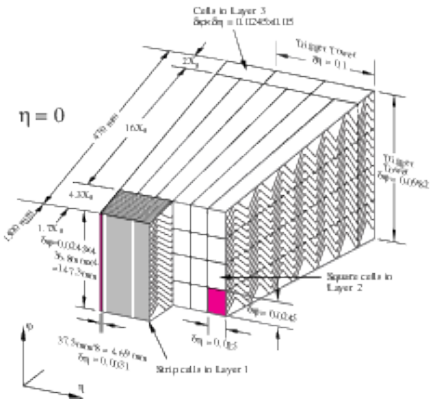
La risoluzione energetica nominale è:

$$\frac{\sigma(E)}{E} \approx \frac{(10\% \div 17\%)}{\sqrt{E}} \pm 0.7\% \tag{3.3}$$

mentre quella angolare è di circa 0.025 rad.



(a) Struttura del calorimetro elettromagnetico



(b) Segmentazione interna del calorimetro

Sezione calorimetro	Copertura in η		Granularità ($\Delta\eta \times \Delta\phi$)
	barrel	end-cap	
Presampler	$\lvert\eta\rvert < 1.54$	$1.5 < \lvert\eta\rvert < 1.8$	0.025×0.1
Strip	$\lvert\eta\rvert < 1.476$	$1.375 < \lvert\eta\rvert < 3.2$	0.003×0.1
Middle	$\lvert\eta\rvert < 1.475$	$1.375 < \lvert\eta\rvert < 3.2$	0.025×0.025
Back	$\lvert\eta\rvert < 1.35$	$1.5 < \lvert\eta\rvert < 2.5$	0.05×0.025

Tabella 3.1: Copertura e granularità delle componenti del calorimetro elettromagnetico

Il calorimetro adronico

All'esterno del calorimetro elettromagnetico si trova il calorimetro adronico, che occupa un cilindro di 8.5 m di diametro e si estende per 6.1 m di lunghezza lungo la direzione del fascio.

Il materiale sensibile è uno scintillatore, mentre il ferro viene usato come materiale assorbente e copre una regione $|\eta| < 4.9$ grazie alla solita struttura barrel + end-cap. Il calorimetro è suddiviso in celle di dimensioni $\Delta\eta \times \Delta\phi = 0.1 \times 0.1$ rad che forniscono quindi una risoluzione minore rispetto al calorimetro elettromagnetico[2].

I tappi che coprono le estremità del cilindro sono costituiti da due ruote indipendenti in cui si alternano piatti in rame e strati sensibili riempiti di argon liquido. La lunghezza totale del calorimetro per $\eta = 0$ è pari a circa 11 volte la lunghezza di interazione nucleare, il che consente un ottimo contenimento longitudinale dei jet ad alta energia.

3.2.3 Lo spettrometro di muoni

Lo spettrometro di muoni è l'ultimo rivelatore vero e proprio di ATLAS, ed espleta la funzione di misurare la quantità di moto dei muoni che lo attraversano tramite la ricostruzione della loro traiettoria e della deviazione di quest'ultima causata dal campo magnetico generato dai magneti esterni. Anch'esso è formato da un cilindro e da due tappi alle estremità e la regione di pseudorapidità coperta è $|\eta| < 2.7$.

[2]La risoluzione angolare è infatti data approssimativamente dalla dimensione di tali celle.

Questo rivelatore è necessario perchè i muoni sono le uniche particelle che attraversano tutti gli strati interni di ATLAS depositandovi una quantità di energia minima. La difficoltà nel catturare i muoni spiega le dimensioni sensibilmente più grandi di questo apparato rispetto ai rivelatori interni: il raggio del detector di muoni va da 4.25 m nella parte più interna a 11 m in quella più esterna.

Il suo funzionamento è simile a quello dell'inner detector: tracciare le traiettorie delle particelle che lo attraversano e misurarne la deviazione dovuta al campo magnetico esterno; le differenze principali rispetto all'inner detector − oltre al volume molto maggiore − sono la minore risoluzione spaziale e la diversa configurazione del campo magnetico (che qui, a differenza che nell'inner detector, non è spazialmente uniforme).

La risoluzione spaziale dello spettrometro è circa di $5 \div 10$ mm, mentre quella temporale è inferiore ai 25 ns grazie a un sistema di camere a trigger con elevata velocità di risposta. Nell'insieme, il detector è progettato per misurare il momento di una particella di 100 GeV con una precisione del 3% e di una particella di 1 TeV con una precisione del 10% circa.

3.2.4 Il sistema di magneti

Lo scopo del sistema magnetico di ATLAS è permettere di misurare con precisione il momento delle particelle prodotte dall'interazione dei fasci adronici. Il momento trasverso può essere ricavato tramite la formula:

$$p_T = r_g |q| B \tag{3.4}$$

dove r_g è il *giroraggio* (o raggio di ciclotrone) della traiettoria della particella con carica q immersa nel campo magnetico di modulo B di cui si vuole misurare il momento.

Il sistema di magneti di ATLAS è formato da due parti: un magnete solenoidale centrale (*Central Solenoid*), che provvede ad immergere l'inner detector in un campo magnetico altamente uniforme, e un sistema di magneti toroidali esterno, formato da otto *Barrel Toroid* centrali più due *End-Cap Toroid* laterali, che circondano lo spettrometro di muoni. Le dimensioni complessive di questo apparato sono 26 m di lunghezza per 20 m di diametro.

Il magnete centrale produce un campo magnetico di 2 T, la cui alta intensità permette di deviare anche particelle con grande momento e la cui uniformità consente di misurare con grande precisione le loro traiettorie. I magneti esterni producono un campo magnetico ancora più intenso (attorno ai 4 T), che consente di deviare apprezzabilmente i muoni che lo attraversano lo spettrometro.

3.3 Sistema di trigger

L'enorme frequenza di produzione di particelle durante le collisioni ($\sim 10^9$ eventi/s) rende praticamente impossibile conservare la totalità dei dati acquisiti da ATLAS, sia perché la frequenza di produzione di questi ultimi (attorno ai 20 MHz) supera di molto la velocità massima di scrittura dell'hardware a disposizione (che è di circa 500 Hz), sia perché lo spazio totale di storage è limitato (inoltre non tutti gli eventi sono fisicamente interessanti).

Interviene quindi un sistema di trigger che esegue una selezione preliminare degli eventi, rigettandone un numero sufficiente a portare la frequenza di dati in entrata al livello della frequenza di scrittura[3] (vedi fig. 7). La selezione è a tre livelli, chiamati LVL1, LVL2 e EF, di cui il primo è a livello hardware e gli altri due a livello software.

[3] Viene in questo modo scartato circa il 99.8% dei dati.

3.3.1 Trigger hardware (LVL1)

Il primo trigger si basa su informazioni ottenute a bassa granularità ($\Delta\eta \times \Delta\phi = 0.1 \times 0.1$ rad). La selezione consiste nell'identificare muoni, elettroni e fotoni ad alto p_T, utilizzando rispettivamente i dati dello spettrometro e del calorimetro elettromagnetico, ed inoltre cercare getti e leptoni τ decaduti in adroni con grande E_T.

Questo primo livello di selezione porta la frequenza di accettazione eventi a 75 kHz. Poiché si richiede che LVL1 identifichi univocamente i singoli *bunch* di protoni incidenti, la latenza di segnale di questo trigger è molto piccola (con un ritardo di circa 2 µs); durante questo periodo i dati sono immagazzinati in una memoria volatile (*pipeline*) integrata nei circuiti posti nei pressi del rivelatore.

3.3.2 Trigger software (LVL2 e EF)

Il secondo trigger, LVL2, analizza tutti e soli i dati provenienti dal primo livello LVL1. Questo livello di selezione, di tipo software, è progettato per ridurre ulteriormente la frequenza dei dati in entrata da 75 kHz a $\sim$ 5 kHz, utilizzando maggiori informazioni rispetto al trigger precedente. La latenza del segnale proveniente da LVL2 è dell'ordine di $1 \div 10$ ms, variabile da evento a evento.

L'ultimo trigger è EF, Event Filter. A questo livello vengono usati algoritmi di selezione utilizzati nell'analisi offline riadattati per l'ambiente online; in particolare, vengono applicate le calibrazioni aggiornate ed integrate le informazioni sull'allineamento delle parti del rivelatore e sulle correzioni ottenute dalle mappe del campo magnetico.

Questa selezione porta la frequenza di input a quella desiderata: circa 500 Hz, che corrisponde ad una frequenza di dati in output pari a $\sim 100\,\mathrm{MB\,s^{-1}}$.

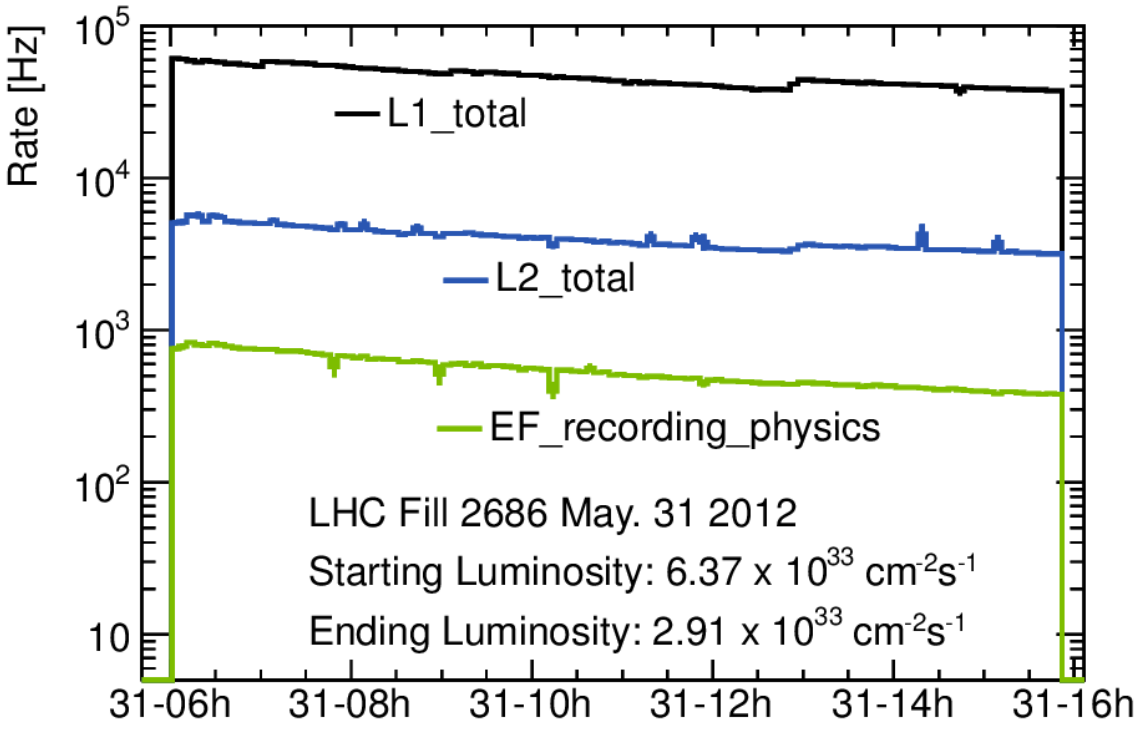


Figura 7: Frequenza di scrittura dei tre livelli di trigger durante un run di 10 ore.

Capitolo 4

Ricostruzione e identificazione di fotoni in ATLAS

4.1 Tipologie di eventi fotonici

Nella collisione *hard* di adroni la produzione di fotoni prompt è data da due tipi di processo: produzione diretta (costituita da scattering Compton e annichilazione quark-antiquark) e produzione per frammentazione (fig. 8).

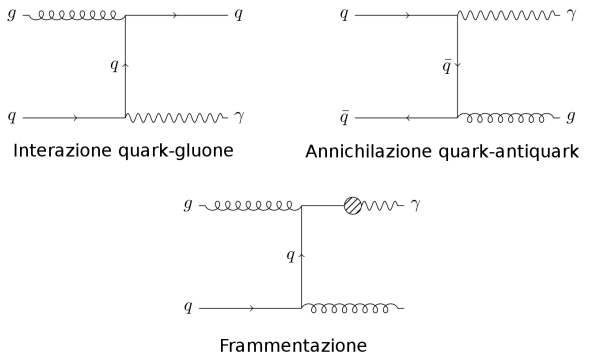


Figura 8: Diagrammi di Feynman per i processi che portano alla produzione di fotoni al *leading order*.

Le funzioni di frammentazione non sono calcolabili analiticamente tramite teorie perturbative, pertanto è necessario ricavarle tramite la comparazione dei dati. Per questo motivo, in generale si tenta di ridurre la componente di frammentazione al minimo mediante un taglio di isolamento.

4.2 Ricostruzione

In ATLAS i fotoni vengono ricostruiti e identificati a partire dai cluster di energia nel calorimetro elettromagnetico con $E_T > 2.5\,\mathrm{GeV}$, misurata in un insieme di 3×5 celle nel secondo comparto. Questi cluster vengono associati ad eventuali tracce rilevate nell'inner detector ed estrapolate al calorimetro: se non è presente alcuna traccia primaria in un intorno del cluster

EM il candidato è considerato un fotone non convertito, altrimenti è considerato inizialmente un elettrone.

Per includere nella ricostruzione anche i fotoni convertiti (quelli che interagiscono producendo coppie elettrone-positrone: vedi fig. 9) vengono considerati come candidati fotoni anche i cluster che sono collegabili a tracce provenienti da un vertice di conversione ricostruito nell'inner detector. Per aumentare l'efficienza, inoltre, vengono considerati anche cluster con un'unica traccia associata proveniente da un vertice secondario.

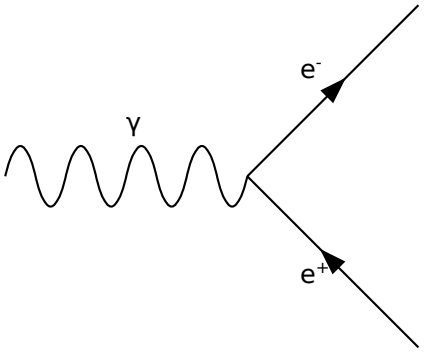


Figura 9: Produzione di coppie ee^- per conversione di un fotone.

Per entrambi i tipi di fotone, la misura dell'energia totale avviene sempre nel calorimetro elettromagnetico, utilizzando un cluster di dimensioni dipendenti dalla classificazione del fotone. Le dimensioni di questi cluster nella parte centrale del calorimetro vanno dalle $\eta \times \phi = 3 \times 5$ celle per i fotoni non convertiti alle 3×7 celle per quelli convertiti[1], mentre nei rivelatori laterali si utilizza un cluster 5×5.

L'efficienza di ricostruzione di fotoni, valutata tramite simulazioni Monte Carlo, è attorno all'85%.

4.3 Preselezione

In questo lavoro utilizziamo gli eventi provenienti dal Run I di ATLAS del 2012; tuttavia, non tutti gli eventi disponibili sono considerati validi per l'analisi: è necessario applicare una preselezione sugli eventi basata su una serie di tagli di qualità. In particolare:

- si accettano solamente gli eventi appartenenti alla e/γ *Good Run List* (**GRL**), una lista di eventi considerati buoni per l'analisi che esclude i *luminosity block*[2] nei quali i rivelatori più importanti (inner detector e calorimetro) hanno presentato malfunzionamenti;
- gli eventi devono passare almeno uno dei trigger di fotone singolo; questi trigger sono chiamati `EF_gXX_loose`, dove `XX` indica il valore minimo in GeV che si richiede in almeno un candidato fotone presente in quell'evento e va da 10 a 120 GeV;
- gli eventi devono superare un ulteriore controllo di qualità delle misurazioni, per escludere problemi in regioni particolari dei calorimetri.

[1] Per questi ultimi si usa una dimensione in ϕ superiore per compensare la deflessione degli $e^+ - e^-$ dovuta al campo magnetico.

[2] circa 2 minuti di presa dati

4.4 Identificazione

Una volta preselezionati gli eventi buoni, è necessario separare i candidati fotoni in segnale effettivo e background. Per fare ciò, si utilizzano alcune variabili di forma relative ai profili energetici longitudinali e trasversali degli sciami calorimetrici.

Si definiscono due gruppi di criteri di selezione: **loose** (letteralmente "largo", "sciolto") e **tight** ("stretto"), che dipendono dalla restrittività dei tagli applicati. Tutti i criteri dipendono dalla pseudorapidità ricostruita del candidato fotone in questione, per compensare le variazioni di spessore del materiale attraversato dall'oggetto in analisi.

4.4.1 Identificazione loose

Questo criterio di selezione di basa su tre variabili:

1 - la frazione di energia nel primo strato di calorimetro adronico (R_{had}), definita come il rapporto tra l'energia trasversa depositata nel primo layer del calorimetro adronico e quella del candidato fotone;

2 - il rapporto tra l'energia depositata in 3×7 celle e 7×7 celle nel secondo strato del calorimetro elettromagnetico (R_η);

3 - la media quadratica della distribuzione di energia lungo η nel secondo strato del calorimetro elettromagnetico (ω_2).

Ci si aspetta che i fotoni veri abbiano un basso rilascio di energia nel calorimetro adronico e che il loro profilo energetico sia molto più concentrato nel centro del proprio cluster rispetto agli eventuali fotoni di background provenienti dai jet. Il criterio loose è lo stesso sia per i fotoni convertiti che per quelli non convertiti.

4.4.2 Identificazione tight

Questo criterio applica una selezione più rigida, che prende in considerazione anche la variabile R_ϕ, definita come il rapporto tra l'energia depositata in 3×3 e 3×7 celle nel secondo strato del calorimetro elettromagnetico, e la forma degli sciami nel primo strato. In funzione del deposito di energia nel primo strato vengono definite diverse quantità utili per distinguere lo sciame generato da un singolo fotone da una coppia di sciami contigui sovrapposti[3], sfruttando l'elevata granularità delle strip nel primo comparto.

Le variabili definite sono le seguenti:

- la larghezza[4] ω_{stot} della distribuzione dell'energia lungo η valutata su tutte le strip del cluster;
- l'asimmetria E_{ratio} valutata tra il primo ed il secondo massimo del profilo energetico nel primo strato (in assenza del secondo massimo, $E_{ratio} = 1$);
- la differenza ΔE tra l'energia del secondo massimo ed il minimo compreso tra i due massimi (in assenza del secondo massimo, $\Delta E = 0$;

[3] questo può verificarsi ad esempio dal decadimento di un mesone neutro in una coppia di fotoni.
[4] intesa come media quadratica (RMS) della distribuzione.

– la frazione di energia F_{side} nelle 7 strip centrate attorno al primo massimo che non è contenuta nelle 3 strip più interne;

– la larghezza ω_{s3} della distribuzione dell'energia valutata nelle 3 strip più interne centrate attorno al primo massimo.

La prima variabile serve a rigettare i candidati il cui sciame è largo e pertanto compatibile con quello di un jet; la seconda e la terza variabile rigettano i candidati che producono sciami con massimi distinti nel primo strato del calorimetro; la quarta e la quinta variabile rigettano gli eventi in cui due sciami sovrapposti producono un unico picco allargato.

A differenza del criterio loose, le variabili tight sono ottimizzate separatamente per fotoni convertiti e non, e vengono definite a partire da simulazioni Monte Carlo, il che consente di ottenere un'efficienza di selezione media pari a circa l'85% dei fotoni ricostruiti.

4.5 Energia trasversale di isolamento

Una delle variabili più importanti per lo studio dei fotoni diretti è l'*energia trasversale di isolamento*, definita come l'energia depositata nel calorimetro entro un cono attorno al candidato fotone meno l'energia del fotone stesso. Il cono in questione, centrato sull'oggetto in analisi, ha un raggio pari a:

$$R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2} = 0.4\,\text{rad} \tag{4.1}$$

nel piano $\eta - \phi$. Da questo cono viene esclusa una zona di 5×7 celle centrali, e si effettuano delle correzioni per l'eventuale energia residua del fotone fuoriuscente da questa zona, calcolata sempre tramite simulazioni. Si sottraggono infine ulteriori contributi all'energia di isolamento dovuti ad *underlying event*, con una tecnica descritta in [11].

L'utilità principale di questa variabile è quella di rigettare jet adronici di alta energia circondati di molta attività adronica e ridurre la componente di frammentazione. In questo lavoro le distribuzioni di segnale e background vengono normalizzate applicando un taglio sulle code oltre una certa energia di isolamento e utilizzate per stimare la purezza del campione.

Capitolo 5

Il framework di analisi

In questo capitolo verrà descritto il framework di analisi dati utilizzato in questa tesi e le motivazioni che mi hanno portato a realizzarlo, nonché le differenze tra l'approccio che ho usato per la mia analisi e quello comunemente adottato in questo campo.

Come verrà descritto in maggiore dettaglio nelle prossime sezioni, le caratteristiche principali del mio framework sono due:

- i dati da analizzare vengono recuperati via rete dalla griglia di calcolo ma l'algoritmo di analisi vero e proprio è lanciato in locale;
- l'analisi procede in parallelo, lanciando una coda di job per ogni sito presente in un file di configurazione (liberamente modificabile ed aggiornabile), consentendo di processare tanti più dati contemporaneamente quanti più siti contenenti i dataset interessanti per l'analisi si conoscono.

5.1 Nomenclatura utilizzata

Prima di analizzare dettagliatamente il framework d'analisi è bene fornire i termini principali che si incontrano lavorando su DDM (*Distributed Data Management*).

WLCG (*Worldwide LHC Computing Grid*): è l'infrastruttura di calcolo distribuito che consente a numerosi centri di ricerca e calcolo in tutto il mondo l'accesso ai dati provenienti da LHC e l'elaborazione degli stessi. Il tutto avviene tramite un sistema gerarchico di *Tier* che si occupa di pre-analizzare e smistare i *raw data* secondo un ordine stabilito in modo da rendere l'analisi dati più organizzata ed efficiente.

sito è l'insieme dei servizi resi disponibili da uno specifico centro di calcolo, che va dalla capacità di storage presso uno o più dischi rigidi locali ai protocolli di rete necessari ad accedere ai file che si trovano salvati in tali storage. I protocolli principalmente utilizzati sono SRM e XRootD.

dataset è una collezione di file ROOT logicamente correlati, ad esempio appartenenti allo stesso run, denominati in modo da fornire numerose informazioni sul proprio contenuto: ad esempio, un dataset chiamato:
data12_8TeV.00204153.physics_Egamma.merge.NTUP_PHOTON.r4065_p1278_p1344_p1345
contiene eventi (in questo caso *n*tuple di fotoni) raccolti nel 2012 con un'energia di collisione di 8 TeV (le tag finali danno informazioni su run number, stream e storia del

dataset). Un'importante distinzione tra dataset è quella tra i dataset *data-* e i dataset *mc-*: i primi contengono dati reali provenienti da LHC, gli ultimi sono invece simulazioni MonteCarlo.

dataset container è una collezione di dataset logicamente correlati. Seguono la stessa nomenclatura dei dataset singoli, ma il loro nome termina in / per distinguerli da questi ultimi.

file è un file fisico (conservato tipicamente in molteplice copia su diversi storage indipendenti), solitamente di estensione *.root*, che contiene una grande quantità di dati[1] in formato compresso, che possono essere analizzati tramite il software messo a disposizione dal CERN. Le dimensioni di uno di questi file può risultare di vari gigabyte.

5.2 Il sistema di Tier

Come accennato in 5.1, la struttura di WLCG è organizzata secondo un sistema gerarchico di *Tier*, numerate in ordine crescente a partire da quella fondamentale (Tier-0) fino ad arrivare a Tier-3, che costituisce il livello gerarchico più basso (vedi fig. 10)

- **Tier-0**, che si trova al CERN, è l'*hub* centrale di WLCG, e si occupa di raccogliere tutti i *raw data* provenienti da LHC, processarli in modo da renderli più facilmente utilizzabili in seguito e distribuirli ai centri di calcolo che costituiscono Tier-1. In totale, Tier-0 fornisce meno del 20% della potenza di calcolo totale di WLCG . [9]

- **Tier-1** è costituita da 13 centri di calcolo sparsi in tutto il mondo dotati di sufficente capacità di storage per contenere i dati di LHC. Oltre a conservare i dati grezzi e i risultati delle simulazioni Monte Carlo, Tier-1 si occupa anche di garantirne l'accesso e a contribuire in larga misura alla potenza di calcolo necessaria a riprocessare i dati e a riorganizzarli in dataset, distribuendoli successivamente a Tier-2. Per massimizzare l'affidabilità e la velocità di trasmissione, Tier-1 è connesso al CERN tramite cavi a fibra ottica da 10 gigabit al secondo.

- **Tier-2** aggiunge ulteriore potenza di calcolo per le rielaborazioni finali dei dati e per le simulazioni Monte Carlo. I dati utilizzati vengono raccolti principalmente da Tier-1, così come a Tier-1 vengono inviati i risultati delle simulazione in modo da conservarli permanentemente negli storage dedicati.

- **Tier-3** è formata da numerosi centri di calcolo che, pur avendo accesso a WLCG, non contribuiscono direttamente al trattamento e alla conservazione dei dati. Le macchine di Tier-3 danno potenza di calcolo ed accesso ai dataset a gran parte degli utenti finali, principalmente ricercatori che compiono analisi dati sui vari esperimenti di LHC. Il centro di calcolo dell'INFN presso il dipartimento di Fisica dell'Università degli Studi di Milano somma le funzioni di Tier-2 e Tier-3.

o

[1] quasi sempre conservati sotto forma di ROOT TTree: un formato ad albero i cui rami possono contenere valori scalari od oggetti strutturati.

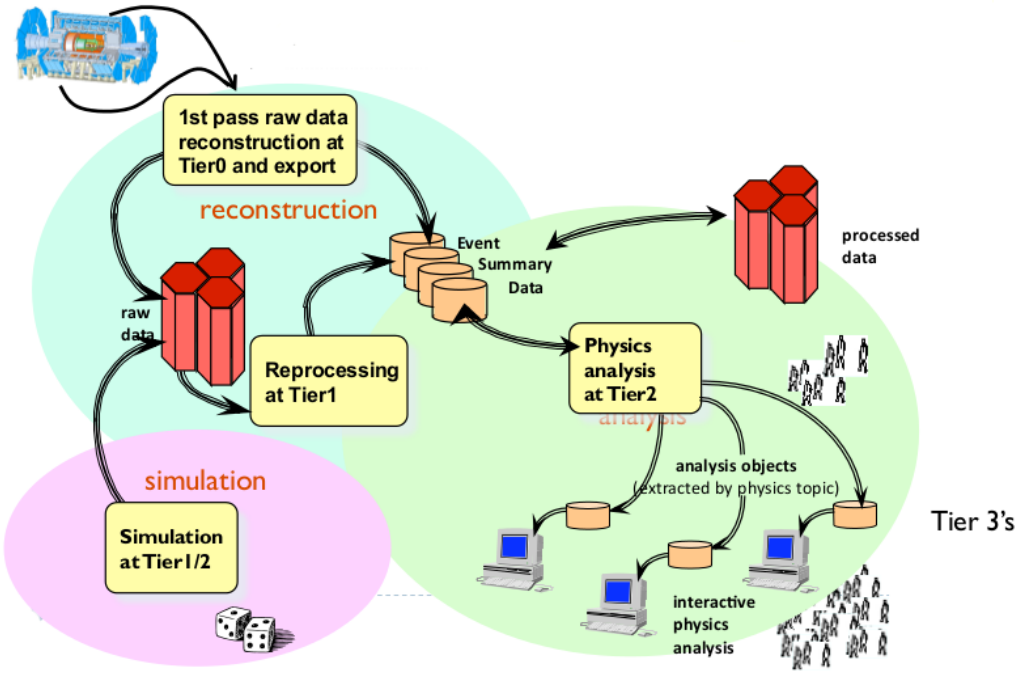


Figura 10: Schema del sistema di Tier di WLCG, con le relative gerarchie e compiti all'interno della rete di calcolo.

Paese	Nome Federazione	Nome Sito
Canada	CA-TRIUMF	TRIUMF-LCG2
Francia	FR-CCIN2P3	IN2P3-CC
Germania	DE-KIT	FZK-LCG2
Italia	IT-INFN-CNAF	INFN-T1
Paesi Bassi	NL-T1	NIKHEF-ELPROD
Paesi Bassi	NL-T1	SARA-MATRIX
Paesi nordici	NDGF	NDGF-T1
Corea del nord	KR-KISTI-GSDC	KR-KISTI-GSDC-01
Spagna	ES-PIC	pic
Taiwan	TW-ASGC	Taiwan-LCG2
Gran Bretagna	UK-T1-RAL	RAL-LCG2
Stati Uniti	US-FNAL-CMS	USCMS-FNAL-WC1
Stati Uniti	US-T1-BNL	BNL-ATLAS

Tabella 5.1: Lista dei centri di calcolo Tier-1

5.3 Analisi dati su WLCG

Naturalmente, una volta ottenuti ed organizzati i dati provenienti dai vari esperimenti di LHC, ciò che resta da fare è analizzarli per ricavarne dei risultati di interesse fisico.

L'analisi dati vera e propria, effettuata da singoli o gruppi di ricercatori affiliati al CERN, avviene principalmente tramite l'utilizzo di ROOT. Si tratta di un software, sviluppato al

CERN, studiato specificatamente per consentire agli utenti la creazione di algoritmi di analisi anche molto complessi, interfacciandosi principalmente con il linguaggio di programmazione C++, ma con vari moduli aggiuntivi che ne consentono l'utilizzo anche in altri linguaggi come Python o Ruby.

In particolare, esiste una suite di pacchetti ROOT dedicata nello specifico all'analisi dati su ATLAS, che ho utilizzato estensivamente nel corso di questo lavoro.

5.3.1 Reperimento dei dati: analisi locale e analisi remota

Una volta scritto il codice per l'analisi dati (in questo caso C++), è necessario fornire al programma risultante i dati che si desidera elaborare[2]. Naturalmente nella maggior parte dei casi i dati desiderati non si trovano inizialmente sulla macchina locale che si ha a disposizione, ma saranno piuttosto salvati in uno o più storage di siti remoti.

Esistono quindi due possibili approcci:

1) trasferire i dati necessari sulla propria macchina e lanciare quindi l'analisi in locale; questo approccio tuttavia è applicabile solamente se la quantità di dati in questione è ragionevolmente piccola, per evitare tempi di attesa e traffici di rete eccessivi. Nel mio caso, non erano disponibili 70 TB di storage necessari a trasferire i dati del Run a 8 TeV del 2012. Inoltre, il trasferimento dei dataset avrebbe richiesto un tempo eccessivo[3];

2) lanciare la propria analisi in remoto, tramite un *submit* del proprio job al sito contenente i dati tramite un apposito protocollo remoto. Questo è l'approccio standard seguito dalla maggior parte degli utenti per l'analisi dati, poiché evita i problemi relativi all'approccio locale e sposta il carico di lavoro computazionale sulle macchine del sito remoto. Tuttavia anche questo metodo non è privo di svantaggi:

   - il job inviato ha una latenza incognita, poiché viene inserito in un sistema di coda ed eseguito quando una macchina è disponibile;
   - è praticamente impossibile eseguire debug o profiling sul proprio job, dal momento che l'intera analisi è nascosta ed inaccessibile a chi l'ha lanciata. In particolare, in caso di crash, l'unico modo per scoprirne la causa è contattare l'amministratore del sito su cui l'analisi era stata lanciata.

Vista l'ingente mole di dati da processare per questo lavoro, entrambi gli approcci risultano poco soddisfacenti: l'analisi locale è impraticabile per via dell'impossibilità pratica di trasferire tutti i dati necessari al centro di calcolo di Milano, mentre l'analisi remota ha il difetto di non essere direttamente controllabile e quindi implica lanciare l'algoritmo di analisi "alla cieca" senza avere indizi nel caso questo fallisca.

Ho pertanto valutato un approccio ibrido, che consiste nel:

i) *leggere* i dati via rete;

ii) salvare in locale *solo le variabili necessarie* per l'analisi;

iii) *eseguire* quindi l'analisi in locale sui dati salvati.

[2] questi possono essere sia dati reali che dati provenienti da simulazioni Monte Carlo: il formato in cui vengono salvati, come visto in 5.1, è il medesimo.

[3] in 7.1 è presentata una stima dei tempi coinvolti

Questo metodo elimina il problema di trasferire tutti i dati in locale[4] e consente inoltre di replicare l'analisi molto velocemente senza dover reperire di nuovo i dati via rete. È addirittura possibile riutilizzare gli stessi dati per un'analisi diversa che necessiti delle stesse variabili o di un loro sottoinsieme.

5.4 Caratterizzazione preliminare dell'analisi

Prima dell'implementazione vera e propria del metodo descritto sopra, si è reso necessario uno studio preliminare su alcune caratteristiche computazionali dell'analisi; l'incognita più importante era la quantità di operazioni Input/Output (I/O) effettuate realmente dal programma: bisogna infatti tenere conto che tutte le operazioni che in locale consistono nella lettura di file da disco rigido si traducono in operazioni di lettura via rete nell'approccio ibrido sopra delineato: è importante che queste operazioni siano una quantità contenuta per evitare congestioni di rete sia sulla macchina locale che su quelle remote.

Lo studio dell'I/O effettivo è stato condotto applicando l'algoritmo di analisi su un singolo file di dati precedentemente trasferito in locale (delle dimensioni di 4.3 GB), tenendo traccia di tutte le operazioni di sistema effettuate dall'analisi tramite l'utility *strace*. L'analisi di tali operazioni ha rivelato che solamente la parte di file contenente le variabili utili all'analisi viene effettivamente letta da ROOT, che si rivela quindi molto efficiente nel leggere i dati salvati nel suo apposito formato (vedi fig. 11). Poiché questi risultati si sono rivelati promettenti per il metodo ibrido descritto in 5.3.1, ho proceduto con lo sviluppo del framework che lo implementa.

5.5 Accesso ai dati remoti con DQ2 e XRootD

Esiste una vasta gamma di strumenti a disposizione degli utenti di WLCG per il reperimento dei dati remoti: i due principali utilizzati nel corso di questa tesi sono stati DQ2 e XRD.

5.5.1 DQ2

La suite di client DQ2 è utilizzata per la maggior parte delle operazioni relative al Distributed Data Management di ATLAS, in particolare trasferimento e catalogazione dei dati. Delle molte funzionalità offerte da DQ2, quelle utilizzate dal mio framework sono: elencare i siti che possiedono un certo dataset, i file in esso contenuti, il numero di repliche dello stesso sull'intera WLCG ed i siti disponibili per l'accesso remoto. Tutte queste operazioni possono essere convenientemente automatizzate sotto forma di script: il framework sfrutta questo fatto per localizzare autonomamente gli URL dei dataset (e relativi file) prendendo in ingresso un file contenente i nomi di tutti i dataset su cui si desidera svolgere l'analisi.

Esiste però una difficoltà inerente a questo procedimento: il protocollo di accesso ai siti da parte di DQ2 è SRM, quindi gli URL da esso restituiti sono relativi a questo protocollo; poiché SRM non permette singole operazioni di I/O sui file, cosa indispensabile affinché l'accesso remoto con ROOT sia efficiente, è necessario tradurre il path fornito da DQ2 sostituendo

[4] tipicamente le variabili necessarie per l'analisi sono solo una piccola frazione del totale disponibile per ogni evento; nel mio caso si trattava di poche decine di variabili utili contro più di 8000 disponibili, corrispondenti a $\sim 0.5\%$ del totale.

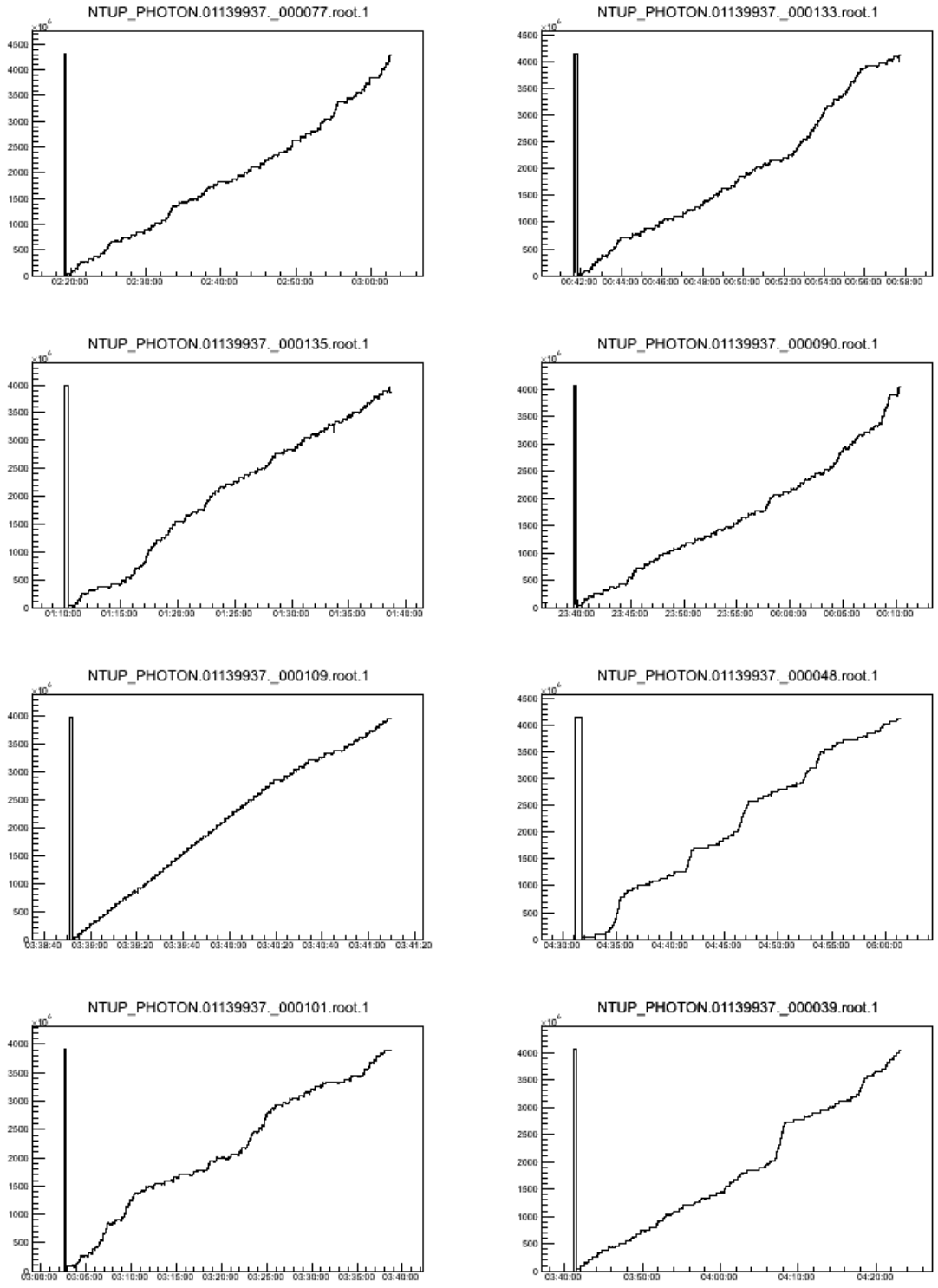


Figura 11: I grafici mostrano l'andamento del *seek pointer* in lettura di un file rispetto al tempo. Il seek pointer punta al contenuto del file che verrà letto alla successiva richiesta di input, quindi i grafici forniscono una panoramica su come i file vengono letti. Si può vedere che ogni file viene letto una volta sola in modo lineare: traducendo queste operazioni in letture via rete, significa che non vengono mai trasferiti più volte gli stessi dati.

all'indirizzo SRM il relativo URL XRootD (`root://...`), accessibile all'analisi. Questa sostituzione è tuttavia non banale, pertanto è possibile effettuarla solo per quei siti per i quali la sostituzione è nota a priori.

Più nel dettaglio, ogni sito su WLCG ha un identificativo: ad esempio, il centro di calcolo di Lione - uno di quelli usati per l'analisi durante questa tesi - è noto come IN2P3-CC_PHYS-HIGGS; un URL SRM restituito da DQ2 per un dataset presente in questo sito avrà la forma: `srm://ccsrm.in2p3.fr`, mentre il corrispettivo XRootD sarebbe `root://ccxrootdatlas.in2p3.fr`. Come si nota, non esiste una convenzione di nomi per questi URL, il che rende impossibile automatizzare del tutto questo processo di sostituzione. Per la mia analisi, ho identificato tre diversi siti contenenti tutti i dataset necessari per i quali la sostituzione era nota, più altri due siti contenenti un sottoinsieme dei dataset necessari, per un totale di cinque siti diversi. Come vedremo in seguito, più siti sono disponibili, più l'analisi condotta via framework può essere parallelizzata.

5.5.2 XRootD

XRootD è un protocollo generico che consente un accesso scalabile, veloce e a bassa latenza ai dati ROOT. Inizialmente sviluppato come strumento autonomo[5], è poi stato integrato completamente in ROOT, ed è utilizzato da quest'ultimo per ogni genere di accesso remoto a file. XRootD comprende anche una funzionalità di redirezione che permette di reperire un file con maggiore efficacia effettuando una query ad un servizio detto *redirector*, che è reso disponibile da molti siti di WLCG. È possibile infatti chiamare XRD fornendogli il path di un determinato file per farsi restituire l'indirizzo IP di un sito qualunque su cui quel file è presente. Il sito corrispondente all'IP restituito può anche non essere presente nei database interrogati da DQ2, perché i due servizi sono indipendenti.

L'utilizzo di XRootD rappresenta un metodo alternativo di reperimento dei dati rispetto a DQ2, con una serie di pro e contro:

Pro
- fornisce un canale d'accesso ai dati alternativo a DQ2, quindi può essere utilizzato assieme a quest ultimo per aumentare la parallelizzazione del lavoro;
- utilizza nativamente il protocollo ROOT, pertanto è esente dal problema della sostituzione degli URL che limita l'utilizzo di DQ2 ai soli siti noti.

Contro
- Opera con una granularità diversa rispetto a DQ2 (singoli file invece che dataset), pertanto a livello di framework deve essere gestito separatamente;
- il meccanismo di redirezione è molto più lento rispetto a DQ2, che interroga semplicemente un database; il vantaggio del redirector sta nella maggiore affidabilità, visto che il database interrogato da DQ2 potrebbe non essere aggiornato; tuttavia gli elevati tempi d'esecuzione di XRD lo rendono quasi inutilizzabile come unico metodo di reperimento dati;
- in rari casi, la ricerca di un file da parte di XRD non restituisce alcun risultato;
- il servizio XRootD di ATLAS è ancora in fase sperimentale: per prima cosa non tutti i siti lo forniscono, e in più possono capitare periodi di *down* imprevedibili (com'è accaduto nel corso della mia analisi).

Il framework utilizza sia DQ2 che XRD in parallelo, in modo da massimizzare la velocità e l'affidabilità di reperimento dei dati.

[5] A SLAC, nell'ambito del progetto Scalla

5.6 Struttura del framework

Il framework d'analisi è strutturato in 3 parti principali (vedi fig. 12):

1) il **launcher**, che si occupa di recuperare i dati tramite DQ2 e XRootD e di lanciare e coordinare i sottoprocessi d'analisi;

2) la **parte di analisi** vera e propria, costituita da script che processano i dati tramite gli algoritmi in ROOT, producono i risultati ed eseguono dei self-check preliminari per verificare che l'analisi non sia corrotta;

3) gli **script di controllo** che, conclusa l'analisi, eseguono degli ulteriori controlli sui log prodotti da launcher e script d'analisi in modo da garantire con maggiore sicurezza la buona riuscita dell'analisi e da generare statistiche sulla performance della stessa.

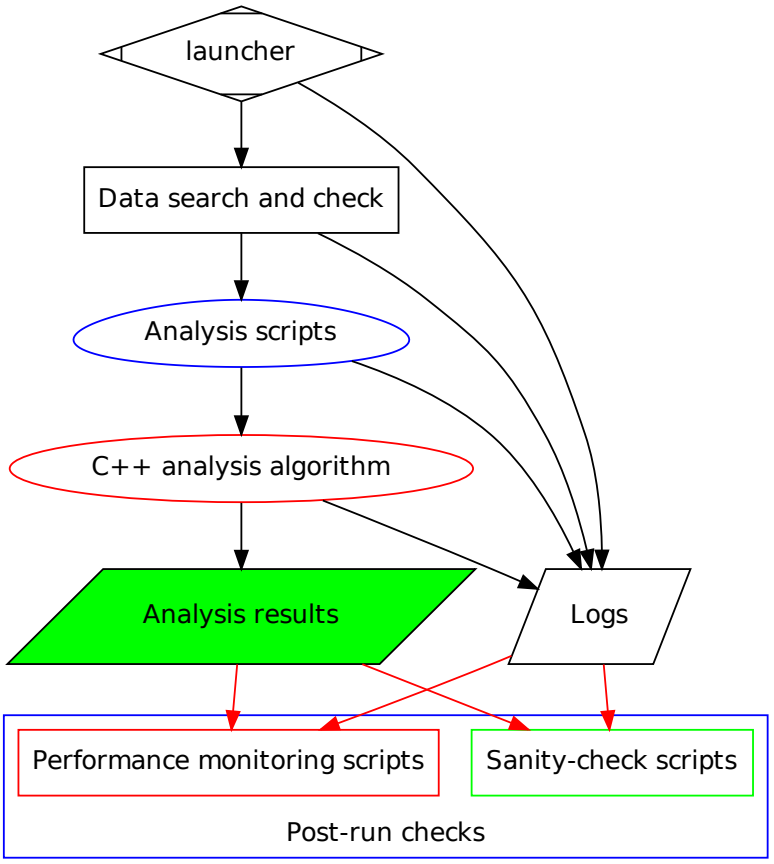


Figura 12: Struttura logica del framework d'analisi

5.6.1 Il launcher e gli script di reperimento dati

Il launcher è la parte fondamentale del framework, quella che si occupa di creare le working directory, leggere i file di configurazione e lanciare i vari sotto-processi che eseguono l'analisi.

Si tratta di un bash script che viene lanciato sulla macchina su cui si intende eseguire l'analisi, fornendogli in input due file di configurazione: uno contenente i nomi di tutti i dataset che devono essere analizzati, e l'altro contenente le informazioni riguardanti i siti da cui si vogliono attingere i dataset, in particolare la sostituzione da path SRM a path ROOT. Una riga di questo file avrà la struttura:

```
Lione  IN2P3-CC_PHYS-HIGGS  srm://ccsrm.in2p3.fr  root://ccxrootdatlas.in2p3.fr/
```

che associa al nome del sito il suo identificativo WLCG, il prefisso SRM e quello ROOT dei file presenti su quel sito.

La prima operazione effettuata dal launcher è effettuare una serie di query DQ2 per identificare tutti i siti che contengono repliche dei dataset da analizzare; al termine di questa operazione viene creata una tabella contenente, dataset per dataset, tutti i siti su cui quel dataset è disponibile.

Il launcher procede dunque a creare le directory necessarie per l'analisi. La logica con cui l'analisi viene effettuata è la seguente: per ogni dataset da analizzare si inserisce un token in un insieme *pool*; i sottoprocessi d'analisi (che operano in parallelo) vanno ciascuno a prelevare un token alla volta dalla pool e, dopo aver spostato quel token in un insieme *running*, effettuano l'analisi sul corrispettivo dataset. Completata l'analisi, e verificatone l'esito positivo, il token viene spostato nell'insieme *complete*, altrimenti viene rimesso nella pool[6]. (vedi fig. 13).

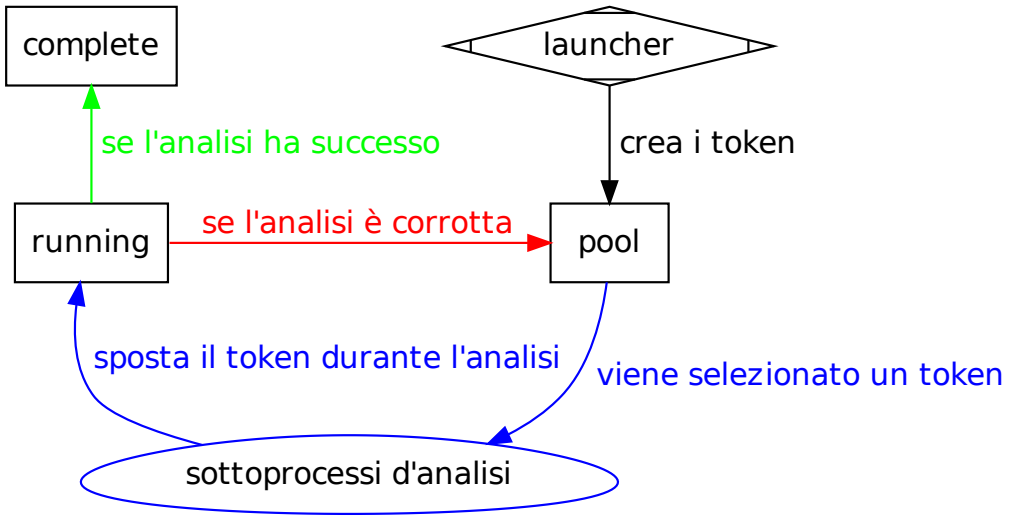


Figura 13: Workflow del framework d'analisi

[6]Nell'implementazione concreta di questa struttura logica, gli insiemi pool, running e complete sono directory e i token sono file; si sfrutta quindi l'atomicità dell'operazione `move` offerta da filesystem per evitare problemi di concorrenza tra i processi.

Prima di far partire la vera analisi, viene effettuato un controllo preliminare sulla tabella creata in precedenza per verificare che ogni dataset abbia almeno un sito associato: in caso contrario, l'analisi non può essere portata a termine con certezza per la totalità dei dataset richiesti, quindi in questa eventualità il processo si blocca e viene restituito un errore[7].

Se il controllo sui dataset va a buon fine, il launcher legge il file con le informazioni sui siti, salva in memoria i siti validi e fa partire il ciclo di analisi vero e proprio. Il ciclo avviene come mostrato in fig. 14. Il numero di iterazioni eseguite dipende da quante analisi falliscono

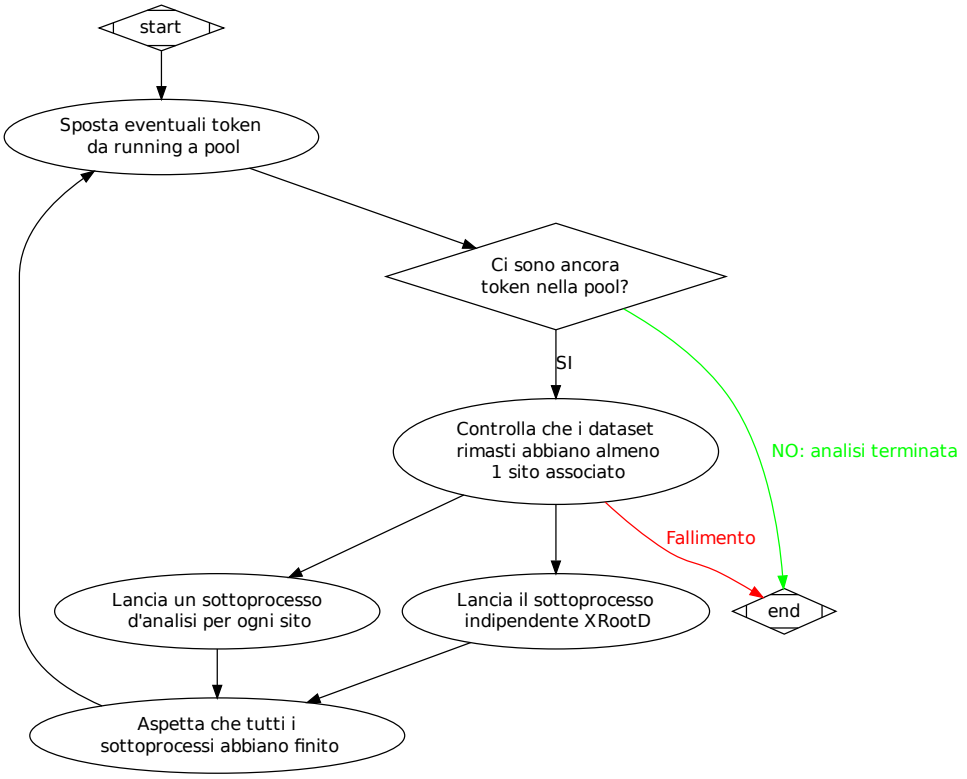


Figura 14: Ciclo principale dell'analisi

all'interno di un ciclo; nel caso ideale, quello in cui tutte le analisi fossero completate con successo al primo tentativo, avverrebbe una sola iterazione. Nella pratica, avviene quasi sempre che almeno un'analisi fallisca e il framework non riesca a recuperare l'errore all'interno dello stesso ciclo; per l'analisi completa effettuata in questo lavoro, sono state necessarie 8 iterazioni per analizzare correttamente tutti i 263 dataset del run 2012.

5.6.2 Gli script di analisi

Gli script di analisi sono tre:

[7] questo controllo viene in realtà effettuato da uno script ausiliario in awk. Questo linguaggio di scripting è stato ampiamente utilizzato per tutti gli script di parsing, rivelandosi particolarmente comodo per questo genere di task.

1) **subproc.py** è uno script Python che recupera i path dei file da analizzare tramite DQ2, facendo uso delle informazioni contenute nel file dei siti passato al launcher. Poiché questo script opera su un singolo sito, il launcher crea tante istanze di questo script quanti sono i siti validi che gli vengono passati, in modo che l'analisi prosegua parallelamente su tutti i siti.

   Ognuno di questi script legge la tabella dei dataset creata precedentemente dal launcher e costruisce una lista dei dataset che è in grado di recuperare dal proprio sito, ordinandoli in modo che i dataset di cui sono presenti meno repliche vengano trattati per primi. È questo stesso script ad occuparsi di spostare i token tra i vari insiemi, recuperare gli URL dei file da analizzare, lanciare l'algoritmo ROOT, analizzare i log prodotti e aggiornare la tabella dei dataset una volta completata l'analisi.

2) **xrootd_subprocess.sh** è un bash script che si occupa di gestire il sottoprocesso d'analisi che lavora via XRootD, senza però condurre direttamente l'analisi vera e propria. Questo script intermedio è necessario a causa della diversa granularità coinvolta nella ricerca via XRootD rispetto a quella via DQ2 (vedi 5.5.2): la sua utilità consiste infatti nel raggruppare le analisi effettuate su singoli file via XRootD in modo che queste appaiano agli occhi del launcher come un'unica analisi effettuata su un dataset.

   Come *subproc.py*, questo script preleva un dataset alla volta dalla pool, ottiene la lista dei file in esso contenuti via DQ2 e, invece di operare la sostituzione specifica per un sito su tali file, esegue una query XRD per il primo di essi, trova l'IP di un sito che contiene quel file e lancia l'analisi file per file, raggruppando risultati e log in modo da restituire la stessa granularità di *subproc.py*.

3) **run_xrootd.py** è lo script che esegue l'analisi sui singoli file risolti da *xrootd_subprocess.sh*. Come *subproc.py*, anche questo script effettua dei controlli di qualità sull'analisi effettuata, e dà errore appena l'analisi su un file fallisce. Per mantenere la granularità del dataset sul processo di analisi, il fallimento su un singolo file comporta che l'analisi su tutto il dataset corrente sia considerata fallita dal launcher.

5.6.3 Gli script di controllo

Una volta completata l'intera analisi, una serie di script ausiliari viene utilizzata "manualmente" per alcune rifiniture finali. Come abbiamo visto nel paragrafo precedente, il controllo degli errori sulle singole analisi viene effettuato automaticamente in runtime, quindi ciò che resta da fare è principalmente raccogliere alcune statistiche overall sull'intero processo avvenuto. Poiché i log sono pensati in modo da mantenere praticamente tutte le informazioni interessanti dell'analisi ed essere al tempo stesso *machine-readable*, risulta abbastanza semplice scrivere degli script che ne eseguano il parsing e restituiscano i risultati desiderati.

Successo/fallimento delle analisi

Un dato di interesse riguardo al processo di analisi condotto via framework è il numero di analisi fallite rispetto al numero totale di analisi tentate. Un'analisi può fallire per numerosi motivi diversi: file non disponibile sul sito remoto, perdita di connessione con quest'ultimo, crash dell'algoritmo di analisi, scadenza del proxy necessario per effettuare query, ...

Dal parsing della totalità dei log prodotti dall'analisi[8] sono emersi i seguenti risultati:

Analisi totali	**Analisi riuscite**	**Analisi fallite**
3903	2950 (75.58%)	947 (24.42%)

Come si vede, il numero di analisi fallite è circa un quarto del totale. Bisogna considerare che i controlli che fanno considerare "fallita" un'analisi sono stati mantenuti molto severi per essere certi di ottenere risultati affidabili anche processando una mole molto grande di dati (263 dataset per un totale di più di 10 000 file).

I controlli effettuati sui log sono i seguenti:

- per ogni analisi su un dataset, si controlla che in tutti i file di log prodotti siano presenti le indicazioni che l'analisi è cominciata e terminata con successo, e che le due corrispondano. In questo modo ci si assicura che non ci siano stati crash durante l'analisi;
- si effettua un parsing riga per riga degli stessi file di log controllando che non siano presenti errori dati dall'algoritmo di analisi o dagli script stessi[9];
- per ogni dataset, si confronta il numero di eventi in entrata (contati da ROOT prima dell'analisi) con quello degli eventi totali processati: se i due numeri non corrispondono l'analisi si considera corrotta;
- appena un'analisi è completata, viene lanciato automaticamente un algoritmo in C++ che legge il file di output prodotto dalla stessa e si assicura che il TTree in esso contenuto sia identico ad un TTree prototipo che si è certi contenere tutte e sole le variabili necessarie.

Velocità di elaborazione

Un altro aspetto interessante da studiare consiste nei tempi di elaborazione medi impiegati dall'analisi a seconda di vari parametri, quali l'ora del giorno, il sito da cui si leggono i dati o il dataset che si processa.

A questo scopo interviene una serie di altri script che, sempre tramite parsing dei log, consentono di estrarre in modo automatico tutti questi dati. I dati sui tempi sono riportati nelle figg. 15,16,17.

[8] il numero di log risultanti dalla mia analisi ammonta a circa 1050 file, lunghi in media 435 righe.

[9] O almeno errori inaccettabili: durante l'analisi vengono sempre restituiti alcuni messaggi di warning o di informazione, ma molti di questi possono essere ignorati in quanto compromettono l'esito dell'analisi.

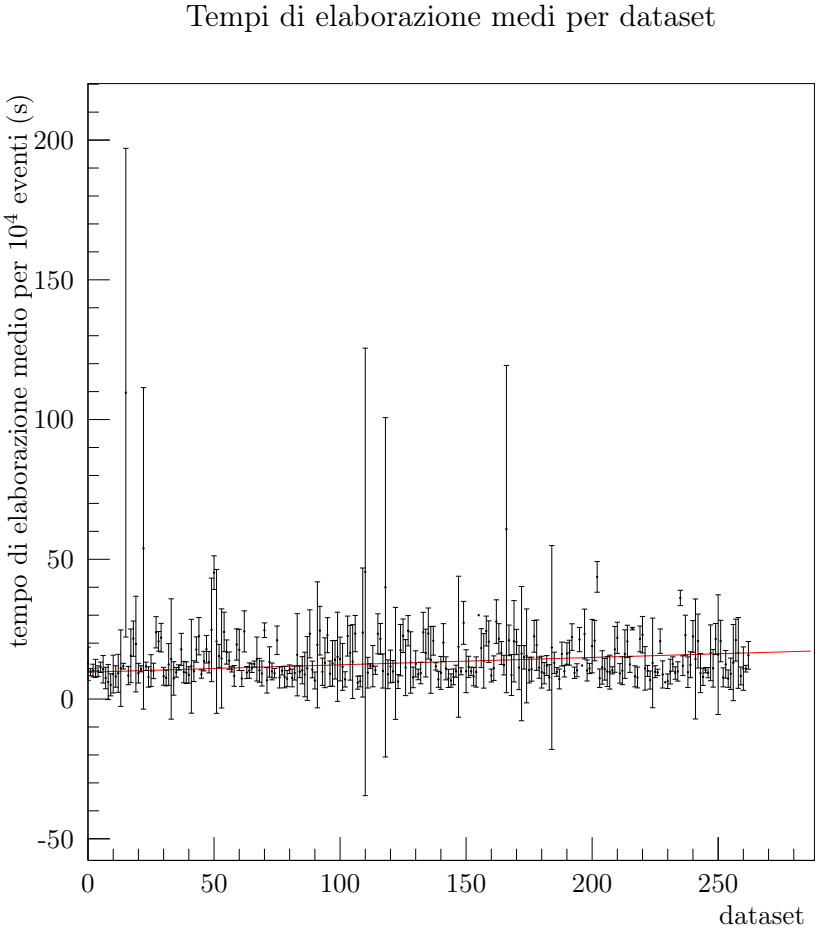


Figura 15: Tempi di elaborazione per dataset. In ascissa, i dataset sono numerati in modo arbitrario; in ordinata, il numero di secondi necessari per processare 10^4 eventi. La media complessiva è $t \sim 10$ s/10^4 eventi.

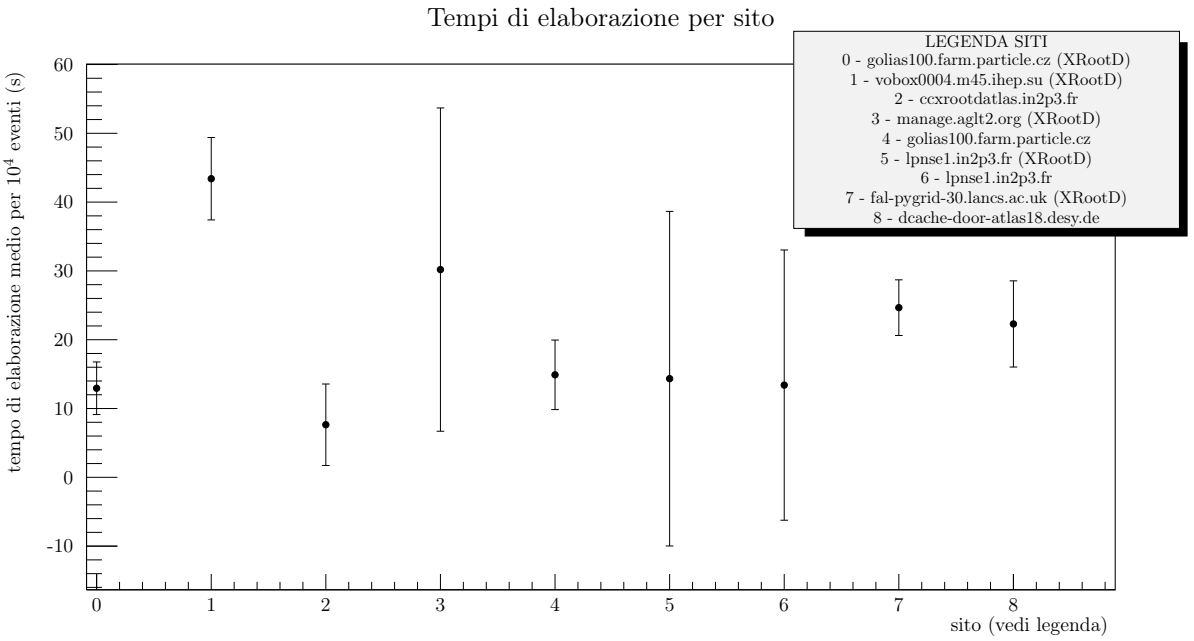


Figura 16: Tempi di elaborazione per sito. Si può apprezzare la differenza di velocità di elaborazione (che si traduce in maggiore o minore latenza) tra i vari siti utilizzati.

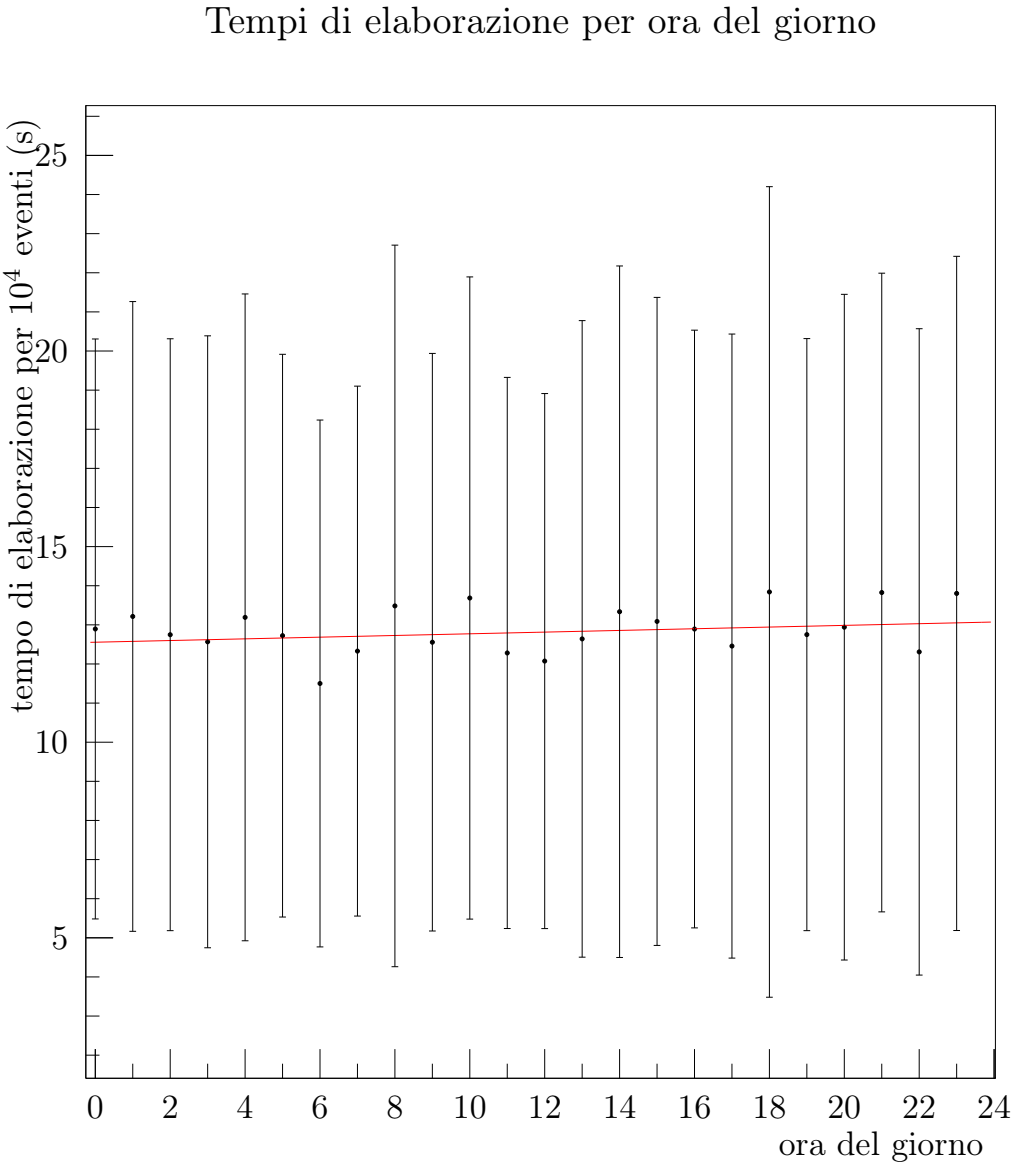


Figura 17: Tempi di elaborazione per ora del giorno. Appare evidente che la velocità di elaborazione non dipende dall'orario durante il quale viene eseguita l'analisi.

Capitolo 6

Calcolo della purezza e isolation template fit

Come accennato nell'Introduzione (cap. 1), l'analisi fisica vera e propria svolta in questa tesi consiste nel calcolo della purezza del campione di fotone inclusivo prodotto nel 2012 dal run a 8 TeV di ATLAS. Prima di presentare i risultati, verrà esposto più nel dettaglio in cosa consiste il metodo utilizzato: l'*isolation template fit*.

6.1 Definizione di purezza

Diamo innanzitutto la definizione di **purezza** di un campione.

Dato un campione di N eventi, definiamo *segnale* la porzione di eventi che è composta effettivamente da veri fotoni[1] (e la indicheremo con N_s), e *background* tutti gli eventi restanti, che chiamiamo N_b. In questo modo si ha naturalmente:

$$N = N_s + N_b \tag{6.1}$$

Date queste definizioni, definiamo la purezza come:

$$P = \frac{N_s}{N} \tag{6.2}$$

ovvero il rapporto tra la porzione di segnale e il numero totale di eventi. Segue immediatamente dalla definizione che $0 \leq P \leq 1$; in un rivelatore ideale vorremmo avere $P = 1$, in modo da poter sapere con certezza che tutti gli eventi selezionati sono effettivamente il segnale che vogliamo. Nella pratica, questo livello di purezza è irraggiungibile, poiché anche a seguito dei tagli e dell'applicazione dei criteri di selezione rimane sempre una porzione di eventi che viene identificata erroneamente come segnale.

D'altra parte, restringere troppo i criteri di selezione, pur contribuendo ad aumentare la purezza, ha l'effetto indesiderato di ridurre l'*efficienza* del rivelatore: altro parametro fondamentale definito come il rapporto tra il numero di eventi di segnale identificati e il numero di eventi di segnale effettivamente prodotti dal'interazione. Una bassa efficienza implica che buona parte degli eventi interessanti prodotti dall'acceleratore non viene identificata come tale, e quindi scartata.

[1] In generale il segnale è la porzione di eventi "interessanti" ai fini dell'analisi.

Di solito si cerca dunque di trovare il miglior compromesso che massimizzi la purezza con la minima perdita di efficienza: nel caso di ATLAS ci aspettiamo che la purezza nominale a regime[2] sia comunque molto elevata (dell'ordine di $\sim 0.9 \div 1$).

6.2 Isolation template fit

L'idea alla base dell'isolation template fit (vedi fig. 18) è quella di prendere la PDF[3] dell'energia d'isolamento del campione di fotoni *tight* (ovvero quella parte di eventi che ha passato tutta la serie di tagli che abbiamo imposto) e scomporla in due parti: la parte di segnale e quella di background, entrambe caratterizzate da una propria *shape* (ovvero da un'opportuna PDF) e da un proprio peso, in modo tale che:

$$M = N_{tight}(f \cdot S + (1 - f) \cdot B) \tag{6.3}$$

dove:

- M è la PDF "modello" (normalizzata in modo che $\int_{-\infty}^{\infty} M(x)\,\mathrm{d}x = N_{tight}$) che andrà a fittare i dati di fotoni tight;
- N_{tight} è il numero totale di eventi tight;
- S e B sono le PDF rispettivamente di segnale e background (normalizzate a 1);
- f è il peso di S rispetto a B nella combinazione lineare che va a formare M (naturalmente $0 \leq f \leq 1$).

Di questa scomposizione N_{tight} è l'unica quantità nota a priori; S e B possono essere ricavate da opportuni campioni di controllo o dal Monte Carlo. In questo lavoro ho utilizzato il seguente metodo:

- Per ricavare B, ho utilizzato i dati *non tight*, ossia tutti gli eventi che *non* hanno esplicitamente passato i tagli di selezione tight su[4]: δ_E, F_{side}, ω_{s3} e E_{ratio} (vedi 4.4.2 per il significato di queste variabili); tramite questi dati ho ottenuto la forma della PDF del background B che sono andato poi ad inserire nella composizione lineare di M. Questo procedimento è valido se si assume che la forma del background non sia influenzata dalla *tightness* e che non ci sia *signal leakage* che causi la presenza di eventi di segnale nella regione non-tight (ovvero, si richiede che i dati non tight siano costituiti puramente di eventi di background).
- Per ricavare S ho eseguito l'analisi su dati provenienti da simulazioni Monte Carlo di fotone inclusivo, considerando come segnale tutti e soli gli eventi di fotone vero in modo da ottenere una PDF di segnale che fosse certamente pura. Per il mio lavoro, ho utilizzato simulazioni Monte Carlo realizzate con Sherpa [12] (*Simulation of High-Energy Reactions of PArticles*), un generatore di eventi per reazioni ad alta energia.

Una volta note queste tre quantità, si procede fittando M ai dati tight, il che permette di ricavare f, che è l'ultima quantità rimasta incognita. Ricavata f, si ha un modello che descrive

[2] Come vedremo, la purezza dipende dall'energia del candidato fotone.

[3] *Probability Density Function*, ovvero la distribuzione di probabilità.

[4] questi sono stati ricavati invertendo uno dei bit della *bitmask* che rappresenta quali tagli loose e tight sono stati superati dall'evento.

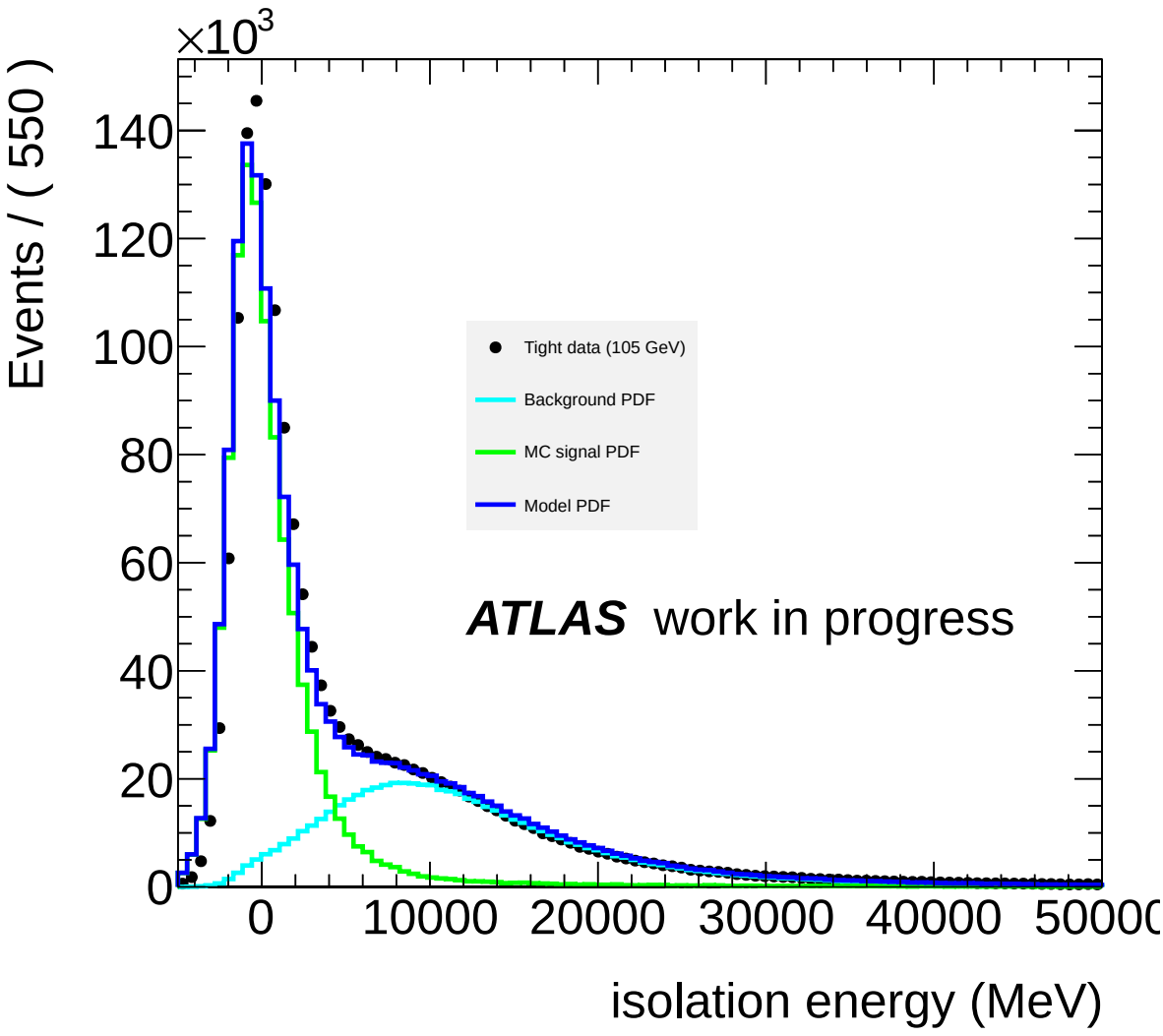


Figura 18: Illustrazione della tecnica dell'isolation template fit. In blu la PDF modello fittata ai dati, in verde quella di segnale (ricavata dal Monte Carlo) e in azzurro quella del background (ricavata dai dati non-tight).

la distribuzione di dati tight in termini di una somma di segnale e background: per ricavare la purezza si integra S (opportunamente pesato con f) in un'opportuna "regione di segnale", ottenendo così N_s, e si divide il risultato per il numero di eventi totali in quella regione. Questo procedimento viene effettuato per diversi bin di p_T del fotone in modo da ottenere un grafico della purezza al variare della sua energia. Nel corso di questa tesi, ho preso come regione di segnale l'intervallo $-5\,\text{GeV} \leq E_{iso} \leq 7\,\text{GeV}$[5] (fig. **??**).

6.2.1 Calcolo degli errori sistematici

Gli errori sistematici sui risultati possono venire da varie sorgenti. In particolare ho eseguito di nuovo l'analisi variando il parametro con cui discrimino gli eventi non-tight da quelli

[5]Ricordando da 4.5 la definizione di energia d'isolamento, notiamo che tale quantità può anche essere negativa, nel caso di un evento con particolarmente poca attività circostante.

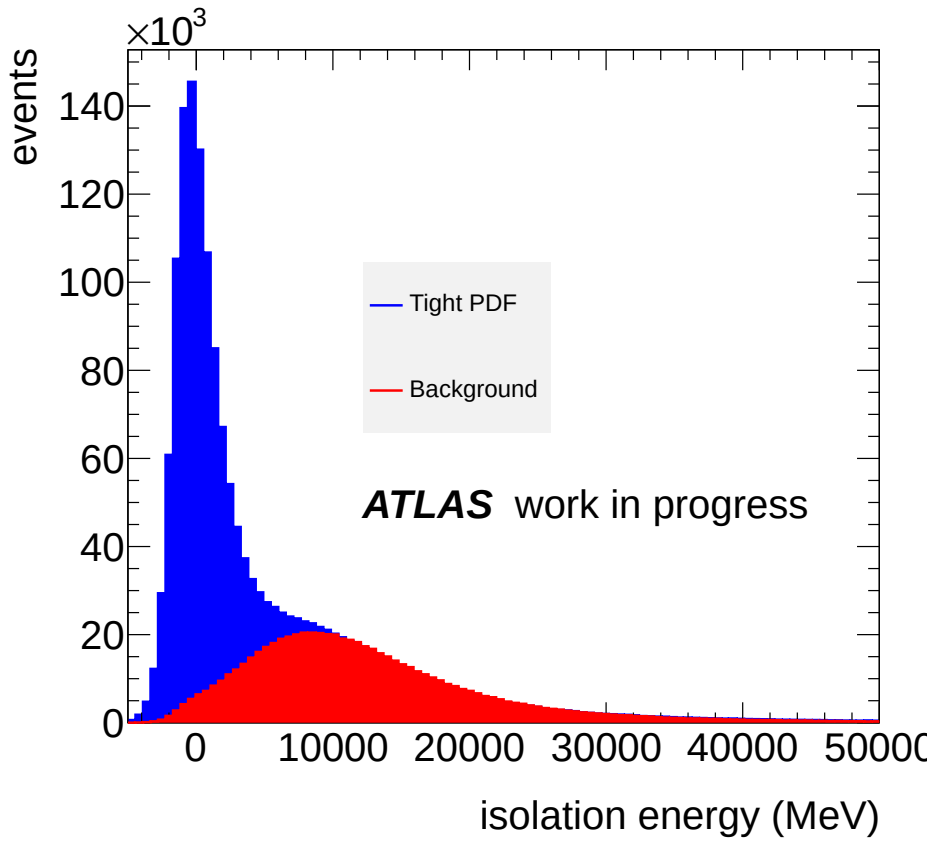


Figura 19: PDF dei fotoni tight. In blu è segnata la porzione di segnale e in rosso quella di background. Si può vedere che oltre $\sim 7\,\text{GeV}$ la porzione di segnale diventa trascurabile.

tight, ovvero la *bitmask* che descrive i tagli d'identificazione. Modificando tale parametro in modo da rendere la richiesta di tightness più o meno stretta, ho ottenuto risultati di purezza diversi rispetto all'analisi che usa il parametro nominale (fig. 26): ho quindi stimato l'errore sistematico come la differenza massima tra i risultati ottenuti bin per bin.

6.3 Risultati

Di seguito presento i risultati per la purezza ottenuti dall'isolation template fit (fig. 27). Come atteso, la purezza tende a 1 all'aumentare dell'energia del fotone.

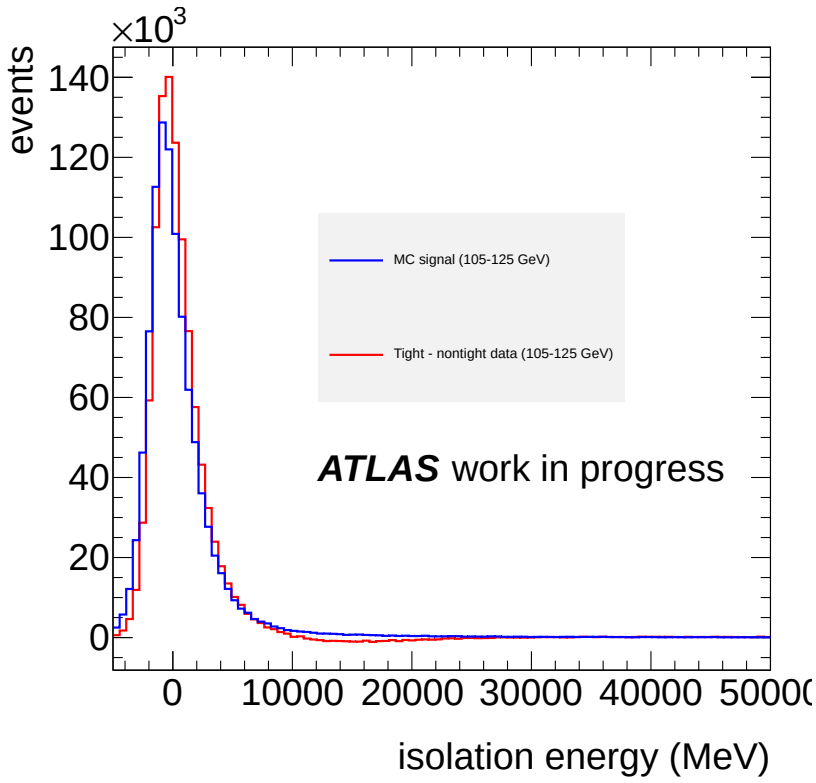


Figura 20: In blu è riportata la PDF di segnale del Monte Carlo, in rosso quella ottenuta dai dati come $tight - nontight$.

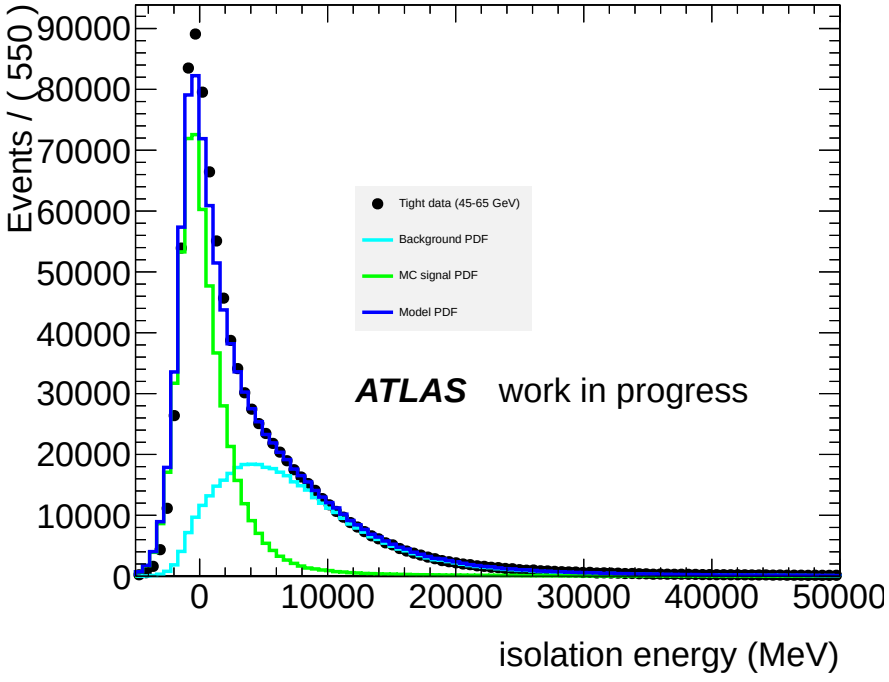


Figura 21: Isolation template fit a $45 \leq p_T \leq 65$GeV

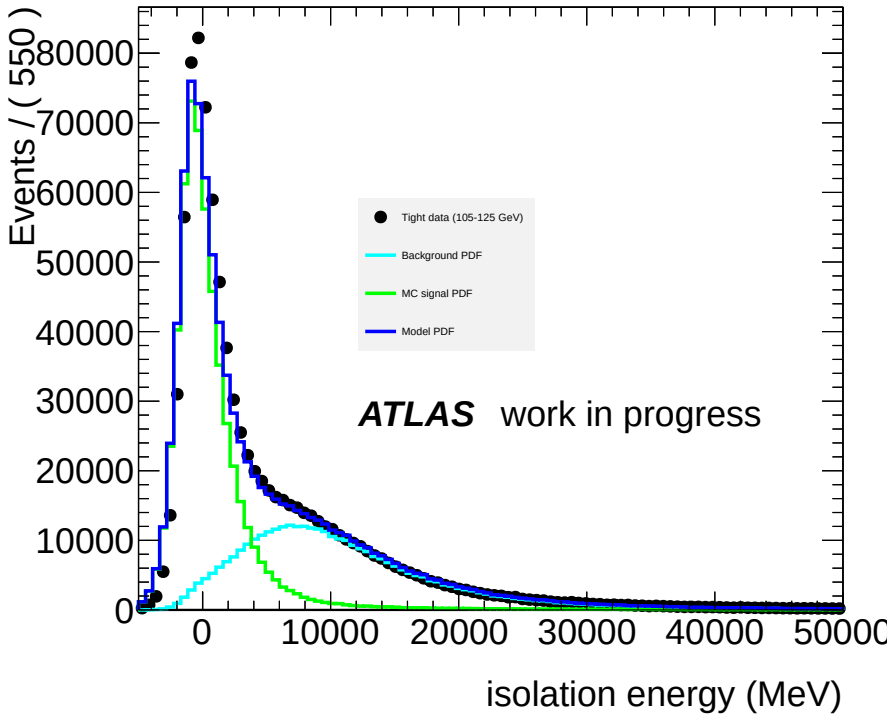


Figura 22: Isolation template fit a $85 \leq p_T \leq 105$GeV

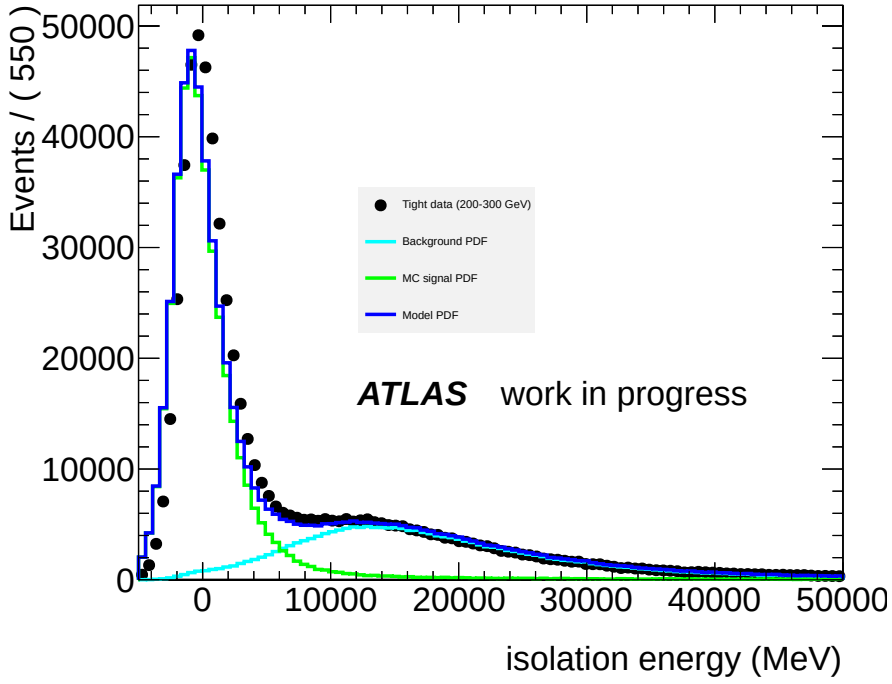


Figura 23: Isolation template fit a $200 \leq p_T \leq 300$GeV

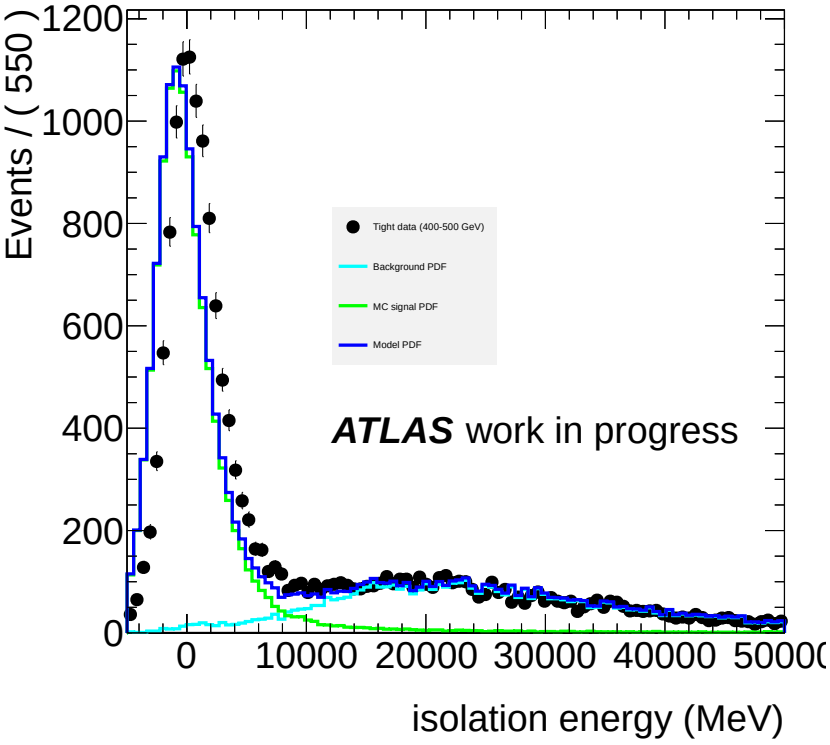


Figura 24: Isolation template fit a $400 \leq p_T \leq 500$GeV

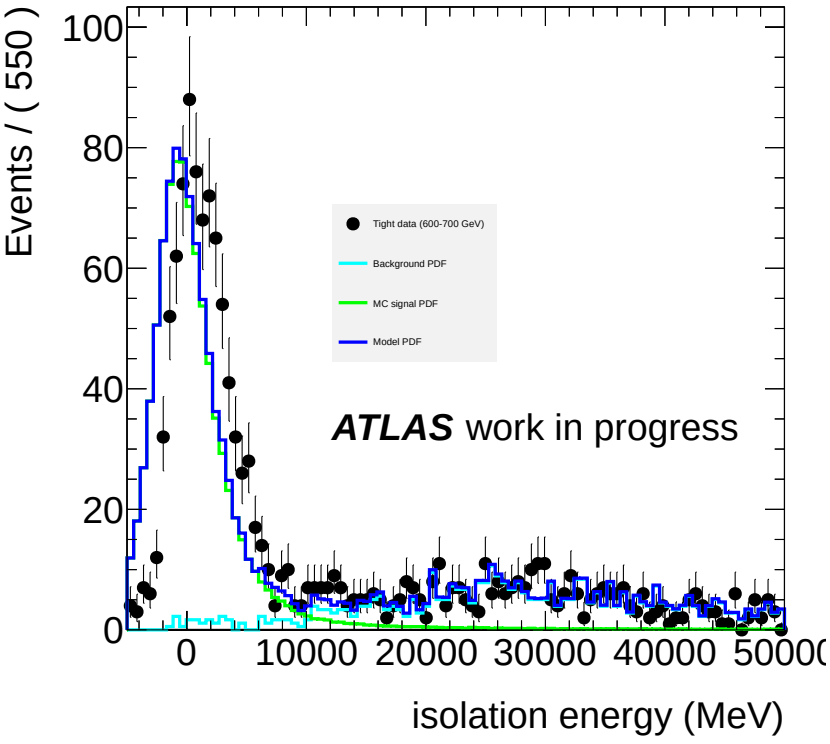


Figura 25: Isolation template fit a $600 \leq p_T \leq 700$GeV

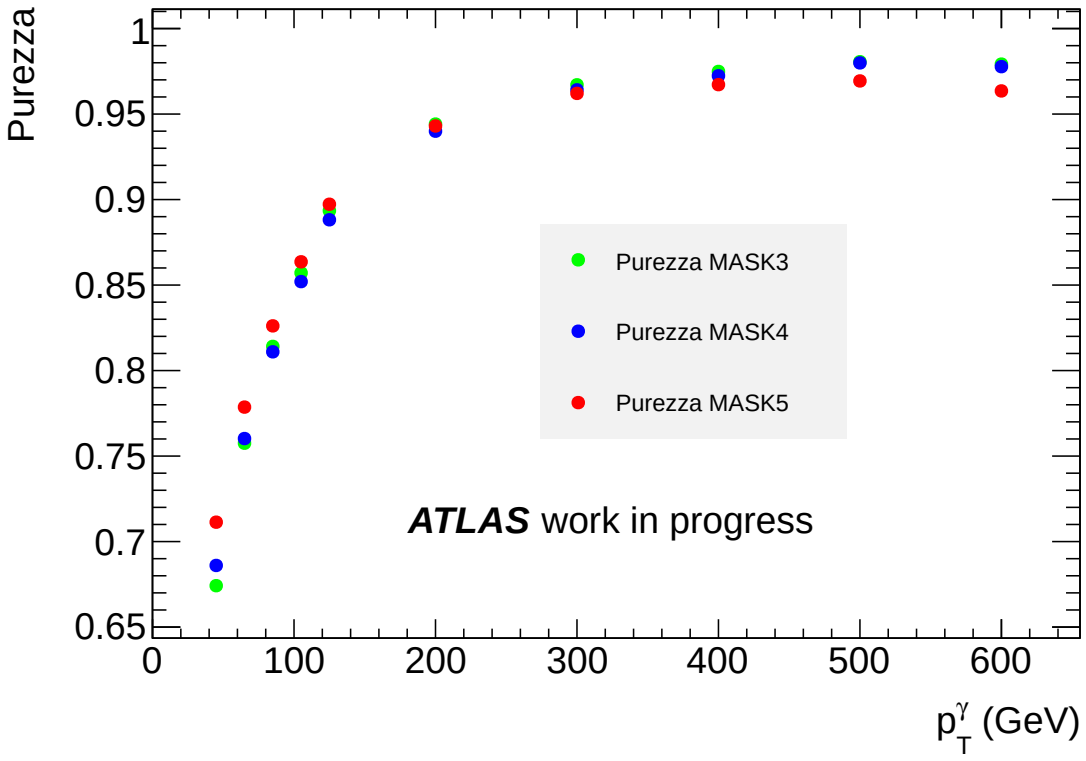


Figura 26: Grafici di purezza per tre diversi criteri di selezione tight. In blu i risultati corrispondenti al criterio nominale utilizzato.

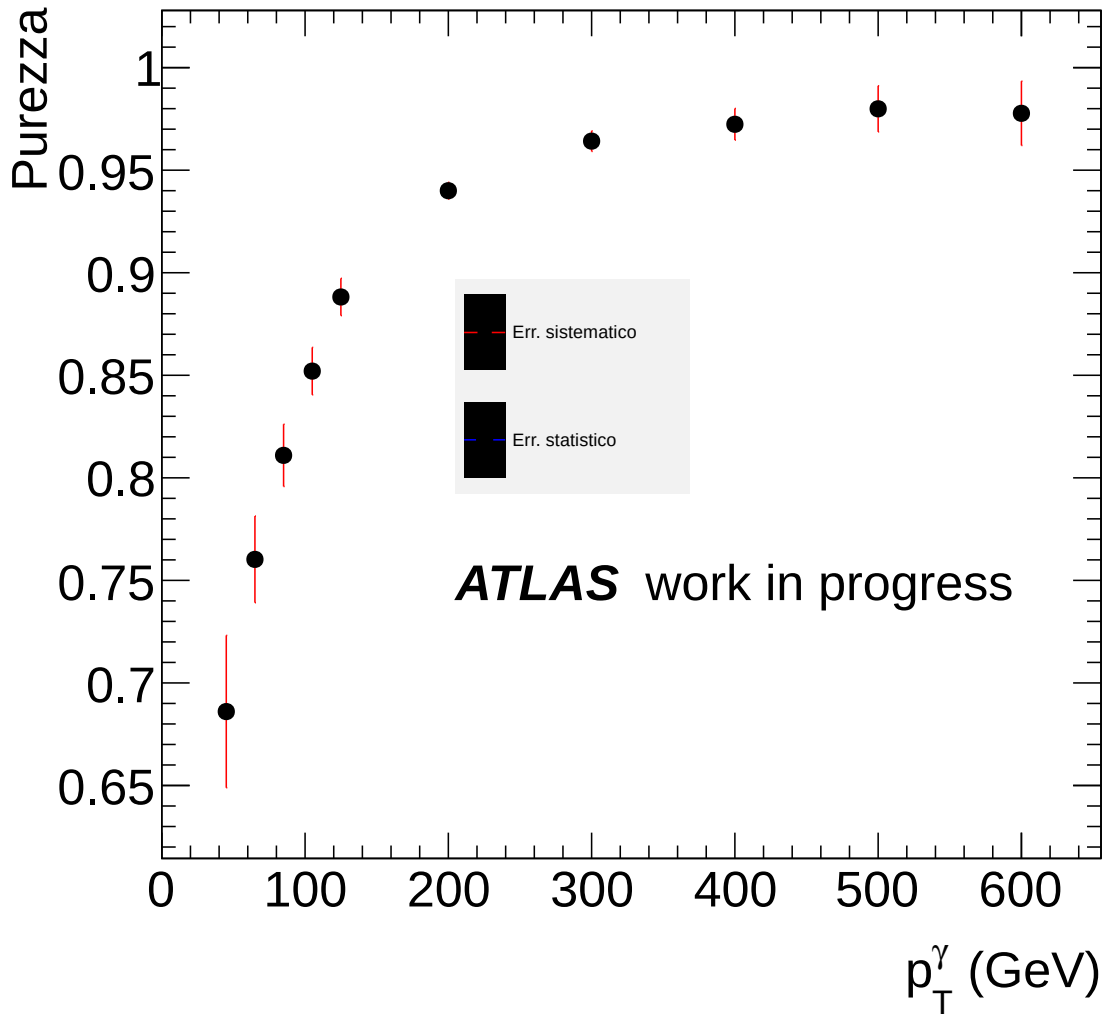


Figura 27: Risultati di purezza. In rosso è riportato l'errore sistematico, come stimato in 6.2.1; l'errore totale è completamente dominato dall'errore sistematico.

Capitolo 7

Conclusioni

Traggo ora le conclusioni sui due aspetti fondamentali di questo lavoro: il framework di analisi e il calcolo della purezza.

7.1 Framework d'analisi

L'approccio utilizzato dal framework si è rivelato soddisfacente: l'analisi sull'intero campione del 2012 ha richiesto circa 5 giorni per elaborare 10 263 file per un totale di ~ 700 milioni di eventi, tempo ragionevole data la grande mole di dati richiesta.

A scopo comparativo, ho confrontato i tempi di analisi di un singolo file piuttosto consistente ($\sim 4\,\mathrm{GB}$) usando prima il metodo di analisi locale e poi quello tramite framework: nel primo caso ho considerato come tempo totale la somma del tempo reale necessario per trasferire il file da un sito remoto in locale tramite XRootD e quello necessario a completare l'analisi vera e propria. Seguono i risultati:

Analisi locale	$t_{trasferimento} + t_{analisi} = 3702\,\mathrm{s} + 15\,\mathrm{s} = 3717\,\mathrm{s}$
Framework	$t_{analisi} = 2034\,\mathrm{s}$

Il tempo di esecuzione via framework risulta essere circa il 55% del tempo totale di esecuzione in locale, ovvero l'analisi risulta essere circa due volte più veloce. Naturalmente questa considerazione non tiene conto di molteplici altri fattori, quali la possibilità che l'analisi fallisca, la possibilità che la rete sia congestionata o che il sito su cui si sta svolgendo l'analisi diventi irraggiungibile o subisca qualche guasto, eccetera.

Si noti in particolare che il tempo di analisi locale è di gran lunga dominato dal trasferimento dei dati: una volta che questi ultimi sono presenti su storage locale l'analisi risulta molto più veloce di quella via framework. Il framework è dunque indicato per svolgere la selezione iniziale dei dati e salvare localmente le variabili utili: questo consente di ripetere l'analisi successivamente anche molte volte in tempi ridotti, avendo già a disposizione i dati.

Esiste inoltre un ampio margine di miglioramento sull'aspetto di gestione dell'analisi automatizzata da parte del framework: per cominciare, l'analisi via XRD è gestita in maniera piuttosto primitiva e semplice, e non sfrutta appieno le potenzialità della redirezione, oltre a gestire in modo relativamente grossolano la granularità dei singoli file[1]. Inoltre il buon anda-

[1] ad esempio, come accennato sopra, il fallimento dell'analisi di un singolo file inficia tutto il dataset corrispondente, per cui in caso di insuccesso vengono rianalizzati *tutti* i file del dataset.

mento dell'analisi è strettamente legato al numero di siti su cui si decide di eseguirla, visto che in caso di irraggiungibilità o altri errori di connessione il framework semplicemente rinuncia a portare avanti l'analisi e riprova nel ciclo successivo. Infine, questo approccio è presumibilmente tanto meno efficiente quanto maggiore è il numero di variabili che è necessario salvare per eseguire l'analisi (nel mio caso tale numero ammonta a poche decine).

7.2 Calcolo della purezza

Il metodo dell'isolation template fit permette di stimare la purezza del campione raccolto con grande precisione, facendo uso dell'intera shape delle distribuzioni di segnale e background, rivelandosi quindi più performante di un semplice conteggio sulle regioni di segnale e controllo; questo metodo inoltre può essere svolto con molte varianti, permettendo un'analisi completamente *data-driven* oppure aiutata da simulazioni Monte Carlo.

Naturalmente l'affidabilità dei risultati ottenuti dipende dalla correttezza delle assunzioni che ho fatto, in particolare:

1) ho assunto che il *signal leakage* sia contenuto, quindi che solo una parte trascurabile del segnale possa finire all'interno della zona di controllo (che abbiamo definito come $p_T > 20\,\text{GeV}$);

2) ho assunto che la forma del background rimanga pressoché invariata al variare della *tightness*.

La purezza cresce rapidamente con p_T, superando il 90% sopra i 100 GeV. L'errore sistematico risulta dominante in tutto il range di p_T esplorato.

I risultati finali appaiono soddisfacenti e compatibili con studi precedenti [15].

Bibliografia

[1] *RootCore* `https://twiki.cern.ch/twiki/bin/viewauth/AtlasComputing/RootCore`

[2] *DQ2 Clients How To* `https://twiki.cern.ch/twiki/bin/viewauth/AtlasComputing/DQ2ClientsHowTo`

[3] *XRootD website* `http://xrootd.org/index.html`

[4] *LHC: The guide* `http://cds.cern.ch/record/1092437/files/CERN-Brochure-2008-001-Eng.pdf`

[5] *ATLAS Collaboration, Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC, Phys. Lett, B, 716 1-29* (2012)

[6] *CMS Collaboration, Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC, Phys. Lett, B, 716 30-61* (2012)

[7] *ATLAS Collaboration, The ATLAS Experiment at the CERN Large Hadron Collider, JINST 3, S08003* (2008)

[8] *ATLAS. Detector and physics performance. Technical Design Report, Vol.1* (1999)

[9] *The Grid: a system of tiers* `http://home.web.cern.ch/about/computing/grid-system-tiers`

[10] *ATLAS Collaboration, Measurements of the photon identification efficiency with the ATLAS detector 4.9 fb−1 of pp collision data collected in 2011*

[11] Cacciari, Matteo and Salam, GavinP. and Sapeta, Sebastian *On the characterisation of the underlying event* (Journal of High Energy Physics, 2010)

[12] T. Gleisberg, S. Hoeche, F. Krauss, M. Schonherr, S. Schumann, et al. *Event generation with SHERPA 1.1* (Journal of High Energy Physics, 2009)

[13] S. Manzoni, *Misura della sezione d'urto per produzione inclusiva di fotoni diretti isolati in collisioni pp a 7 TeV nel centro di massa con il rivelatore ATLAS* (tesi di laurea Università degli Studi di Milano, 2009-2010)

[14] S. Mazza, *Stima in situ della purezza del campione di candidati fotoni nell'esperimento ATLAS a LHC* (tesi di laurea Università degli Studi di Milano, 2010-2011)

[15] E. Guiraud *Studio della purezza del campione di fotone inclusivo del Run I di ATLAS con il metodo 2D-sidebands e caratterizzazione del modello di accesso ai dati* (tesi di laurea Università degli Studi di Milano, 2012-2013)