



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

ESTUDIO DE LAS PROPIEDADES DE
FRAGMENTACIÓN DE JETS PRODUCIDOS EN
COLISIONES HADRÓNICAS UTILIZANDO DATOS
ABIERTOS DEL EXPERIMENTO CMS EN EL CERN.

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

FÍSICO

P R E S E N T A :

SAKSEVUL ARIAS SANTIZ

TUTOR

DR. HERMES LEÓN VARGAS



CIUDAD UNIVERSITARIA, Cd. Mx., 2020



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Ta stojol jtot-jme'...

Prólogo

Una de las preguntas más simples y fundamentales de la física ¿de qué están hechas las cosas? Si bien desde la antigüedad se ha tratado de responder de manera intuitiva a esto, Aristóteles aseguraba que la materia en la Tierra estaba compuesta por uno o más de los cuatro elementos fundamentales (agua, tierra, aire, fuego), ha sido hasta hace relativamente poco tiempo que se ha comenzado a intentar dar una respuesta más rigurosa. Poco a poco fue descubierto que los átomos están compuestos por electrones y un núcleo, que los núcleos a su vez están compuestos por nucleones (protones y neutrones). Y estos últimos están constituidos por unas partículas fundamentales llamadas *quarks* y gluones.

Una vez que consideramos a una partícula como fundamental, nos gustaría conocer su comportamiento. Esto debido al hecho de que la combinación de un número pequeño de bloques (partículas elementales) puede explicar la gran diversidad de construcciones (materia) que existen. En este sentido, fue el estudio de hadrones lo que incitó a Gell-Mann y a Zweig a proponer a los *quarks* como las piezas fundamentales que constituyen a los hadrones [1].

Conocer las propiedades de los *quarks* y gluones nos permitiría entender mejor la materia hadrónica, la cual constituye más del 99.9% de la materia de los átomos, y dado que toda la materia que conocemos está constituida por átomos, entonces podríamos plantear la siguiente interrogante: ¿Por qué no simplemente separar los *quarks* de los núcleos atómicos e intentar estudiarlos individualmente? Si es posible dividir el núcleo atómico para generar energía nuclear, consecuentemente, separar un *quark* de un nucleón no ha de ser tan difícil, ¿verdad? Bueno, aunque seamos capaces de desarrollar la metodología para “cortar” en pedazos a un protón, nos

sobrepasaría una propiedad de los *quarks* llamada “carga de color”, pues cualquier partícula hadrónica (es decir, constituida de *quarks* y gluones) que existe *libre* en el Universo debe tener color neutro. ¿Color? ¿Partículas de colores?

Ciertamente este término de *color* es un poco confuso pero es simple notación. Históricamente fue introducido para satisfacer el principio de exclusión de Pauli en bariones¹ conformados por 3 *quarks* similares. Aquí la idea principal es la misma que utilizan la mayoría de los televisores, monitores, celulares, entre otros, que cuentan con pantallas LED. En cada pixel hay 3 leds de colores rojo, verde y azul. Con variaciones en la intensidad de la iluminación de cada led respecto a los otros 2, podemos obtener prácticamente toda la gama de colores que conocemos. En particular si ponemos los 3 leds a la misma intensidad obtendremos el color blanco. Así pues, en términos de física de partículas: la combinación de rojo, verde y azul genera un color neutro. Este concepto es fundamental, pues únicamente podemos encontrar hadrones con color neutro en la naturaleza.

Sin embargo, aún existe otra manera menos intuitiva de generar un color neutro. Todas las partículas conocidas en el Universo tienen asociada una antipartícula con todos los números cuánticos invertidos. De esta manera, si existe un *quark*, digamos *u*, con carga de color verde, entonces debe de existir un *quark anti-u* con carga *anti-verde*. Extraño, ¿no? Lamentablemente la respuesta es no. No tiene nada de extraño. No olvidemos que esta etiqueta de “color” no tiene nada que ver con lo que cotidianamente llamamos color. Es simplemente una manera de llamar a esta propiedad de los *quarks*, por lo que tener una carga de color anti-verde no es nada descabellado.

En síntesis, no podemos obtener *quarks* o gluones libres porque su carga de color no sería cero. Pero si podemos experimentar con combinaciones de éstos que tengan un color neutro. Es por esto que para estudiar a los *quarks* se necesita estudiar a los hadrones que los contienen. ¡Esto es una gran noticia! ¿No? Si consideramos que la gran mayoría de la materia que conocemos en el Universo (protones y neutrones) son hadrones entonces basta con echar un vistazo alrededor para estudiar los constituyentes del Cosmos. Una vez más, la respuesta es no.

¹Hadrones constituidos por 3 *quarks*

Según se sabe, existen por lo menos 6 tipos o sabores (de nuevo, es simple notación) de *quarks* con sus respectivos anti-*quarks*. Los protones y neutrones están constituidos por *quarks* de los 2 sabores más ligeros. A saber, están formados por los *quarks* *u* y *d*. Entonces, si sólo miramos a nuestro alrededor nos estamos perdiendo de 4 sabores que nunca conoceremos, y además, estos *quarks* son inestables, lo que significa que no “viven” por “mucho” tiempo y decaen a sabores más ligeros y estables. Por lo que si no podemos encontrarlos, únicamente resta “fabricarlos” en el laboratorio.

Este concepto de “crear” y detectar partículas es la labor principal de un colisionador de partículas. La idea es relativamente simple. Según la famosa ecuación de Einstein, que relaciona la masa con la energía, $E = \gamma mc^2$, tenemos que la energía de una partícula es proporcional a su masa en reposo, una constante (c^2 , la velocidad de la luz en el vacío al cuadrado) y a un factor (γ) que depende de la velocidad de dicha partícula. Resultando en que, a mayor velocidad de una partícula, mayor será su energía. Lo hermoso de esta ecuación es justamente el hecho que si la energía y la masa son equivalentes entonces podemos obtener una a partir de la otra. En otras palabras, mientras que en un reactor nuclear se convierte masa en energías, ¡en un colisionador de partículas podemos obtener materia a partir de energía!

Un colisionador de partículas utiliza dos conjuntos de partículas que viajan a altas velocidades en sentido contrario, cuando se colisionan estos dos conjuntos, la energía que tenían debido a las altas velocidades a las que viajan se transforma en materia. Como resultado, de dos protones que colisionan podemos obtener cientos o miles de nuevas partículas viajando mucho menos rápido que los protones originales (esto último por conservación de la energía). Tomando en cuenta que mientras más pesada sea una partícula, menos probable es que se genere, y mientras la energía inicial lo permita, no hay una restricción para el tipo y la cantidad de partículas generadas.

En palabras de los científicos del detector de partículas CMS: *“Los colisionadores de partículas son las herramientas más poderosas que tenemos para estudiar los ladrillos de construcción del nuestro Universo y las leyes que lo gobiernan. Los detectores gigantes como el CMS, actúan como cámaras que toman “fotografías” de las colisiones entre partículas permitiéndonos comprobar nuestro entendimiento de la na-*

*tural*eza [2]”, y es justamente lo que se hará en este trabajo, intentar entender un poco más nuestro Universo mediante los datos obtenidos y reconstruidos por el detector CMS en el CERN².

En términos generales, ésta tesis se estructura con la intención de poder servir como manual para que cualquier persona con acceso a internet y a una computadora pueda hacer análisis de datos abiertos del CERN. En este sentido, existen 4 partes y cada una tiene un propósito específico.

La primera es la Introducción, la cual se centra en dar un breve preámbulo a las herramientas y los conceptos utilizados en ésta tesis. El lector versado en el área de la física de partículas puede omitir este capítulo o consultarlo para temas puntuales. Es de recalcar que esta sección será mejor entendida por personas que tengan conocimientos física básica y nociones elementales de programación.

La segunda es la Metodología y se enfoca en desarrollar las herramientas utilizadas para llevar a cabo este trabajo. Se centra en la implementación de herramientas computacionales para obtener resultados de física.

La tercera son los Resultados, consiste en interpretar y estudiar en términos de física los resultados obtenidos.

La cuarta son las Conclusiones. Su objetivo es recapitular todo el trabajo hecho y los resultados obtenidos a fin dar una visión general de todo el camino recorrido.

Espero que disfrutes, te asombren y aprendas al leer esta tesis, tanto como yo lo hice al escribirla! Y si tienes dudas, no dudes en escribir a saksevul@ciencias.unam.mx.

²La Organización Europea para la Investigación Nuclear (CERN) es centro de investigación en física de partículas más grande del mundo; se ubica en Ginebra, Suiza.

Agradecimientos

Yox ora j-ta, chib muk' ta vinik-ants j-ta k'uchal jtot-jme' voch'oik tsots abt'ejemik s-junul s-kuxlejalik. La skoltaikun k'uchal j-ta batel j-be, x-chiuk la s-chanumbtasikun k'u x-elan ti lekil kuxlejal-e, la s-tsatsubtasik k'onton k'alal k'un ka-ay j-ba, xchiuk yak'objikun k'usi tsots s-k'oplale: s-k'anojikun x-chiuk skotol yo-ontunik. Kolaval s-yojoviletik, ta a tojolik ch-bat li' kabtele. J-k'anojoxikot tajek.

Kolyal schanibal kich'oktak Vero, Emi, Ulises x-chiuk Camila. Vo'oxuk la vak'bikun ta ilel ti skotol k'usi xu' xk'ot ta pasel ku'une, oy boch'o yan ts-chan spasel. Jech x-tok kolavalik vixubtak, Diana x-chiuk Brisa, oy yik'al muk' ko'olok lij ch'i tal, pisil-e la koltaikun k'alal bik'itun ox-e.

Graicas a los Yamamoto: Yuriko, Leticia, Jaime y Tombo, en especial a ti mi querida Yu, que me han abierto las puertas de su casa y corazón. Prácticamente se han convertido en una familia para mí.

A mis amigos de la facultad Alan, Martín, Brandon, Marianne, Alison, Santiago, Mónica, entre otros, por brindar su apoyo durante todo este tiempo y estar ahí si los necesitaba. Gracias por todas las aventuras que vivimos juntos, son los mejores.

A mis amigos Elena, Jacqueline y Luis. Siempre ha sido una locura pasar tiempo con ustedes.

A Hermes por asesorarme, brindarme consejo y ayuda durante todo este tiempo, sí se pudo. A mis compañeros Janette, Diego, Daniel, Abel y Reina, por haberme ayudado siempre que lo necesité y por todas las experiencias que vivimos juntos durante este tiempo.

Al Dr. Ángel Prieto y a Youssef Sarkis por haber confiado en mí para ser su ayudante, brindarme su apoyo, y por sus valiosos consejos.

A mis sinodales, de los cuales tuve el privilegio ser alumno, gracias por sus clases, tiempo y dedicación. En especial a la Dra. Mirna Villavicencio quien revisó mi tesis tan pronto cómo se la entregué, al Dr. Eleazar Cuautle por retarme a aprender a utilizar Pythia, a la Dra. Elizabeth Martínez que fue quien más molestias se tomó al revisar este trabajo, y al Dr. Ángel Sanchez por su siempre buena disposición.

A Carlos, Toño y Lucio con los que viví durante este tiempo. Aunque hemos tenido altibajos, han sido unos excelentes compañeros.

A la Universidad Nacional Autónoma de México por abrirme sus puertas y proveerme con una educación de calidad.

Al CERN por haber publicado sus datos para el libre acceso, así debería ser la ciencia. En especial a Kati Lassila-Perini, Edgar F. Carrera y Matevz Tadel del equipo CMS por resolver mis dudas respecto a CMS *Open-Data*.

Al Sistema de Becas para Estudiantes Indígenas por haberme apoyado durante la carrera. En especial a Gabriela, Carlos y Emiliano quienes fueron mis tutores y siempre estuvieron ahí para brindar su mano.

Agradezco el apoyo del proyecto CONACyT 254964 y el Fondo Sectorial de Investigación para la Educación.

Glosario

AOD (*Analysis Object Data*): Datos de Objetos de Análisis, un formato de datos utilizado para almacenar información referente a colisiones de partículas.

CERN (*Conseil Européen pour la Recherche Nucléaire*): Organización Europea para la Investigación Nuclear.

Clusters: Agrupaciones o amontonamientos, por ejemplo de energía o de partículas.

CMS (*Compact Muon Solenoid*): Solenoide Compacto de Muones, el cual es un detector de partículas multipropósito y se encuentra en el LHC.

ECAL (Calorímetro Electromagnético): Parte del detector CMS que detecta la energía de los electrones y fotones.

Evento(s): Esta palabra cuyo significado depende del contexto; ver secciones 1.3.1, 1.4.3 y 1.9.1.

FastJet: Es una paquetería computacional el cual se dedica a la reconstrucción de jets.

HCAL (Calorímetro Hadrónico): Parte del detector CMS que detecta la energía de los hadrones.

Jet: Conjunto de partículas que viajan en una misma dirección y sentido. La característica principal de estas partículas es que han sido generadas a partir de un mismo partón (*quark* o gluón).

LHC (*Large Hadron Collider*): Gran Colisionador de Hadrones, es un acelerador de partículas encontrado en el CERN.

Macro (Macro-instrucción): En términos prácticos es una instrucción compleja, formada por otras instrucciones más sencillas, la cual es interpretada y ejecutada mediante algún lenguaje de programación.

MC: Monte Carlo, método computacional basado en números aleatorios.

MV (Máquina Virtual): En la explicación más banal, es una computadora simulada dentro de otra computadora.

PF (ParticleFlow): Algoritmo de reconstrucción de partículas a partir de la información recolectada por un detector de partículas.

Pseudo-rapidez (η): Es una coordenada espacial utilizada por CMS, proviene de la transformación de la coordenada θ del sistema coordenado esférico.

Pythia: Programa computacional para la simulación de colisiones de partículas.

Root: Programa dedicado al análisis datos.

Tracks: Trayectorias de partículas cargadas, o simplemente, partículas partículas.

Trigger: Sistema de selección. Funciona como un gatillo que se ejecuta cuando se cumple con ciertas condiciones preestablecidas.

tunings: Son parámetros especializados de Pythia para modificar una simulación.

UNAM: Universidad Nacional Autónoma de México.

Unfoldings: En física de partículas, son las transformaciones hechas para obtener las distribuciones verdaderas al tomar en cuenta la resolución y aceptación de los detectores.

Resumen

Esta tesis se enfoca en estudiar las propiedades de jets producidos en colisiones protón-protón con Pythia 8, a una energía en el centro de masa de 7 TeV, mediante el uso de datos públicos (abiertos) del experimento CMS en el CERN, en el cual no participa la UNAM, pero que gracias a la iniciativa de datos abiertos del CERN son accesibles.

A diferencia de los centros de investigación de física de partículas en donde se utilizan herramientas de súper cómputo para llevar a cabo análisis, este trabajo se distingue por haberse realizado utilizando una computadora portátil (Intel Core i5, 8 GB en Ram y tarjeta gráfica de 1GB), un disco duro externo de 10 TB y una conexión a internet, por lo que está estructurado a fin de poder ser utilizado como manual para analizar datos abiertos del CERN. En este sentido, existen dos apartados principales. El primero se centra en desarrollar la metodología computacional para el análisis de datos, se construyen *macros* para: descargar archivos de datos (AODs), extraer la información relevante de estos y hacer análisis recursivos. Los AODs utilizados son de los conjuntos *BTag*, *Jet*, *MinBias* y *MultiJet*, que corresponden a colisiones del año 2011 a una energía de 7 TeV. El segundo apartado hace referencia a la simulación, en la cual la reconstrucción de jets es llevada a cabo mediante la paquetería FastJet 3 y se utilizan 4 algoritmos distintos (igual que en los datos abiertos). De la misma manera que sucede con el análisis de datos, se desarrollan diversos *macros* para la simulación.

Con el propósito de presentar los resultados obtenidos en el análisis de datos y la simulación, se define una Función de Fragmentación (FF) para jets y se hace un desglose respecto a la energía de estos últimos. Mediante la FF se explora el comportamiento de la fragmentación de jets para los 5 tipos de partículas que son detectables: muones, fotones, electrones y hadrones cargados y neutros. De manera satisfactoria, se encuentran concordancias entre la simulación y los conjuntos de datos, al considerar al menos 3 algoritmos de reconstrucción de jets, para 4 tipos de partículas.

Todo el material computacional desarrollado para este trabajo se encuentra disponible en la siguiente página de GitHub:

<https://github.com/Saksevul/T>

Índice general

Prólogo	II
Agradecimientos	VI
Glosario	VIII
Resumen	x
1. Introducción	1
1.1. El Modelo Estándar	2
1.1.1. Interacciones y partículas	3
1.2. Cromodinámica Cuántica	4
1.3. <i>Conseil Européen pour la Recherche Nucléaire</i> (CERN)	5
1.3.1. <i>Large Hadron Collider</i> (LHC)	6
1.3.2. Portal de Datos Abiertos del CERN	6
1.4. Detector <i>Compact Muon Solenoid</i> (CMS)	7
1.4.1. Convención de coordenadas	7
1.4.2. Configuración del CMS	9
1.4.3. Eventos en CMS	11
1.4.4. Sistema de <i>trigger</i> de CMS	11
1.4.5. Convención de unidades para software	12
1.5. <i>Open-Data</i> de CMS	12
1.5.1. Jerarquía de los datos de CMS	13
1.5.2. <i>Analysis Object Data</i> (AOD)	13

1.5.3.	Conjuntos de AODs utilizados	14
1.5.4.	Máquina Virtual (MV)	15
1.6.	<i>Particle Flow</i> (PF)	15
1.7.	Funciones de Fragmentación (FFs)	17
1.8.	Jets	17
1.8.1.	Fragmentación de jet	18
1.8.2.	Algoritmos de reconstrucción secuencial de jets	19
1.8.3.	Tipos de jets	21
1.9.	Pythia	21
1.9.1.	Eventos en Pythia	22
1.9.2.	Convención de unidades	22
1.9.3.	Configuración de una simulación	23
1.10.	FastJet	23
1.10.1.	Parámetros de reconstrucción de jets	24
1.11.	Root	24
2.	Metodología	25
2.1.	Análisis	25
2.1.1.	Descripción de la descarga de archivos AOD	26
2.1.2.	Archivos AODs descargados	29
2.1.3.	Información utilizada	30
2.1.4.	Extracción de información de los AODs	34
2.1.5.	Extracción recursiva de información contenida en distintos AODs	36
2.1.6.	Límites de detección del CMS	37
2.2.	Simulación	41
2.2.1.	<i>Macro</i> de simulación	41
2.2.2.	Generalización del <i>macro</i> de simulación	45
3.	Resultados	48
3.1.	Resultados de la FF propuesta	51
3.1.1.	Muones	51

3.1.2. Fotones	53
3.1.3. Electrones	60
3.1.4. Hadrones neutros	61
3.1.5. Hadrones cargados	64
4. Conclusiones	71
A. Apéndice A	73
A.1. Estructura de AODs	73
A.2. Ventana Deslizante	74
A.3. Área del jet	75
A.4. Ramas de muones y electrones dentro de los AODs	75
B. Apéndice B	78
B.1. Estructura de los directorios relevantes	78
B.2. Carpeta compartida en la MV	78
B.3. Configuración del entorno de Root	80
B.3.1. Navegar por los AODs	81
B.4. Metodología recursiva para AODs	82
B.4.1. Aumentando la cantidad de información analizada	83
B.4.2. Considerando distintos algoritmos de reconstrucción de jets PF	85
B.4.3. Utilizando distintos tipos de AODs	86
B.5. Conectar Pythia con otros programas	87
B.6. Información de usuario en FastJet	89
C. Apéndice C	92
C.1. Prueba de límites inferiores de energía en la simulación	92
C.2. Desglose completo de energías de la FF	93
C.2.1. ak5Jets	94
C.2.2. ak7Jets	104
C.2.3. kt4Jets	114
C.2.4. kt6Jets	124

Índice de figuras

1.1.	Representación de la interacción entre los constituyentes (<i>quarks</i> y gluones) de dos protones colisionando. El momento del protón 1 es \vec{p}_{P_1} , el momento del protón 2 es \vec{p}_{P_2} , el momento del partón 1 es $\vec{p}_{parton1}$ y el momento del partón 2 es $\vec{p}_{parton2}$ [3].	2
1.2.	Las partículas elementales del Modelo Estándar [4].	3
1.3.	Modelo a escala 1:200 del detector CMS construido con piezas LEGO®[5].	7
1.4.	Sistema coordinado en el experimento CMS [6].	8
1.5.	Distribución de los sub-detectores dentro de CMS [7].	9
1.6.	Visualización de un evento <i>MinBias</i> en CMS utilizando herramientas del portal de datos abiertos del CERN [8]. En la figura se muestran las trayectorias de las partículas cargadas (líneas amarillas), un muón (línea roja) y dos jets (conos amarillos).	18
1.7.	Formación de jets a partir de la separación de un <i>quark</i> y un anti- <i>quark</i> .	19
2.1.	Estructura jerárquica dentro de un AOD. El AOD se llama <code>0001.root</code> , nuestro árbol tiene el nombre de <code>Eventos</code> , el cual almacena a la rama <code>reocoPFJets_ak5PFJets__RECO.</code> , dentro de la cual está contenida la rama <code>obj.</code> y en esta última rama tenemos a las hojas <code>pt_</code> , <code>eta_</code> <code>yphi_</code> .	33
2.2.	Histograma de la distribución de la energía entre la multiplicidad para electrones dentro de <code>ak5PFJets</code>	39
2.3.	Distribuciones de energía entre la multiplicidad para los 4 constituyentes restantes de <code>ak5PFJets</code> en los AODs.	39

3.1. Función de fragmentación para fotones usando ak5MCJets . Los diferentes histogramas son el resultado de usar distintas funciones para describir la relación entre el número de eventos y el \hat{p}_T^{\min}	50
3.2. Función de fragmentación respecto a 4 constituyentes en jets simulados (ak5MCJets). Los distintos colores denotan distintas relaciones utilizadas entre el número de eventos y el \hat{p}_T^{\min}	50
3.3. Función de fragmentación para muones contenidos en ak5Jets . El número de entradas en cada bin es $\frac{dN}{d(E^{\text{Muon}}/E^{\text{ak5Jets}})}$, adicionalmente, el histograma se encuentra normalizado al número total de entradas: N_{Tot} . La variable ξ , descrita al inicio de éste capítulo, es utilizada para obtener una medida cuantitativa del ajuste de la simulación a los conjuntos de datos.	52
3.4. Función de fragmentación para muones contenidos en ak7Jets	52
3.5. Función de fragmentación para muones contenidos en kt6Jets	53
3.6. Comportamiento de la FF para fotones contenidos en ak5Jets	54
3.7. Comportamiento de la FF para fotones contenidos en ak7Jets	54
3.8. Comportamiento de la FF para fotones contenidos en kt4Jets	55
3.9. Comportamiento de la FF para fotones contenidos en kt6Jets	55
3.10. Función de fragmentación para fotones contenidos en ak5Jets cuya energía $\in [30, 100)$ GeV.	56
3.11. Función de fragmentación para fotones contenidos en kt4Jets cuya energía $\in [30, 100)$ GeV.	57
3.12. Función de fragmentación para fotones contenidos en kt6Jets cuya energía $\in [30, 100)$ GeV.	57
3.13. Función de fragmentación para fotones contenidos en ak5Jets cuya energía $\in [100, 500)$ GeV.	58
3.14. Función de fragmentación para fotones contenidos en ak7Jets cuya energía $\in [100, 500)$ GeV.	59
3.15. Función de fragmentación para fotones contenidos en kt4Jets cuya energía $\in [100, 500)$ GeV.	59

3.16. Función de fragmentación para fotones contenidos en kt6Jets cuya energía $\in [100, 500)$ GeV.	60
3.17. Función de fragmentación para electrones contenidos en ak5Jets	61
3.18. Función de fragmentación para electrones contenidos en ak5Jets cuya energía $\in (0, 100)$ GeV.	62
3.19. Función de fragmentación para electrones contenidos en ak5Jets cuya energía < 100 GeV.	62
3.20. Función de fragmentación para hadrones neutros contenidos en ak5Jets	63
3.21. Función de fragmentación para hadrones neutros contenidos en ak7Jets	63
3.22. Función de fragmentación para hadrones neutros contenidos en ak5Jets cuya energía $\in [100, 400)$ GeV.	64
3.23. Función de fragmentación para hadrones neutros contenidos en ak7Jets cuya energía $\in [100, 400)$ GeV.	65
3.24. Función de fragmentación para hadrones neutros contenidos en kt4Jets cuya energía $\in [100, 400)$ GeV.	65
3.25. Función de fragmentación para hadrones neutros contenidos en kt6Jets cuya energía $\in [100, 400)$ GeV.	66
3.26. Función de fragmentación para hadrones cargados contenidos en ak5Jets	66
3.27. Función de fragmentación para hadrones cargados contenidos en ak7Jets	67
3.28. Función de fragmentación para hadrones cargados contenidos en kt4Jets	67
3.29. Función de fragmentación para hadrones cargados contenidos en kt6Jets	68
3.30. Función de fragmentación para hadrones cargados contenidos en ak5Jets cuya energía $\in [60, 800)$ GeV.	69
3.31. Función de fragmentación para hadrones cargados contenidos en ak7Jets cuya energía $\in [60, 800)$ GeV.	69
3.32. Función de fragmentación para hadrones cargados contenidos en kt4Jets cuya energía $\in [60, 800)$ GeV.	70
3.33. Función de fragmentación para hadrones cargados contenidos en kt6Jets cuya energía $\in [60, 800)$ GeV.	70

A.1. Representación simplificada de la disposición de los datos en un archivo AOD.	73
A.2. Disposición del método “ventana deslizante” [9].	74
A.3. Área de jets reconstruidos con algoritmos <i>ak</i> y <i>kt</i> , utilizando con el mismo conjunto de datos y el mismo radio [10]. Podemos notar una alta irregularidad en el área de los jets en el algoritmo <i>kt</i> en comparación con el algoritmo <i>ak</i> . Para el algoritmo <i>kt</i> la forma exacta de las áreas son determinadas por el conjunto específico de fantasmas utilizados y cambia cuando los fantasmas son modificados [10]. Los distintos colores son utilizados para representar distintos jets y sus respectivas áreas.	75
B.1. Distribución de los directorios principales utilizados en el desarrollo de este trabajo.	79
B.2. Navegador TBrowser de Root.	82
B.3. <i>Macro</i> de extracción recursiva de datos. Este procedimiento es llevado a cabo por el <i>macro</i> Aumentar.bash para considerar información de distintos AODs.	84
B.4. <i>Macro</i> de extracción recursiva de datos. Este procedimiento es llevado a cabo por el <i>macro</i> Recursivo.bash para considerar información de distintos AODs y los 4 algoritmos de reconstrucción de jets PF disponibles en los AODs.	86
C.1. Distribución de momento transversal, p_T , de muones contenidos en jets reconstruidos con el algoritmo <i>ak5</i>	92
C.2. Distribución de momento transversal, p_T , de electrones contenidos en jets reconstruidos con el algoritmo <i>ak5</i>	93

Capítulo 1

Introducción

Una de las colisiones más comunes llevadas a cabo en los colisionadores de partículas es la de protón-protón. Los protones forman parte de la familia de los hadrones, lo que significa que están conformados por *quarks* y es por este motivo que, en una colisión a alta energía¹ entre protones, se estudian las interacciones entre sus constituyentes, los *quarks* y los gluones.

Por lo tanto, el contenido de *quarks* en un protón está dado por los *quarks* de valencia (*u* y *d* -ver sección 1.1-), y el mar de *quarks* y gluones. Por lo tanto, en una colisión de protones, cualquier combinación de *quarks* o gluones puede contribuir al proceso de dispersión fuerte. La probabilidad de que un partón² específico, con una fracción de momento del protón, participe en un proceso fuerte es conocido como función de distribución partónica y se mide usando datos de experimentos de dispersión inelástica profunda [11]. La energía de cada partón (*quark* o gluón) es una fracción desconocida de la energía del protón del que forma parte.

Una representación ilustrativa de este fenómeno la podemos encontrar en la figura 1.1, donde se muestra a dos protones a punto de colisionar. Notemos que dentro de ellos está representado el mar de *quarks* y gluones.

Si bien es cierto que dentro de cada protón está un mar de *quarks* y gluones, en

¹En este trabajo consideraremos del orden de GeV para la energía en el centro de masa en las colisiones, pues a esto nos restringe los datos abiertos.

²De la misma manera que un nucleón puede ser un protón o un neutrón, llamamos partón a un *quark* o a un gluón.

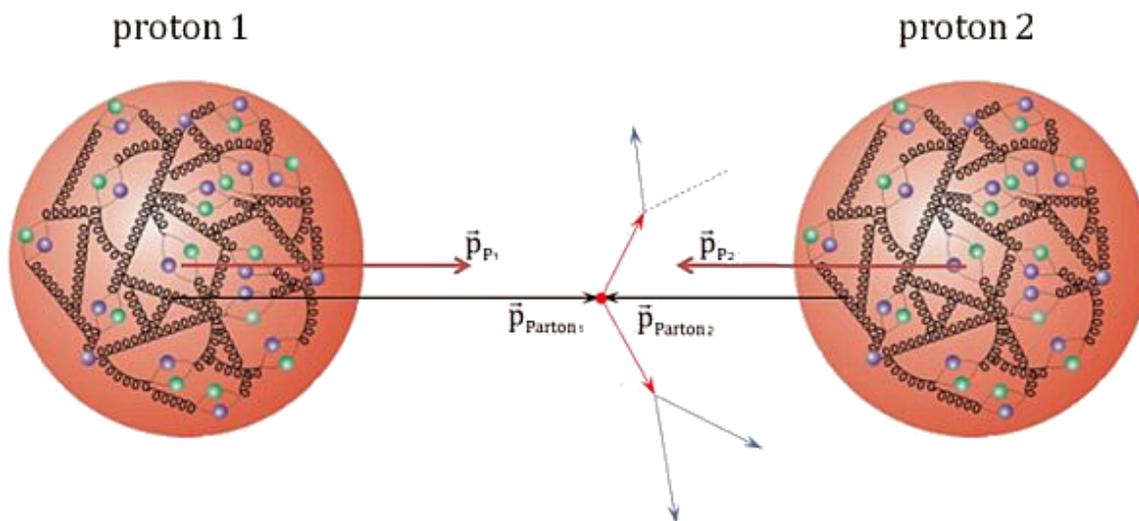


Figura 1.1: Representación de la interacción entre los constituyentes (*quarks* y gluones) de dos protones colisionando. El momento del protón 1 es \vec{p}_{P_1} , el momento del protón 2 es \vec{p}_{P_2} , el momento del partón 1 es \vec{p}_{parton_1} y el momento del partón 2 es \vec{p}_{parton_2} [3].

realidad los que le dan la identidad de protón a cada protón son los llamados *quarks* de valencia.

Para saber un poco más a cerca de los procesos llevados a cabo en las colisiones, vale la pena estudiar cuales son las partículas fundamentales de la física y cómo interactúan entre ellas. Con este fin se comienza presentando el Modelo Estándar.

1.1. El Modelo Estándar

Hoy en día, sabemos que toda la materia visible en el Cosmos está compuesta por un número pequeño de partículas, cuyo comportamiento está gobernado por 4 fuerzas distintas. El Modelo Estándar, en física de partículas, es un compendio de las partículas y fuerzas elementales que describen nuestro Universo visible (figura 1.2). Este modelo consiste en seis *quarks* y seis leptones agrupados en tres generaciones. Además, cuenta con partículas mediadoras, normalmente llamadas bosones de norma. Por último, cuenta con el bosón de Higgs el cual proporciona masa a los *quarks*, leptones y bosones W^\pm y Z^0 .

Las partículas de una generación más alta se desintegran a través de interacciones débiles a las partículas de una generación más baja, explicando porqué el mundo

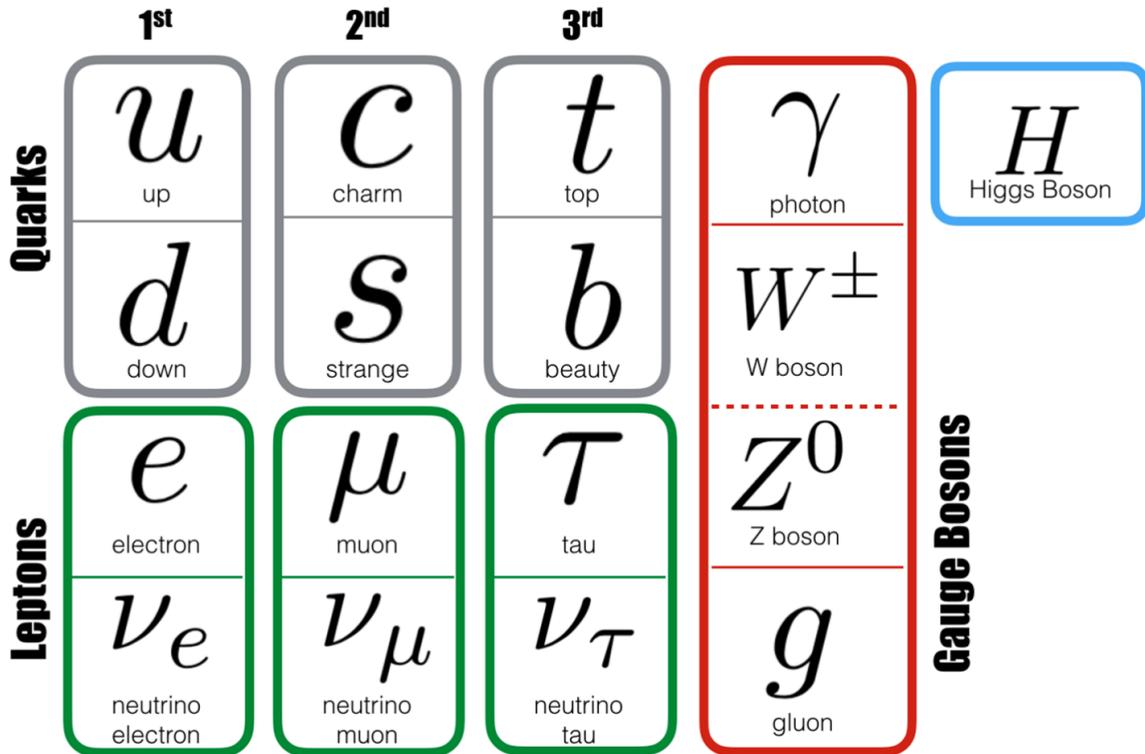


Figura 1.2: Las partículas elementales del Modelo Estándar [4].

de baja energía, como se le conoce hoy, consiste sólo de partículas de la primera generación [11].

1.1.1. Interacciones y partículas

En el Universo existen 4 fuerzas fundamentales mediante las cuales las partículas interactúan entre si: *fuerte*, *electromagnética*, *débil* y *gravitacional* [12]. Estas fuerzas son enlistadas en la tabla 1.1.

Interacción	Partículas afectadas	Rango de interacción	Fuerza (relativa a la fuerte)
Fuerte	<i>Quarks</i> y Hadrones	$\sim 10^{-15}m$	1
Electromagnética	Partículas con carga eléctrica	∞	$\sim 10^{-2}$
Débil	<i>Quarks</i> y Leptones	$\sim 10^{-18}m$	$\sim 10^{-5}$
Gravitacional	Todas	∞	$\sim 10^{-39}$

Tabla 1.1: Las cuatro interacciones fundamentales en física [13]

Cada una de estas fuerzas es mediada por el intercambio de partículas. La fuerza fuerte es mediada por ocho tipos de gluones, la fuerza electromagnética es mediada por los fotones, mientras que las interacciones débiles son mediadas por los bosones W^\pm y Z . La gravedad es mediada por el gravitón, sin embargo la mayoría de los científicos creen que la gravedad es simplemente muy débil para jugar un papel significativo en la física de partículas elementales [12].

1.2. Cromodinámica Cuántica

La Cromodinámica Cuántica (QCD -por sus siglas en inglés-) es la teoría de las interacciones fuertes, que junto con el Modelo Estándar (ME), describe los bloques de construcción de las partículas que interactúan fuertemente (como protones y neutrones) y las fuerzas que actúan entre ellas [14]. Los constituyentes fundamentales de estas partículas son los *quarks*, cuyo espín es $\frac{1}{2}$.

En general, las partículas formadas por *quarks* reciben el nombre de hadrones. En particular si una partícula está compuesta por 2 *quarks* la llamamos mesón, mientras que si está compuesta por 3 *quarks* la llamamos barión.

Carga de color

Un problema al considerar a los bariones, es la presencia de tres *quarks* del mismo sabor en una sola partícula (por ejemplo el barión Ω , el cual contiene 3 *quarks* s) pues se podría pensar que se viola el principio de exclusión de Pauli. Para solventar este problema fue sugerida la existencia de una propiedad adicional de los *quarks*, la cual debía tener un total de 6 valores distintos. Esta es la **carga de color** (o simplemente **color**), con 3 valores posibles para los *quarks* (rojo, verde y azul) y sus tres respectivos anti-valores para los anti-*quarks* (antirrojo, antiverde y antiazul) [13].

Esto resuelve el problema del barión Ω , pues ahora tenemos además de la restricción respecto a los espines, la restricción de color. Finalmente, para que esta restricción de color quede completa debemos pedir que los bariones tengan una carga de color neutra (color blanco) y esto se consigue de dos maneras distintas. Para los bariones

(compuestos de 3 *quarks*) debemos tener 1 *quark* de cada color, así, la combinación rojo-azul-verde resulta en color blanco (anti-rojo, anti-azul y anti-verde resulta en color blanco). Y en el caso de los mesones (compuesto de un *quark* y un anti-*quark*) uno de los constituyentes debe de tener la carga de color inversa al otro (por ejemplo azul y anti-azul).

Principio de Confinamiento

Las leyes de la naturaleza nos impiden tener *quarks* aislados. Este hecho es conocido como **Principio de Confinamiento**. Este principio nos dice que los *quarks* no pueden aparecer solos ya que deben estar confinados dentro de un hadrón [14]. Es por esto que para estudiar a un *quark* de cierto sabor, se debe de estudiar a los hadrones que lo contienen y también a sus subproductos.

1.3. *Conseil Européen pour la Recherche Nucléaire* (CERN)

El Consejo Europeo de Investigación Nuclear (CERN por sus siglas en francés) es el mayor centro para la investigación en física de partículas. Fue concebido en la década de los 50's y desde entonces nuestro entendimiento de la materia va mucho más allá del núcleo y la principal área de investigación en el CERN es la física de partículas. Es por esto que generalmente es conocido como Laboratorio Europeo para Física de Partículas [15]. En términos generales, la misión principal de CERN es conocer de qué está hecho y cómo funciona el Cosmos [16].

El CERN es una organización europea de investigación que se encuentra en el cantón de Ginebra en la frontera Franco-Suiza y opera el más grande laboratorio de física de partículas en el mundo.

1.3.1. *Large Hadron Collider (LHC)*

Para estudiar la física de partículas, en el CERN se tiene al Gran Colisionador de Hadrones (LHC), un acelerador y colisionador de partículas que consta de dos anillos de 27 km de circunferencia y está construido dentro de un túnel. Las motivaciones para la construcción de este colisionador son principalmente el descubrimiento del bosón de Higgs y el estudio de eventos³ con una energía de colisión en el centro de masa de hasta 14 TeV. El LHC está diseñado como un colisionador protón-protón [17].

Para estudiar las colisiones producidas por el LHC, el CERN cuenta con 4 detectores principales: ALICE, ATLAS, CMS y LHCb. Los dos detectores más grandes, ATLAS y CMS son detectores de partículas diseñados para ver una amplia gama de partículas y fenómenos producidos en colisiones de alta energía[18]. Mientras que ALICE y LHCb son detectores especializados, por ejemplo LHCb se especializa en física del *quark* b.

1.3.2. **Portal de Datos Abiertos del CERN**

El 20 de Noviembre de 2014 el CERN lanza su Portal de Datos Abiertos (*Open Data Portal*) donde datos provenientes de colisiones producidas en los experimentos del LHC fueron puestos accesibles al público en general por primera vez en la historia [19].

Este portal difunde materiales de diversas investigaciones, incluido software y documentación complementaria que son necesarias para entender y analizar los datos que están siendo compartidos [20]. Actualmente cuenta con más de 2 petabytes de datos abiertos de física de partículas.

Este portal es accesible mediante cualquier navegador web a través de la página web:

<http://opendata.cern.ch/>

³En términos experimentales se llama **evento** a la información de todos los productos de una colisión de partículas.

En esta página podemos seleccionar el tipo de información que se desea explorar, así como centrarse en algún experimento en particular, como ALICE, ATLAS, CMS o LHCb.

1.4. Detector *Compact Muon Solenoid* (CMS)

El LHC es capaz de colisionar grupos de protones a una velocidad cercana a la de la luz, 40 millones de veces por segundo. Cuando esto sucede, parte de la energía de la colisión se convierte en masa y en partículas de vida corta, antes no observadas, que podrían dar pistas sobre cómo se comporta la naturaleza en un nivel fundamental.

El Solenoide Compacto de Muones (CMS) es un detector de propósito general diseñado para estudiar la física de colisiones protón-protón con una energía en el centro de masa de 14 TeV. Este detector está diseñado para medir la energía, el momento y la trayectoria de fotones, electrones, muones y otras partículas cargadas con una alta precisión [21].

En términos generales el CMS es un detector que consiste en dos calorímetros [1]. Las partículas electromagnéticas (electrones y fotones) son detenidas y medidas en el primero; las partículas hadrónicas son medidas en ambos calorímetros y son detenidas en el segundo. Un sistema de *tracking* mide las trayectorias de todas las partículas cargadas. Y finalmente, el sistema más externo detecta las partículas cargadas que cruzan ambos calorímetros, esto es, muones y anti-muones.

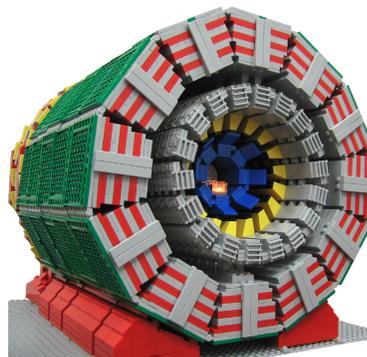


Figura 1.3: Modelo a escala 1:200 del detector CMS construido con piezas LEGO®[5].

1.4.1. Convención de coordenadas

Para describir los eventos detectados y reconstruidos por CMS, se utiliza un sistema coordinado (ver figura 1.4) que tiene el origen centrado en el punto de colisión nominal, dentro del experimento [22]. El eje x apuntando paralelo a la horizontal y

en dirección hacia el centro del LHC. El eje y apuntando en dirección a la vertical y el eje z en dirección del haz, apuntando en dirección hacia las montañas *Jura*. El ángulo azimutal ϕ es medido desde el eje x y se encuentra contenido en el plano $x-y$. El ángulo polar θ se mide a partir del eje z .

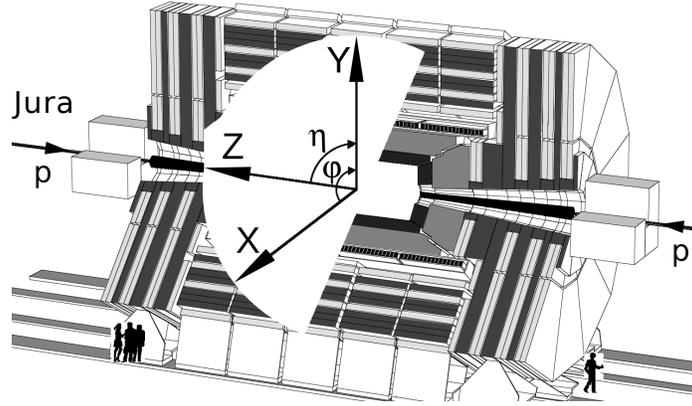


Figura 1.4: Sistema coordenado en el experimento CMS [6].

A partir del momento es posible obtener el **momento transverso** (\mathbf{p}_T), el cual se define como la cantidad de momento que tiene cualquier partícula resultante en dirección perpendicular al eje de colisión. Es decir, la proyección del momento en el plano $x-y$.

Por otra parte, es posible obtener una coordenada espacial llamada *pseudo-rapidez*, η , la cual se define como un ángulo de movimiento, para alguna partícula resultante, con respecto al eje de colisión [23]:

$$\eta \equiv -\ln \left[\tan \left(\frac{\theta}{2} \right) \right]$$

donde θ es el ángulo polar del sistema coordenado esférico.

De manera equivalente, es posible escribir esta ecuación en términos del momento total (\vec{p}) de la partícula generada:

$$\eta = \frac{1}{2} \ln \left(\frac{p_{\parallel}}{|\vec{p}|} \right)$$

donde p_{\parallel} es la proyección del momento total de la partícula en dirección paralela al eje de colisión.

1.4.2. Configuración del CMS

La configuración del CMS se explica a continuación [22].

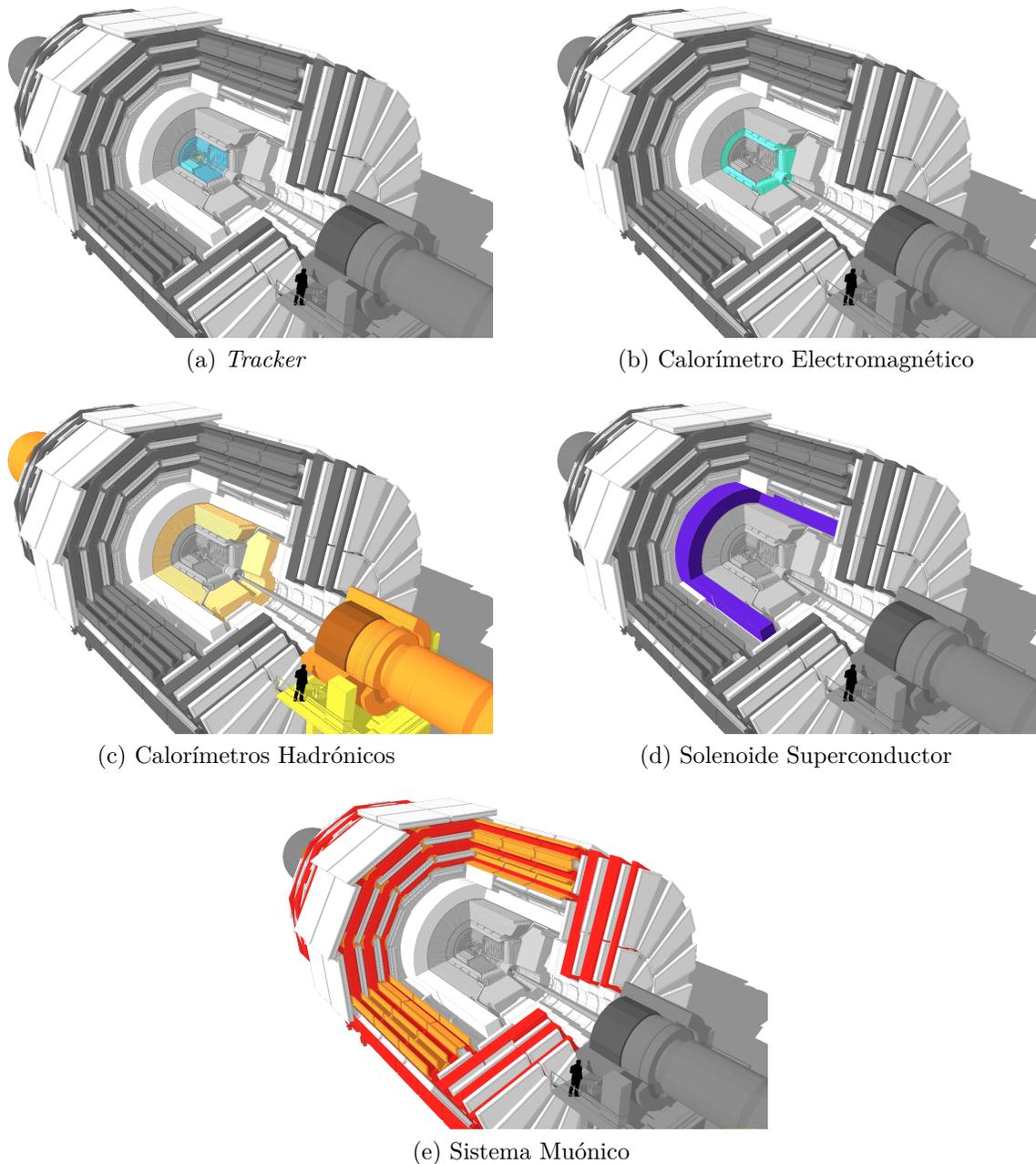


Figura 1.5: Distribución de los sub-detectores dentro de CMS [7].

En el corazón del CMS se sitúa un superconductor solenoidal de 13 m de largo y 5.9 m de diámetro interno (figura 1.5d), el cual produce un campo magnético de 3.8 Teslas. El campo magnético fuera del solenoide es capaz de saturar 1.5 m de hierro,

lo cual permite integrar 4 *estaciones* de muones para asegurar una buena y completa cobertura geométrica. Dentro del solenoide se encuentra el *tracker* y la mayoría de los calorímetros.

El sistema de *tracking* (figura 1.5a) ocupa un espacio cilíndrico de 5.8 m y un diámetro de 2.6 m. El propósito de este detector es proveer medición de puntos de alta precisión en tres dimensiones a lo largo de las trayectorias curvas de las partículas cargadas hasta una pseudorapidez de $|\eta| < 2.5$. Sin embargo, la mejor eficiencia del *tracker* se encuentra en la región $0 < |\eta| < 0.9$ [24]. Para lidiar con una multiplicidad alta en las detecciones, el CMS utiliza 10 capas de detectores microscópicos de silicio, los cuales proveen la segmentación y precisión necesaria.

El Calorímetro Electromagnético (ECAL) es un calorímetro hermético y homogéneo, compuesto por cristales de plomo-tungsteno ($PbWO_4$) y cuenta con una cobertura en la pseudorapidez de hasta $|\eta| < 3.0$ (figura 1.5b). Este calorímetro consta de dos partes, la primera es el *Electromagnetic Barrel* (EB) el cual tiene una forma cilíndrica con un radio interno de 129 cm y cubre un intervalo de pseudorapidez de $0 < |\eta| < 1.479$. La segunda parte son los *Electromagnetic Endcap* (EE) los cuales tienen forma de disco, se sitúan a una distancia de 314 cm del vértice, en dirección z , y cubren un rango de pseudorapidez de $1.479 < |\eta| < 3.0$.

El Calorímetro Hadrónico (HCAL), al igual que la mayoría de los calorímetros en el CMS, se encuentra localizado dentro del solenoide magnético, y además rodea al ECAL. Este calorímetro está compuesto por 4 partes (figura 1.5c). La primera de ellas es el *Hadron Barrel* (en color anaranjado claro en la figura 1.5c) el cual cubre una región de pseudorapidez de $0 < |\eta| < 1.4$ y tiene una segmentación de $\Delta\eta \times \Delta\phi = 0.087 \times 0.087$. Por otro lado tenemos los *Hadron Endcap* (HE) que se encuentran en cada uno de los extremos del HB y cubren una región de $1.3 < |\eta| < 3.0$ (en color anaranjado claro en la figura 1.5c). Después, tenemos el *Hadron Outer* (HO) el cual se encuentra dentro del sistema muónico y por lo tanto constreñido por la geometría y construcción de dicho sistema. El HO cubre una región de $0 < |\eta| < 1.26$ y ayuda a mejorar la resolución y la medición de la E_T^{miss} en el HCAL (se muestra color anaranjado en la figura 1.5e). Finalmente, tenemos el calorímetro Hadrónico Frontal

(HF) que se encuentra a una distancia, en dirección del haz, de 11.2 m del centro del detector y se sitúa a cada lado (en color rojizo en la figura 1.5c). Ningún *tracker* está en frente del HF, así que no hay información para determinar si las partículas que detecta son cargadas o neutras. Sin embargo es capaz de distinguir si las partículas detectadas son hadrónicas o electromagnéticas [25]. Su principal función es ampliar el rango de detección en pseudorapidez, pues cubren una región de $3.0 < |\eta| < 5.0$.

El sistema detector de muones en el CMS (de color rojo en la figura 1.5e) tiene 3 funciones principales: *triggering*, identificación y medición de momento. Este sistema cuenta con una región de *barrel* y dos *endcaps*, y se encuentran fuera del solenoide magnético. El *barrel* cubre un región de $0 < |\eta| < 1.2$ y el sistema muónico completo cubre hasta $|\eta| < 2.4$ [26].

1.4.3. Eventos en CMS

Un “evento” en el registro de datos del CMS se refiere a la información reconstruida y procesada acerca de los productos de una colisión de partículas (en esta tesis colisión protón-protón). Estos eventos son reconstruidos usando el algoritmo *Particle Flow*⁴, el cual reconstruye e identifica partículas individuales mediante una combinación optimizada de toda la información de los sub-detectores [27].

1.4.4. Sistema de *trigger* de CMS

En el LHC colisionan conjuntos de protones cada 25 ns con un máximo de 60 interacciones protón-protón por conjunto [28]. Teniendo así del orden de $\sim 10^8$ colisiones ¡por segundo! Sin embargo, el detector CMS solamente es capaz de almacenar $\sim 10^2$ eventos completos cada segundo [29], por lo que se hace evidente que necesitamos un sistema de selección de los mismos.

En esencia, un *trigger* es un mecanismo que selecciona rápidamente eventos para ser almacenados para un análisis posterior [28]. CMS tiene un sistema de trigger de dos niveles [30]:

⁴Ver la sección 1.6, en donde se discute a detalle este algoritmo.

- *Trigger* de Nivel-1: Simple y de corta duración. Compuesto por un sistema de electrónica diseñado especialmente para este detector.
- *High Level Trigger* (HLT): Sofisticado y requiere significativamente más tiempo para ser ejecutado. Es un sistema de software el cual realiza labores de reconstrucción y filtrado. En términos de hardware, está conformado por PC's comerciales.

Para que un evento sea almacenado por CMS, éste tiene que ser aceptado por *Trigger* de Nivel-1 y necesita satisfacer por lo menos una de las selecciones de HLT.

1.4.5. Convención de unidades para software

El código de software de CMS utiliza las siguientes convenciones [31]:

La Energía es medida en unidades de GeV , de esta manera el momento y la masa son dadas en $\frac{GeV}{c}$ y $\frac{GeV}{c^2}$, respectivamente. La distancia y posiciones están en centímetros (cm), mientras que el tiempo está denotado en nanosegundos (ns). El campo magnético es medido en Teslas (T) y su orientación coordenada es tal que el campo magnético solenoidal está a lo largo del eje z . La carga eléctrica es medida en unidades de carga elemental $|e|$.

1.5. *Open-Data* de CMS

Al ser CMS un detector en el CERN, se ha puesto para el acceso libre distintos formatos de datos que contienen diversos grados de detalles, tamaño y refinamiento para su uso en múltiples escenarios. A su vez, estos datos se agrupan dentro de archivos con múltiples formatos de eventos, de acuerdo con el origen o contenido de estos.

En el portal de datos abiertos del CMS podemos encontrar conjuntos de datos, documentación, software, herramientas de visualización de eventos, entre otros.

1.5.1. Jerarquía de los datos de CMS

Los datos de CMS se organizan en una jerarquía de niveles de datos. Cada evento de física se escribe en cada nivel de datos, donde los niveles contienen un tipo diferente de información sobre el evento. Los tres niveles principales para datos de CMS son [32]:

1. RAW: Contiene información *cruda* proveniente del detector (señales en los elementos del detector). Este tipo de datos no es utilizado para hacer análisis.
2. RECO: Obtiene su nombre de *reconstructed data*, es el primer paso de procesamiento de datos. Este nivel contiene objetos de física (productos de decaimiento como muones, fotones, electrones, hadrones, jets, etc.) reconstruidos pero la información que provee sigue siendo muy detallada. Este tipo de datos puede ser utilizado para análisis pero es demasiado grande, o pesado, para su uso frecuente cuando CMS ha almacenado una muestra de datos sustancial.
3. AOD: El *Analysis Object Data* es la versión “destilada” de RECO y se espera sea usada para análisis. AOD provee un equilibrio entre tamaño de evento y complejidad de la información disponible, esto con el fin de optimizar la flexibilidad y velocidad de los análisis.

1.5.2. *Analysis Object Data* (AOD)

En este trabajo serán utilizados únicamente los datos AOD, por lo que vale la pena ahondar un poco más en ellos, discutir su estructura y su contenido.

Como ya se ha mencionado, los datos AOD son derivados de la información RECO y proveen información para el análisis físico en un formato conveniente y compacto. Estos datos están listos para su uso directo en el análisis. Los archivos AOD contienen información a cerca de vértices, *tracks*, partículas reconstruidas (muones, fotones, electrones, etcétera), jets y *clusters* de los calorímetros [33]. Estos archivos no pueden ser entendidos como simples tablas de datos, pues requieren Root⁵ para ser leídos

⁵Ver sección 1.11 para obtener más información sobre Root.

[34]. En otras palabras, estos archivos vienen en formato `.root`.

La estructura de estos archivos se basa en árboles, ramas y hojas [35]. Los árboles están optimizados para reducir espacio en el disco y mejorar la velocidad de acceso. Los árboles son capaces de manejar todo tipo de datos (variables), como objetos o arreglos. La organización en ramas permite la optimización para el uso posterior, de este modo, si hay dos variables relevantes para un análisis, la manera más eficiente para ser leídas es ponerlas juntas en una misma rama. Finalmente, las variables dentro de una rama son llamadas hojas. Una descripción visual de la estructura de los AODs, se encuentra el apéndice A.1.

1.5.3. Conjuntos de AODs utilizados

Los eventos adquiridos por el CMS son organizados en conjuntos de datos primarios, de acuerdo con los resultado del HLT (descrito en la sub-sección 1.4.4). En la página de datos abiertos del CERN [20] podemos encontrar diferentes tipos de AODs los cuales se especializan en distintos tipos de información.

Al momento de comenzar este trabajo únicamente se encontraban disponibles datos de los años 2010, 2011 y 2012. En este trabajo se considerarán los AODs *Jet* [39], *B-Tagging (BTag)* [40], *Minimum Bias (MinBias)* [41] y *Multiple Jet (MultiJet)* [42]. La tabla 1.2 provee información del *trigger* utilizado y la física de los eventos en cada caso. Los AODs *Jet* y *MultiJet* se enfocan específicamente en jets y su comportamiento, y únicamente se encuentran disponibles en los conjuntos de datos correspondientes a 2010 y 2011. *B-Tag* ofrece información de la física generada por el *quark* b y en particular de jets generados por el *quark* b . Por otro lado, tenemos el conjunto *MinBias* que servirá como referencia a procesos generados por colisiones inelásticas, es decir, que no se enfocan en jets. Estos dos últimos conjuntos de datos se encuentran disponibles para los 3 años anteriormente mencionados. Debido a que la cantidad de *data* disponible para 2010 resulta ser insuficiente para mis fines, decidí utilizar datos correspondientes a 2011 y en este año sólo se encuentran disponibles colisiones llevadas a cabo con una energía en el centro de masa de 7 GeV.

AOD	<i>Trigger</i>	Evento
Jet	A partir del <i>Trigger</i> de Nivel-1, basado en la técnica llamada ventana deslizante [9] (ver apéndice A.2).	Jet generado por la hadronización de cualquier partón.
BTag	Explota el hecho de que los hadrones <i>b</i> típicamente tienen mayores vidas-medias en comparación con los hadrones compuestos por <i>quarks</i> más ligeros. En consecuencia, las trayectorias y los vértices de sus productos de decaimiento están significativamente desplazados del vértice primario. Del mismo modo, los hadrones <i>b</i> decaen más frecuentemente a estados finales con leptones que sus contrapartes ligeras [9].	Jets que provienen de la hadronización de un <i>quark</i> <i>b</i> [36].
MinBias	Se lleva a cabo utilizando el Calorímetro Hadrónico Frontal (HF). El trigger se dispara cuando un mínimo de 10 torres detectan energías mayores a 1.4 GeV [37].	Corresponde a una colisión inelástica [38].
MultiJet	El <i>trigger</i> se activa al detectar al menos 1 jet con $p_T > 250$ GeV en $ \eta < 1.3$, balanceado por un sistema de retroceso compuesto por dos o más jets con $p_T \in (25, 750)$ GeV [27].	Jets reconstruidos dada dicha configuración del <i>trigger</i> .

Tabla 1.2: Conjunto de datos primarios utilizados.

1.5.4. Máquina Virtual (MV)

En términos generales, una máquina virtual es un software que permite emular un sistema operativo dentro de un sistema operativo diferente. La MV de CMS [43] es la herramienta recomendada para el análisis de datos abiertos, y se debe de instalar utilizando VirtualBox, el cual es una plataforma para el uso de máquinas virtuales [44]. Esta máquina virtual está basada en Scientific Linux 6 e incluye Root y CMSSW. CMSSW se refiere al conjunto de herramientas de software utilizadas por CMS.

1.6. *Particle Flow* (PF)

Aunque no siempre es posible “observar” directamente a las partículas creadas en las colisiones, sus productos de decaimiento dejan señales en los sub-detectores de CMS, y posteriormente el *software* usa estas señales para reconstruir los productos de decaimiento. Es importante señalar que estos productos son sólo interpretaciones

de las señales observadas por CMS y, como tales, están sujetas a diversas fuentes de incertidumbre (eficiencias, identificaciones erróneas, por mencionar algunas) [2].

Para mejorar la descripción de cada evento, dentro del detector se reconstruyen las propiedades de las partículas al correlacionar la información proveniente de todos los sub-detectores. Esto se hace con el fin de identificar cada partícula de estado final. A este enfoque holístico se le llama reconstrucción *Particle-Flow* [45].

El detector CMS es especialmente capaz de llevar a cabo la reconstrucción PF. Esto es debido a que CMS cuenta con un *tracker* altamente segmentado, un calorímetro electromagnético de grano fino, un calorímetro hadrónico hermético, un campo magnético del orden de Teslas, y un excelente espectrómetro de muones [45]. Sin embargo CMS no es perfecto, por ejemplo, una parte del momento llevado por los hadrones neutros es reconstruida como si fuera proveniente de fotones debido a los depósitos de energía producidos por los hadrones en el ECAL [45].

El algoritmo *Particle Flow* tiene como objetivo reconstruir e identificar todas las partículas estables en un evento [2], es decir: muones, fotones, electrones, hadrones neutros y hadrones cargados. A cada una de las partículas se asocia con un 4-momento, la estimación de la energía depositada en los calorímetros y otros marcadores más especializados [46].

Es importante mencionar que PF no hace ningún intento por distinguir las diversas especies de hadrones neutros y cargados [45].

Algoritmo computacional PF

El algoritmo computacional PF consiste en los siguientes pasos [46]:

1. Obtener la información necesaria:
 - El *cluster* de los calorímetros.
 - Los *tracks*, incluido el del sistema muónico.
 - Pre-identificación de electrones.
2. La asociación de elementos topológicamente conectados (señales, de distintos sub-detectores, que están relacionadas).
3. Identificación y reconstrucción de partículas.

1.7. Funciones de Fragmentación (FFs)

Las Funciones de Fragmentación describen cómo los partones (*quarks* o gluones), con carga de color, se convierten en partículas sin color como hadrones [47] o fotones [48]. Estas funciones pueden ser utilizadas para estudiar y caracterizar el comportamiento de los jets. Para llevarlo a cabo esto, se considera una simulación y esta será comparada con los cuatro conjuntos de datos abiertos con los que se está trabajando.

En esta tesis se definirá una FF en términos de la energía de los constituyentes de un jet y la energía total del jet. Además, se estudiará el comportamiento de la FF con respecto a la relación entre el número de eventos simulados y el momento transversal mínimo (\hat{p}_T^{\min}).

1.8. Jets

Los jets son los chorros colimados de partículas que resultan de la fragmentación de un *quark*, o gluón, energético [49] en colisiones de partículas de alta energía. En este sentido, la reconstrucción de jets juega un papel muy importante puesto que permite obtener, de manera indirecta, información sobre los constituyentes más fundamentales de la materia hadrónica: los *quarks* y los gluones[50].

Para cualquier herramienta que sea utilizada para la reconstrucción de jets, su comportamiento debe de ser bien definido y reproducible: uno debería tener reglas que proyecten un conjunto de partículas dentro de un conjunto de jets. Dicho conjunto de reglas es referido como un **algoritmo de reconstrucción de jets**. Usualmente, un algoritmo de jets involucra uno o más parámetros que gobiernan su comportamiento detallado y la combinación de un algoritmo de jet y sus parámetros es conocida como una **definición de jet** [49].

El contenido de partículas de los jets, en términos de tipo de partícula y distribución de energía, puede ser descrito mediante las funciones de fragmentación y depende del sabor del partón que inicializa el jet. En promedio, 65 % de la energía del jet es

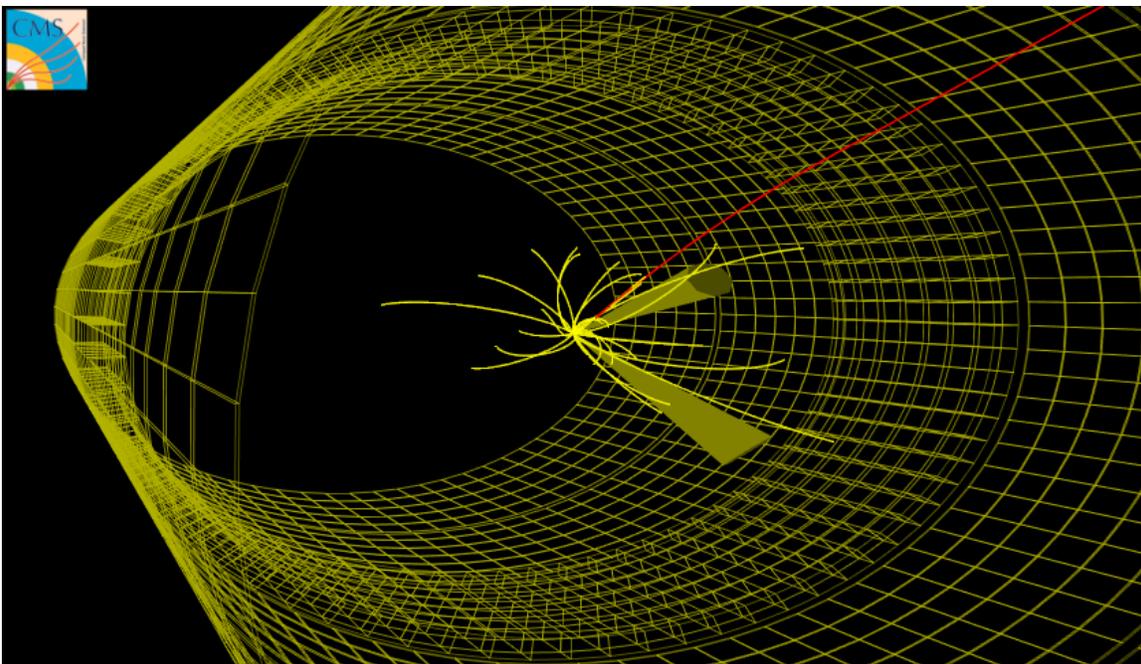


Figura 1.6: Visualización de un evento *MinBias* en CMS utilizando herramientas del portal de datos abiertos del CERN [8]. En la figura se muestran las trayectorias de las partículas cargadas (líneas amarillas), un muón (línea roja) y dos jets (conos amarillos).

llevada por partículas con carga eléctrica, 25 % por fotones y 10 % por hadrones neutros [45]. Y en promedio, los jets de gluones presentan más partículas de baja energía que los jets de *quarks* [51].

1.8.1. Fragmentación de jet

Consideremos un *quark* q y un anti-*quark* \bar{q} alejándose el uno del otro en el sistema de referencia del centro de masa (figura 1.7). A diferencia del electromagnetismo, las líneas del campo de color no se esparcirán en todo el espacio sino que estarán constreñidas a una estrecha región tubular. Una vez que q y \bar{q} se separen aproximadamente 1 fm (10^{-15} m), la energía potencial aumenta tanto que uno o más pares $q\bar{q}$ son creados. Entonces, el sistema original se rompe en piezas cada vez más pequeñas, hasta que solamente quedan hadrones ordinarios. Algunos de estos llamados hadrones pueden ser inestables y decaer en aún más hadrones, leptones y fotones; que son, de hecho, los observados. Eventualmente, toda la energía es degradada en dos jets moviéndose más o menos a lo largo del eje $q\bar{q}$ original [52], [53].

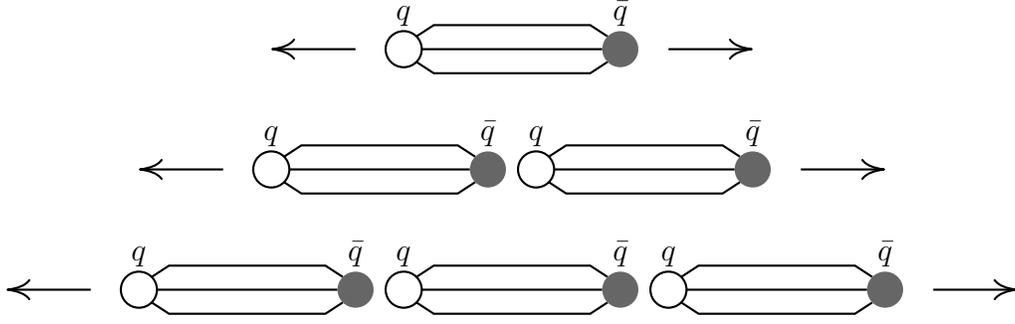


Figura 1.7: Formación de jets a partir de la separación de un *quark* y un anti-*quark*.

Debido al confinamiento de color nunca se tendrá a un jet originado por un solo *quark* apareciendo en el vacío.

1.8.2. Algoritmos de reconstrucción secuencial de jets

Se introducen las distancias d_{ij} y d_{iB} . Donde d_{ij} representa la distancia entre las entradas (partículas - **pseudojets**⁶-) i y j ; d_{iB} es la distancia en el espacio de momentos entre la entrada i y el haz⁷ B [10] y [54].

$$d_{ij} = \min [k_{ti}^{2p}, k_{tj}^{2p}] \left(\frac{\Delta_{ij}}{R^2} \right)^2 \quad (1.1)$$

$$d_{iB} = k_{ti}^{2p} \quad (1.2)$$

en estas ecuaciones k_{tn} (con $n \in i, j$) se refiere al momento transverso de la entrada n , $\Delta_{ij} = \sqrt{(\eta_i - \eta_j)^2 + (\phi_i - \phi_j)^2}$ corresponde a la distancia en el espacio $\eta - \phi$ entre las entradas i y j , y R es el radio del jet, que es un parámetro escalar. El parámetro p gobierna la potencia relativa de la energía contra las escalas geométricas (Δ_{ij}).

El algoritmo de reconstrucción procede a identificar la menor de las distancias, si está es d_{ij} entonces las entradas i y j son re combinadas, por el otro lado si es d_{iB} entonces llamamos a i como un jet y lo quitamos de la lista de entradas. Finalmente, estas distancias son recalculadas y el procedimiento repetido hasta que todas las

⁶En términos computacionales, un **pseudojet** es un conjunto de partículas que son consideradas para formar jets.

⁷En este caso nos referimos al haz de partículas colisionantes.

partículas sean parte de algún jet y la distancia entre los ejes de todos los jet sea mayor a R (reconstrucción inclusiva). O hasta que un número deseado de jets ha sido encontrado (reconstrucción exclusiva).

Algoritmo kt

Considerando la ecuación 1.1 con el parámetro $p = 1$ obtenemos el algoritmo kt .

$$d_{ij} = \min [k_{ti}^2, k_{tj}^2] \left(\frac{\Delta_{ij}}{R^2} \right)^2 \quad (1.3)$$

Sin embargo, en términos generales para cualquier $p > 0$ obtendremos el mismo comportamiento porque lo que importa es el ordenamiento entre las entradas, y para todo Δ_{ij} finito, este ordenamiento es mantenido para todo valor positivo de p .

Como podemos ver en la ecuación (1.3), en el parámetro $\min [k_{ti}^2, k_{tj}^2]$ tenemos preferencia para momentos transversos bajos. Por lo tanto, el algoritmo kt prefiere agrupar primero las partículas de bajo momento transverso en una región que fluctúa considerablemente al inicio.

Algoritmo $anti-kt$ (ak)

Tomando $p = -1$ en la ecuación (1.1), obtenemos el algoritmo ak mostrado explícitamente en la ecuación (1.4).

$$d_{ij} = \min \left[\frac{1}{k_{ti}^2}, \frac{1}{k_{tj}^2} \right] \left(\frac{\Delta_{ij}}{R^2} \right)^2 \quad (1.4)$$

Cabe mencionar que, análogamente al caso de kt , cualquier $p < 0$ se produciría el mismo comportamiento.

En general, este algoritmo actúa de tal manera que las partículas más energéticas “absorben” a las menos energéticas. Esto lo podemos ver si consideramos el parámetro $\min \left[\frac{1}{k_{ti}^2}, \frac{1}{k_{tj}^2} \right]$, pues es de notar que existe preferencia para momentos transversos altos. Esto da como resultado una preferencia en agrupar partículas de bajo momento transverso con partículas de alto momento transverso, siempre y cuando las escalas geométricas Δ_{ij} y R lo permitan.

1.8.3. Tipos de jets

Cuando se clasifican los jets se deben de tomar en cuenta lo siguiente [55]:

- **El algoritmo de reconstrucción de jets:** Estos algoritmos hacen referencia a los criterios utilizados para asociar las partículas a los jets. Dos ejemplos de estos algoritmos son *kt* y *ak*.
- **El radio del jet.** Es decir, el parámetro de distancia utilizado para agrupar las partículas dentro de un mismo jet, este parámetro está medido en términos de las coordenadas angulares ϕ y η . De tal modo que $R = \sqrt{\Delta\phi + \Delta\eta}$.
- **El tipo de partículas utilizadas.** Por ejemplo, PF es un algoritmo utilizado para reconstruir las partículas en un detector pero no es el único, otro algoritmo es el *Calo*, el cual utiliza únicamente información proveniente de los calorímetros. Otro ejemplo son las partículas generadas mediante simulaciones.

De este modo, `kt6PFJet` es un jet reconstruido con el algoritmo *kt* cuyo parámetro de distancia de $R = 0.6$, y que utiliza partículas reconstruidas con el algoritmo PF.

1.9. Pythia

Pythia es un programa, basado en el método *Monte Carlo*^{8,9} (MC), para la generación de eventos de física de alta energía. Es decir, para la descripción de colisiones a altas energías entre partículas elementales tales como e^+ , e^- , p y \bar{p} en diversas combinaciones [58].

Este programa computacional es de suma utilidad pues nos permite conocer las predicciones de los modelos actuales que rigen la descripción de las interacciones en la física de partículas. Además de que es posible estudiar los efectos y resultados en diferentes tipos de colisiones entre partículas en amplios rangos de energías.

⁸Pese a no haber una definición oficial del método *Monte Carlo*, podemos expresarlo de la siguiente manera. “En lugar de llevar a cabo largos cálculos complejos, mejor llevar a cabo un gran número de experimentos usando un generador de números aleatorios” [56].

⁹Debido a que el mundo subatómico es gobernado por la mecánica cuántica, la cual es una teoría probabilística, únicamente podemos calcular la probabilidad de un cierto resultado. Por ende, el generador de eventos es también probabilístico y por ello el nombre de Monte Carlo [57].

Pese a ser Pythia una poderosa herramienta computacional, lo cierto es que no es posible hacer un análisis de datos apropiado únicamente con ella, pues los resultados (histogramas y gráficas) son mostrados a través de la terminal, obteniendo así una visualización pobre y rudimentaria.

1.9.1. Eventos en Pythia

Para Pythia, y en general para cualquier simulador de colisiones de partículas de altas energías, un evento es la simulación de una sola colisión (en esta tesis una colisión protón-protón) [57].

El “Registro de Eventos” de Pythia almacena toda la información referente a un evento. Este registro está configurado para almacenar cada paso en la evolución de un proceso con baja multiplicidad (número de partículas) inicial (dos partículas colisionantes), hasta un estado final de alta multiplicidad (todos los productos finales de la colisión) [59]. Este registro es un vector de partículas que se expande para ajustarse a las necesidades del evento actual. De este modo, el `Event[n]` es la n -ésima partícula del evento actual y es posible estudiar sus propiedades utilizando los distintos métodos disponibles (en términos técnicos: `Event[n].methods()`).

1.9.2. Convención de unidades

Como era de esperarse, Pythia tiene una convención interna para las unidades utilizadas en la medición de las magnitudes físicas. Dicha convención se enlista a continuación [60]:

- **Giga electron volts (GeV):** Para todas las energías, momentos y masas.
- **Milímetros (mm):** Para todas las distancias y tiempos.
- **Milibarns (mb):** Para todas las secciones transversales. Un barn es igual a 10^{-28} m^2 .

Esta convención considera que la velocidad de la luz (c) es igual a la unidad, $c = 1$.

1.9.3. Configuración de una simulación

Para generar una simulación se necesitan definir algunos parámetros:

- **Energía en el centro de masa** (en el código `Beams:eCM`): Define la energía en el centro de masa de la colisión. Para este trabajo se realizarán colisiones a 7 TeV, la misma energía de los datos abiertos.
- **Activar QCD Dura** (en el código `HardQCD:all=on`): Activa los procesos duros, es decir, dos partones interactúan y entre ellos hay una transferencia de momento transversal mayor a 2 GeV.
- \hat{p}_T^{\min} (en el código `pTHatMin`): El momento transversal mínimo lo utilizamos para forzar una mayor transferencia de momento transversal entre partones. Es decir, aumentamos el valor de 2 GeV del inciso anterior hasta un valor máximo de 7 TeV utilizado en esta tesis.
- **Número de Eventos**: Esto se refiere a la cantidad de colisiones protón-protón que se desea simular. Se define dentro del código como un número entero mayor a 0.
- **Partículas colisionantes**: Pythia (versión 8240) realiza colisiones protón-protón por *default*. Por lo tanto no hace falta especificar el tipo de partículas colisionantes en este trabajo.

Estos son los parámetros necesarios para hacer una simulación en Pythia.

1.10. FastJet

FastJet es una paquetería que proporciona implementaciones rápidas de algoritmos de reconstrucción de jets basados en recombinación secuencial, en particular, incluye los algoritmos *kt* y *ak*. Esta reconstrucción de jets se basa en los métodos geométricos descritos en la sección 1.8.2 y discutidos más a detalle en [61], [62] y [63].

1.10.1. Parámetros de reconstrucción de jets

FastJet es una paquetería de C++ que proporciona una amplia gama de herramientas de análisis y reconstrucción de jets [49]. Por tanto, para reconstruir jets que sean geoméricamente iguales a los que se tienen en los datos abiertos, es necesario definir algunos parámetros.

- **Algoritmo de Reconstrucción de Jets (ARJ):** En este caso se utilizarán algoritmos de reconstrucción secuencial de jets (en el código se define como `fastjet::genkt_algorithm`), a saber, se utilizarán los algoritmos *kt* y *ak*.
- **Radio del jet:** Dependiendo del caso, utilizaremos radios de 0.4, 0.5, 0.6 y 0.7.
- **p_T mínimo de jets:** Esto define el p_T mínimo necesario para que los jets y su contenido sea considerado para el análisis. Este valor fue tomado a partir de los datos abiertos y es de $3 \text{ GeV}/c$.

1.11. Root

Root es un marco de trabajo orientado a objetos y dirigido a resolver los retos de análisis de datos en física de altas energías [35]. Proporciona las funcionalidades necesarias para tratar el procesamiento de *big data*, y el análisis estadístico y la visualización de datos [64]. Por este motivo, en este trabajo se realizará el análisis de datos mediante el programa Root [65], el cual además permite conectarlo con Pythia¹⁰ para tener una salida directa de la información proveniente de las simulaciones en formato `.root`.

¹⁰Ver apéndice B.5 para obtener más detalles sobre esta conexión.

Capítulo 2

Metodología

En este capítulo se explica el proceso completo de análisis de los datos reales (abiertos) del CMS, así como el de la simulación de colisiones hecha con Pythia.

Para esta tesis, obviando la conexión a internet, se han utilizado como únicas herramientas de cómputo un disco duro externo de 10 TB y una laptop con un procesador Intel Core i5-3210M, tarjeta gráfica AMD Turks de 1 GB, memoria RAM de 8 GB y SSD de 240 GB; cuyo sistema operativo es Ubuntu 18.04 en la versión de 64 bits.

Los tiempos de cómputo necesarios para llevar completamente a cabo el análisis de datos y la simulación son aproximadamente de 30 y 60 horas, respectivamente.

2.1. Análisis

En esta sección se explica la metodología desarrollada para trabajar con los datos abiertos del CMS. Del mismo modo, se da un ejemplo del *macro* de análisis utilizado y se detallará paso a paso su funcionalidad.

La página de datos abiertos del CERN [20] cuenta con una documentación muy amplia y con diversos tutoriales. Sin embargo, carece de la información necesaria para el uso específico que se le dará a los datos, por lo que se desarrollan de manera independiente herramientas útiles para este trabajo.

Para el análisis de datos, CMS recomienda utilizar una máquina virtual (MV) para cada generación de datos [43], es decir, para cada año de toma de datos existe una máquina virtual distinta. Esto tiene ventajas y desventajas. Una de las principales ventajas es tener disponible todas las herramientas para el análisis, tales como todas las paqueterías y el *software* necesario. No obstante, la principal desventaja es que al ser una máquina virtual, es incapaz de utilizar todos los recursos de la computadora en donde ha sido instalada. Esto resulta en tiempos de cómputo mucho mayores y como se trabajará con un total de 9.8 TB de datos reales (abiertos), esto es especialmente importante. Finalmente, cabe mencionar que las herramientas de software contenidas en la MV fueron desarrolladas a la par que la toma de datos, *i. e.*, antes del año 2013) y en algunos casos pueden resultar anticuadas.

Así que para esta tesis se ha decidido instalar todas las herramientas necesarias directamente en el Sistema Operativo. Con esto mejoran los tiempos de cómputo hasta ser capaz de analizar los 9.8 TB en 24 horas.

La MV se emplea únicamente para la descarga de archivos AOD.

2.1.1. Descripción de la descarga de archivos AOD

Si se utiliza la MV, es posible acceder a los archivos AOD de manera remota. Tal como se realiza en [34] o en [66]. Sin embargo, el simple acceso a los archivos resulta poco práctico por el considerable tiempo requerido. Fuera del entorno de MV no es posible acceder a los archivos AOD de manera remota ya que se carecen de los permisos y *software* necesarios.

Dicho lo anterior, resulta evidente que es necesario descargar los AODs para poder trabajar con ellos de manera continua. La descarga de los AODs se puede hacer de dos maneras distintas las cuales se mencionan a continuación:

- Explorando el sitio web de datos abiertos del CERN [20], navegar en los conjuntos de datos disponibles y descargar directamente (esto se puede hacer desde cualquier navegador web). Este camino resulta inviable si se desea descargar grandes conjuntos de datos del orden de TB.

- Descargar los archivos directamente utilizando la MV con el servicio `Xrootd`. Este es el método aquí utilizado y el recomendado para descargar múltiples archivos.

Descripción de la metodología de descarga

Lo primero que se necesita es descargar e instalar la MV de CMS para el conjunto de datos abiertos correspondientes al año 2011, este procedimiento es descrito por CERN en [43]. En este trabajo se utilizó la versión *CMS-Open-Data-1.3.0*, esta MV está basada en Scientific Linux 6. Posterior a eso debemos seguir los siguientes pasos:

1. Acceder a la MV del CMS.
2. Abrir una terminal.
3. Movernos al Directorio de Trabajo (`/home/cms-opendata/CMSSW_5_3_32/src`):

```
cd /home/cms-opendata/CMSSW_5_3_32/src
```

4. Cargar el entorno de *software* de CMS:

```
cmsenv
```

5. Descargar el AOD deseado con el siguiente comando:

```
xrdcp DirecciónDeDescarga/NombreDeLAOD.root  
↔ RutaLocal/NombreLocal.root
```

Podemos encontrar toda la colección de datos abiertos en [20] y accediendo a cada tipo de dato, es posible conocer la dirección de descarga y el nombre correspondiente a cada AOD.

Descargas Múltiples

En cada tipo de datos podemos encontrar hasta del orden de 2000 archivos, lo que hace que sea muy complicada obtener toda la colección si se realizan descargas individuales. Por lo tanto vale la pena describir una manera de descargar múltiples archivos de manera sencilla y eficiente.

Dado que la descarga se realiza desde la terminal y podemos obtener la dirección de descarga de todos los archivos directamente de la página de datos abiertos, [20], la forma más fácil de realizar múltiples descargas es haciendo listas de descarga dentro de un archivo en formato `.bash`. Estas listas son básicamente el paso 5 descrito arriba pero con la diferencia de que a cada archivo le corresponde una dirección de descarga distinta, un nombre distinto y debe de tener un nombre distinto al ser almacenada localmente, de tal modo que esta lista tendría la siguiente forma:

```
1  !/bin/bash
2
3  xrdcp -v -f Dirección1/Nombre1.root RutaLocal/NombreLocal1.root
4  xrdcp -v -f Dirección2/Nombre2.root RutaLocal/NombreLocal2.root
5  xrdcp -v -f Dirección3/Nombre3.root RutaLocal/NombreLocal3.root
```

Lo que encontramos en la página web es la dirección de descarga y el nombre de cada archivo. Nosotros debemos de asignar la ruta local de almacenamiento del archivo, así como su nombre local. Las opciones adicionales `-v` y `-f` sirven para mostrar un resumen sobre las descargas y para sobrescribir archivos ya existentes, respectivamente.

Finalmente, si a nuestro *macro* lo llamamos `ListaDescarga.bash` y lo ubicamos en el directorio de trabajo (`/home/cms-opendata/CMSSW_5_3_32/src`) entonces para ejecutarlo basta con escribir la siguiente orden en la terminal:

```
bash /home/cms-opendata/CMSSW_5_3_32/src/ListaDescarga.bash
```

Una muestra real de este *macro* es el siguiente:

```
1  !/bin/bash
2
3  xrdcp -v -f
   ↪ root://eospublic.cern.ch//eos/opendata/cms/Run2011A/MinimumBias
   ↪ /AOD/120ct2013-v1/20000/00658730-8546-E311-B40B-003048F23FE2.root
   ↪ /mnt/shared/Pawahtun/CMS_Run2011A/MinimumBias_20000/0001.root
4  xrdcp -v -f
   ↪ root://eospublic.cern.ch//eos/opendata/cms/Run2011A/MinimumBias
   ↪ /AOD/120ct2013-v1/20000/006DE055-8646-E311-8C31-0025904B3072.root
   ↪ /mnt/shared/Pawahtun/CMS_Run2011A/MinimumBias_20000/0002.root
5  xrdcp -v -f
   ↪ root://eospublic.cern.ch//eos/opendata/cms/Run2011A/MinimumBias
   ↪ /AOD/120ct2013-v1/20000/008C3D2F-9145-E311-828D-0025901AF548.root
   ↪ /mnt/shared/Pawahtun/CMS_Run2011A/MinimumBias_20000/0003.root
```

En este caso obtenemos AODs del tipo *MinimumBias* correspondientes al conjunto de datos del año 2011 [41]. Y los datos se almacenan en un disco duro externo (llamado “Pawahtun”) al cual se accede a través de una carpeta compartida en la MV (ver apéndice B.2). Pese a que el *macro* muestra la descarga de únicamente 3 archivos, no existe un límite para la cantidad de descargas.

En resumen, utilizando el comando `xrdcp -v -f` es posible descargar datos abiertos AOD simplemente con conocer su dirección y nombre de descarga y asignándole una ubicación y nombre local.

2.1.2. Archivos AODs descargados

Como se explicó en la subsección 1.5.3, para este trabajo fueron descargados AODs de los conjuntos de datos primarios *Jet*, *B-Tag*, *MinBias* y *MultiJet*, correspondientes a colisiones protón-protón con una energía en el centro de masa de 7 GeV y llevadas a cabo durante el año 2011. En la tabla 2.1 se resume la cantidad y tamaño total de los AODs descargados.

Esta información fue almacenada en un disco duro externo de 10 TB de capacidad.

AOD	Cantidad	Tamaño [TB]
Jet	799	3.1
BTag	488	1.8
MinBias	900	2.4
MultiJet	1200	2.5

Tabla 2.1: Tipo, cantidad y tamaño de AODs utilizados en este trabajo.

2.1.3. Información utilizada

El análisis realizado considera únicamente la información contenida en las ramas correspondientes a los jets PF de los AOD. Tomando como ejemplo la rama `recoPFJets_ak5PFJets__RECO.obj`, contenida en el árbol `Eventos` de cualquier AOD, procedemos a mencionar las hojas contenidas en ésta. La información se estructura de manera similar en cualquiera de las 4 ramas correspondientes a cada uno de los algoritmos de reconstrucción de jets PF utilizados por CMS: (`ak5`, `ak7`, `kt4` y `kt6`).

- `recoPFJets_ak5PFJets__RECO.obj.pt_` Esta hoja corresponde al momento transversal del jet.
- `recoPFJets_ak5PFJets__RECO.obj.eta_` Coordenada espacial η del jet.
- `recoPFJets_ak5PFJets__RECO.obj.phi_` Coordenada espacial ϕ del jet.
- `recoPFJets_ak5PFJets__RECO.obj.mass` Masa del jet. Se calcula a partir la masa invariante medida de sus constituyentes.
- `recoPFJets_ak5PFJets__RECO.obj.vertex_.fCoordinates.fX` Componente x del vértice de producción del jet.
- `recoPFJets_ak5PFJets__RECO.obj.vertex_.fCoordinates.fY` Componente y del vértice de producción del jet.
- `recoPFJets_ak5PFJets__RECO.obj.vertex_.fCoordinates.fZ` Componente z del vértice de producción del jet.
- `recoPFJets_ak5PFJets__RECO.obj.pdgId_` Número de identificación según la convención de Particle Data Group. En este caso no aplica pues el PDG Id es únicamente para partículas y los jets no son partículas *per se*.
- `recoPFJets_ak5PFJets__RECO.obj.mJetArea` Área del jet (ver apéndice A.3), sin embargo, este es un concepto que no se utilizará en esta tesis.

- recoPFJets_ak5PFJets__RECO.obj.mPileupEnergy Energía de las partículas de colisiones no asociadas, es decir, provenientes de otros eventos. [50].
- recoPFJets_ak5PFJets__RECO.obj.m_specific.mChargedHadronEnergy Energía total de los hadrones cargados dentro del jet.
- recoPFJets_ak5PFJets__RECO.obj.m_specific.mNeutralHadronEnergy Energía total de los hadrones neutros dentro del jet.
- recoPFJets_ak5PFJets__RECO.obj.m_specific.mPhotonEnergy Energía total de los fotones dentro del jet.
- recoPFJets_ak5PFJets__RECO.obj.m_specific.mElectronEnergy Energía total de los electrones dentro del jet.
- recoPFJets_ak5PFJets__RECO.obj.m_specific.mMuonEnergy Energía total de los muones dentro del jet.
- recoPFJets_ak5PFJets__RECO.obj.m_specific.mHFHadronEnergy Energía total de los hadrones dentro del jet. Esta medición se lleva a cabo solamente por el calorímetro HF.
- recoPFJets_ak5PFJets__RECO.obj.m_specific.mHFEMEnergy Energía total de las partículas electromagnéticas dentro del jet. Esta medición se lleva a cabo únicamente por el calorímetro HF.
- recoPFJets_ak5PFJets__RECO.obj.m_specific.mChargedHadronMultiplicity Multiplicidad de los hadrones cargados dentro del jet.
- recoPFJets_ak5PFJets__RECO.obj.m_specific.mNeutralHadronMultiplicity Multiplicidad de los hadrones neutros dentro del jet.
- recoPFJets_ak5PFJets__RECO.obj.m_specific.mPhotonMultiplicity Multiplicidad de los fotones dentro del jet.
- recoPFJets_ak5PFJets__RECO.obj.m_specific.mElectronMultiplicity Multiplicidad de los electrones dentro del jet.
- recoPFJets_ak5PFJets__RECO.obj.m_specific.mMuonMultiplicity Multiplicidad de los muones dentro del jet.
- recoPFJets_ak5PFJets__RECO.obj.m_specific.mHFHadronMultiplicity Multiplicidad de los hadrones dentro del jet. Esta medición se lleva a cabo solamente por el calorímetro HF.
- recoPFJets_ak5PFJets__RECO.obj.m_specific.mHFEMMultiplicity Multiplicidad de las partículas electromagnéticas dentro del jet. Esta medición se lleva

- a cabo únicamente por el calorímetro HF.
- `recoPFJets_ak5PFJets__RECO.obj.mChargedEmEnergy` Energía total de las partículas electromagnéticas cargadas dentro del jet.
- `recoPFJets_ak5PFJets__RECO.obj.mChargedMuEnergy` Energía total de los muones dentro del jet.
- `recoPFJets_ak5PFJets__RECO.obj.mNeutralEmEnergy` Energía total de las partículas electromagnéticas neutras dentro del jet.
- `recoPFJets_ak5PFJets__RECO.obj.mChargedMultiplicity` Multiplicidad de partículas cargadas dentro del jet.
- `recoPFJets_ak5PFJets__RECO.obj.mNeutralMultiplicity` Multiplicidad de partículas neutras dentro del jet.

Dado que los datos que consideraremos en este trabajo se encuentran contenidos en la misma rama, `recoPFJets_ak5PFJets__RECO.obj.`, a partir de ahora llamaremos a las hojas simplemente por su nombre. Por ejemplo `pt_`, `eta_`, `phi_`, etc. La figura 2.1 muestra esquemáticamente la estructura de las hojas contenidas en la rama `recoPFJets_ak5PFJets__RECO.obj.`, se muestran como ejemplo las hojas `pt_`, `eta_` y `phi_` pero la misma estructura aplica para cualquiera de las otras hojas.

Los jets PF están reconstruidos de tal manera que sus vértices de producción están contenidos en el origen. Es decir, las entradas de `vertex_.fCoordinates.fX`, `vertex_.fCoordinates.fY` y `vertex_.fCoordinates.fZ` son 0 para cualquier jet.

Etiqueta *specific*

Esta es una etiqueta asignada durante el procesamiento de datos y se refiere a cantidades específicas de cada jet PF [67, 68]. En las hojas de las ramas de jets PF, la etiqueta *specific* se denota con la palabra `m_specific`.

Datos HF

El acrónimo HF hace referencia al calorímetro *Hadronic Forward*. Dentro de las ramas de los jets PF, en los AODs, encontramos hojas las cuales tienen dicho acrónimo, por ejemplo `HFHadronEnergy` y `HFEMEnergy`, que hacen referencia a detecciones hadrónicas y electromagnéticas en el HF, respectivamente [67].

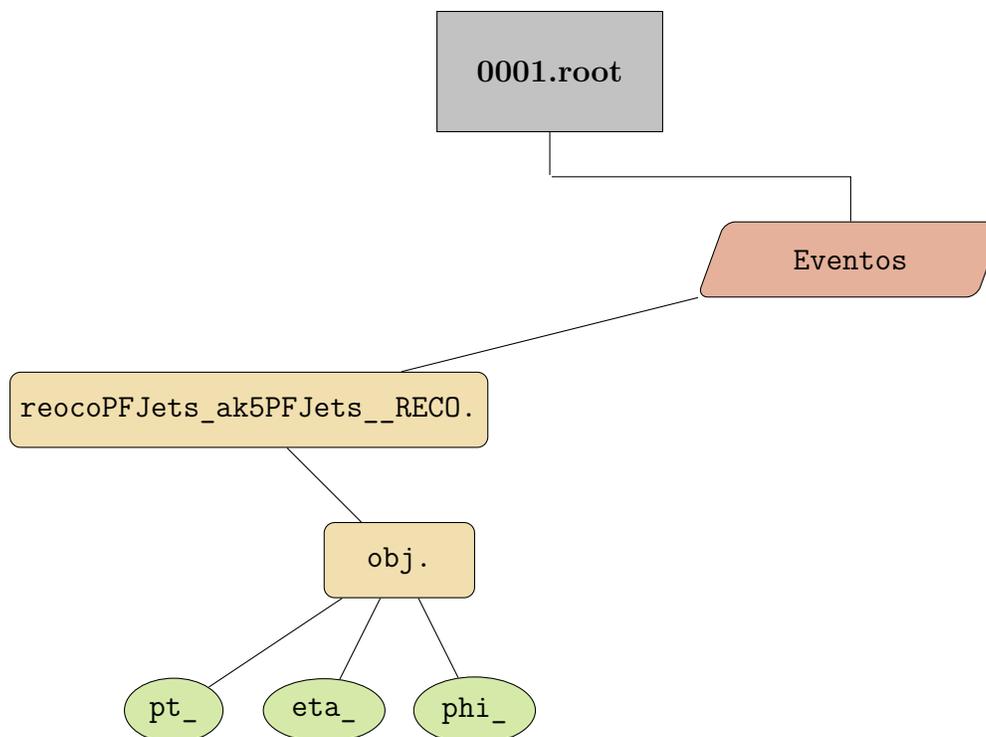


Figura 2.1: Estructura jerárquica dentro de un AOD. El AOD se llama `0001.root`, nuestro árbol tiene el nombre de `Eventos`, el cual almacena a la rama `reocoPFJets_ak5PFJets__RECO.`, dentro de la cual está contenida la rama `obj.` y en esta última rama tenemos a las hojas `pt_`, `eta_` y `phi_`.

Datos Obsoletos

En las ramas de jets PF tenemos hojas que son obsoletas, esto es, la información que contienen estas hojas puede ser obtenida a partir de otras hojas de PFJets.

- `mChargedEmEnergy`
- `mChargedMuEnergy`
- `mNeutralEmEnergy`
- `mChargedMultiplicity`
- `mNeutralMultiplicity`

La única razón por la cual esta información se conserva es por razones de retro-compatibilidad [68].

Dicho lo anterior, podemos reducir nuestro conjunto de datos de análisis a las siguientes hojas:

- pt_
- eta_
- phi_
- m_specific.mChargedHadronEnergy
- m_specific.mNeutralHadronEnergy
- m_specific.mPhotonEnergy
- m_specific.mElectronEnergy
- m_specific.mMuonEnergy
- m_specific.mChargedHadronMultiplicity
- m_specific.mNeutralHadronMultiplicity
- m_specific.mPhotonMultiplicity
- m_specific.mElectronMultiplicity
- m_specific.mMuonMultiplicity

2.1.4. Extracción de información de los AODs

Ahora, nos interesa no sólo conocer la distribución de los datos, sino también poder trabajar con los datos allí contenidos. Para esto fue necesario desarrollar un *macro* que pueda ser entendido por Root. Este *macro* se llama `Histo_pT_jet.C`, está escrito en C y la estructura se muestra a continuación. Cabe mencionar que pese a mostrarse la metodología utilizada para una sola hoja, este procedimiento es válido para cualquier otra hoja.

```

1 void Histo_pT_jet() {
2     // Abrimos el archivo AOD que nos interesa. En este caso se
3     ↪ muestra la ruta completa al archivo llamado 0001.root.
4     TFile *InputFile = new TFile(
5     ↪ "/media/saksevul/Pawahtun/CMS_Run2011A/MinBias_20000/0001.root"
6     ↪ );
7     // Accedemos al árbol de interés. Este árbol recibe el nombre de
8     ↪ "Events". El nombre deriva del hecho de que este árbol
9     ↪ contiene la información referente a todos los Eventos
10    ↪ contenidos en el AOD.
11    TTree *Eventos = (TTree*)InputFile->Get("Events");
12    //Crear el archivo de salida (tiene que ser definido después del
13    ↪ InputFile), el cual ak5PFJets_pt.root.

```

```

7   TFile* OutputFile = new TFile(
   ↪   "/home/saksevul/T/OpenData/MinBias_20000/ak5PFJets_pt.root",
   ↪   "RECREATE" );
8   // Declaramos apuntadores a las hojas que utilizaremos del árbol.
9   TLeaf *Jets_pt_ =
   ↪   Eventos->GetLeaf("recoPFJets_ak5PFJets__RECO.obj.pt_");
10  TLeaf *Jets_eta_ =
   ↪   Eventos->GetLeaf("recoPFJets_ak5PFJets__RECO.obj.eta_");
11  // Creamos el histograma que será utilizado para almacenar la
   ↪   información extraída.
12  TH1F *h_Jets_pt_ = new TH1F( "ak5PFJet_pt_",
   ↪   "recoPFJets_PFJets__RECO.obj.pt_; p_{T} [GeV]; Frecuencia",
   ↪   1200, 0, 1200 );
13
14  // Comenzamos un ciclo for para todos los Eventos contenidos en
   ↪   el árbol "Eventos".
15  for (Int_t e=0; e<Eventos->GetEntries(); e++) {
16     // Cargamos la información correspondiente a los jets del
   ↪     e-ésimo Evento.
17     Jets_pt_->GetBranch()->GetEntry(e);
18     Jets_eta_->GetBranch()->GetEntry(e);
19     // Este ciclo es para obtener la información de todos los jets
   ↪     contenidos en cada Evento. La opción Jets_pt_->GetLen() nos
   ↪     da la cantidad de jets contenidos en el e-ésimo Evento.
20     for (Int_t i=0; i<Jets_pt_->GetLen(); i++) {
21        // Extraeremos solamente la información de los jets que se
   ↪        encuentran dentro del rango (-3.0,3.0) en eta.
22        if (fabs(Jets_eta_->GetValue(i)) > 3.0) continue;
23        // Esta es la manera en que llenamos nuestro histograma con
   ↪        la información proveniente del archivo AOD. Nótese que
   ↪        estamos obteniendo el valor de pT correspondiente al
   ↪        i-ésimo jet del e-ésimo Evento.
24        h_Jets_pt_ -> Fill(Jets_pt_->GetValue(i));
25        } // Cerramos el ciclo for i.
26    } // Cerramos el ciclo for e.
27

```

```

28 // De esta manera escribimos nuestro histograma en nuestro
   ↪ archivo de salida.
29 h_Jets_pt_ -> Write();
30 // Finalmente cerramos el archivo AOD abierto.
31 InputFile -> Close();
32 // Y eliminamos la variable llamada OutputFile, sin embargo, esto
   ↪ no afecta en lo absoluto al archivo de salida.
33 delete OutputFile;
34 }

```

Finalmente, para ejecutar este *macro* se hace uso de la terminal. Una vez cargadas las variables del entorno Root (ver Apéndice B.3) se procede a ejecutar el *macro* `Histo_pT_jet.C` del siguiente modo:

```
root -q /RutaAlMacro/Histo_pT_jet.C
```

Al terminar obtendremos únicamente al archivo de salida `ak5PFJets_pT.root` con la información extraída del AOD graficada en forma de un histograma 1-D. Es decir, perderemos la estructura de la hoja y en cambio obtendremos un archivo que ocupa menos espacio de almacenamiento y es más fácil de manipular y analizar.

2.1.5. Extracción recursiva de información contenida en distintos AODs

En la subsección anterior mostramos cómo extraer información de un solo archivo AOD y hacer un histograma. Para aumentar la cantidad de la información analizada es necesario ser capaz de obtener información proveniente de diversos AODs. Por otra parte, dentro de los AOD existen 4 distintos tipos de jets reconstruidos, a saber, `ak5PFJets`, `ak7PFJets`, `kt4PFJets` y `kt6PFJets`. Finalmente, para ampliar el panorama de nuestro análisis tenemos distintos tipos de AODs que contienen información centrada en distintas índoles, como por ejemplo: *BTag*, *Jet*, *MinBias* y *MultiJet* (ver la subsección 1.4.4).

Entonces, en términos del ejemplo mostrado en la subsección 2.1.4, se desea aumentar la cantidad de la información analizada para estudiar de la distribución de p_T para distintos algoritmos buscadores de jets PF contenidos en diversos tipos de archivos AOD. Esta es la motivación para presentar esta sección donde se menciona dicha metodología.

Sean i el i -ésimo archivo AOD, j el j -ésimo tipo de AOD, m el m -ésimo algoritmo buscador de jets y N_j el número de archivos para cada tipo de AOD. De esta manera $m \in \{ak5, ak7, kt4, kt6\}$, $j \in \{BTag, Jet, MinBias, MultiJet\}$, $i \in \{1, 2, \dots, N_j\}$, donde $N_j \in \{N_{BTag}, N_{Jet}, N_{MinBias}, N_{MultiJet}\}^1$ e $i, N_j \in \mathbb{N}$. Con esto en mente es posible describir un análisis recursivo en términos matemáticos, considerando a `ExtraerPt.C` en función de i , j y m , `ExtraerPt.C (i, j, m)`.

$$\sum_{m=ak5}^{kt6} \left\{ \sum_{j=BTag}^{MultiJet} \left[\sum_{i=1}^{N_j} ExtraerPt.C(i, j, m) \right] \right\} \quad (2.1)$$

Con esto estaríamos ejecutando un total de $4 \{488 + 799 + 900 + 1250\} = 13748$ veces el `macro ExtraerPt.C`. De esta manera aumentamos la cantidad de la información analizada y la variedad de la información analizada.

Una descripción más detallada de esta metodología se proporciona en el apéndice B.4. Para conocer el `macro` específico utilizado en este trabajo, visitar el siguiente enlace

<https://github.com/Saksevul/T/blob/master/OpenData/bashMacro.sh>

2.1.6. Límites de detección del CMS

El detector CMS fue diseñado tomando la detección del bosón de Higgs como punto de referencia [22]. Esta es una referencia particular debido a que existen distintos canales de decaimiento para este bosón. Si a esto le sumamos el hecho de que las partículas cargadas de baja energía no alcanzan los detectores debido al intenso campo

¹Ver tabla 2.1 para conocer explícitamente estos valores.

magnético, tenemos que dependiendo del tipo de partículas, existen distintos umbrales inferiores de detección en el aparato.

Dado que el objetivo principal de la simulación es compararla con los datos abiertos del CMS, es necesario considerar estas limitantes del aparato dentro de la simulación. En este sentido se define un “umbral de detección” y un “umbral de selección” para cada partícula. Estos umbrales estarán en función de la energía promedio de cada tipo de partícula dentro del jet, $\frac{E^{\text{Particula}}}{Mult^{\text{Particula}}}$, por lo que se estudian haciendo uso de la información contenida en los jets PF de los AODs, a saber, se toma la información de las siguientes hojas:

- `m_specific.mChargedHadronEnergy`
- `m_specific.mNeutralHadronEnergy`
- `m_specific.mPhotonEnergy`
- `m_specific.mElectronEnergy`
- `m_specific.mMuonEnergy`
- `m_specific.mChargedHadronMultiplicity`
- `m_specific.mNeutralHadronMultiplicity`
- `m_specific.mPhotonMultiplicity`
- `m_specific.mElectronMultiplicity`
- `m_specific.mMuonMultiplicity`

El **umbral de detección** es una aproximación del límite inferior de detección de CMS para cada tipo de partículas en función de la energía promedio de estas. En la figura 2.2 se muestra el histograma perteneciente a los electrones contenidos en los AODs dentro de los jets reconstruidos con el algoritmo `ak5PF`. La lectura es la siguiente: El ancho de cada bin es $d\frac{E^{\text{Electron}}}{Mult^{\text{Electron}}}$, el número de entradas es dN , en este sentido, $\frac{dN}{d(E^{\text{Electron}}/Mult^{\text{Electron}})}$ corresponde al número de entradas en cada bin; adicionalmente, el histograma se encuentra normalizado al número total de entradas: N_{Tot} . Es de notar que en el rango aproximado de 0 a 3.0 GeV la tendencia es claramente creciente. Lo cual es contrario a lo que se esperaría por el hecho de que las partículas menos energéticas deberían dominar la distribución, pues estas requieren menos energía para ser creadas y por tanto su multiplicidad debería ser mayor. Por ende, concluimos que CMS es ineficiente para detectar electrones con

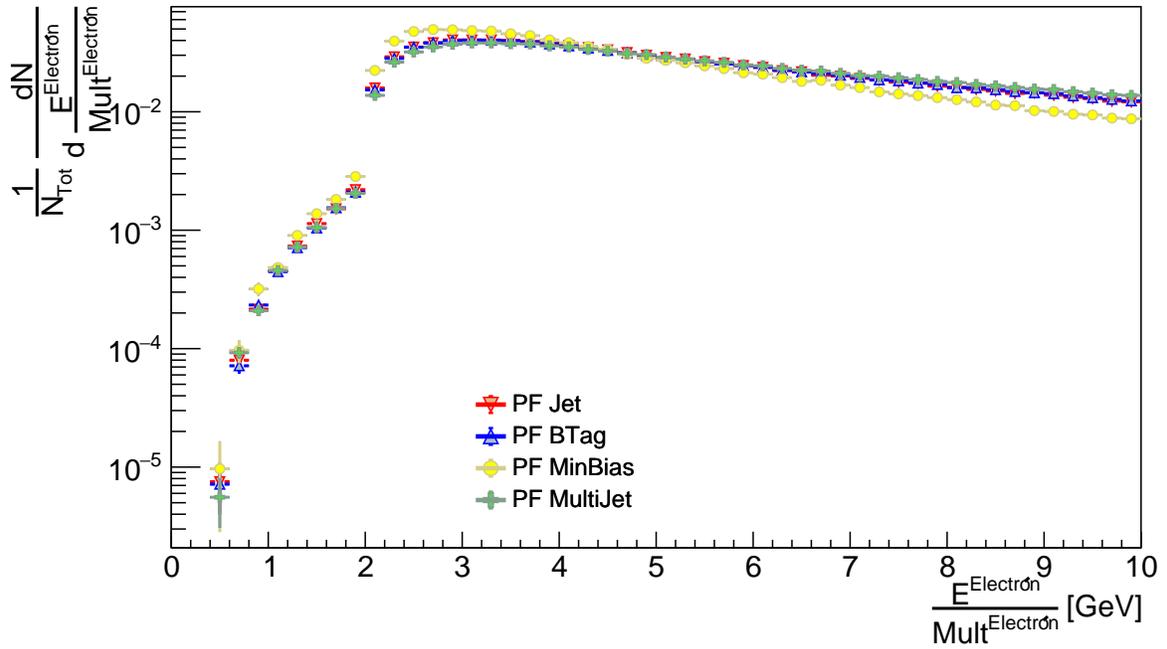
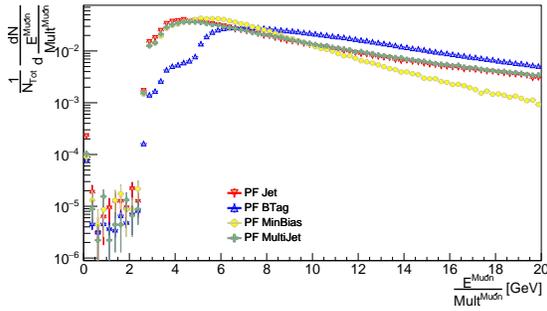
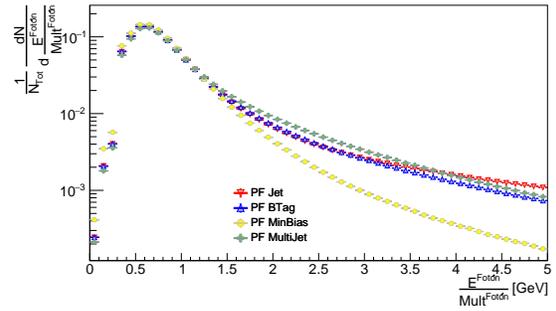


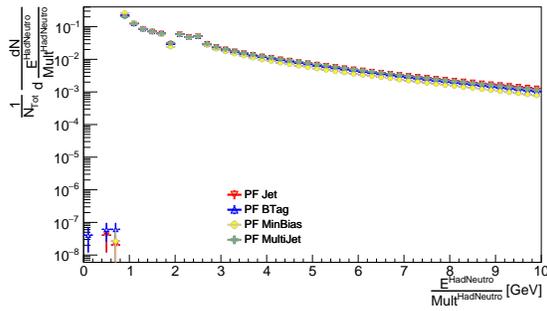
Figura 2.2: Histograma de la distribución de la energía entre la multiplicidad para electrones dentro de ak5PFJets.



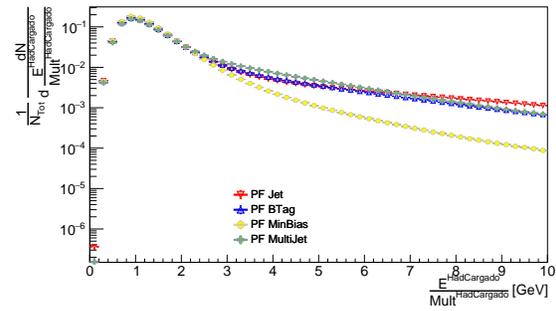
(a) Muones



(b) Fotonos



(c) Hadrones Neutros



(d) Hadrones Cargados

Figura 2.3: Distribuciones de energía entre la multiplicidad para los 4 constituyentes restantes de ak5PFJets en los AODs.

energías menores a ~ 3.0 GeV. Consecuentemente, para los electrones individuales, se propone un umbral de detección de 2.2 GeV dentro de la simulación.

Adicionalmente se propone un **umbral de selección**, tanto para el análisis de datos como para la simulación, el cual es un límite inferior a partir del cual las distribuciones de $\frac{E}{Mult}$ comienzan a comportarse de manera decreciente, justo como es de esperar. De modo que, en esta tesis, sólo se analizan jets cuyos electrones tengan una energía promedio mayor a 3.8 GeV (la cual es el umbral de selección para electrones).

De la misma manera se obtienen histogramas análogos para los muones, fotones, hadrones neutros y hadrones cargados (figura 2.3). Y podemos proponer valores análogos a los propuestos para los electrones. La tabla 2.2 muestra los valores del umbral de detección y el umbral de selección para cada conjunto de partículas en la simulación.

Partícula	Umbral de detección [GeV]	Umbral de selección [GeV]
Muones	4.00	8.0
Fotones	0.32	0.8
Electrones	2.20	3.8
Hadrones Neutros	0.80	2.6
Hadrones Cargados	0.60	1.0

Tabla 2.2: Umbral de detección y umbral de selección utilizados en la simulación para que la información proveniente de ésta sea similar a la de los datos abiertos.

2.2. Simulación

La simulación se enfoca en comparar los resultados predichos por modelos físicos con la información que fue encontrada con el experimento e incorporada a los AODs. La diferencia aquí es que en lugar de utilizar partículas reconstruidas como sucede con PF, utilizaremos partículas generadas mediante eventos de Pythia, y posteriormente reconstruiremos los jets utilizando la paquetería FastJet.

Entonces el preámbulo de esta simulación es conectar Pythia con FastJet y Root. Por lo que es necesario seguir los pasos detallados en el apéndice B.5.

2.2.1. *Macro* de simulación

En las sub-secciones 1.9.3 y 1.10.1 fueron mencionados los parámetros necesarios para la generación de eventos con Pythia y la reconstrucción de jets con FastJet, ahora se muestra el núcleo de la simulación, la mejor forma para hacerlo es mediante un código, así que a continuación se muestra la estructura del *macro* Sim.cc.

```

1 // Necesario para utilizar archivos de entrada y salida.
2 #include <iostream>
3 // Para acceder a los elementos de Pythia 8.
4 #include "Pythia8/Pythia.h"
5 // Entorno Root para realizar histogramas.
6 #include "TH1.h"
7 // Para graficas interactivas en Root.
8 #include "TVirtualPad.h"
9 #include "TApplication.h"
10 // Para guardar archivos en formato Root.
11 #include "TFile.h"
12 // Habilita la inicialización automática de los objetos de
13   ↪ fastjet::PseudoJet.
14 #include "Pythia8Plugins/FastJet3.h"
15 // Necesario para poder utilizar información definida por el usuario (ver
16   ↪ apéndice B.6).
17 #include "fastjet/MyInfo.h"

```

```

17 using namespace Pythia8;
18 int main(int argc, char* argv[]) {
19     // Creamos la aplicación del entorno Root.
20     TApplication theApp("hist", &argc, argv);
21     // Creamos el archivo de salida donde los histogramas se guardarán.
22     TFile* OutputFile = new
23     ↪ TFile("/home/saksevul/T/Pythia/FastJet/ak5MC-10.root", "RECREATE");
24     // Creamos los histogramas que utilizaremos.
25     TH1F* h_Jets_pt_ = new TH1F("Jets_pt_", "Espectro de p_{T} de
26     ↪ ak5MCJets; p_{T} [GeV]; Frecuencia", 100, 0, 1200);
27     TH1F *h_Muons_pt_ = new TH1F("Muons_pt_", "p_{T} de Muones contenidos
28     ↪ en ak5MCJets; p_{T} [GeV]; Frecuencia", 100, 0, 300);
29     // Generador de eventos.
30     Pythia Pythia;
31     // Tipografía de Eventos.
32     Event& event = Pythia.event;
33
34     // Selección de procesos (ver subsección 1.9.3).
35     // Generamos procesos QCD duros. Es decir, con pTHatMin mayor a 2 GeV.
36     Pythia.readString("HardQCD:all = on");
37     // Aquí forzamos a que los procesos QCD tengan un pTHatMin mayor a 10
38     ↪ GeV
39     Pythia.readString("PhaseSpace:pTHatMin = 10.0");
40     // Inicializamos la simulación a la energía (en el centro de masa) del
41     ↪ LHC correspondiente a los datos abiertos del años 2011.
42     Pythia.readString("Beams:eCM = 7000.0");
43     Pythia.init();
44     // Número de eventos generados.
45     int nEvent = 12000;
46
47     // Seleccionamos los parámetros comunes para la reconstrucción de jets
48     ↪ utilizando de FastJet (ver subsección 1.10.1).
49     // Algoritmo de reconstrucción de jets, ak= -1 y kt = 1.
50     int JCA = -1;
51     // Radio del jet.
52     double R = 0.5;
53     // pT mínimo necesario para almacenar la información del jet [49]. Este
54     ↪ valor se tomó para ser el mismo que el de los datos abiertos.
55     double pTMin = 3.0;

```

```

49 // Rango máximo en pseudorapidez del detector.
50 double etaMax = 1.479;
51
52 // Definir el algoritmo de reconstrucción de jets con los parámetros
   ↪ anteriores.
53 fastjet::JetDefinition jetDef(fastjet::genkt_algorithm, R, JCA);
54 // Definimos una variable para almacenar la información de las
   ↪ partículas que conformarán los jets.
55 std::vector <fastjet::PseudoJet> fjInputs;
56
57 // Comenzamos el ciclo for sobre los eventos generados. Saltamos los
   ↪ errores (en caso de haber).
58 for (int iEvent=0; iEvent<nEvent; ++iEvent){ if (!Pythia.next())
   ↪ continue;
59 // Comenzamos el análisis de FastJet y extraemos la información
   ↪ necesaria de las partículas del registro de eventos.
60 fjInputs.resize(0); Vec4 pTemp; double mTemp;
61 // Comenzamos un ciclo for para todas las partículas generadas en el
   ↪ iEvent-ésimo evento. Y consideramos únicamente a las partículas de
   ↪ estado final, esto es, partículas que no han decaído [59].
62 for (int i = 0; i < event.size(); ++i) if (event[i].isFinal()) {
63 // Elegimos partículas "visibles" dentro del detector. Por "visible"
   ↪ nos referimos a que posean carga eléctrica, carga fuerte, o que
   ↪ se compongan de este tipo de partículas.
64 if (!event[i].isVisible()) continue;
65 // Delimitamos en  $\eta$  las partículas que serán utilizadas para el
   ↪ Algoritmo de Reconstrucción de jets.
66 if (abs(event[i].eta()) > 2*etaMax) continue;
67 // Limitamos la energía mínima necesaria para cada tipo de
   ↪ partículas para poder ser parte de un jet (ver subsección
   ↪ 2.1.6). Los id's siguen la convención utilizada en [69].
68 if (abs(event[i].id())==13 && event[i].e()<4.00) continue;
69 else if (abs(event[i].id())==22 && event[i].e()<0.32) continue;
70 else if (abs(event[i].id())==11 && event[i].e()<2.20) continue;
71 else if (abs(event[i].isNeutral()) && event[i].e()<0.80) continue;
72 else if (abs(event[i].isCharged()) && event[i].e()<0.60) continue;
73 // Generamos un pseudojet a partir de las partículas generadas por
   ↪ Pythia. particleTemp son las partículas candidatas a formar
   ↪ parte del jet.

```

```

74     fastjet::PseudoJet particleTemp = event[i];
75     // Esto es para poder almacenar el índice de evento de cada
       ↪ constituyente del jet (ver apéndice B.6). Esto es sumamente útil
       ↪ pues el registro de eventos contiene toda la información acerca
       ↪ de las partículas.
76     particleTemp.set_user_info( new MyInfo( i ) );
77     // Guardamos las partículas aceptables para poder formar parte de
       ↪ algún jet.
78     fjInputs.push_back( particleTemp);
79 } // Cerramos ciclo for i.
80 // Ejecutamos el algoritmo FastJet.
81 vector <fastjet::PseudoJet> inclusiveJets, sortedJets;
82 // Indicamos las partículas a utilizar para conformar los jets e
       ↪ indicamos el tipo de ARJ deseado.
83 fastjet::ClusterSequence clustSeq(fjInputs, jetDef);
84 // pT mínimo necesario para incluir a un jet en el registro.
85 inclusiveJets = clustSeq.inclusive_jets(pTMin);
86 // Ordenamos los jets con respecto a su pT.
87 sortedJets    = sorted_by_pt(inclusiveJets);
88
89 // Realizamos un ciclo for j sobre todos los jets generados en el
       ↪ iEvent-ésimo Evento.
90 for (size_t j=0; j<sortedJets.size(); j++){
91     // Delimitamos el rango en  $\eta$  para los jets que utilizaremos.
92     if (abs(sortedJets[j].eta()) > etaMax) continue;
93     // Estos son criterios de selección utilizados para tener
       ↪ información y estadística similar proveniente de los datos y la
       ↪ simulación (ver subsección 2.1.6).
94     if (MuonMultiplicity>0 && MuonEnergy/MuonMultiplicity<8.0) continue;
95     if (PhotonMultiplicity>0 && PhotonEnergy/PhotonMultiplicity<0.8)
       ↪ continue;
96     if (ElectronMultiplicity>0 &&
       ↪ ElectronEnergy/ElectronMultiplicity<3.8) continue;
97     if (NeutralHadMultiplicity>0 &&
       ↪ NeutralHadEnergy/NeutralHadMultiplicity<2.6) continue;
98     if (ChargedHadMultiplicity>0 &&
       ↪ ChargedHadEnergy/ChargedHadMultiplicity<1.0) continue;
99     // Llenamos el histograma con el pT de cada jet.
100    h_Jets_pt_ -> Fill(sortedJets[j].pt());

```

```

101     // Ciclo for k sobre todos los constituyentes del j-ésimo jet en el
        ↪ iEvent-ésimo Evento.
102     for (size_t k=0; k<sortedJets[j].constituents().size(); k++) {
103         // Esta variable nos sirve para acceder al registro de Eventos del
        ↪ k-ésimo constituyente del j-ésimo jet en el iEvent-ésimo
        ↪ Evento.
104         Int_t JCIndex =
        ↪ sortedJets[j].constituents()[k].user_info<MyInfo>().Index();
105         // Seleccionamos los muones dentro de los jets.
106         if (abs(event[JCIndex].id())==13) {
107             // Llenamos el histograma correspondiente al pT de los muones
        ↪ contenidos dentro de jets.
108             h_Muons_pt_ -> Fill(sortedJets[j].constituents()[k].pt());
109         } // Ciclo if.
110     } // Ciclo for k.
111 } // Ciclo for j.
112 } // Ciclo for iEvent.
113 // Escribimos los histogramas en el archivo .root de salida.
114 h_Jets_pt_ -> Write();
115 h_Muons_pt_ -> Write();
116 // Eliminamos la variable relacionada al archivo de salida.
117 delete OutputFile;
118 // Finalizamos el programa.
119 return 0;
120 }

```

Este *macro* muestra únicamente la manera de extraer el momento transversal (p_T) de los jets reconstruidos y el p_T de los muones contenidos dentro de éstos, esto es más que suficiente para presentar la forma básica de utilizar Pythia, FastJet y Root juntos (ver apéndice B.5); además de incorporar la información del usuario definida en el apéndice B.6 y el tratamiento de los jets y sus componentes.

2.2.2. Generalización del *macro* de simulación

Previamente se mostró la manera de simular colisiones protón-protón en Pythia, utilizar FastJet para reconstruir los jets y obtener los archivos de salida en formato *.root*. Sin embargo, ese *macro* de muestra nos limita a utilizar un único \hat{p}_T^{\min} y un

solo algoritmo de reconstrucción de jets, en dicho caso $\hat{p}_T^{\min} = 10$ y el algoritmo de reconstrucción de jets ha sido `ak5`.

Con el fin de mejorar esta limitación se hace uso de un *macro* en `bash` el cual modificará a `Sim.cc` para variar el \hat{p}_T^{\min} y posteriormente el algoritmo de reconstrucción de jets (ARJ). De esta manera, tendremos a $\hat{p}_T^{\min} \in \{1, 2, 3, \dots, 3500\}$ y $ARJ \in \{ak5, ak7, kt4, kt6\}$. En otras palabras, se recorrerá todo el rango de \hat{p}_T^{\min} para cada ARJ de la siguiente manera

$$\sum_{ARJ=ak5}^{kt6} \left[\sum_{\hat{p}_T^{\min}=1}^{3500} \text{Sim.cc}(\hat{p}_T^{\min}, ARJ) \right] \quad (2.2)$$

Este proceso proporciona las ventajas de ampliar el rango de energía de las partículas generadas gracias al incremento de \hat{p}_T^{\min} y esto nos permite hacer una comparación con respecto a la información contenida en los AODs.

Para saber exactamente la forma del *macro* utilizado en este trabajo, ver el siguiente archivo:

`https://github.com/Saksevul/T/blob/master/Pythia/bashMacro.sh`

el cual es justamente el *macro* `bash` mencionado arriba.

Número de Eventos en función del \hat{p}_T^{\min}

Recordemos que \hat{p}_T^{\min} denota la cantidad de momento transversal mínimo que llevan las partículas y sus subproductos después de colisionar. De manera intuitiva esperamos que en una colisión real sea más probable tener un \hat{p}_T^{\min} *bajo* que uno *alto*². Esto último debido a que, en términos clásicos, el parámetro de impacto es inversamente proporcional al p_T de los productos de la colisión. Dicho esto, para simular una colección de eventos que se asemeje a lo que tenemos en los datos abiertos, debemos de reducir el número de eventos simulados a medida que aumenta el \hat{p}_T^{\min} .

²Entiéndase *bajo* o *alto* en comparación a la energía disponible en el centro de masa.

Cabe destacar que esto no es un *tuning*³ para Pythia *per se*, sino una manera de aumentar la cantidad de eventos con alto \hat{p}_T^{\min} a la simulación.

Ahora falta conocer la relación adecuada entre \hat{p}_T^{\min} y el número de eventos, en este trabajo se proponen y se estudian 3 posibles relaciones. Estas relaciones serán escritas como el número de eventos en función del \hat{p}_T^{\min} y tendrán un dominio de $\hat{p}_T^{\min} \in \{0, 1, \dots, 3500\}$.

Distribución	Función	Eventos totales
Exponencial	$NoE(\hat{p}_T^{\min}) = 1200 \left(10^{-\hat{p}_T^{\min}/1000} - 10^{-3.5} \right)$	516203
Potencial	$NoE(\hat{p}_T^{\min}) = 3000 \left(\left(\hat{p}_T^{\min} \right)^{-0.1} - 3500^{-0.1} \right)$	511301
Lineal	$NoE(\hat{p}_T^{\min}) = \frac{1}{12} \left(3500 - \hat{p}_T^{\min} \right)$	508377

Tabla 2.3: Relaciones para el número de eventos (NoE) en función del \hat{p}_T^{\min} .

El rango de las funciones se escogió de acuerdo el siguiente razonamiento. Dado que las colisiones simuladas son de protón-protón, con una energía en el centro de masa de 7 TeV, entonces la energía relativista de cada protón es $E_p = \sqrt{p^2 c^2 + m^2 c^4} = 3500$ GeV (donde c es la velocidad de la luz en el vacío). Por lo que sus subproductos pueden tener un p_T máximo de ~ 3500 GeV/ c después de la colisión. Por otra parte, la forma de las funciones se escogió de tal manera que $NoE(\hat{p}_T^{\min})$ sea estrictamente decreciente y que al evaluar en 3500 GeV/ c , el resultado sea cero. Esto último para asegurar que \hat{p}_T^{\min} se lleve máximo todo el momento inicial.

Los resultados obtenidos para este análisis se muestran en el siguiente capítulo en función de la FF que ahí se define.

³El *tuning* de MC consiste en encontrar valores para los parámetros del generador de modo que las distribuciones para estas variables en MC se parezca a la de los datos [70].

Capítulo 3

Resultados

Los AODs se dividen en 4 conjuntos: *Jet*, *BTag*, *MinBias* y *MultiJet*. Dentro de cada AOD existe información de 4 tipos de jets reconstruidos utilizando partículas PF, a saber: *ak5PFJets*, *ak7PFJets*, *kt4PFJets* y *kt6PFJets*; los cuales difieren entre sí por el algoritmo de reconstrucción de jets utilizado (ver sección 1.8.2). Finalmente e independientemente del tipo de AOD y del algoritmo de reconstrucción de jets utilizado, dentro de cada rama correspondiente a **PFJets** tenemos información de la energía de los 5 posibles componentes de los jets (muones, fotones, electrones, hadrones neutros y hadrones cargados) y justamente se utilizarán estas energías y la energía total del jet para reportar los resultados obtenidos.

Para describir los resultados obtenidos se debe definir una Función de Fragmentación (FF). Esta FF será el cociente de la energía de los distintos constituyentes de cada jet respecto a la energía total del jet:

$$FF = \frac{E^{\text{Particula}}}{E^{\text{Jet}}} \quad (3.1)$$

La presentación de los resultados está organizado de la siguiente manera: histogramas para cada constituyente del jet, en la parte superior, se muestra el cociente de energía de esta partícula respecto a la energía total del jet. Esto se mostrará para los 4 conjuntos de datos analizados y para la simulación. En la parte inferior de los histogramas se mostrará la distribución de la variable ξ , la cual se define como:

$$\xi = \frac{E^{\text{Datos}} - E^{\text{Simulacion}}}{E^{\text{Simulacion}}} \quad (3.2)$$

Dada la definición de esta variable, solamente se tendrá una distribución por cada conjunto de datos. Esta variable será de utilidad para tener una medida cuantitativa de cuanto se ajusta la simulación a cada conjunto de datos.

Entonces, se busca encontrar similitudes entre la simulación y los conjuntos de datos, para hacer esto se toman distintos rangos de energía y en estos se analizan los cocientes de energía con el fin de estudiar su comportamiento para dar una descripción cualitativa de la fragmentación. En caso de encontrar discrepancias se considerará a la simulación inadecuada para describir la fragmentación de jets, esto pues, acorde con la referencia [71], los conjuntos de datos abiertos son lo suficientemente adecuados para prescindir de “correcciones”¹.

Como panorama general se tendrá el siguiente número de histogramas de cocientes de energía (los cuales están disponibles para su consulta en el apéndice C.2):

$$\sum_{ak5}^{kt6} \left[\sum_{\text{Muon}}^{\text{HadCargado}} (15) \right] = 300 \quad (3.3)$$

pues se utilizan 15 rangos de energía para cada tipo de partículas.

Número de eventos en función del \hat{p}_T^{\min}

Lo primero que buscamos conocer de la simulación es el comportamiento de FF dependiendo de la relación entre el número de eventos y \hat{p}_T^{\min} . Con este objetivo mostramos la figura 3.1, la cual representa la distribución de la FF correspondiente a los fotones dentro del jet. Los distintos histogramas difieren en la relación entre el número de eventos y el \hat{p}_T^{\min} utilizado para la simulación; las relaciones explícitas son las mostradas en la tabla 2.3.

Lo que podemos notar de primera instancia es que, en el caso de los fotones, la distribución de la FF propuesta no depende de la relación entre el número de eventos

¹Textualmente: *Even without detector unfolding, we find very good agreement [...]*. Refiriéndose a la exactitud de los datos.

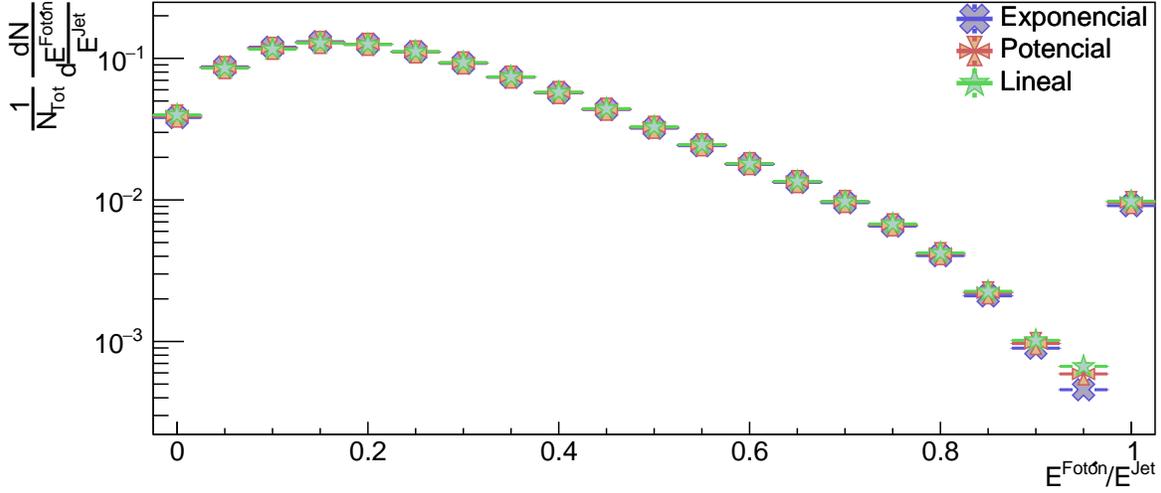


Figura 3.1: Función de fragmentación para fotones usando **ak5MCJets**. Los diferentes histogramas son el resultado de usar distintas funciones para describir la relación entre el número de eventos y el \hat{p}_T^{\min} .

(NoE) y el \hat{p}_T^{\min} . Del mismo modo, se presentan los otros 4 tipos de partículas restantes (figura 3.2) y obtenemos la misma independendencia que el caso de los fotones. Con esto, tenemos la información necesaria para considerar que nuestra FF es independiente respecto a la relación entre el número de eventos simulados y el \hat{p}_T^{\min} .

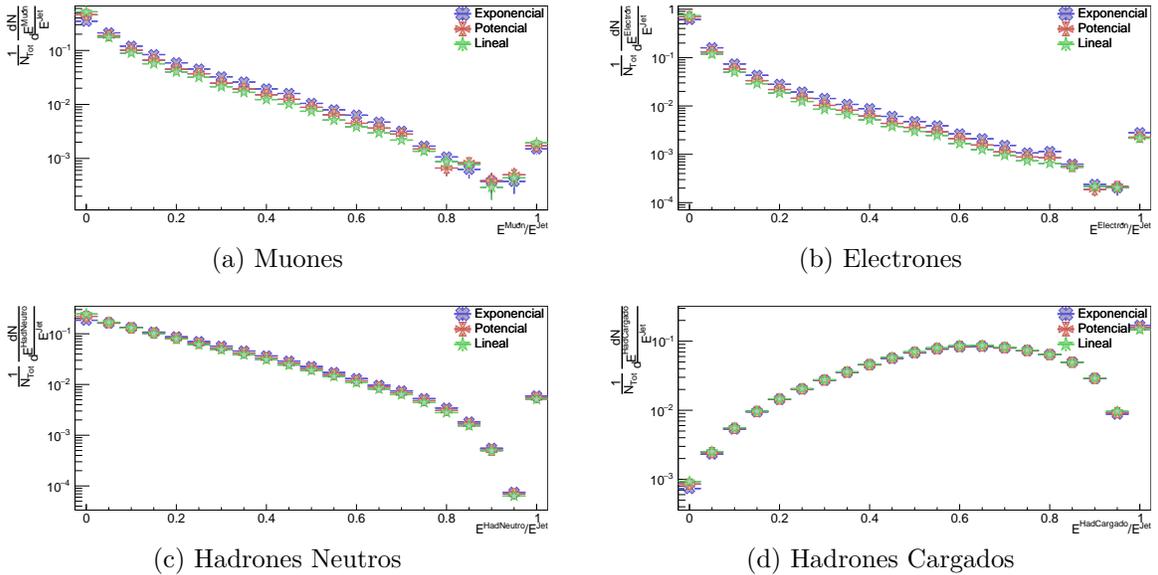


Figura 3.2: Función de fragmentación respecto a 4 constituyentes en jets simulados (**ak5MCJets**). Los distintos colores denotan distintas relaciones utilizadas entre el número de eventos y el \hat{p}_T^{\min} .

Por lo tanto, para el análisis posterior se utiliza la relación de potencia entre el \hat{p}_T^{\min} y el número de eventos (NoE), pues esta resulta ser la más “natural”, a saber,

$$\text{NoE}(\hat{p}_T^{\min}) = 3000 \left((\hat{p}_T^{\min})^{-0.1} - 3500^{-0.1} \right) \quad (3.4)$$

3.1. Resultados de la FF propuesta

En esta sección serán mostrados y discutidos los resultados más sobresalientes encontrados con la FF para los 5 tipos de partículas. Se cuenta con 15 intervalos de energía para estudiar el comportamiento de cada tipo de partícula, y dada la naturaleza de la FF propuesta, estos intervalos son respecto a la energía total del jet y son los siguientes:

- $0 \leq E^{\text{Jet}} [GeV]$ ▪ $100 \leq E^{\text{Jet}} < 200[GeV]$ ▪ $600 \leq E^{\text{Jet}} < 700 [GeV]$
- $0 \leq E^{\text{Jet}} < 10 [GeV]$ ▪ $200 \leq E^{\text{Jet}} < 300[GeV]$ ▪ $700 \leq E^{\text{Jet}} < 800 [GeV]$
- $10 \leq E^{\text{Jet}} < 30 [GeV]$ ▪ $300 \leq E^{\text{Jet}} < 400[GeV]$ ▪ $800 \leq E^{\text{Jet}} < 900 [GeV]$
- $30 \leq E^{\text{Jet}} < 60 [GeV]$ ▪ $400 \leq E^{\text{Jet}} < 500[GeV]$ ▪ $900 \leq E^{\text{Jet}} < 1000[GeV]$
- $60 \leq E^{\text{Jet}} < 100[GeV]$ ▪ $500 \leq E^{\text{Jet}} < 600[GeV]$ ▪ $1000 \leq E^{\text{Jet}} [GeV]$

Para presentar de manera concisa la información obtenida, algunos de estos rangos son omitidos o agrupados con otros que contengan información similar, dependiendo del caso. Los histogramas correspondientes a los 15 rangos de energías, para cada uno de los 5 tipos de partículas estudiados, se encuentran disponibles en el apéndice C.2.

3.1.1. Muones

Comenzamos con mostrar en la figura 3.3 los resultados obtenidos para los muones contenidos en jets reconstruidos con el algoritmo *ak5*. Pese a que los jets MC no se acoplan perfectamente a la distribución de ningún conjunto de datos, notamos que en el rango $0.20 \leq \frac{E^{\text{Muon}}}{E^{\text{Jet}}} \leq 0.85$ la simulación se ajusta relativamente bien al conjunto *Jet* y casi al final notamos un breve dominio del conjunto *MultiJet*.

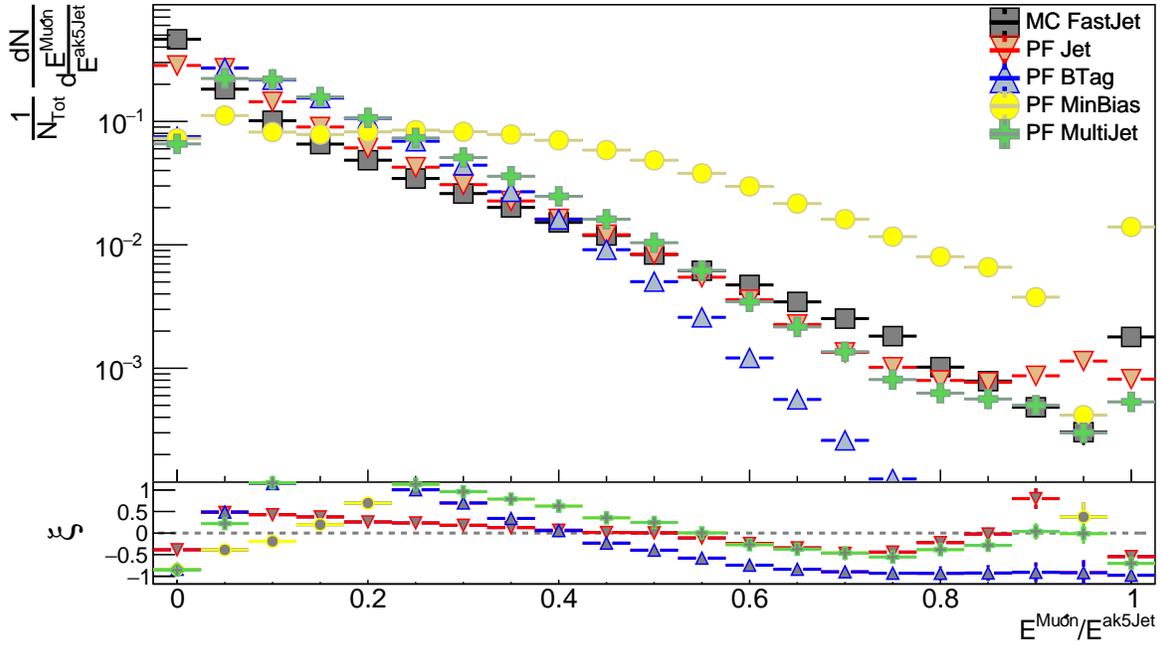


Figura 3.3: Función de fragmentación para muones contenidos en $ak5Jets$. El número de entradas en cada bin es $\frac{dN}{d(E^{Muon}/E^{ak5Jets})}$, adicionalmente, el histograma se encuentra normalizado al número total de entradas: N_{Tot} . La variable ξ , descrita al inicio de éste capítulo, es utilizada para obtener una medida cuantitativa del ajuste de la simulación a los conjuntos de datos.

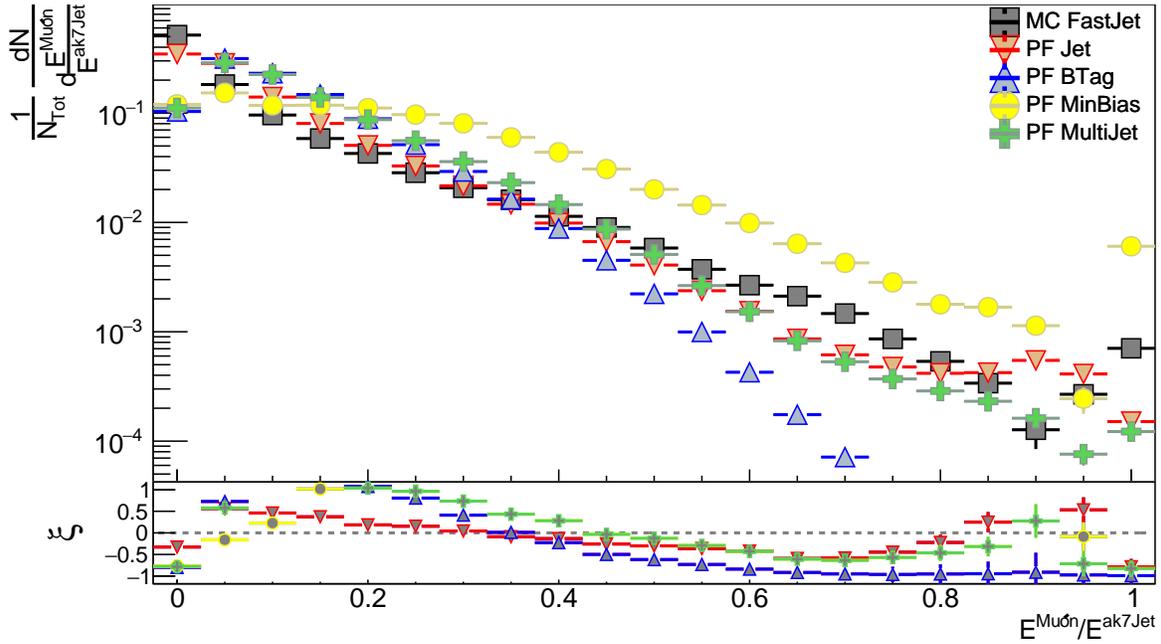


Figura 3.4: Función de fragmentación para muones contenidos en $ak7Jets$.

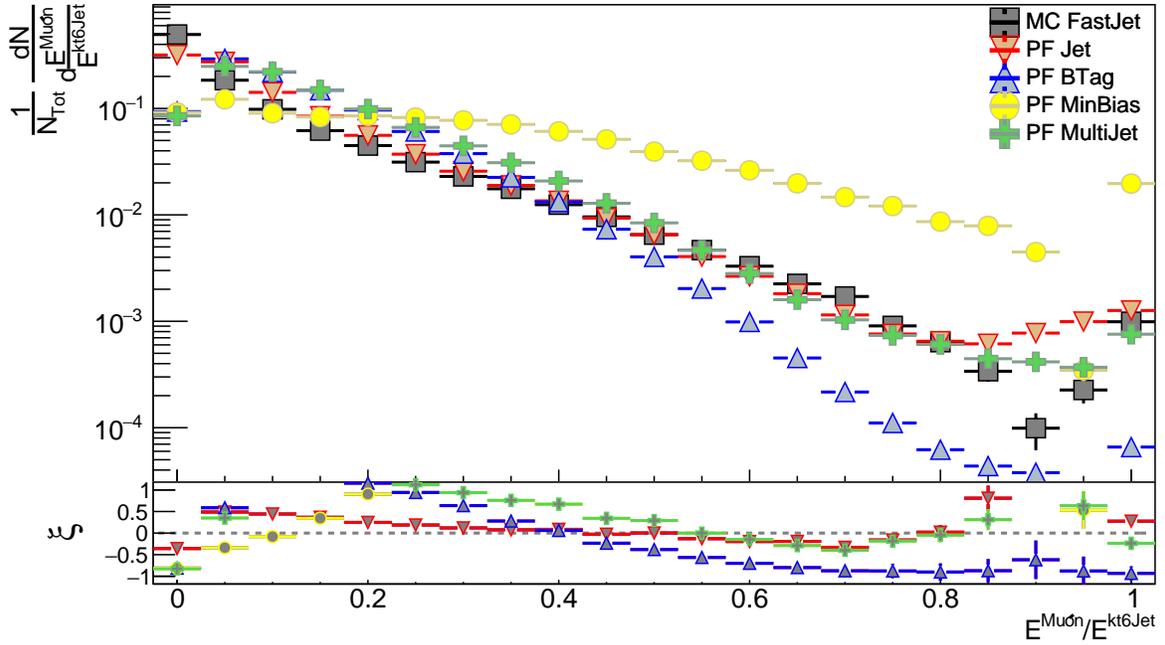


Figura 3.5: Función de fragmentación para muones contenidos en $kt6Jets$.

La tendencia anterior la podemos notar con jets reconstruidos con los algoritmos $ak7$, mostrada en la figura 3.4 en el rango $0.15 \leq \frac{E^{Muon}}{E^{Jet}} \leq 0.85$, y $ak6$, de la figura 3.5 en el rango $0.20 \leq \frac{E^{Muon}}{E^{Jet}} < 0.85$. Con lo que podemos aseverar que esta conducta no es propia del algoritmo ak , sino que es una tendencia para jets cuyo radio es mayor o igual a 5.

Para poder dar una mejor explicación a este comportamiento, se han desglosado estas figuras para distintos rangos de energía del jet. Sin embargo, la configuración de Pythia utilizada no es capaz de describir de manera refinada la fragmentación de jets en muones. Por tanto, se ha decidido revisar el comportamiento de las otras partículas que conforman los jets.

3.1.2. Fotones

El siguiente tipo de partícula analizadas es el fotón. Para esto comenzamos con la distribución de la FF en todo el rango de energía disponible para los jets reconstruidos con el algoritmo $ak5$ (figura 3.6).

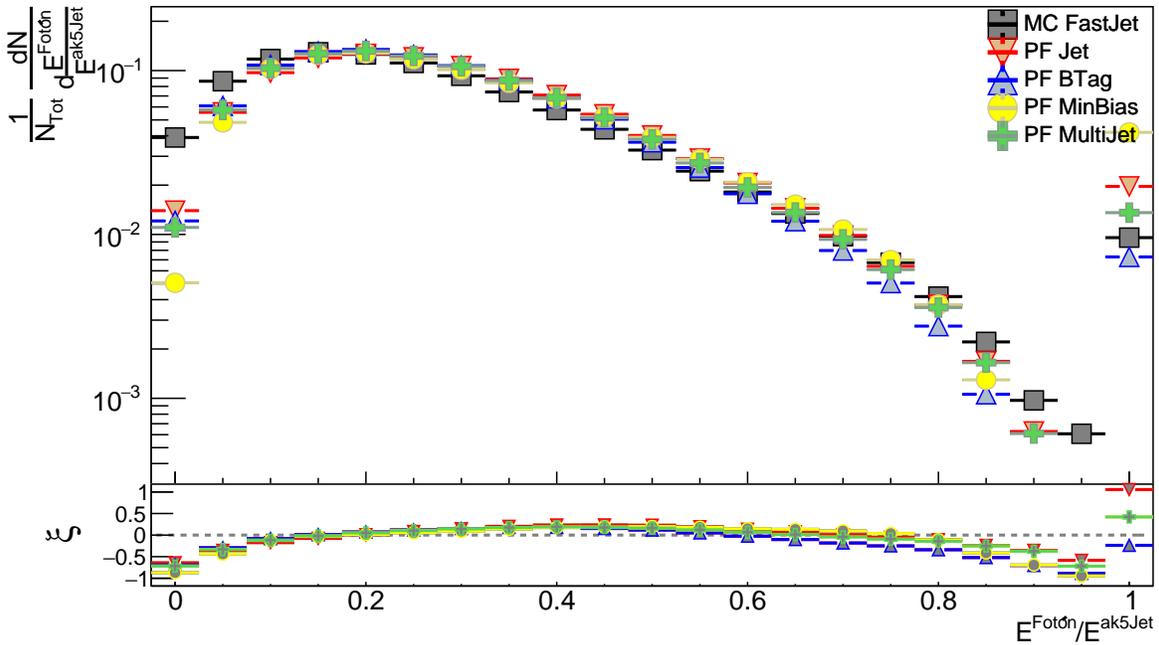


Figura 3.6: Comportamiento de la FF para fotones contenidos en ak5Jets.

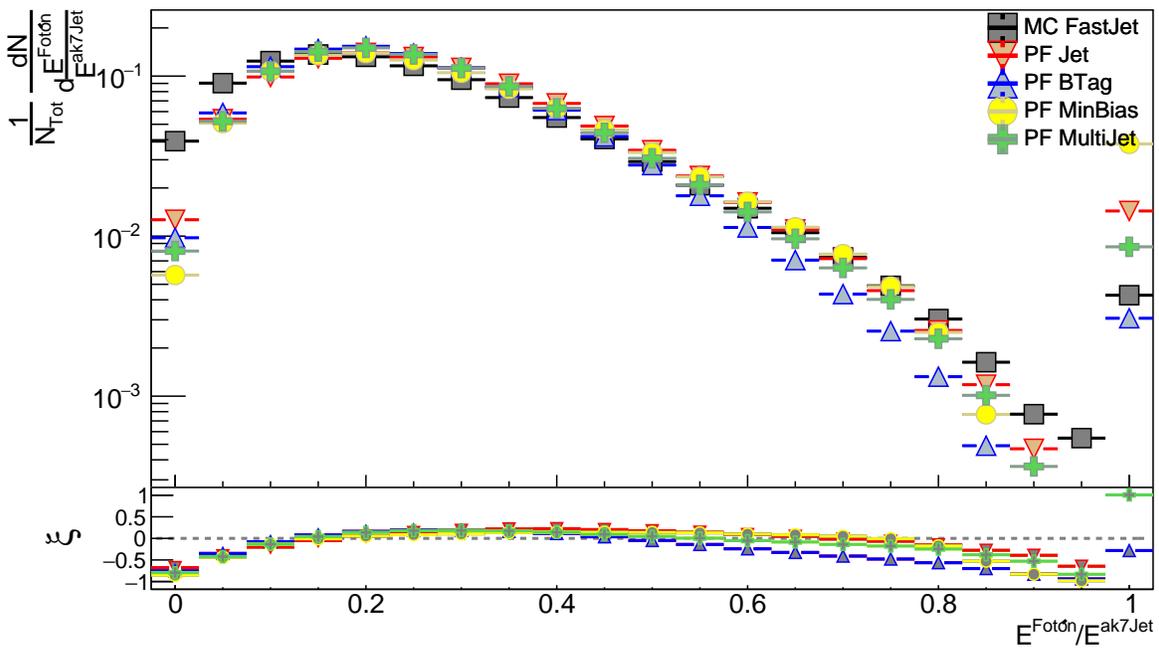


Figura 3.7: Comportamiento de la FF para fotones contenidos en ak7Jets.

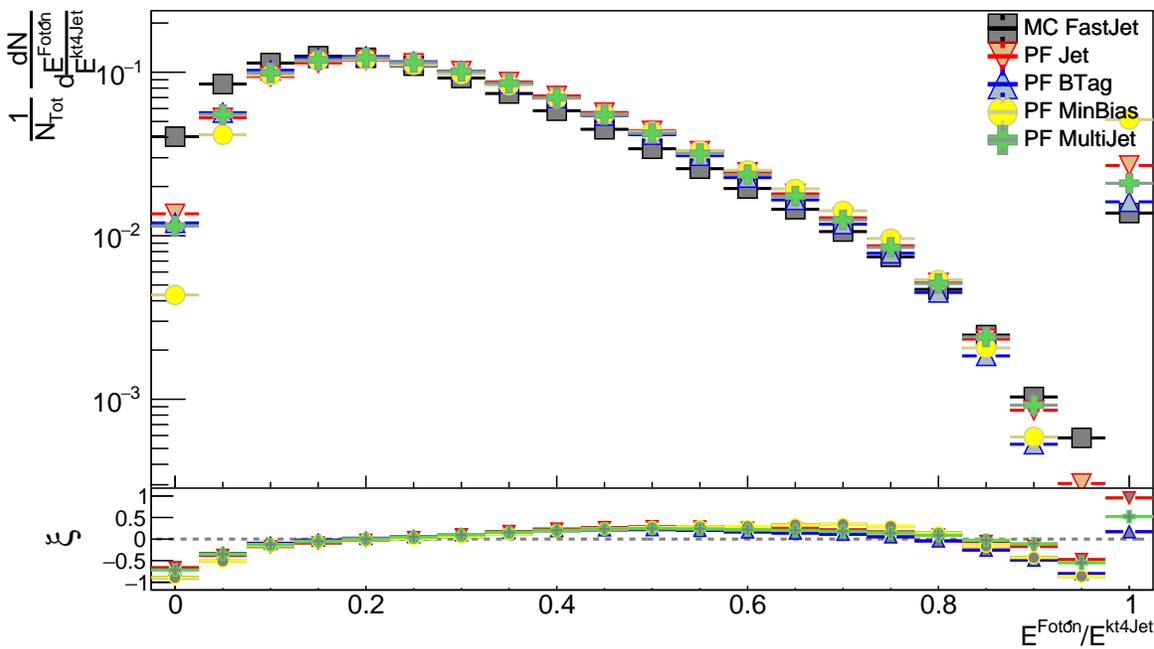


Figura 3.8: Comportamiento de la FF para fotones contenidos en kt4Jets.

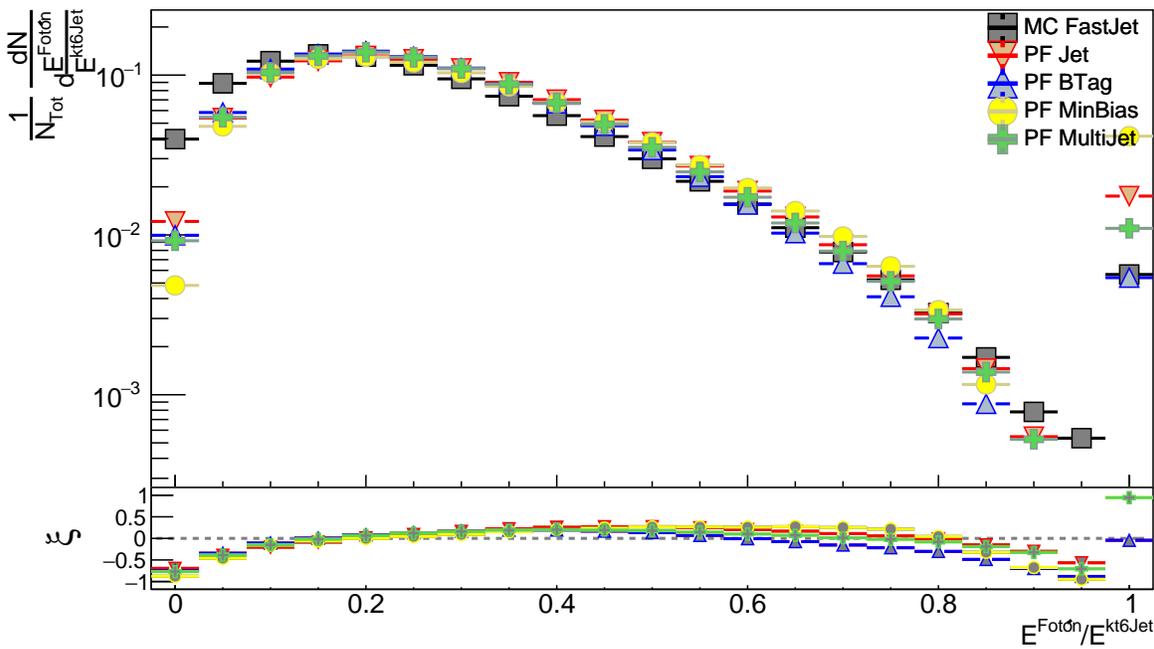


Figura 3.9: Comportamiento de la FF para fotones contenidos en kt6Jets.

En la figura 3.6 notamos que los jets MC se ajusta de buena manera a los conjuntos de datos abiertos, en especial, dentro del rango de $0.05 \leq \frac{E^{\text{Foton}}}{E^{\text{Jet}}} \leq 0.90$, una tendencia que se repite cuando consideramos los demás algoritmos de reconstrucción de jets (figuras 3.7, 3.8 y 3.9). Entonces, continuamos con mostrar el comportamiento de esta FF en algunos rangos de energía del jet.

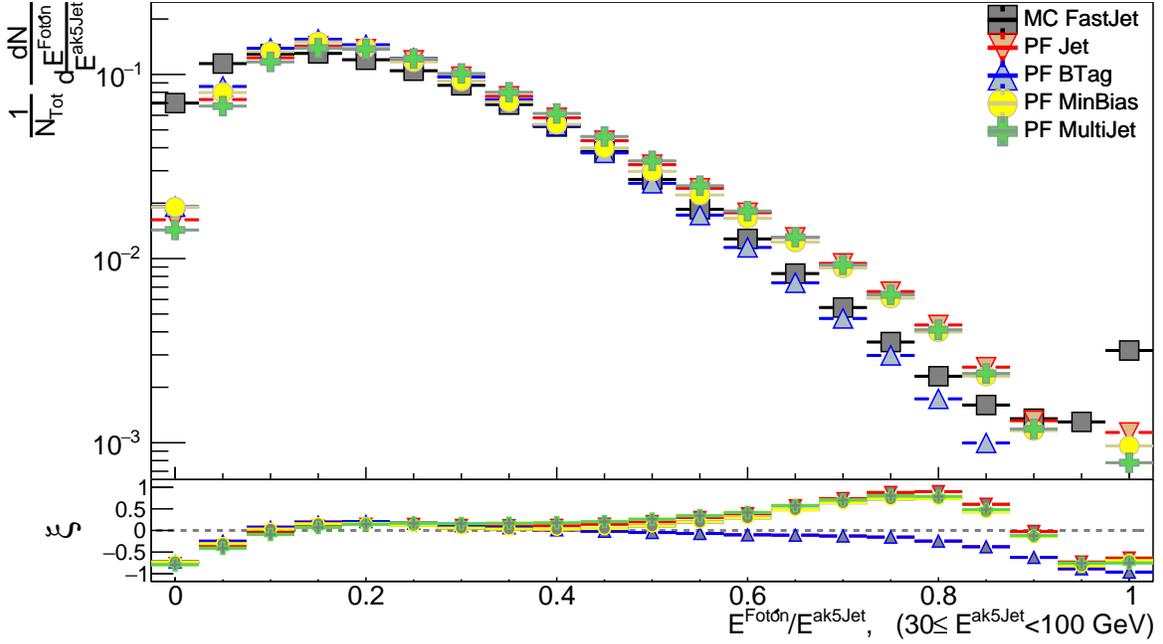


Figura 3.10: Función de fragmentación para fotones contenidos en `ak5Jets` cuya energía $\in [30, 100)$ GeV.

Las figuras 3.10, 3.11 y 3.12 muestran la fragmentación de jets en fotones con jets cuya energía $\in [30, 100)$ GeV, en todas estas figuras existe una concordancia entre los jets MC con el conjunto *BTag*. La primera es respecto al algoritmo *ak5* y sucede en el rango de $0.40 \leq \frac{E^{\text{Foton}}}{E^{\text{Jet}}} \leq 0.85$. La segunda figura muestra *kt4Jets* y el rango es $0.30 \leq \frac{E^{\text{Foton}}}{E^{\text{Jet}}} \leq 0.90$. La tercera es con el algoritmo *kt6* en $0.45 \leq \frac{E^{\text{Foton}}}{E^{\text{Jet}}} \leq 0.85$. Con esto podemos concluir que en el rango $0.45 \leq \frac{E^{\text{Foton}}}{E^{\text{Jet}}} \leq 0.85$, para jets cuyo radio es menor a 7, la fragmentación en fotones MC es consistente con el conjunto de datos abiertos *BTag*, el cual es reconstruido a partir de la hadronización de *quarks b*, independientemente del algoritmo de reconstrucción de jets considerado.

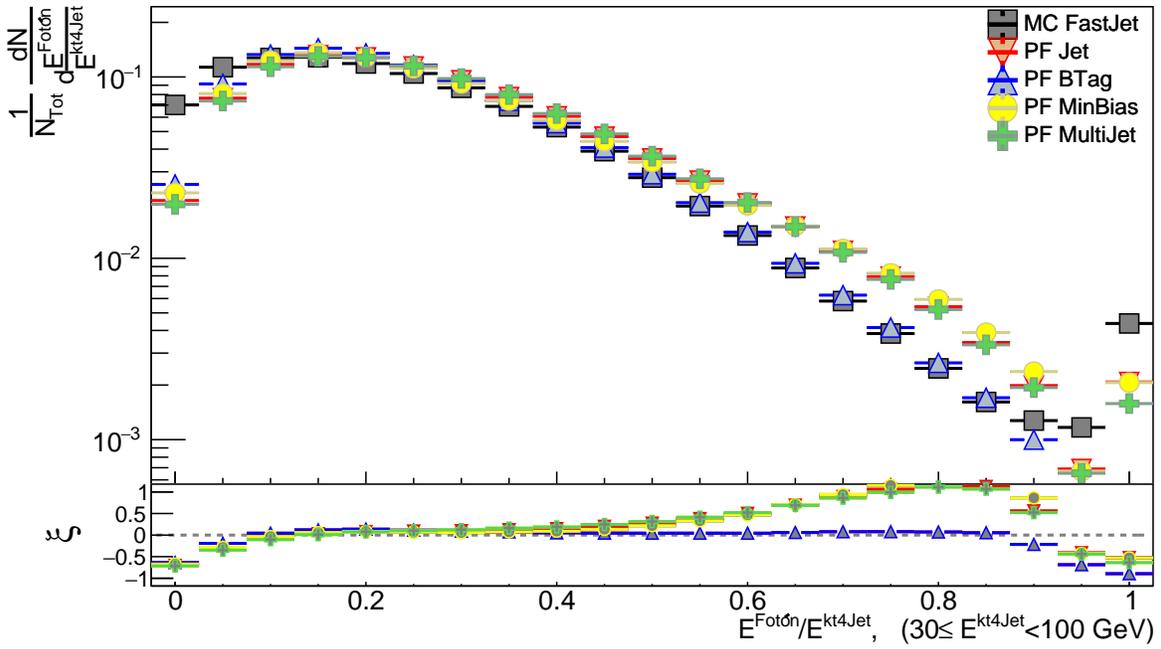


Figura 3.11: Función de fragmentación para fotones contenidos en kt4Jets cuya energía $\in [30, 100)$ GeV.

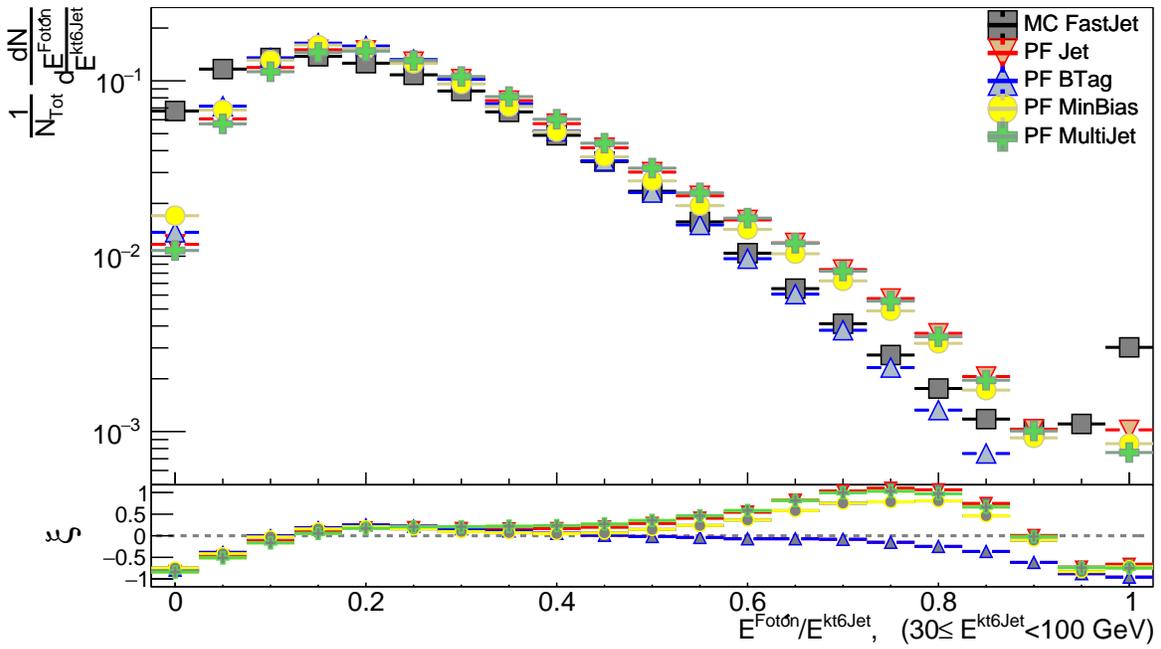


Figura 3.12: Función de fragmentación para fotones contenidos en kt6Jets cuya energía $\in [30, 100)$ GeV.

Por otra parte, la tabla 3.1 muestra los rangos, en $\frac{E^{\text{Foton}}}{E^{\text{Jet}}}$ para jets cuya energía $100 \leq E^{\text{Jet}} < 500$ GeV, en los cuales la simulación es consistente con el conjunto de datos *MultiJets*. Es decir, en estos rangos los eventos predominantes son en los que tenemos 1 jet con $p_T > 250$ GeV en $|\eta| < 1.3$, balanceado por un sistema de retroceso compuesto por dos o más jets con $p_T \in (25, 750)$ GeV [27].

Algoritmo	Rango de $\frac{E^{\text{Foton}}}{E^{\text{Jet}}}$	Ver figura
ak5	[0.70,0.90]	3.13
ak7	[0.55,0.85]	3.14
kt4	[0.70,0.90]	3.15
kt6	[0.60,0.90]	3.16

Tabla 3.1: Rangos en $\frac{E^{\text{Foton}}}{E^{\text{Jet}}}$ de consistencia de jets MC con el conjunto *MultiJet*, donde $100 \leq E^{\text{Jet}} < 500$ GeV.

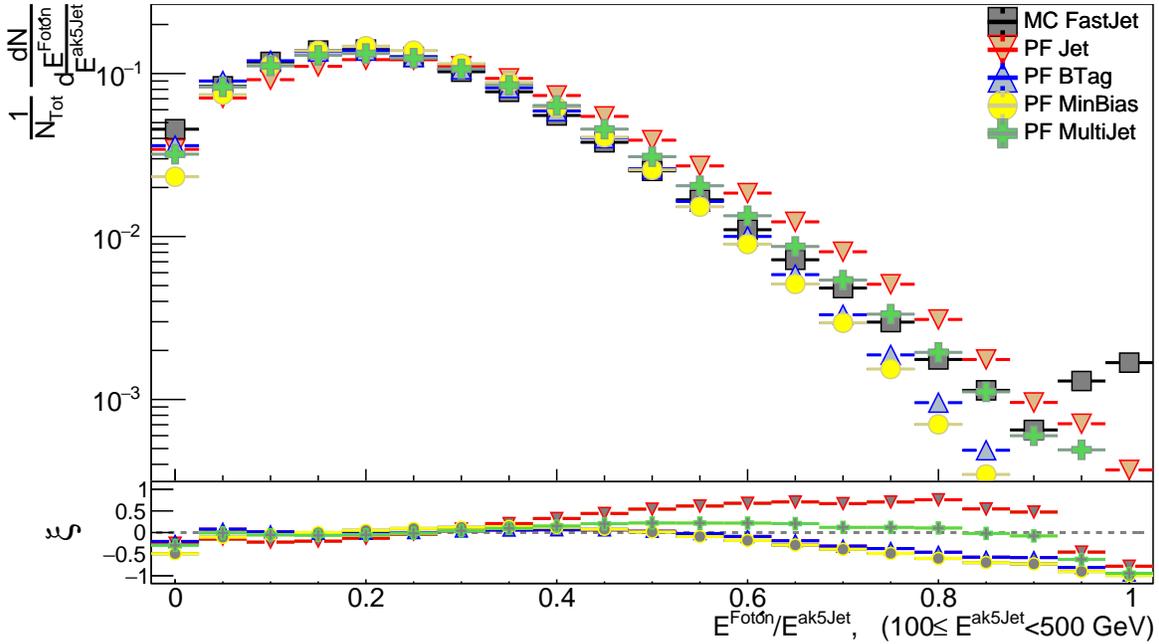


Figura 3.13: Función de fragmentación para fotones contenidos en *ak5Jets* cuya energía $\in [100, 500)$ GeV.

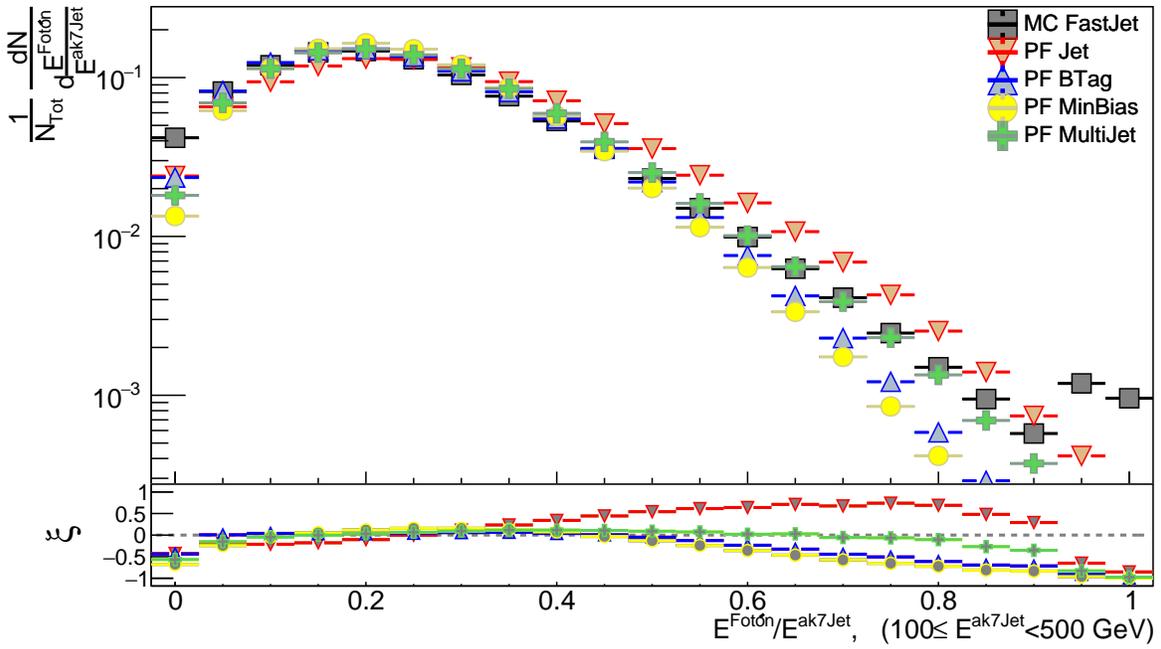


Figura 3.14: Función de fragmentación para fotones contenidos en ak7Jets cuya energía $\in [100, 500)$ GeV.

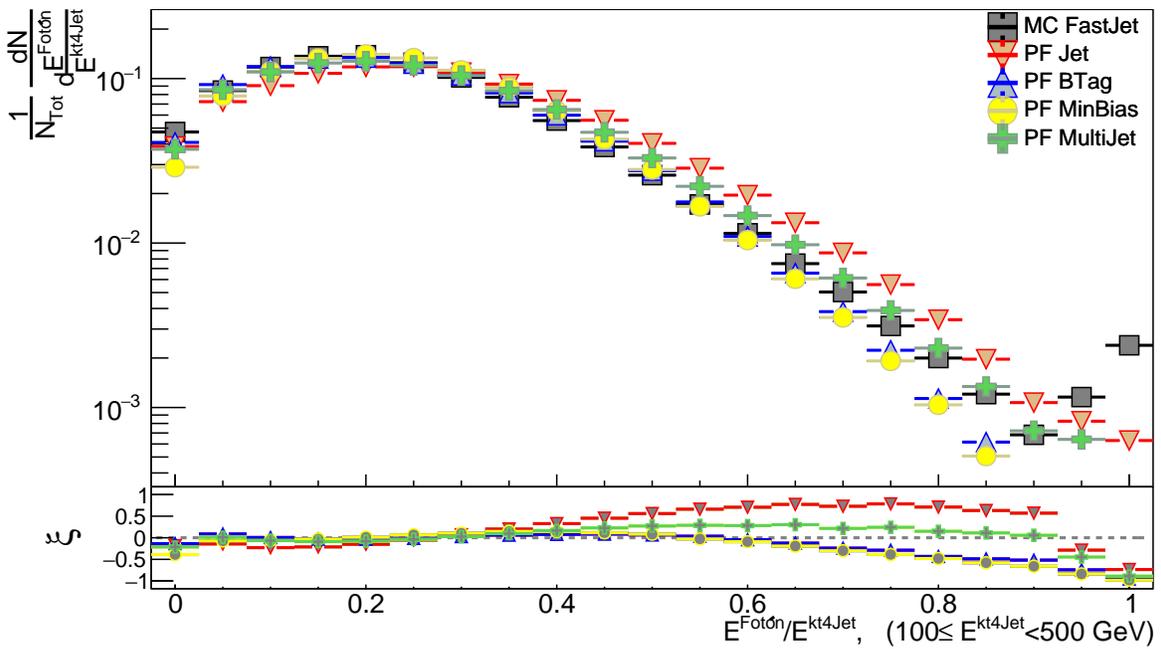


Figura 3.15: Función de fragmentación para fotones contenidos en kt4Jets cuya energía $\in [100, 500)$ GeV.

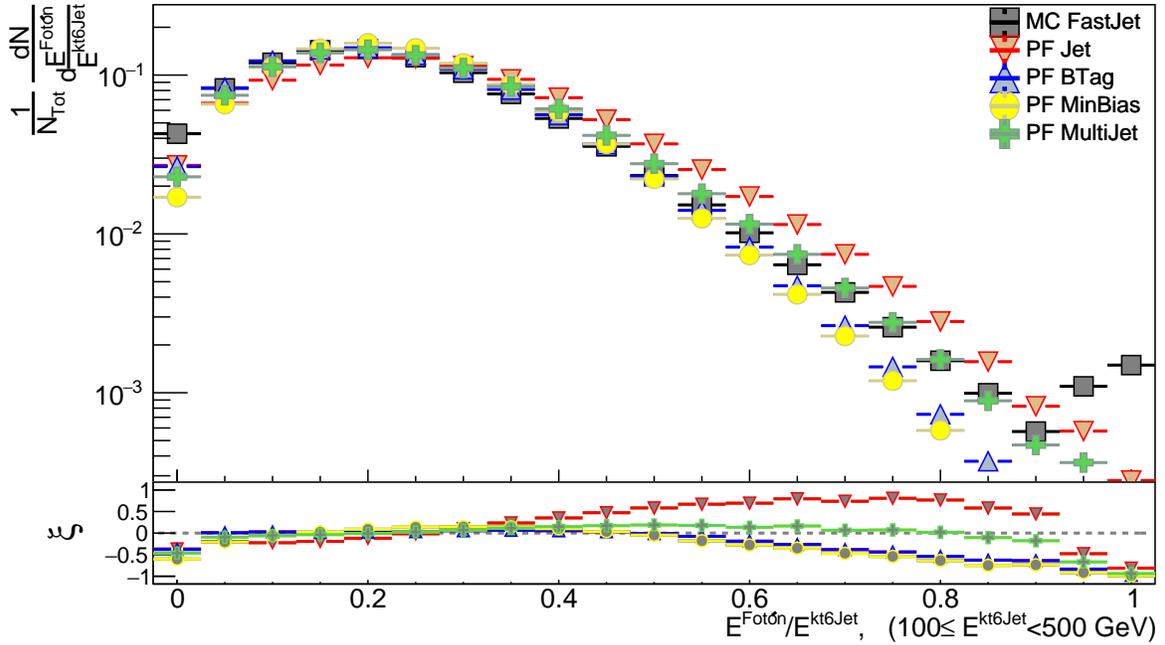


Figura 3.16: Función de fragmentación para fotones contenidos en $kt6Jets$ cuya energía $\in [100, 500)$ GeV.

3.1.3. Electrones

Ahora pasamos a mostrar la FF correspondiente a los electrones que se muestra en la figura 3.17, en la cual se considera el algoritmo $ak5$. Como podemos notar al considerar la variable ξ , la simulación se ajusta pobremente a los conjuntos de datos. Sin embargo, las figuras 3.18 y 3.19 desglosan en diferentes rangos de energía la distribución de la FF para electrones. Con esto, resulta evidente que para jets con energías menores a 100 GeV, la FF se ajusta bien a los conjuntos de datos, sin concordar exactamente con ninguno de ellos. Por otra parte, para jets con energías mayores a 100 GeV notamos que la simulación tiene un comportamiento notablemente distinto al de los conjuntos de datos.

El mismo comportamiento podemos encontrar al considerar los otros tres algoritmos de reconstrucción de jets. Sin embargo, no son mostrados explícitamente en este capítulo por no aportar nada nuevo. En cualquier caso, el desglose completo para cualquier algoritmo de reconstrucción puede ser encontrado en el apéndice C.2.

Finalmente, podemos concluir que la fragmentación en electrones es descrita ade-

cuadramente por la simulación para jets con energías menores a 100 GeV.

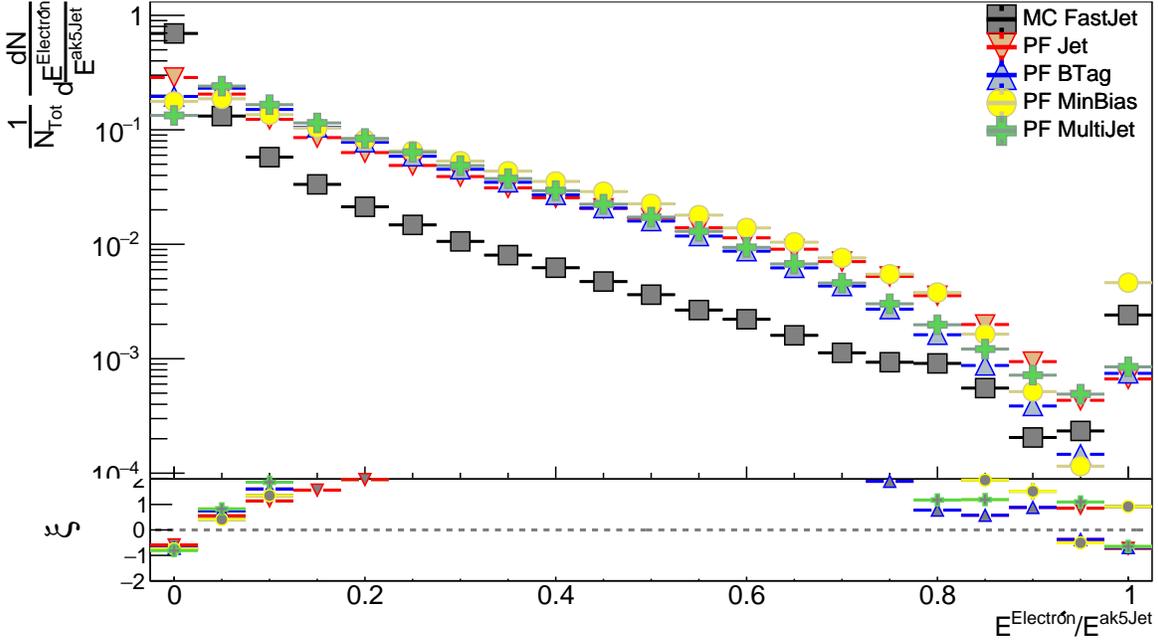


Figura 3.17: Función de fragmentación para electrones contenidos en ak5Jets.

3.1.4. Hadrones neutros

En la figura 3.20 se muestra la distribución de la FF para los hadrones neutros contenidos en ak5Jets. Lo primero que notamos es que la simulación se ajusta relativamente bien en todo el dominio de $\frac{E^{\text{HadNeutro}}}{E^{\text{Jet}}}$ al conjunto *BTag*, tendencia que se confirma al considerar la figura 3.21 correspondiente a ak7Jets. Por contraparte, resulta evidente que el conjunto peor descrito por la simulación es el *MinBias*.

Si analizamos la FF para jets con energías entre $100 \leq E^{\text{Jet}} < 400$ GeV, tenemos que para los algoritmos *ak5* (figura 3.22) y *kt4* (figura 3.24) la simulación concuerda con el conjunto *MultiJet* en el rango $0.55 \leq \frac{E^{\text{HadNeutro}}}{E^{\text{Jet}}} < 1.0$. Del mismo modo, para los algoritmos *ak7* (figura 3.23) y *kt6* (figura 3.25) esta concordancia se da en el rango $0.50 \leq \frac{E^{\text{HadNeutro}}}{E^{\text{Jet}}} < 1.0$. Esto quiere decir que los eventos predominantes son en los que tenemos por lo menos 1 jet con $p_T > 250$ GeV en $|\eta| < 1.3$, balanceado por un

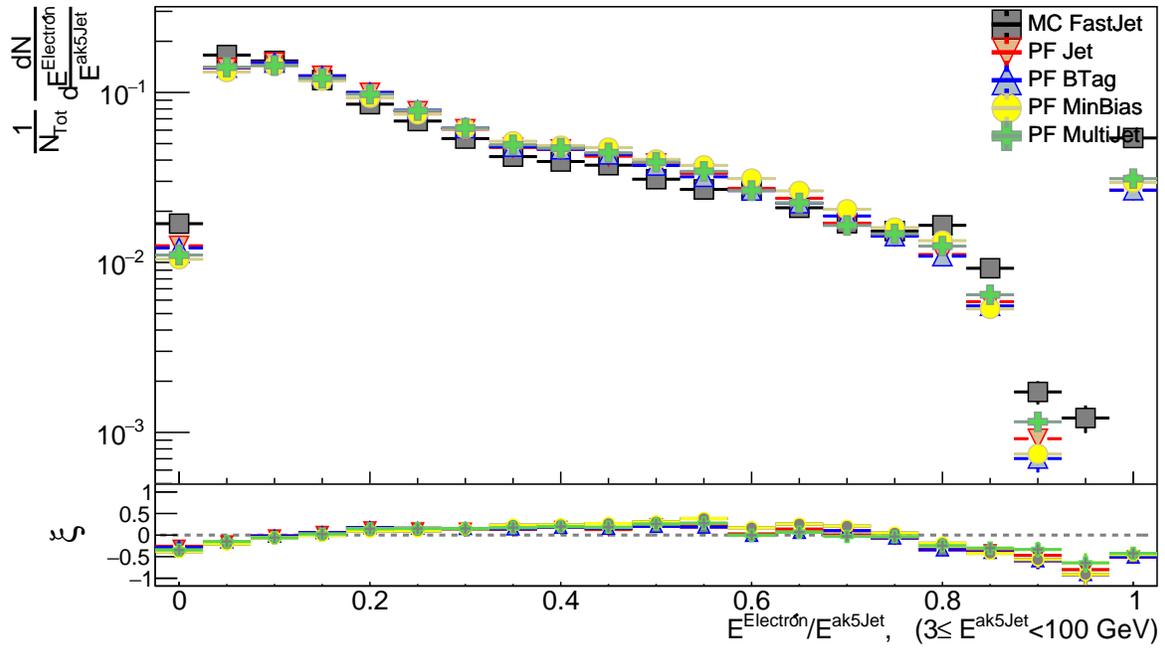


Figura 3.18: Función de fragmentación para electrones contenidos en $ak5Jets$ cuya energía $\in (0, 100)$ GeV.

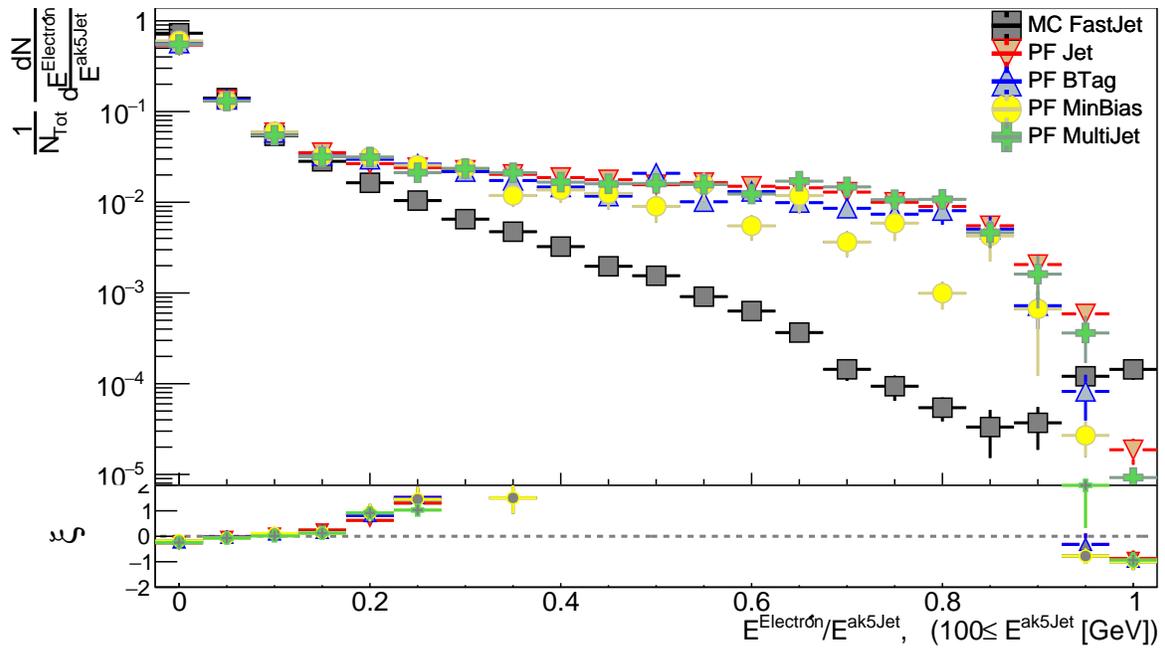


Figura 3.19: Función de fragmentación para electrones contenidos en $ak5Jets$ cuya energía < 100 GeV.

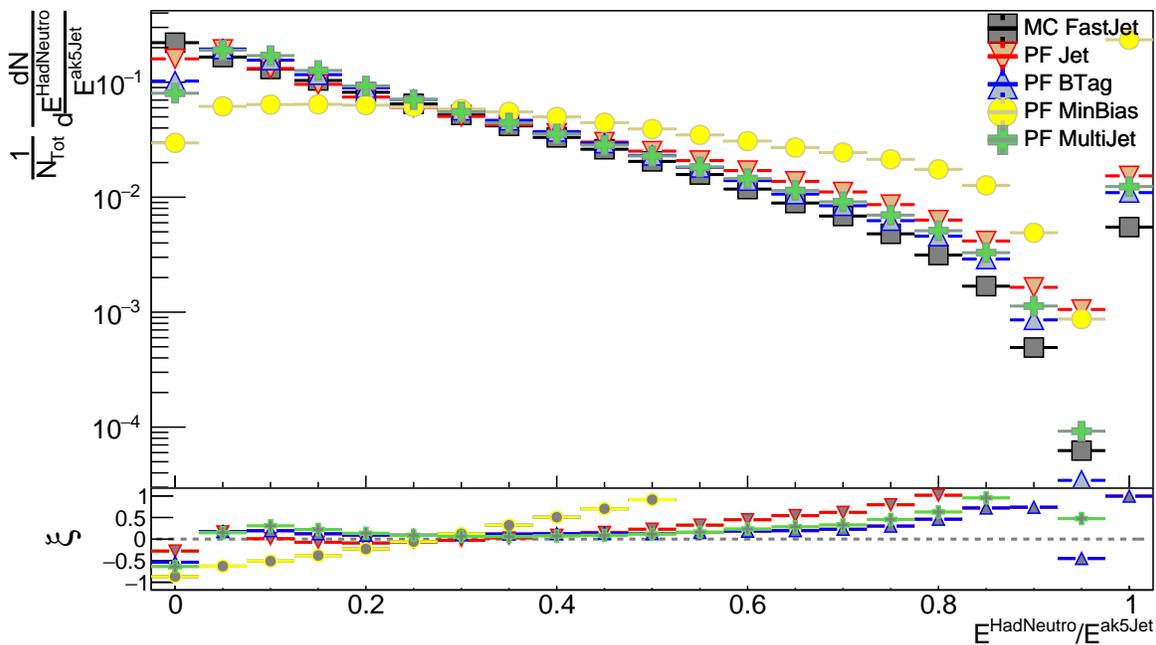


Figura 3.20: Función de fragmentación para hadrones neutros contenidos en ak5Jets.

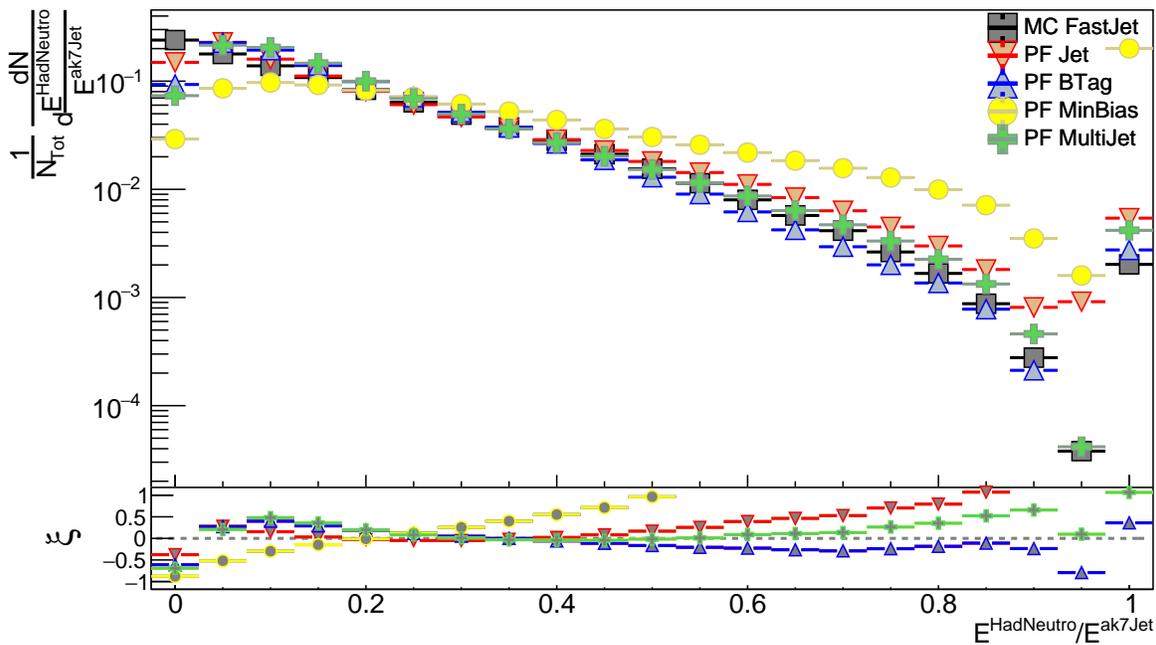


Figura 3.21: Función de fragmentación para hadrones neutros contenidos en ak7Jets.

sistema de retroceso compuesto por dos o más jets con $p_T \in (25, 750)$ GeV [27]; y más aún, esto es un comportamiento independiente del algoritmo de reconstrucción de jets utilizado.

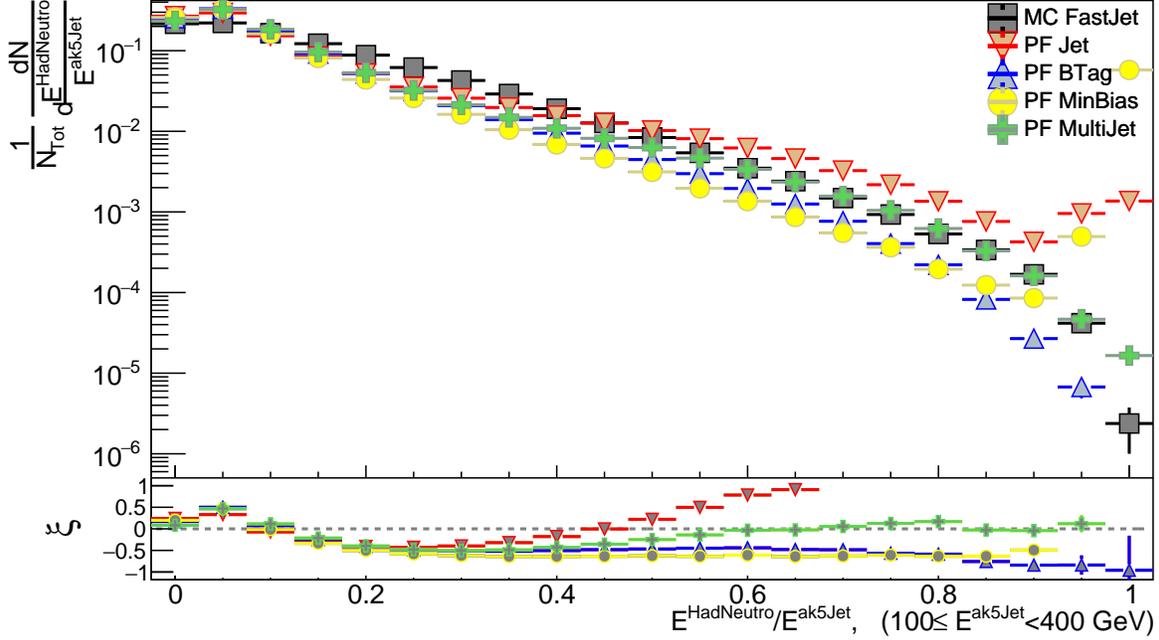


Figura 3.22: Función de fragmentación para hadrones neutros contenidos en *ak5Jets* cuya energía $\in [100, 400)$ GeV.

3.1.5. Hadrones cargados

La FF para hadrones cargados se muestra en las figuras 3.26, 3.27, 3.28 y 3.29 para los algoritmos *ak5*, *ak7*, *kt4* y *kt6*, respectivamente. Es de resaltar que en general los jets MC se ajustan de buena manera a los cuatro conjuntos de datos abiertos para todo el dominio de $\frac{E^{\text{HadCargado}}}{E^{\text{Jet}}}$, y en particular el conjunto mejor descrito es *Jet*. Por lo que se puede inferir que la simulación describe adecuadamente la fragmentación de jets en hadrones cargados.

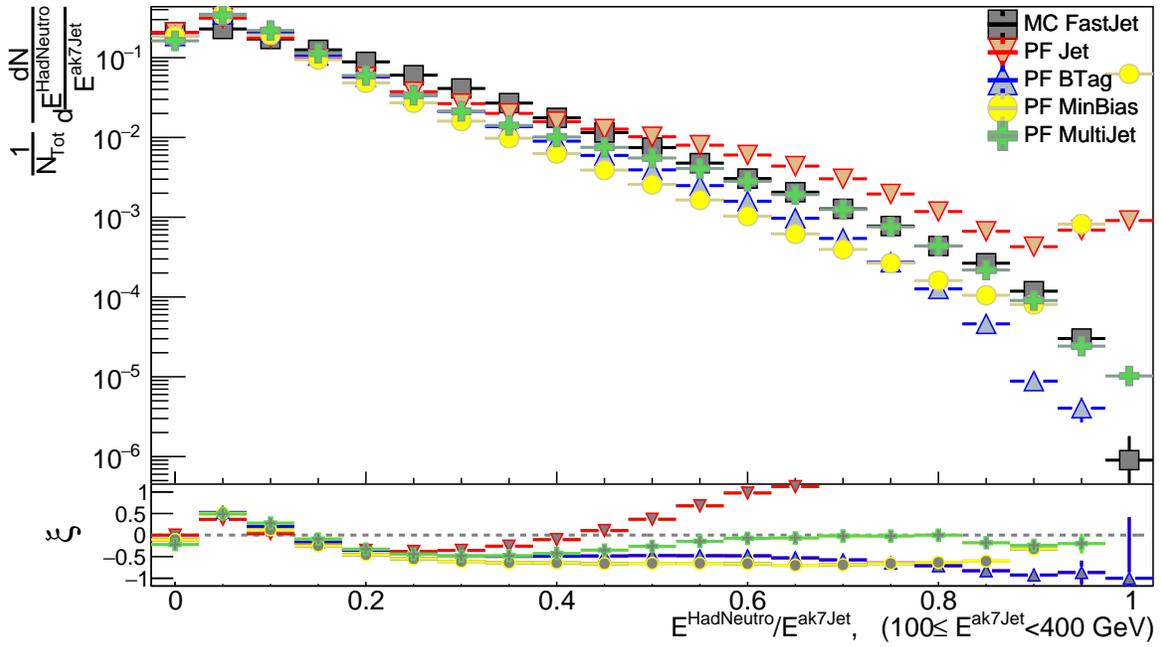


Figura 3.23: Función de fragmentación para hadrones neutros contenidos en ak7Jets cuya energía $\in [100, 400)$ GeV.

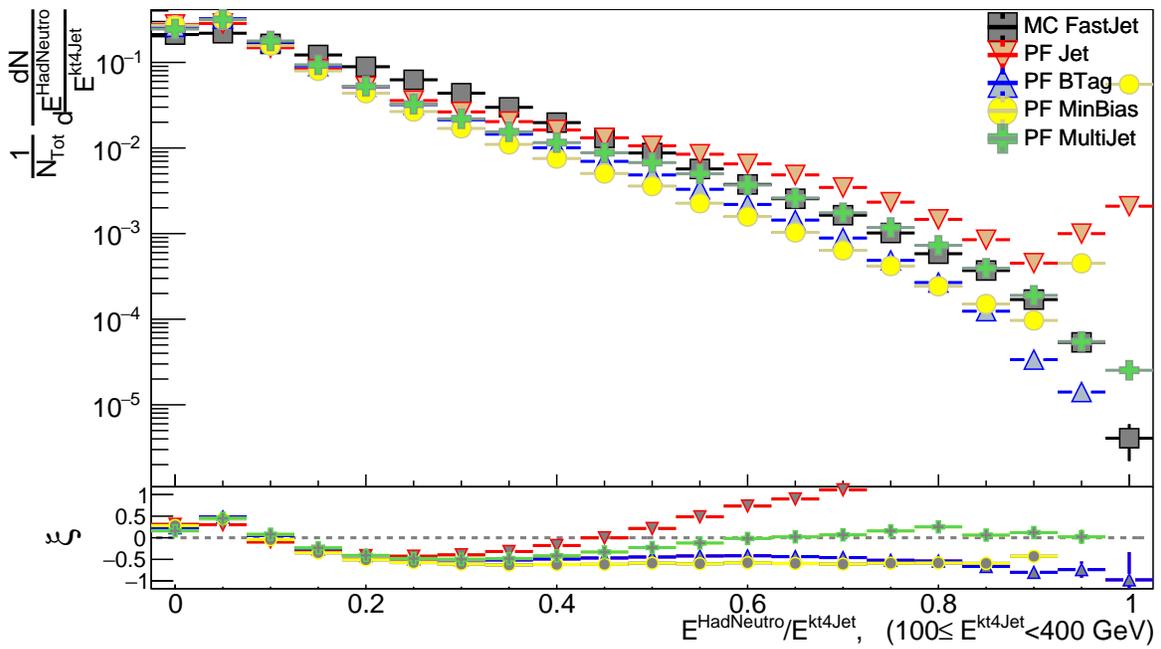


Figura 3.24: Función de fragmentación para hadrones neutros contenidos en kt4Jets cuya energía $\in [100, 400)$ GeV.

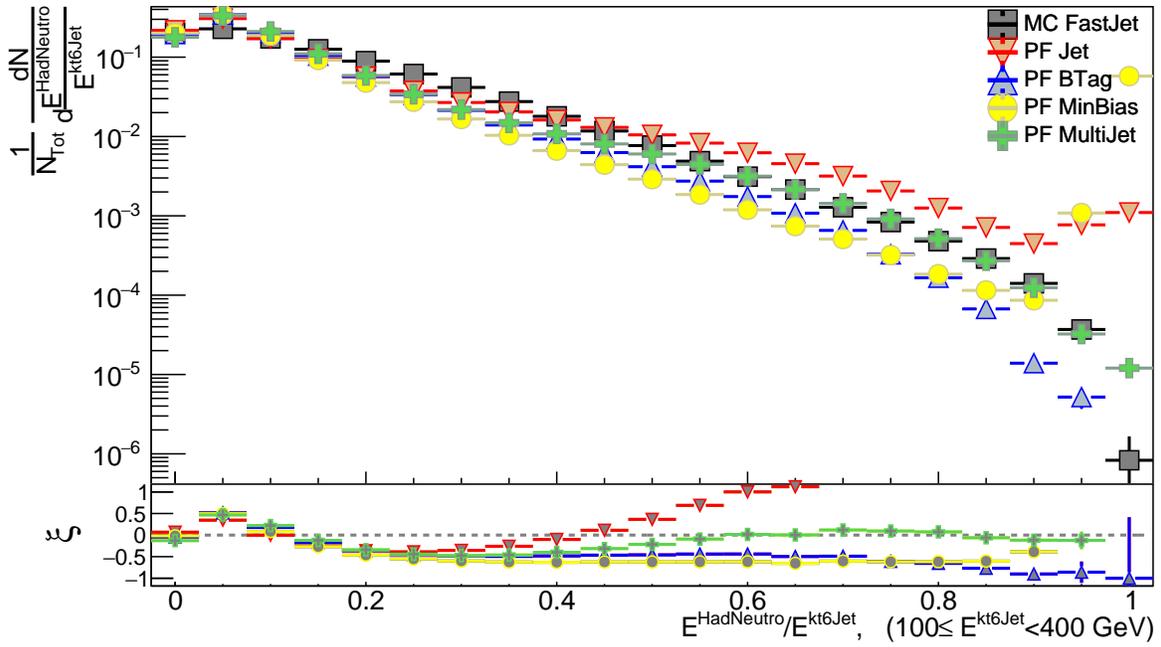


Figura 3.25: Función de fragmentación para hadrones neutros contenidos en kt6Jets cuya energía $\in [100, 400)$ GeV.

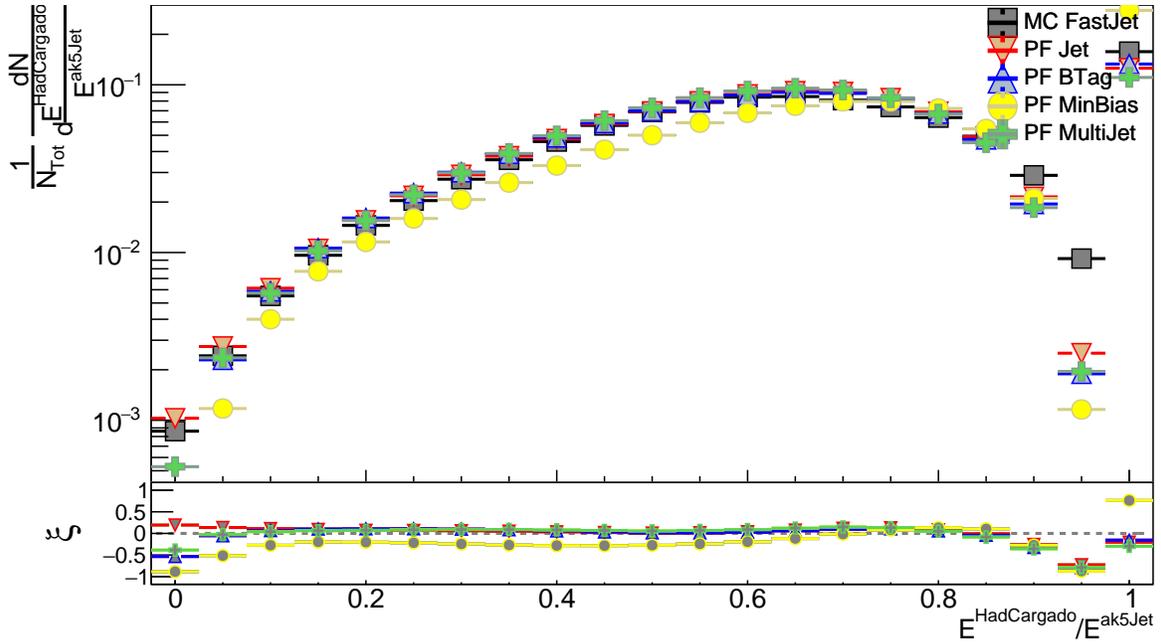


Figura 3.26: Función de fragmentación para hadrones cargados contenidos en ak5Jets.

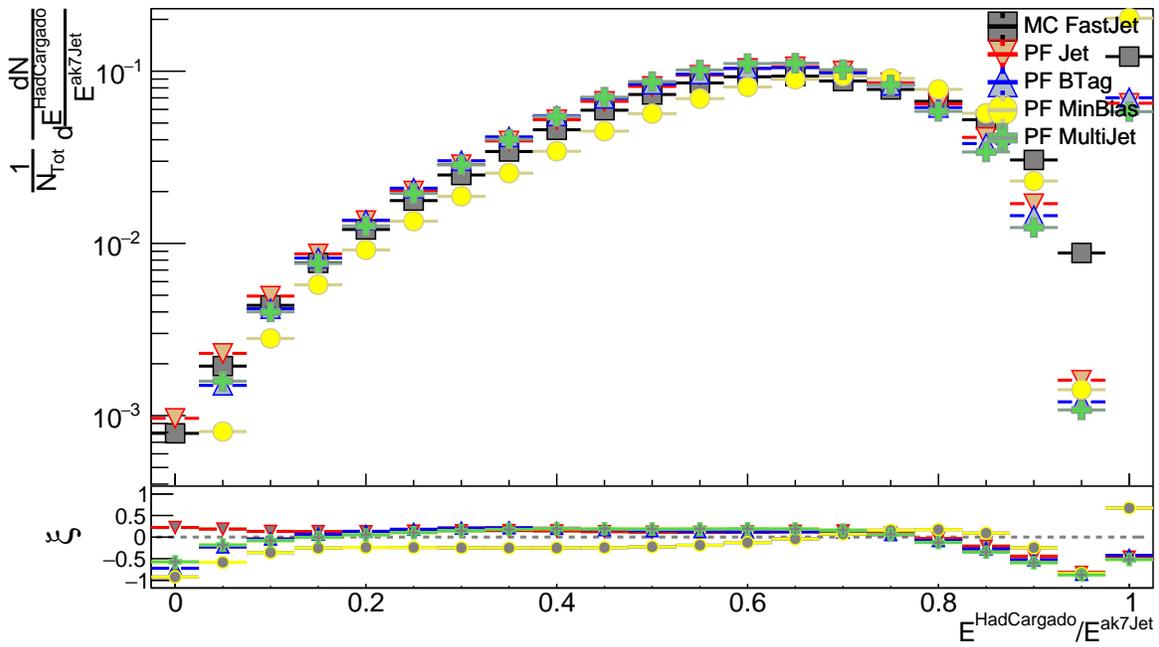


Figura 3.27: Función de fragmentación para hadrones cargados contenidos en ak7Jets.

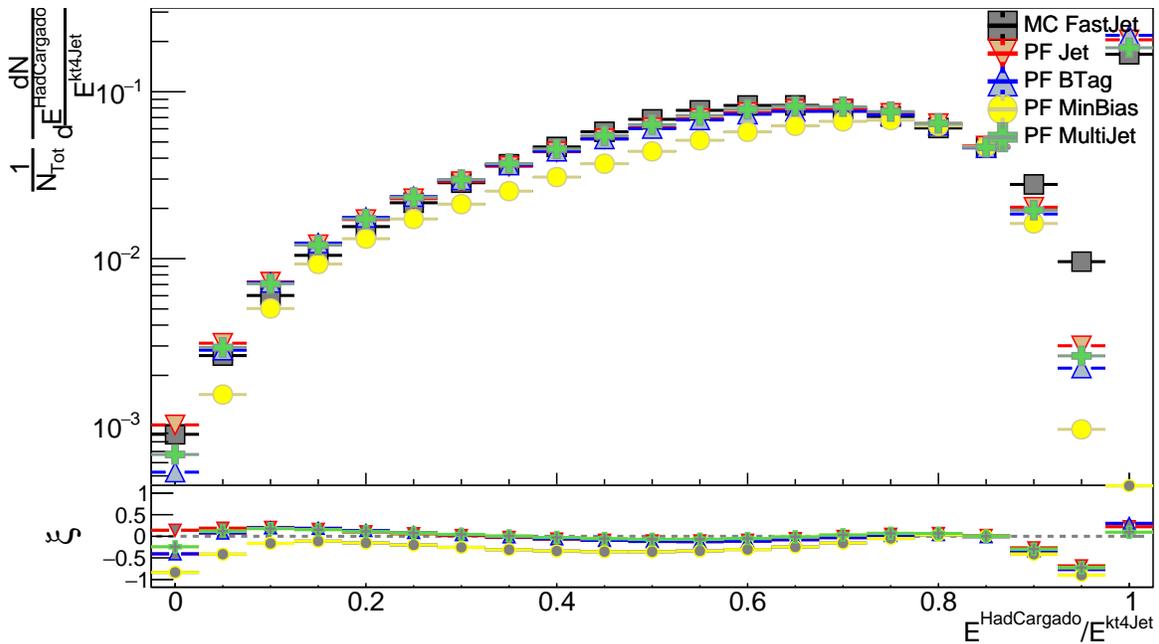


Figura 3.28: Función de fragmentación para hadrones cargados contenidos en kt4Jets.

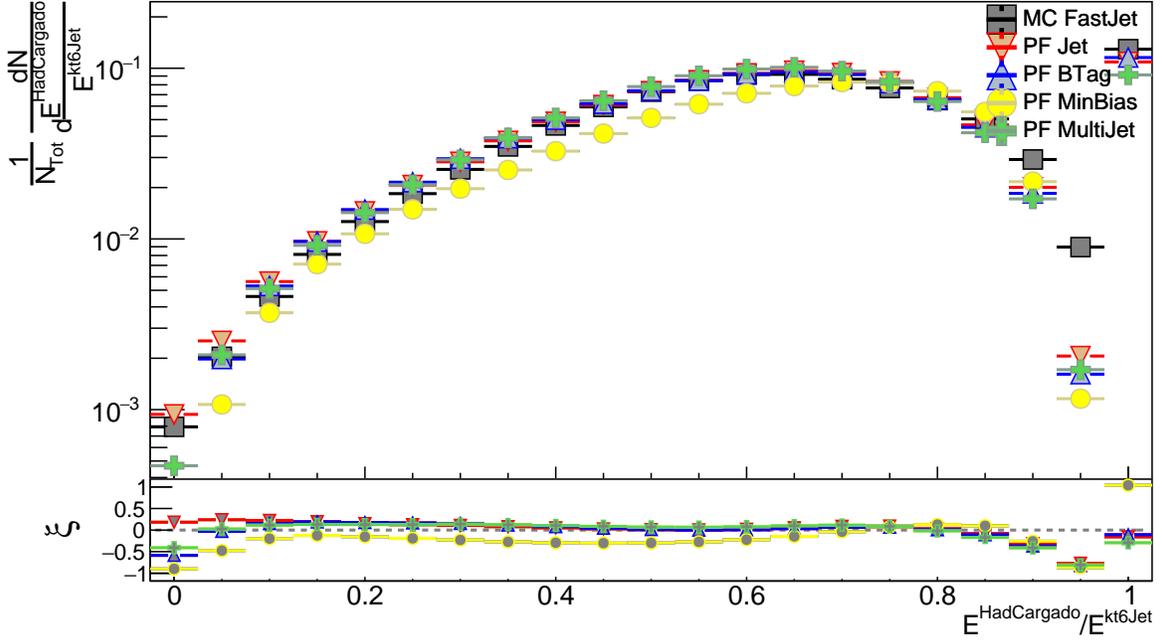


Figura 3.29: Función de fragmentación para hadrones cargados contenidos en $kt6Jets$.

Por último, para jets con energías entre $[60,800)$ GeV la tabla 3.2 muestra el rango en $\frac{E^{HadCargado}}{E^{Jet}}$, para el cual la simulación se ajusta mejor al conjunto *MinBias* dependiendo del algoritmo de reconstrucción utilizado. Con lo que en este rango los hadrones cargados dentro de jets MC concuerdan con aquellos que provienen de eventos inelásticos difractivos reales.

Algoritmo	Rango de $\frac{E^{HadCargado}}{E^{Jet}}$	Ver figura
<i>ak5</i>	[0.05,0.35]	3.30
<i>ak7</i>	[0.05,0.25]	3.31
<i>kt4</i>	[0.05,0.35]	3.32
<i>kt6</i>	[0.05,0.30]	3.33

Tabla 3.2: Rangos en $\frac{E^{HadCargado}}{E^{Jet}}$ de consistencia de jets MC con el conjunto *MultiJet*, donde $60 \leq E^{Jet} < 800$ GeV.

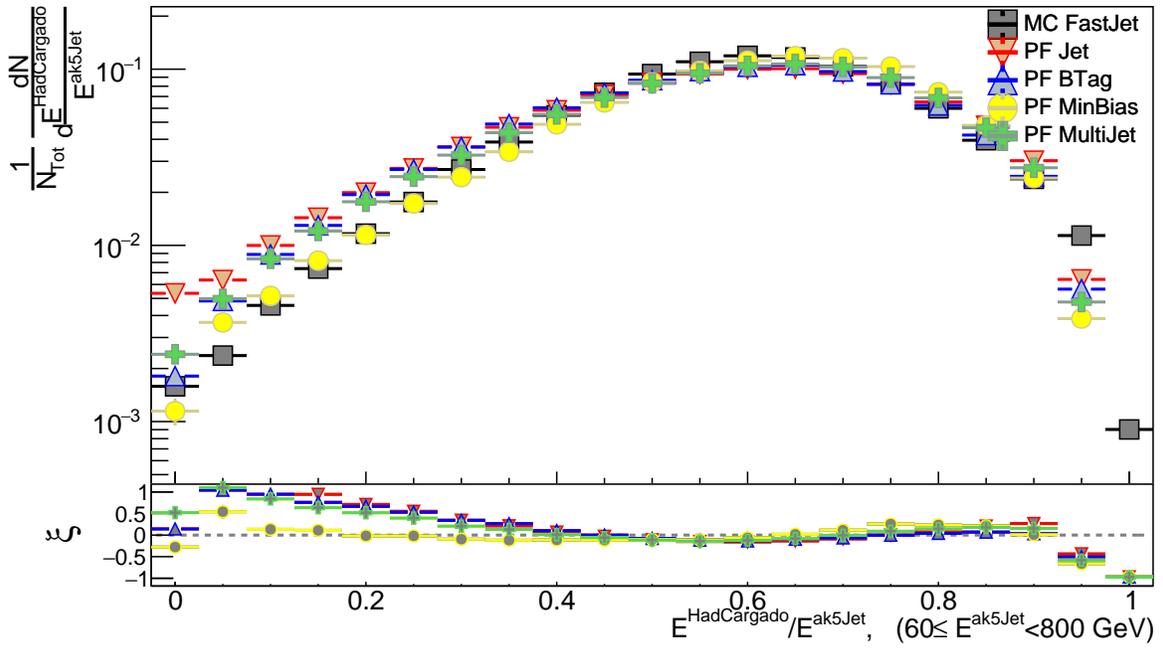


Figura 3.30: Función de fragmentación para hadrones cargados contenidos en $ak5Jets$ cuya energía $\in [60, 800)$ GeV.

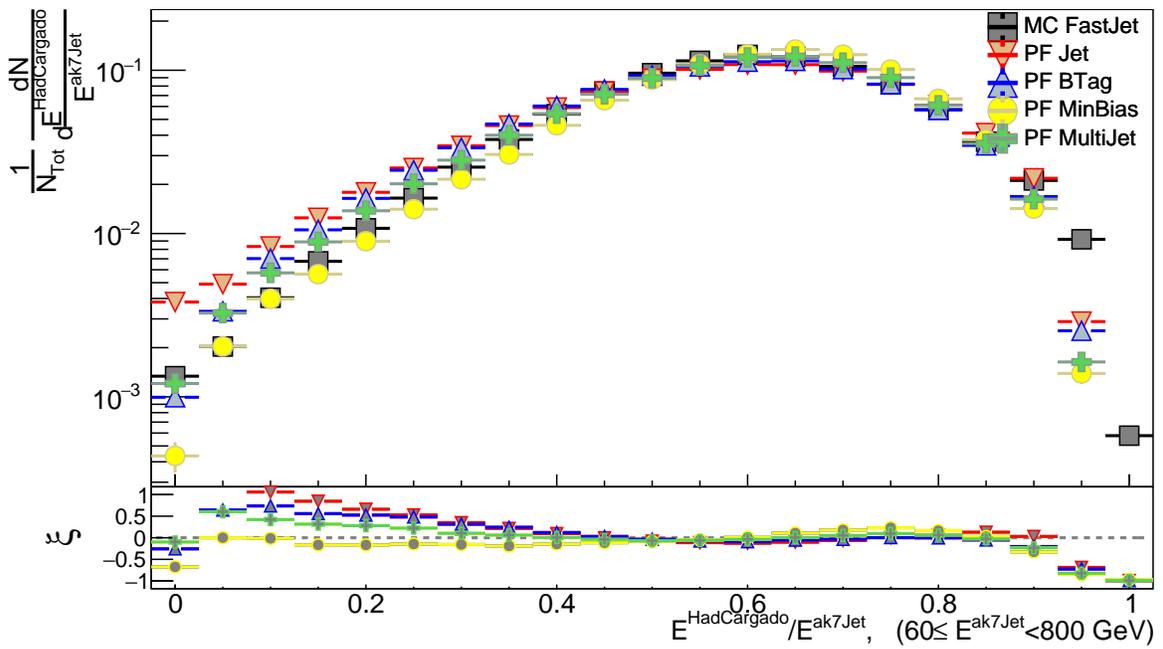


Figura 3.31: Función de fragmentación para hadrones cargados contenidos en $ak7Jets$ cuya energía $\in [60, 800)$ GeV.

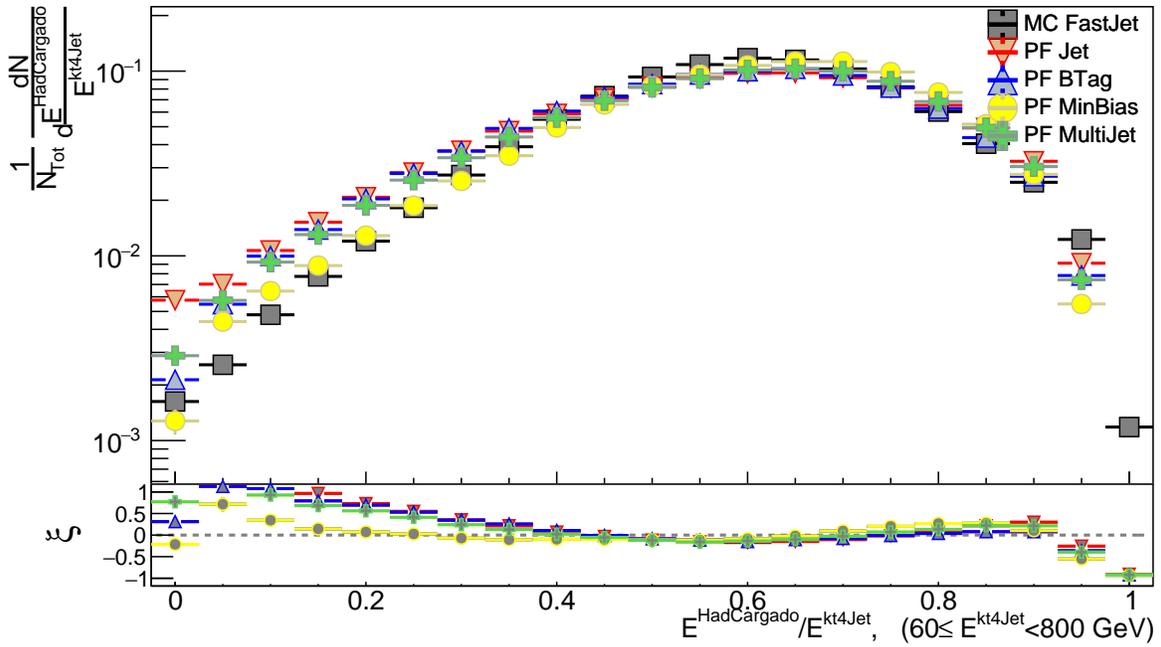


Figura 3.32: Función de fragmentación para hadrones cargados contenidos en $kt4Jets$ cuya energía $\in [60, 800)$ GeV.

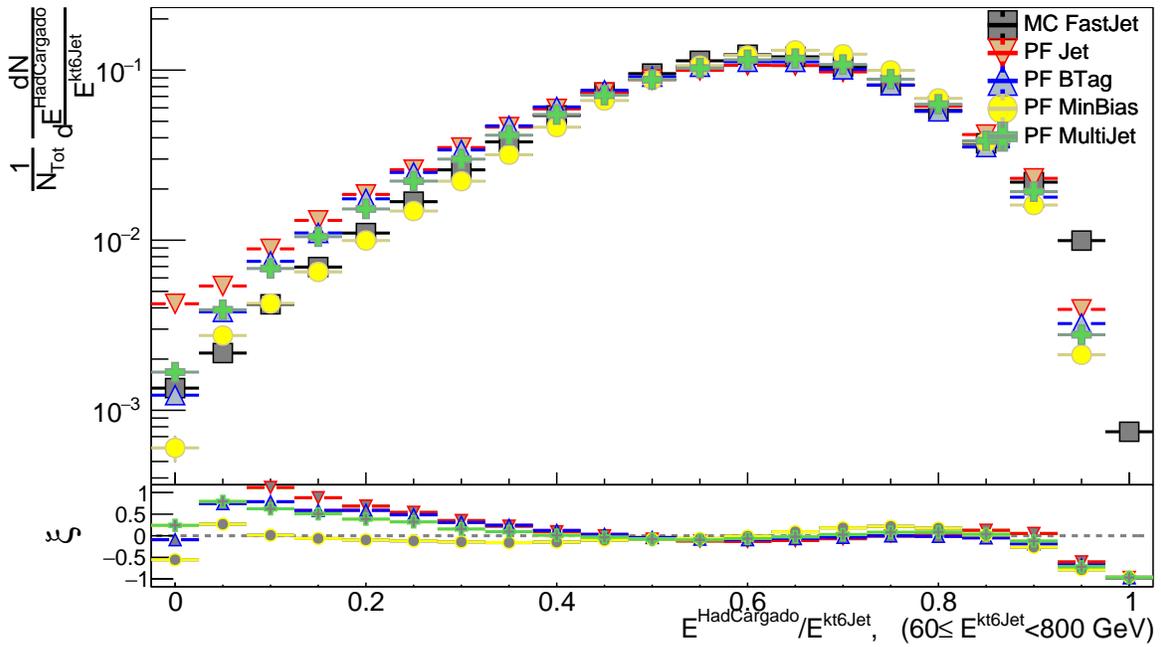


Figura 3.33: Función de fragmentación para hadrones cargados contenidos en $kt6Jets$ cuya energía $\in [60, 800)$ GeV.

Capítulo 4

Conclusiones

Lo primero que podemos resaltar es que ha sido posible desarrollar una metodología completa para el análisis de datos abiertos de LHC. Al igual que sucede con la proveída por el CERN, ésta metodología de análisis es accesible para cualquier persona con una computadora y acceso a internet. Sin embargo, presenta tras ventajas principales: primeramente, los tiempos de cómputo se reducen significativamente, pues el almacenamiento y análisis de datos se hace de manera local; seguidamente, todos los *macros* están disponibles a través de GitHub; por último, estos *macros* son aptos para ser modificados y obtener información de cualquier rama de los AODs, no sólo de las correspondientes a los jets. Por otra parte, cabe resaltar que para el procesamiento y análisis de datos no se ha tomado en cuenta la respuesta (*unfolding*¹) del detector ni la estimación de incertidumbres sistemáticas pues CMS no provee esta información en su portal de datos abiertos y acorde con [71], los conjuntos de datos son lo suficientemente adecuados para ser analizados directamente; además de que los AODs han sido desarrollados para su análisis directo.

En segundo término, es de notar que se ha logrado obtener una simulación lo suficientemente aceptable para poder ser comparada con los conjuntos de datos abiertos utilizados, pese a no haber utilizado *tunings*² para Pythia 8 (pues esto estaría más

¹Las distribuciones medidas en física de partículas están distorsionadas por la resolución y la aceptación de los detectores. La transformación a la distribución verdadera se llama *unfolding* [72].

²Los *tunings* son parámetros técnico y especializados de Pythia. Estos se pueden variar independientemente, pero los requisitos físicos conducen a correlaciones y anti-correlaciones entre ellos.

allá de los propósitos de esta tesis). Esto ha sido crucial al estudiar la fragmentación de jets utilizando la FF definida.

En tercer lugar, al estudiar los resultados de la simulación en términos de los datos reales, tenemos que los resultados obtenidos son capaces de describir la fragmentación de los jets para todas las partículas contenidos en ellos, excepto para los electrones. La mayoría de estos resultados concuerdan al considerar al menos 3 algoritmos de reconstrucción de jets distintos. Con lo que es posible concluir que son comportamientos generales para la fragmentación de jets. Esto es importante porque de este modo podemos estudiar la fragmentación de jets mediante nuestra FF y discernir en qué rangos de $\frac{E^{\text{Particula}}}{E^{\text{Jet}}}$ la fragmentación implementada en el MC corresponde a lo que se observa en los datos.

Si bien los resultados pueden parecer poco desarrollados, debemos recordar que estos son simplemente una parte de la totalidad. Cabe enfatizar que este trabajo se centró en desarrollar una metodología completa para el análisis de datos abiertos del CMS, además de ser capaces de generar una simulación consistentes con los datos abiertos. Finalmente, esto se ha unificado y presentado mediante nuestra FF. En este sentido podemos afirmar que este trabajo ha cumplido con su propósito.

Para finalizar, es de mencionar que esta tesis puede conducir a futuros trabajos si se considera la respuesta (*unfolding*) del detector, así como los errores sistemáticos en éste último. Del mismo modo es posible utilizar *tunings* de Pythia para ajustar mejor la simulación a cada conjunto de datos y como resultado dar una descripción más detallada de la fragmentación de jets.

Apéndice A

A.1. Estructura de AODs

La información contenida en los AODs puede ser entendida como se muestra en la figura A.1.

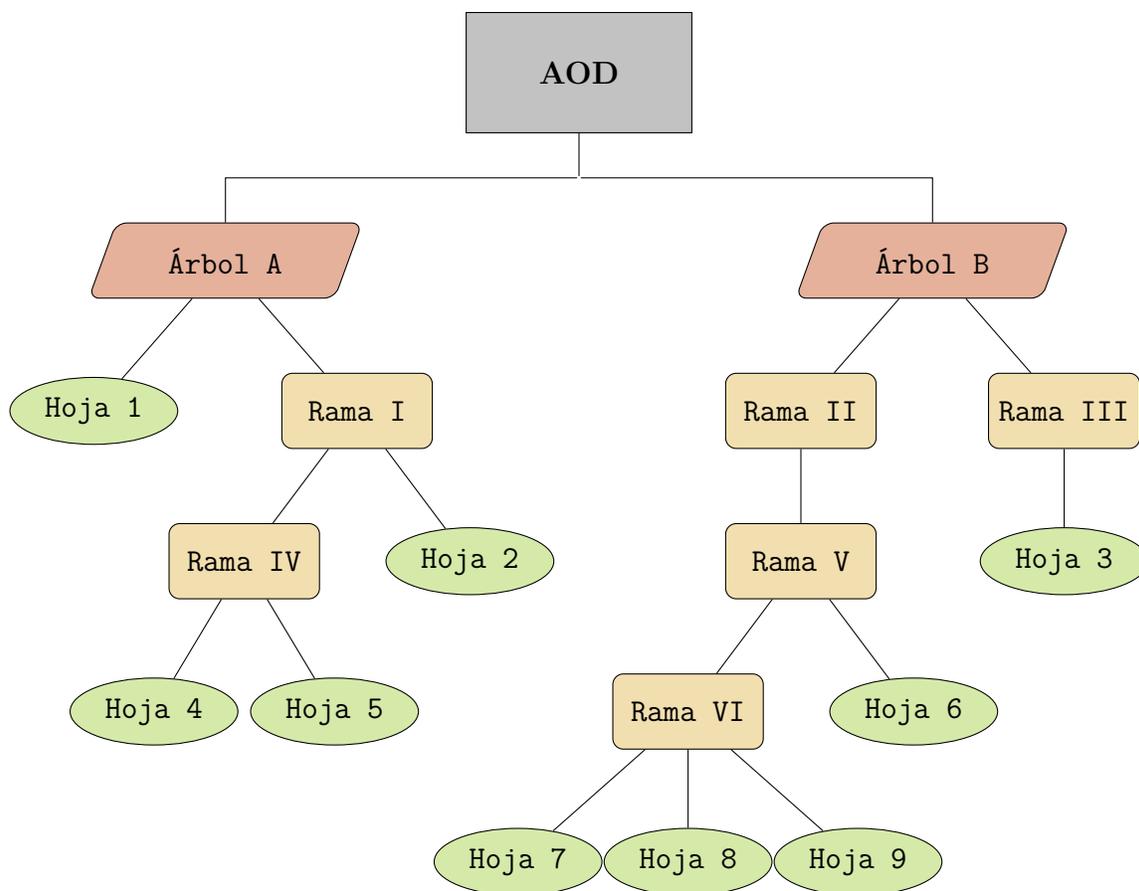


Figura A.1: Representación simplificada de la disposición de los datos en un archivo AOD.

Es importante mencionar que pueden haber más de dos árboles en cada archivo AOD. De la misma manera, podemos tener muchas más ramas dentro de cada árbol y dentro de otras ramas. Similarmente es posible tener más hojas en cada rama o árbol. En particular dentro de los AODs correspondientes a datos abiertos de CMS del año 2011 existen 6 Árboles, hasta ~ 350 ramas por árbol y un máximo de ~ 150 hojas por rama.

En esencia, son las hojas en donde se almacena la información y a ellas hay que acceder para extraer la *data*. Para lograr este objetivo, es necesario acceder a través de cada conjunto (árbol y/o rama(s)) que las contiene.

A.2. Ventana Deslizante

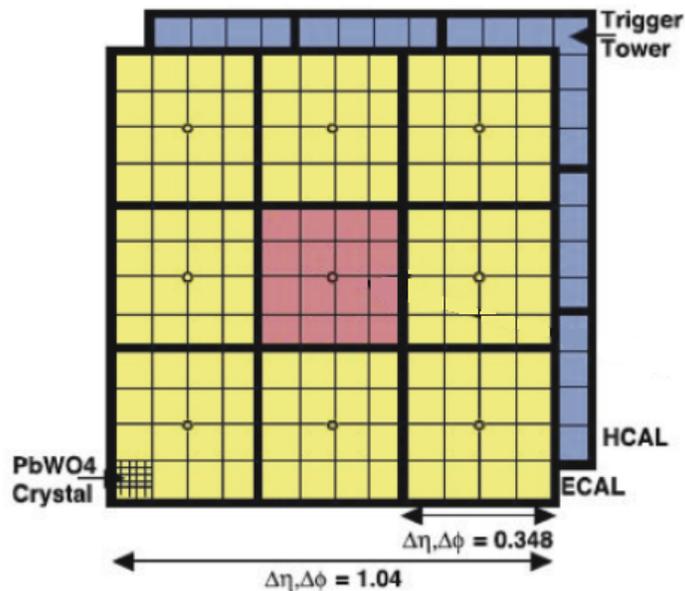


Figura A.2: Disposición del método “ventana deslizante” [9].

El *trigger* del conjunto de datos primarios Jet correspondiente al año 2011 se implementa a partir del *Trigger* de Nivel-1 y se basa en una técnica llamada “ventana deslizante”. La condición para encontrar un jet es que la región central de la matriz 3x3 que se muestra en la figura A.2, es que se necesita tener un E_T superior a cierto umbral específico y este E_T debe de ser mayor al detectado por cualquiera de sus 8 vecinos.

A.3. Área del jet

El área del jet es un concepto no trivial en la medida en que un jet consiste en partículas puntuales, las cuales por sí mismas no tienen un área intrínseca. Para definir un área uno introduce partículas de energía infinitesimal (“fantasmas”) e identifica la región en $\eta - \phi$ donde dichos fantasmas son reconstruidos dentro de un jet dado [62]. La extensión de esta región da una medida del área (adimensional) del jet. Una representación del área del jet la podemos encontrar en la figura A.3.

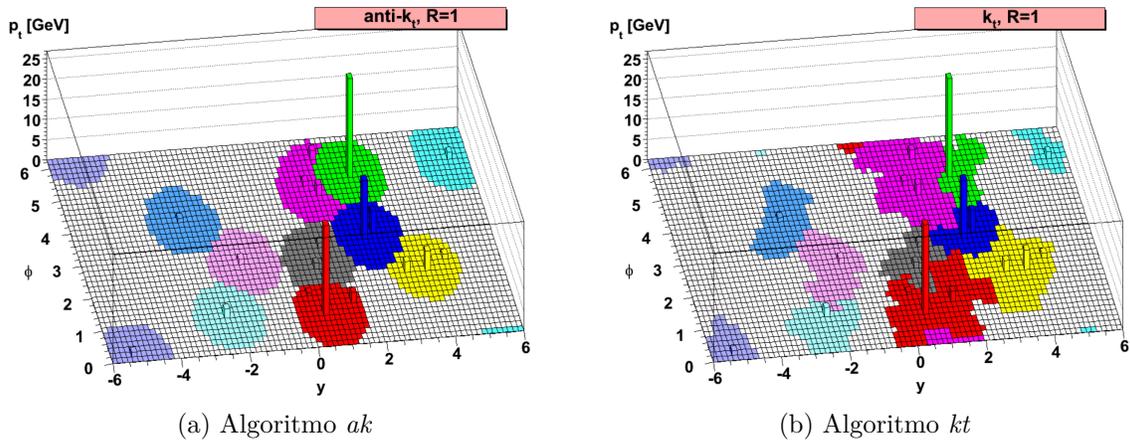


Figura A.3: Área de jets reconstruidos con algoritmos ak y kt , utilizando con el mismo conjunto de datos y el mismo radio [10]. Podemos notar una alta irregularidad en el área de los jets en el algoritmo kt en comparación con el algoritmo ak . Para el algoritmo kt la forma exacta de las áreas son determinadas por el conjunto específico de fantasmas utilizados y cambia cuando los fantasmas son modificados [10]. Los distintos colores son utilizados para representar distintos jets y sus respectivas áreas.

A.4. Ramas de muones y electrones dentro de los AODs

En términos del CMS llamamos muones (o anti-muones) a, por lo menos, 1 de 2 trayectorias (*tracks*) una detectada en el *Tracker* y otro en el sistema muónico [1]. Dada su naturaleza específica de detección, dentro de los AODs la rama que contiene la información relevante a todos los muones detectados es:

- `recoMuons_muons__RECO.obj`.

Cuyas hojas de interés para nosotros serán:

- `pt_` Momento transverso.
- `eta_` Coordenada angular η .
- `phi_` Coordenada angular ϕ .
- `vertex_.fCoordinates.fX` Componente x del vértice de creación.
- `vertex_.fCoordinates.fY` Componente y del vértice de creación.
- `vertex_.fCoordinates.fZ` Componente z del vértice de creación.

Para reconstruir a los electrones se utiliza un algoritmo llamado *Gaussian Sum Filter* (GSF), el cual modela la distribución de pérdida de energía de los electrones, a través del detector, como una suma ponderada de distribuciones gaussianas (*Gaussian Sum*) [73]. Por lo tanto, a la rama de los electrones reconstruidos, dentro de los AODs, lleva el nombre de `GsfElectrons` y la rama específica utilizada para los resultados de la sección C.1 es:

- `recoGsfElectrons_gsfElectrons__RECO.obj`.

la cual contiene las siguientes hojas de interés:

- `pt_` Momento transverso.
- `eta_` Coordenada angular η .
- `phi_` Coordendad angular ϕ .
- `vertex_.fCoordinates.fX` Componente x del vértice de creación.
- `vertex_.fCoordinates.fY` Componente y del vértice de creación.
- `vertex_.fCoordinates.fZ` Componente z del vértice de creación.

Muones y/o electrones dentro de un jet

Para considerar muones y/o electrones contenidos dentro de un jet debemos primero conocer el vértice de creación del jet. En el caso de Pythia los eventos son generados justo en el origen del sistema coordenado, por lo que los jets se encuentran centrados en el origen. Como vimos en la sección 2.1, para el caso de los jets reconstruidos utilizando partículas encontradas mediante el algoritmo PF, tenemos también

que todos los jets reconstruidos tienen el vértice de creación centrado en el origen del sistema coordinado. Por lo que en ambos casos consideramos que los jets se originan en las coordenadas $(0,0,0)$.

Ahora, no perdamos de vista que las ramas `recoMuons_muons__RECO.obj`. y `recoGsfElectrons_gsfElectrons__RECO.obj`. contienen la información de todos los muones y electrones detectados por CMS. Esto significa que no necesariamente todos estos muones y electrones fueron considerados para formar parte de algún jet. De esta forma, tenemos que los muones y electrones contenidos en los jets no debería tener vértices de creación *muy* distintos al de los jets $(0,0,0)$. En este trabajo se implementaron tres filtros para considerar sólo a los muones y electrones contenidos en jets. La metodología es la misma para ambos así que describiremos el proceso refiriéndonos a cualquiera de ellos como *leptón*.

1. Se considera la multiplicidad (m) del leptón dentro de cada jet.
2. Si la multiplicidad es $m > 0$. Entonces buscamos los m leptones más cercanos, en el espacio angular $\phi - \eta$, al jet que los contiene.
3. Finalmente, solamente consideraremos a los m leptones que cumplen con que, además, su vértice de creación se encuentre contenido en una esfera de radio 0.5 cm y centrada en el origen.

Apéndice B

B.1. Estructura de los directorios relevantes

Para describir la metodología desarrollada, la estructura de los directorios merece ser mencionada. Esta estructura se muestra esquemáticamente en la figura B.1. El sistema operativo utilizado para este trabajo es la distribución Ubuntu 18.04 de Linux.

Los directorios `fastjet332`, `Pythia8240` y `root61406` son los directorios donde se han instalado nuestros programas principales a utilizar. El directorio `T` es el directorio de trabajo, es decir, en donde se encuentran almacenados los *macros* de análisis, simulación y comparación. En el directorio `OpenData` encontramos los *macros* de análisis de datos y 4 subdirectorios, uno por cada tipo de AOD. Por otro lado, tenemos al directorio `Parallelism` en el cual se encuentran los *macros* responsables de comparar los resultados del análisis de datos con los de la simulación y los resultados son guardados en `Comparisons`. Por otra parte, encontramos al directorio `Pythia` en donde tenemos a los *macros* de simulación de colisiones y los resultados son guardados en `FastJet`. Finalmente, tenemos a `Pawahtun` que es un disco duro externo y dentro del él se encuentra el directorio `CMS_Run211A` el cual contiene los distintos tipos de AODs con los que se trabaja.

B.2. Carpeta compartida en la MV

Una carpeta compartida en la Máquina Virtual (MV) es un directorio que se encuentra en nuestro Sistema Operativo (SO) al cual podemos acceder mediante la MV.

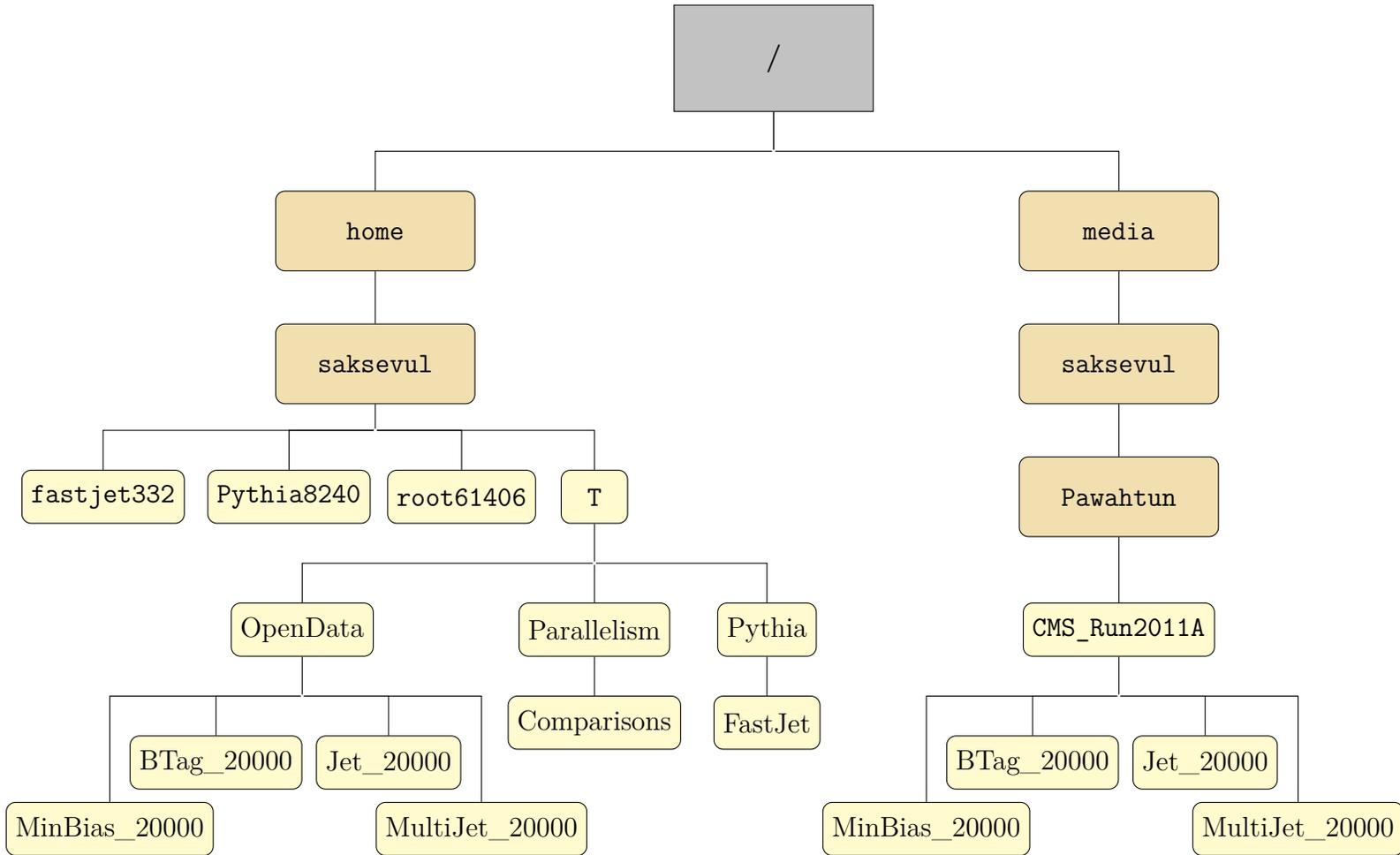


Figura B.1: Distribución de los directorios principales utilizados en el desarrollo de este trabajo.

Para configurar un directorio compartido mediante VirtualBox simplemente seleccionamos nuestra máquina virtual y pulsamos la opción **Configuración**. Después seleccionamos la opción **Carpetas compartidas** y finalmente escogemos **Agregar nueva carpeta compartida**. Con esto seleccionamos la carpeta que deseamos compartir con la MV, nos pedirá la ruta de la carpeta (su ubicación en nuestro SO), un nombre para mostrarlo en la MV y seleccionaremos la opción **Hacer permanente**.

En mi caso, la carpeta compartida es `/media/saksevul/Pawahtun` y su ruta corresponde al de un dispositivo de almacenamiento externo (Disco Duro) llamado “Pawahtun”. El nombre para mostrar en la MV es `Pawahtun`.

Después de estos sencillos pasos necesitamos reiniciar la MV, en caso de estarla utilizando. O, caso contrario, simplemente encenderla para poder tener acceso a la carpeta compartida. Esta carpeta compartida se encontrará en el directorio `/mnt/shared` de la MV.

De esta manera el directorio mostrado en el capítulo 2.1.1, dentro de la MV,

```
/mnt/shared/Pawahtun/CMS_Run2011A/MinimumBias_20000
```

hace referencia a una carpeta compartida en la cual se encuentran contenidos los directorios `CMS_Run2011A/MinimumBias_20000`.

B.3. Configuración del entorno de Root

Una vez instalado Root, para ejecutar Root, es necesario configurar antes el entorno. Esto se puede hacer muy fácilmente al ejecutar el siguiente comando en la terminal.

```
source (/path/to/install-or-build/dir)/bin/thisroot.sh
```

en mi caso este comando se reduce a:

```
source ~/root61406/bin/thisroot.sh
```

Para evitar tener que ejecutar este comando cada vez que se abra la terminal y se desee iniciar Root, se recomienda editar el archivo `.bashrc`. Dicho archivo se encuentra en el directorio `~` para editarlo nos servimos de algún editor de textos, por ejemplo *gedit*, y escribimos en la terminal:

```
gedit ~/.bashrc
```

Una vez que se abra el editor de texto escribiremos las siguientes líneas al final del archivo.

```
source ~/root61406/bin/thisroot.sh
```

Esto sirve para configurar automáticamente el entorno para ejecutar Root cada vez que se abra alguna terminal. De esta manera, podemos olvidarnos de este paso cada vez que queramos ejecutar Root. Y para ejecutar Root simplemente necesitamos el siguiente comando en la terminal:

```
root
```

B.3.1. Navegar por los AODs

Una vez que hemos descargado algún archivo AOD, lo siguiente que nos gustaría hacer es navegar por los archivos para familiarizarnos más con ellos. Para este fin nos valdremos del navegador de Root: *TBrowser*.

Esto es sencillo de hacer. Basta con abrir la terminal, configurar el entorno Root (ver apéndice B.3) e iniciar Root

```
root
```

Una vez iniciado Root, procedemos a acceder al navegador, esto se hace con la siguiente orden

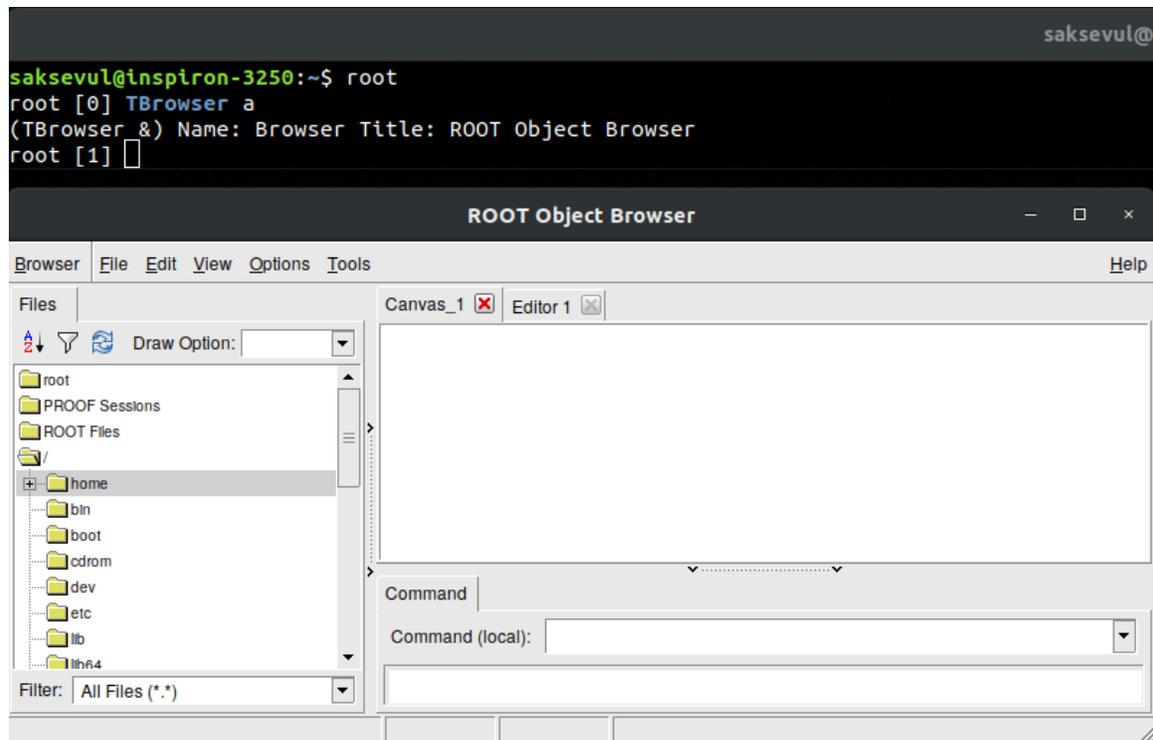


Figura B.2: Navegador TBrowser de Root.

TBrowser a

Esto nos abrirá una ventana con el navegador de Root. Tal como se muestra en la figura B.2. Como podemos ver, en la columna izquierda encontramos el directorio raíz (/) de nuestro sistema operativo y será a través de él que encontraremos la ubicación de nuestros AODs.

B.4. Metodología recursiva para AODs

En esta sección mostraremos, con ayuda del ejemplo `Histo_pT_jet.C` de la sección 2.1.4, como extraer información recursiva de distintos AODs para aumentar tanto la cantidad como la variedad de la información analizada.

B.4.1. Aumentando la cantidad de información analizada

Generaremos un *macro* que lea la información disponible a cerca del pT de ak5PF jets proveniente de distintos archivos AOD del mismo tipo y genere un archivo `.root` de salida para cada AOD. Posteriormente este *macro* fusionará todos los archivos AOD y obtendremos un único archivo final con la estadística contenida en todos los AODs utilizados. La visualización esquemática del procedimiento desarrollado se muestra en la figura B.3.

Dicho lo anterior, el código `Aumentar.bash` tendría la siguiente forma, donde estamos utilizando la distribución de directorios mostrada en el apéndice B.1.

```

1  # Definimos los parámetros de entrada. Esto es útil para hacer
   ↪ cambios rápido y como apoyo para tener un código más sucinto.
2  # Ruta al Macro Maestro (MM). En este caso el macro es
   ↪ Histo\_pT\_jet.C. El nombre es debido al acrónimo Path to the
   ↪ Master Macro (pttMM).
3  PttMM=/home/saksevul/T/OpenData/Histo\_pT\_jet.C
4  # Dirección de la Lista de archivos AOD que se utilizarán. Llamado
   ↪ así por Path to the AOD Files List (pttAFL).
5  PttAFL=/home/saksevul/CMS_Run2011A/MinBias_20000/AODFilesList
6  # Directorio de almacenamiento de los archivos de salida. Path to
   ↪ the Output AOD Files Directory (PttOAFD).
7  PttOAFD=/home/saksevul/T/OpenData/MinBias_20000
8  # Archivo AOD de inicio para el análisis. previous AOD File (pAF).
9  pAF=0001.root
10  # Definido para restaurar el MM a su estado final. final AOD File
   ↪ (pAF).
11  fAF=$pAF
12
13  # Realizamos el ciclo for sobre los archivos enlistados en AOD
   ↪ Files List (AFL). Es decir, aumentamos la estadística al
   ↪ considerar la información contenida en distintos AODs.
14  for AF in $(cat $PttAFL)
15  do
16  # Utilizamos el i-ésimo archivo AOD de entrada en el Master Macro
   ↪ (MM).

```

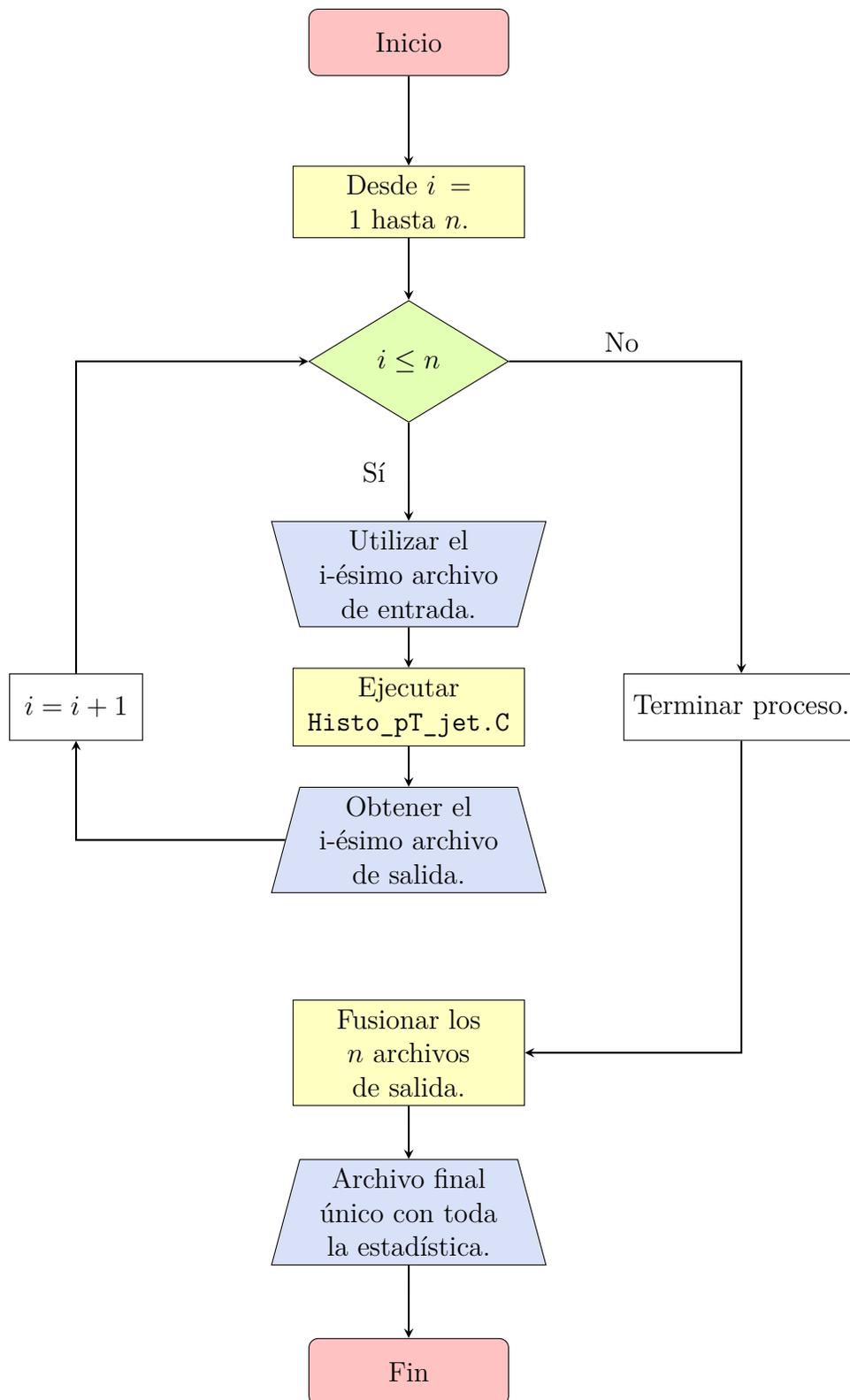


Figura B.3: *Macro* de extracción recursiva de datos. Este procedimiento es llevado a cabo por el *macro* `Aumentar.bash` para considerar información de distintos AODs.

```

17  sed -i "s/$pAF/$AF/g" $PttMM
18  # Ejecutamos el MM. (Sin mensajes de error).
19  root -l -q 2> /dev/null $PttMM
20  # Redefinimos pAF para el siguiente ciclo.
21  pAF=$AF
22  done # Fin del ciclo for para AF.
23
24  # Eliminamos los archivos viejos, pues serán remplazados.
25  rm $PttOAFD/$ak5PFJets_pt.root
26  # Generamos un archivo .root final único con toda la estadística.
27  hadd $PttOAFD/$ak5PFJets_pt.root $PttOAFD/$ak5PFJets_pt-*.root
28  # Eliminamos permanentemente los archivos individuales.
29  rm $PttOAFD/$ak5PFJets_pt-*.root
30
31  # Regresamos el MM a su estado original.
32  sed -i "s/$AF/$fAF/g" $PttMM
33
34  # Eliminamos todas las variables de bash utilizadas.
35  exec bash

```

Con esto tenemos el archivo `ak5PFJets_pT.root` conteniendo no sólo la información de un único archivo AOD, sino conteniendo la información de n AODs.

B.4.2. Considerando distintos algoritmos de reconstrucción de jets PF

Ahora que somos capaces de extraer información proveniente de más de un AOD para aumentar la cantidad de información utilizada para nuestro análisis, el siguiente paso es considerar los distintos tipos de PFJets contenidos en los AODs. Para esto tomaremos ahora como base el *macro* `Recursivo.bash` y lo modificaremos de tal manera que agreguemos un ciclo `for` que tome los distintos tipos de jets. La metodología la podemos considerar como en la figura B.4.

En términos de programación esta fase consiste en agregar un ciclo `for` que contenga al ciclo `for` de `Aumentar.bash`. En otras palabras, seguiremos los pasos de

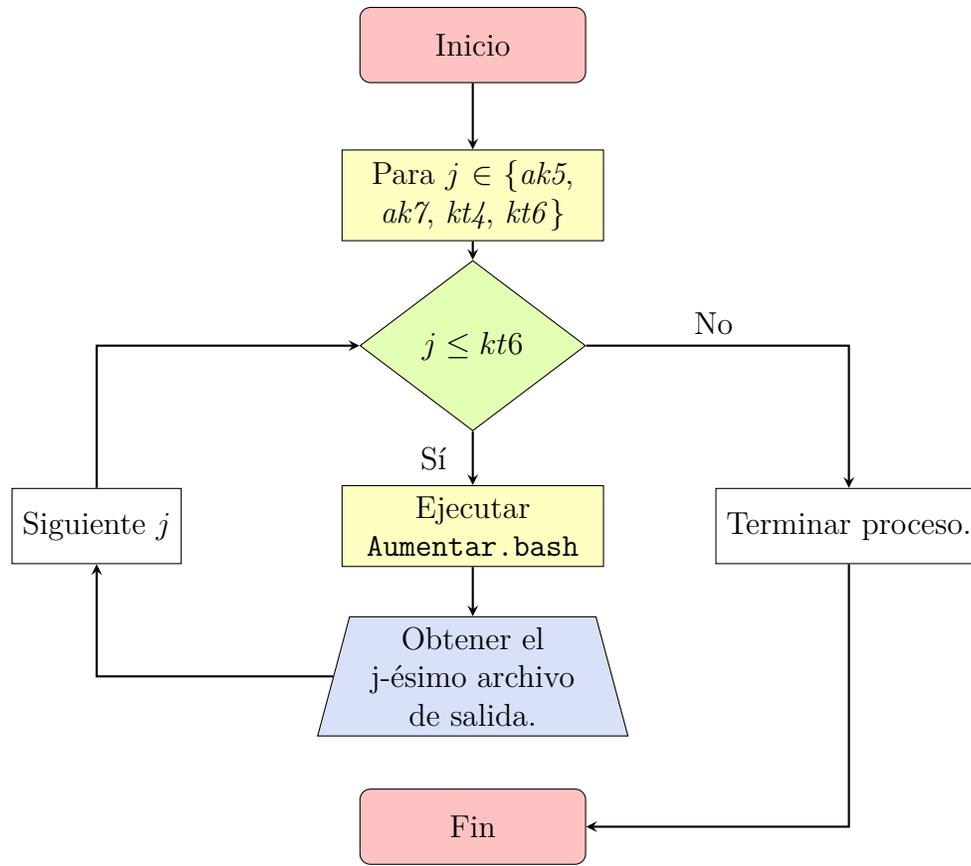


Figura B.4: *Macro* de extracción recursiva de datos. Este procedimiento es llevado a cabo por el *macro* `Recursivo.bash` para considerar información de distintos AODs y los 4 algoritmos de reconstrucción de jets PF disponibles en los AODs.

`Aumentar.bash` para cada uno de los tipos de jet PF. Con esto obtendremos no uno, sino 4 archivos finales (uno por cada algoritmo de reconstrucción de jet PF) que contengan la estadística de los n AODs utilizados en `Aumentar.bash`.

Dado que el código necesario para llevar a cabo este procedimiento es similar al expuesto en la subsección B.4.1. No generaremos un código propio para esta parte.

B.4.3. Utilizando distintos tipos de AODs

Como hemos mencionado anteriormente, es posible considerar distintos tipos de AODs para un análisis más amplio. De este modo procederemos a mencionar la forma de incluir esto en nuestra extracción recursiva de información.

Tal como se hizo en la subsección B.4.2, el *quid* asunto es encapsular en un ciclo

for adicional el procedimiento realizado por `Recursivo.bash`. De tal manera que obtengamos 4 archivos finales para cada tipo de AOD. En esta tesis considero 4 tipos de AOD, a saber *BTag*, *Jet*, *MinBias* y *MultiJet*. Por lo tanto, al final de este *macro*, llamado `bashMacro.bash`, obtendremos 16 archivos finales.

Para conocer el *macro* específico utilizado en este trabajo, ver el siguiente archivo en mi página de GitHub:

```
https://github.com/Saksevul/T/blob/master/OpenData/bashMacro.sh
```

B.5. Conectar Pythia con otros programas

Por defecto los resultados de las simulaciones en Pythia son mostrados en la terminal. Esto acarrea consigo muchas limitaciones al querer realizar análisis de la información generada. Otra limitación de Pythia es que, pese a poseer herramientas para la reconstrucción de jets¹, no es tan eficiente en comparación a utilizar paqueterías dedicadas para este fin.

Para sortear estos inconvenientes es necesario conectar a Pythia 8240 con programas dedicados a cada tarea. A saber, conectarlo a un programa especializado en la reconstrucción de jets y a otro especializado en el análisis de datos. En este caso utilicé FastJet 3.3.2 para reconstruir los jets y Root 6.14/06 como formato de salida para su posterior análisis.

Entonces, en el directorio de instalación de Pythia corremos los siguientes comandos en la terminal:

```
./configure --with-fastjet3=fastjet3-installation-directory  
--with-root=root-installation-directory
```

¹Pythia viene instalado con 3 buscadores de jets [60]: `ClusterJet` para eventos $e^+ e^-$ y `SlowJet` y `CellJet` para colisiones hadrónicas.

```
make
```

Finalmente, debemos editar el archivo `Makefile` (que se encuentra en mismo directorio que `Simulacion.cc`) y agregar una sección especial para poder compilar nuestro *macro* `Simulacion.cc`, el cual utiliza Pythia con FastJet y Root. Esto es necesario pues no existe ninguna instrucción, por default, para entrelazar a Pythia con FastJet y Root.

```
# FASTJET3 and ROOT.
Simulacion: $$@.cc $(PREFIX_LIB)/libPythia8.a
ifeq ($(FASTJET3_USE)$(ROOT_USE),true)
$(CXX) $< -o $@ -I$(FASTJET3_INCLUDE) -I$(ROOT_INCLUDE)
↪ $(CXX_COMMON)\
  `$(ROOTBIN)root-config --cflags`\
  -Wl,-rpath,$(ROOT_LIB) `$(ROOT_BIN)root-config --glibs`\
  -L$(FASTJET3_LIB) -Wl,-rpath,$(FASTJET3_LIB) -lfastjet\
  $(GZIP_INC) $(GZIP_FLAGS)
else
@echo "Error: $@ requires FASTJET3 and ROOT"
endif
```

Error encontrado

Al terminar los pasos siguientes e intentar compilar `Simulacion.cc` obtenía un mensaje de error en la salida de la terminal. Leyendo los mensajes de error y el archivo `Makefile`, logré encontrar la causa. El problema se debe a que en el archivo `Makefile.inc` (el cual se localiza donde `Simulacion.cc`) la información referente a las dependencias necesarias de Root está incompleta. Para resolverlo necesitamos agregar las siguientes líneas resaltadas en amarillo:

```
# ROOT configuration.
ROOT_USE=true
ROOT_BIN=/home/saksevul/root61406/bin/
```

```

ROOT_INCLUDE=/home/saksevul/root61406/include
ROOT_LIBS=-L/home/saksevul/root61406/lib -lGui -lCore -lImt -lRIO
↪ -lNet -lHist
-lGraf -lGraf3d -lGpad -lROOTVecOps -lTree -lTreePlayer -lRint
↪ -lPostscript
-lMatrix -lPhysics -lMathCore -lThread -lMultiProc -lROOTDataFrame
↪ -pthread
-lm -ldl -rdynamic
CXX_ROOT=-O2 -W -Wall -Wshadow -fPIC -pthread -std=c++11 -m64
↪ -I/home/saksevul/
root61406/include

```

Una vez hecho esto, el *macro* `Simulacion.cc` debería correr sin problema alguno. Finalmente, es necesario resaltar que este error en específico surgió al utilizar las versiones FastJet 3.3.2, Pythia 8240 y Root 6.14/06.

B.6. Información de usuario en FastJet

FastJet es una paquetería que reconstruye jets a partir de información de partículas. En este trabajo, dichas partículas y su respectiva información es generada mediante un programa de simulación de física de partículas: Pythia. Al utilizar FastJet para agrupar partículas generadas en procesos simulados en Pythia y formar jets, obtenemos la salida final de la información referente a las partícula que conforma al *i*-ésimo jet en `jets[i].constituyentes` que es una especie de vector de vectores² y cada una de sus entradas almacena información básica de cada partícula que constituye al jet. Es decir, si el `Evento[n]`, de Pythia, es utilizado para generar un jet entonces se guardará la información necesaria de `Evento[n]` en `jets[i].constituyentes[j]`.³

Ahora, el problema es que `jets[i].constituyentes[j]` ya no poseen toda la información que poseía `Evento[n]`. `jets[i].constituyentes[j]` almacena únicamente información básica: la energía, las componentes de su momento, masa del jet y

²En realidad es un vector de C++ cuya clase es *PseudoJet*. Dentro del código de programación se denota como `vector<PseudoJet>`.

³Para mayor información consultar [74].

combinaciones o transformaciones de estas (Capítulo 3.1 [49]). Sin embargo, para realizar todos los análisis de este trabajo es necesario conocer el `id()`⁴ de las partículas que forman al jet.

Para sortear este percance es posible definir información adicional que sea conservada para los constituyentes de cada jet. Esto se hará definiendo *Información del usuario* en la clase `PseudoJet` (apéndice B de [49]). Esto se hará siguiendo la idea planteada en [75]. El código utilizado se llama `MyInfo.h` y es el siguiente y se encuentra disponible en mi página personal de GitHub a través del siguiente link:

<https://github.com/Saksevul/T/blob/master/MyInfo.h>

```

1  using namespace std;
2  using namespace fastjet;
3  // Para asociar información adicional a PseudoJet debemos crear una
   ↪ clase utilizando fastjet::UserInfoBase, el cual contendrá dicha
   ↪ información.
4  class MyInfo : public PseudoJet::UserInfoBase {
5      public:
6          // Definimos la información adicional y el tipo de esta.
7          MyInfo( const int & Index ) :
8              // Índice de Event[índice].
9              _Index(Index){}
10         // Definir la manera de acceder al índice.
11         int Index() const { return _Index;}
12     protected:
13         // El valor asociado al índice.
14         int _Index;
15 }

```

La única razón por la que no se integró este código en el código de simulación principal (`Simulacion.cc`) es para hacer más entendible el código principal. La ubicación de `MyInfo.h` se encuentra en `/home/saksevul/fastjet332/include/fastjet` (ver figura B.1).

⁴Según la convención de *Particle Data Group* [69].

Esta información extra dentro de los constituyentes de los jets se define, dentro de `Simulacion.cc`, de la siguiente manera [76]:

```
// Esto es para poder conocer índice del evento de cada  
↪ constituyentes del jet.  
particleTemp.set_user_info( new MyInfo( i ) );
```

Pues `particleTemp` son las partículas candidatas a formar parte de algún jet.

Por último, podemos recuperar la información definida de la siguiente manera [76]:

```
sortedJets[i].constituents()[j].user_info<MyInfo>().Index()
```

Donde, para algún evento, estamos pidiendo el índice de evento (el valor de n en `Event[n]`) del j -ésimo constituyente (partícula) del i -ésimo jet.

Apéndice C

C.1. Prueba de límites inferiores de energía en la simulación

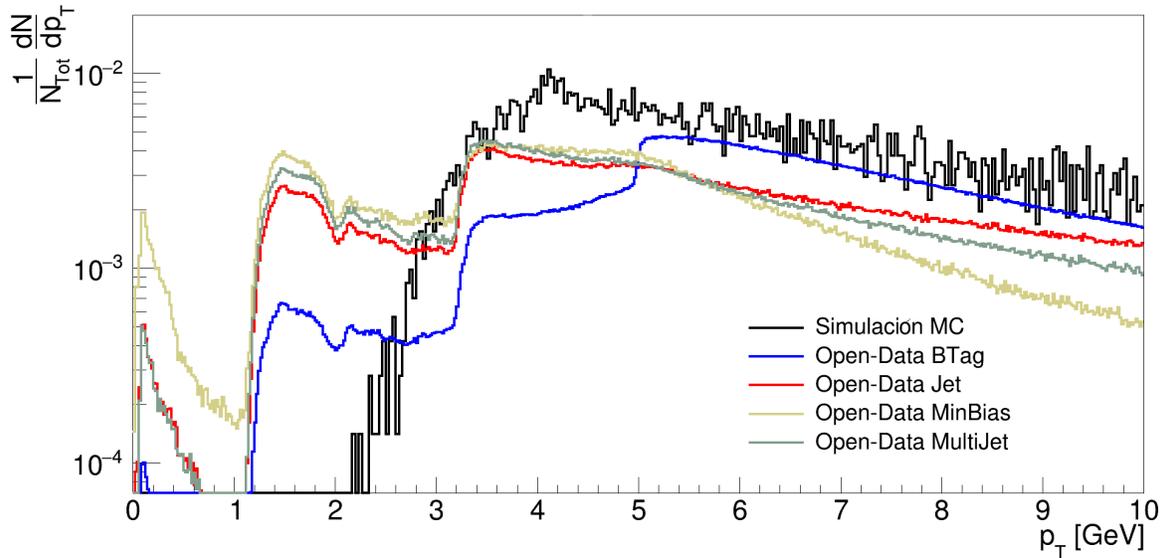


Figura C.1: Distribución de momento transversal, p_T , de muones contenidos en jets reconstruidos con el algoritmo ak5.

Una manera de obtener una prueba indirecta de la validez de los límites de energía inferiores impuestos en las simulaciones con Pythia es mediante el momento transversal de los muones y electrones contenidos dentro de los jets. Nuestra meta es obtener una distribución similar a la que tenemos en los datos abiertos para los muones y electrones contenidos en los jets (ver apéndice A.4). Utilizaremos estas partículas pues dentro de los AODs tenemos ramas dedicadas específicamente a cada una de ellas (ver apéndice A.4) .

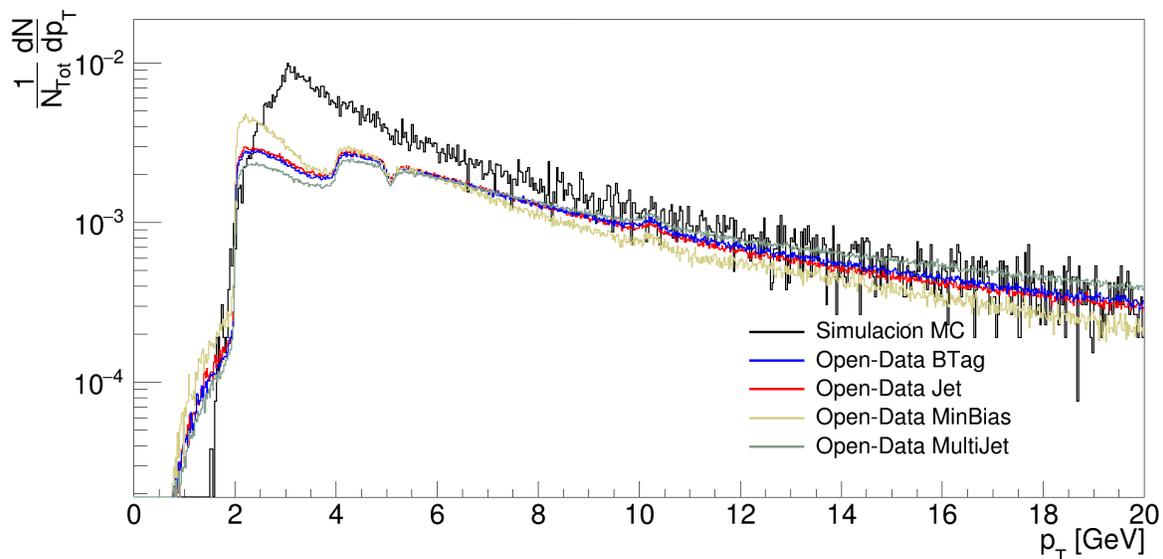


Figura C.2: Distribución de momento transverso, p_T , de electrones contenidos en jets reconstruidos con el algoritmo ak5.

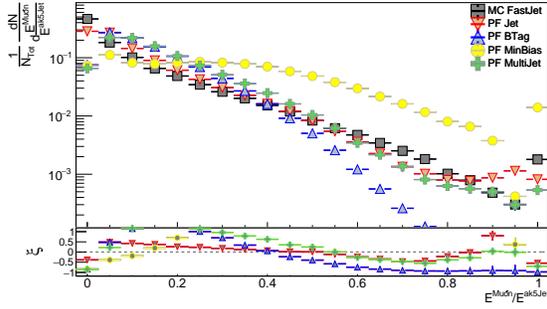
En la figura C.1 se muestra un histograma del momento transverso de los muones contenidos en los jets ak5 y en la figura C.2 se muestra el momento transverso de los electrones contenidos en los jets ak5.

Si bien los resultados no son exactamente iguales (debido en parte a que CMS tiene muchas otras restricciones, desde materiales hasta electrónicas), considero que es lo suficientemente aceptable para estimar los mínimos valores de momento transverso con que se identifican a las partículas.

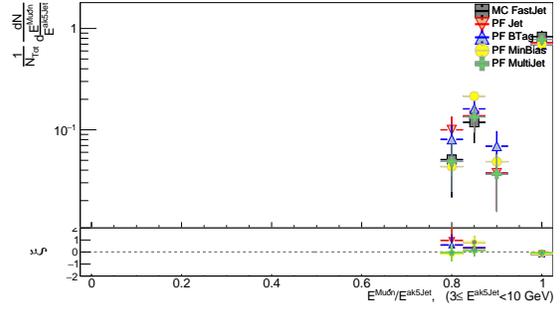
C.2. Desglose completo de energías de la FF

A continuación se muestran las gráficas de la FF para los distintos tipos de partículas en todos los rangos de energía utilizados para los jets. Cada subsección muestra los resultados para jets reconstruidos con un algoritmo específico.

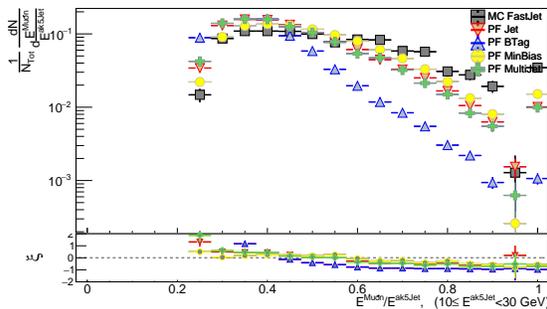
C.2.1. ak5Jets



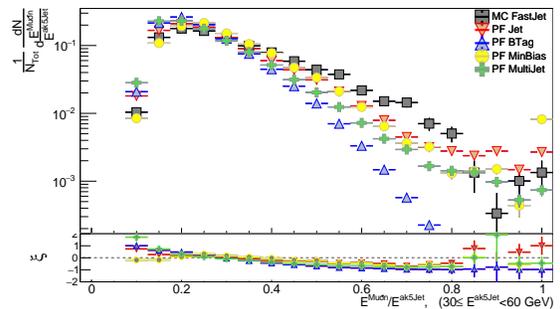
(I) Muones



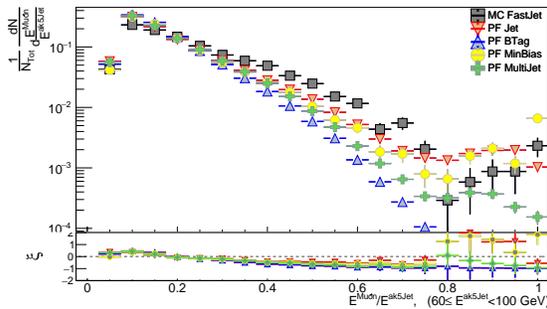
(II) Muones ($3 \leq E^{ak5Jet} < 10$ GeV)



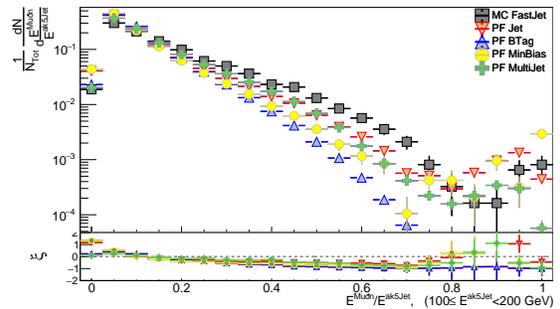
(III) Muones ($10 \leq E^{ak5Jet} < 30$ GeV)



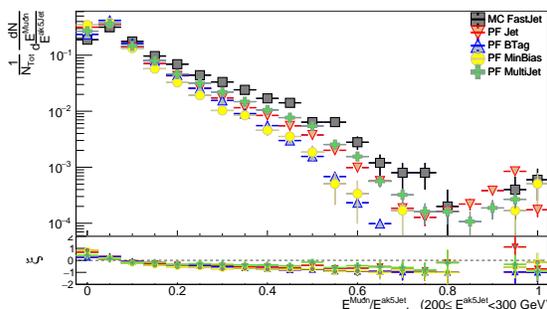
(IV) Muones ($30 \leq E^{ak5Jet} < 60$ GeV)



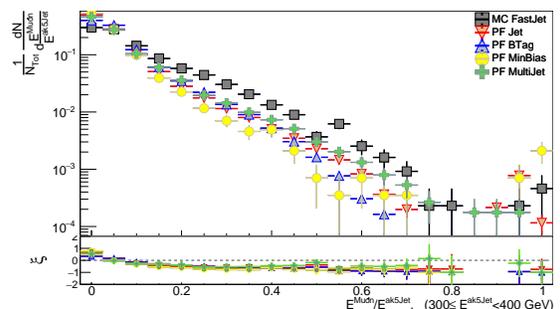
(V) Muones ($60 \leq E^{ak5Jet} < 100$ GeV)



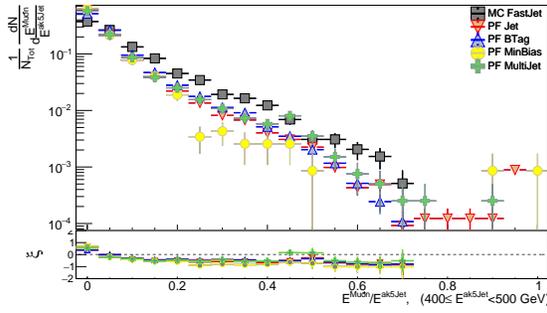
(VI) Muones ($100 \leq E^{ak5Jet} < 200$ GeV)



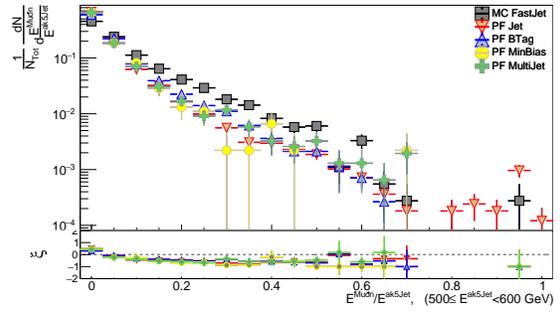
(VII) Muones ($200 \leq E^{ak5Jet} < 300$ GeV)



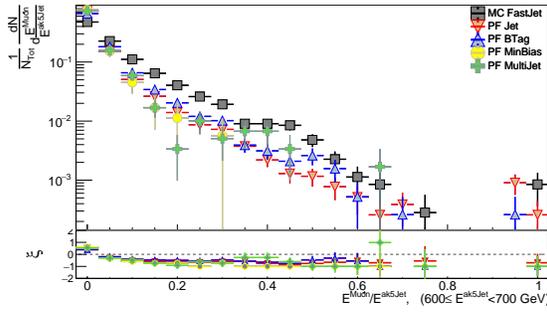
(VIII) Muones ($300 \leq E^{ak5Jet} < 400$ GeV)



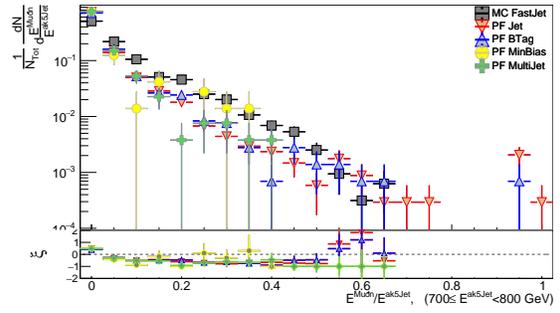
(IX) Muones ($400 \leq E^{ak5Jet} < 500$ GeV)



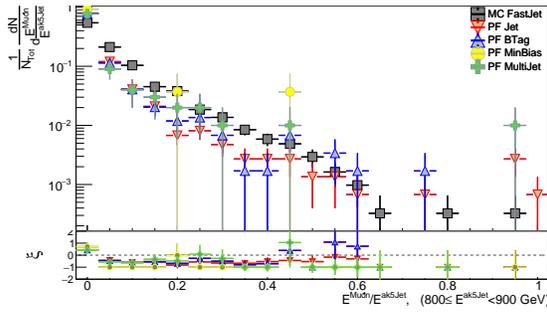
(X) Muones ($500 \leq E^{ak5Jet} < 600$ GeV)



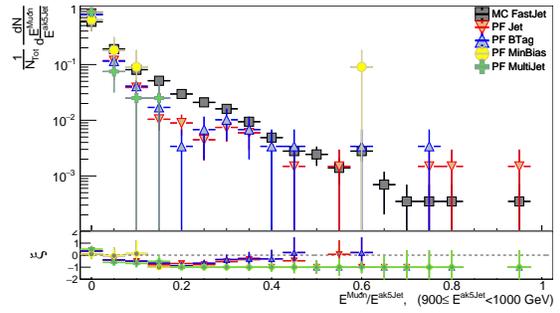
(XI) Muones ($600 \leq E^{ak5Jet} < 700$ GeV)



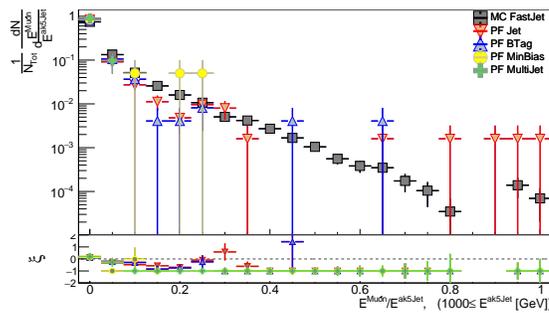
(XII) Muones ($700 \leq E^{ak5Jet} < 800$ GeV)



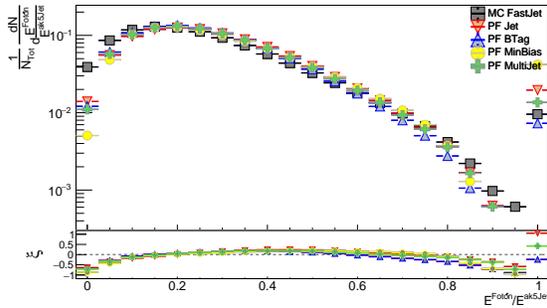
(XIII) Muones ($800 \leq E^{ak5Jet} < 900$ GeV)



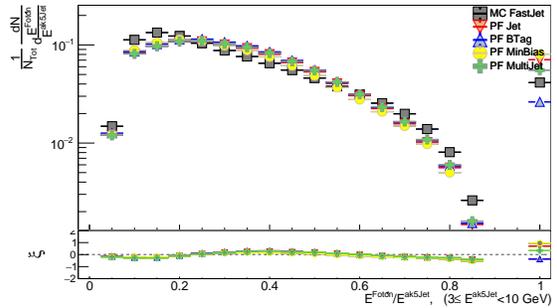
(XIV) Muones ($900 \leq E^{ak5Jet} < 1000$ GeV)



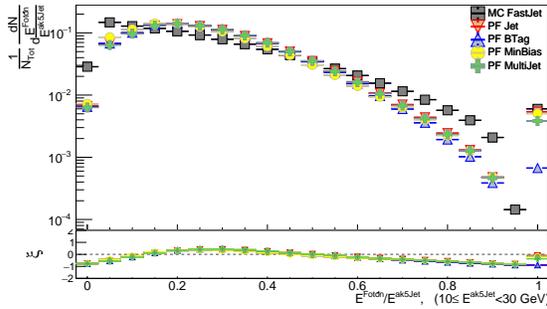
(XV) Muones ($1000 \leq E^{ak5Jet}$ [GeV])



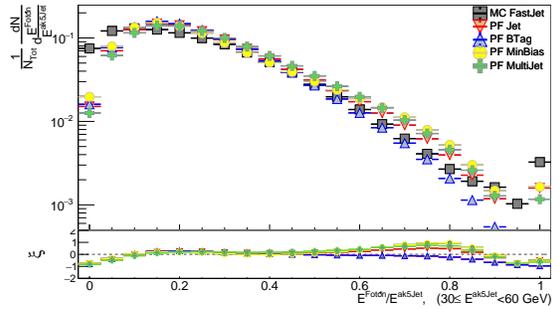
(XVI) Fotonos



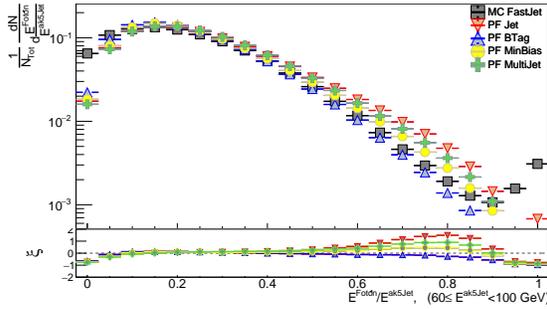
(XVII) Fotonos ($3 \leq E^{ak5Jet} < 10$ GeV)



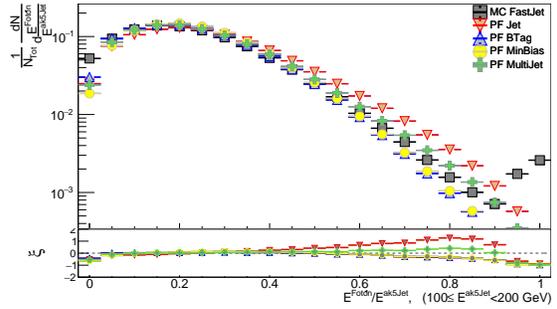
(XVIII) Fotonos ($10 \leq E^{ak5Jet} < 30$ GeV)



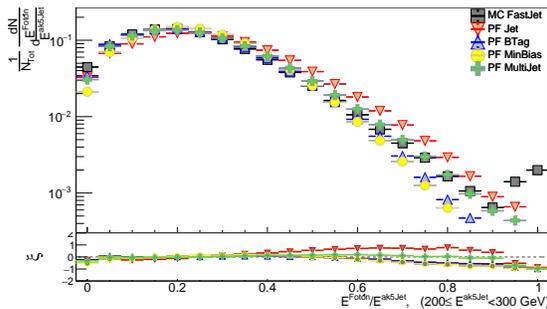
(XIX) Fotonos ($30 \leq E^{ak5Jet} < 60$ GeV)



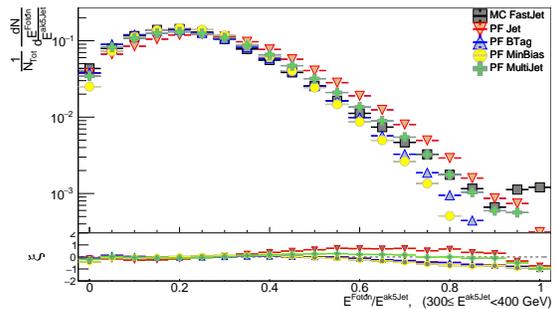
(XX) Fotonos ($60 \leq E^{ak5Jet} < 100$ GeV)



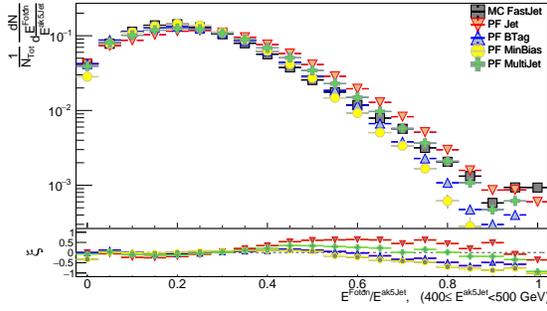
(XXI) Fotonos ($100 \leq E^{ak5Jet} < 200$ GeV)



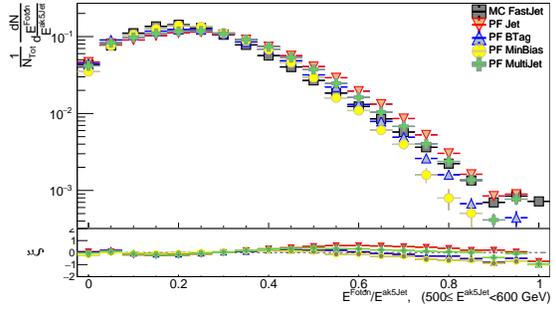
(XXII) Fotonos ($200 \leq E^{ak5Jet} < 300$ GeV)



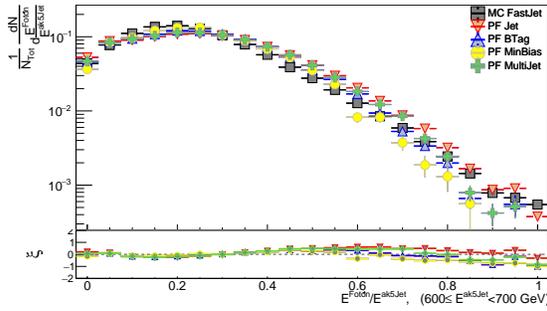
(XXIII) Fotonos ($300 \leq E^{ak5Jet} < 400$ GeV)



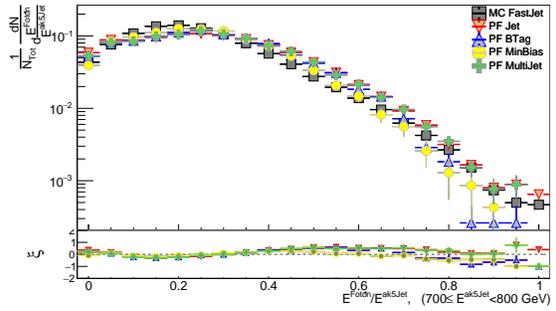
(XXIV) Fotonos ($400 \leq E^{ak5Jet} < 500$ GeV)



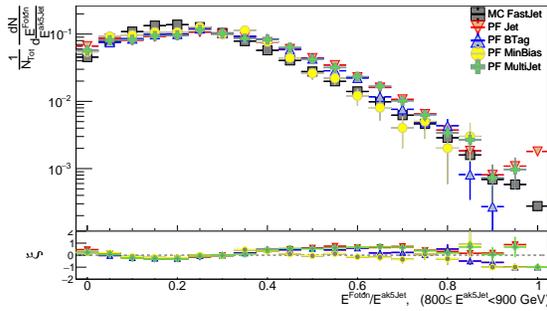
(XXV) Fotonos ($500 \leq E^{ak5Jet} < 600$ GeV)



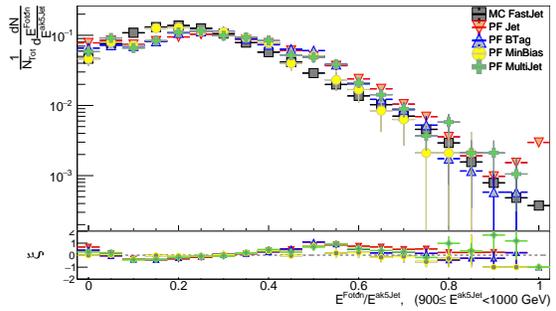
(XXVI) Fotonos ($600 \leq E^{ak5Jet} < 700$ GeV)



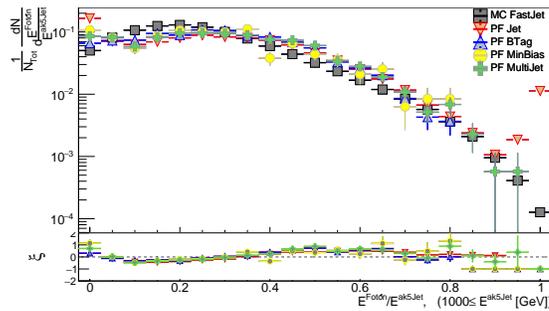
(XXVII) Fotonos ($700 \leq E^{ak5Jet} < 800$ GeV)



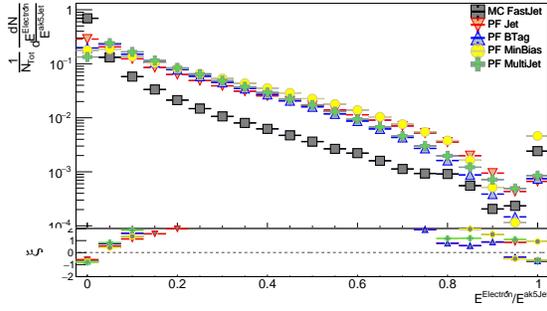
(XXVIII) Fotonos ($800 \leq E^{ak5Jet} < 900$ GeV)



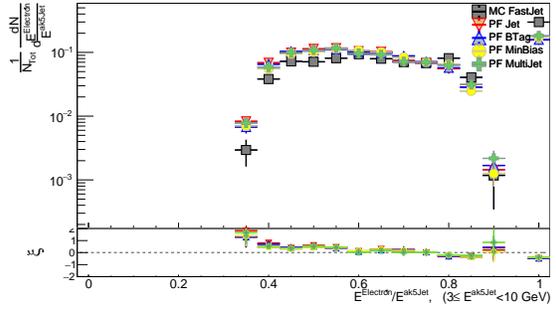
(XXIX) Fotonos ($900 \leq E^{ak5Jet} < 1000$ GeV)



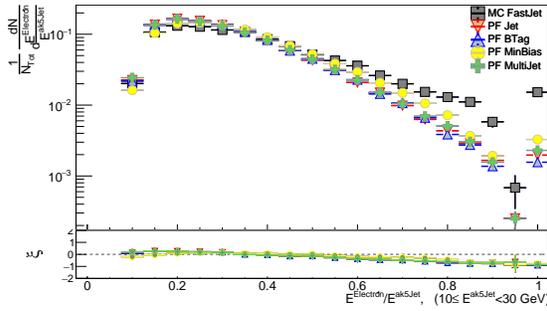
(XXX) Fotonos ($1000 \leq E^{ak5Jet}$ [GeV])



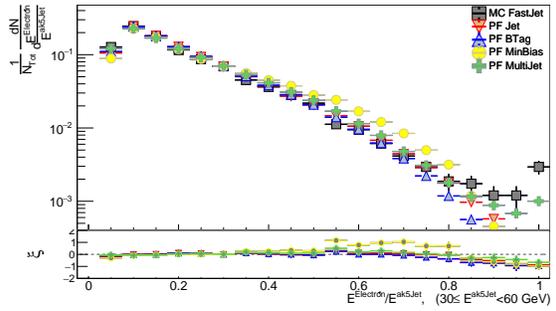
(XXXI) Electrones



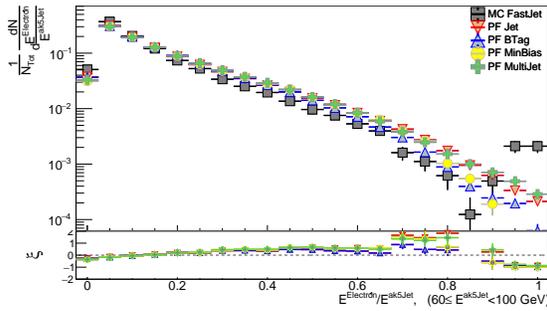
(XXXII) Electrones ($3 \leq E^{ak5Jet} < 10$ GeV)



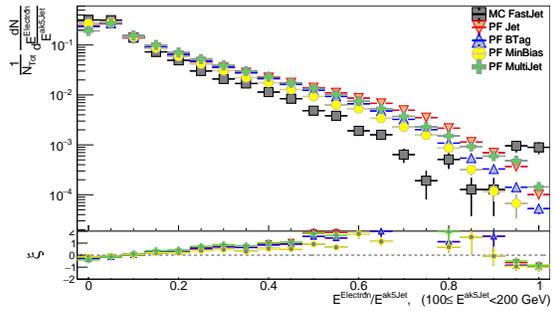
(XXXIII) Electrones ($10 \leq E^{ak5Jet} < 30$ GeV)



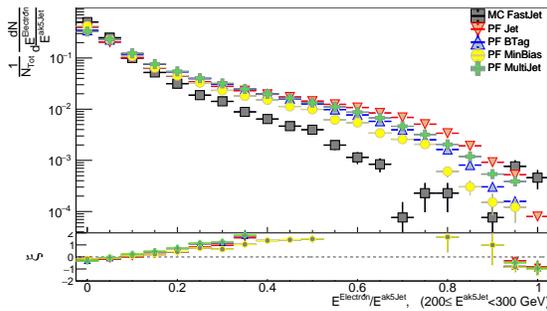
(XXXIV) Electrones ($30 \leq E^{ak5Jet} < 60$ GeV)



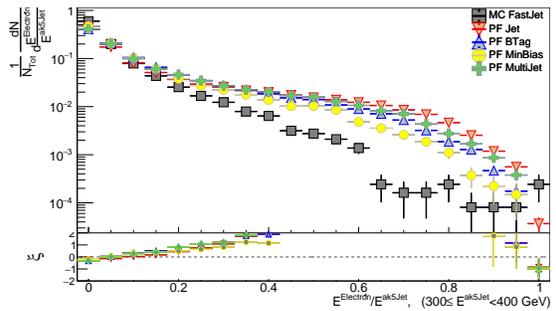
(XXXV) Electrones ($60 \leq E^{ak5Jet} < 100$ GeV)



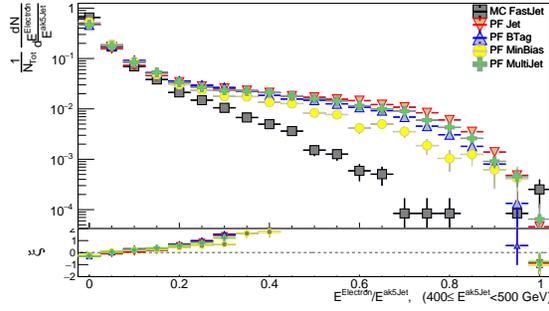
(XXXVI) Electrones ($100 \leq E^{ak5Jet} < 200$ GeV)



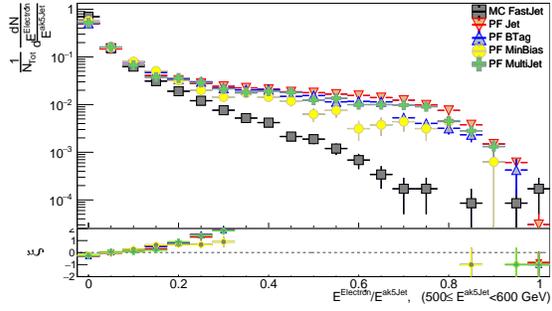
(XXXVII) Electrones ($200 \leq E^{ak5Jet} < 300$ GeV)



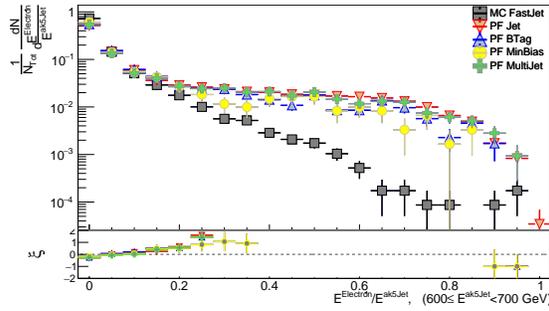
(XXXVIII) Electrones ($300 \leq E^{ak5Jet} < 400$ GeV)



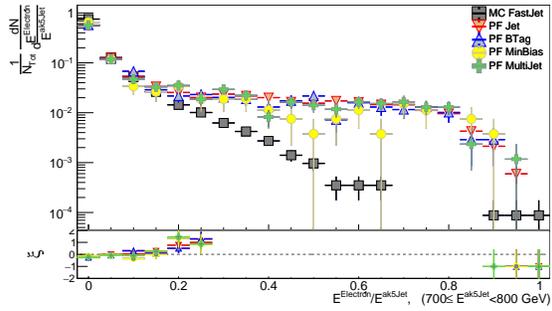
(XXXIX) Electrones ($400 \leq E^{ak5Jet} < 500$ GeV)



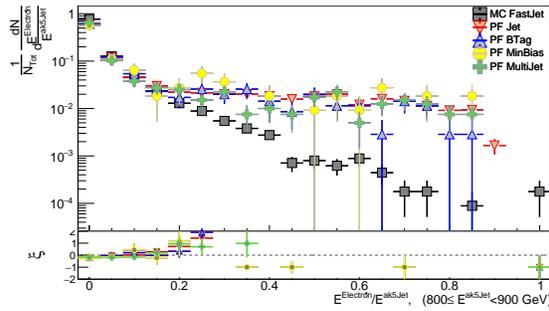
(XL) Electrones ($500 \leq E^{ak5Jet} < 600$ GeV)



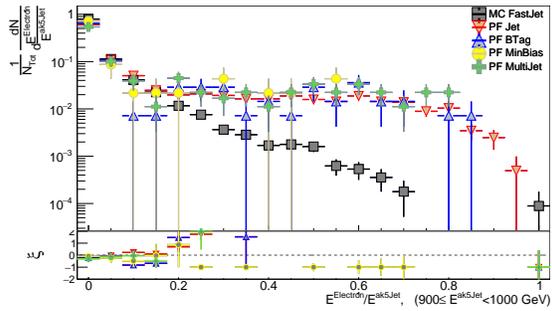
(XLI) Electrones ($600 \leq E^{ak5Jet} < 700$ GeV)



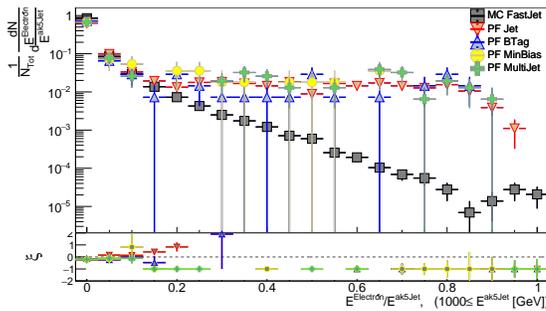
(XLII) Electrones ($700 \leq E^{ak5Jet} < 800$ GeV)



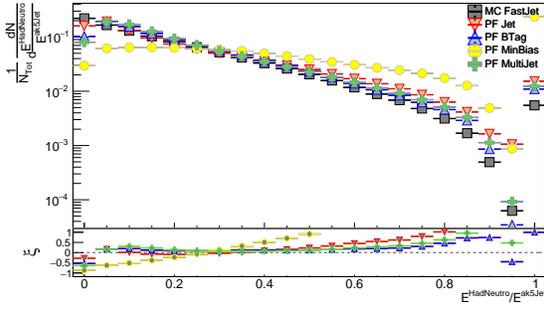
(XLIII) Electrones ($800 \leq E^{ak5Jet} < 900$ GeV)



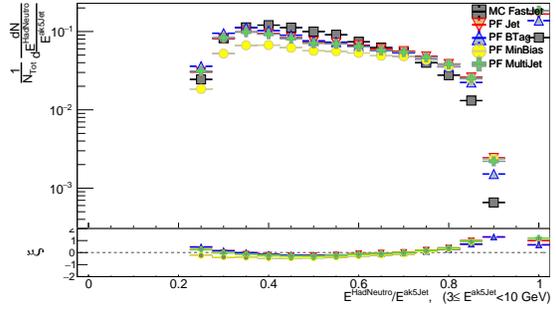
(XLIV) Electrones ($900 \leq E^{ak5Jet} < 1000$ GeV)



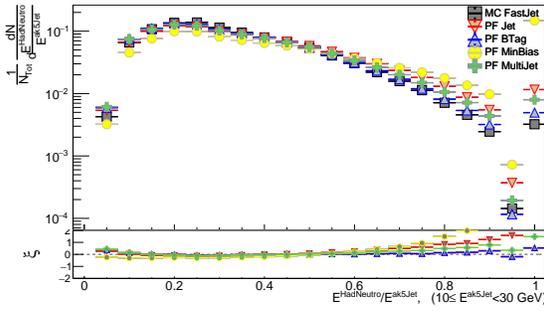
(XLV) Electrones ($1000 \leq E^{ak5Jet}$ [GeV])



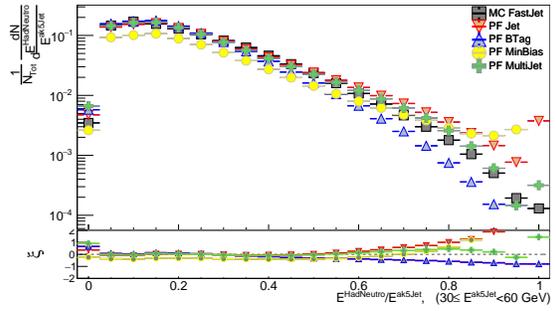
(XLVI) Hadrones neutros



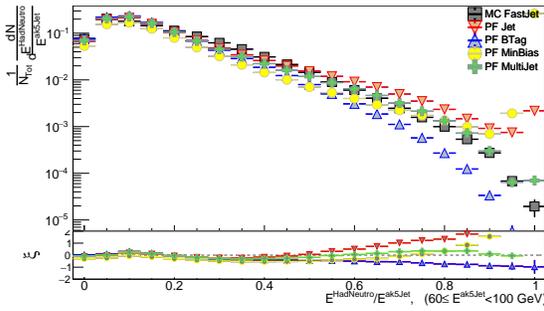
(XLVII) Hadrones neutros ($3 \leq E^{ak5Jet} < 10$ GeV)



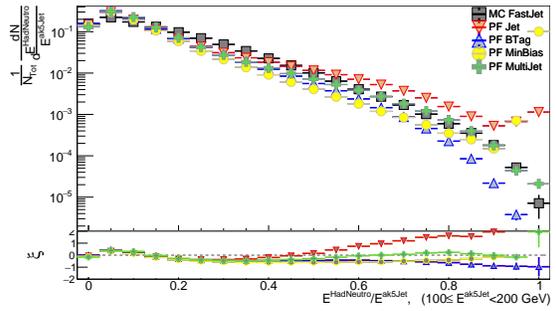
(XLVIII) Hadrones neutros ($10 \leq E^{ak5Jet} < 30$ GeV)



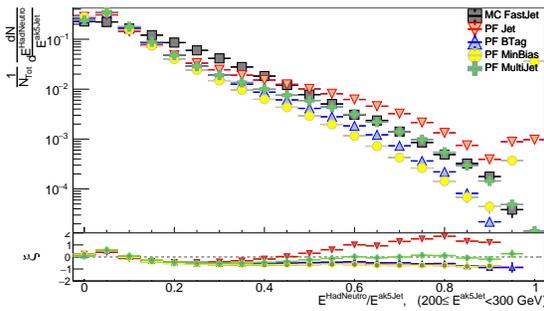
(XLIX) Hadrones neutros ($30 \leq E^{ak5Jet} < 60$ GeV)



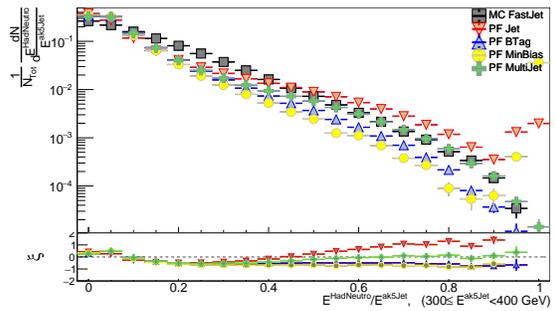
(L) Hadrones neutros ($60 \leq E^{ak5Jet} < 100$ GeV)



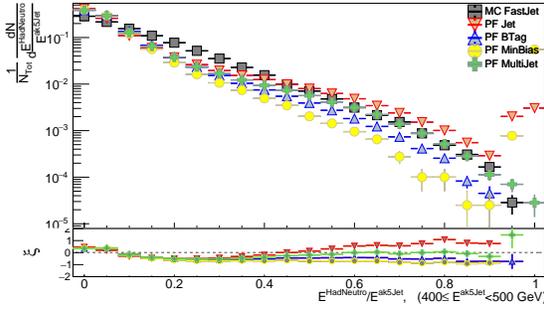
(LI) Hadrones neutros ($100 \leq E^{ak5Jet} < 200$ GeV)



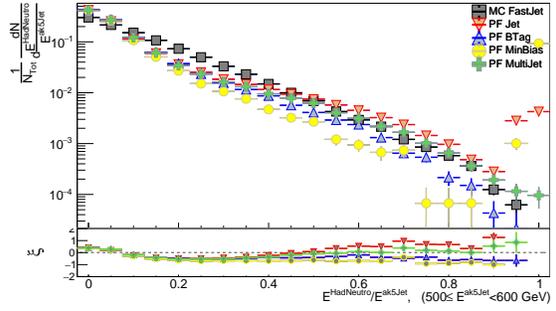
(LII) Hadrones neutros ($200 \leq E^{ak5Jet} < 300$ GeV)



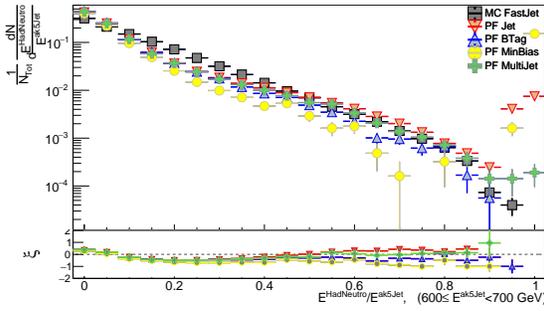
(LIII) Hadrones neutros ($300 \leq E^{ak5Jet} < 400$ GeV)



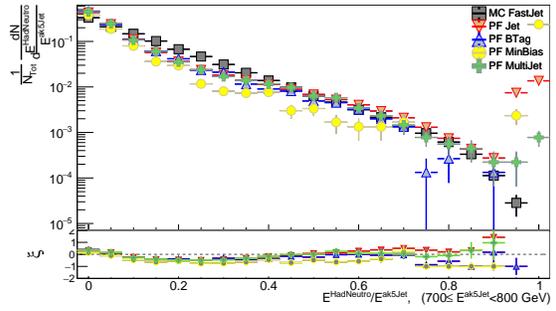
(LIV) Hadrones neutros ($400 \leq E^{ak5Jet} < 500$ GeV)



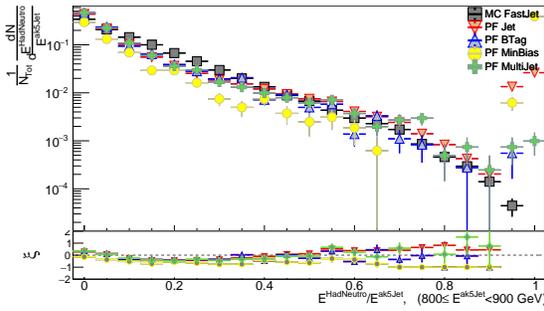
(LV) Hadrones neutros ($500 \leq E^{ak5Jet} < 600$ GeV)



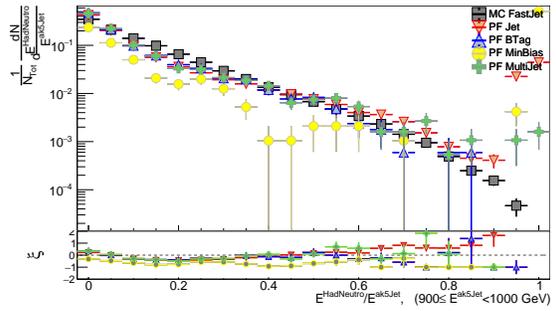
(LVI) Hadrones neutros ($600 \leq E^{ak5Jet} < 700$ GeV)



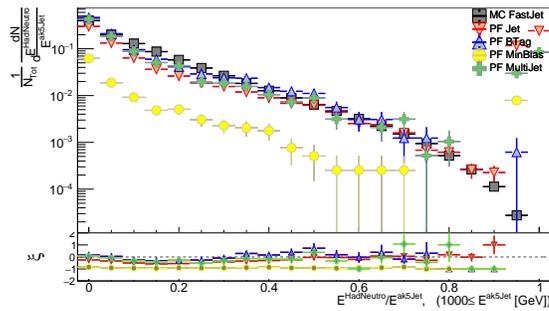
(LVII) Hadrones neutros ($700 \leq E^{ak5Jet} < 800$ GeV)



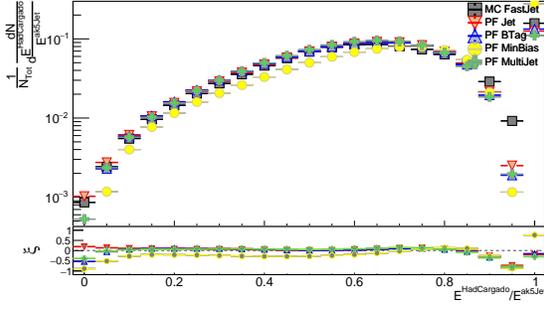
(LVIII) Hadrones neutros ($800 \leq E^{ak5Jet} < 900$ GeV)



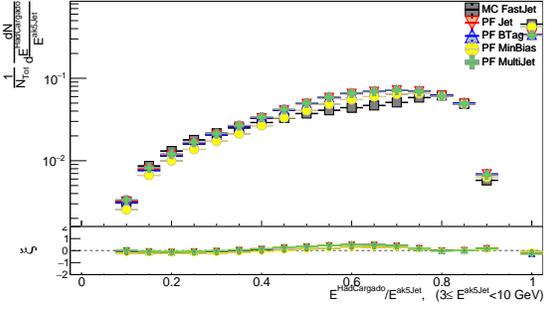
(LIX) Hadrones neutros ($900 \leq E^{ak5Jet} < 1000$ GeV)



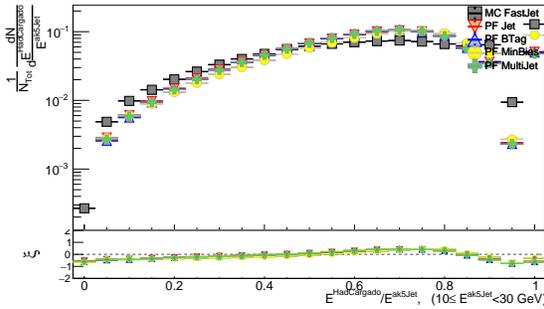
(LX) Hadrones neutros ($1000 \leq E^{ak5Jet}$ [GeV])



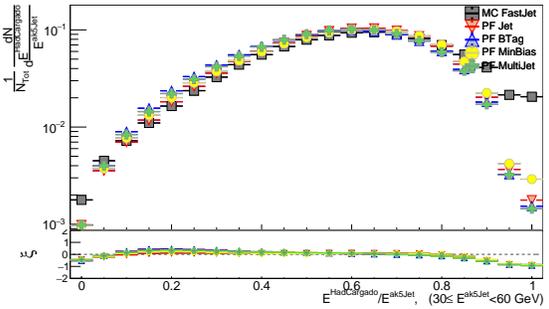
(LXI) Hadrones cargados



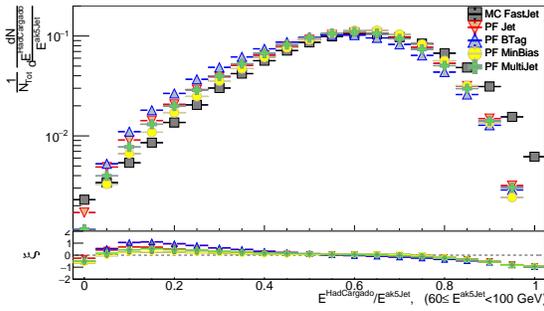
(LXII) Hadrones cargados ($3 \leq E^{ak5Jet} < 10$ GeV)



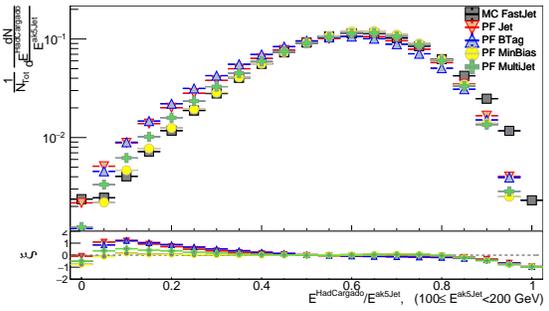
(LXIII) Hadrones cargados ($10 \leq E^{ak5Jet} < 30$ GeV)



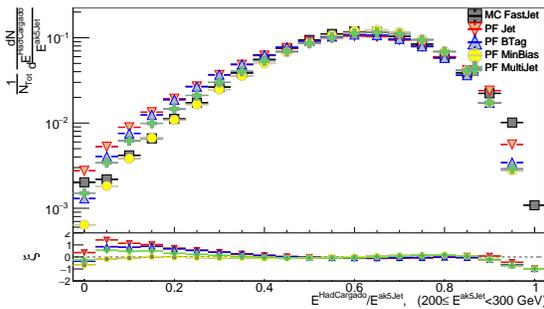
(LXIV) Hadrones cargados ($30 \leq E^{ak5Jet} < 60$ GeV)



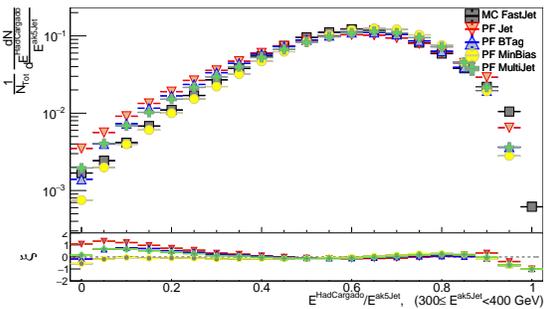
(LXV) Hadrones cargados ($60 \leq E^{ak5Jet} < 100$ GeV)



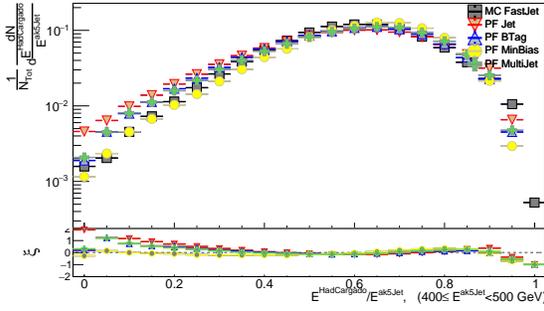
(LXVI) Hadrones cargados ($100 \leq E^{ak5Jet} < 200$ GeV)



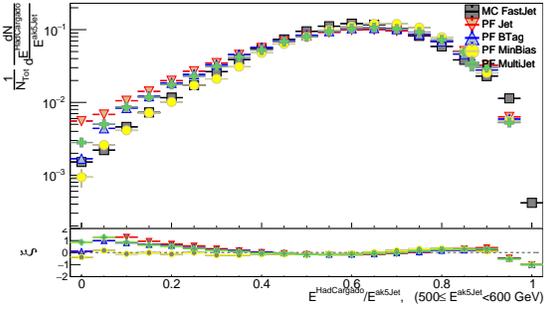
(LXVII) Hadrones cargados ($200 \leq E^{ak5Jet} < 300$ GeV)



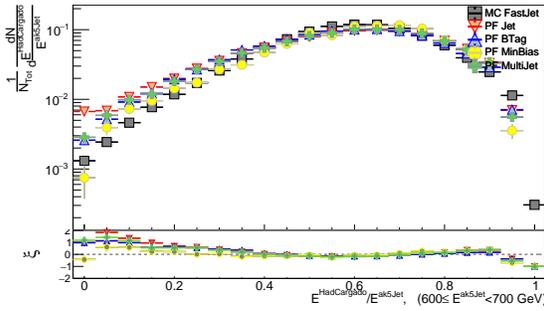
(LXVIII) Hadrones cargados ($300 \leq E^{ak5Jet} < 400$ GeV)



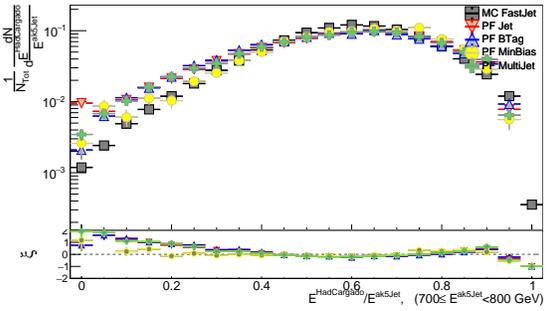
(LXIX) Hadrones cargados ($400 \leq E^{ak5Jet} < 500$ GeV)



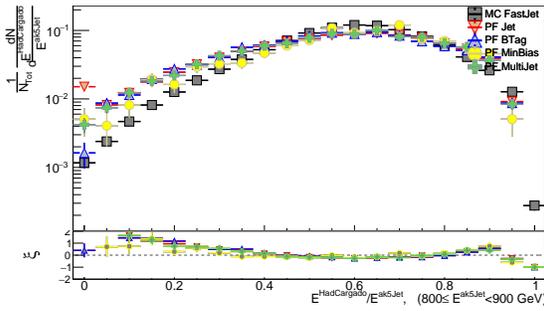
(LXX) Hadrones cargados ($500 \leq E^{ak5Jet} < 600$ GeV)



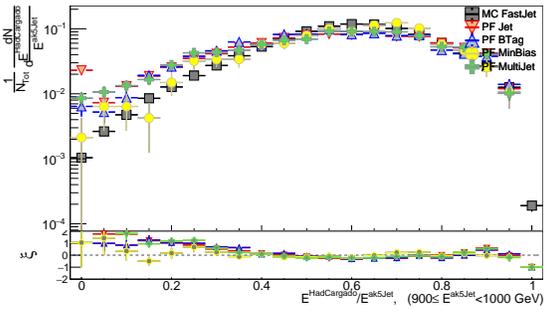
(LXXI) Hadrones cargados ($600 \leq E^{ak5Jet} < 700$ GeV)



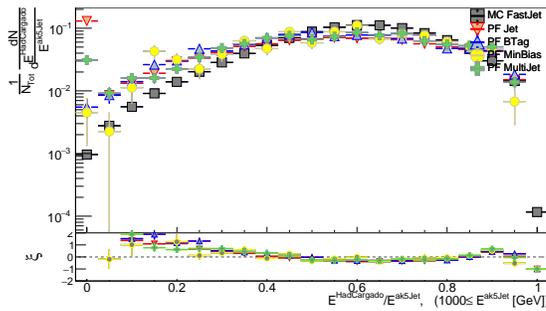
(LXXII) Hadrones cargados ($700 \leq E^{ak5Jet} < 800$ GeV)



(LXXIII) Hadrones cargados ($800 \leq E^{ak5Jet} < 900$ GeV)

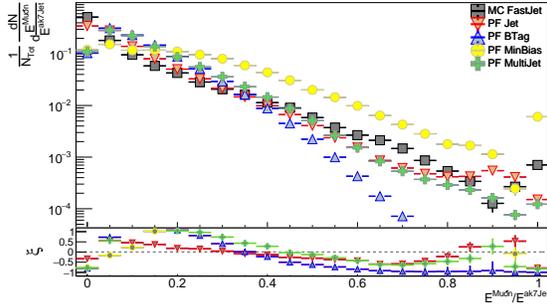


(LXXIV) Hadrones cargados ($900 \leq E^{ak5Jet} < 1000$ GeV)

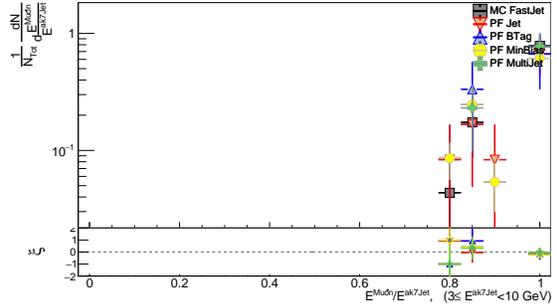


(LXXV) Hadrones cargados ($1000 \leq E^{ak5Jet}$ [GeV])

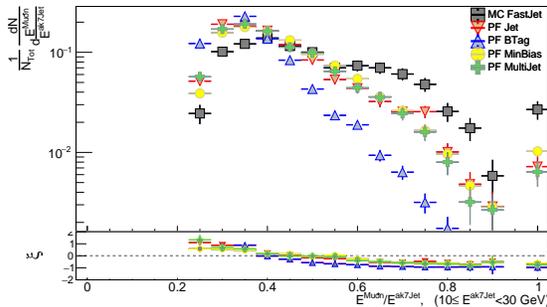
C.2.2. ak7Jets



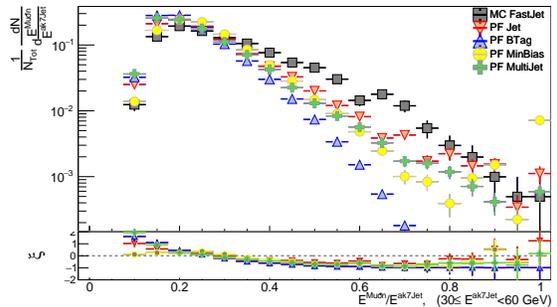
(LXXVI) Muones



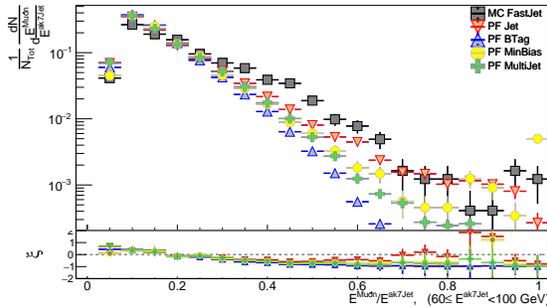
(LXXVII) Muones ($3 \leq E^{ak7Jet} < 10$ GeV)



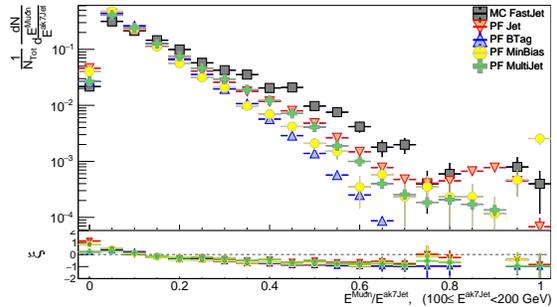
(LXXVIII) Muones ($10 \leq E^{ak7Jet} < 30$ GeV)



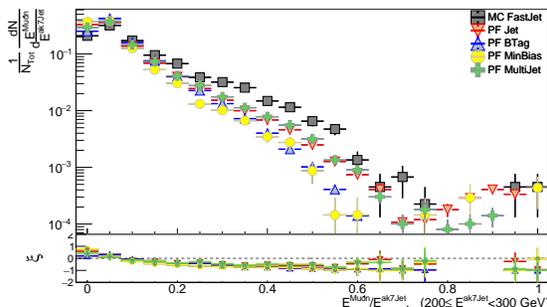
(LXXIX) Muones ($30 \leq E^{ak7Jet} < 60$ GeV)



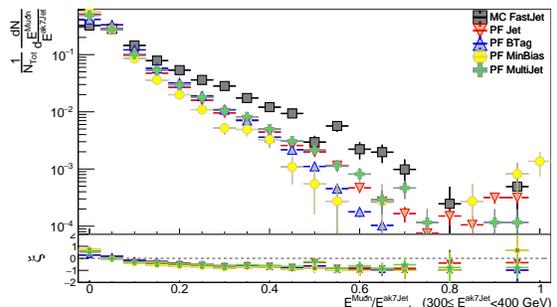
(LXXX) Muones ($60 \leq E^{ak7Jet} < 100$ GeV)



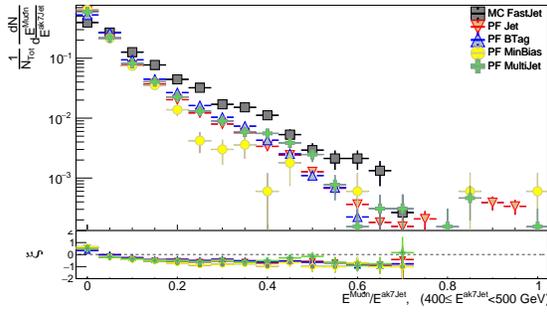
(LXXXI) Muones ($100 \leq E^{ak7Jet} < 200$ GeV)



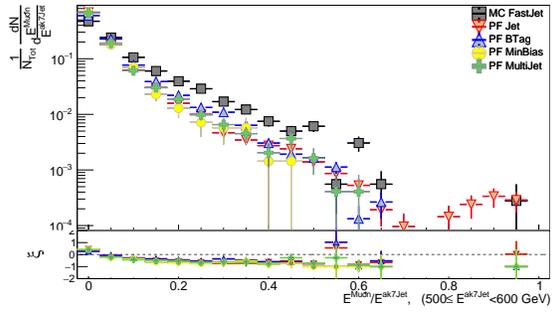
(LXXXII) Muones ($200 \leq E^{ak7Jet} < 300$ GeV)



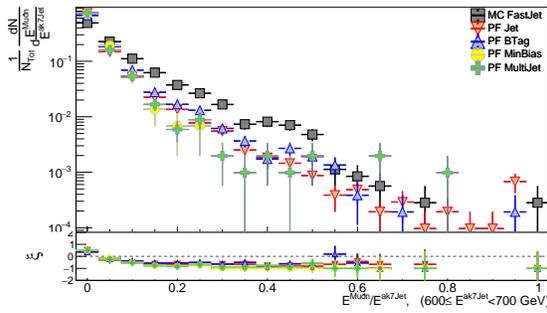
(LXXXIII) Muones ($300 \leq E^{ak7Jet} < 400$ GeV)



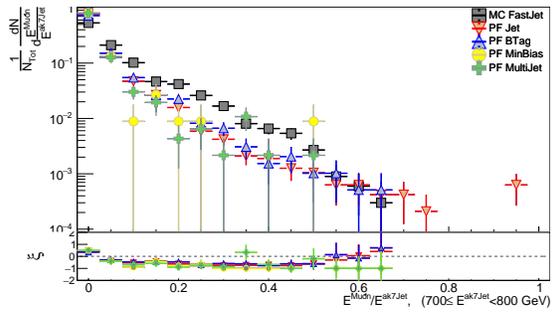
(LXXXIV) Muones ($400 \leq E^{ak7Jet} < 500$ GeV)



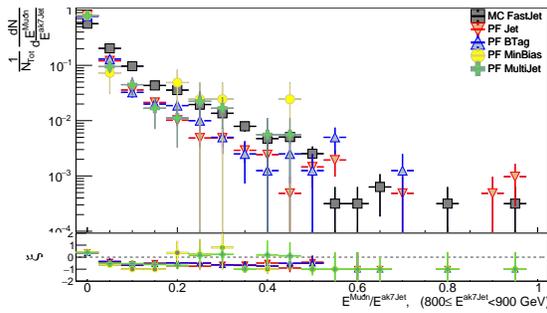
(LXXXV) Muones ($500 \leq E^{ak7Jet} < 600$ GeV)



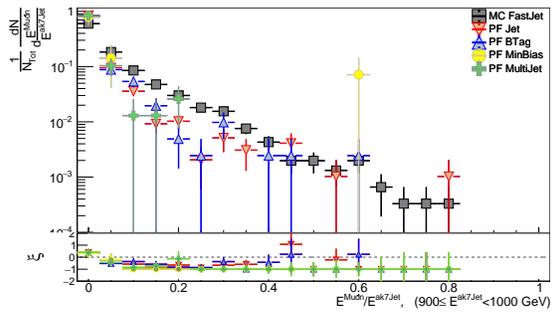
(LXXXVI) Muones ($600 \leq E^{ak7Jet} < 700$ GeV)



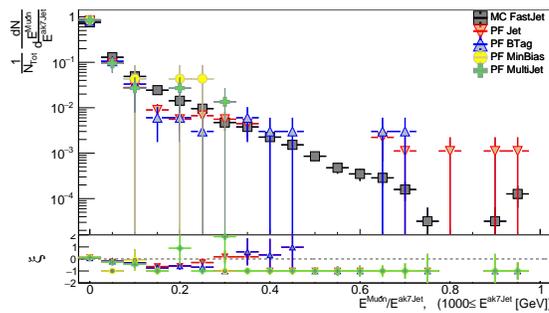
(LXXXVII) Muones ($700 \leq E^{ak7Jet} < 800$ GeV)



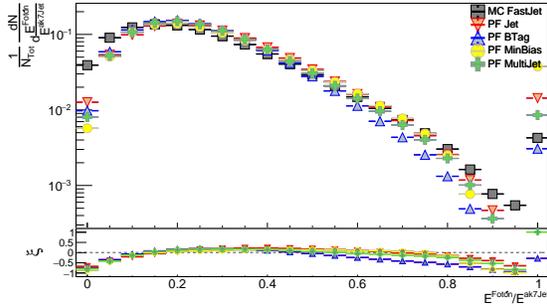
(LXXXVIII) Muones ($800 \leq E^{ak7Jet} < 900$ GeV)



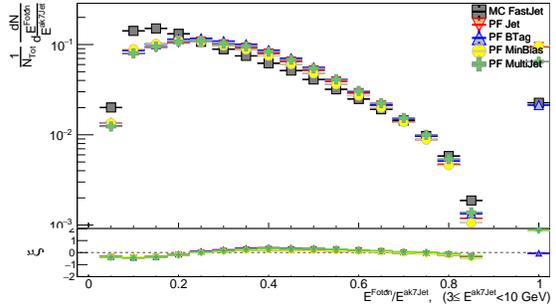
(LXXXIX) Muones ($900 \leq E^{ak7Jet} < 1000$ GeV)



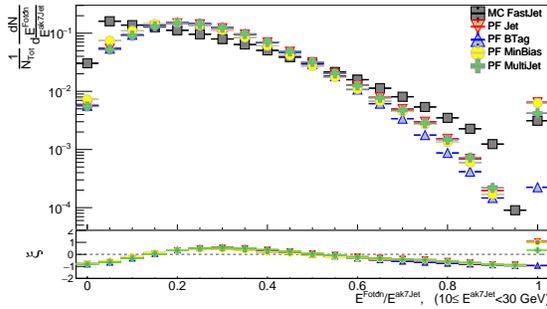
(XC) Muones ($1000 \leq E^{ak7Jet}$ [GeV])



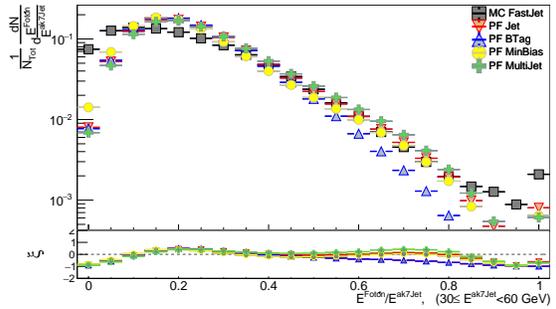
(XCI) Fotonas



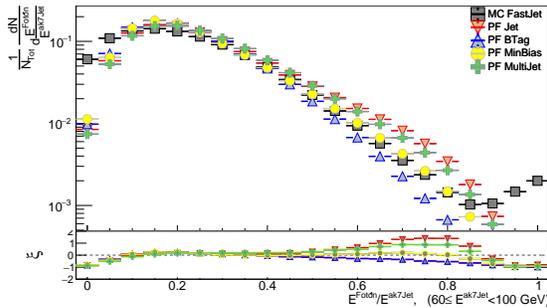
(XCII) Fotonas ($3 \leq E^{ak7Jet} < 10$ GeV)



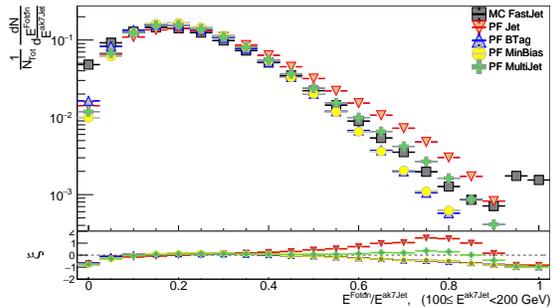
(XCIII) Fotonas ($10 \leq E^{ak7Jet} < 30$ GeV)



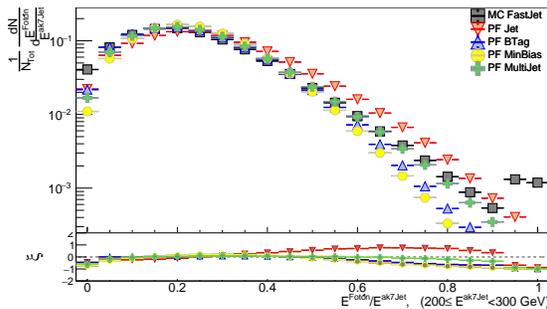
(XCIV) Fotonas ($30 \leq E^{ak7Jet} < 60$ GeV)



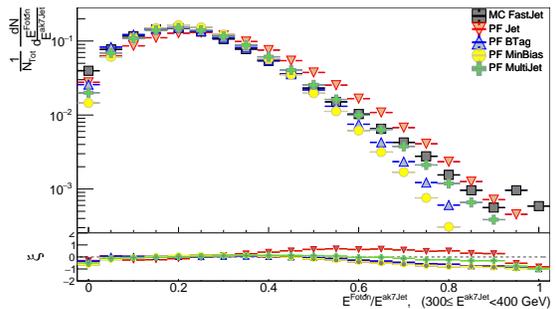
(XCV) Fotonas ($60 \leq E^{ak7Jet} < 100$ GeV)



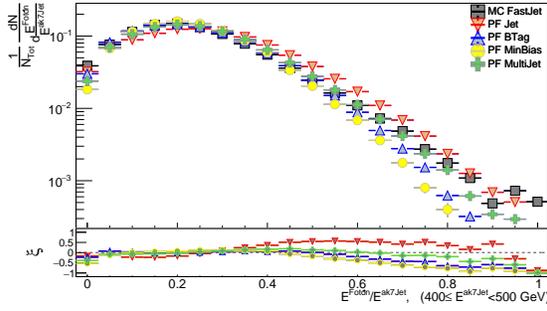
(XCVI) Fotonas ($100 \leq E^{ak7Jet} < 200$ GeV)



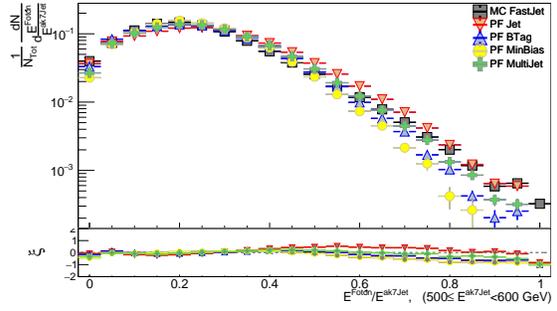
(XCVII) Fotonas ($200 \leq E^{ak7Jet} < 300$ GeV)



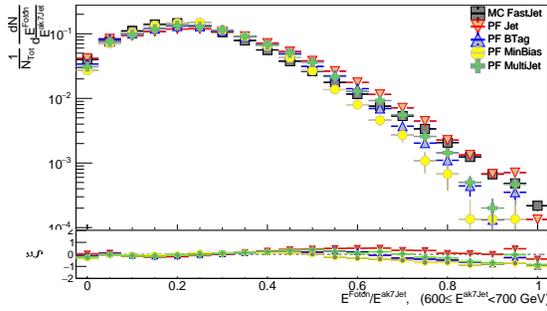
(XCVIII) Fotonas ($300 \leq E^{ak7Jet} < 400$ GeV)



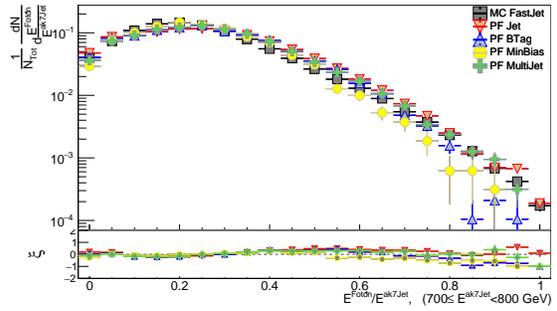
(XCIX) Fotonos ($400 \leq E^{ak7Jet} < 500$ GeV)



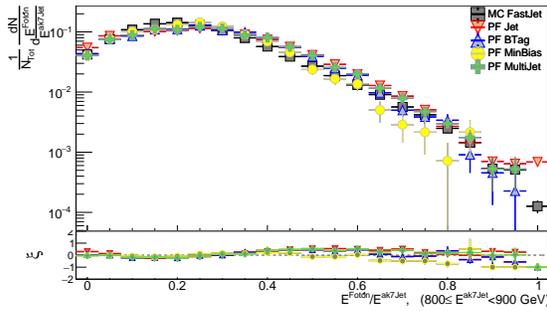
(C) Fotonos ($500 \leq E^{ak7Jet} < 600$ GeV)



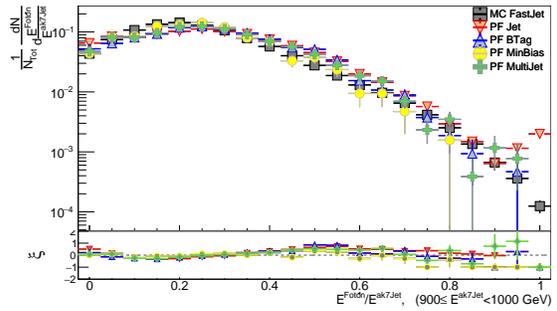
(CI) Fotonos ($600 \leq E^{ak7Jet} < 700$ GeV)



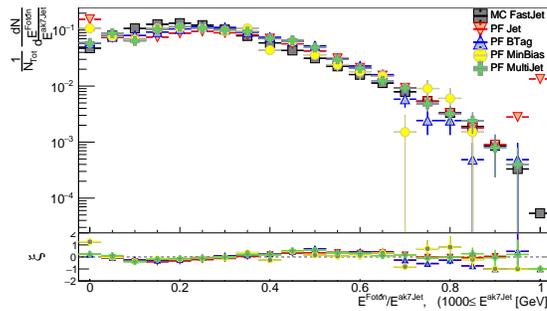
(CII) Fotonos ($700 \leq E^{ak7Jet} < 800$ GeV)



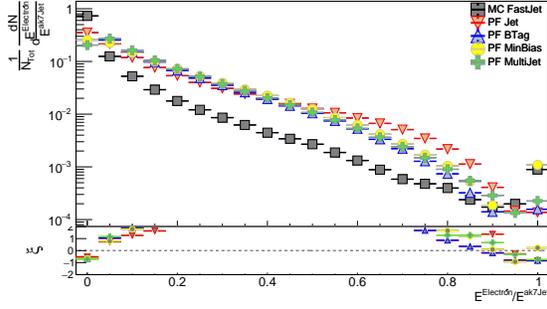
(CIII) Fotonos ($800 \leq E^{ak7Jet} < 900$ GeV)



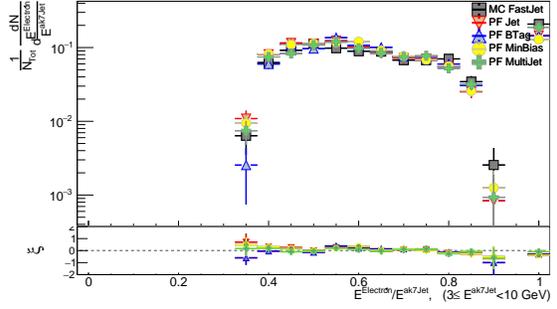
(CIV) Fotonos ($900 \leq E^{ak7Jet} < 1000$ GeV)



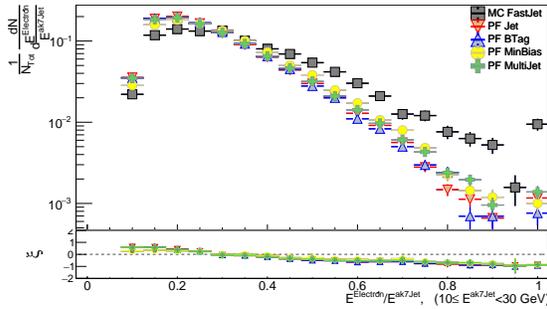
(CV) Fotonos ($1000 \leq E^{ak7Jet}$ [GeV])



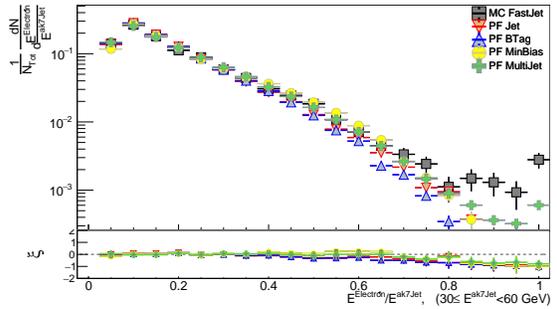
(CVI) Electrones



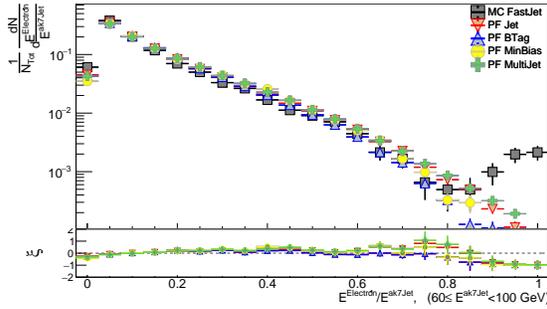
(CVII) Electrones ($3 \leq E^{ak7Jet} < 10$ GeV)



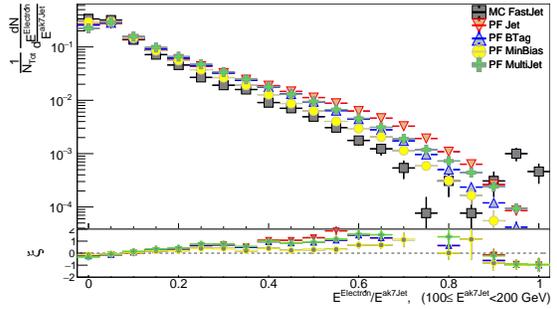
(CVIII) Electrones ($10 \leq E^{ak7Jet} < 30$ GeV)



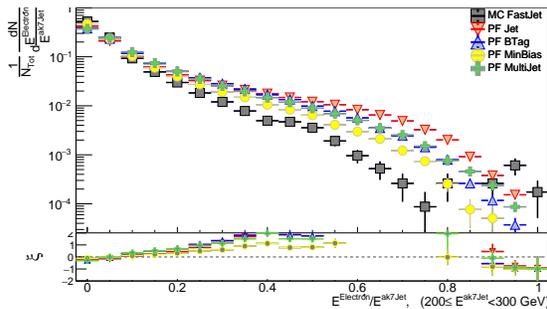
(CIX) Electrones ($30 \leq E^{ak7Jet} < 60$ GeV)



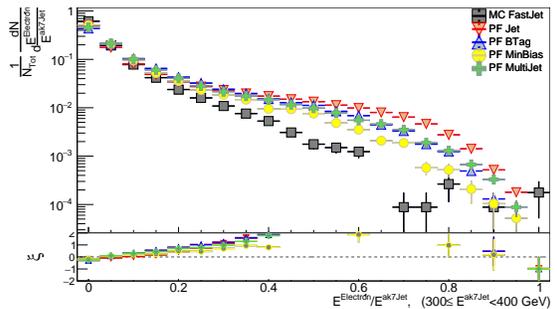
(CX) Electrones ($60 \leq E^{ak7Jet} < 100$ GeV)



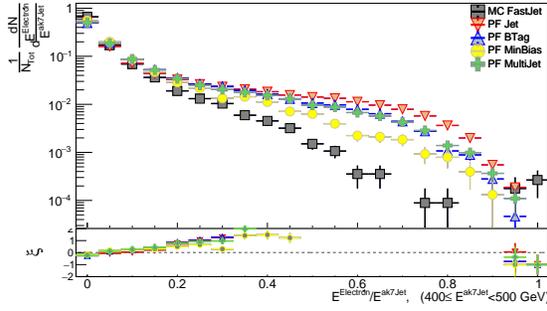
(CXI) Electrones ($100 \leq E^{ak7Jet} < 200$ GeV)



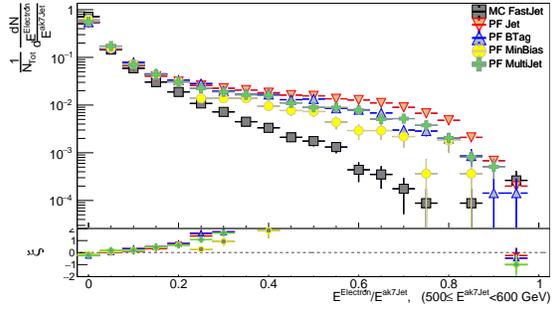
(CXII) Electrones ($200 \leq E^{ak7Jet} < 300$ GeV)



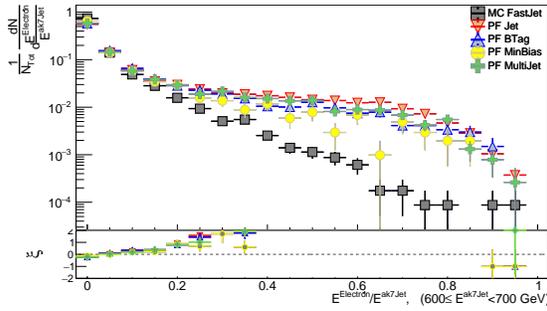
(CXIII) Electrones ($300 \leq E^{ak7Jet} < 400$ GeV)



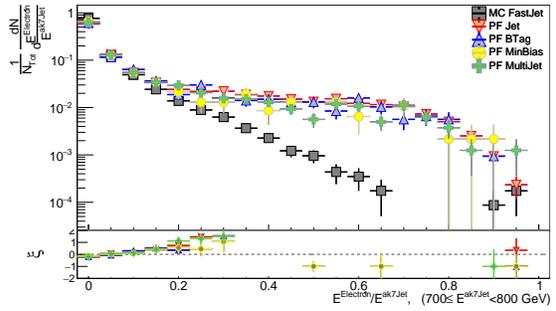
(CXIV) Electrones ($400 \leq E^{ak7Jet} < 500$ GeV)



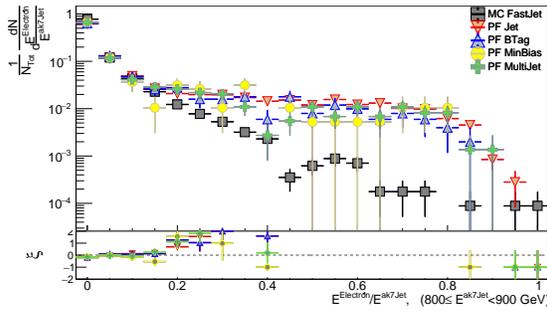
(CXV) Electrones ($500 \leq E^{ak7Jet} < 600$ GeV)



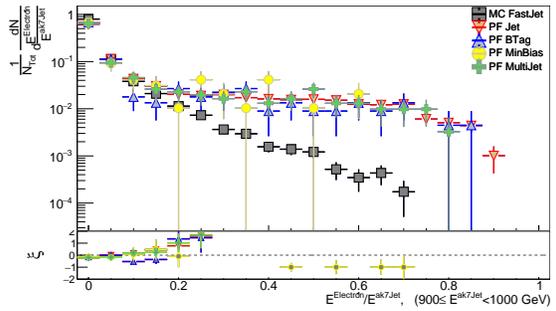
(CXVI) Electrones ($600 \leq E^{ak7Jet} < 700$ GeV)



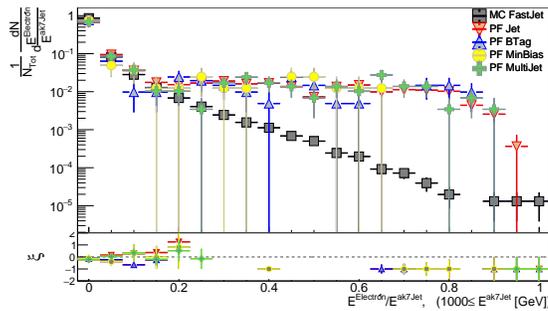
(CXVII) Electrones ($700 \leq E^{ak7Jet} < 800$ GeV)



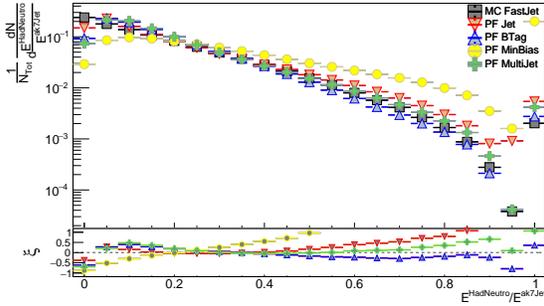
(CXVIII) Electrones ($800 \leq E^{ak7Jet} < 900$ GeV)



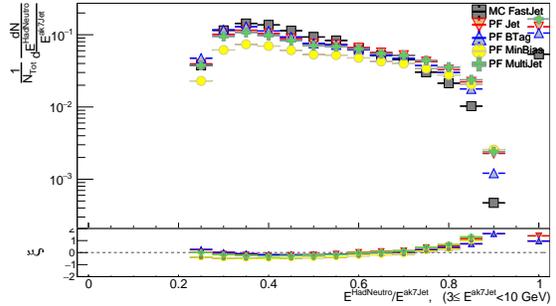
(CXIX) Electrones ($900 \leq E^{ak7Jet} < 1000$ GeV)



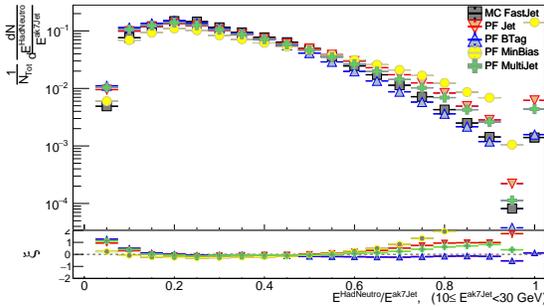
(CXX) Electrones ($1000 \leq E^{ak7Jet}$ [GeV])



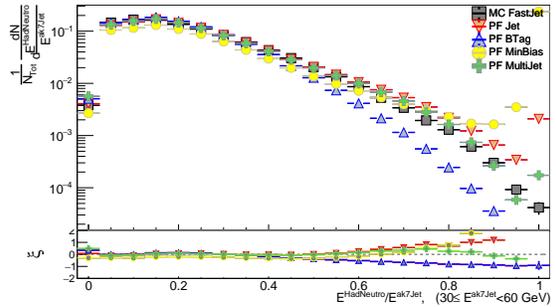
(CXXI) Hadrones neutros



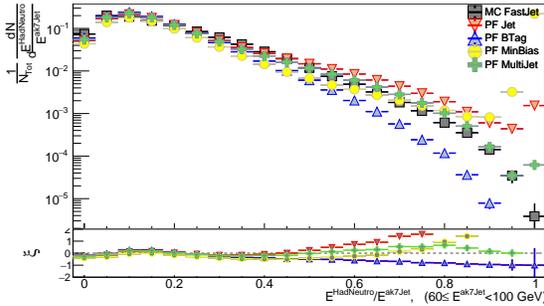
(CXXII) Hadrones neutros ($3 \leq E^{ak7Jet} < 10$ GeV)



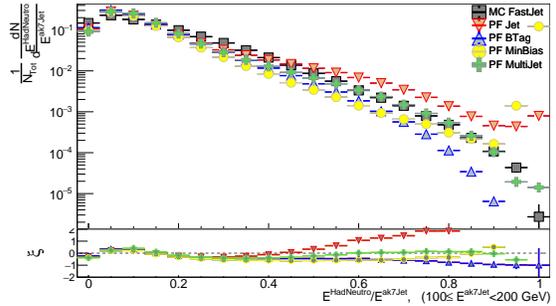
(CXXIII) Hadrones neutros ($10 \leq E^{ak7Jet} < 30$ GeV)



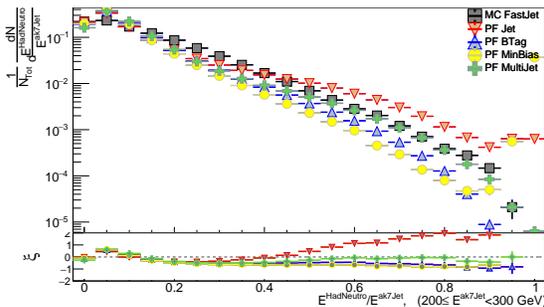
(CXXIV) Hadrones neutros ($30 \leq E^{ak7Jet} < 60$ GeV)



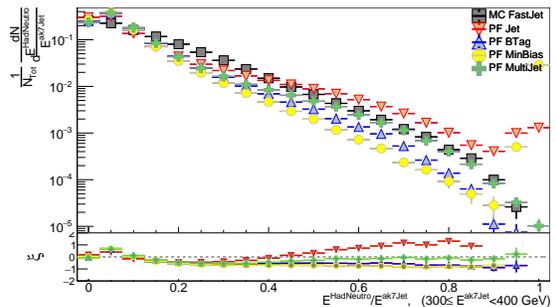
(CXXV) Hadrones neutros ($60 \leq E^{ak7Jet} < 100$ GeV)



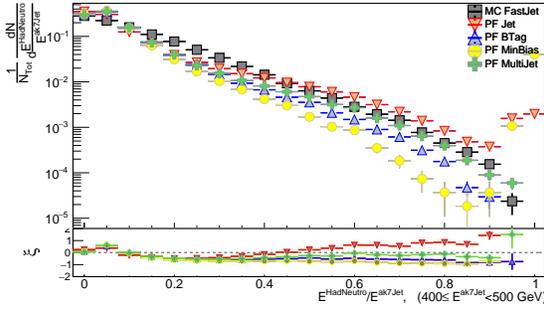
(CXXVI) Hadrones neutros ($100 \leq E^{ak7Jet} < 200$ GeV)



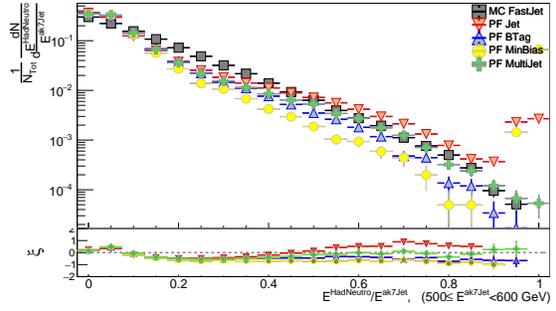
(CXXVII) Hadrones neutros ($200 \leq E^{ak7Jet} < 300$ GeV)



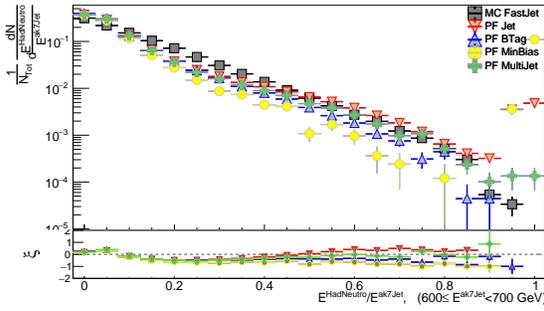
(CXXVIII) Hadrones neutros ($300 \leq E^{ak7Jet} < 400$ GeV)



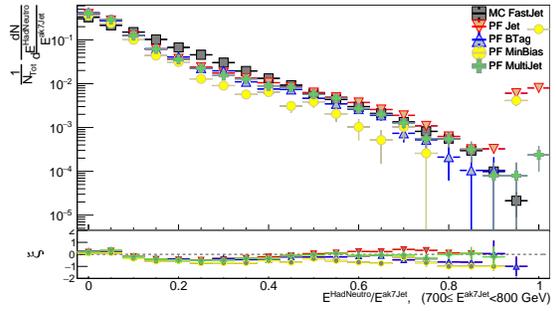
(CXXIX) Hadrones neutros ($400 \leq E^{ak7Jet} < 500$ GeV)



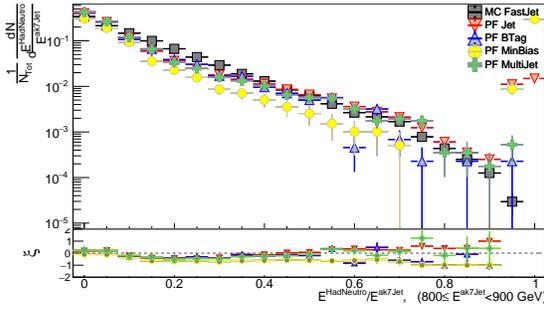
(CXXX) Hadrones neutros ($500 \leq E^{ak7Jet} < 600$ GeV)



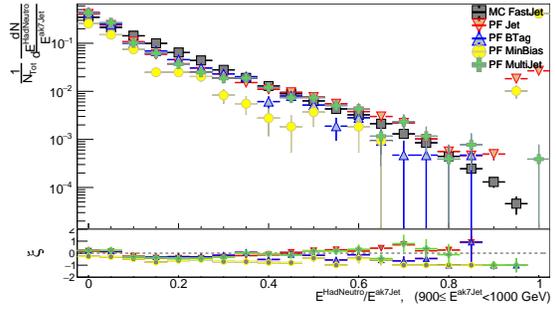
(CXXXI) Hadrones neutros ($600 \leq E^{ak7Jet} < 700$ GeV)



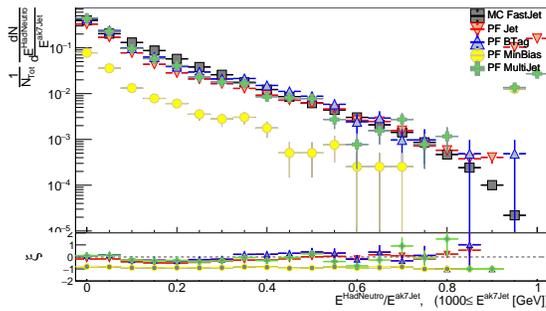
(CXXXII) Hadrones neutros ($700 \leq E^{ak7Jet} < 800$ GeV)



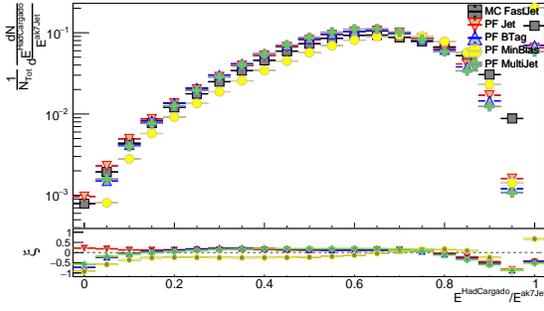
(CXXXIII) Hadrones neutros ($800 \leq E^{ak7Jet} < 900$ GeV)



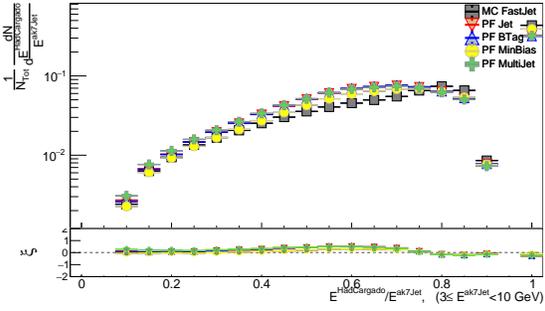
(CXXXIV) Hadrones neutros ($900 \leq E^{ak7Jet} < 1000$ GeV)



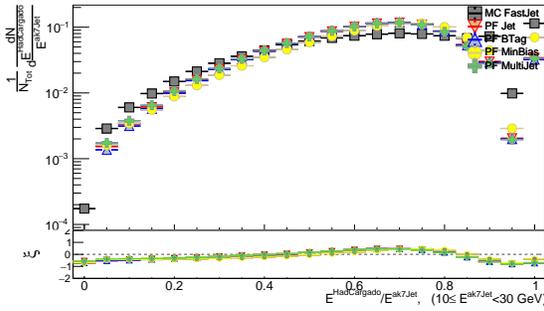
(CXXXV) Hadrones neutros ($1000 \leq E^{ak7Jet}$ [GeV])



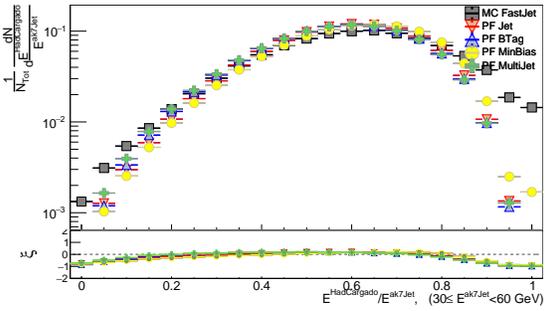
(CXXXVI) Hadrones cargados



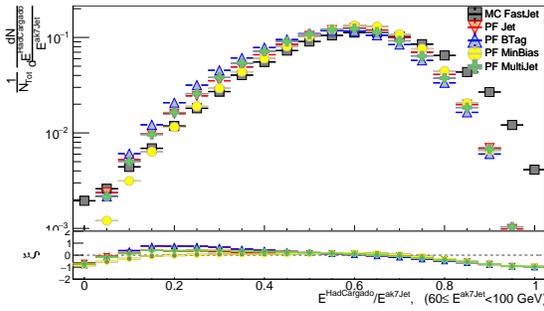
(CXXXVII) Hadrones cargados ($3 \leq E^{ak7Jet} < 10$ GeV)



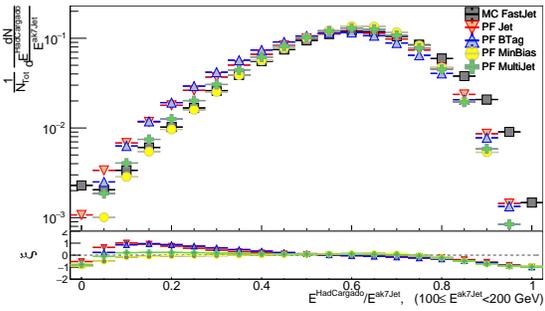
(CXXXVIII) Hadrones cargados ($10 \leq E^{ak7Jet} < 30$ GeV)



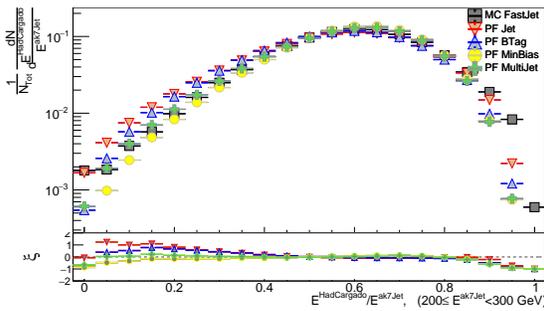
(CXXXIX) Hadrones cargados ($30 \leq E^{ak7Jet} < 60$ GeV)



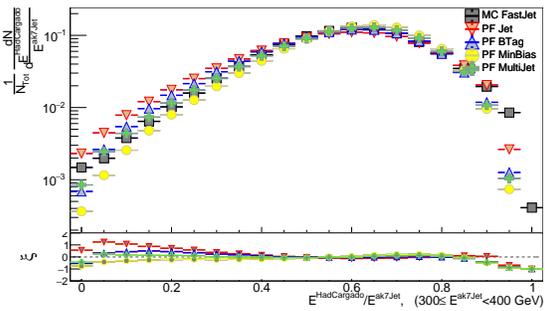
(CXL) Hadrones cargados ($60 \leq E^{ak7Jet} < 100$ GeV)



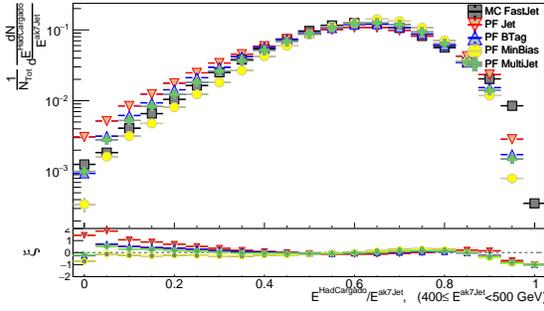
(CXLI) Hadrones cargados ($100 \leq E^{ak7Jet} < 200$ GeV)



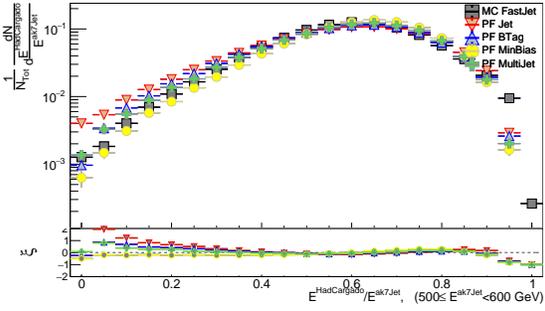
(CXLII) Hadrones cargados ($200 \leq E^{ak7Jet} < 300$ GeV)



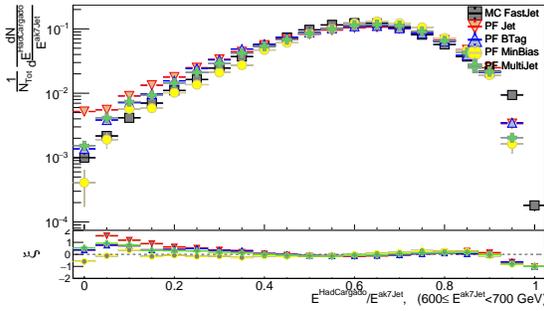
(CXLIII) Hadrones cargados ($300 \leq E^{ak7Jet} < 400$ GeV)



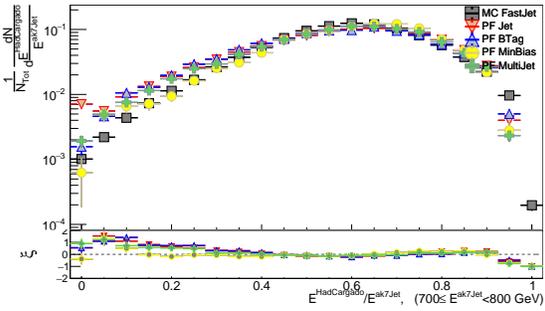
(CXLIV) Hadrones cargados ($400 \leq E^{ak7Jet} < 500$ GeV)



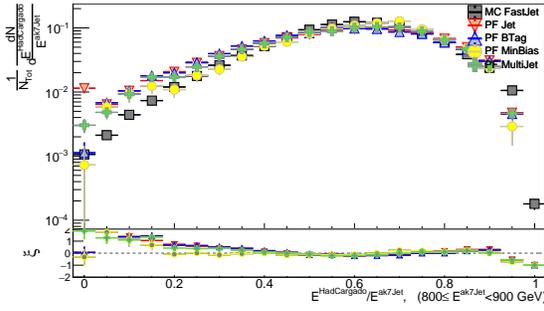
(CXLV) Hadrones cargados ($500 \leq E^{ak7Jet} < 600$ GeV)



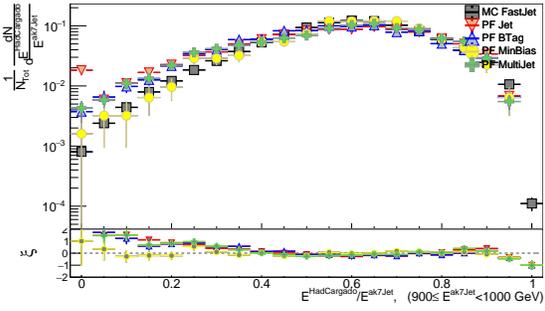
(CXLVI) Hadrones cargados ($600 \leq E^{ak7Jet} < 700$ GeV)



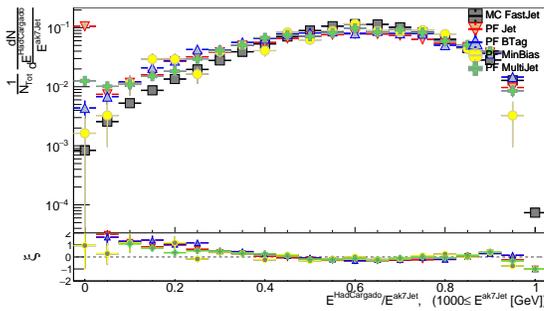
(CXLVII) Hadrones cargados ($700 \leq E^{ak7Jet} < 800$ GeV)



(CXLVIII) Hadrones cargados ($800 \leq E^{ak7Jet} < 900$ GeV)

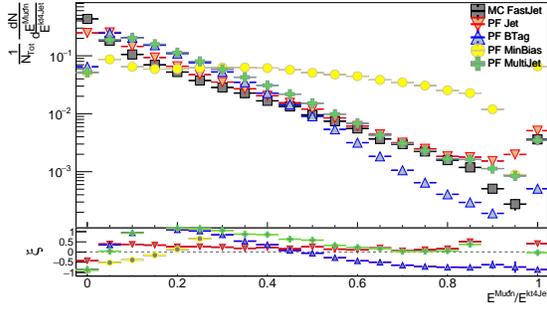


(CXLIX) Hadrones cargados ($900 \leq E^{ak7Jet} < 1000$ GeV)

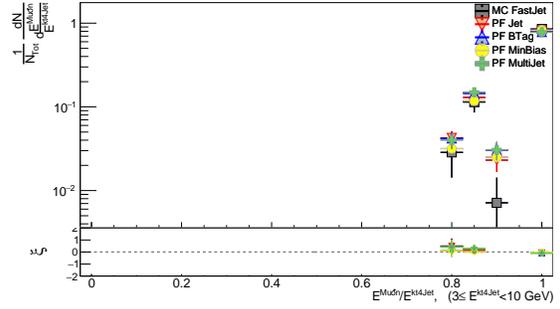


(CL) Hadrones cargados ($1000 \leq E^{ak7Jet}$ [GeV])

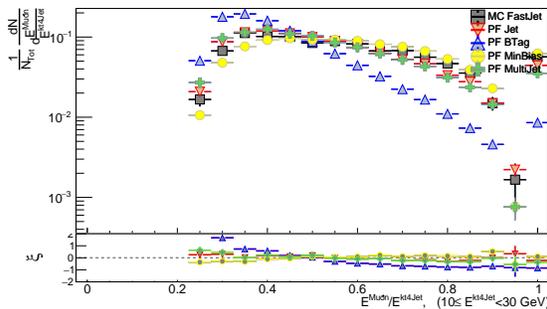
C.2.3. kt4Jets



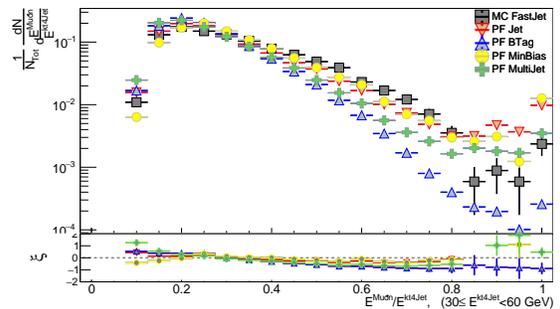
(CLI) Muones



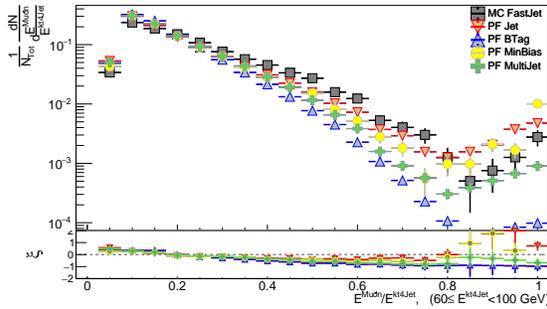
(CLII) Muones ($3 \leq E^{kt4Jet} < 10$ GeV)



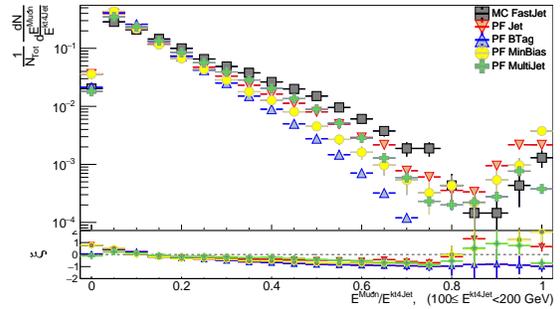
(CLIII) Muones ($10 \leq E^{kt4Jet} < 30$ GeV)



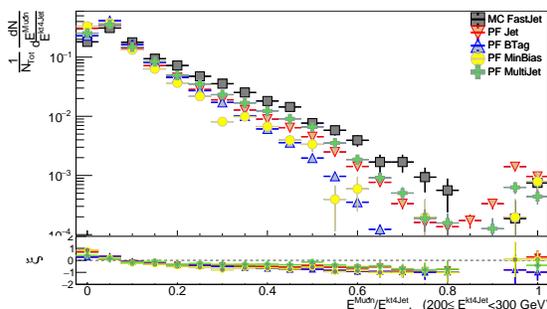
(CLIV) Muones ($30 \leq E^{kt4Jet} < 60$ GeV)



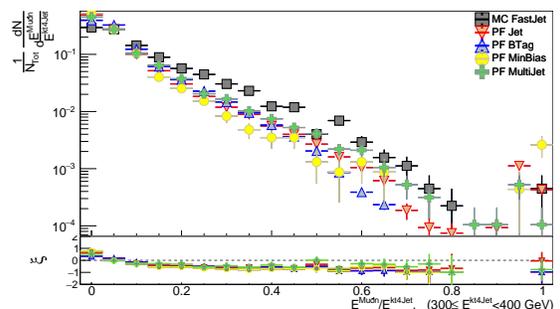
(CLV) Muones ($60 \leq E^{kt4Jet} < 100$ GeV)



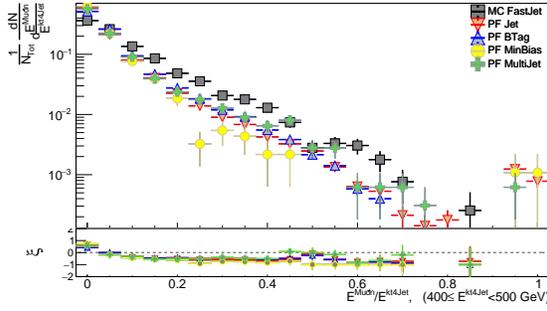
(CLVI) Muones ($100 \leq E^{kt4Jet} < 200$ GeV)



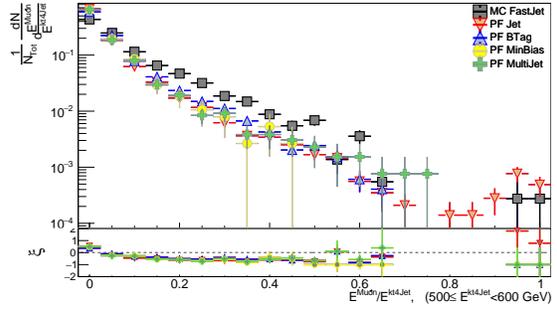
(CLVII) Muones ($200 \leq E^{kt4Jet} < 300$ GeV)



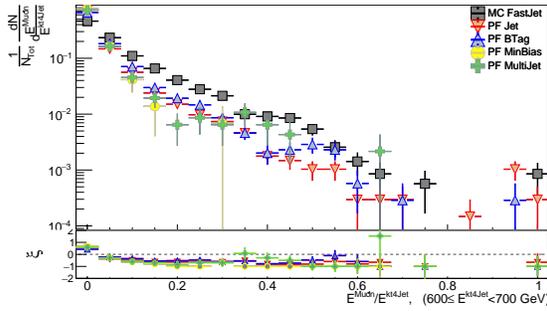
(CLVIII) Muones ($300 \leq E^{kt4Jet} < 400$ GeV)



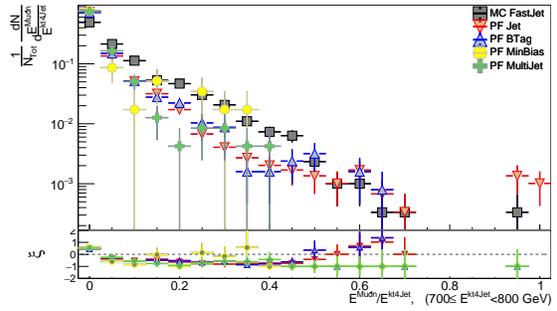
(CLIX) Muones ($400 \leq E^{kt4Jet} < 500$ GeV)



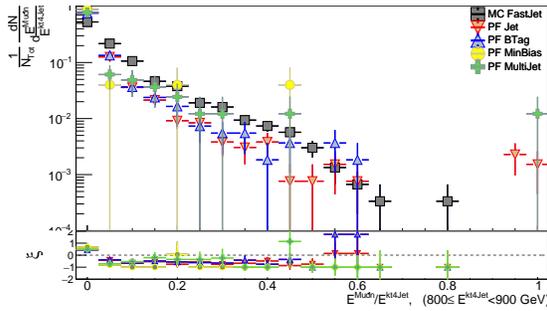
(CLX) Muones ($500 \leq E^{kt4Jet} < 600$ GeV)



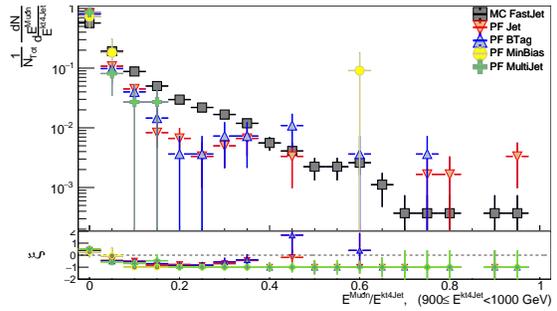
(CLXI) Muones ($600 \leq E^{kt4Jet} < 700$ GeV)



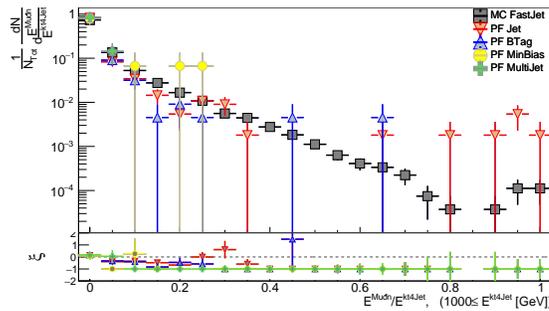
(CLXII) Muones ($700 \leq E^{kt4Jet} < 800$ GeV)



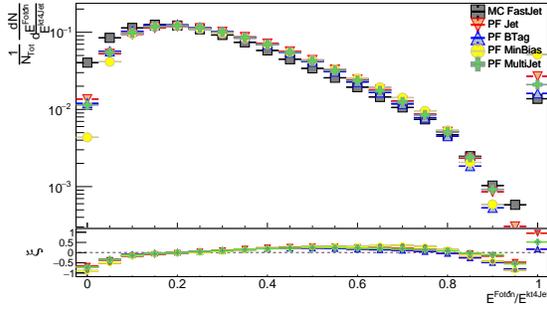
(CLXIII) Muones ($800 \leq E^{kt4Jet} < 900$ GeV)



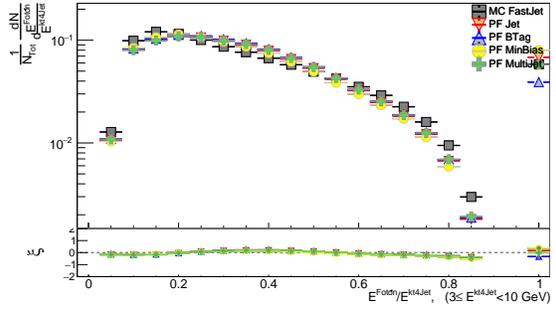
(CLXIV) Muones ($900 \leq E^{kt4Jet} < 1000$ GeV)



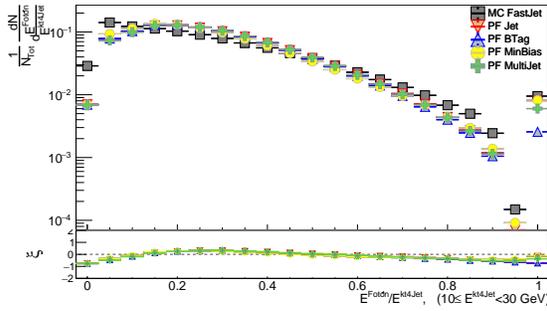
(CLXV) Muones ($1000 \leq E^{kt4Jet}$ [GeV])



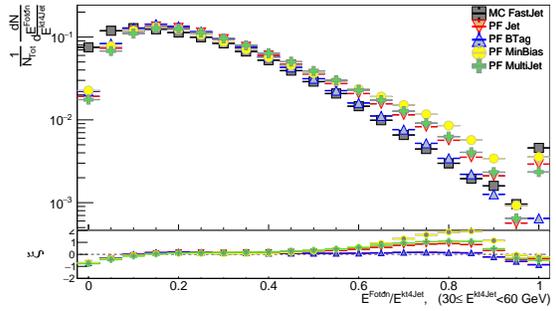
(CLXVI) Fotonos



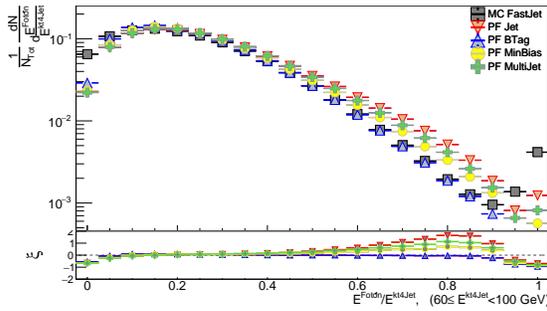
(CLXVII) Fotonos ($3 \leq E^{kt4Jet} < 10$ GeV)



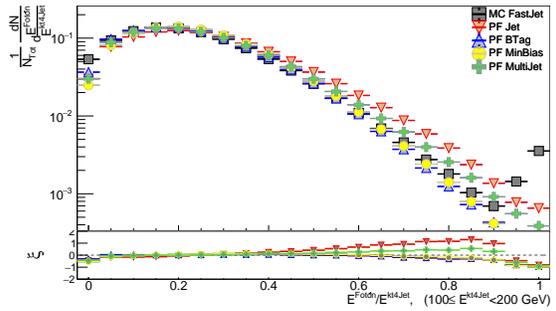
(CLXVIII) Fotonos ($10 \leq E^{kt4Jet} < 30$ GeV)



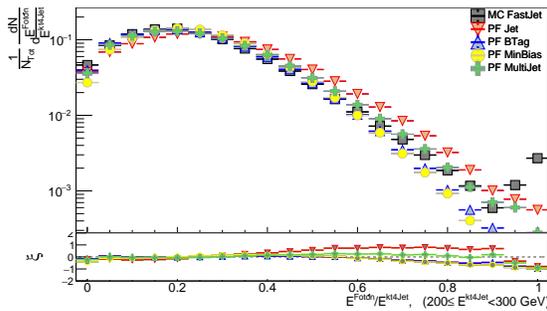
(CLXIX) Fotonos ($30 \leq E^{kt4Jet} < 60$ GeV)



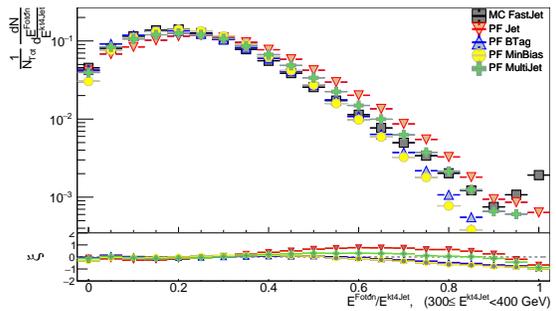
(CLXX) Fotonos ($60 \leq E^{kt4Jet} < 100$ GeV)



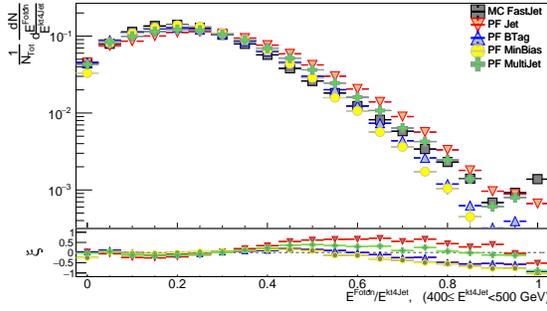
(CLXXI) Fotonos ($100 \leq E^{kt4Jet} < 200$ GeV)



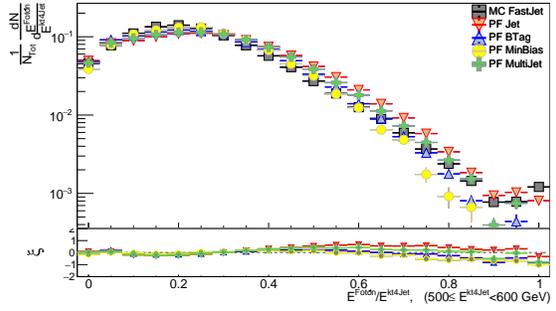
(CLXXII) Fotonos ($200 \leq E^{kt4Jet} < 300$ GeV)



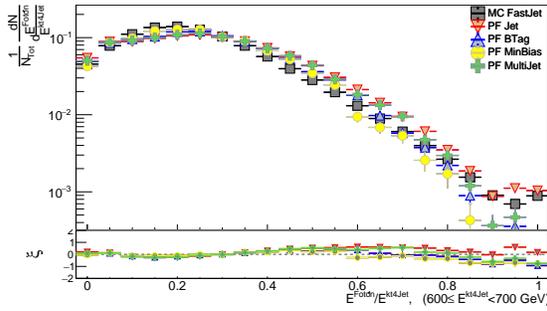
(CLXXIII) Fotonos ($300 \leq E^{kt4Jet} < 400$ GeV)



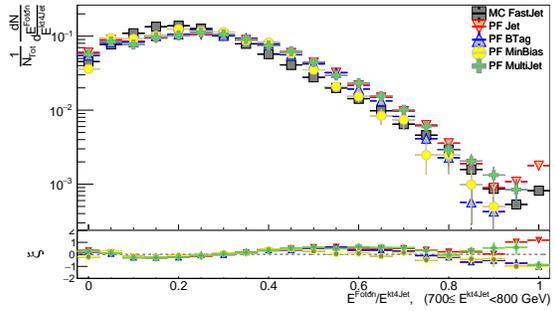
(CLXXIV) Fotonos ($400 \leq E^{kt4Jet} < 500$ GeV)



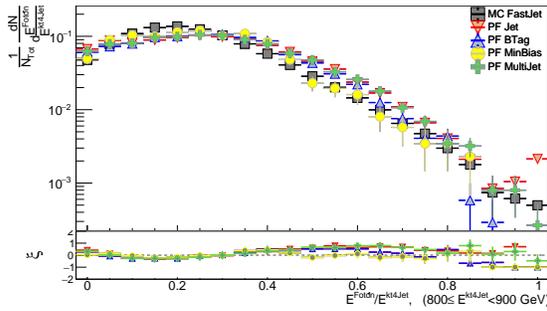
(CLXXV) Fotonos ($500 \leq E^{kt4Jet} < 600$ GeV)



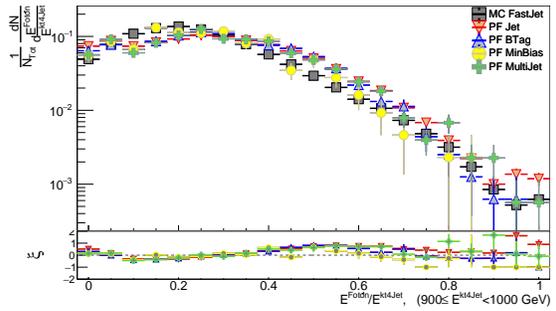
(CLXXVI) Fotonos ($600 \leq E^{kt4Jet} < 700$ GeV)



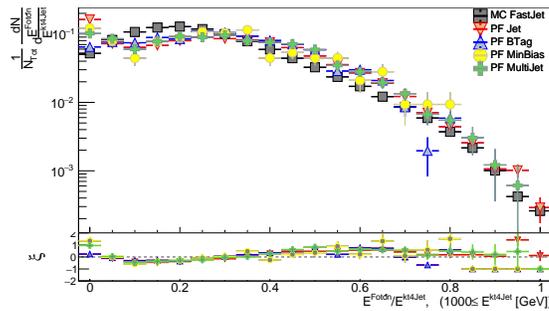
(CLXXVII) Fotonos ($700 \leq E^{kt4Jet} < 800$ GeV)



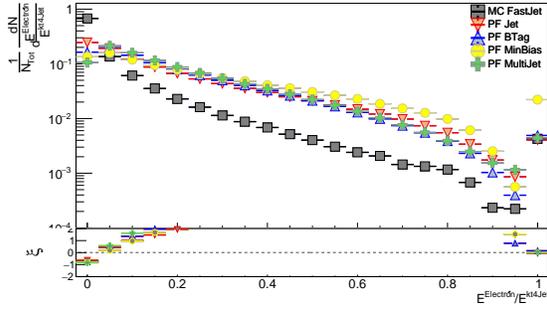
(CLXXVIII) Fotonos ($800 \leq E^{kt4Jet} < 900$ GeV)



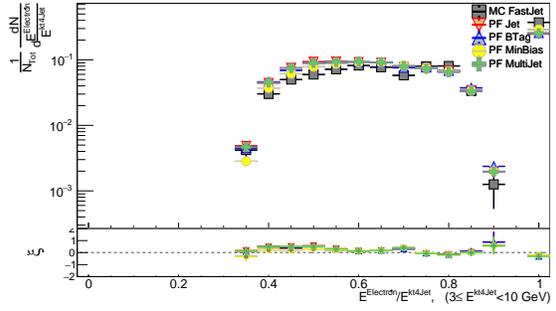
(CLXXIX) Fotonos ($900 \leq E^{kt4Jet} < 1000$ GeV)



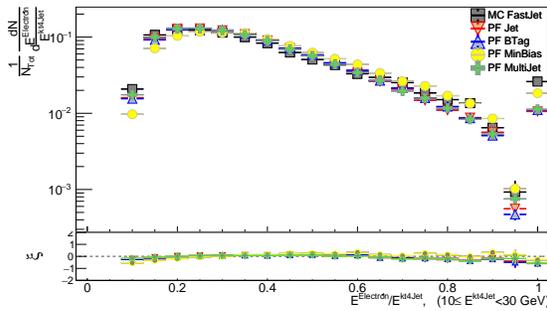
(CLXXX) Fotonos ($1000 \leq E^{kt4Jet}$ [GeV])



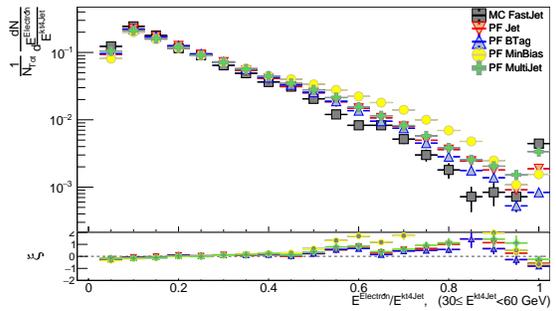
(CLXXXI) Electrones



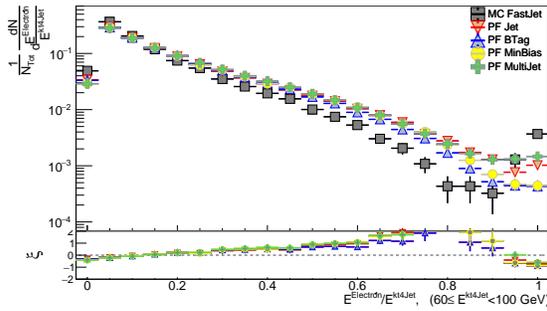
(CLXXXII) Electrones ($3 \leq E^{kt4Jet} < 10$ GeV)



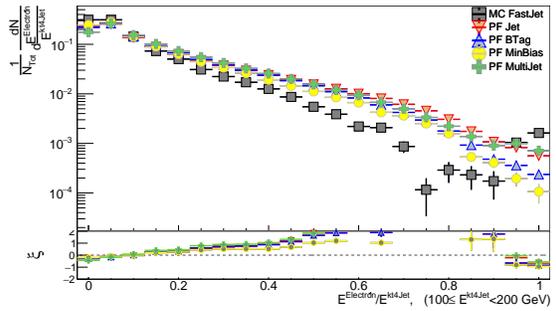
(CLXXXIII) Electrones ($10 \leq E^{kt4Jet} < 30$ GeV)



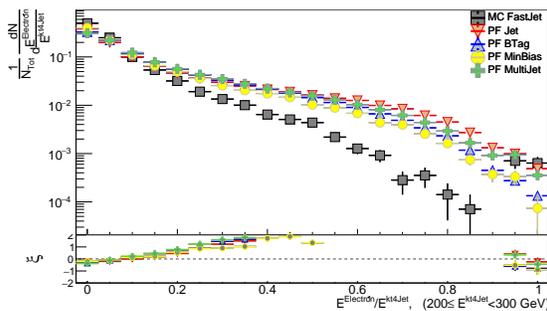
(CLXXXIV) Electrones ($30 \leq E^{kt4Jet} < 60$ GeV)



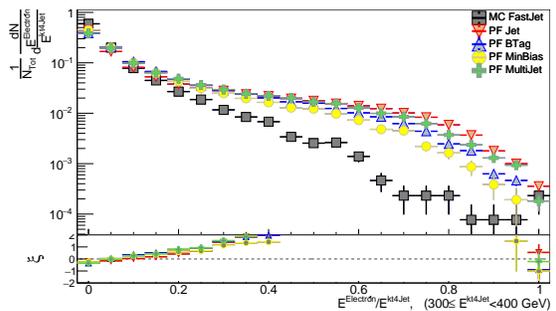
(CLXXXV) Electrones ($60 \leq E^{kt4Jet} < 100$ GeV)



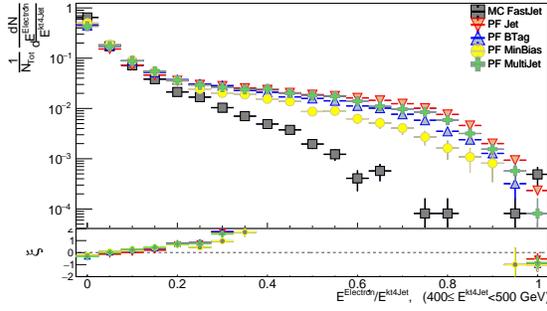
(CLXXXVI) Electrones ($100 \leq E^{kt4Jet} < 200$ GeV)



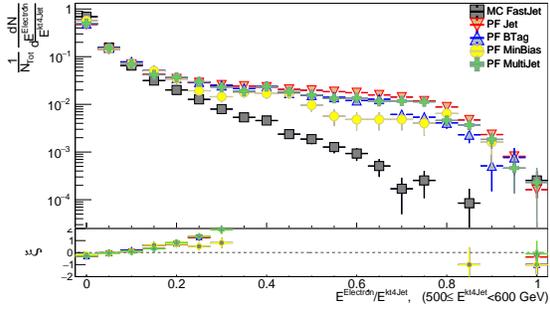
(CLXXXVII) Electrones ($200 \leq E^{kt4Jet} < 300$ GeV)



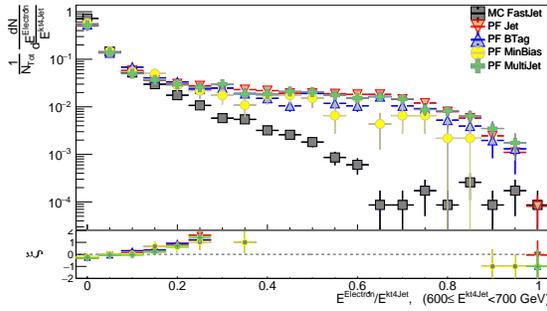
(CLXXXVIII) Electrones ($300 \leq E^{kt4Jet} < 400$ GeV)



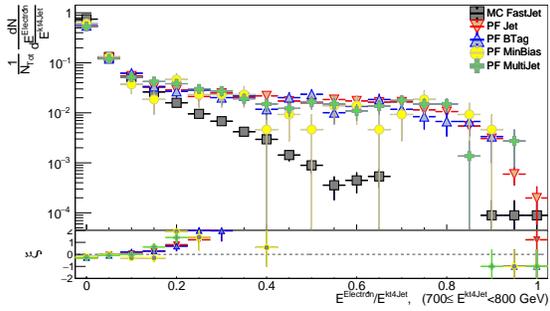
(CLXXXIX) Electrones ($400 \leq E^{kt4Jet} < 500$ GeV)



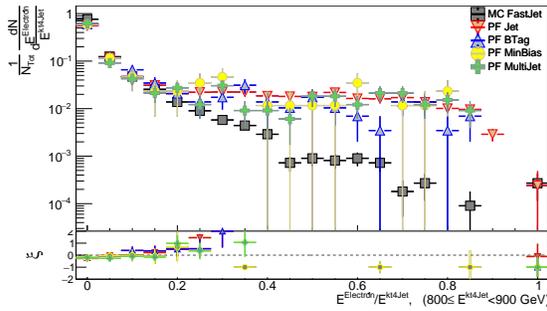
(CXC) Electrones ($500 \leq E^{kt4Jet} < 600$ GeV)



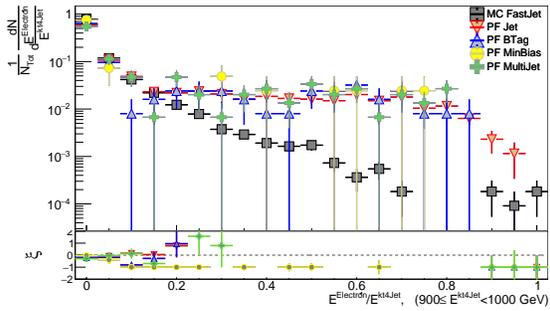
(CXCI) Electrones ($600 \leq E^{kt4Jet} < 700$ GeV)



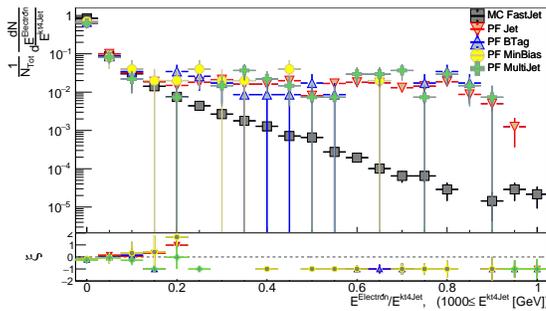
(CXCII) Electrones ($700 \leq E^{kt4Jet} < 800$ GeV)



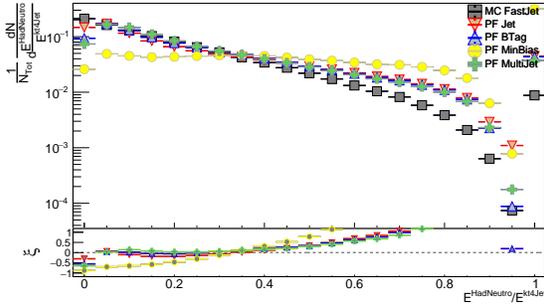
(CXCIII) Electrones ($800 \leq E^{kt4Jet} < 900$ GeV)



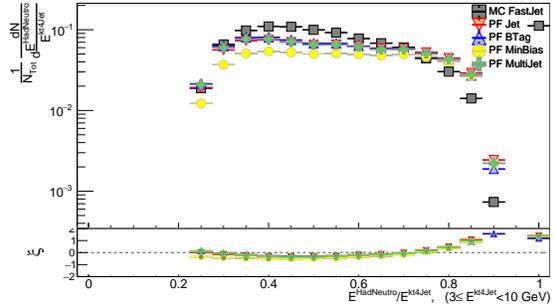
(CXCIV) Electrones ($900 \leq E^{kt4Jet} < 1000$ GeV)



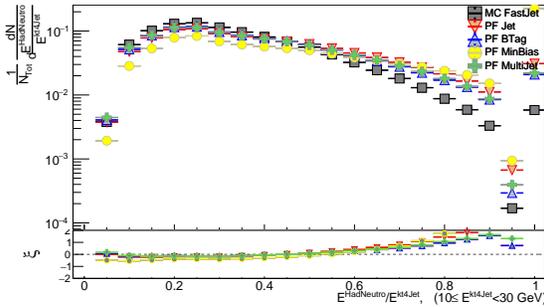
(CXCV) Electrones ($1000 \leq E^{kt4Jet}$ [GeV])



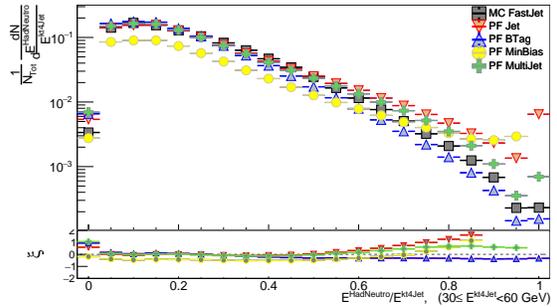
(CXCVI) Hadrones neutros



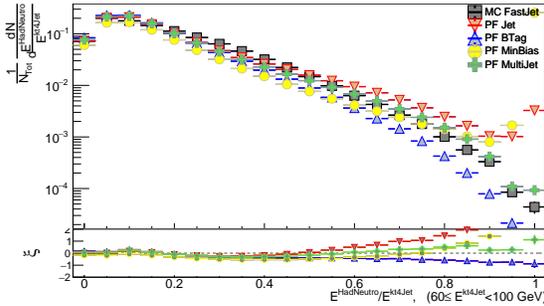
(CXCVII) Hadrones neutros ($3 \leq E^{kt4Jet} < 10$ GeV)



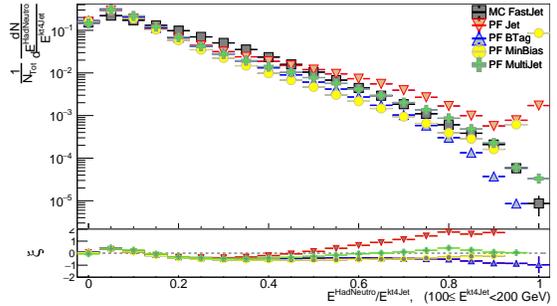
(CXCVIII) Hadrones neutros ($10 \leq E^{kt4Jet} < 30$ GeV)



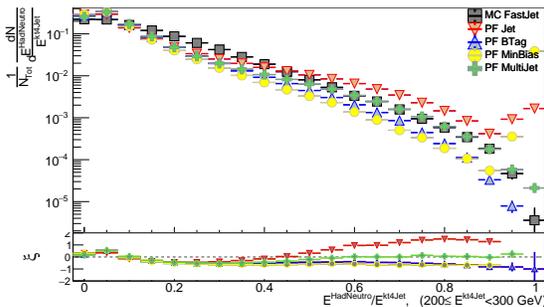
(CXCIX) Hadrones neutros ($30 \leq E^{kt4Jet} < 60$ GeV)



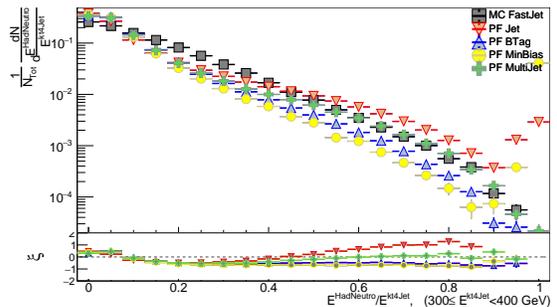
(CC) Hadrones neutros ($60 \leq E^{kt4Jet} < 100$ GeV)



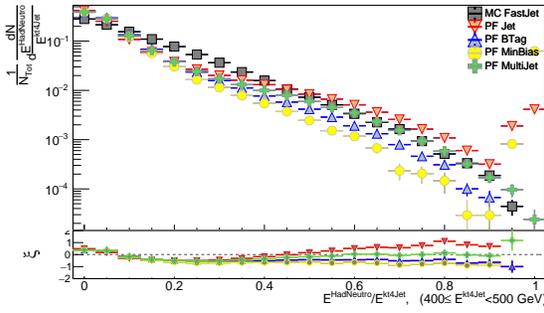
(CCI) Hadrones neutros ($100 \leq E^{kt4Jet} < 200$ GeV)



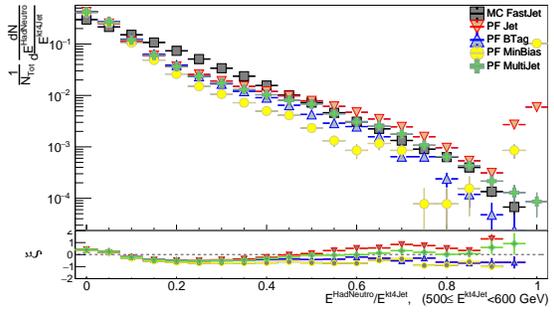
(CCII) Hadrones neutros ($200 \leq E^{kt4Jet} < 300$ GeV)



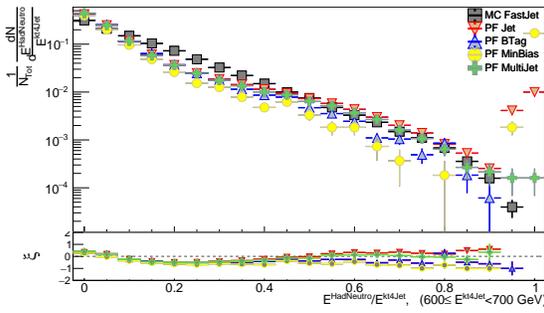
(CCIII) Hadrones neutros ($300 \leq E^{kt4Jet} < 400$ GeV)



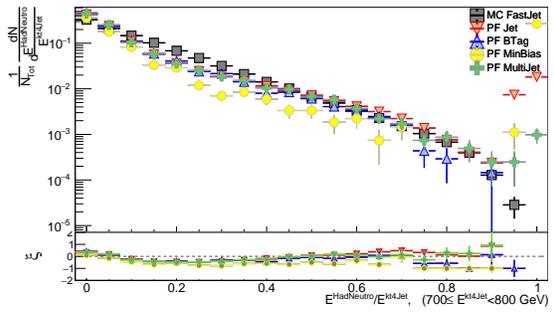
(CCIV) Hadrones neutros ($400 \leq E^{kt4Jet} < 500$ GeV)



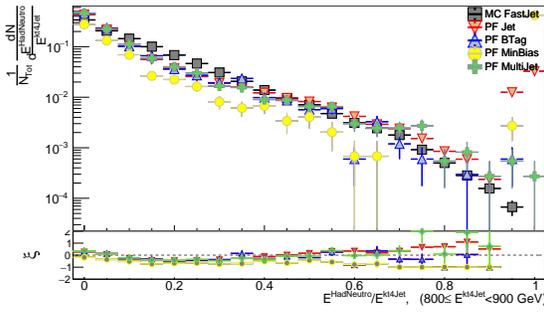
(CCV) Hadrones neutros ($500 \leq E^{kt4Jet} < 600$ GeV)



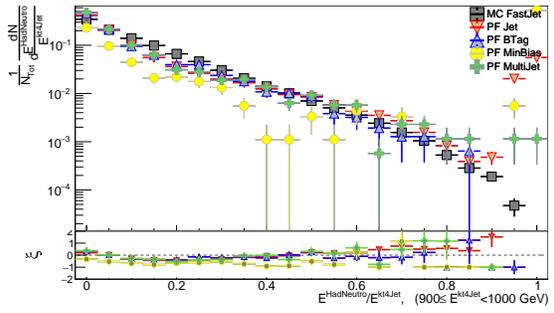
(CCVI) Hadrones neutros ($600 \leq E^{kt4Jet} < 700$ GeV)



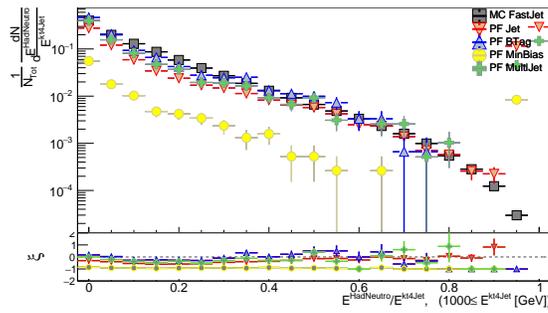
(CCVII) Hadrones neutros ($700 \leq E^{kt4Jet} < 800$ GeV)



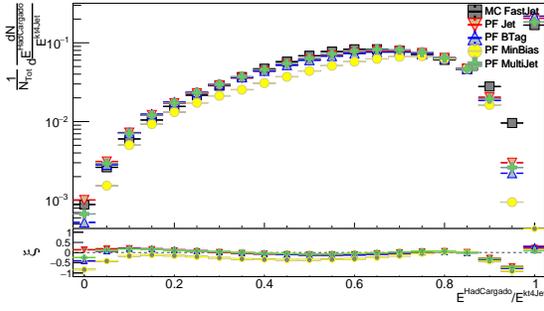
(CCVIII) Hadrones neutros ($800 \leq E^{kt4Jet} < 900$ GeV)



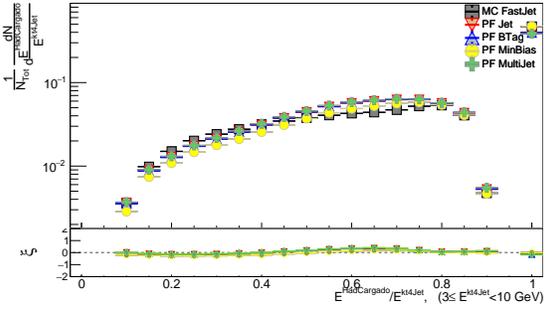
(CCIX) Hadrones neutros ($900 \leq E^{kt4Jet} < 1000$ GeV)



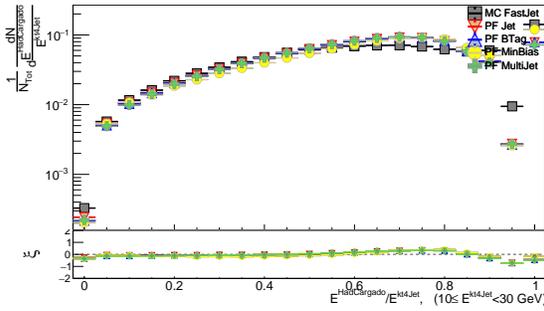
(CCX) Hadrones neutros ($1000 \leq E^{kt4Jet}$ [GeV])



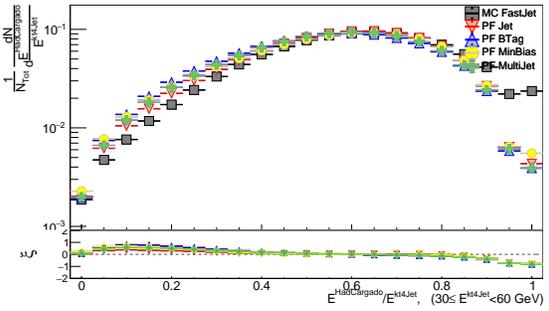
(CCXI) Hadrones cargados



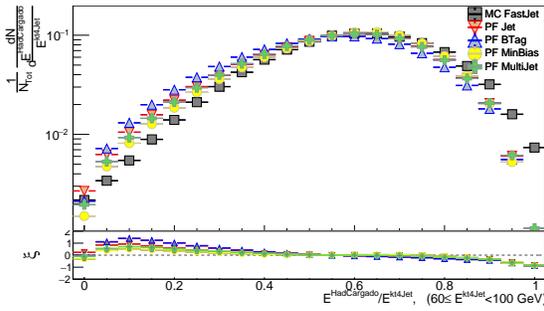
(CCXII) Hadrones cargados ($3 \leq E^{kt4Jet} < 10$ GeV)



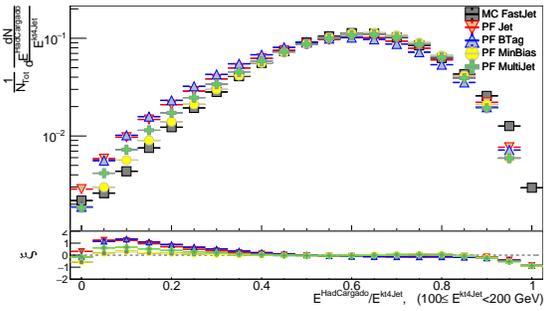
(CCXIII) Hadrones cargados ($10 \leq E^{kt4Jet} < 30$ GeV)



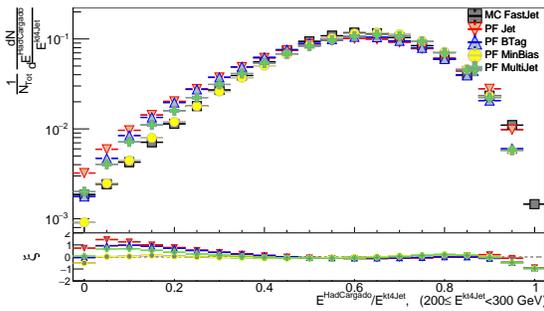
(CCXIV) Hadrones cargados ($30 \leq E^{kt4Jet} < 60$ GeV)



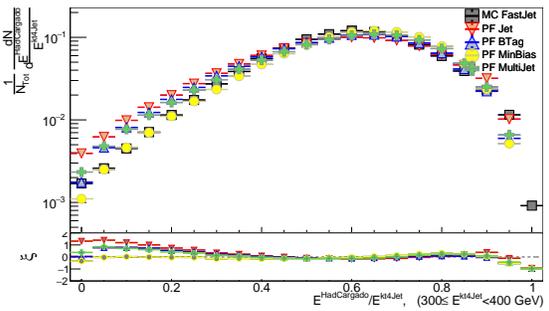
(CCXV) Hadrones cargados ($60 \leq E^{kt4Jet} < 100$ GeV)



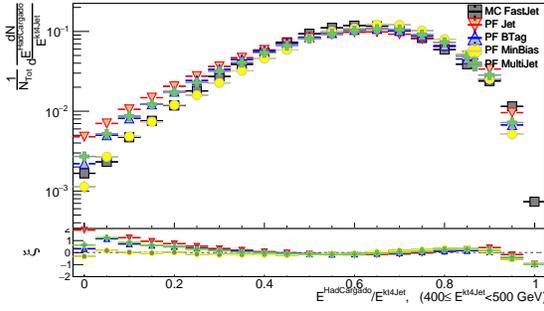
(CCXVI) Hadrones cargados ($100 \leq E^{kt4Jet} < 200$ GeV)



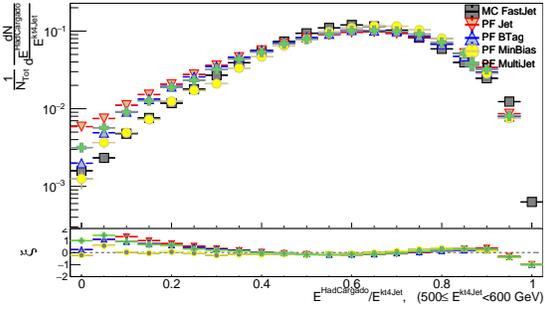
(CCXVII) Hadrones cargados ($200 \leq E^{kt4Jet} < 300$ GeV)



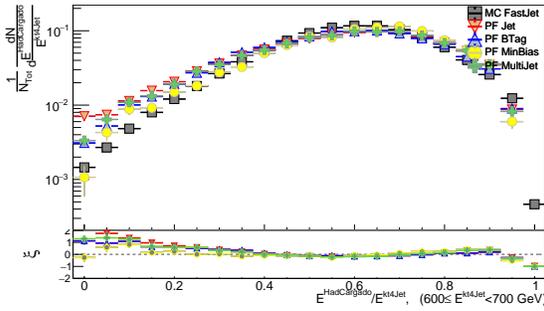
(CCXVIII) Hadrones cargados ($300 \leq E^{kt4Jet} < 400$ GeV)



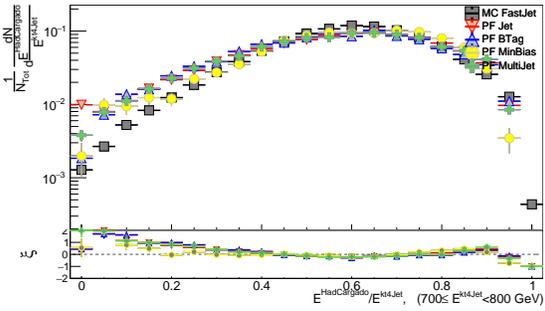
(CCXIX) Hadrones cargados ($400 \leq E^{kt4Jet} < 500$ GeV)



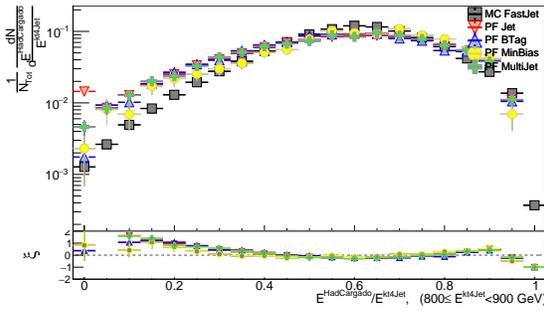
(CCXX) Hadrones cargados ($500 \leq E^{kt4Jet} < 600$ GeV)



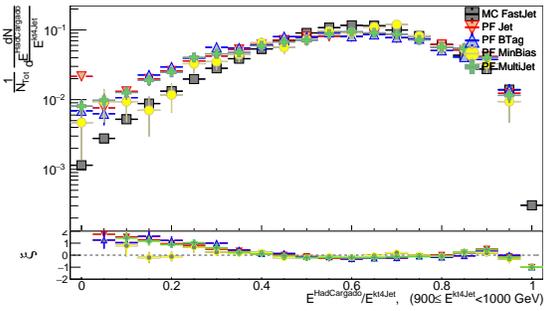
(CCXXI) Hadrones cargados ($600 \leq E^{kt4Jet} < 700$ GeV)



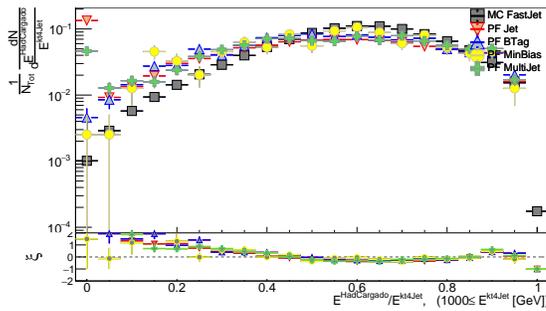
(CCXXII) Hadrones cargados ($700 \leq E^{kt4Jet} < 800$ GeV)



(CCXXIII) Hadrones cargados ($800 \leq E^{kt4Jet} < 900$ GeV)

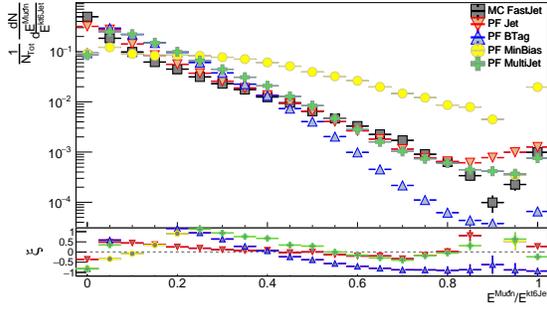


(CCXXIV) Hadrones cargados ($900 \leq E^{kt4Jet} < 1000$ GeV)

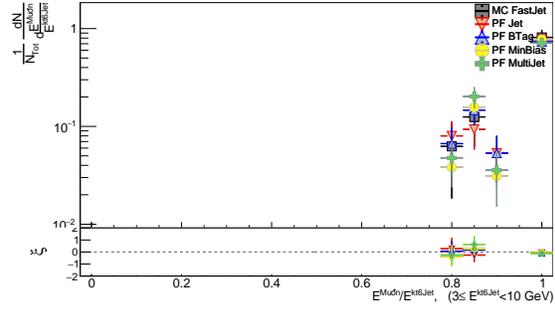


(CCXXV) Hadrones cargados ($1000 \leq E^{kt4Jet}$ [GeV])

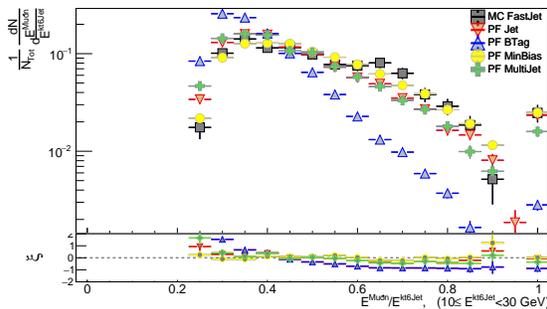
C.2.4. kt6Jets



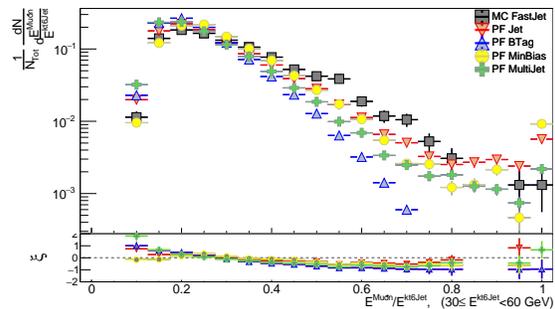
(CCXXVI) Muones



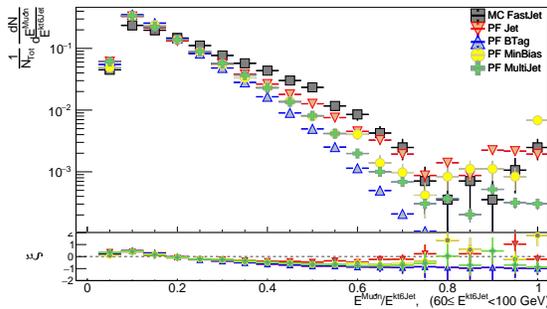
(CCXXVII) Muones ($3 \leq E^{kt6Jet} < 10$ GeV)



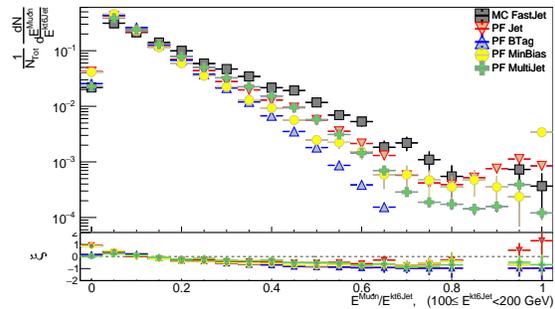
(CCXXVIII) Muones ($10 \leq E^{kt6Jet} < 30$ GeV)



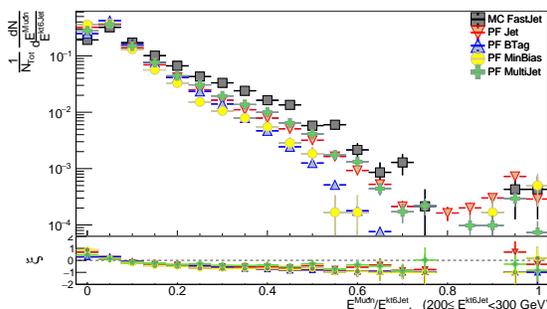
(CCXXIX) Muones ($30 \leq E^{kt6Jet} < 60$ GeV)



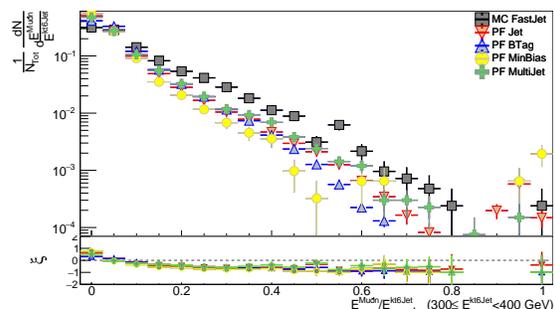
(CCXXX) Muones ($60 \leq E^{kt6Jet} < 100$ GeV)



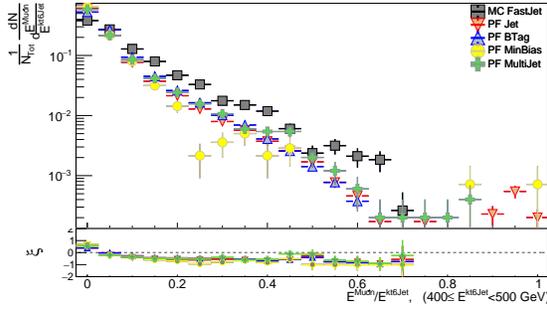
(CCXXXI) Muones ($100 \leq E^{kt6Jet} < 200$ GeV)



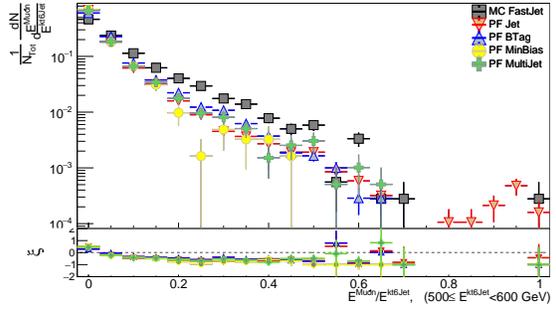
(CCXXXII) Muones ($200 \leq E^{kt6Jet} < 300$ GeV)



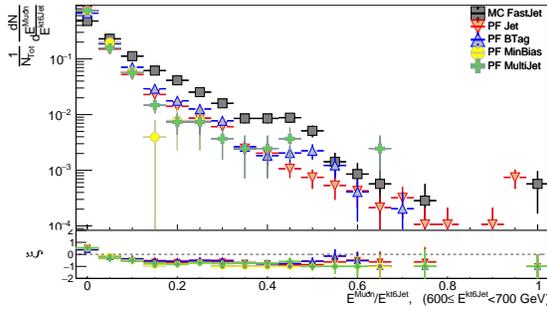
(CCXXXIII) Muones ($300 \leq E^{kt6Jet} < 400$ GeV)



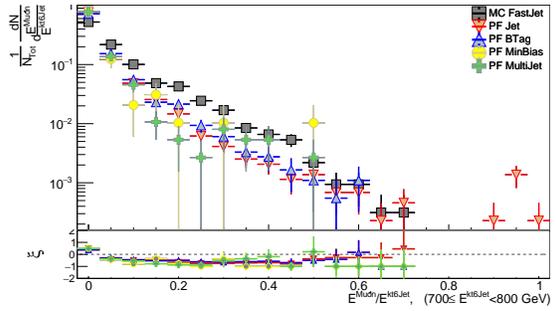
(CCXXXIV) Muones ($400 \leq E^{kt6Jet} < 500$ GeV)



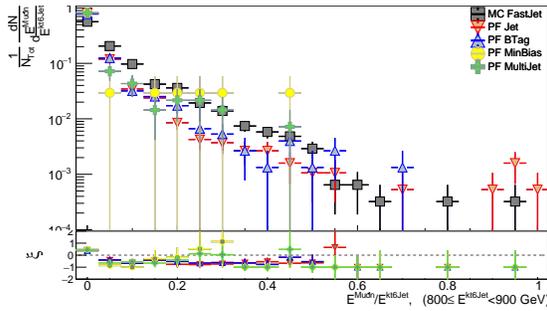
(CCXXXV) Muones ($500 \leq E^{kt6Jet} < 600$ GeV)



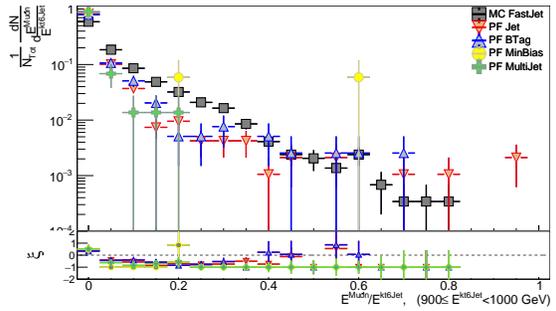
(CCXXXVI) Muones ($600 \leq E^{kt6Jet} < 700$ GeV)



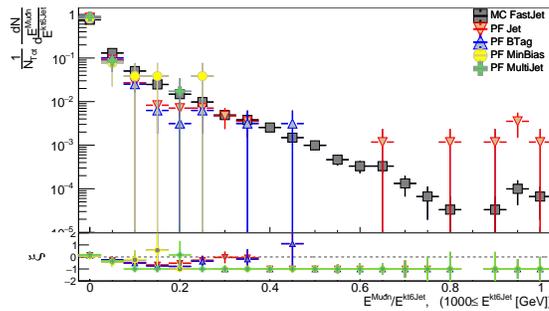
(CCXXXVII) Muones ($700 \leq E^{kt6Jet} < 800$ GeV)



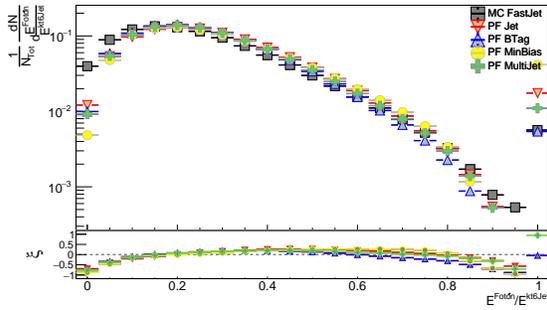
(CCXXXVIII) Muones ($800 \leq E^{kt6Jet} < 900$ GeV)



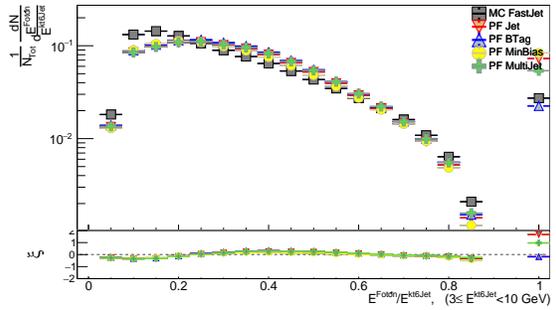
(CCXXXIX) Muones ($900 \leq E^{kt6Jet} < 1000$ GeV)



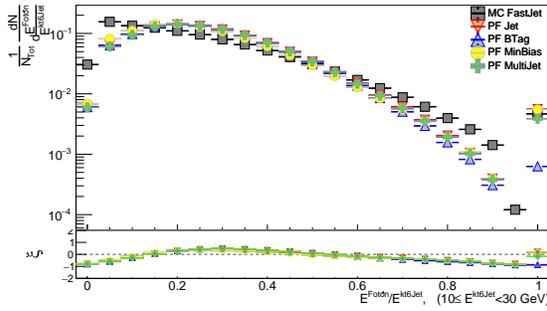
(CCXL) Muones ($1000 \leq E^{kt6Jet}$ [GeV])



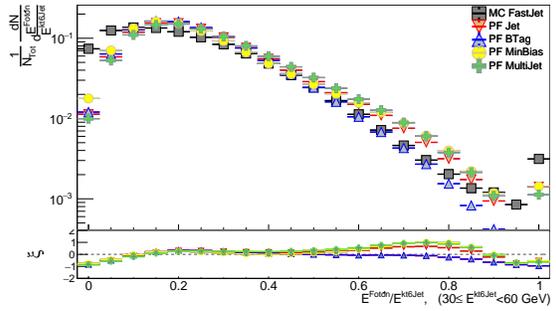
(CCXLI) Fotonos



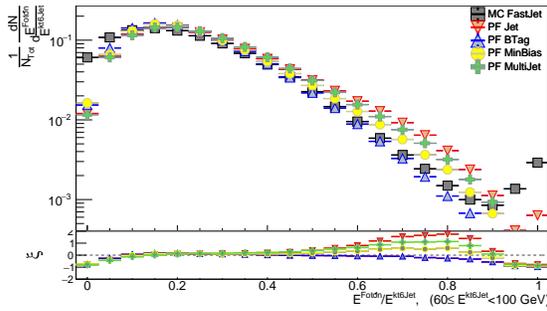
(CCXLII) Fotonos ($3 \leq E^{kt6Jet} < 10$ GeV)



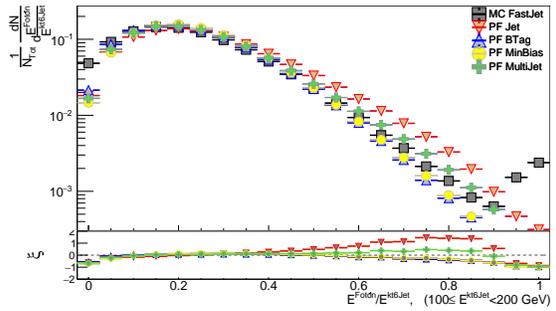
(CCXLIII) Fotonos ($10 \leq E^{kt6Jet} < 30$ GeV)



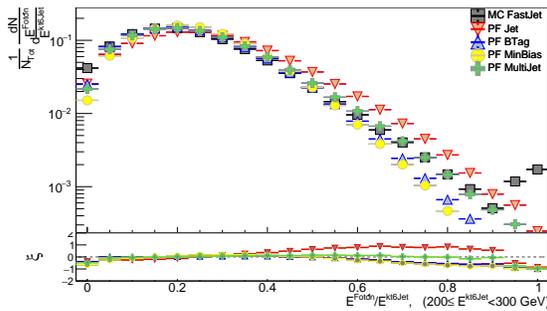
(CCXLIV) Fotonos ($30 \leq E^{kt6Jet} < 60$ GeV)



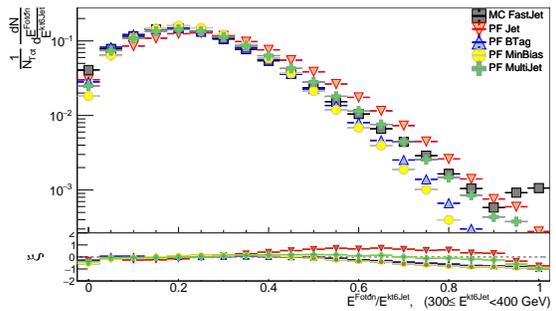
(CCXLV) Fotonos ($60 \leq E^{kt6Jet} < 100$ GeV)



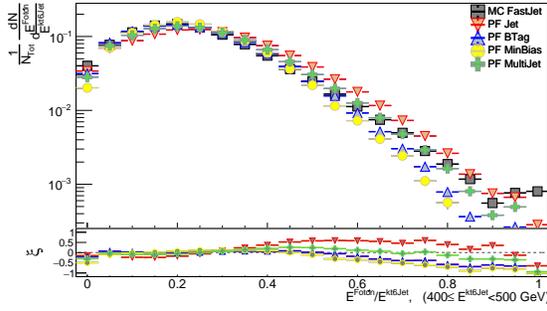
(CCXLVI) Fotonos ($100 \leq E^{kt6Jet} < 200$ GeV)



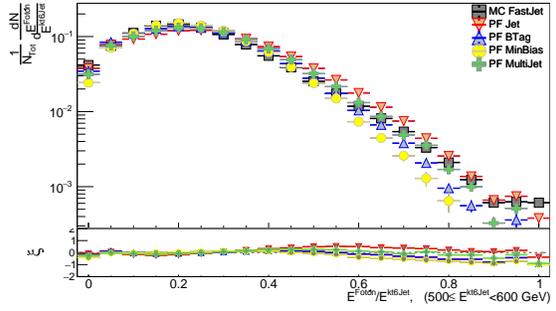
(CCXLVII) Fotonos ($200 \leq E^{kt6Jet} < 300$ GeV)



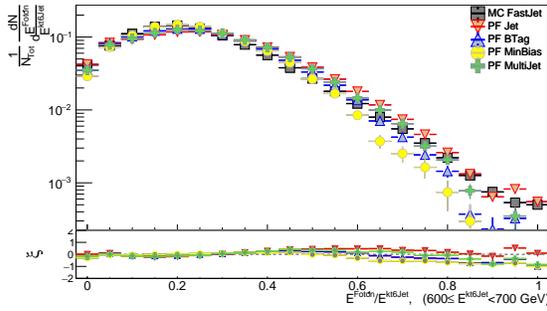
(CCXLVIII) Fotonos ($300 \leq E^{kt6Jet} < 400$ GeV)



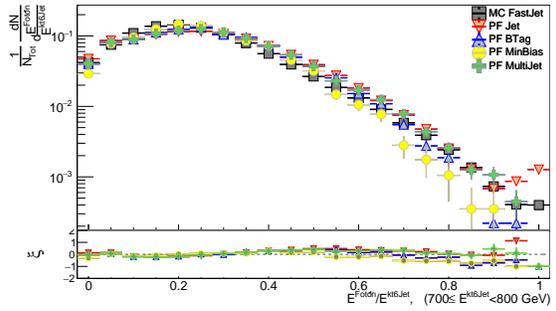
(CCXLIX) Fotonos ($400 \leq E^{kt6Jet} < 500$ GeV)



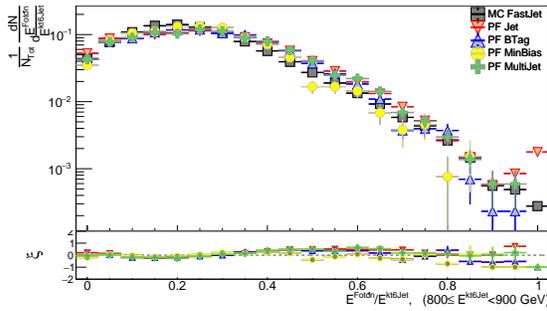
(CCL) Fotonos ($500 \leq E^{kt6Jet} < 600$ GeV)



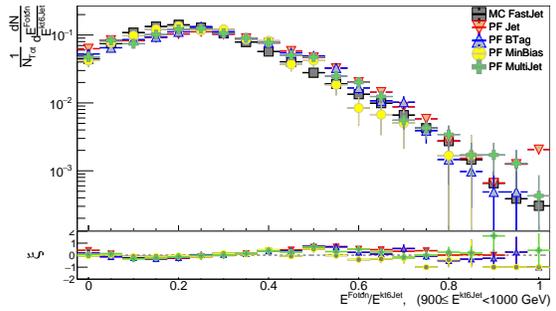
(CCLI) Fotonos ($600 \leq E^{kt6Jet} < 700$ GeV)



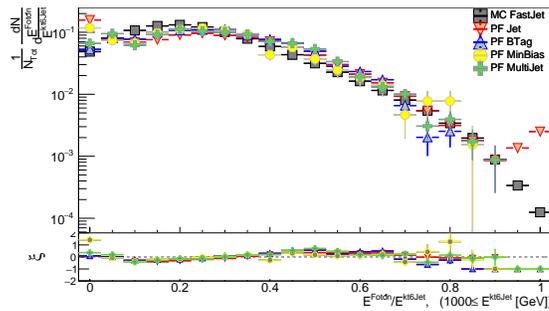
(CCLII) Fotonos ($700 \leq E^{kt6Jet} < 800$ GeV)



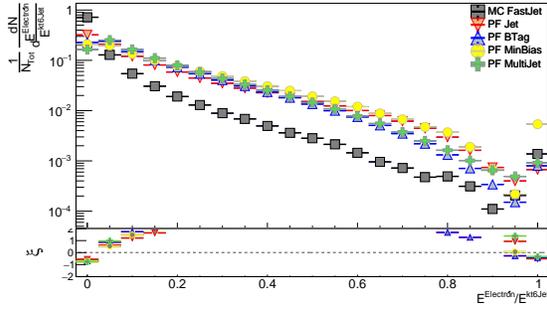
(CCLIII) Fotonos ($800 \leq E^{kt6Jet} < 900$ GeV)



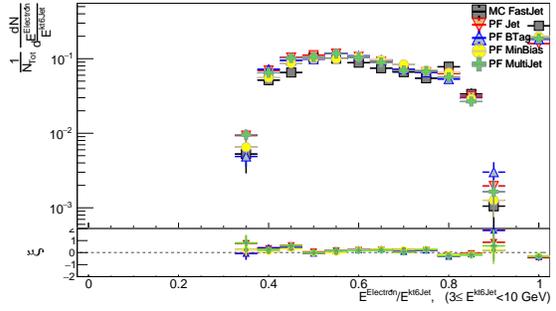
(CCLIV) Fotonos ($900 \leq E^{kt6Jet} < 1000$ GeV)



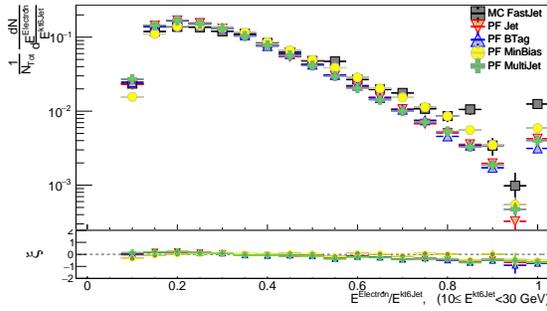
(CCLV) Fotonos ($1000 \leq E^{kt6Jet}$ [GeV])



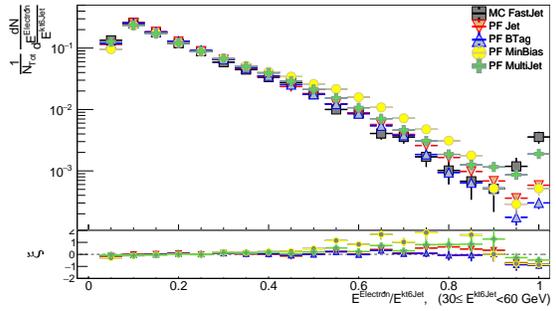
(CCLVI) Electrones



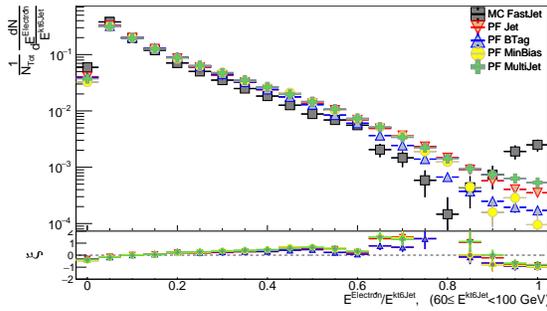
(CCLVII) Electrones ($3 \leq E^{kt6Jet} < 10$ GeV)



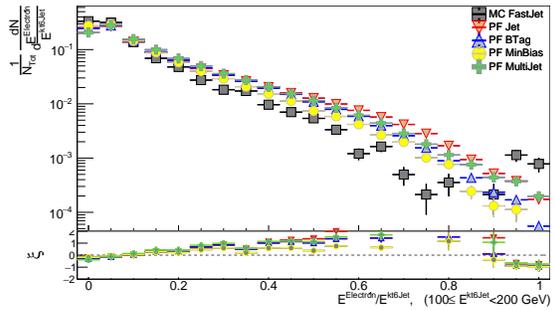
(CCLVIII) Electrones ($10 \leq E^{kt6Jet} < 30$ GeV)



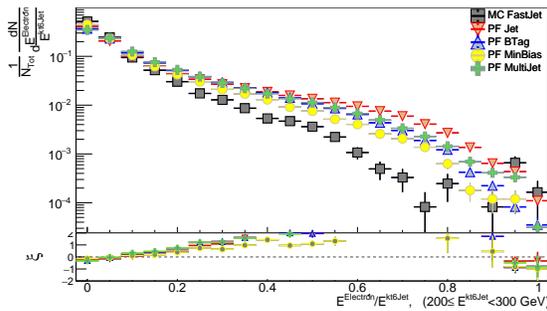
(CCLIX) Electrones ($30 \leq E^{kt6Jet} < 60$ GeV)



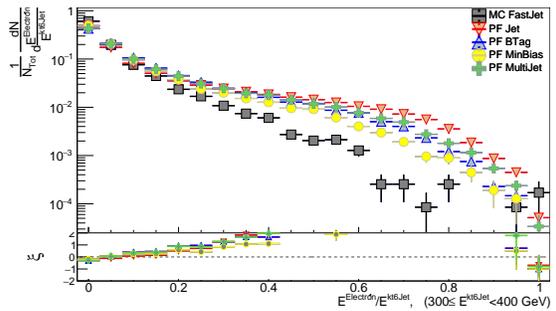
(CCLX) Electrones ($60 \leq E^{kt6Jet} < 100$ GeV)



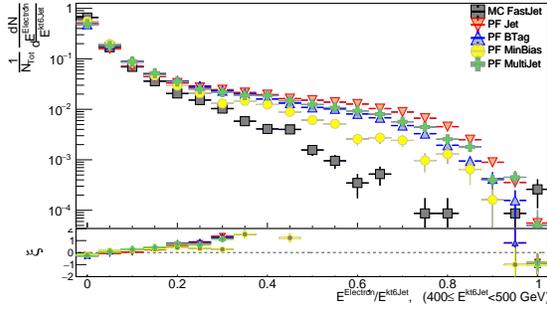
(CCLXI) Electrones ($100 \leq E^{kt6Jet} < 200$ GeV)



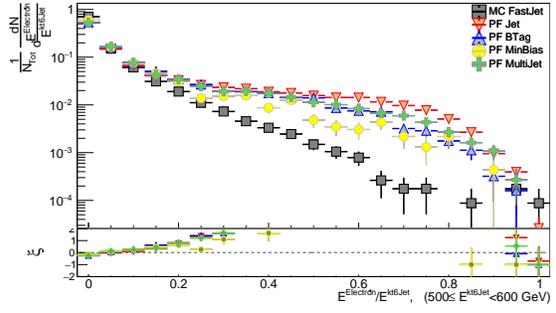
(CCLXII) Electrones ($200 \leq E^{kt6Jet} < 300$ GeV)



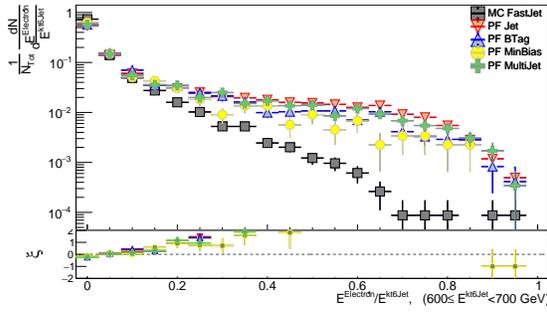
(CCLXIII) Electrones ($300 \leq E^{kt6Jet} < 400$ GeV)



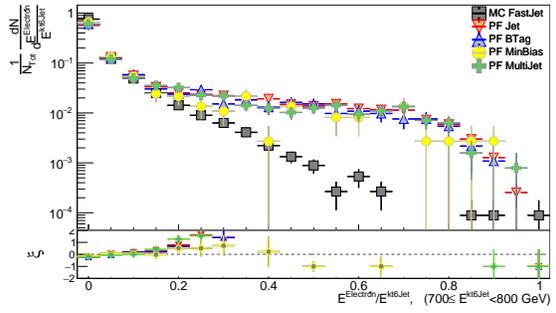
(CCLXIV) Electrones ($400 \leq E^{kt6Jet} < 500$ GeV)



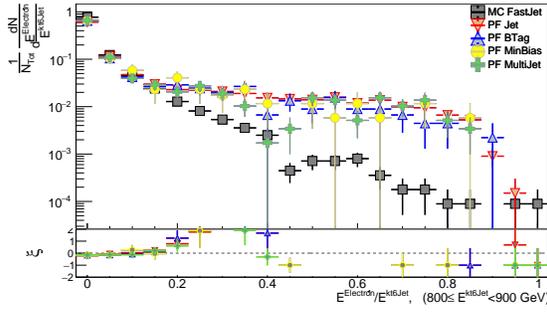
(CCLXV) Electrones ($500 \leq E^{kt6Jet} < 600$ GeV)



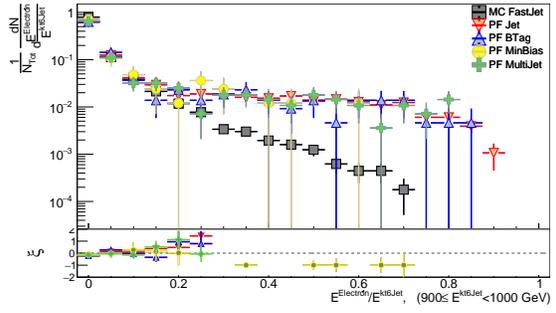
(CCLXVI) Electrones ($600 \leq E^{kt6Jet} < 700$ GeV)



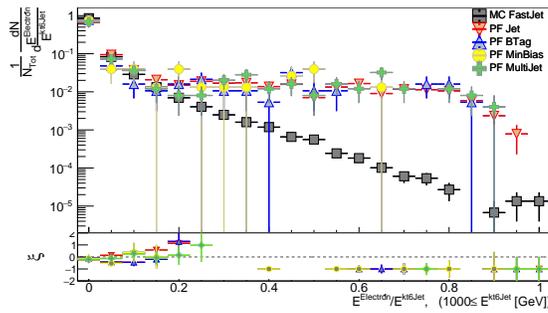
(CCLXVII) Electrones ($700 \leq E^{kt6Jet} < 800$ GeV)



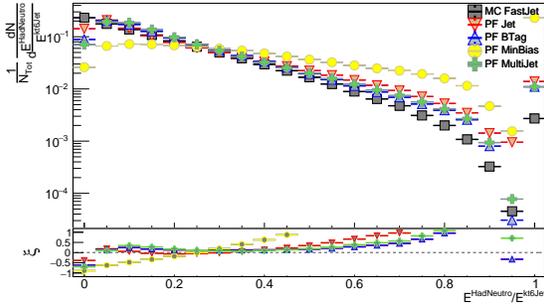
(CCLXVIII) Electrones ($800 \leq E^{kt6Jet} < 900$ GeV)



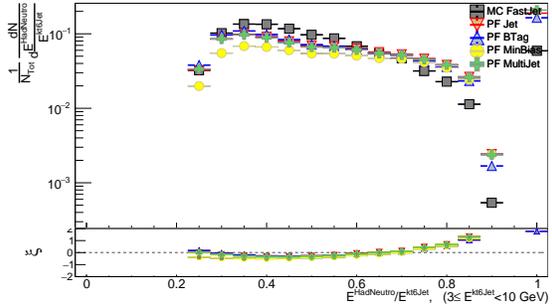
(CCLXIX) Electrones ($900 \leq E^{kt6Jet} < 1000$ GeV)



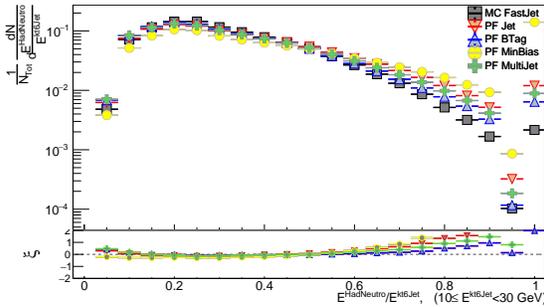
(CCLXX) Electrones ($1000 \leq E^{kt6Jet}$ [GeV])



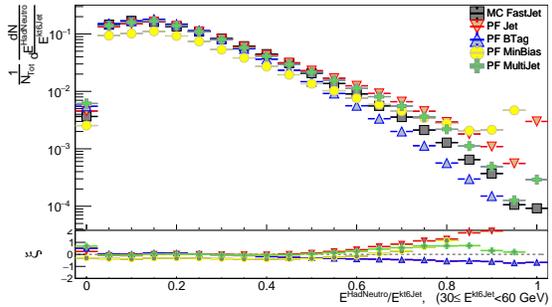
(CCLXXI) Hadrones neutros



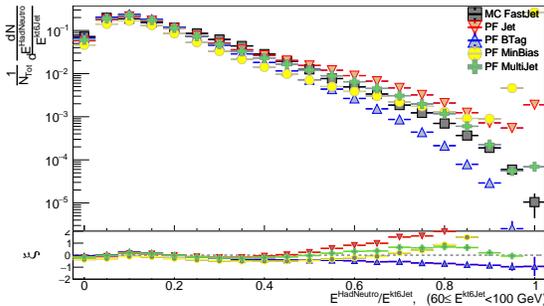
(CCLXXII) Hadrones neutros ($3 \leq E^{kt6,Jet} < 10$ GeV)



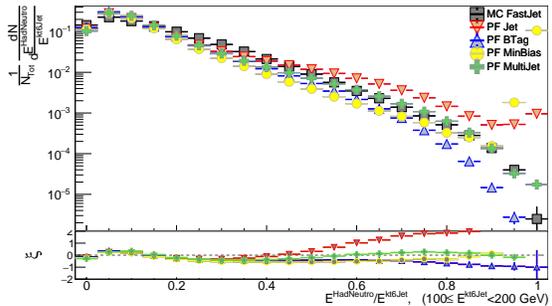
(CCLXXIII) Hadrones neutros ($10 \leq E^{kt6,Jet} < 30$ GeV)



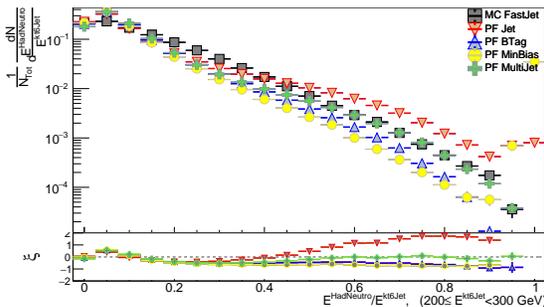
(CCLXXIV) Hadrones neutros ($30 \leq E^{kt6,Jet} < 60$ GeV)



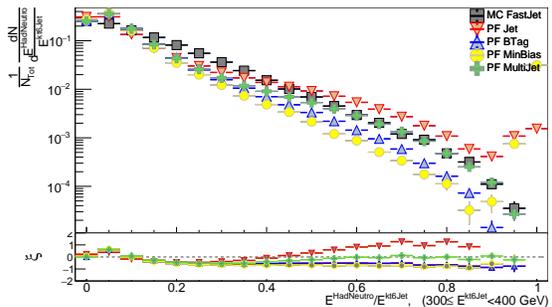
(CCLXXV) Hadrones neutros ($60 \leq E^{kt6,Jet} < 100$ GeV)



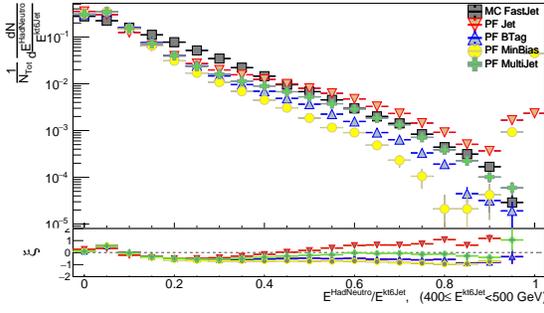
(CCLXXVI) Hadrones neutros ($100 \leq E^{kt6,Jet} < 200$ GeV)



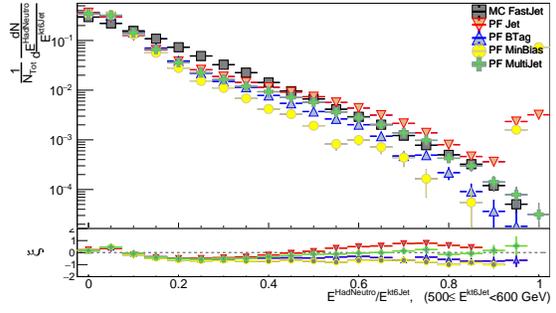
(CCLXXVII) Hadrones neutros ($200 \leq E^{kt6,Jet} < 300$ GeV)



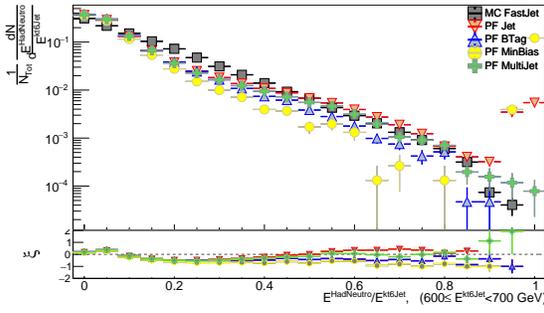
(CCLXXVIII) Hadrones neutros ($300 \leq E^{kt6,Jet} < 400$ GeV)



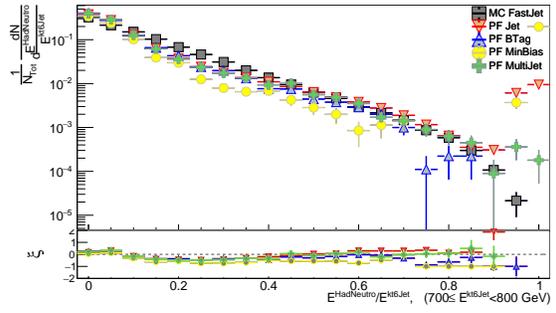
(CCLXXIX) Hadrones neutros ($400 \leq E^{kt6Jet} < 500$ GeV)



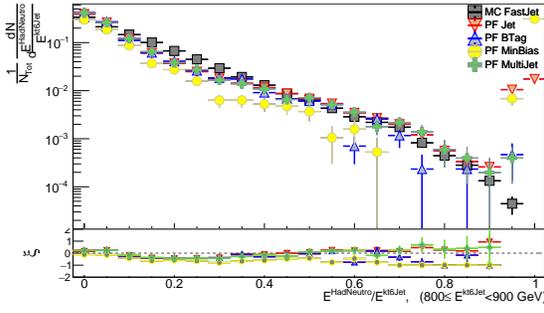
(CCLXXX) Hadrones neutros ($500 \leq E^{kt6Jet} < 600$ GeV)



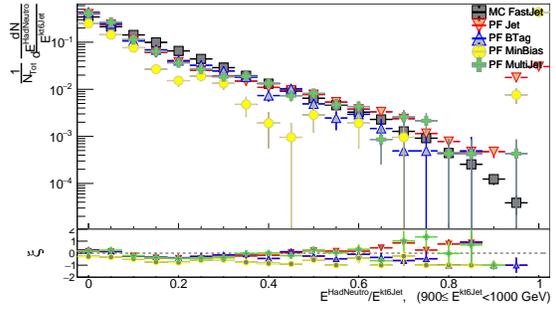
(CCLXXXI) Hadrones neutros ($600 \leq E^{kt6Jet} < 700$ GeV)



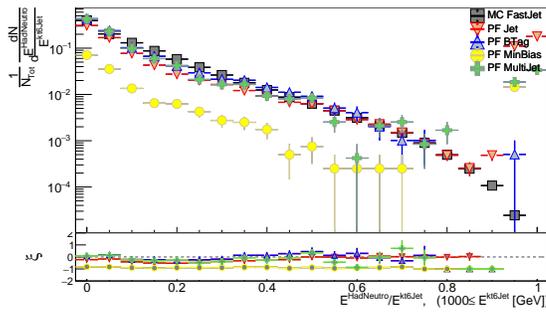
(CCLXXXII) Hadrones neutros ($700 \leq E^{kt6Jet} < 800$ GeV)



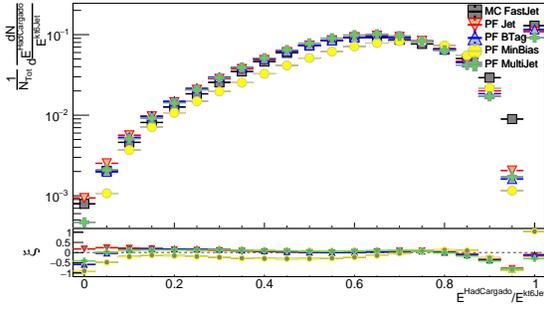
(CCLXXXIII) Hadrones neutros ($800 \leq E^{kt6Jet} < 900$ GeV)



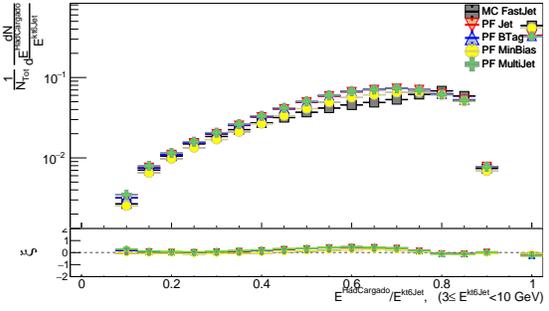
(CCLXXXIV) Hadrones neutros ($900 \leq E^{kt6Jet} < 1000$ GeV)



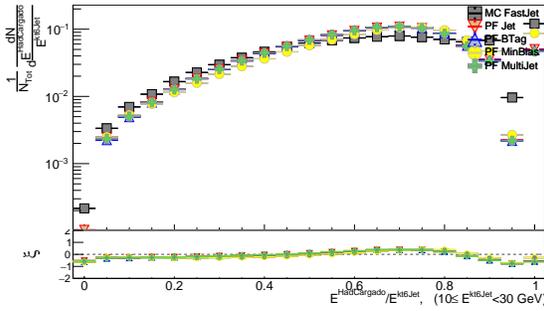
(CCLXXXV) Hadrones neutros ($1000 \leq E^{kt6Jet}$ [GeV])



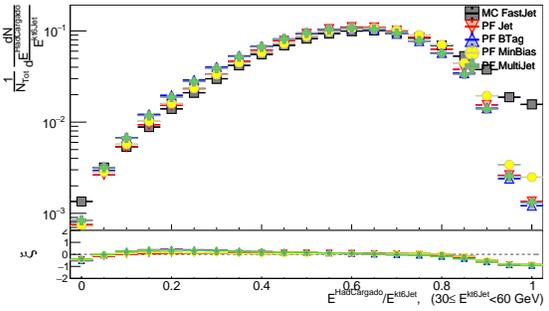
(CCLXXXVI) Hadrones cargados



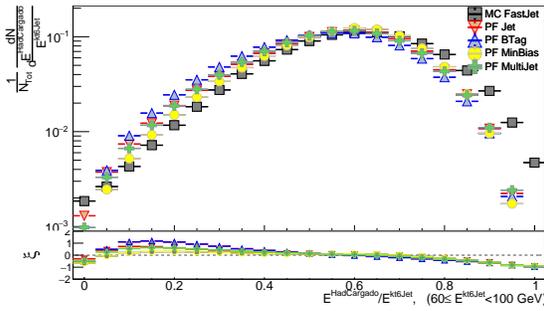
(CCLXXXVII) Hadrones cargados ($3 \leq E^{kt6Jet} < 10$ GeV)



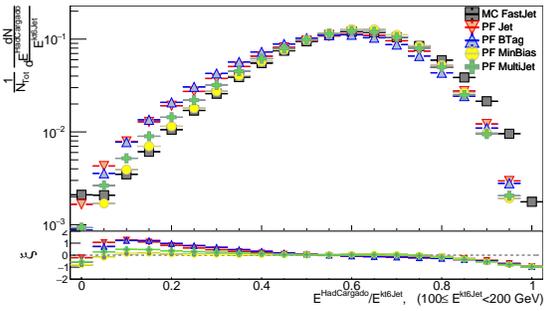
(CCLXXXVIII) Hadrones cargados ($10 \leq E^{kt6Jet} < 30$ GeV)



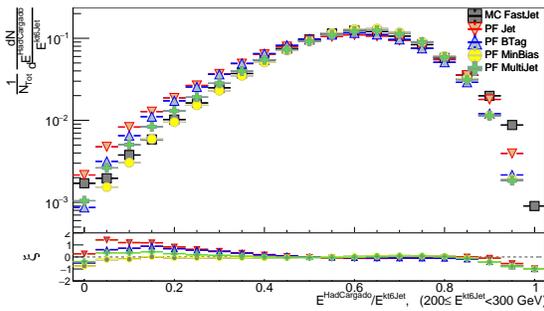
(CCLXXXIX) Hadrones cargados ($30 \leq E^{kt6Jet} < 60$ GeV)



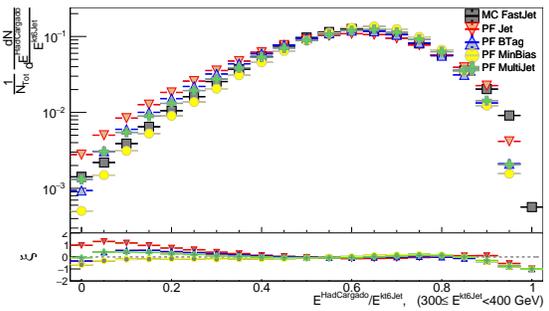
(CCXC) Hadrones cargados ($60 \leq E^{kt6Jet} < 100$ GeV)



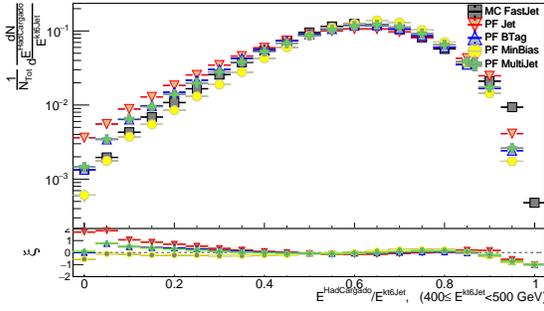
(CCXCI) Hadrones cargados ($100 \leq E^{kt6Jet} < 200$ GeV)



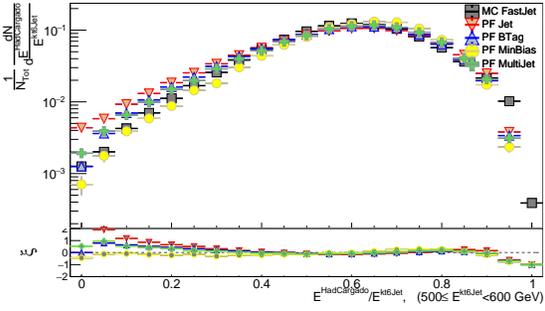
(CCXCII) Hadrones cargados ($200 \leq E^{kt6Jet} < 300$ GeV)



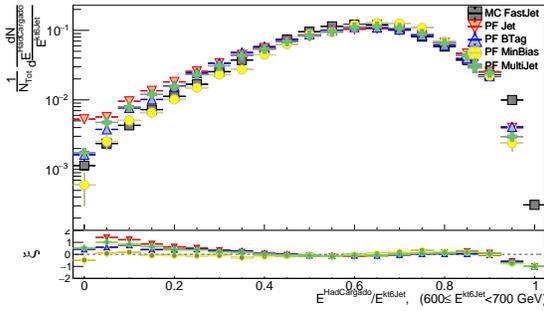
(CCXCIII) Hadrones cargados ($300 \leq E^{kt6Jet} < 400$ GeV)



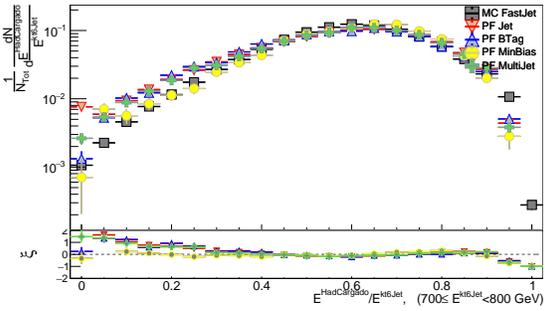
(CCXCIV) Hadrones cargados ($400 \leq E^{kt6Jet} < 500$ GeV)



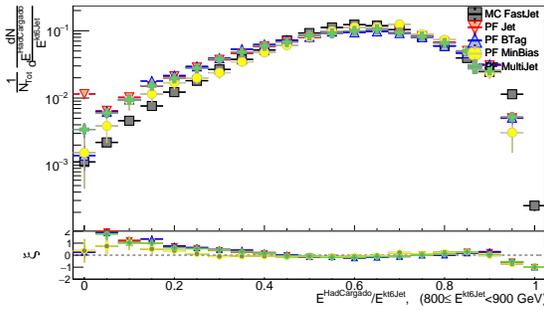
(CCXCV) Hadrones cargados ($500 \leq E^{kt6Jet} < 600$ GeV)



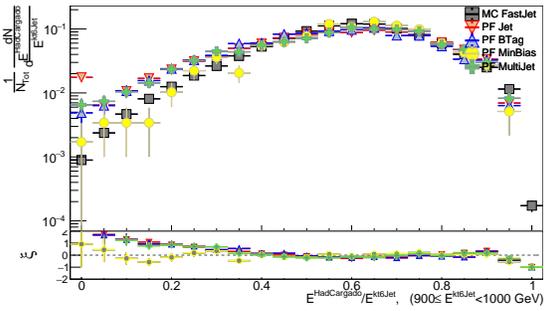
(CCXCVI) Hadrones cargados ($600 \leq E^{kt6Jet} < 700$ GeV)



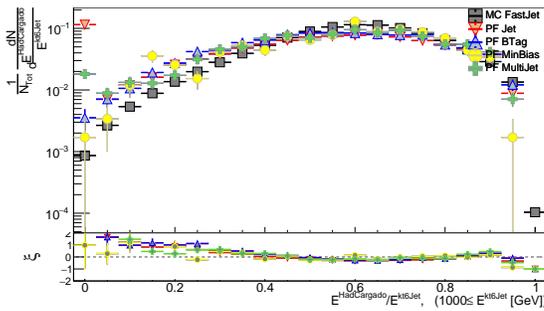
(CCXCVII) Hadrones cargados ($700 \leq E^{kt6Jet} < 800$ GeV)



(CCXCVIII) Hadrones cargados ($800 \leq E^{kt6Jet} < 900$ GeV)



(CCXCIX) Hadrones cargados ($900 \leq E^{kt6Jet} < 1000$ GeV)



(CCC) Hadrones cargados ($1000 \leq E^{kt6Jet}$ [GeV])

Bibliografía

- [1] Clémentine Broutin. *Electron Measurements and Search for Higgs Bosons in Multi-Lepton Channels with the CMS Experiment at LHC*. PhD thesis, Ecole polytechnique, 2011.
- [2] CMS collaboration (2016). *CMS Physics Objects 2011*. CERN Open Data Portal. <https://docs.google.com/document/d/11dP0zW0BelW3kNt6UTszetc8AoVMDGQfC5Cc9McSDQE/edit> [Consultado Enero 13, 2019].
- [3] CERN. *When protons collide*. http://atlas.physicsmasterclasses.org/en/zpath_protoncollisions.htm [Consultado Mayo 07, 2019].
- [4] University of Zurich. The lhcb experiment at uzh, standard model. <https://www.physik.uzh.ch/en/researcharea/lhcb/outreach/StandardModel.html> [Consultado Enero 17, 2020].
- [5] The CMS Experiment at CERNu (Marguerite Tonjes and Jeff Temple). *CMS made with Lego*. <https://cms.cern/news/cms-made-lego%C2%AE> [Consultado Enero 12, 2020].
- [6] CMS Collaboration. CMS: The hadron calorimeter technical design report. 1997.
- [7] David Barney and Tai Sakuma. Sketchup images highlighting the sub-detectors. Sep 2017.
- [8] Colaboración CMS. *CMS Open-Data Event Display*. <http://opendata.cern.ch/visualise/events/cms> [Consultado Enero 12, 2020].

- [9] Vardan Khachatryan et al. The CMS trigger system. *JINST*, 12(01):P01020, 2017.
- [10] Matteo Cacciari, Gavin P. Salam, and Gregory Soyez. The anti- k_t jet clustering algorithm. *JHEP*, 04:063, 2008.
- [11] M. Bachtis. *Heavy Neutral Particle Decays to Tau Pairs: Detected with CMS in Proton Collisions at $\sqrt{s} = 7\text{TeV}$* . Springer Theses. Springer International Publishing, 2013.
- [12] David J Griffiths. *Introduction to elementary particles; 2nd rev. Edition*. Physics textbook. Wiley, New York, NY, 2008.
- [13] Arthur Beiser. *Concepts of Modern Physics*. McGraw Hill, 6ta edition, 2003.
- [14] Ahmed Ali and Gustav Kramer. *Jets and QCD: A Historical Review of the Discovery of the Quark and Gluon Jets and its Impact on QCD*. *Eur. Phys. J.*, H36:245–326, 2011.
- [15] CERN. *The history of CERN*. <https://home.cern/about/who-we-are/our-history> [Consultado Enero 19, 2020].
- [16] CERN. *What is CERN's mission?* <https://home.cern/about/who-we-are/our-mission> [Consultado Enero 19, 2020].
- [17] Oliver Sim Brüning, Paul Collier, P Lebrun, Stephen Myers, Ranko Ostojic, John Poole, and Paul Proudlock. *LHC Design Report*. CERN Yellow Reports: Monographs. CERN, Geneva, 2004.
- [18] CERN. *What is CMS?* <http://cms.web.cern.ch/news/what-cms> [Consultado Junio 12, 2019].
- [19] CERN. *CERN makes public first data of LHC experiment*. <https://home.cern/news/news/accelerators/cern-makes-public-first-data-lhc-experiments> [Consultado Agosto 12, 2019].

- [20] CERN. *CERN Open Data Portal*. <http://opendata.cern.ch/docs/about> [Consultado Julio 19, 2019].
- [21] CMS Collaboration. *ECAL Technical Design Report*. CERN/LHCC, Enero 2006.
- [22] Bayatian G L, Sergey Chatrchyan, Gevorg Hmayakyan, Albert Sirunyan, Adam W, Thomas Bergauer, Marko Dragicevic, Ero J, Friedl M, Rudolf Frühwirth, Ghete V, Glaser P, Hrubec J, M Jeitler, Krammer M, Ildefons Magrans, Mikulec I, Mitaroff W, Noebauer T, and Bekhzod Yuldashev. *CMS Physics : Technical Design Report Volume 1: Detector Performance and Software*. Geneva: CERN, Enero 2006.
- [23] Cheuk-Yin Wong. *Introduction to high-energy heavy ion collisions*. World Scientific, 1994.
- [24] V. Veszpremi. *Operation and performance of the CMS tracker*. *JINST*, 9:C03005, 2014.
- [25] David Gruft Sheffield. *Search for low-mass and high-mass narrow dijet resonances with the CMS detector at $\sqrt{s} = 13$ TeV*. PhD thesis, Rutgers, The State University of New Jersey, 2017.
- [26] Serguei Chatrchyan et al. *The Performance of the CMS Muon Detector in Proton-Proton Collisions at $\sqrt{s} = 7$ TeV at the LHC*. *JINST*, 8:P11002, 2013.
- [27] Vardan Khachatryan et al. Jet energy scale and resolution in the CMS experiment in pp collisions at 8 TeV. *JINST*, 12(02):P02014, 2017.
- [28] Oleg Kuprash. *Triggers at ATLAS and CMS*, 2018. https://indico.cern.ch/event/681549/contributions/2930934/attachments/1664379/2667597/2018_06_07_LHCP18_Trigger_in_ATLAS_and_CMS.pdf [Consultado Enero 15, 2020].
- [29] CERN. The CMS Offline WorkBook. In *The CMS Offline SW Guide*, chapter 2.3 CMSSW Application Framework. CERN, Julio 2018. <https://twiki.cern.ch/>

- twiki/bin/view/CMSPublic/WorkBookCMSSWFramework [Consultado Julio 04, 2019].
- [30] P. Bortignon. *Description of the CMS Trigger Design and Performance*, 2018. https://indico.cern.ch/event/659612/contributions/2836315/attachments/1593071/2521964/201802_Bortignon_TDII_PueblaMexico_31Jan_2.pdf [Consultado Diciembre 15, 2019].
- [31] T. Bergauer. *Design, Construction and Commissioning of the CMS Tracker at CERN and Proposed Improvements for Detectors at the Future International Linear Collider*. PhD thesis, Vienna, Tech. U., 2008.
- [32] CERN. The CMS Offline WorkBook. In *The CMS Offline SW Guide*, chapter 2.2 Computing Model. CERN, Julio 2018. <https://twiki.cern.ch/twiki/bin/view/CMSPublic/WorkBookComputingModel> [Consultado Junio 01, 2019].
- [33] CERN. The CMS Offline WorkBook. In *The CMS Offline SW Guide*, chapter 3.1 Analysis Overview: an Introduction. CERN, Julio 2018. <https://twiki.cern.ch/twiki/bin/view/CMSPublic/WorkBookAnalysisOverviewIntroduction> [Consultado Julio 03, 2019].
- [34] CERN. *Getting Started with CMS 2011 Open Data*. <http://opendata.cern.ch/docs/cms-getting-started-2011> [Consultado Enero 08, 2019].
- [35] CERN. *ROOT User's Guide*, Mayo 2018. <https://root.cern.ch/root/html/doc/guides/users-guide/ROOTUsersGuideA4.pdf> [Consultado Junio 14, 2019].
- [36] Serguei Chatrchyan et al. Identification of b-Quark Jets with the CMS Experiment. *JINST*, 8:P04013, 2013.
- [37] Yuan Chao. Minimum-Bias and Underlying Event Studies at CMS. In *Proceedings, 28th International Conference on Physics in Collision (PIC 2008): Perugia, Italy, June 25-28, 2008*, 2008.

- [38] Michael Leyton. Minimum Bias and Underlying Event Measurements with ATLAS. In *Proceedings, 3rd International Workshop on Multiple Partonic Interactions at the LHC (MPI@LHC 2011): Hamburg, Germany, 21-25 Nov 2011*, pages 11–19, 2012.
- [39] CMS collaboration (2016). *Jet primary dataset in AOD format from RunA of 2011 (/Jet/Run2011A-12Oct2013-v1/AOD)*. CERN Open Data Portal. <http://opendata.cern.ch/record/21> [Consultado Enero 17, 2019].
- [40] CMS collaboration (2016). *BTag primary dataset in AOD format from RunA of 2011 (/BTag/Run2011A-12Oct2013-v1/AOD)*. CERN Open Data Portal. <http://opendata.cern.ch/record/15> [Consultado Enero 17, 2019].
- [41] CMS collaboration (2016). *MinimumBias primary dataset in AOD format from RunA of 2011 (/MinimumBias/Run2011A-12Oct2013-v1/AOD)*. CERN Open Data Portal. <http://opendata.cern.ch/record/24> [Consultado Enero 17, 2019].
- [42] CMS collaboration (2016). *MultiJet primary dataset in AOD format from RunA of 2011 (/MultiJet/Run2011A-12Oct2013-v1/AOD)*. CERN Open Data Portal. <http://opendata.cern.ch/record/28> [Consultado Enero 17, 2019].
- [43] Colaboración CMS. *CMS 2011 Virtual Machines: How to install*. <http://opendata.cern.ch/docs/cms-virtual-machine-2011> [Consultado Enero 11, 2019].
- [44] Colaboración VirtualBox. *VirtualBox User Manual's (Version 6.0.10)*. <https://www.virtualbox.org/manual/UserManual.html> [Consultado Agosto 15, 2019].
- [45] A. M. Sirunyan et al. *Particle-flow reconstruction and global event description with the CMS detector*. *JINST*, 12(10):P10003, 2017.
- [46] CERN. Particle Flow. In *The CMS Offline SW Guide*. CERN, Noviembre 2018. <https://twiki.cern.ch/twiki/bin/view/CMSPublic/SWGuideParticleFlow#Jets> [Consultado Junio 10, 2019].

- [47] V. Bertone, N. P. Hartland, E. R. Nocera, J. Rojo, and L. Rottoli. *Charged hadron fragmentation functions from collider data*. *Eur. Phys. J.*, C78(8):651, 2018.
- [48] Andreas Metz and Anselm Vossen. *Parton Fragmentation Functions*. *Prog. Part. Nucl. Phys.*, 91:136–202, 2016.
- [49] Matteo Cacciari, Gavin P. Salam, and Gregory Soyez. *FastJet User Manual (for version 3.3.2)*. page 80. <http://fastjet.fr/repo/fastjet-doc-3.3.2.pdf>.
- [50] Jeff Tseng and Hannah Evans. *Sequential recombination algorithm for jet clustering and background subtraction*. *Phys. Rev.*, D88:014044, 2013.
- [51] The DELPHI Collaboration. *Measurement of the gluon fragmentation function and a comparison of the scaling violation in gluon and quark jets*. *Eur. Phys. J., C - Particles and Fields*, 13, Abril 2000.
- [52] B. Andersson, G. Gustafson, G. Ingelman, and T. Sjöstrand. *Parton fragmentation and string dynamics*. *Physics Reports*, 97(2):31 – 145, 1983.
- [53] F. Halzen and A.D. Martin. *Quarks and Leptons: An Introductory Course in Modern Particle Physics*. Wiley India Pvt. Limited, 2008.
- [54] Ryan Atkin. Review of jet reconstruction algorithms. *Journal of Physics: Conference Series*, 645, Octubre 2015.
- [55] CERN. The CMS Offline WorkBook. In *The CMS Offline SW Guide*, chapter 7.2 Jet Analysis . CERN, Julio 2009. <https://twiki.cern.ch/twiki/bin/view/CMSPublic/WorkBookJetAnalysis> [Consultado Junio 10, 2019].
- [56] Tomasz Wlodek. Monte Carlo methods in HEP. In -. University of Texas, -.
- [57] Christian Bierlich. *Rope Hadronization, Geometry and Particle Production in pp and pA Collisions*. PhD thesis, Lund University, 2016.

- [58] Torbjörn Sjöstrand, Stefan Ask, Jesper R. Christiansen, Richard Corke, Nishita Desai, Philip Ilten, Stephen Mrenna, Stefan Prestel, Christine O. Rasmussen, and Peter Z. Skands. *An Introduction to PYTHIA 8.2. Comput. Phys. Commun.*, 191:159–177, 2015.
- [59] Torbjörn Sjöstrand and Peter Skands and Stefan Prestel. *PYTHIA 8 Worksheet*. Colaboración PYTHIA, Enero 2014.
- [60] Colaboración PYTHIA. *PYTHIA 8 (Online Manual)*. <http://home.thep.lu.se/~torbjorn/pythia81php/Welcome.php> [Consultado Abril 19, 2019].
- [61] Matteo Cacciari and Gavin P. Salam. Dispelling the N^3 myth for the k_t jet-finder. *Phys. Lett.*, B641:57–61, 2006.
- [62] Matteo Cacciari and Gavin P. Salam. Pileup subtraction using jet areas. *Physics Letters B*, 659(1):119 – 126, 2008.
- [63] Gavin Salam and Matteo Cacciari. Jet clustering in particle physics, via a dynamic nearest neighbour graph implemented with CGAL. 04 2006.
- [64] Rene Brun and Fons Rademakers. *ROOT - An Object Oriented Data Analysis Framework*. <http://root.cern.ch/> [Consultado Mayo 27, 2019].
- [65] Rene Brun and Fons Rademakers. Root - an object oriented data analysis framework. In *AIHENP'96 Workshop, Lausanne*, volume 389, pages 81–86, 1996.
- [66] Achim Geiser, Irene Dutta, Harri Hirvonsalo, and Bridget Sheeran. *Example code to produce the di-muon spectrum from a CMS 2011 or 2012 primary dataset*, 2017. <http://opendata.cern.ch/record/5001> [Consultado Enero 13, 2019].
- [67] Colaboración CMS. *JetSpecific.cc*. Lineas 37, 247-249 y 321-333. Código público, ver. cfbdc8b, <https://github.com/cms-sw/cmssw/blob/master/RecoJets/JetProducers/src/JetSpecific.cc> [Consultado Mayo 13, 2019].

- [68] Colaboración CMS. *PFJet.h*. Lineas 65-71, 179 y 180. Código público, ver. 1bafe59, https://github.com/cms-sw/cmssw/blob/CMSSW_5_3_X/DataFormats/JetReco/interface/PFJet.h#L111 [Consultado Mayo 13, 2019].
- [69] L. Garren, F. Krauss, C. J. Lin, Sergio Navas, P. Richardson, and T. Sjöstrand. *Monte Carlo Particle Numbering Scheme*. -, Octubre 2017.
- [70] Jennifer Kile and Julian von Wimmersperg-Toeller. Monte Carlo Tuning for $e^+e^- \rightarrow$ Hadrons and Comparison with Unfolded LEP Data. 2017.
- [71] Patrick T. Komiske, Radha Mastandrea, Eric M. Metodiev, Preksha Naik, and Jesse Thaler. Exploring the Space of Jets with CMS Open Data. *Phys. Rev.*, D101(3):034009, 2020.
- [72] Volker Blobel. *Unfolding Methods in Particle Physics*. <https://indico.cern.ch/event/107747/contributions/32645/attachments/24317/35000/blobel.pdf> [Consultado Febrero 29, 2020].
- [73] CERN. Track Reconstruction Offline Guide. In *The CMS Offline SW Guide*, chapter 2 Global Reconstruction. CERN, Julio 2009. <https://twiki.cern.ch/twiki/bin/view/CMSPublic/SWGuideGsFitter> [Consultado Junio 10, 2019].
- [74] Torbjorn Sjostrand, Stephen Mrenna, and Peter Z. Skands. A Brief Introduction to PYTHIA 8.1. *Comput. Phys. Commun.*, 178:852–867, 2008.
- [75] Colaboración FastJet. *09-user_info.cc*. Lineas 65-95. Código público, http://fastjet.fr/repo/doxygen-3.0beta1/09-user__info_8cc_source.html [Consultado Mayo 17, 2019].
- [76] Colaboración FastJet. *const L& fastjet::PseudoJet::user_info () const [inline]*. Código público, http://fastjet.fr/repo/doxygen-3.0.0/classfastjet_1_1PseudoJet.html#ab2fd8ad20a61b97deebbbc85187020de [Consultado Mayo 17, 2019].