

¿Qué hace el programa `higgs4mu_select` al ejecutarse?

Física + Programación, paso a paso

Aldo

29 de septiembre de 2025

Objetivo y panorama general

Objetivo del ejecutable

- Seleccionar eventos con 4 muones (4μ) compatibles con $H \rightarrow ZZ^* \rightarrow 4\mu$.
- Aplicar preselecciones por muón: p_T , $|\eta|$, ID, aislamiento (ISO), IP ($|d_{xy}|$, $|dz|$), SIP3D.
- Construir y elegir el **pareo OS-OS** óptimo \Rightarrow formar $Z_a \approx M_Z$ y Z_b (off-shell).
- Aplicar cortes finales (20/10 en Z_a ; opcional 20/10/7/5 en 4μ ; ventanas de masa).
- Guardar histogramas y un TTree con variables clave del sistema 4μ .

Entradas \rightarrow Salidas

- *Entrada*: ROOT con TTree (p.ej. analisis_final_clasificados.root:Events).
- *Salida*: ROOT con histogramas + FourMuTree (p.ej. out_zz4mu.root).

Cómo se ejecuta y cómo lee la configuración

Ejemplo de ejecución

```
g++ -std=c++17 higgs4mu_select.cpp -o higgs4mu_select \  
$(root-config --cflags --libs) -O2 -DCOMPPILE_STANDALONE
```

```
./higgs4mu_select analisis_final_clasificados.root Events out_zz4mu.root \  
--comb-search \  
--pt-lead=20 --pt-sub=10 \  
--mza-min=50 --mza-max=106 --mzb-min=12 --mzb-max=120
```

Parsing de argumentos (programación)

- `main()` fija por defecto: archivo de entrada, TTree y archivo de salida.
- `parse_cli()` interpreta flags numéricos y booleanos (`-use-iso`, `-comb-search`, etc.).
- Se imprime un **resumen de configuración** para reproducibilidad.

Razonamiento físico

- Los cortes y ventanas controlan balance señal/ruido y la región cinemática de interés para $H \rightarrow ZZ^*$.
- Se prioriza que Z_a esté cerca de $M_Z \Rightarrow$ se minimiza $|m_{Z_a} - M_Z|$.

Apertura de ROOT y activación de ramas

Programación

- `TFile fin(..., "READ")` y `TTree* tr = fin.Get(treeName)`.
- `tr->SetBranchStatus(*,0)` y luego se activan **sólo** las ramas necesarias (performance).
- **Obligatorias:** `numbermuon`, `muon_pt/eta/phi`, `muon_ch`, `muon_isLoose/Medium/Tight`.
- **Opcionales (IP/SIP):** `muon_dxy`, `muon_dz`, `muon_sip3d` (conexión *condicional*).
- **ISO:** si `-use-iso`, conectar `muon_pfireliso04DBCorr` u otra vía `-iso-branch=`.

Física

- Las ramas contienen la kinemática, carga, calidad e información de aislamiento/impacto para un 4μ "limpio".

Programación

- Cutflow global `h_cutflow` con 11 pasos.
- Pares: `h_mZij_best` (siempre) y `h_mZij_pre` (diagnóstico opcional).
- Finales: `h_mZa`, `h_mZb`, `h_m4mu`.
- Por categorías: AllTight/AllMedium/AllLoose y por composición $T\{nT\}M\{nM\}L\{nL\}$.
- TTree FourMuTree: masas, 4-vector del sistema, categoría, niveles por muón e índices de origen.

Física

- `h_mZa` muestra un pico en Z ; `h_m4mu` puede insinuar H cerca de 125 GeV.
- El TTree permite rehacer cortes y estudios de sistema.

Preselección por etapas (pipeline instrumentado)

Etapas

- 1 **PT**: $p_T > \text{PT_MIN_BASE}$.
- 2 $|\eta|$: si APPLY_ETA, exigir $|\eta| < \text{ETA_MAX}$.
- 3 **ID** \geq **Loose**: si APPLY_ID.
- 4 **ISO**: si USE_ISO y la rama existe, $\text{relISO} \leq \text{ISO_MAX}$.
- 5 **IP (dxy, dz)**: si APPLY_IP y existen ramas, $|dxy| < \text{DXY_MAX}$, $|dz| < \text{DZ_MAX}$.
- 6 **SIP3D**: si APPLY_SIP y existe, $|\text{SIP3D}| < \text{SIP3D_MAX}$.

Programación

- Tubería de índices: $\text{idx_all} \rightarrow \text{idx_pt} \rightarrow \text{idx_eta} \rightarrow \text{idx_id} \rightarrow \text{idx_iso} \rightarrow \text{idx_ip} \rightarrow \text{idx_sip}$.
- En cada etapa, histogramas diagnósticos “after X” y un cutflow por etapas (h_evt_pre_cf).

Física

- PT/ η : aceptación y calidad de reconstrucción.
- ID/ISO: pureza de muones prompt; IP/SIP: supresión de no-prompt o mal medidos.

Construcción del cuádruple y pareo OS-OS

Programación

- Con muones que sobreviven todas las etapas:
 - *Top4*: si `USE_COMB_SEARCH=false`, ordena por p_T y toma los 4 mayores.
 - *Combinatorio*: si `USE_COMB_SEARCH=true`, prueba todas las combinaciones de 4 muones.
- Para cada cuádruple, evaluar combinaciones OS-OS: (12+34), (13+24), (14+23).
- Criterio lexicográfico de óptimo:
 - 1 Minimizar $|m_{Za} - M_Z|$.
 - 2 Minimizar $|m_{Zb} - M_Z|$.
 - 3 Maximizar $p_T(Z_a) + p_T(Z_b)$.

Física

- En $H \rightarrow ZZ^*$ se espera un bosón Z on-shell ($Z_a \approx M_Z$) y otro off-shell.
- Elegir la combinación adecuada mejora la pureza del canal 4μ .

Cortes finales sobre el candidato 4μ

Programación

- **Corte en p_T de Z_a :** exige PT_LEAD, PT_SUB (20/10 típico).
- **Corte 20/10/7/5 en 4μ** (opcional): si ENFORCE_4MU_PT.
- **Ventanas de masa:** $m_{Za} \in (MZA_MIN, MZA_MAX)$, $m_{Zb} \in (MZB_MIN, MZB_MAX)$.
- **Sanity check:** $m_{4\mu} > 0$.

Física

- 20/10 y 20/10/7/5 están motivados por triggers y aceptación.
- Ventanas de masa recortan regiones de fondo, manteniendo señal potencial.

Programación

- Pares del pareo ganador: `h_mZij_best` (sólo las dos parejas efectivas, Z_a y Z_b).
- Finales: `h_mZa`, `h_mZb`, `h_m4mu`.
- Por categorías: `AllTight/AllMedium/AllLoose`.
- Por composición: `m4mu_by_comp_*`, `mZa_by_comp_*`, `mZb_by_comp_*`.
- `FourMuTree`: masas, 4-vector del sistema, índices, niveles y `pairing_code`.

Física

- `h_mZa` \Rightarrow pico en Z . `h_m4mu` \Rightarrow posible exceso cerca de 125 GeV.

Programación

- Cutflow (11 pasos): desde *All entries* hasta *filled*.
- Escritura en `out_zz4mu.root` de todos los histogramas y del FourMuTree.

Física

- El cutflow diagnostica dónde se pierden eventos y orienta la optimización de cortes (mejor S/\sqrt{B}).

Realzar el pico en Z (control)

- `-mza-min=80 -mza-max=100 -pt-min=5 -eta-max=2.4`
`-use-iso -iso-max=0.35 -sip3d-max=4`

Buscar señal de Higgs (más estricto)

- `-comb-search -pt-lead=20 -pt-sub=10 -enforce-4mu-pt`
`-mza-min=50 -mza-max=106 -mzb-min=12 -mzb-max=120`

Diagnóstico de emparejamientos

- `-fill-pairs-pre` para histogramas de pares *antes* de elegir el mejor.

- **Crash/compilación por IP/SIP:** asegurar **declaración y conexión condicional** de `muon_dxy/dz/sip3d`.
- **ISO no existe:** omitir `-use-iso` o ajustar `-iso-branch=` al nombre correcto.
- **Muy pocos eventos:** revisar `h_evt_pre_cf` para identificar la etapa que vacía y relajar ese corte.
- **Pareos raros:** usar `-comb-search` y/o inspeccionar `h_mZij_pre`.

Física vs Programación (visión de conjunto)

Física

- 4μ de bosones Z prompt.
- $Z_a \approx M_Z$, Z_b off-shell.
- Cortes motivados por triggers/aceptancia.
- Señal: posible exceso en $m_{4\mu} \sim 125$ GeV.

Programación

- Pipeline de preselección instrumentado y trazable.
- Pareo OS-OS óptimo con criterio lexicográfico.
- Histos + TTree para análisis posteriores.
- Cutflow para auditoría y tuning.

Apéndice: variables del TTree FourMuTree

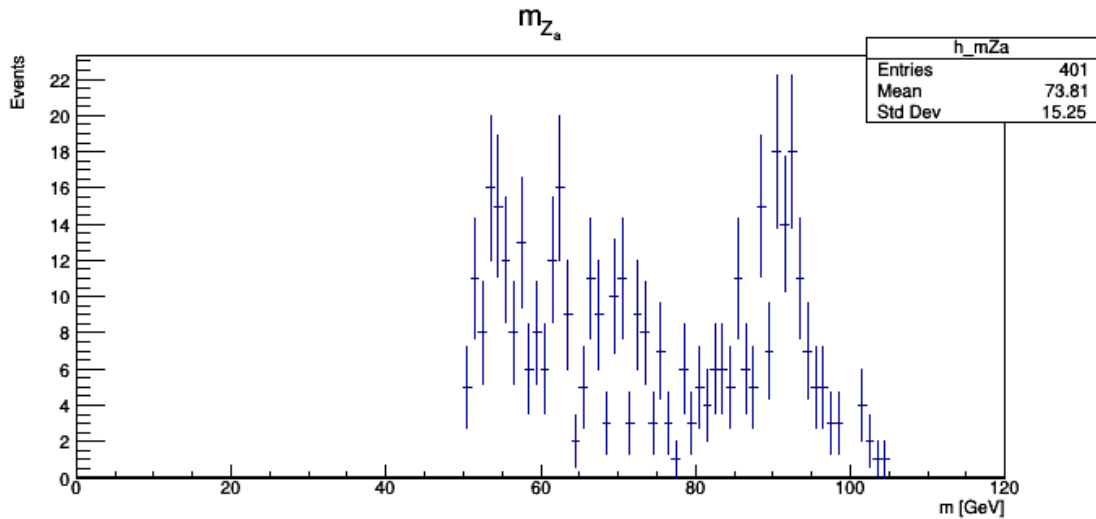
- **4-vector sistema:** `mass4mu`, `pt_4mu`, `eta_4mu`, `phi_4mu`, `px4mu`, `py4mu`, `pz4mu`, `E4mu`.
- **Pares:** `mZa`, `mZb`, $dMZa = |mZa - M_Z|$, $dMZb = |mZb - M_Z|$.
- **Clasificación:** `category` (3=AllTight, 2=AllMedium, 1=AllLoose, 0=Unclassified), `nT`, `nM`, `nL`.
- **Trazabilidad:** `mu_levels[4]`, `idxSel[4]`, `idxZa[2]`, `idxZb[2]`, `pairing_code`.

Código: ejemplo de ejecución reproducible

```
./higgs4mu_select analisis_final_clasificados.root Events out_zz4mu.root \  
  --comb-search --pt-lead=20 --pt-sub=10 --enforce-4mu-pt \  
  --mza-min=50 --mza-max=106 --mzb-min=12 --mzb-max=120
```

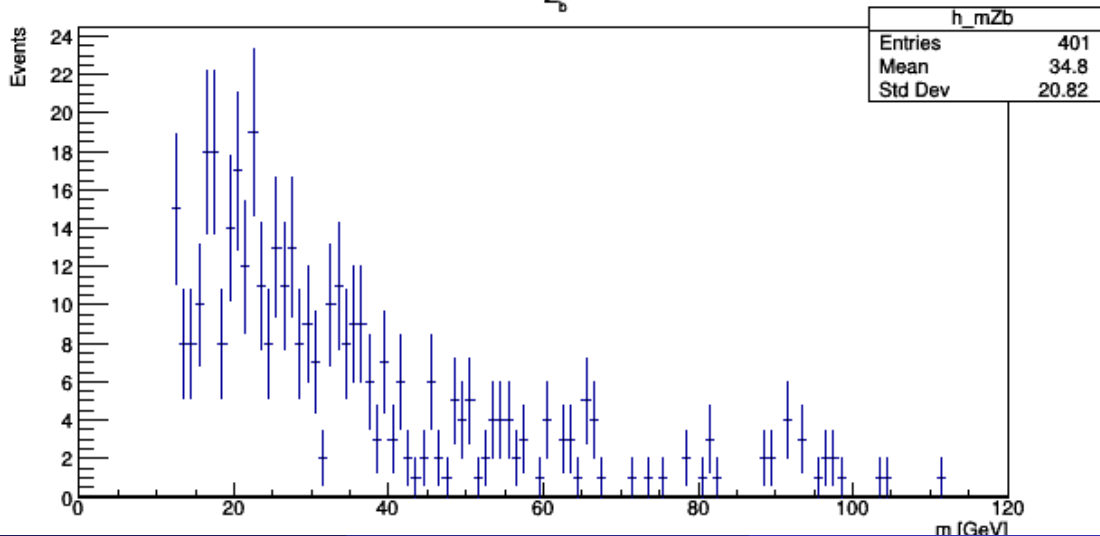
Sugerencia: guarda el comando en un `.sh` con fecha y seed (si aplica) para trazabilidad.

Resultados: m_{Z_a}

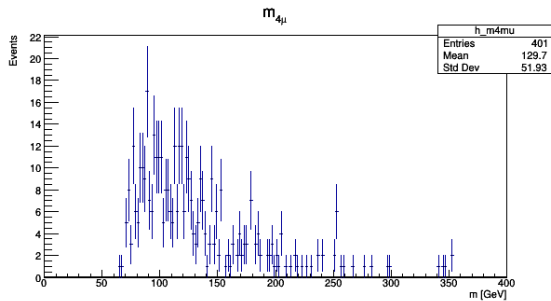


Resultados: m_{Z_b}

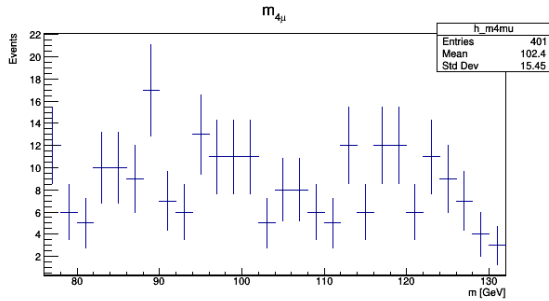
m_{Z_b}



Resultados: $m_{4\mu}$ (completo y zoom 80–130 GeV)



$m_{4\mu}$ (completo)



$m_{4\mu}$ (zoom 80–130 GeV)

¡Gracias!
Codigo en GitHub