

# Plan analizy

- Wybieramy ciekawy proces fizyczny.
  - W LHCb przeważnie jest to rozpad mezonów i barionów zawierających kwark c lub b.
  - Wybieramy np. produkcję i rozpad mezonu  $D^0$ .
- Sprawdzamy w [Particle Data Group](#) sposoby rozpadu
- Szacujemy szanse obserwacji w spektrometrze LHCb (tryger)
- Przygotowujemy kryteria selekcji.
- Przeprowadzamy analizę.
- Publikujemy wynik.

**CHARMED MESONS**  
**( $C = \pm 1$ )**  
 $D^+ = c\bar{d}, D^0 = c\bar{u}, \bar{D}^0 = \bar{c}u, D^- = \bar{c}d$ , similarly for  $D^{*}$ 's

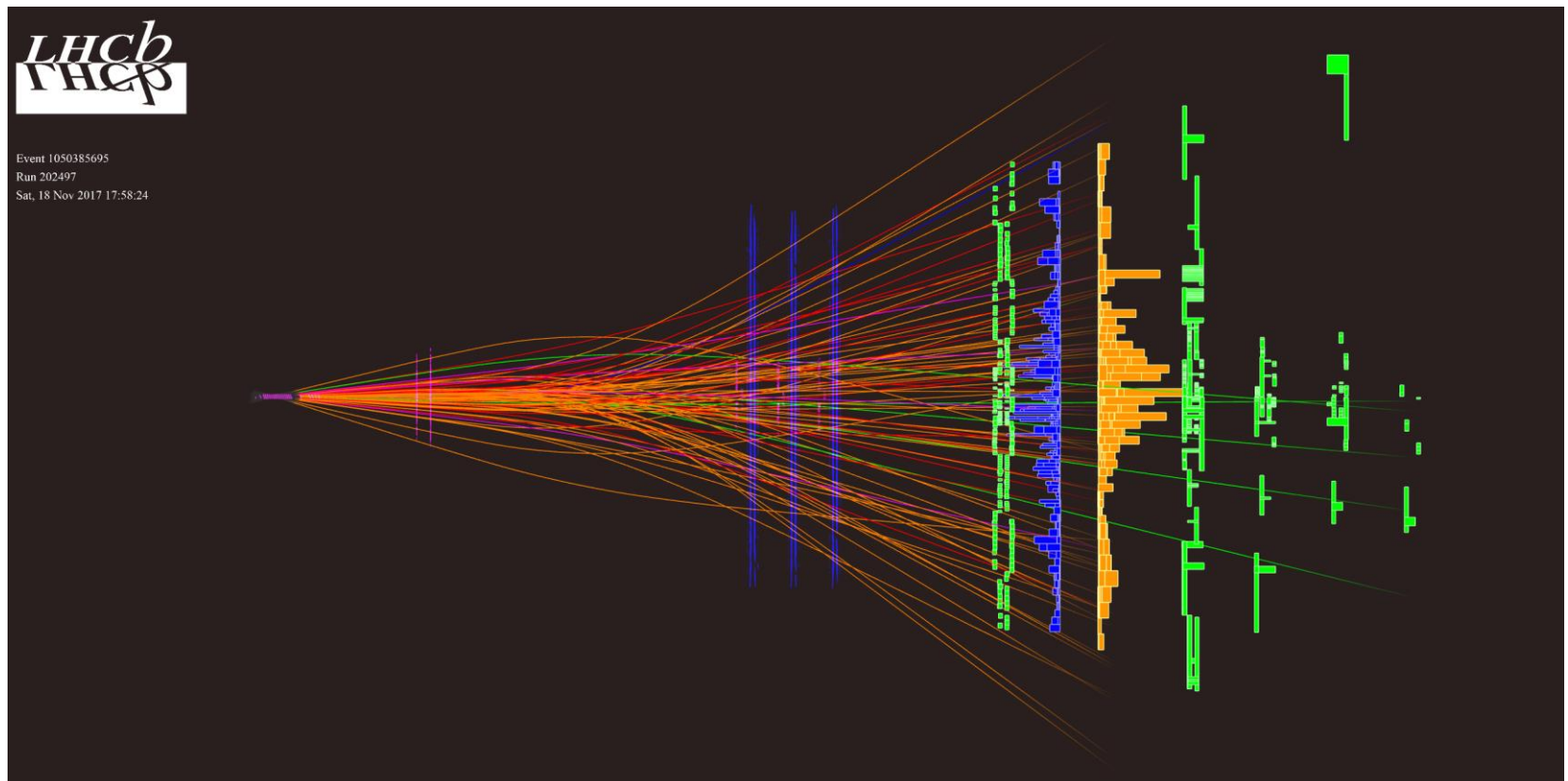
$D^0$ DECAY MODES	Fraction ( $\Gamma_i/\Gamma$ )
<b>Hadronic modes with one <math>\bar{K}</math></b>	
$\Gamma_{32} \quad K^- \pi^+$	( 3.89 ± 0.04 ) %
$\Gamma_{33} \quad K^+ \pi^-$	( 1.385 ± 0.027 ) × 10 <sup>-4</sup>
$\Gamma_{34} \quad K_S^0 \pi^0$	( 1.19 ± 0.04 ) %
$\Gamma_{35} \quad K_L^0 \pi^0$	( 10.0 ± 0.7 ) × 10 <sup>-3</sup>
$\Gamma_{36} \quad K_S^0 \pi^+ \pi^-$	[e] ( 2.75 ± 0.18 ) %
$\Gamma_{37} \quad K_S^0 \rho^0$	( 6.2 $\begin{smallmatrix} + 0.6 \\ - 0.8 \end{smallmatrix}$ ) × 10 <sup>-3</sup>
$\Gamma_{38} \quad K_S^0 \omega, \omega \rightarrow \pi^+ \pi^-$	( 2.0 ± 0.6 ) × 10 <sup>-4</sup>
$\Gamma_{39} \quad K_S^0 (\pi^+ \pi^-)_{S\text{-wave}}$	( 3.3 ± 0.7 ) × 10 <sup>-3</sup>

# Analiza

- Mezon  $D^0$ 
  - może być wyprodukowany bezpośrednio w zderzeniach proton-proton,
  - może pochodzić z rozpadu innej cząstki, np.  $B^0$

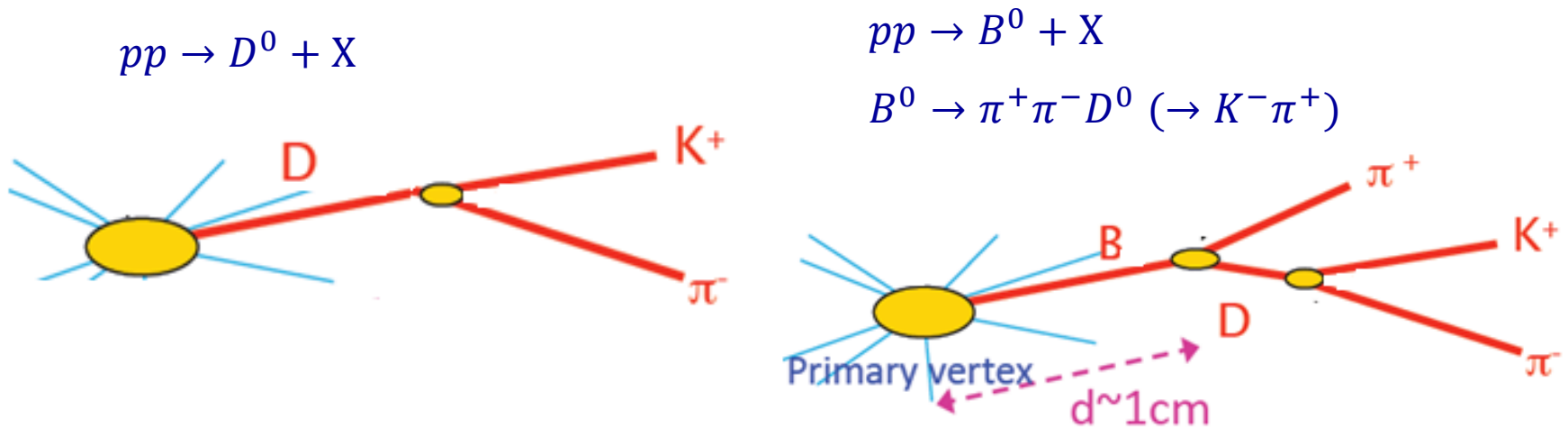
$\Gamma_{43} \quad \bar{D}^0 \pi^+ \pi^-$

$( 8.8 \pm 0.5 ) \times 10^{-4}$



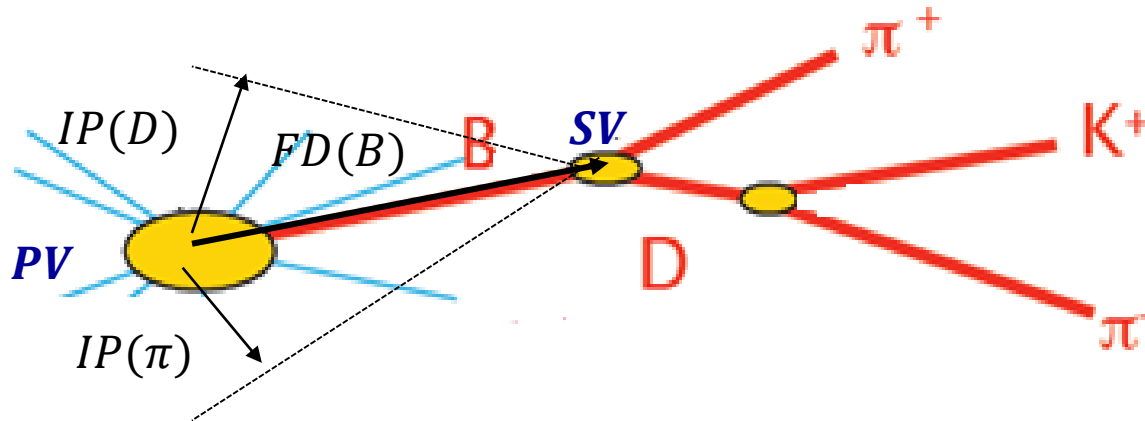
# Analiza

- Mezon  $D^0$ 
  - może być wyprodukowany bezpośrednio w zderzeniach proton-proton,
  - może pochodzić z rozpadu innej cząstki, np.  $B^0$
- Znajdujemy parametry, które wybiorą z ogółu przypadków rozpad:  $D^0 \rightarrow K^- \pi^+$  oraz/i pozwolą na odróżnienie produkcji bezpośredniej (prompt) i wtórnej.



# Parametry mezonu D

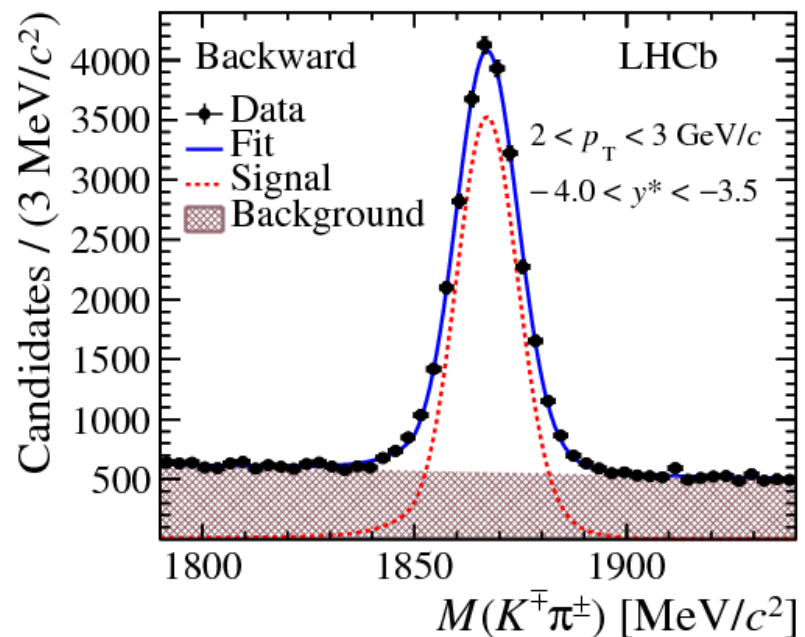
- Mezon  $D^0$ 
  - może być wyprodukowany bezpośrednio w zderzeniach proton-proton,
  - może pochodzić z rozpadu innej cząstki, np.  $B^0$
- Znajdujemy parametry, które wybiorą z ogółu przypadków rozpad:  $D^0 \rightarrow K^- \pi^+$  oraz/i pozwolą na odróżnienie produkcji bezpośredniej (prompt) i wtórnej.



- $IP(D)$  – parametr zderzenia:
- czas życia policzony z drogi (FD) pomiędzy wierzchołkiem produkcji i rozpadu,
- jakość dopasowania wierzchołków (pierwotnego PV i wtórnego SV)
- identyfikacja pionów i kaonów

# Jak to można zrobić?

- Analizę przeprowadzimy na przygotowanym zbiorze danych MC.
- Zbiór z danymi ma strukturę Ntupla (n-krotki).
- Ntupel jest właściwie bazą danych, do której kierujemy zapytania, np:
  - narysuj pęd dodatnio naładowanej cząstki, która miała  $IP > 0.1$ ,
  - narysuj masę cząstek o pędach powyżej 1 GeV, itp.
- Zapytania piszemy w C++ (lub Pythonie) w architekturze ROOT.
- Efekt końcowy: rozkład masy (np.) z dopasowaniem.



# Struktura Ntupla

ROOT Object Browser

Przykład analizy „wyklikanej” w roocie.....

Browser File Edit View Options Tools Help

Files

- root
- PROOF Sessions
- ROOT Files
  - Projekt2017.root
    - DecayTree;77
      - h1\_OWNPV\_X
      - h1\_OWNPV\_Y
      - h1\_OWNPV\_Z
      - h1\_IP\_OWNPV
      - h1\_ORIVX\_X
      - h1\_ORIVX\_Y
      - h1\_ORIVX\_Z
      - h1\_ORIVX\_CHI2
      - h1\_PX**
      - h1\_PY
      - h1\_PZ
      - h1\_M
      - h1\_ID
      - h2\_OWNPV\_X
      - h2\_OWNPV\_Y
      - h2\_OWNPV\_Z
      - h2\_IP\_OWNPV
      - h2\_ORIVX\_X
      - h2\_ORIVX\_Y

Filter: All Files (\*.\*)

Canvas\_1 Editor 1

h1\_PX

htemp	
Entries	1821072
Mean	49.65
Std Dev	2967

Command

Command (l) Można też napisać skrypt lub program do analizy

# Struktura Ntupla

- PX, PY, PZ, M – odpowiednio współrzędne pędu i masa, w jednostkach naturalnych.
- ID – flaga mówiąca o rodzaju cząstki (identyfikacja), można ją rozkodować patrząc na tablicę: <http://pdg.lbl.gov/2010/reviews/rpp2010-rev-monte-carlo-numbering.pdf>
- ORIVX – współrzędne miejsca powstania cząstki, czyli miejsca rozpadu jej „matki”,
- OWNPV – współrzędne wierzchołka pierwotnego przypadku, czyli punktu interakcji protonów.

Informacje o ORIVX i OWNPV (w mm) pozwalają rozstrzygnąć, czy zrekonstruowana cząstka powstała bezpośrednio w zderzeniach protonów, czy też jest produktem rozpadu innej cząstki,

- IP\_OWNPV – parametr zderzenia – pomaga w odróżnieniu cząstek powstałych bezpośrednio w zderzeniu od tych, które pochodzą z rozpadów.
- D0\_ORIVX – współrzędne wierzchołka, w którym powstał (jeśli jest dobrze wybrany) mezon  $D^0$

... h1\_OWNPV\_X  
... h1\_OWNPV\_Y  
... h1\_OWNPV\_Z  
... h1\_IP\_OWNPV  
... h1\_ORIVX\_X  
... h1\_ORIVX\_Y  
... h1\_ORIVX\_Z  
... h1\_ORIVX\_CHI2  
... h1\_PX  
... h1\_PY  
... h1\_PZ  
... h1\_M  
... h1\_ID

# Skrypt do analizy

```
void Make_meson() {
```

```
    Bchain = new TChain("DecayTree");  
    Bchain->Add("Projekt2017.root");  
    TTree* MyBtree = Bchain;
```

Wczytywanie pliku z danymi i jego struktury („drzewa”)

```
    Double_t    h1_px, h1_py, h1_pz, h1_m;  
    Double_t    h2_px, h2_py, h2_pz, h2_m;  
    Double_t    hh_px, hh_py, hh_pz, hh_m;  
    Double_t    h1_e, h2_e;
```

Deklaracja potrzebnych zmiennych

```
    MyBtree->SetBranchAddress("h1_PX", &h1_px);  
    MyBtree->SetBranchAddress("h1_PY", &h1_py);  
    MyBtree->SetBranchAddress("h1_PZ", &h1_pz);  
    MyBtree->SetBranchAddress("h1_M", &h1_m);
```

Masy i pędy cząstek z ntupla (czyli bazy danych) są przepisywane do lokalnych zmiennych px, py...

```
    MyBtree->SetBranchAddress("h2_PX", &h2_px);  
    MyBtree->SetBranchAddress("h2_PY", &h2_py);  
    MyBtree->SetBranchAddress("h2_PZ", &h2_pz);  
    MyBtree->SetBranchAddress("h2_M", &h2_m);
```

Deklaracja histogramu. Jak dobrać zakres i binowanie?

```
    Int_t NBINS=100;  
    Double_t GeV=0.001;  
    Double_t min_mass = 0.6; // dyskusja  
    Double_t max_mass = 5;
```

```
    TH1D* HH_m= new TH1D( "HH_mass", "HH_mass ",NBINS, min_mass
```



# Wykonanie ćwiczenia

```
Double_t mass_h12;

Int_t Evt_tot = MyBtree->GetEntries();
for(Int_t event = 0; event < Evt_tot; ++event){
    Bchain->GetEvent(event);

    hh_px = h1_px+h2_px;
    .....

    h1_e = sqrt(h1_px*h1_px+h1_py*h1_py+h1_pz*h1_pz+
h1_m*h1_m);
    .....
    mass_h12= .....

    HH_m->Fill(mass_h12*GeV);

    TCanvas* mass_h12_can=newTCanvas("mass_h12_can","mass_h12_can",0,0,800,600);
    mass_h12_can->cd(1);
    HH m->Draw();
}
```

Właściwa analiza.  
Tutaj dla każdego przypadku  
można dokonać obliczeń,  
nałożyć kryteria, a na końcu  
wczytać wartość do histogramu.

Deklaracja pola na rysunek i  
wykonanie rysunku histogramu

# Wykonanie ćwiczenia

1. Analizę można wykonać pisząc skrypt lub program wykonywalny pod Rootem.
2. W najprostszej wersji wykonujemy go poprzez:

```
root [2] .X Make_meson.cpp
```

Ocenie podlega:

- sposób przeprowadzenia selekcji: próby różnych kryteriów, różne kombinacje ładunków, szukanie przypadków z nieprawidłową identyfikacją.
- sposób napisania programu: prosty skrypt, napisanie funkcji, klasy do selekcji, użycie `TLorentzVector`, etc.
- sposób opisu: wstęp teoretyczny (krótki, ale sensowny), poprawność wykonania rysunków, porównanie z wartościami tablicowymi, wnioski.