

MC information

- MC matching information

Yonsei University

Chanho Kim

2023/04/11

content

- ◆ MC matching
 - ◆ Usage
 - ◆ Method
- ◆ Matching failure case
- ◆ Decay String Grammar in basf2
- ◆ MC matching variable

MC matching

- Usage :
 - Optimization of selection
 - => Labeling the training data for MVA
 - Calculation of signal efficiencies
 - Investigation of background source
 - => TopoAna is more popular for this purpose in these days
- Method :
 - Relate mdst dataObjects (Tracks, ECLClusters, KLMClusters) with MCParticles (with weights)
 - Relate reconstructed particles with MC particles

Matching failure case (`variable == NaN`)

- CloneTracks :
 - isCloneTrack / isOrHasCloneTrack
- Fake Tracks
- Wrong charge
 - isWrongCharge
- Overlapping clusters
 - nMCMatches
- Cluster split-offs
- Beam background

Decay String Grammar in basf2

- Particles and decays

- Particles => ex : K+, pi+, K_S0, etc ...
- each particles has corresponding pdg code such as B+ : 521, B- : -521
- Decays are represented as arrow : D0 -> K- pi+
or for complex decay trees, D*0 -> [D0 -> K- pi+] pi+

- Marker and keywords

- @ : consider as signal regardless to the daughters and its mc information
@Xsd : 0 -> K_S0, @Xsd:1 -> K+ pi- => Xsd:0 and Xsd:1 are both considered as signal
- '...' : missing massive final state particles are ignored for mc matching
- '?nu' : missing neutrinos are ignored for mc matching

Ex) D0 -> K- pi+ ... nu? : D0 considered as signal only considering K- pi+ not additional massive particles and nu

- Arrow

- -> : treat as signal considering only final state particles but not radiated photons
- =direct=> : treat as signal considering about intermediate state but not radiated photons
- =norad=> : treat as signal considering about radiated photons but not intermediate state
- = exact=> : direct + norad case

MC matching variables (mcErrors)

- mcErrors :

bit-wise error flags indicate what is wrong in MC matching

<code>c_ Correct = 0</code>	This Particle and all its daughters are perfectly reconstructed.
<code>c_ MissFSR = 1</code>	A Final State Radiation (FSR) photon is not reconstructed (based on <code>MCParticle::c_IsFSRPhoton</code>).
<code>c_ MissingResonance = 2</code>	The associated <code>MCParticle</code> decay contained additional non-final-state particles (e.g. a rho) that weren't reconstructed. This is probably O.K. in most cases.
<code>c_ DecayInFlight = 4</code>	A Particle was reconstructed from the secondary decay product of the actual particle. This means that a wrong hypothesis was used to reconstruct it, which e.g. for tracks might mean a pion hypothesis was used for a secondary electron.
<code>c_ MissNeutrino = 8</code>	A neutrino is missing (not reconstructed).
<code>c_ MissGamma = 16</code>	A photon (not FSR) is missing (not reconstructed).
<code>c_ MissMassiveParticle = 32</code>	A generated massive FSP is missing (not reconstructed).
<code>c_ MissKlong = 64</code>	A Klong is missing (not reconstructed).
<code>c_ MisID = 128</code>	One of the charged final state particles is mis-identified (wrong signed PDG code).
<code>c_ AddedWrongParticle = 256</code>	A non-FSP Particle has wrong PDG code, meaning one of the daughters (or their daughters) belongs to another Particle.
<code>c_ InternalError = 512</code>	No valid match was found. For tracks, it indicates that there is not a true track related to the reconstructed one. Might indicate fake or background track or cluster.
<code>c_ MissPHOTOS = 1024</code>	A photon created by PHOTOS was not reconstructed (based on <code>MCParticle::c_IsPHOTOSPhoton</code>).
<code>c_ AddedRecoBremsPhoton = 2048</code>	A photon added with the bremsstrahlung recovery tools (<code>correctBrems</code> or <code>correctBremsBelle</code>) has no MC particle assigned, or it doesn't belong to the decay chain of the corrected lepton mother.

MC matching variables (isSignal)

- Possible value : NaN(failure), 0(fake), 1(signal)
- Aliases for mcErrors variable
 - isSignal : mcErrors == 0
 - isSignalAcceptMissingNeutrino : mcErrors == 0 or mcErrors == 8
 - isSignalAcceptMissingGamma : mcErrors == 0 or mcErrors == 16
 - isSignalAcceptBremsPhotons : mcErrors == 0 or mcErrors == 2048
etc ...
- user define method : `create_isSignal_alias(aliasname,flags)`
 - For example
`create_isSignal_alias("isSignalAcceptMissingNeutrinoAndGamma",[8,16])`

Reference

- [Bremsstrahlung and MC matching \(belle2.org\)](#)
- [7.1.2. DecayString — basf2 light-2303-iriomote documentation \(belle2.org\)](#)