Belle II Analysis Software Tutorial

Belle II Analysis Software Overview

Anže Zupanc
Jožef Stefan Institute and University of Ljubljana

Setup the environment

- Follow instructions posted on today's tutorial confluence page

  https://confluence.desy.de/display/BI/Physics+HandsOnAnalysisTutorialJune2017

- Confirm that everything works for you by executing

  basf2 --info

  basf2 modularAnalysis.py

  basf2 variables.py

Understanding the input: mDST format

- Physics analyses are performed on mDST (mini Data Summary Table) data format
  - *Do you know what objects does it include?*
  - *Do you know how are these objects related to each other?*

Understanding the input: mDST format

- Physics analyses are performed on mDST (mini Data Summary Table) data format
  - *Do you know what objects does it include?*
  - *Do you know how are these objects related to each other?*

It's very easy to find out! Let's print the content of the DataStore for 10 events in one generic MC7 file:

```
basf2 $BELLE2_RELEASE_DIR/analysis/examples/printDataStore.py -n 10 -i input_mdst_file.root
```

Understanding the input: mDST format

- Physics analyses are performed on mDST (mini Data Summary Table) data format
    - *Do you know what objects does it include?*
    - *Do you know how are these objects related to each other?*

It's very easy to find out! Let's print the content of the DataStore for 10 events in one generic MC7 file:

```
basf2 $BELLE2_RELEASE_DIR/analysis/examples/printDataStore.py -n 10 -i input_mdst_file.root
```

- `$BELLE2_RELEASE_DIR/analysis/examples/printDataStore.py`: steering file
- `-n 10`: process only 10 events
- `-i input_mdst_file.root`: use this file as input

Understanding the input: mDST format

- Physics analyses are performed on mDST (mini Data Summary Table) data format
  - *Do you know what objects does it include?*
  - *Do you know how are these objects related to each other?*

It's very easy to find out! Let's print the content of the DataStore for 10 events in one generic MC7 file:

```
basf2 $BELLE2_RELEASE_DIR/analysis/examples/printDataStore.py -n 10 -i input_mdst_file.root
```

↑ steering file (points to $BELLE2_RELEASE_DIR/analysis/examples/printDataStore.py)

↑ process only 10 events (points to -n 10)

↑ use this file as input (points to -i input_mdst_file.root)

MC7 can be found here:
https://confluence.desy.de/display/BI/MC7+samples+for+analysis+users

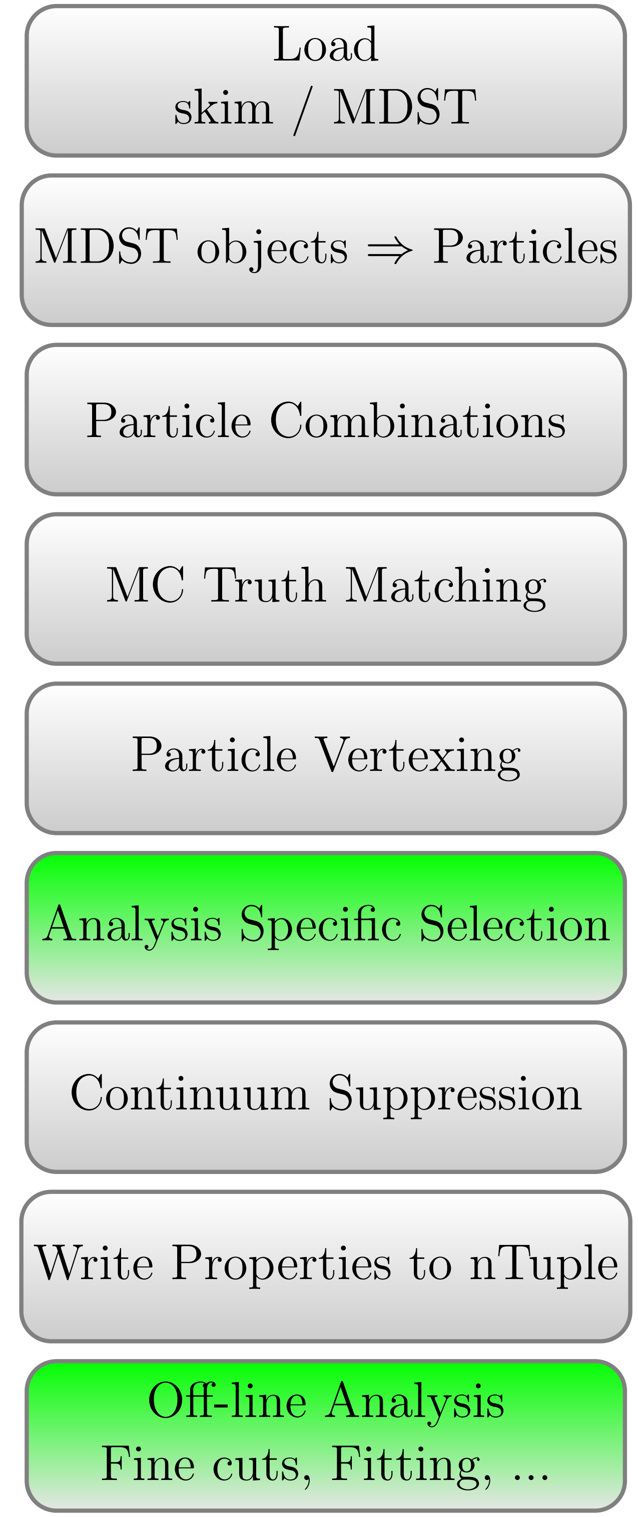
Understanding the input: mDST format

```
[INFO] =============================================================================
[INFO] DataStore collections in event 260010
[INFO] =============================================================================
[INFO] Type                 Name                      #Entries           <Event>
[INFO] RelationContainer    ECLClustersToMCParticles
[INFO] EventMetaData        EventMetaData
[INFO] RelationContainer    KLMClustersToECLClusters
[INFO] RelationContainer    KLMClustersToMCParticles
[INFO] RelationContainer    TracksToECLClusters
[INFO] RelationContainer    TracksToKLMClusters
[INFO] RelationContainer    TracksToMCParticles
[INFO] RelationContainer    TracksToPIDLikelihoods
[INFO] ECLCluster[]         ECLClusters               39
[INFO] KLMCluster[]         KLMClusters               2
[INFO] MCParticle[]         MCParticles               107
[INFO] PIDLikelihood[]      PIDLikelihoods            17
[INFO] TrackFitResult[]     TrackFitResults           37
[INFO] Track[]              Tracks                    17
[INFO] V0[]                 V0s                       10
[INFO]
[INFO] -----------------------------------------------------------------------------
[INFO] Type                 Name                      #Entries      <Persistent>
[INFO] FileMetaData         FileMetaData
[INFO] ProcessStatistics    ProcessStatistics
[INFO] BackgroundInfo[]     BackgroundInfos           1
[INFO]
[INFO] =============================================================================
```

Belle II Analysis Software

- Modular, **common** set of configurable algorithms with intuitive python steering and sequencing
    - **Analysis Specific tasks**
        - your work, **the physics**
    - **Generic tasks**
        - similar in each analysis
        - **Standardized** $\rightarrow$ *less error prone*

- *User writes simple python scripts to reconstruct decays of interest and writes desired quantities to nTuples for off-line analysis*



Further information

Analysis Software confluence page **Tutorials**

Belle II Analysis Software

- Modular, **common** set of configurable algorithms

```
# load input ROOT file
inputMdst('inputFile.root')

# create and fill ParticleLists, '' means no cut, take all
fillParticleList('e-', '') # labels preferred, but not required
fillParticleList('gamma:all', '')

# create and fill with candidates that pass cuts
fillParticleList('e+:good', 'eid > 0.1')
fillParticleList('gamma:highE', 'E > 1.0')

# create tuples first
muGood = ('mu+:good', 'muid > 0.1')
piGood = ('pi+:good', 'piid > 0.1')
fillParticleLists([muGood, piGood])
```

Load

articles

ions

ing

ng

lection

ssion

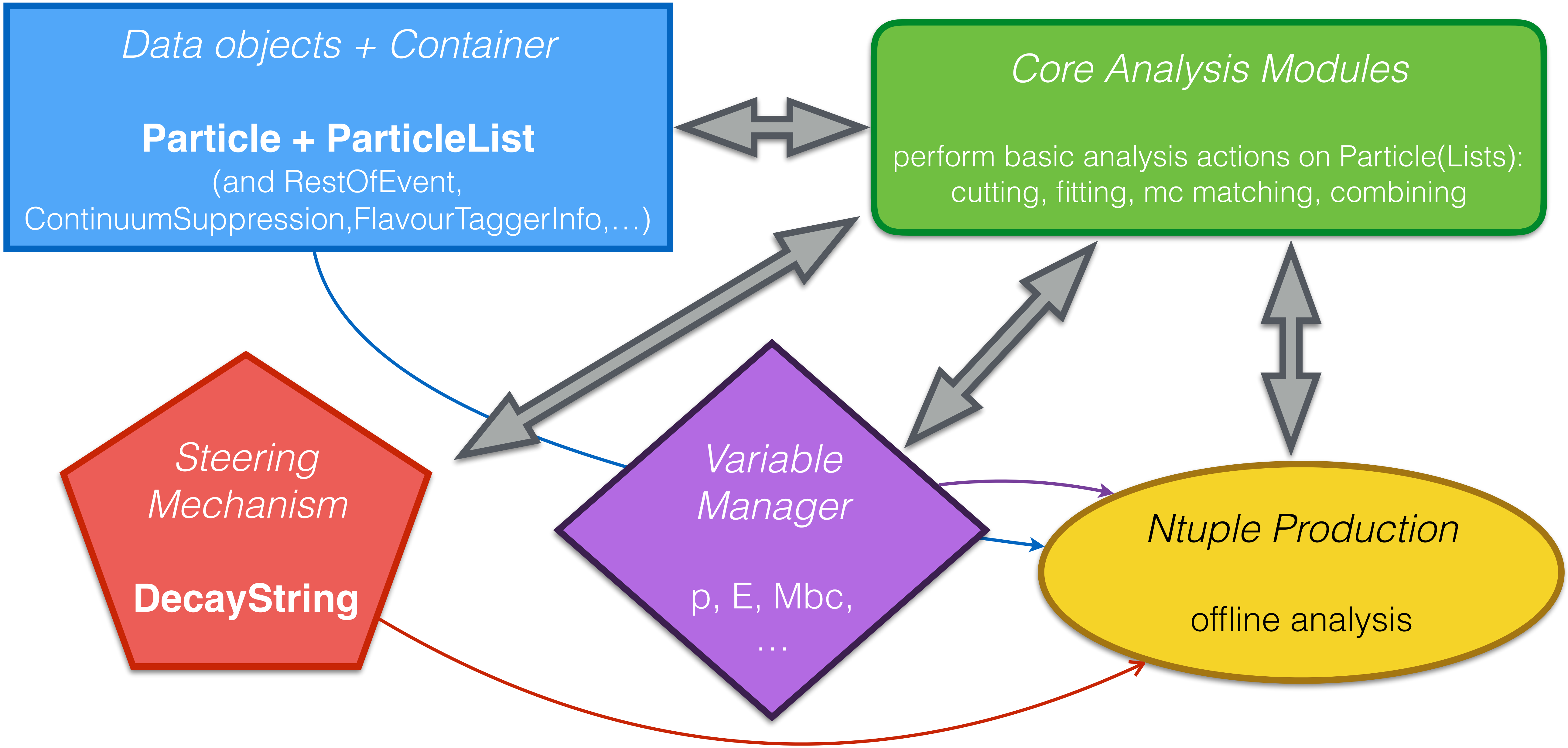
nTuple

Off-line Analysis
Fine cuts, Fitting, ...

Analysis Software confluence page

Tutorials

Belle II Analysis Software



Python functions (for more human readable analysis steering scripts)

Data objects: Particle class

Particle class is a common representation of all particle types

- final states particles detected at Belle II
  - charged e/μ/π/K/p reconstructed as Tracks
  - γ reconstructed as ECLClusters
  - K_{long} reconstructed as KLMClusters in the ECL/KLM
- kinematically reconstructed (composite) particles
  - π^0, K_S, D, B, ...

Private members of the *Particle* are limited to only those which define a particle and are common to all particle types (momentum, position, PDG code, covariance m., …).

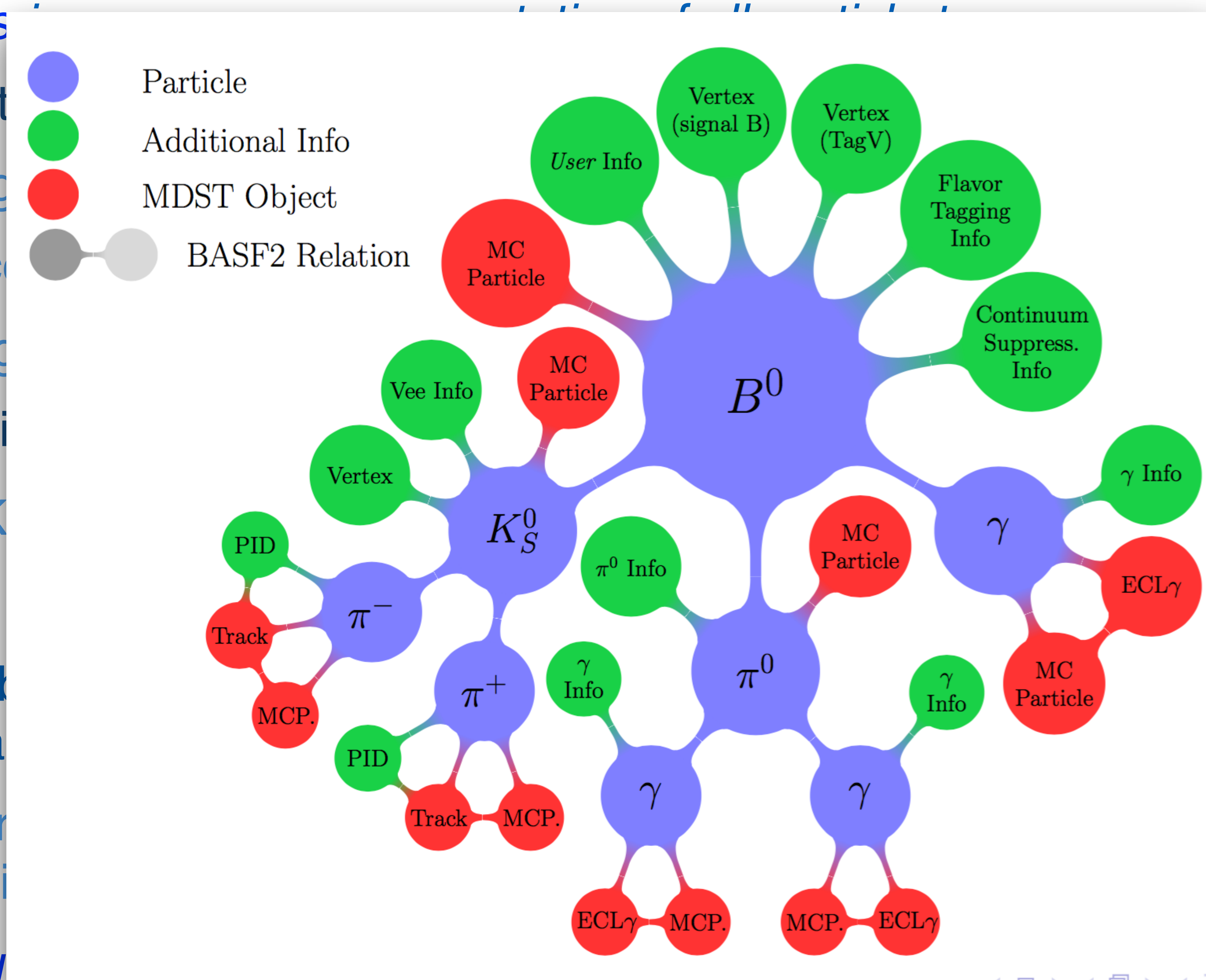
All other information which exists and is relevant only for certain types of particles is saved in the independent data-objects accessible via BASF2 Relations:

- *ContinuumSupression* (various FW moments, angles, …)
- *FlavourTaggerInfo*
- *ExtraInfo* (any user-defined floating-point value identified by a string key)

Data objects: Particle class

Particle class

- final stat
  - charg
  - γ rec
  - Klong
- kinemati
  - π^0, K



Private memb … e a particle and are common to a … ance m., …).

All other infor … s of particles is saved in the i … ns:

- *Continu*
- *FlavourTaggerInfo*
- *ExtraInfo* (any user-defined floating-point value identified by a string key)

ParticleList: collection of Particles

ParticleList provides ability to group together particles and anti-particles that logically belong together:

- all π^0 candidates that have invariant mass within certain window
- all D^0 candidates reconstructed in $D^0 \to K^-\pi + (\overline{D}^0 \to K^+\pi^-)$ decay mode

ParticleList can store only particles with same PDG code (which however can be reconstructed in different decay modes).

ParticleList doesn't have ownership of *Particle* objects that it collects.

ParticleList is the dataobject with which the analysis modules operate (input/output).

Example:

- *reconstruction of D0 -> K- pi+ pi0 candidates with* ***ParticleCombiner***

```
reconstructDecay('D0:myD0 -> K-:tight pi+:all pi0:loose','')
```

ParticleList: collection of Particles

ParticleList provides ability to group together particles and anti-particles that logically belong together:

- all π^0 candidates that have invariant mass within certain window
- all D^0 candidates reconstructed in $D^0 \to K^-\pi + (\overline{D}^0 \to K^+\pi^-)$ decay mode

ParticleList can store only particles with same PDG code (which however can be reconstructed in different decay modes).

ParticleList doesn't have ownership of *Particle* objects that it collects.

ParticleList is the dataobject with which the analysis modules operate (input/output).

Example:

- *reconstruction of D0 -> K- pi+ pi0 candidates with* ***ParticleCombiner***

reconstructDecay('D0:myD0 -> K-:tight pi+:all pi0:loose','')

Python module to wrap ***ParticleCombiner*** analysis module

ParticleList: collection of Particles

ParticleList provides ability to group together particles and anti-particles that logically belong together:

- all π^0 candidates that have invariant mass within certain window
- all D^0 candidates reconstructed in $D^0 \to K^-\pi + (\overline{D}^0 \to K^+\pi^-)$ decay mode

ParticleList can store only particles with same PDG code (which however can be reconstructed in different decay modes).

ParticleList doesn't have ownership of *Particle* objects that it collects.

ParticleList is the dataobject with which the analysis modules operate (input/output).

Example:

- *reconstruction of D0 -> K- pi+ pi0 candidates with* ***ParticleCombiner***

DecayString

reconstructDecay('D0:myD0 -> K-:tight pi+:all pi0:loose','')

ParticleList: collection of Particles

ParticleList provides ability to group together particles and anti-particles that logically belong together:

- all π^0 candidates that have invariant mass within certain window
- all D^0 candidates reconstructed in $D^0 \to K^-\pi + (\overline{D}^0 \to K^+\pi^-)$ decay mode

ParticleList can store only particles with same PDG code (which however can be reconstructed in different decay modes).

ParticleList doesn't have ownership of *Particle* objects that it collects.

ParticleList is the dataobject with which the analysis modules operate (input/output).

Example:

- *reconstruction of D0 -> K- pi+ pi0 candidates with* ***ParticleCombiner***

reconstructDecay('D0:myD0 -> K-:tight pi+:all pi0:loose','')

output — D0:myD0; input — K-:tight; input — pi+:all; input — pi0:loose

ParticleList(s)

ParticleList: collection of Particles

The unique identifier of the ParticleList is its name. According to the naming convention the ParticleList's name has to be of the form:

ParicleListName = particle_name:label

where particle_name is the name of the particle as given in the evt.pdl and the label can be any string indicating the selection criteria or decay mode (or anything else) used to reconstruct the particles. Examples are:

- pi+:loose - pi+ candidates passing loose PID requirements
- D0:kpi - D0 candidates reconstructed in D0->Kpi decays
- B+:myVeryOwnBCandidates - my very own precious B+ candidates

evt.pdl = $BELLE2_EXTERNALS_DIR/share/evtgen/evt.pdl

ParticleList: collection of Particles

What if I’m looking for an hypothetical particle that is not included in the evt.pdl?

- Add it in your steering file by yourself!

```
import pdg
pdg.add_particle(name, pdg, mass(GeV), width(GeV),
                 charge(e), spin, max_width(GeV), lifetime(0), pythiaID)
```

 evt.pdl = $BELLE2_EXTERNALS_DIR/share/evtgen/evt.pdl

Python functions

List of Steering functions

Function name	Short description	Defined in
Input/Output		
inputMdst	loads content of the specified root file to the `DataStore` with the `RootInput` module	modularAnalysis.py
loadMdstList	loads content of the specified root files to the `DataStore` with the `RootInput` module	modularAnalysis.py
outputMdst	saves only mDST-level persistent dataobjects from `DataStore` to the specified file with `RootOutput` module	modularAnalysis.py
add_mdst_output	saves only mDST-level dataobjects from `DataStore` to the specified file with `RootOutput` module	reconstruction.py
`outputUdst`	saves microDST-level dataobjects from `DataStore` to the specified file with `RootOutput` module	modularAnalysis.py
`removeParticlesNotInLists`	Removes all Particles that are not in a given list of ParticleLists (or daughters of those). All relations from/to Particles, daughter indices, and other ParticleLists are fixed.	modularAnalysis.py
MC Production		
generateY4S	generates e+e- $\rightarrow$ Y(4S) events where Y(4S) decays in user specified way	modularAnalysis.py
generateContinuum	generates e+e- $\rightarrow \gamma \rightarrow$ qq-bar events where light quarks hadronize and decay in user specified way	modularAnalysis.py
add_simulation	simulates detector response	simulation.py
add_reconstruction	performs reconstruction	reconstruction.py
`loadGearbox`	Loads Gearbox module to the path.	modularAnalysis.py
Reconstruction - Final State Particles		
`fillParticleList`	Creates Particles of the desired type from the corresponding MDST dataobjects. The following types of the particles can be loaded: o) charged final state particles (input MDST type = Tracks): `e+`, `mu+`, `pi+`, `K+`, `p`, `deuteron` (and charge conjugated particles) o) neutral final state particles `gamma` (input MDST type = ECLCluster) and `K_S0`, `Lambda0` (input MDST type = V0)	modularAnalysis.py
`fillParticleLists`	Same as above (multiple lists are created)	modularAnalysis.py
`fillConvertedPhotonsList`	Creates photon Particle object for each e+e- combination in the V0 StoreArray and adds it to the ParticleList	modularAnalysis.py
`fillParticleListFromMC`	Creates Particle object for each MCParticle of the desired type found in the `StoreArray` adds it to the ParticleList	modularAnalysis.py

The full list of all functions (with parameters explained) that are defined can be printed out with

```
basf2 modularAnalysis.py
basf2 vertex.py
```

Exercise 1

- Use MC7 sample from before
- Create charged pion, photon and Kshort candidates (without any selection criteria)
- print content of DataStore as before
- print momentum of each pion, photon and invariant mass of each Kshort

Exercise 1

- Use MC7 sample from before
- Create charged pion, photon and Kshort candidates (without any selection criteria)
- print content of DataStore as before
- print momentum of each pion, photon and invariant mass of each Kshort

```
[INFO] ===========================================================================
[INFO] DataStore collections in event 260010
[INFO] ===========================================================================
[INFO] Type                 Name                       #Entries           <Event>
[INFO] RelationContainer    ECLClustersToMCParticles
[INFO] EventMetaData        EventMetaData
[INFO] RelationContainer    KLMClustersToECLClusters
[INFO] RelationContainer    KLMClustersToMCParticles
[INFO] ParticleList         K_S0:all
[INFO] ParticleExtraInfoMap  ParticleExtraInfoMap
[INFO] RelationContainer    ParticlesToMCParticles
[INFO] RelationContainer    ParticlesToPIDLikelihoods
[INFO] RelationContainer    TracksToECLClusters
[INFO] RelationContainer    TracksToKLMClusters
[INFO] RelationContainer    TracksToMCParticles
[INFO] RelationContainer    TracksToPIDLikelihoods
[INFO] ParticleList         gamma:all
[INFO] ParticleList         pi+:all
[INFO] ParticleList         pi-:all
[INFO] ECLCluster[]         ECLClusters                39
[INFO] KLMCluster[]         KLMClusters                2
[INFO] MCParticle[]         MCParticles                107
[INFO] PIDLikelihood[]      PIDLikelihoods             17
[INFO] Particle[]           Particles                  59
[INFO] TrackFitResult[]     TrackFitResults            37
[INFO] Track[]              Tracks                     17
[INFO] V0[]                 V0s                        10
[INFO]
[INFO] ---------------------------------------------------------------------------
[INFO] Type                 Name                       #Entries      <Persistent>
[INFO] FileMetaData         FileMetaData
[INFO] ProcessStatistics    ProcessStatistics
[INFO] BackgroundInfo[]     BackgroundInfos            1
[INFO]
[INFO] ===========================================================================
```

```
[INFO] ===========================================================================
[INFO] [ParticlePrinterModule] START ----------------------------
[INFO]  ParticleLists: pi+:all (7+0) + pi-:all (10+0)
[INFO]  - 0 = 211[0]
[INFO]     o) p = 0.28879
[INFO]  - 4 = 211[1]
[INFO]     o) p = 0.546838
[INFO]  - 5 = 211[2]
[INFO]     o) p = 0.4153
[INFO]  - 6 = 211[3]
[INFO]     o) p = 0.39559
[INFO]  - 8 = 211[4]
[INFO]     o) p = 0.628753
[INFO]  - 9 = 211[5]
[INFO]     o) p = 0.39385
[INFO]  - 11 = 211[6]
[INFO]     o) p = 0.260749
[INFO]  - 1 = -211[7]
```

```
[INFO] [ParticlePrinterModule] START -----------------------------
[INFO]  ParticleList : gamma:all (0+24)
[INFO]  - 17 = 22[0]
[INFO]     o) E = 0.276833
[INFO]  - 18 = 22[1]
[INFO]     o) E = 0.19579
[INFO]  - 19 = 22[2]
[INFO]     o) E = 0.0225137
```

```
[INFO] [ParticlePrinterModule] START -----------------------------
[INFO]  ParticleList : K_S0:all (0+6)
[INFO]  - 43 = 310[0]
[INFO]     o) daughter indices =  41 42
[INFO]     o) M = 0.828873
```

Exercise 1

Why is number of Kshorts not equal to number of V0 dataobjects?

- Use MC7 sample from before
- Create charged pion, photon and Kshort candidates (without any selection criteria)
- print content of DataStore as before
- print momentum of each pion, photon and invariant mass of each Kshort

```
[INFO] ==========================================================================
[INFO] DataStore collections in event 260010
[INFO] ==========================================================================
[INFO] Type                 Name                       #Entries          <Event>
[INFO] RelationContainer    ECLClustersToMCParticles
[INFO] EventMetaData        EventMetaData
[INFO] RelationContainer    KLMClustersToECLClusters
[INFO] RelationContainer    KLMClustersToMCParticles
[INFO] ParticleList         K_S0:all
[INFO] ParticleExtraInfoMap  ParticleExtraInfoMap
[INFO] RelationContainer    ParticlesToMCParticles
[INFO] RelationContainer    ParticlesToPIDLikelihoods
[INFO] RelationContainer    TracksToECLClusters
[INFO] RelationContainer    TracksToKLMClusters
[INFO] RelationContainer    TracksToMCParticles
[INFO] RelationContainer    TracksToPIDLikelihoods
[INFO] ParticleList         gamma:all
[INFO] ParticleList         pi+:all
[INFO] ParticleList         pi-:all
[INFO] ECLCluster[]         ECLClusters                39
[INFO] KLMCluster[]         KLMClusters                2
[INFO] MCParticle[]         MCParticles                107
[INFO] PIDLikelihood[]      PIDLikelihoods             17
[INFO] Particle[]           Particles                  59
[INFO] TrackFitResult[]     TrackFitResults            37
[INFO] Track[]              Tracks                     17
[INFO] V0[]                 V0s                        10
[INFO]
[INFO] --------------------------------------------------------------------------
[INFO] Type                 Name                       #Entries     <Persistent>
[INFO] FileMetaData         FileMetaData
[INFO] ProcessStatistics    ProcessStatistics
[INFO] BackgroundInfo[]     BackgroundInfos            1
[INFO]
[INFO] ==========================================================================
```

```
[INFO] ==========================================================================
[INFO] [ParticlePrinterModule] START ----------------------------
[INFO]  ParticleLists: pi+:all (7+0) + pi-:all (10+0)
[INFO]  - 0 = 211[0]
[INFO]      o) p = 0.28879
[INFO]  - 4 = 211[1]
[INFO]      o) p = 0.546838
[INFO]  - 5 = 211[2]
[INFO]      o) p = 0.4153
[INFO]  - 6 = 211[3]
[INFO]      o) p = 0.39559
[INFO]  - 8 = 211[4]
[INFO]      o) p = 0.628753
[INFO]  - 9 = 211[5]
[INFO]      o) p = 0.39385
[INFO]  - 11 = 211[6]
[INFO]      o) p = 0.260749
[INFO]  - 1 = -211[7]
```

```
[INFO] [ParticlePrinterModule] START ------------------------------
[INFO]  ParticleList : gamma:all (0+24)
[INFO]  - 17 = 22[0]
[INFO]      o) E = 0.276833
[INFO]  - 18 = 22[1]
[INFO]      o) E = 0.19579
[INFO]  - 19 = 22[2]
[INFO]      o) E = 0.0225137
```

```
[INFO] [ParticlePrinterModule] START ------------------------------
[INFO]  ParticleList : K_S0:all (0+6)
[INFO]  - 43 = 310[0]
[INFO]     o) daughter indices =  41 42
[INFO]      o) M = 0.828873
```

Exercise 2

- Use steering file from Exercise 1
- Perform a cut (E > 0.3 GeV) on the photon particle list
- Copy charged pions with piid>0.1 from the existing list to a new one
- Print again the energy of photons and momentum of pions from the new list
- Compare the numbers of all Particles in DataStore (for the same event) from exercises 1 and 2

Exercise 2

- Use steering file from Exercise 1
- Perform a cut (E > 0.3 GeV) on the photon particle list
- Copy charged pions with piid>0.1 from the existing list to a new one
- Print again the energy of photons and momentum of pions from the new list
- Compare the numbers of all Particles in DataStore (for the same event) from exercises 1 and 2

```
[INFO] ===========================================================================
[INFO] DataStore collections in event 260010
[INFO] ===========================================================================
[INFO] Type                 Name                       #Entries           <Event>
[INFO] RelationContainer    ECLClustersToMCParticles
[INFO] EventMetaData        EventMetaData
[INFO] RelationContainer    KLMClustersToECLClusters
[INFO] RelationContainer    KLMClustersToMCParticles
[INFO] ParticleList         K_S0:all
[INFO] ParticleExtraInfoMap  ParticleExtraInfoMap
[INFO] RelationContainer    ParticlesToMCParticles
[INFO] RelationContainer    ParticlesToPIDLikelihoods
[INFO] RelationContainer    TracksToECLClusters
[INFO] RelationContainer    TracksToKLMClusters
[INFO] RelationContainer    TracksToMCParticles
[INFO] RelationContainer    TracksToPIDLikelihoods
[INFO] ParticleList         gamma:all
[INFO] ParticleList         pi+:all
[INFO] ParticleList         pi+:good
[INFO] ParticleList         pi-:all
[INFO] ParticleList         pi-:good
[INFO] ECLCluster[]         ECLClusters                39
[INFO] KLMCluster[]         KLMClusters                2
[INFO] MCParticle[]         MCParticles                107
[INFO] PIDLikelihood[]      PIDLikelihoods             17
[INFO] Particle[]           Particles                  59
[INFO] TrackFitResult[]     TrackFitResults            37
[INFO] Track[]              Tracks                     17
[INFO] V0[]                 V0s                        10
[INFO]
[INFO] ---------------------------------------------------------------------------
[INFO] Type                 Name                       #Entries      <Persistent>
[INFO] FileMetaData         FileMetaData
[INFO] ProcessStatistics    ProcessStatistics
[INFO] BackgroundInfo[]     BackgroundInfos            1
[INFO]
[INFO] ===========================================================================
```

```
[INFO] ===========================================================================
[INFO] [ParticlePrinterModule] START ------------------------------
[INFO]  ParticleLists: pi+:good (7+0) + pi-:good (9+0)
[INFO]  - 0 = 211[0]
[INFO]      o) p = 0.28879
[INFO]  - 4 = 211[1]
[INFO]      o) p = 0.546838
[INFO]  - 5 = 211[2]
[INFO]      o) p = 0.4153
[INFO]  - 6 = 211[3]
[INFO]      o) p = 0.39559
```

```
[INFO] [ParticlePrinterModule] END   ------------------------------
[INFO] [ParticlePrinterModule] START ------------------------------
[INFO]  ParticleList : gamma:all (0+2)
[INFO]  - 20 = 22[0]
[INFO]      o) E = 0.321549
[INFO]  - 25 = 22[1]
[INFO]      o) E = 0.395711
```

Exercise 2

- Use steering file from Exercise 1
- Perform a cut (E > 0.3 GeV) on the photon particle list
- Copy charged pions with piid>0.1 from the existing list to a new one
- Print again the energy of photons and momentum of pions from the new list
- Compare the numbers of all Particles in DataStore (for the same event) from exercises 1 and 2

Why is the number of all Particles in the event same as before?

```
[INFO] ===========================================================================
[INFO] DataStore collections in event 260010
[INFO] ===========================================================================
[INFO] Type                 Name                      #Entries            <Event>
[INFO] RelationContainer    ECLClustersToMCParticles
[INFO] EventMetaData        EventMetaData
[INFO] RelationContainer    KLMClustersToECLClusters
[INFO] RelationContainer    KLMClustersToMCParticles
[INFO] ParticleList         K_S0:all
[INFO] ParticleExtraInfoMap  ParticleExtraInfoMap
[INFO] RelationContainer    ParticlesToMCParticles
[INFO] RelationContainer    ParticlesToPIDLikelihoods
[INFO] RelationContainer    TracksToECLClusters
[INFO] RelationContainer    TracksToKLMClusters
[INFO] RelationContainer    TracksToMCParticles
[INFO] RelationContainer    TracksToPIDLikelihoods
[INFO] ParticleList         gamma:all
[INFO] ParticleList         pi+:all
[INFO] ParticleList         pi+:good
[INFO] ParticleList         pi-:all
[INFO] ParticleList         pi-:good
[INFO] ECLCluster[]         ECLClusters               39
[INFO] KLMCluster[]         KLMClusters               2
[INFO] MCParticle[]         MCParticles               107
[INFO] PIDLikelihood[]      PIDLikelihoods            17
[INFO] Particle[]           Particles                 59
[INFO] TrackFitResult[]     TrackFitResults           37
[INFO] Track[]              Tracks                    17
[INFO] V0[]                 V0s                       10
[INFO]
[INFO] ---------------------------------------------------------------------------
[INFO] Type                 Name                      #Entries       <Persistent>
[INFO] FileMetaData         FileMetaData
[INFO] ProcessStatistics    ProcessStatistics
[INFO] BackgroundInfo[]     BackgroundInfos           1
[INFO]
[INFO] ===========================================================================
```

```
[INFO] ===========================================================================
[INFO] [ParticlePrinterModule] START ------------------------------
[INFO]  ParticleLists: pi+:good (7+0) + pi-:good (9+0)
[INFO]  - 0 = 211[0]
[INFO]     o) p = 0.28879
[INFO]  - 4 = 211[1]
[INFO]     o) p = 0.546838
[INFO]  - 5 = 211[2]
[INFO]     o) p = 0.4153
[INFO]  - 6 = 211[3]
[INFO]     o) p = 0.39559
```

```
[INFO] [ParticlePrinterModule] END   ------------------------------
[INFO] [ParticlePrinterModule] START ------------------------------
[INFO]  ParticleList : gamma:all (0+2)
[INFO]  - 20 = 22[0]
[INFO]     o) E = 0.321549
[INFO]  - 25 = 22[1]
[INFO]     o) E = 0.395711
```

Exercise 3

- Use steering file from Exercise 2
- Reconstruct D0 -> pi+ pi- candidates using the new pion list
- Print the invariant mass of all D0 candidates
- Compare the numbers of all Particles in DataStore (for the same event) from exercises 2 and 3
  - Is the difference equal to the number of D0 candidates?

Exercise 3

- Use steering file from Exercise 2
- Reconstruct D0 -> pi+ pi- candidates using the new pion list
- Print the invariant mass of all D0 candidates
- Compare the numbers of all Particles in DataStore (for the same event) from exercises 2 and 3
  - Is the difference equal to the number of D0 candidates?

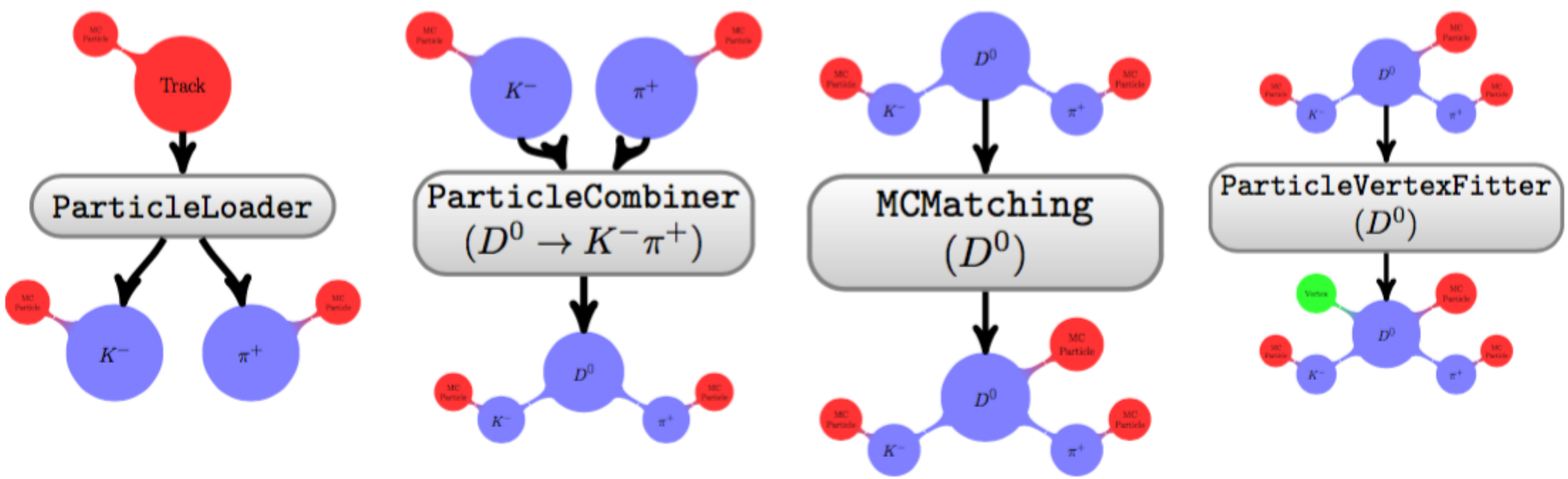
```
[INFO] ==========================================================================
[INFO] DataStore collections in event 260010
[INFO] ==========================================================================
[INFO] Type                 Name                       #Entries           <Event>
[INFO] ParticleList         D0:all
[INFO] RelationContainer    ECLClustersToMCParticles
[INFO] EventMetaData        EventMetaData
[INFO] RelationContainer    KLMClustersToECLClusters
[INFO] RelationContainer    KLMClustersToMCParticles
[INFO] ParticleList         K_S0:all
[INFO] ParticleExtraInfoMap  ParticleExtraInfoMap
[INFO] RelationContainer    ParticlesToMCParticles
[INFO] RelationContainer    ParticlesToPIDLikelihoods
[INFO] RelationContainer    TracksToECLClusters
[INFO] RelationContainer    TracksToKLMClusters
[INFO] RelationContainer    TracksToMCParticles
[INFO] RelationContainer    TracksToPIDLikelihoods
[INFO] ParticleList         anti-D0:all
[INFO] ParticleList         gamma:all
[INFO] ParticleList         pi+:all
[INFO] ParticleList         pi+:good
[INFO] ParticleList         pi-:all
[INFO] ParticleList         pi-:good
[INFO] ECLCluster[]         ECLClusters                39
[INFO] KLMCluster[]         KLMClusters                2
[INFO] MCParticle[]         MCParticles                107
[INFO] PIDLikelihood[]      PIDLikelihoods             17
[INFO] Particle[]           Particles                  122
[INFO] TrackFitResult[]     TrackFitResults            37
[INFO] Track[]              Tracks                     17
[INFO] V0[]                 V0s                        10
[INFO]
[INFO] --------------------------------------------------------------------------
[INFO] Type                 Name                       #Entries      <Persistent>
[INFO] FileMetaData         FileMetaData
[INFO] ProcessStatistics    ProcessStatistics
[INFO] BackgroundInfo[]     BackgroundInfos            1
[INFO]
[INFO] ==========================================================================
```

```
[INFO] ==========================================================================
[INFO] [ParticlePrinterModule] START ------------------------------
[INFO]  ParticleLists: D0:all (0+63) + anti-D0:all (0+63)
[INFO]  - 59 = 421[0]
[INFO]     o) daughter indices =  0 1
[INFO]      o) M = 0.497041
[INFO]  - 60 = 421[1]
[INFO]     o) daughter indices =  4 1
[INFO]      o) M = 0.942033
[INFO]  - 61 = 421[2]
[INFO]     o) daughter indices =  5 1
[INFO]      o) M = 0.814606
[INFO]  - 62 = 421[3]
[INFO]     o) daughter indices =  6 1
[INFO]      o) M = 0.566222
[INFO]  - 63 = 421[4]
[INFO]     o) daughter indices =  8 1
[INFO]      o) M = 0.880142
[INFO]  - 64 = 421[5]
[INFO]     o) daughter indices =  9 1
[INFO]      o) M = 0.731587
[INFO]  - 65 = 421[6]
[INFO]     o) daughter indices =  11 1
[INFO]      o) M = 0.430149
[INFO]  - 66 = 421[7]
[INFO]     o) daughter indices =  0 2
[INFO]      o) M = 0.329027
```

Analysis modules

BASF2 analysis module performs a single well defined action

- makes combinations, performs vertex fits, performs mc matching, calculates continuum suppression variables, …
- *each module usually creates a new Particle or other data object or modifies the existing one*



User doesn't need to write C++ code, but provides instead a BASF2 python steering file where he specifies the action to be performed on given input ParticleList(s)

Analysis modules

Modules

Module	Short description
`ParticleLoader`	Loads MDST dataobjects as Particle objects and collects them in specified ParticleList
`ParticleCombiner`	Creates particle combinations
`ParticleVertexFitter`	Performs kinematic fits
`ParticleListManipulator`	Manipulates ParticleLists: copies/merges/performs particle selection
`ParticleSelector`	Removes Particles from given ParticleList that do not pass specified selection criteria
`ParticlePrinter`	Prints specified variables for all particles (or event based variables) in the specified ParticleList to the screen (useful for debugging)
`ParticleStats`	Print out summary for specific ParticleList at the end of the job (retention, multiplicity, ...)
`MCMatcherParticles`	Performs MC truth matching (sets relation Particle->MCParticle) for all particles (and its (grand)-daughter particles) in the ParticleList
`ParticleMCDecayString`	Adds the Monte Carlo decay string to a Particle (useful for identification of generated decay chain)
`TMVAExpert`	Adds an TMVAExpert output as an ExtraInfo to the Particle objects in given ParticleLists. Requires TMVATeacher to be executed first.
`TMVATeacher`	Trains TMVA method with given particle lists as training samples
`NtupleMaker`	Creates and fills flat ntuples with user-specified set of ntuple tools or variables
`VariablesToHistogram`	alculate variables specified by the user for a given ParticleList and save them into a TH1F
`BestCandidateSelection`	Selects best Particles or ranks them in the ParticleList according to the values of any user-specified `variable`
`RemoveParticlesNotInLists`	Removes all Particles that are not in one of the given ParticleLists (or daughters of Particles in the lists)
`SkimFilter`	Filter based on ParticleLists, by setting return value to true if at least one of the given lists is not empty
`TagVertex`	Tag side Vertex Fitter
`ContinuumSuppression`	Creates for each Particle in the given ParticleLists a ContinuumSuppression dataobject and makes BASF2 relation between them
`RestOfEventBuilder`	Creates for each Particle in the given ParticleList a RestOfEvent dataobject and makes BASF2 relation between them
`RestOfEventInterpreter`	Creates a mask for tracks and clusters in the RestOfEvent
`MCDecayFinder`	Find decays in MCParticle list matching a given DecayString and create Particles from them
`VariableToReturnValue`	Calculate event-based variable specified by the user and sets return value of the module accordingly
`VariablesToExtraInfo`	For each particle in the input list the selected variables are saved in an extra-info field with the given name
`ParticleCopier`	Replaces each Particle in the ParticleList with its copy

splay a menu

Analysis modules

Modules

Module	Short description
ParticleLoader	Loads MDST dataobjects as Particle objects and collects them in specified ParticleList
ParticleCombiner	Creates particle combinations
ParticleVertexFitter	
ParticleListManipulat	
ParticleSelector	
ParticlePrinter	ugging)
ParticleStats	
MCMatcherParticles	articleList
ParticleMCDecayStrinc	
TMVAExpert	cuted first.
TMVATeacher	
NtupleMaker	
VariablesToHistogram	
BestCandidateSelectic	
RemoveParticlesNotInI	
SkimFilter	
TagVertex	
ContinuumSuppression	n them
RestOfEventBuilder	
RestOfEventInterprete	
MCDecayFinder	Find decays in MCParticle list matching a given DecayString and create Particles from them
VariableToReturnValue	Calculate event-based variable specified by the user and sets return value of the module accordingly
VariablesToExtraInfo	For each particle in the input list the selected variables are saved in an extra-info field with the given name
ParticleCopier	Replaces each Particle in the ParticleList with its copy

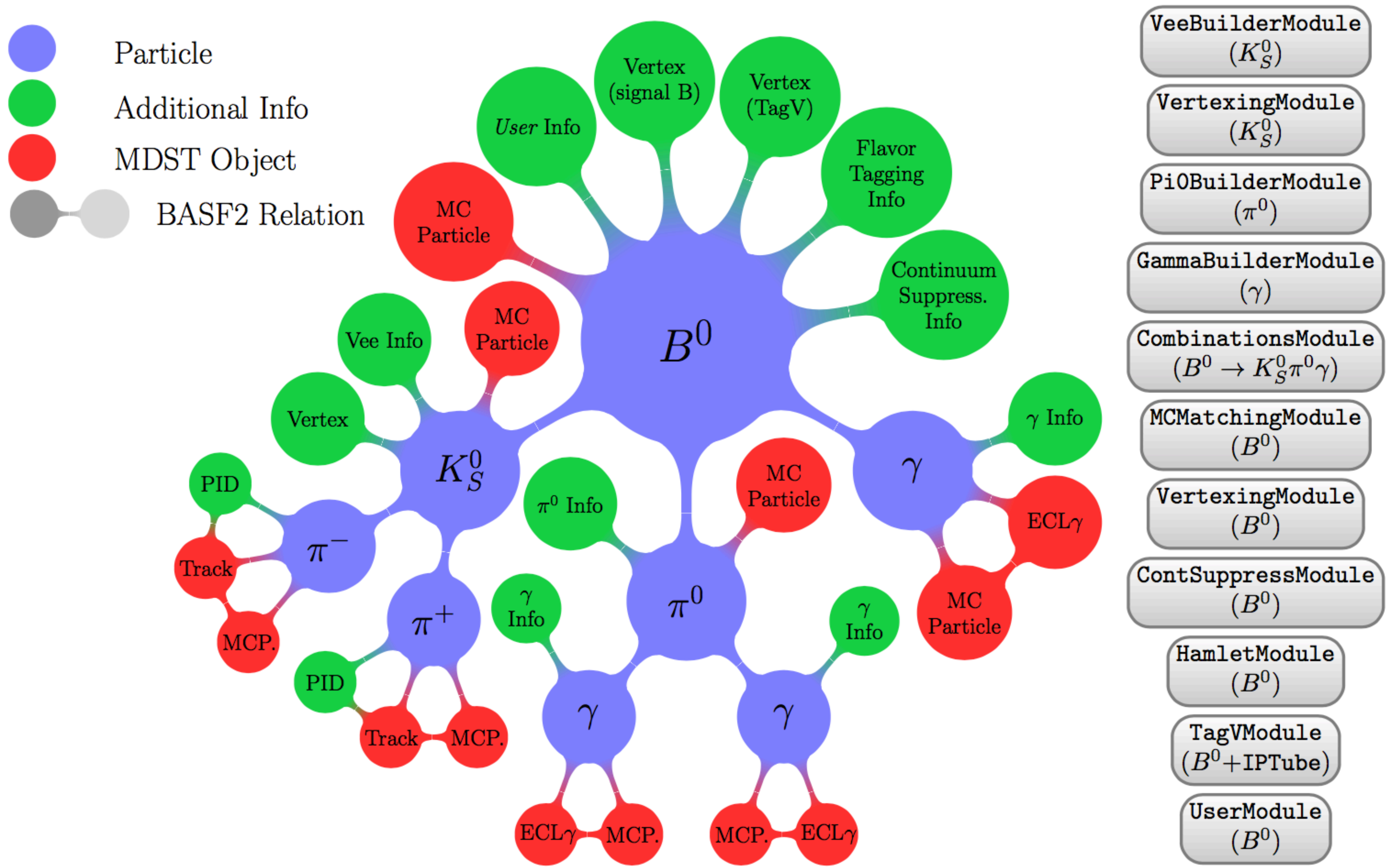
splay a menu

- Core analysis modules are developed
- *All analysis modules perform their action on charged-conjugated list/process automatically*

1. reconstructDecay('D0:myD0 -> K-:tight pi+:all pi0:loose','')
   # creates anti-D0 -> K+ pi- pi0 candidates as well and stores them to ***'anti-D0:myD0'*** *ParticleList*

2. vertexKFit('D0:myD0')
   # performs vertex fits on all candidates in the charge-conjugated list ('anti-D0:myD0') as well

Example Analysis Sequence and Data Structure: $B^0 \to K^*(K_S^0\pi^0)\gamma$



Decay String

DecayString

The DecayString is an elegant way of telling the analysis modules about the structure and the particles of a decay tree.

It has to be stressed that the decay string purely specifies a decay tree. It does not contain any physical interpretation of the decay, for example if it is allowed or not.

Examples:

- *specifying the decay to be reconstructed by the* **ParticleCombiner**

reconstructDecay('D0:myD0 -> K-:tight pi+:all pi0:loose','')

DecayString that tells ParticleCombiner module to create 'D0:myD0' ParticleList and fill it with D0 particle candidates created by making all combinations of K- candidates from 'K-:tight', pi+ candidates form 'pi+:all' and pi0 candidates from 'pi0:loose' ParticleList

Decay String

DecayString

The DecayString is an elegant way of telling the analysis modules about the structure and the particles of a decay tree.
It has to be stressed that the decay string purely specifies a decay tree. It does not contain any physical interpretation of the decay, for example if it is allowed or not.

Examples:

- *selecting a Particle in the decay chain for which we wish to save particular Variable or nTupleTool*

```
tools += ['PID', 'D*+ -> [D0 -> ^K- ^pi+] ^pi+']
```

DecayString that tells NtupleMaker module to create PID-related branches only for charged kaons and pions in this D+ -> D0 pi+; D0 -> K- pi+; decay chain. (PID-related quantities are only defined for charged FSPs.)*

```
^ – is used to “select” particle in the DecayString
```

Variable Manager

[Variables and VariableManager]

- *VariableManager* library is the central place in the analysis package for calculations of various simple or derived quantities needed for
  - *performing (on-line) selection*
  - *flat ntuple production for offline analysis*
- All *Variables* registered to the *VariableManager* can be used inside C++, and as parameters for analysis modules in python steering scripts

Examples:

- *specifying the* **ParticleCombiner** *to keep only B candidates with*

$$M_{\mathrm{bc}} > 5.2 \text{ GeV and } |\Delta E| < 0.2 \text{ GeV}$$

```
reconstructDecay('B+ -> anti-D0:myD0 pi+:all','Mbc > 5.2 and abs(deltaE) < 0.2')
```

Variable Manager

[Variables and VariableManager]

- V... ...kage for
  c...
- A... ...side C++, and
  a...

Exa...

r... ...bs(deltaE) < 0.2')

Kinematics

p	momentum magnitude
E	energy
px	momentum component x
py	momentum component y
pz	momentum component z
pt	transverse momentum
cosTheta	momentum cosine of polar angle
cth	momentum cosine of polar angle
phi	momentum azimuthal angle in degrees
p_CMS	CMS momentum magnitude
E_CMS	CMS energy
px_CMS	CMS momentum component x
py_CMS	CMS momentum component y
pz_CMS	CMS momentum component z
pt_CMS	CMS transverse momentum
cosTheta_CMS	CMS momentum cosine of polar angle
cth_CMS	CMS momentum cosine of polar angle
phi_CMS	CMS momentum azimuthal angle in degrees
cosThetaBetweenParticleAndTrueB	cosine of angle between momentum the particle and a true B particle. Is somewhere between -1 and 1 if only a massless particle like a neutrino is missing in the reconstruction.
cosAngleBetweenMomentumAndVertexVector	cosine of angle between momentum and vertex vector (vector connecting ip and fitted vertex) of this particle
distance	distance relative to interaction point
significanceOfDistance	significance of distance relative to interaction point
dx	x in respect to IP
dy	y in respect to IP
dz	z in respect to IP
dr	transverse distance in respect to IP
M	invariant mass (determined from particle's 4-momentum vector)
dM	mass minus nominal mass
Q	released energy in decay
dQ	released energy in decay minus nominal one
Mbc	beam constrained mass
deltaE	energy difference
InvM	invariant mass (determined from particle's daughter 4-momentum vectors)
ErrM	uncertainty of invariant mass (determined from particle's daughter 4-momentum vectors)
SigM	signed deviation of particle's invariant mass from its nominal mass

List of all defined [Variables]

Executing basf2 variables.py' returns a complete list

Variable Manager

Variables and VariableManager

- V… c… …kage for
- A… a…

Exa…

r…

Kinematics

p	momentum magnitude
E	energy
px	momentum component x
py	momentum component y
pz	momentum component z
pt	transverse momentum
cosTheta	momentum cosine of polar angle
cth	mome…
phi	mome…
p_CMS	CMS…
E_CMS	CMS…
px_CMS	CMS…
py_CMS	CMS…
pz_CMS	CMS…
pt_CMS	CMS…
cosTheta_CMS	CMS…
cth_CMS	CMS…
phi_CMS	CMS…
cosThetaBetweenPart…	like…
cosAngleBetweenMome…	
distance	dist…
significanceOfDista…	
dx	x in…
dy	y in…
dz	z in…
dr	tran…
M	inva…
dM	mass…
Q	rele…
dQ	rele…
Mbc	beam…
deltaE	ener…
InvM	inva…
ErrM	unce…
SigM	sign…

List of all defined Variables

- *Variable* types
  - *simple* (no additional arguments/parameters)
    - *p, px, DLLKaon, Mbc, M, dM, isSignal, R2, …*
  - *parameter* (require additional arguments/parameters)
    - *daughterInvariantMass(0,1)* = invariant mass of first and second daughter
    - *massDifference(0)* = difference between invariant masses of this and particle and first daughter
  - *meta* (‘combinations’ of functions/variables)
    - *daughter(1, M)* = invariant Mass of the first daughter
    - *abs(deltaE)* = absolute value of DeltaE
    - *extraInfo(signalProbability)* = signalProbability stored as extraInfo
    - *formula(v1 + v2 * v3 - v4 / v5^v6)* = simple formulas of any variables

Variables

Variable aliases

Variable Aliases:

- sometimes variable names can get very long
    - define alias instead!

```
from variables import variables
variables.addAlias('sigProb', 'extraInfo(SignalProbability)')
variables.addAlias('rank', 'extraInfo(sigProb_rank)')
variables.addAlias('dmID', 'extraInfo(decayModeID)')
variables.addAlias('uniqueSignal', 'extraInfo(uniqueSignal)')
variables.addAlias('d0_dmID', 'daughter(0,extraInfo(decayModeID))')
variables.addAlias('d1_dmID', 'daughter(1,extraInfo(decayModeID))')
variables.addAlias('d0_d0_dmID', 'daughter(0,daughter(0,extraInfo(decayModeID)))')
variables.addAlias('d1_d0_dmID', 'daughter(1,daughter(0,extraInfo(decayModeID)))')
variables.addAlias('d1d2_M', 'daughterInvariantMass(1,2)')
variables.addAlias('d1d3_M', 'daughterInvariantMass(1,3)')
variables.addAlias('d1d4_M', 'daughterInvariantMass(1,4)')
variables.addAlias('d2d3_M', 'daughterInvariantMass(2,3)')
variables.addAlias('d2d4_M', 'daughterInvariantMass(2,4)')
variables.addAlias('d3d4_M', 'daughterInvariantMass(3,4)')
variables.addAlias('d1d2d3_M', 'daughterInvariantMass(1,2,3)')
variables.addAlias('d2d3d4_M', 'daughterInvariantMass(2,3,4)')
variables.addAlias('d1d2d3d4_M', 'daughterInvariantMass(1,2,3,4)')
```

```
# apply some very loose cuts to reduce the number
# of Btag candidates
applyCuts('B0:generic', 'Mbc>5.24 and abs(deltaE)<0.200 and sigProb>0.001')
applyCuts('B+:generic', 'Mbc>5.24 and abs(deltaE)<0.200 and sigProb>0.001')
```

Exercise 4

- Create new steering file
- Reconstruct B -> J/psi K0s; J/psi -> ee and mumu;
- Require that at least one of the leptons from J/psi is positively identified as an electron or muon (e.g. by requiring that eid>0.1 or muid>0.1)
- Run over MC7 signal MC (link)
  - include one file for J/psi->ee (B->JpsiKL MC; sorry no B->JpsiKS MC available)
  - include one file for J/psi -> mumu (B->J/psiKs MC)
- print summary of B0 list at the end

Exercise 4

- Create new steering file
- Reconstruct B -> J/psi K0s; J/psi -> ee and mumu;
- Require that at least one of the leptons from J/psi is positively identified as an electron or muon (e.g. by requiring that eid>0.1 or muid>0.1)
- Run over MC7 signal MC (link)
    - include one file for J/psi->ee (B->JpsiKL MC; sorry no B->JpsiKS MC available)
    - include one file for J/psi -> mumu (B->J/psiKs MC)
- print summary of B0 list at the end

```
[INFO] ParticleStats Summary:

========================================================================
                        |Retention|          0| Unique
    B0:jpsiks( 0)       |       0.2152  1.0000  1.0000

=======================================================================
 Average Candidate Multiplicity (ACM) and ACM for Passed Events (ACMPE)
                        | All Particles                | Particles                    | AntiParticles
                        | ACM           | ACMPE        | ACM           | ACMPE        | ACM           | ACMPE
    B0:jpsiks( 0)       |        0.2575        1.1966        0.0000        0.0000        0.0000        0.0000        0.2575        1.1966

=======================================================
Total Retention: 0.2152
Total Number of Particles created in the DataStore: 5200267
=======================================================
```

Flat nTuples

How to create ntuple

NTupleMaker

NTupleMaker creates and fills flat ntuples with user specified content for offline analysis. The content of the ntuple can be defined in two ways:

1. Using nTupleTools with predefined content

```
toolsB  = ['EventMetaData', '^B-']
toolsB += ['Kinematics',    '^B- -> [^D0 -> ^K- ^pi+] ^pi-']
toolsB += ['DeltaEMbc',     '^B-']
toolsB += ['PID',           'B- -> [D0 -> ^K- ^pi+] ^pi-']
toolsB += ['MassBeforeFit', 'B- -> ^D0 pi-']
toolsB += ['MCTruth',       '^B- -> ^D0 ^pi-']
```

List of tools

2. Using CustomFloats nTupleTool with user defined content

```
toolsGamma = ['CustomFloats[px:py:pz:goodGamma]', '^gamma']
ntupleFile('gamma-output.root')
ntupleTree('gammaTree', 'gamma:all', toolsGamma)
```

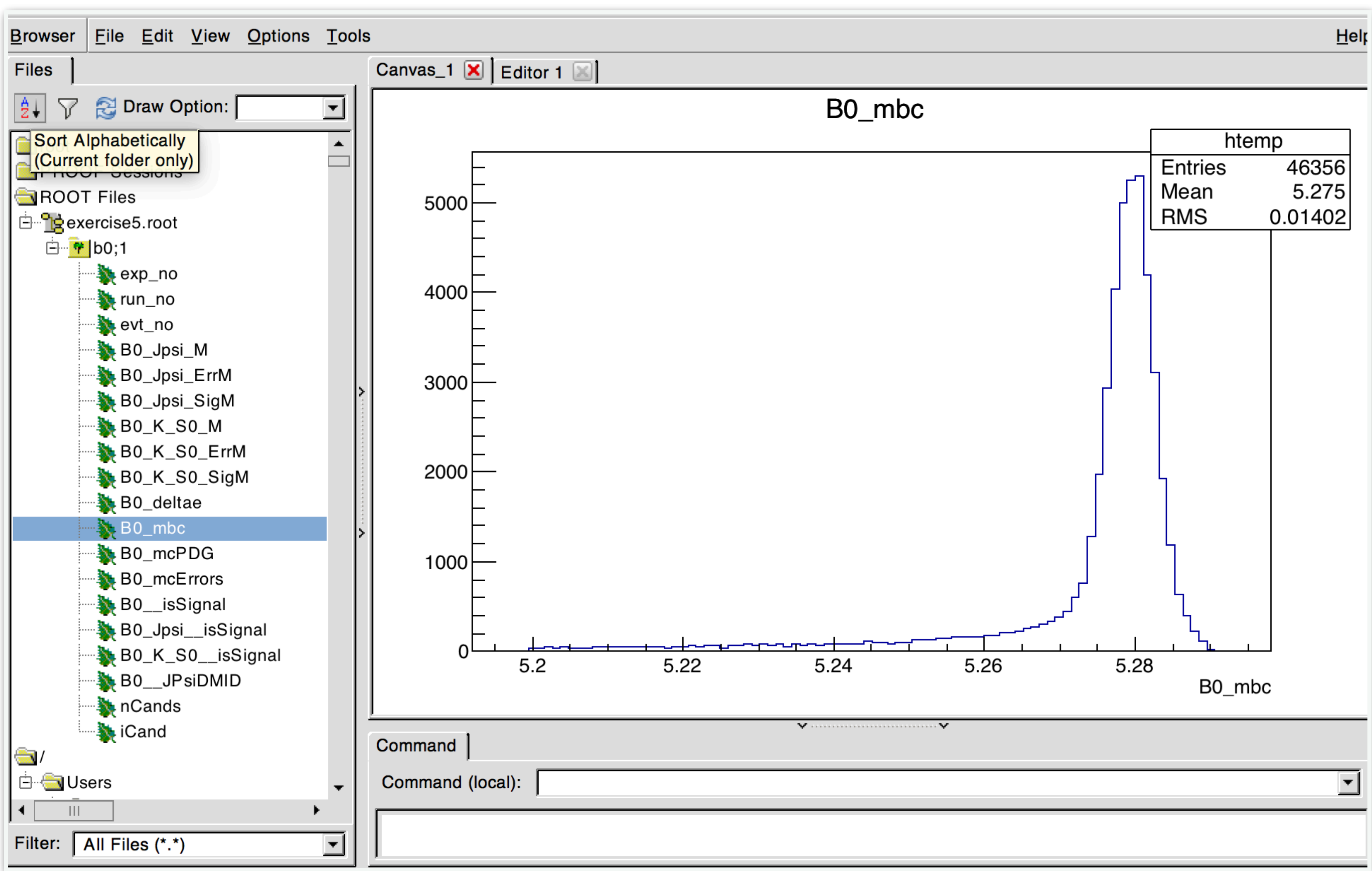
all Variables (meta, alias, parameter, …), separated with ':'

Exercise 5

- Use the steering file from exercise 4
- Create ntuple and fill it with:
  - DeltaE and Mbc of B candidates
  - J/psi and KShort invariant masses
  - J/psi decay mode ID (define alias!)
  - eid and muid of J/psi daughters

Exercise 5

- Use the steering file from exercise 4
- Create ntuple and fill it with:
  - DeltaE and Mbc of B candidates
  - J/psi and KShort invariant masses
  - J/psi decay mode ID (define alias!)
  - eid and muid of J/psi daughters



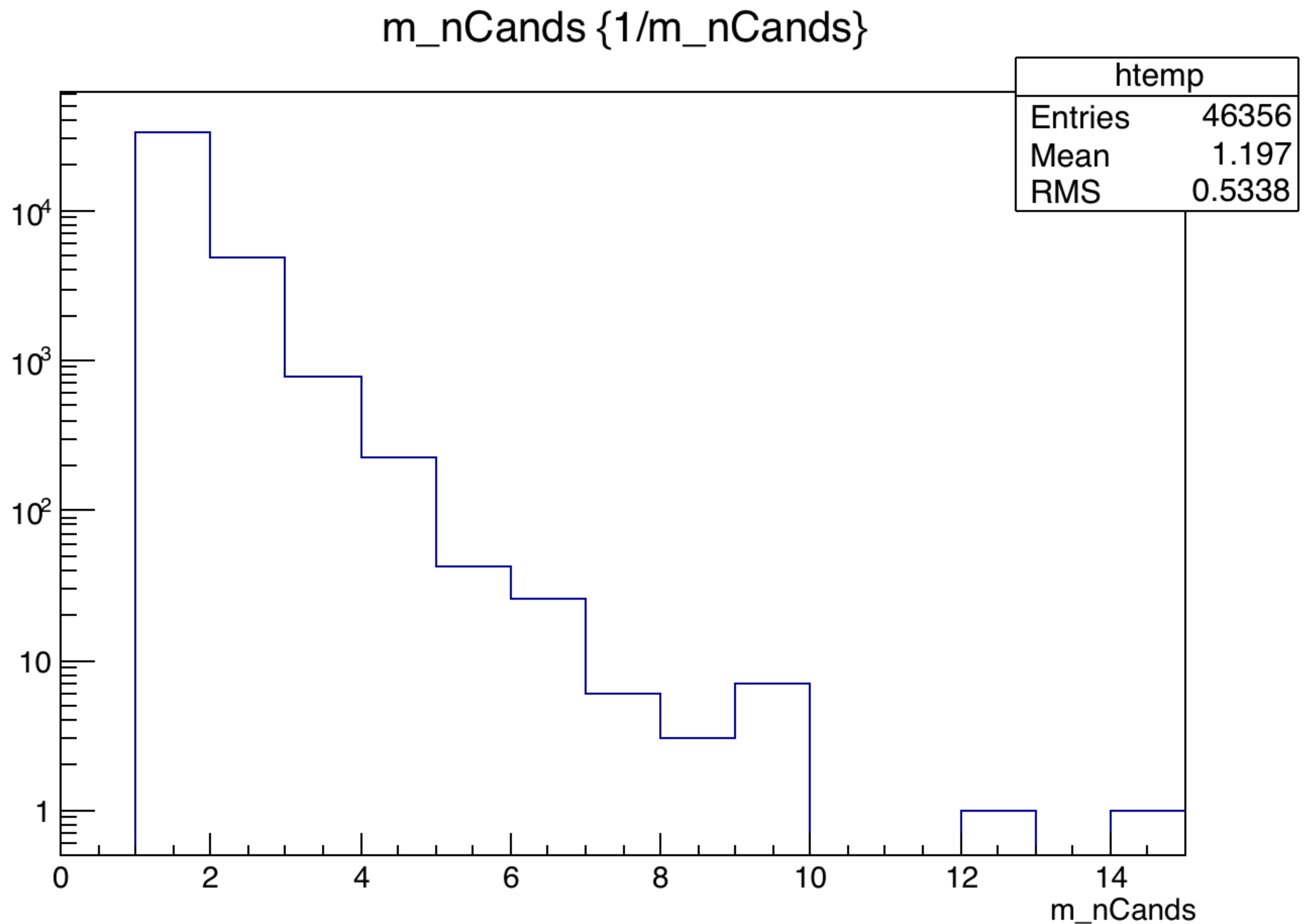
Exercise 6

- Plot number of candidates per event

```
b0->Draw("m_nCands","1/m_nCands")
```

Exercise 6

- Plot number of candidates per event



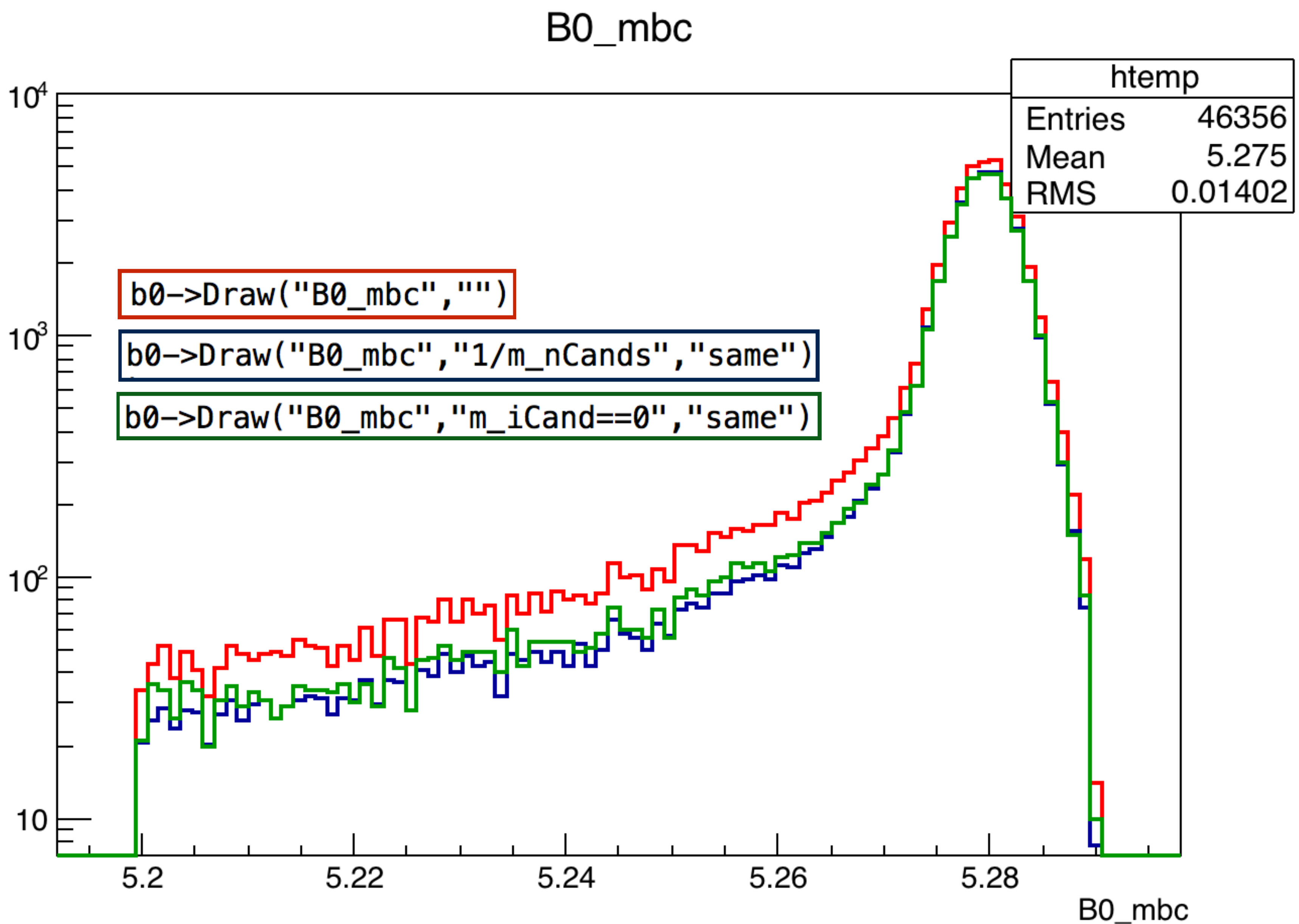

```
b0->Draw("m_nCands","1/m_nCands")
```

Exercise 6

- Can I draw only one candidate per event?
  - At this stage (no best candidate selection yet) there are two possibilities:
    1. Draw only 1st candidate in an event
    2. Draw all candidates in an event but weighted with 1/nCandidates

Exercise 6

- Can I draw only one candidate per event?
  - At this stage (no best candidate selection yet) there are two possibilities:
    1. Draw only 1st candidate in an event
    2. Draw all candidates in an event but weighted with 1/nCandidates



Exercise 6

- What is the source of multiple candidates?

Exercise 6

- What is the source of multiple candidates?

```
root [6] b0->Scan("m_nCands:B0_mbc:B0_deltae:B0_Jpsi_M:B0_K_S0_M:B0__JPsiDMID:B0__isSignal","m_nCands>1")
************************************************************************************************
*    Row   * m_nCands *    B0_mbc * B0_deltae * B0_Jpsi_M * B0_K_S0_M * B0__JPsiD * B0__isSig *
************************************************************************************************
*       11 *        5 * 5.2660489 * -0.168615 * 3.1063382 * 0.4541613 *         2 *         0 *
*       12 *        5 * 5.2550258 * -0.239294 * 2.8623807 * 0.6721092 *         2 *         0 *
*       13 *        5 * 5.2152891 * 0.1349816 * 3.1063382 * 1.4950423 *         2 *         0 *
*       14 *        5 * 5.2151985 * 0.1340994 * 2.8623807 * 0.5146154 *         2 *         0 *
*       15 *        5 * 5.2811255 * 0.0018234 * 3.1063382 * 0.4702372 *         2 *         1 *
*       18 *        2 * 5.2327518 * -0.101432 * 3.0929150 * 0.5148170 *         2 *         0 *
*       19 *        2 * 5.2788667 * -0.003403 * 3.0929150 * 0.5092093 *         2 *         1 *
*       32 *        2 * 5.2650704 * -0.194976 * 3.1009950 * 0.4528838 *         2 *         0 *
*       33 *        2 * 5.2871685 * -0.058925 * 3.1009950 * 0.4934948 *         2 *         0 *
*       34 *        2 * 5.2835822 * 0.0140079 * 3.0937235 * 0.4813628 *         2 *         1 *
*       35 *        2 * 5.2800521 * 0.0014306 * 3.0937235 * 0.8446744 *         2 *         0 *
*       37 *        2 * 5.2690300 * -0.098116 * 3.0782930 * 0.7704534 *         2 *         0 *
*       38 *        2 * 5.2774133 * -0.050596 * 3.0782930 * 0.5785743 *         2 *         1 *
*       44 *        2 * 5.2722535 * -0.188015 * 3.0939481 * 0.5695912 *         2 *         0 *
*       45 *        2 * 5.2782278 * -0.002700 * 3.0939481 * 0.5244799 *         2 *         1 *
*       48 *        3 * 5.2393484 * 0.1429377 * 3.1177580 * 0.4997866 *         2 *         0 *
*       49 *        3 * 5.2653894 * -0.135387 * 3.1177580 * 0.4327980 *         2 *         0 *
*       50 *        3 * 5.2788553 * 0.0169916 * 3.1177580 * 0.4446834 *         2 *         1 *
*       54 *        3 * 5.2617774 * -0.235137 * 3.1487865 * 0.6366187 *         2 *         0 *
*       55 *        3 * 5.2694711 * -0.235613 * 3.0929985 * 0.5894577 *         2 *         0 *
*       56 *        3 * 5.2736740 * -0.006364 * 3.0929985 * 0.4853006 *         2 *         1 *
*       59 *        2 * 5.2130451 * -0.199016 * 3.0543851 * 0.8891012 *         2 *         0 *
*       60 *        2 * 5.2749471 * -0.035054 * 3.0543851 * 0.3657057 *         2 *         1 *
*       69 *        2 * 5.2626767 * -0.035757 * 3.0966794 * 0.3643627 *         2 *         0 *
*       70 *        2 * 5.2754530 * 0.0037377 * 3.0966794 * 0.5315880 *         2 *         1 *
Type <CR> to continue or q to quit ==>
```

Exercise 7

- Use the steering file from exercise 5
- Sort candidates (ascending order) according to

$$\left|\frac{m_{\ell\ell} - M_{J/\psi}}{\sigma(m_{\ell\ell})}\right| + \left|\frac{m_{\pi\pi} - M_{K^0_S}}{\sigma(m_{\pi\pi})}\right|$$

- add rank to ntuple
- print above quantities on ~100 events to better understand them before running over all events

Hint: check ‘SigM’ variable

Exercise 7

- Use the steering file from exercise 5
- Sort candidates (ascending order) according to

$$\left|\frac{m_{\ell\ell} - M_{J/\psi}}{\sigma(m_{\ell\ell})}\right| + \left|\frac{m_{\pi\pi} - M_{K_S^0}}{\sigma(m_{\pi\pi})}\right|$$

- add rank to ntuple
- print above quantities on ~100 events to better understand them before running over all events

Hint: check ‘SigM’ variable

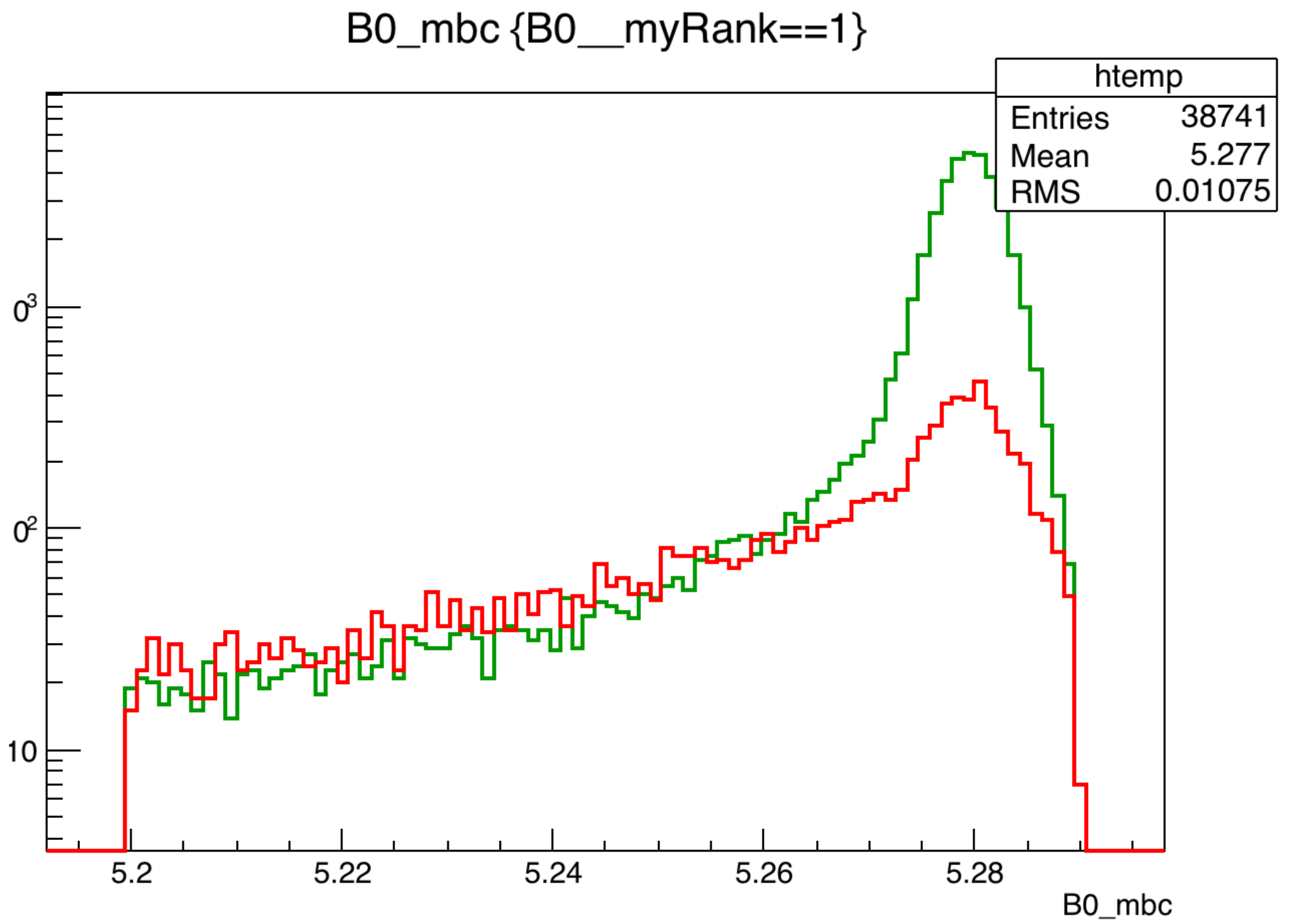
```
➜  Downloads root -l exercise7.root
root [0]
Attaching file exercise7.root as _file0...
root [1] b0->Show()
======> EVENT:-1
 exp_no           = 0
 run_no           = 0
 evt_no           = 0
 B0_Jpsi_M        = 0
 B0_Jpsi_ErrM     = 0
 B0_Jpsi_SigM     = 0
 B0_K_S0_M        = 0
 B0_K_S0_ErrM     = 0
 B0_K_S0_SigM     = 0
 B0_deltae        = 0
 B0_mbc           = 0
 B0_mcPDG         = 0
 B0_mcErrors      = 0
 B0__isSignal     = 0
 B0_Jpsi__isSignal = 0
 B0_K_S0__isSignal = 0
 B0__JPsiDMID     = 0
 B0__myRank       = 0
 m_nCands         = 0
 m_iCand          = 0
```

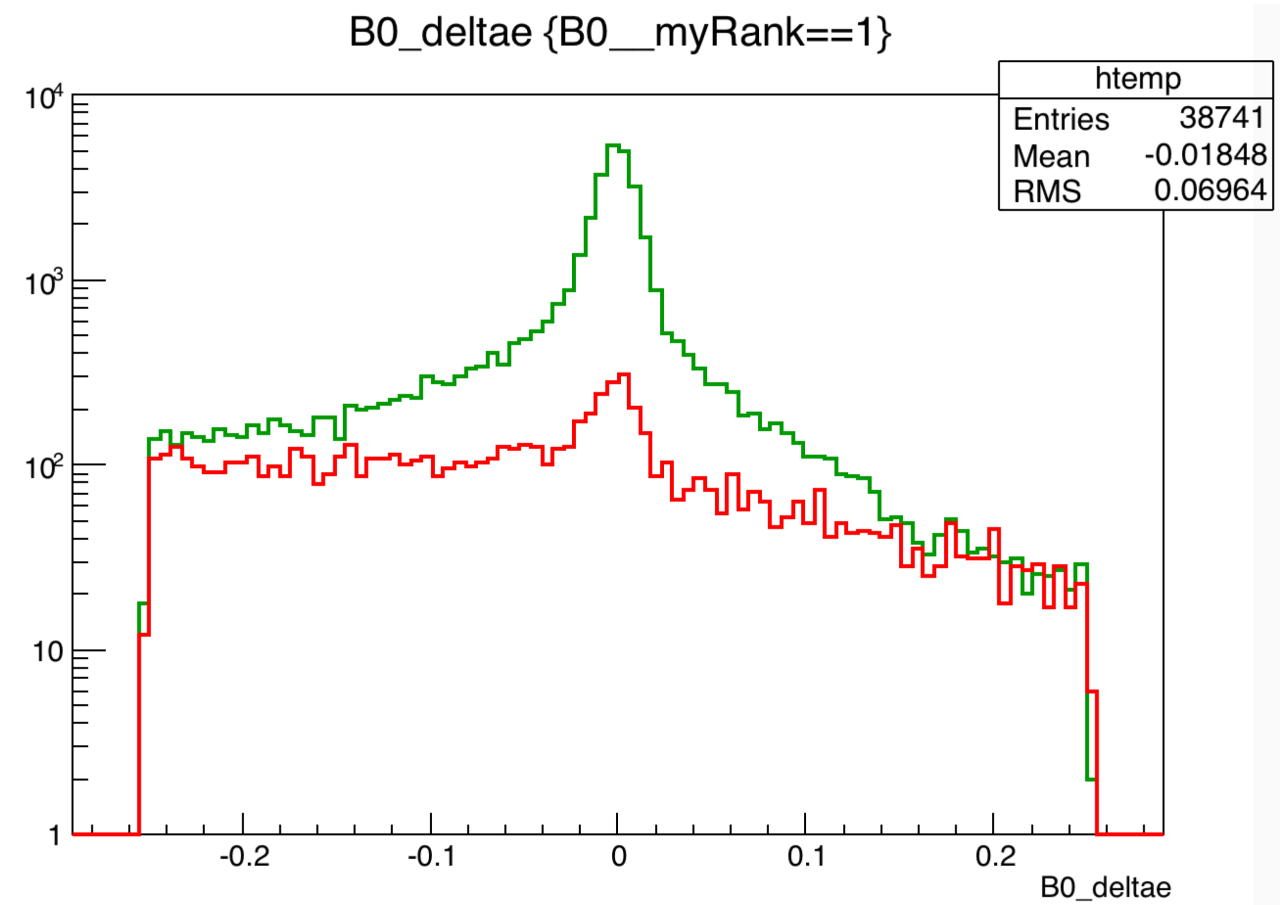
Exercise 8

- Draw the best (and the rest) candidate(s)

Exercise 8

- Draw the best (and the rest) candidate(s)






```
b0->Draw("B0_mbc","B0__myRank==1","")
```

```
b0->Draw("B0_mbc","B0__myRank!=1","same")
```

```
b0->Draw("B0_deltae","B0__myRank==1")
```

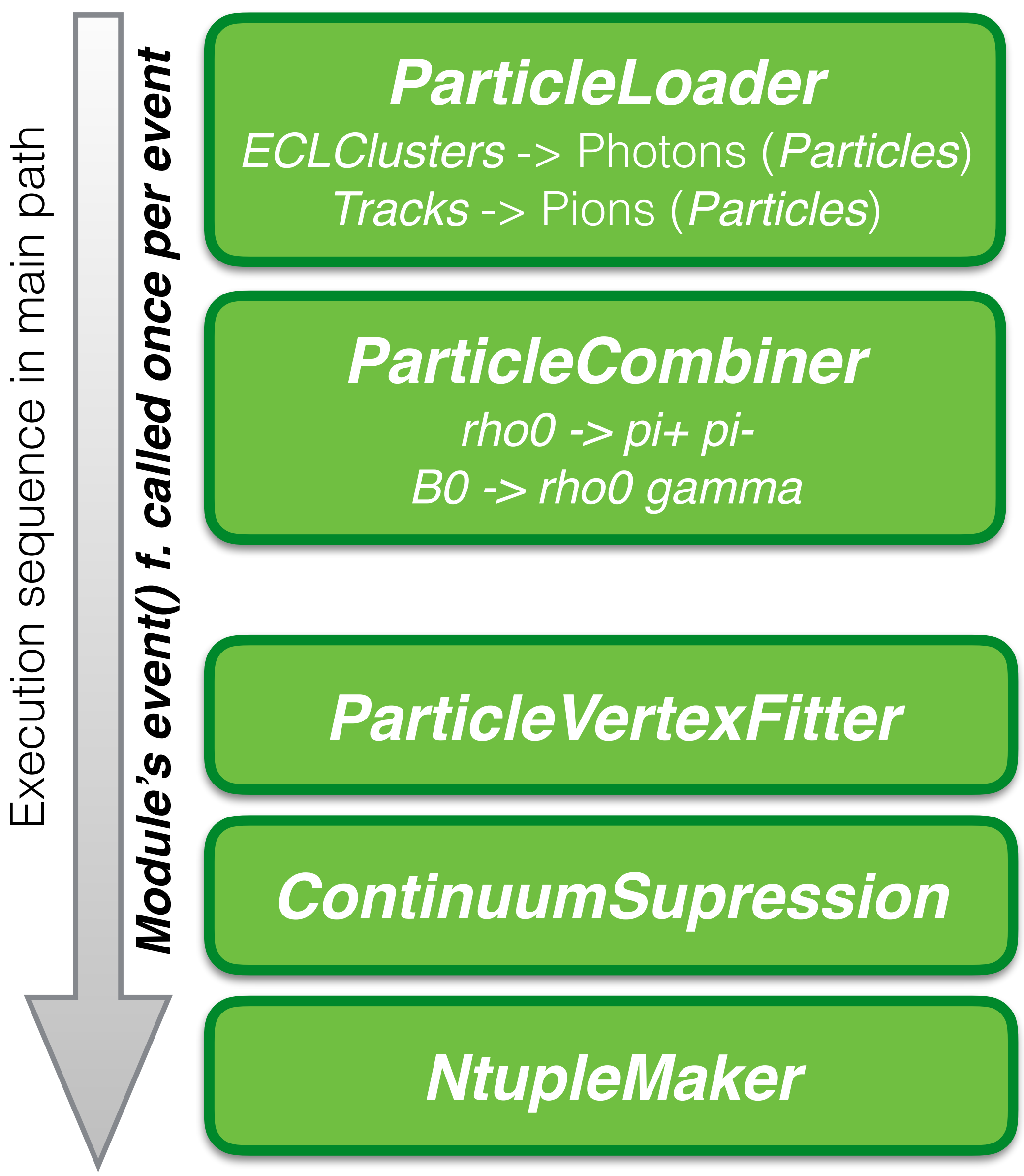
```
b0->Draw("B0_deltae","B0__myRank!=1","same")
```

Analysis modules and execution path

Normally, a module's ***event()*** function is called once per event while any repeated processing is performed inside this call by e.g. looping over an array of input data.

Example:

- *reconstruction of **B0 -> rho0 gamma** decay*

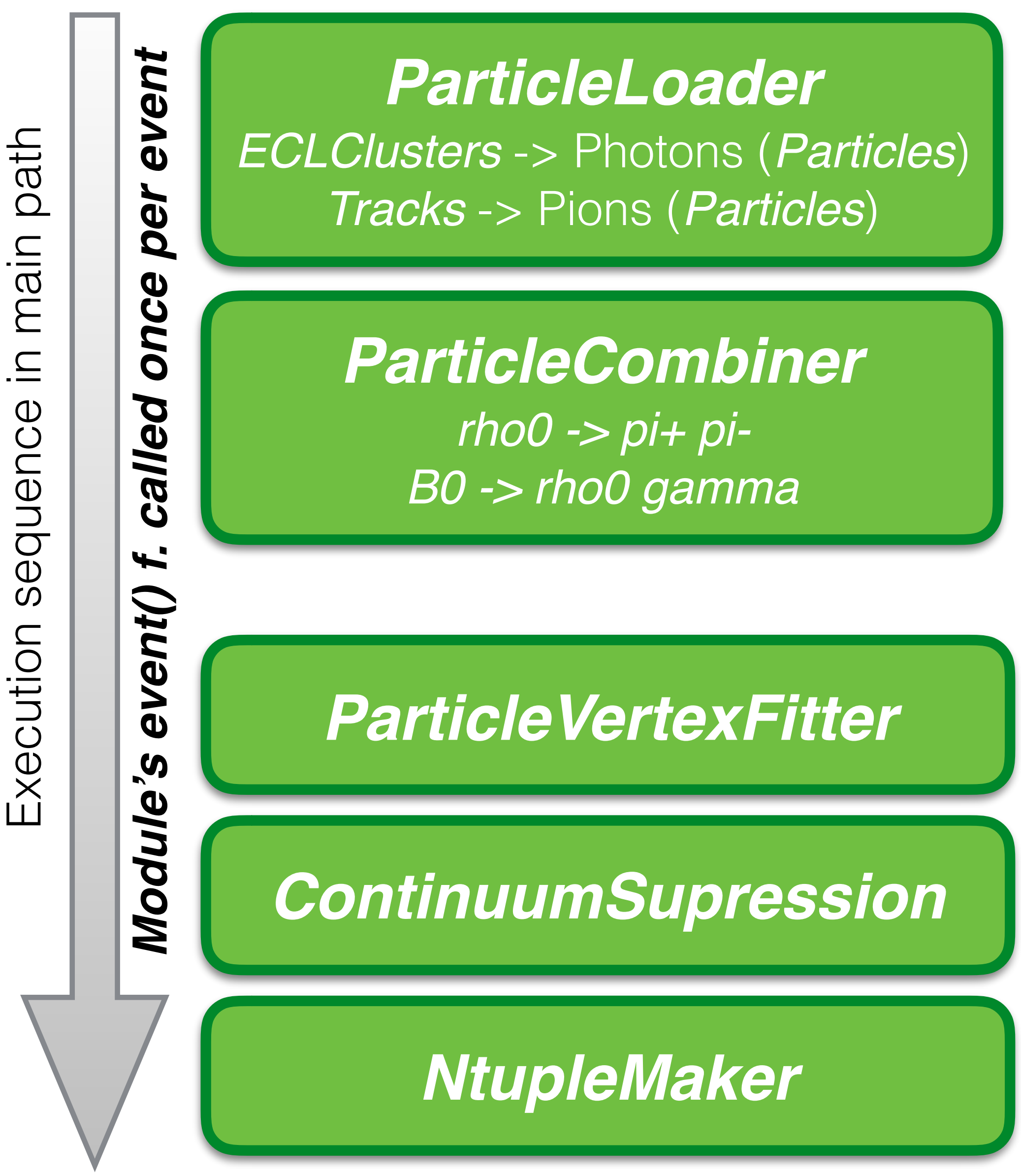


Analysis modules and execution path

Normally, a module's ***event()*** function is called once per event while any repeated processing is performed inside this call by e.g. looping over an array of input data.

Example:

- *reconstruction of* ***B0 -> rho0 gamma*** *decay*



Q: Does signal photon candidate originate from a π^0 decay?

A: Combine signal photon candidate with other photons in an event and check if the pair is consistent with π^0 hypothesis.

Analysis modules and execution path

Normally, a module's ***event()*** function is called once per event while any repeated processing is performed inside this call by e.g. looping over an array of input data.

Example:

- *reconstruction of **B0 -> rho0 gamma** decay*

Q: Does signal photon candidate originate from a π^0 decay?

A: Combine signal photon candidate with other photons in an event and check if the pair is consistent with π^0 hypothesis.

Execution sequence in main path

Module's event() f. called once per event

ParticleLoader
ECLClusters -> Photons (*Particles*)
Tracks -> Pions (*Particles*)

ParticleCombiner
rho0 -> pi+ pi-
B0 -> rho0 gamma

ParticleVertexFitter

ContinuumSupression

NtupleMaker

For each B0 candidate in an event:

Module's event() f. called
once per entry in given StoreArray

ParticleLoader
Remaining *ECLClusters* -> other Photons

ParticleCombiner
pi0 -> gamma (sig) gamma (other)

MVA classifier
construct pi0 probability for given signal photon

Analysis modules and execution path

Normally, a module's ***event()*** function is called once per event while any repeated processing is performed inside this call by e.g. looping over an array of input data.

Example:

- *reconstruction of **B0 -> rho0 gamma** decay*

Execution sequence in main path

Module's event() f. called once per event

ParticleLoader
ECLClusters -> Photons (*Particles*)
Tracks -> Pions (*Particles*)

ParticleCombiner
rho0 -> pi+ pi-
B0 -> rho0 gamma

ParticleVertexFitter

ContinuumSupression

NtupleMaker

Q: Does signal photon candidate originate from a π^0 decay?

A: Combine signal photon candidate with other photons in an event and check if the pair is consistent with π^0 hypothesis.

For each B0 candidate in an event:

Module's event() f. called
once per entry in given StoreArray

ParticleLoader
Remaining *ECLClusters* -> other Photons

ParticleCombiner

All analysis modules can be executed in path for each entry in some array

Exercise 9

- Use the steering file from exercise 7
- Create RestOfEvent object for each B0 candidate
- Loop over tracks in RestOfEvent and attach the eid and muid of the most electron/muon like track to B0 candidate
- dump this info to ntuple
- but first inspect the content of ROE by printing it out on few 10 events

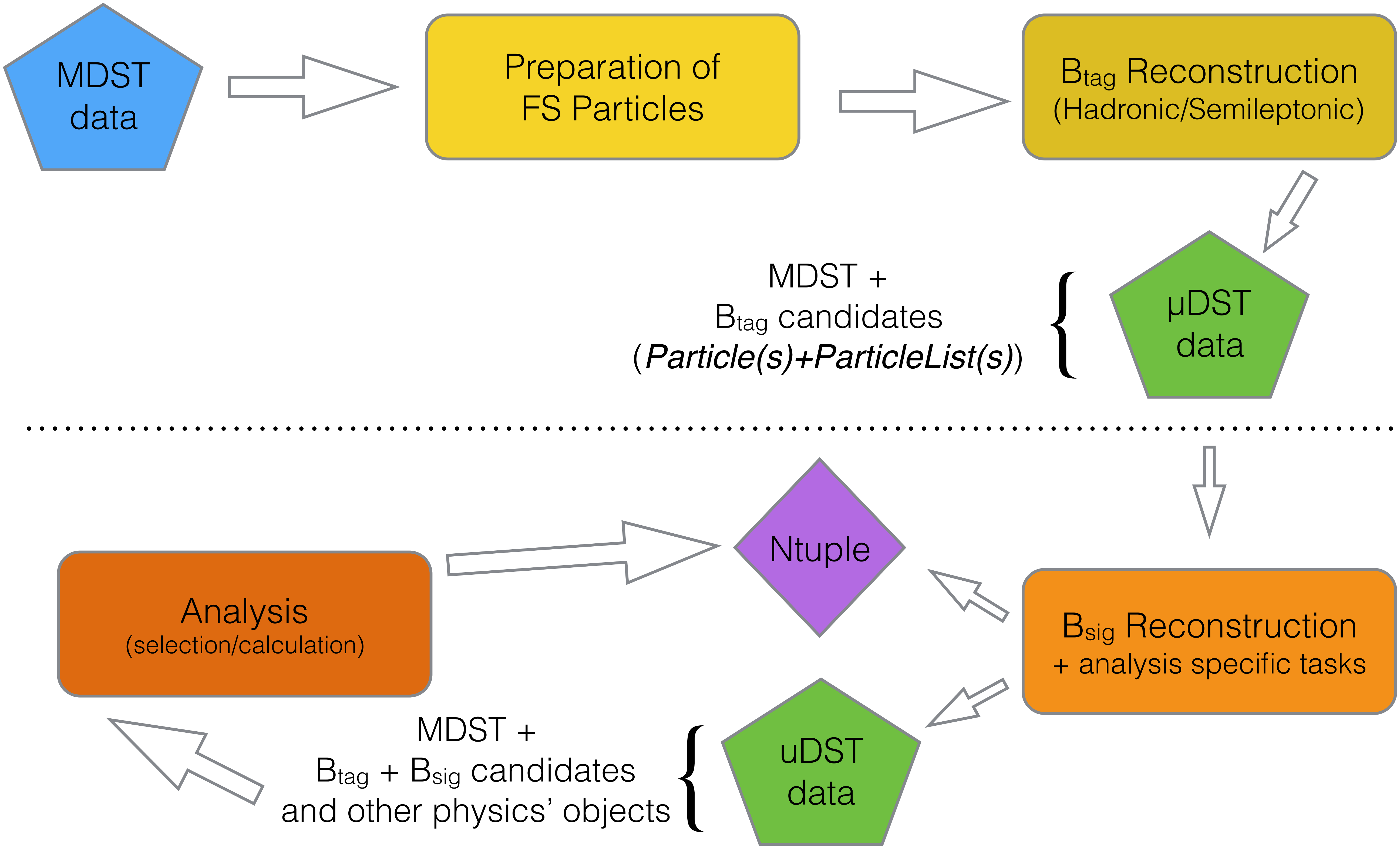
Hint: check $BELLE2_RELEASE_DIR/analysis/examples/tutorials/B2A306-B02RhoGamma-withPi0Veto.py for reference

Exercise 9

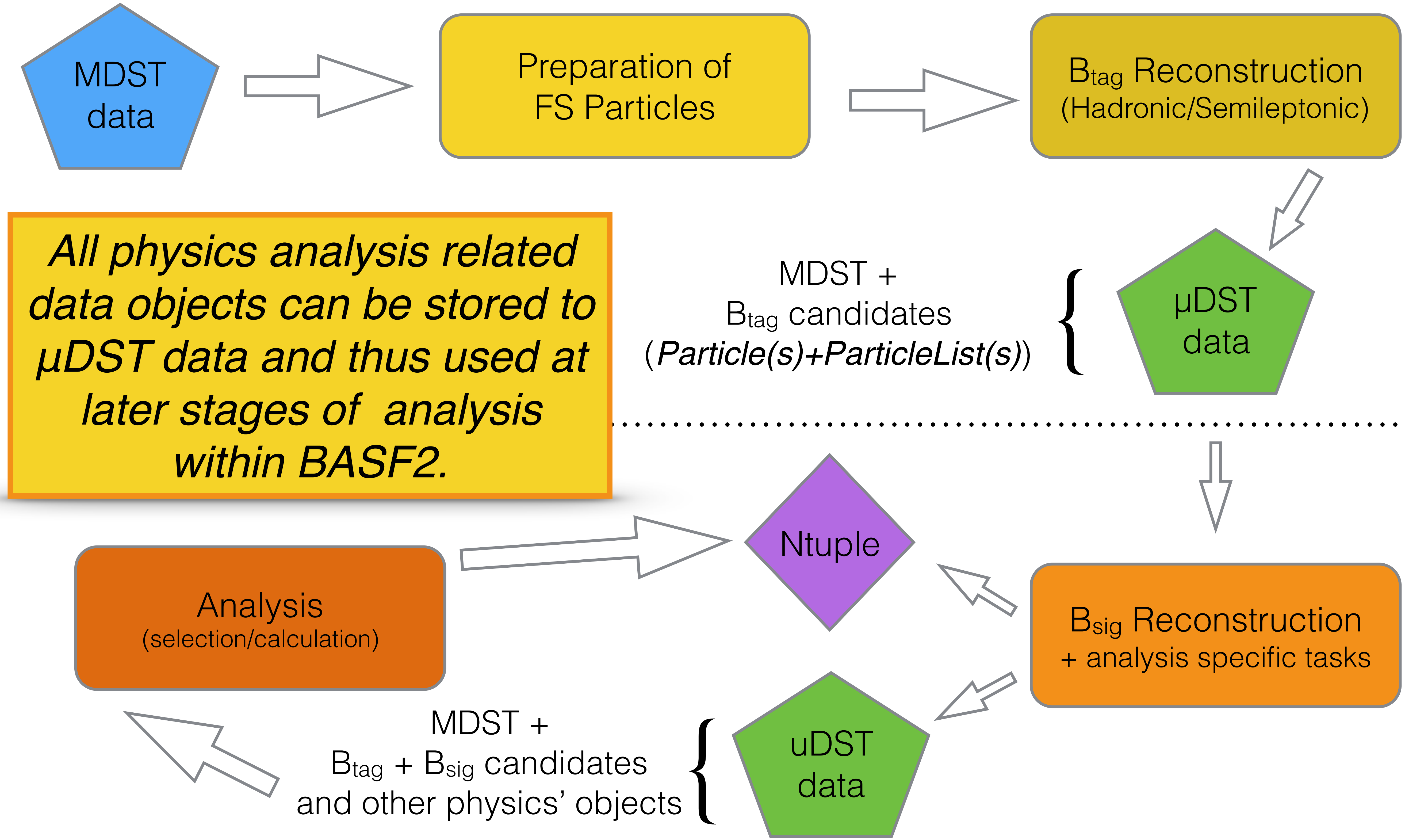
```
[INFO] [ParticlePrinterModule] START ----------------
[INFO]  ParticleLists: mu+:all (6+0) + mu-:all (6+0)
[INFO]  - 2 = -13[0]
[INFO]      o) muid = 5.78151e-17
[INFO]      o) mdstIndex = 2
[INFO]  - 3 = -13[1]
[INFO]      o) muid = 2.42386e-23
[INFO]      o) mdstIndex = 3
[INFO]  - 5 = -13[2]
[INFO]      o) muid = 1
[INFO]      o) mdstIndex = 5
[INFO]  - 6 = -13[3]
[INFO]      o) muid = 1.15364e-05
[INFO]      o) mdstIndex = 6
[INFO]  - 8 = -13[4]
[INFO]      o) muid = 0.892305
[INFO]      o) mdstIndex = 8
[INFO]  - 9 = -13[5]
[INFO]      o) muid = 0.42731
[INFO]      o) mdstIndex = 9
[INFO]  - 0 = 13[6]
[INFO]      o) muid = 0.221382
[INFO]      o) mdstIndex = 0
[INFO]  - 1 = 13[7]
[INFO]      o) muid = 0.11241
[INFO]      o) mdstIndex = 1
[INFO]  - 4 = 13[8]
[INFO]      o) muid = 1
[INFO]      o) mdstIndex = 4
[INFO]  - 7 = 13[9]
[INFO]      o) muid = 0.00380894
[INFO]      o) mdstIndex = 7
[INFO]  - 10 = 13[10]
[INFO]      o) muid = 0.57608
[INFO]      o) mdstIndex = 10
[INFO]  - 11 = 13[11]
[INFO]      o) muid = 2.49246e-10
[INFO]      o) mdstIndex = 11
```

```
[INFO] [ParticlePrinterModule] START ------------------------------
[INFO]  ParticleLists: B0:jpsiks (0+1) + anti-B0:jpsiks (0+1)
[INFO]  - 29 = 511[0]
[INFO]      o) daughter indices =  28 26
[INFO]       o) daughter(0,daughter(0,muid)) = 1
[INFO]       o) daughter(0,daughter(1,muid)) = 1
[INFO]       o) daughter(0,daughter(0,mdstIndex)) = 5
[INFO]       o) daughter(0,daughter(1,mdstIndex)) = 4
[INFO] [ParticlePrinterModule] END   ------------------------------
[INFO] [RestOfEventPrinterModule] START ------------------------------
[INFO]  - ROE related to particle with PDG: 511
[INFO]  - ROE related to MC particle with PDG: -511
[INFO]  - No. of Tracks in ROE: 8
[INFO]  - No. of ECLClusters in ROE: 39
[INFO] [RestOfEventPrinterModule] END --------------------------------
[INFO] [ParticlePrinterModule] START ------------------------------
[INFO]  ParticleLists: mu+:roe (1+0) + mu-:roe (0+0)
[INFO]  - 50 = -13[0]
[INFO]       o) muid = 0.892305
[INFO]       o) mdstIndex = 8
[INFO] [ParticlePrinterModule] END   ------------------------------
```

RootOuput - standard BASF2 output module for μDST production



RootOuput - standard BASF2 output module for μDST production



Exercise 10

- Use the steering file from exercise 9
- Create microDST file (miniDST + physics analysis data objects, like ParticleLists, Particles, ROE, …)
- After you create the microDST file check print the its DataStore content

Exercise 10

- Use the steering file from exercise 9
- Create microDST file (miniDST + physics analysis data objects, like ParticleLists, Particles, ROE, …)
- After you create the microDST file check print the its DataStore content

```
[INFO] ===========================================================================
[INFO] DataStore collections in event 700001
[INFO] ===========================================================================
[INFO] Type                 Name                        #Entries          <Event>
[INFO] ParticleList         B0:jpsiks
[INFO] RelationContainer    ECLClustersToMCParticles
[INFO] EventExtraInfo       EventExtraInfo
[INFO] EventMetaData        EventMetaData
[INFO] RelationContainer    KLMClustersToECLClusters
[INFO] RelationContainer    KLMClustersToMCParticles
[INFO] ParticleExtraInfoMap  ParticleExtraInfoMap
[INFO] RelationContainer    ParticlesToMCParticles
[INFO] RelationContainer    ParticlesToPIDLikelihoods
[INFO] RelationContainer    ParticlesToRestOfEvents
[INFO] RelationContainer    TracksToECLClusters
[INFO] RelationContainer    TracksToKLMClusters
[INFO] RelationContainer    TracksToMCParticles
[INFO] RelationContainer    TracksToPIDLikelihoods
[INFO] ParticleList         anti-B0:jpsiks
[INFO] ECLCluster[]         ECLClusters                 42
[INFO] KLMCluster[]         KLMClusters                 8
[INFO] MCParticle[]         MCParticles                 39
[INFO] PIDLikelihood[]      PIDLikelihoods              12
[INFO] Particle[]           Particles                   54
[INFO] RestOfEvent[]        RestOfEvents                1
[INFO] TrackFitResult[]     TrackFitResults             16
[INFO] Track[]              Tracks                      12
[INFO] V0[]                 V0s                         2
[INFO]
[INFO] ---------------------------------------------------------------------------
```

Exercise 11

- Run over microDST file produced in exercise 10
- fill the same ntuple as before and compare the content with previous ones without reconstructing the J/psi, B0, …, again

Exercise 11

- Run over microDST file produced in exercise 10
- fill the same ntuple as before and compare the content with previous ones without reconstructing the J/psi, B0, …, again