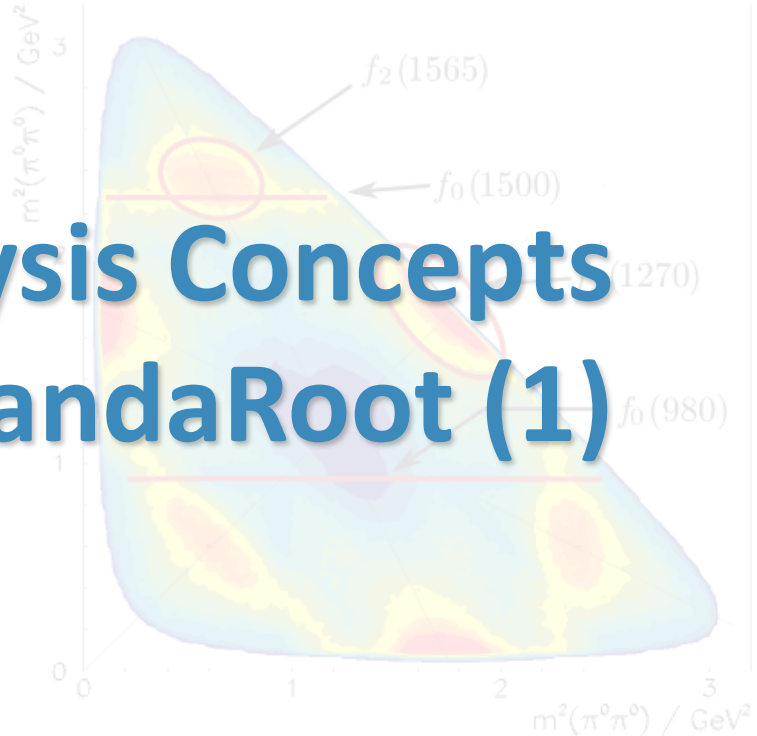
Physics Analysis Concepts with PandaRoot (1)



PANDA Lecture Week 2017

GSI, Dec 11 - 15, 2017

Klaus Götzen
GSI Darmstadt

Topics

- Data Levels
- Data Access *(PndAnalysis)*
- Analysis Basics
- Particle Candidates *(RhoCandidate, PndPidCandidate)*
- Combinatorics *(RhoCandList)*

Wiki Tutorial Page

- A all times up-to-date tutorial can be found here:
  http://panda-wiki.gsi.de/cgi-bin/view/Computing/PandaRootRhoTutorial

Sie sind hier: PANDA Wiki > Computing Web > PandaRoot > PandaRootRhoTutorial (30 Jan 2017, KlausGoetzen)

Simulation and Analysis in PandaRoot with RHO (Updated: Jan. 24th, 2017; Tested with rev 29680)

1. Simulating the Signal Events

1.1. Event Generation

In order to run the simulation we first of all need to generate events. This can be done on the fly with the simulation macro, the only thing we need is a decay file for EvtGen, which defines our decay.

In our example the decay file is given by `pp_jpsi2pi_jpsi_mumu.dec` which lists as

```
noPhotos

Decay pbarpSystem
  1.0 J/psi pi+ pi-    PHSP;
Enddecay

Decay J/psi
  1.0 mu+ mu-      VLL;
Enddecay
```

Wiki Rho Documentation

http://panda-wiki.gsi.de/cgi-bin/view/Computing/PandaRootAnalysisJuly13

Edit Attach Printable

Computing.PandaRootAnalysisJuly13

Documentation of Rho Classes

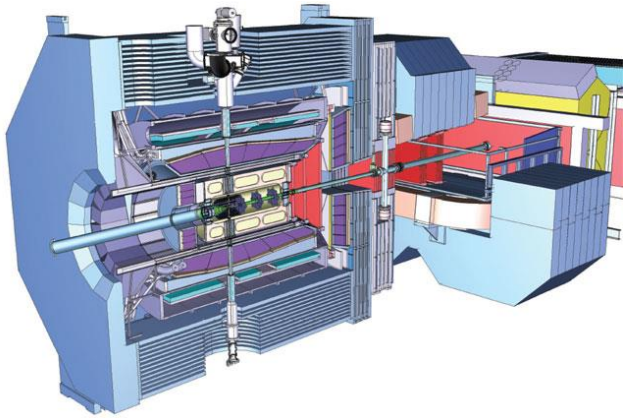
Index

Here you can find information about:

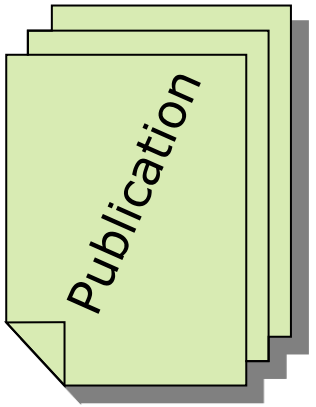
- **PndAnalysis** - Interface - Lists and Keys - PID Algo Names - MC Truth Match
- **RhoCandidate** - Interface
- **RhoCandList** - Interface
- **Particle Selectors** - Kinematic Selectors - PID Selectors
- **Fitters** - Vertex Fitting - 4C Fitting - Kinematic Fitting

A tutorial for Rho with an example simulation and analysis can be found on this Wiki page.

The Aim



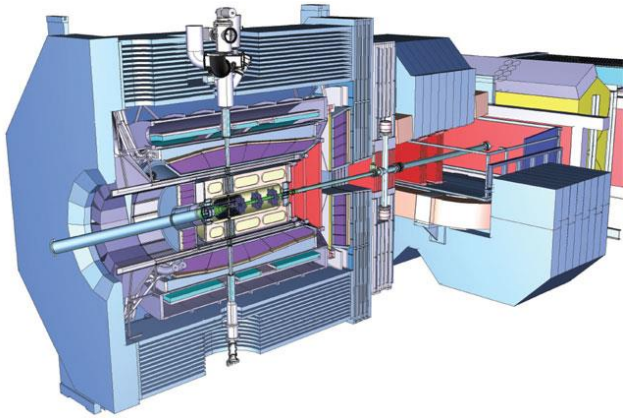
Data Analysis



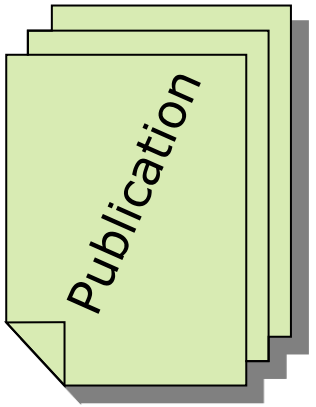
- What do you want to analyse/investigate?
- How can it be measured (what are the observables)?

- Reconstruct the corresponding signal channels!
- Determine the needed quantities!
- Estimate the errors (statistic & systematic)!
- Write paper
- Get invited to Stockholm

The Aim



Data Analysis



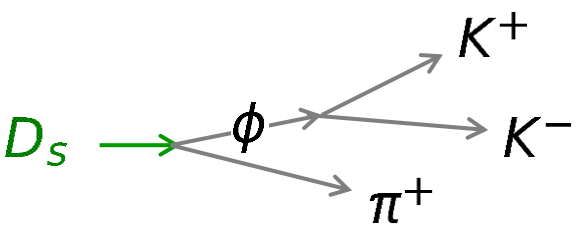
- What do you want to analyse/investigate?
- How can it be measured (what are the observables)?

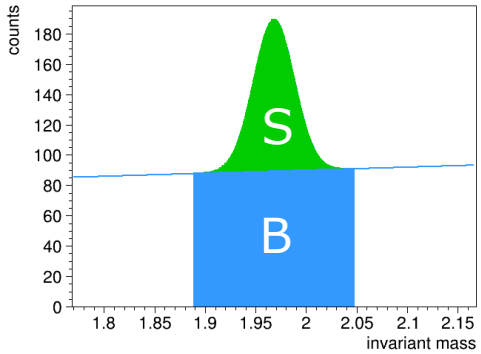
- Reconstruct the corresponding signal channels!
- Determine the needed quantities!
- Estimate the errors (statistic & systematic)!
- Write paper
- Get invited to Stockholm

Gonna talk about that one!

Example: Count Number of D_s in Data

- Reconstruct signal in data; maximize e.g. significance $\frac{S}{\sqrt{S+B}}$





S = entries in signal peak
B = entries below peak

- Reconstruct signal in dedicated signal Monte Carlo events; number of *MC truth matched* signals give efficiency

$$\varepsilon = \frac{N_{reco,MC}}{N_{0,MC}}$$

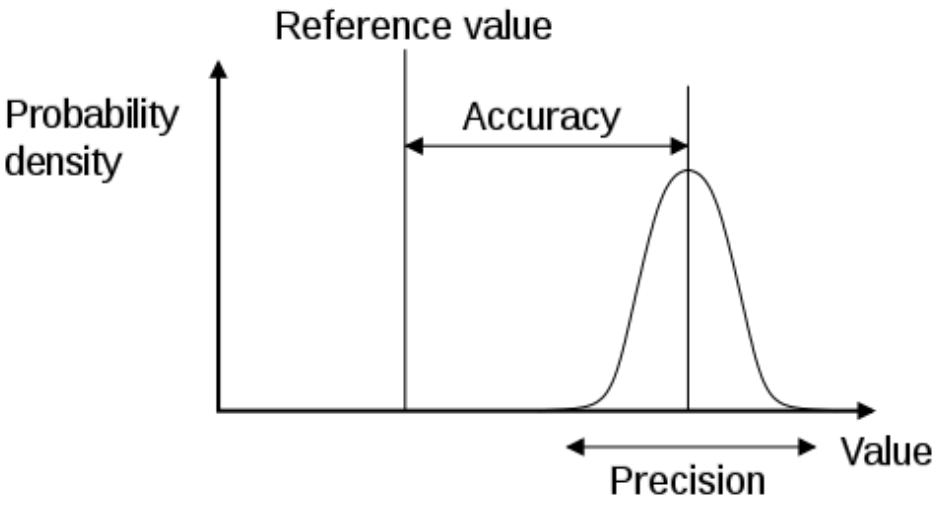
- Number of D_s in data is (taking into account BR's)

$$N_{D_s} = \frac{S}{\varepsilon \cdot \mathcal{BR}(D_s \to \phi\pi^{\pm}) \cdot \mathcal{BR}(\phi \to K^+K^-)}$$

Goal for Experimentalist

- **Theoreticians**: interested in measured **value**
  - $M_{Higgs} \approx$ **125** GeV

- **Experimentalists:** interested in **precision/accuracy (error)**
  - $M_{Higgs} =$ X **±** **2** GeV



High accuracy
(low systematic error)

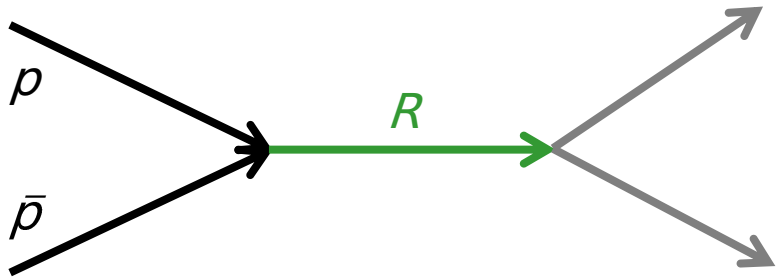
High precision
(low statistic error)

BASICS

Reactions in PANDA

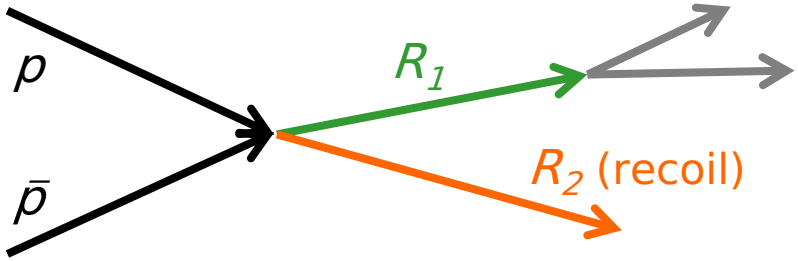
There are two different kinds of principle reactions in PANDA

- **Formation** reactions
  The $\bar{p}p$ system transforms into single resonance state R, which decays afterwards



$\bar{p}$ beam momentum ($\rightarrow \sqrt{s}$) is fixed by mass of R

- **Production** reactions
  The $\bar{p}p$ system transforms into more than one resonance



$\bar{p}$ beam momentum can vary due to kinematic recoil R_2

Data Levels

What kind of data (levels) do we have in an experiment?

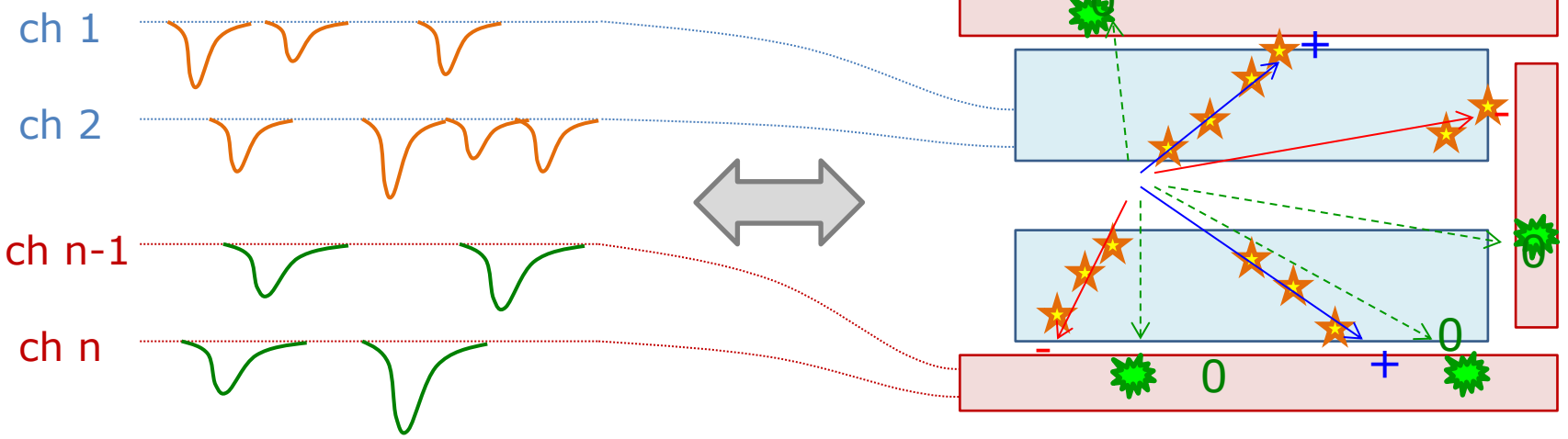
- Raw data
  The stream of digitized hits from the detectors
  Basis for track/neutral cluster/PID reconstruction

- Analysis Object Data (AOD) / Data Summary Tape (DST)
  Reconstructed objects like tracks and neutrals with associated PID information, bundled to event based data packages
  Basis for physics analysis

- Tag level data
  Condensed information of events like total energy or momentum, multiplicities (tracks, gammas, kaons, etc.)
  Fast data filtering

Data Levels

RAW data



TAG data

2 positive charged (+)
2 negative charged (-)
4 neutral (0)
...

Data Access in PandaRoot

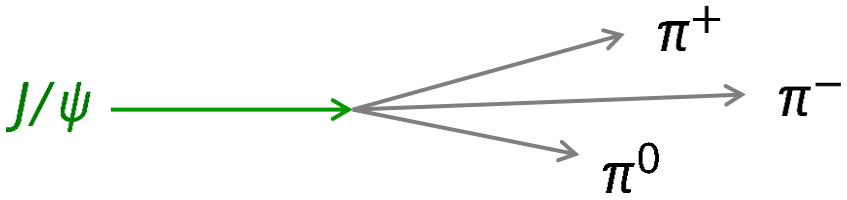
- PandaRoot object: **PndAnalysis**

```
FairRunAna *fRun = new FairRunAna();
fRun->SetInputFile("pid_complete.root");

PndAnalysis *ana = new PndAnalysis();
RhoCandList eplus, eminus, muplus, muminus, mct;
TString myPidAlgosElectron = "PidAlgoEmcBayes;PidAlgoDrc"
TString myPidAlgosMuon     = "PidAlgoMdtHardCuts"
...
while ( ana->GetEvent() )
{
  ana->FillList( eplus,   "ElectronTightPlus",  myPidAlgosElectron );
  ana->FillList( eminus,  "ElectronTightMinus", myPidAlgosElectron );
  ana->FillList( muplus,  "MuonTightPlus",      myPidAlgosMuon );
  ana->FillList( muminus, "MuonTightMinus",     myPidAlgosMuon );
  ana->FillList( mct,     "McTruth" );
  ...
}
```

- Features:
  - Simple access to reco candidates and McTruth objects
  - Various PID algorithms directly accessible

The basis for Analysis: Invariant masses

- Q: How do we detect a certain resonance decay?



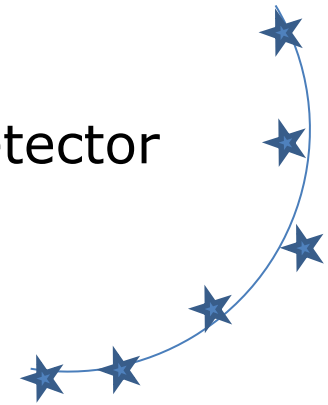
- A: *Estimate* the 4-vectors of the final states (FS), sum up and compute the invariant mass

$$m_{inv} = \sqrt{\left(\sum E\right)^2 - \left(\sum p_x\right)^2 - \left(\sum p_y\right)^2 - \left(\sum p_z\right)^2}$$

It should be ‚close' to the resonance rest mass.

Why did I write estimate?
Usually only momentum (charged) or energy (neutrals) from FS particles is measured; the so-called *mass hypothesis* has to be set during analysis, typically based on PID detector information.

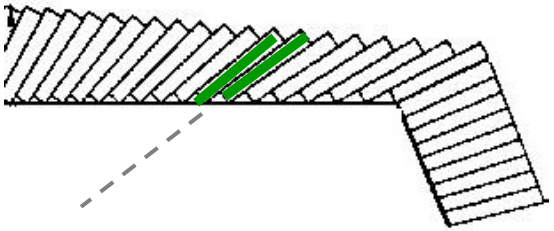
How 4-vectors are reconstructed



Charged particles

- Reconstruction of trajectory with tracking/vertex detector
- Curvature in magnetic field → momentum p
- Assume position on trajectory → 3-vector (p_x, p_y, p_z)
- Assume mass hypothesis m → $E = \sqrt{m^2 + p^2}$

Neutral particles (gammas)

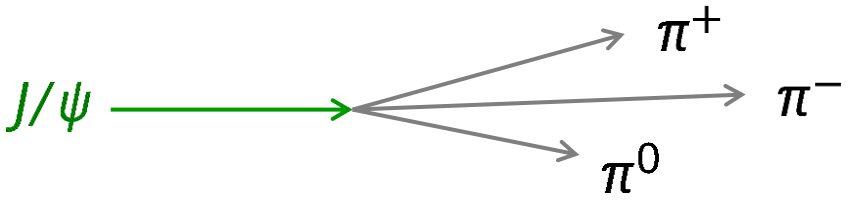


- Reconstruction of cluster in EM-calorimeter
- Cluster energy → particle energy E
- Assume gamma mass $m=0$ → $p = E$
- Assume gamma comes from IP → 3-vector (p_x, p_y, p_z)

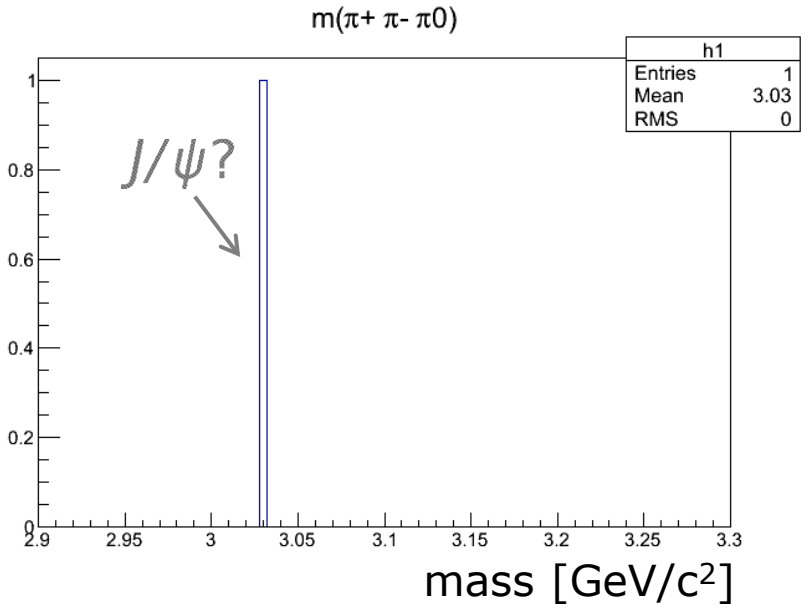
⇒ Determination of 4-vectors is not an unbiased process!

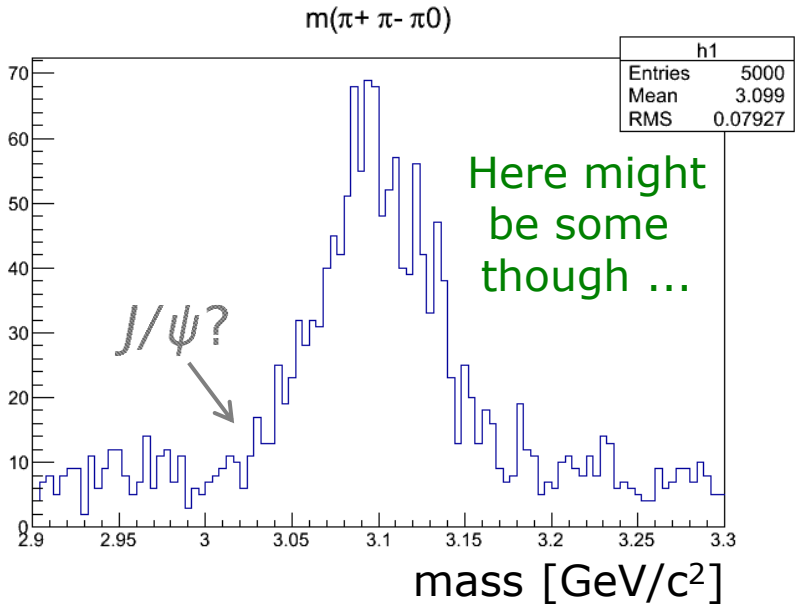
The basis for Analysis: Invariant masses

- Q: How do we detect a certain resonance decay?



- A2: Not at all (in principle) for a single decay!





Particle Candidate in PandaRoot

- PandaRoot object: **RhoCandidate**
- Some important accessors:

Return type	Method	Information
TLorentzVector	P4()	4-vector
TVector3	P3()	Momentum vector
TVector3	Pos()	Origin
Double_t	Charge()	Charge
Double_t	M(), P(), E()	Mass, Momentum, Energy
Int_t	NDaughters()	Number of daughters
RhoCandidate*	Daughter(i)	i-th daughter in decay tree
RhoCandidate*	TheMother()	Mother in decay tree
RhoCandidate*	Combine(TCandidate &c,...)	Creates composite candidate
Bool_t	Overlaps(TCandidate &c)	Checks for collision in tree
RhoCandidate*	GetMcTruth()	MC truth object
Int_t	PdgCode()	PDG code
FairRecoCandidate*	GetRecoCandidate()	Access to reco information

Micro Candidate in PandaRoot

- PandaRoot object: **PndPidCandidate** *(FairRecoCandidate)*
- Some accessors:

Return type	Method	Information
Int_t	GetMvdHits()	Number of MVD hits
Float_t	GetMvdDEDX()	dE/dx measurm. from MVD
Int_t	GetSttHits()	Number of STT hits
Float_t	GetSttMeanDEDX()	dE/dx measurm. from STT
Float_t	GetDrcThetaC()	θ_C measurement from DIRC
Int_t	GetDrcNumberOfPhotons()	Number Photons in DIRC
Float_t	GetEmcCalEnergy()	Calibrated EMC cluster energy
Int_t	GetEmcNumberOfCrystals()	Number of crystals in cluster
Int_t	GetMuoNumberOfLayers()	Number of layers hit in MUO det
Float_t	GetMuoProbability()	Probability for being a muon

Work with RhoCandidate/PndPidCandidate

- Using PndPidCandidate

```
...
PndAnalysis *pndana= new PndAnalysis();

RhoCandList piplus, piminus, goodpiplus, goodpiminus;

while (pndana->GetEvent())
{
  pndana->FillList(piplus,  "PionAllPlus");  // access to reco candidates
  pndana->FillList(piminus, "PionAllMinus");
  goodpiplus.Cleanup();

  for (int i=0; i<piplus.GetLength(); ++i)
  {
     // get micro candidate and use it for selection
     PndPidCandidate *mic = (PndPidCandidate*) piplus[i]->GetRecoCandidate();

     if (mic->GetSttHits()>5) goodpiplus.Add(piplus[i]);
  }

 ...
}
```

General handling of RhoCandidate

- RhoCandidate = basic analysis object, many instances needed
- **RhoFactory** does the book-keeping of instances

- Creation of new RhoCandidates *(when done by hand):*

RhoCandidate *c = RhoFactory::Instance()->NewCandidate();

RhoCandidate *c = new RhoCandidate(); NEVER!!

- Delete RhoCandidates:

Not necessary!! *RhoFactory takes care of it!*

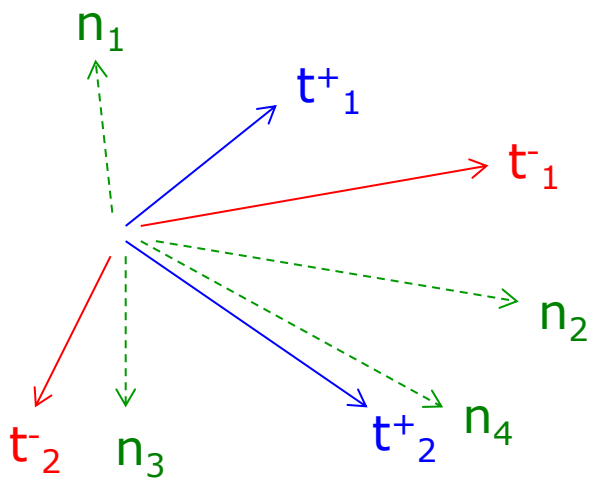
delete c; NEVER!!

actually, it's not super-bad, but life is easier w/o

COMBINATORICS

Combining Candidates

- Remember the example event shown above



- Question: How do we find out, whether there was a decay

$$J/\psi \rightarrow \pi^+\pi^-\pi^0(\rightarrow \gamma\gamma)$$

Combinatorics

- Answer: Create all combinations of FS particles, which match the decay pattern and create histogram of inv. mass.

- Our example:
  2 positive (t^+_1, t^+_2), 2 negative tracks (t^-_1, t^-_2), 4 neutrals (n_k)

- π^0 candidates → 6 combinations

  $\pi^0_1=(n_1+n_2)$, (n_1+n_3), (n_1+n_4), (n_2+n_3), (n_2+n_4), $\pi^0_6=(n_3+n_4)$

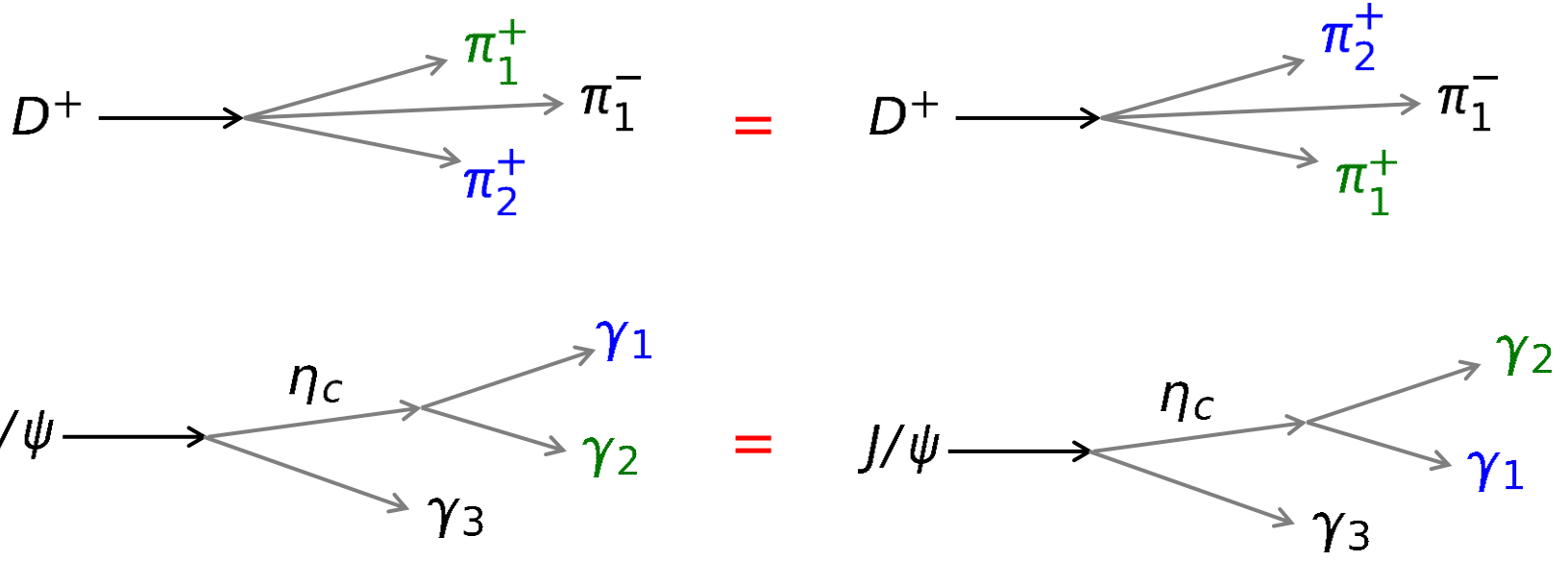
- J/ψ candidates → 24 combinations

  $J/\psi_1 = (t^+_1 + t^-_1 + \pi^0_1)$, $J/\psi_2 = (t^+_1 + t^-_1 + \pi^0_2)$, ...
  ...
  $J/\psi_{24} = (t^+_2 + t^-_2 + \pi^0_6)$

Caveat: Double counting

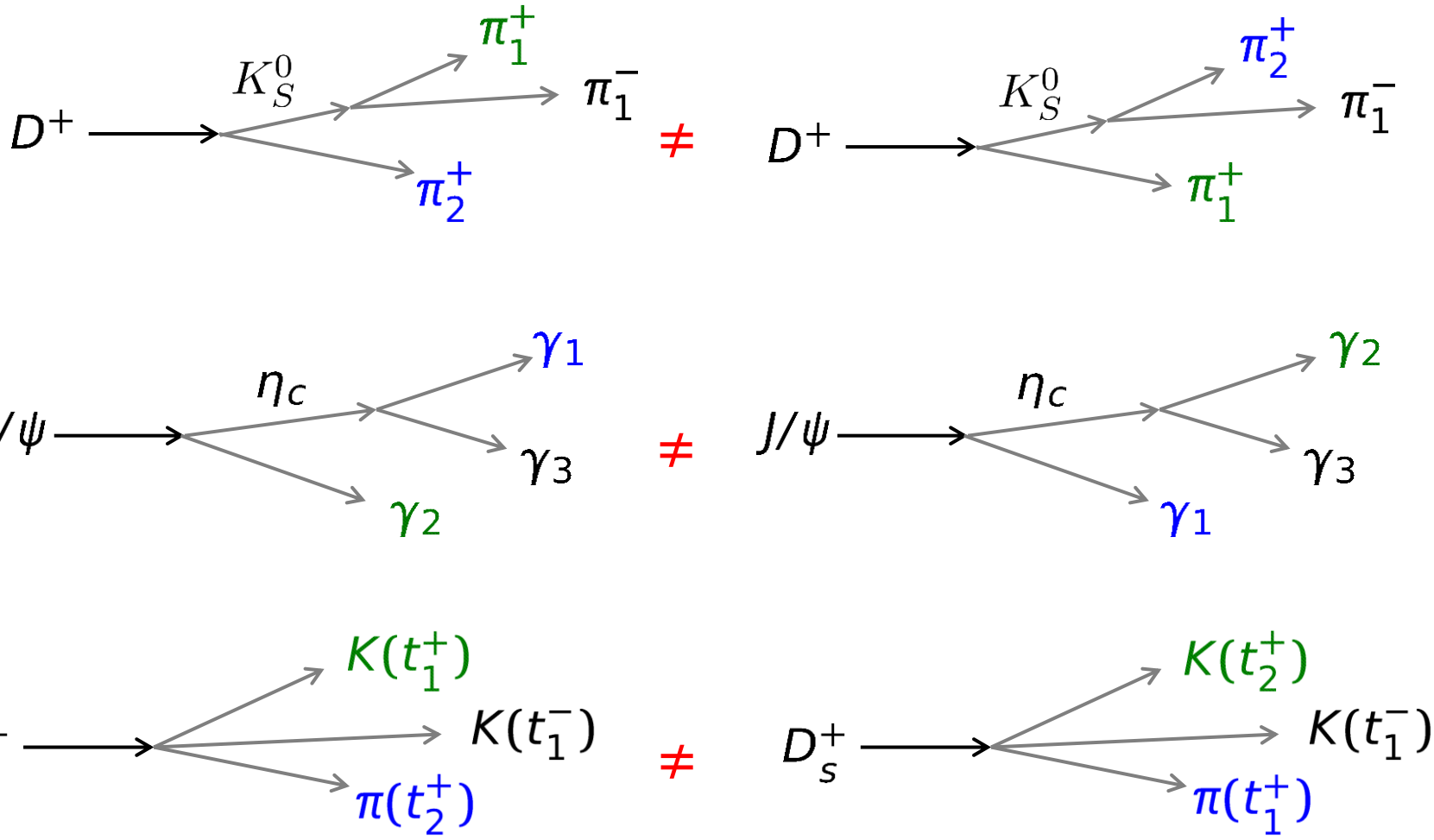
- Double counting = erroneous creation of identical combinations leading to multiple entries in spectra/lists
- Can happen in wrong coded nested loops

- Examples for double counting:



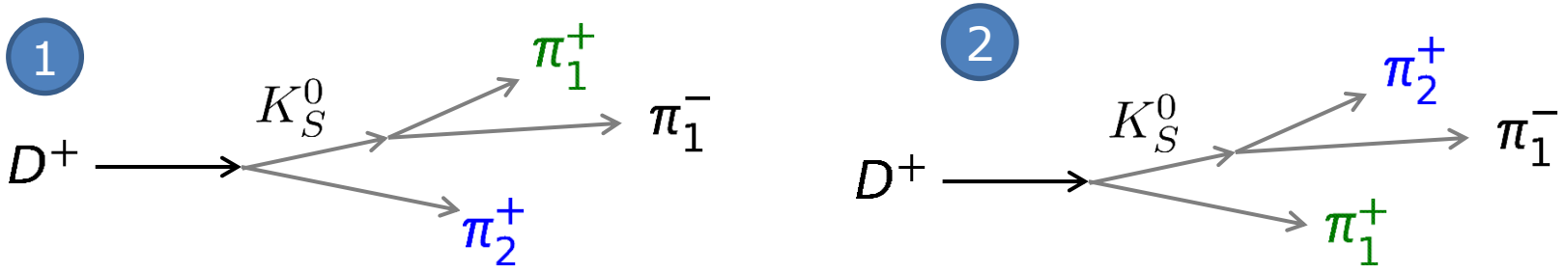
Caveat: Double counting

No double counting here *(decay trees have different topology!)*:



Caveat: Double counting

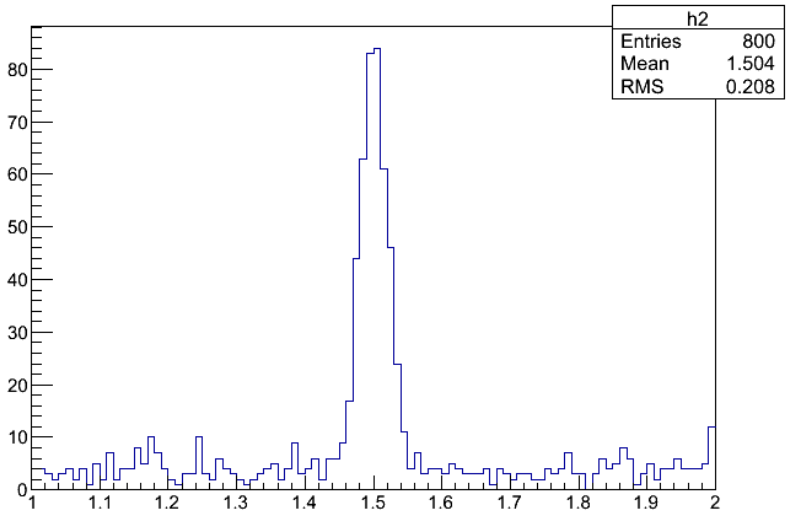
But be aware:

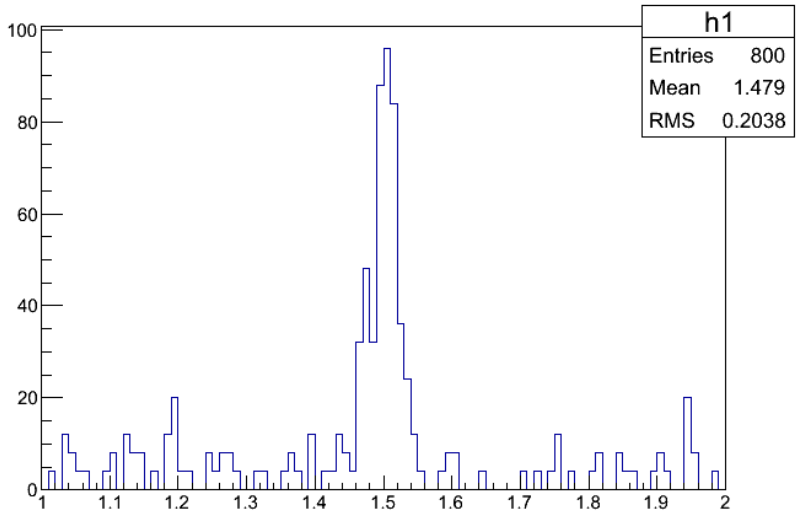


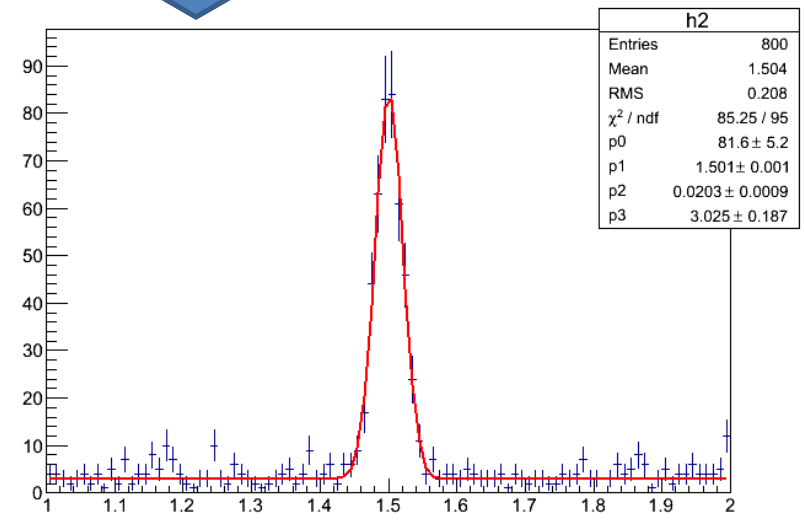
- If cut applied to K_S^0
  → only one of (1) and (2) might be selected.
- If both selected
  → two different decay trees with identical 4-vector of the D^+
- If K_S^0 fit → probably will have two different 4-vectors

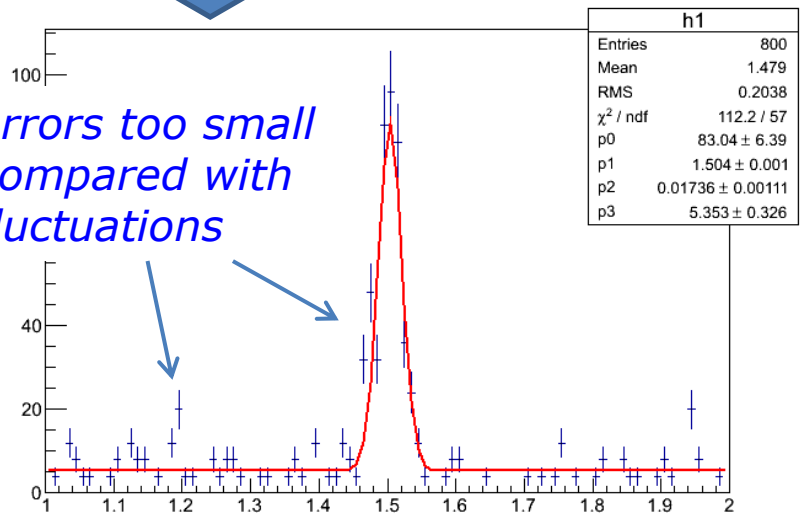
- *Is this now double counting?*
- Simple recipe: *Just keep one (the best) solution per event*

Double counting: How it might look like



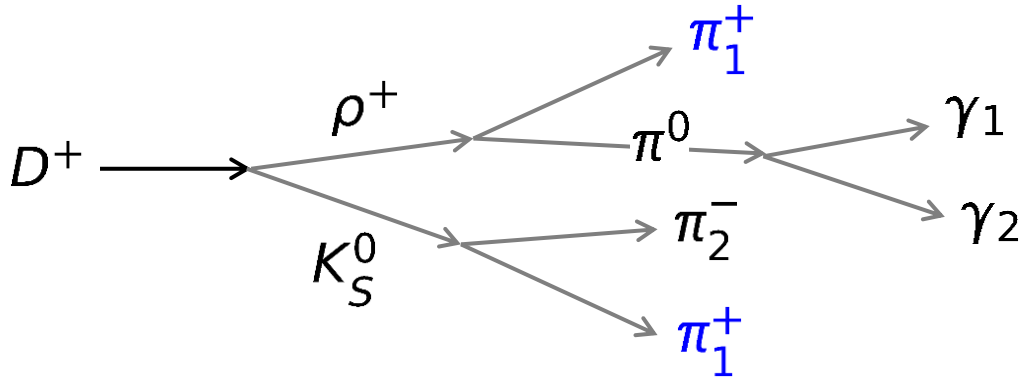


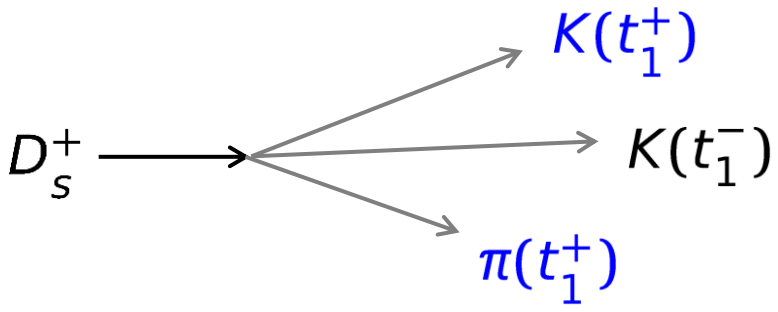




Caveat: Overlaps

- Overlap = erroneous multiple use of the same reco object in a decay tree
- Can happen when working with composite candidates or different mass hypotheses





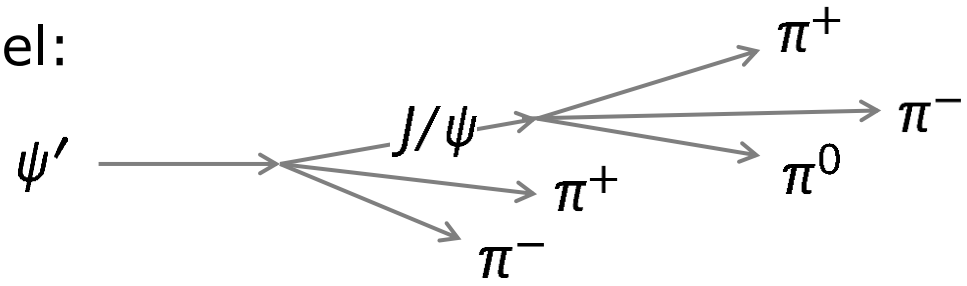
Particle List in PandaROOT

- PandaROOT object: **RhoCandList**
- Some important accessors:

Method	Information
Int_t GetLength()	number of particles in list
void Add(RhoCandidate*)	Appends candidate to list
void Remove(RhoCandidate*)	Removes candidate from list
RhoCandidate* operator[](Int_t)	Returns a candidate
void Append(RhoCandList&)	Appends candidates from another list
void Combine(RhoCandList&,...)	Combinatorics with other RhoCandList(s)
void Select(VAbsPidSelector*)	Selects (=modifies) list
void Select(RhoCandList &, VAbsPid...)	Selects from another list
void Cleanup()	Empties list

Combinatorics in PandaROOT

- PandaROOT objects: **RhoCandList & RhoCandidate**
- Example channel:




```
RhoCandList piplus, piminus, gamma, pi0, jpsi, psip;  // define RhoCandList's

pndana->FillList(piplus,  "PionAllPlus");  // positive chrg trks with pion hypo
pndana->FillList(piminus, "PionAllMinus"); // negative chrg trks with pion hypo

pndana->FillList(gamma, "Neutral");        // neutral particle candidates

pi0.Combine(gamma, gamma);                 // cares about double counting
jpsi.Combine(piplus, piminus, pi0);
psip.Combine(jpsi, piplus, piminus);       // cares about overlaps

for  (int j=0; j<psip.GetLength(); ++j)    // loop over candidates in TCandList
{
    masshisto.Fill( psip[j]->M() );        // and fill e.g. a mass histo
}
```

Combinatorics in PandaROOT (by hand)

- Example channel: $J/\psi \rightarrow \pi^+\pi^-\pi^0(\rightarrow \gamma\gamma)$

```
RhoCandList piplus, piminus, gamma, pi0, jpsi;            // define TCandList's
...
  pi0.Cleanup(); jpsi.Cleanup();                          // clean lists !!

  for (int i=0; i<gamma.GetLength()-1; i++) {             // nested loops
    for (int j=i+1; j<gamma.GetLength(); j++) {
      if (gamma[i]->E()>Emin || gamma[j]->E()>Emin) {      // do e.g. some check
         RhoCandidate *comb = gamma[i]->Combine(gamma[j]); // create comb. candidate
         pi0.Add(comb);                                    // add it to list
      }
    }
  }

  for (int i=0; i<piplus.GetLength(); i++) {              // and the same for
    for (int j=0; j<piminus.GetLength(); j++) {           // the J/psi candidates
      for (int k=0; k<pi0.GetLength(); k++) {
          RhoCandidate *comb = piplus[i]->Combine(piminus[j], pi0[k]);
          jpsi.Add(comb);
      }
    }
  }
...
```

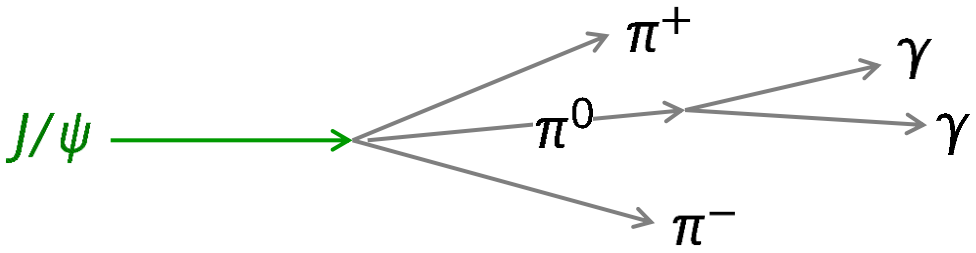
Combinatorics in PandaROOT (by hand)

- Example channel (overlap check):

```
RhoCandList pipm, kplus, kminus, ds;                    // define RhoCandList's
...
  ds.Cleanup();                                         // clean list

  for (int i=0; i<kplus.GetLength(); i++)               // nested loops
  {
    for (int j=0; j<kminus.GetLength(); j++)
    {
      RhoCandidate *phi = kplus[i]->Combine(kminus[j]); // create phi candidate

      for (int k=0; k<pipm.GetLength(); k++)
      {
        if (!pipm[k]->Overlaps(phi))                    // check for overlap
        {
          RhoCandidate *comb = phi->Combine(pipm[k]);
          ds.Add(comb);
        }
      }
    }
  }
...
```

Access to Genealogy

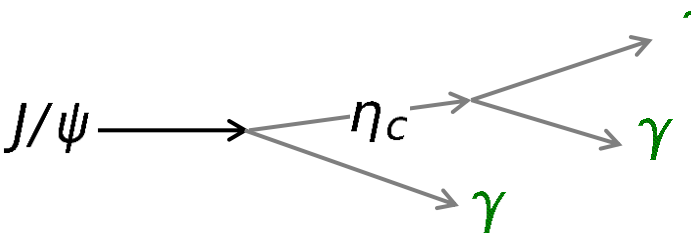
- PandaROOT object: **RhoCandidate**
- Example channel:



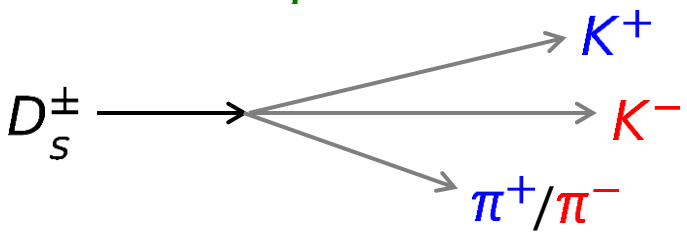

```
pi0.Combine(gamma, gamma);              // create pi0 candidates
jpsi.Combine(piplus, piminus, pi0);     // create J/psi candidates
...
cout << jpsi[i]->NDaughters() <<endl;  // prints number of daughters (3, not 4!)

RhoCandidate *pip = jpsi[i]->Daughter(0); // the pi+ candidate
RhoCandidate *pim = jpsi[i]->Daughter(1); // the pi- candidate
RhoCandidate *pi0 = jpsi[i]->Daughter(2); // the pi0 candidate (composite!)

RhoCandidate *gam1 = pi0->Daughter(0);   // the 1st gamma
RhoCandidate *gam2 = pi0->Daughter(1);   // the 2nd gamma
```

Combinatoric Exercises

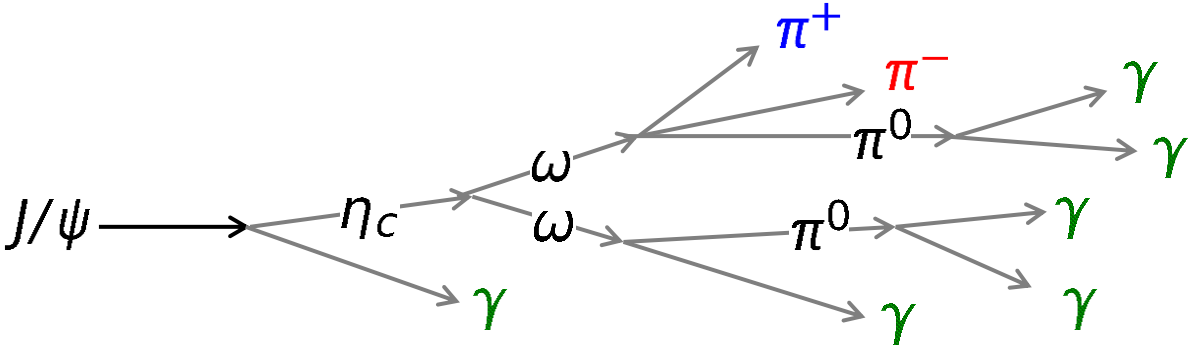
- Particles in event: $3t^+$, $3t^-$, $6n$; no PID information used
- How many combinations for
  - $J/\psi \to \gamma\eta_c \to \gamma\gamma\gamma$? **60**



  - $D_s^{\pm} \to K^+ K^- \pi^{\pm}$? **36**



  - $J/\psi \to \gamma\eta_c \to \gamma\omega\omega \to \gamma\ \pi^+\pi^-\pi^0\ \pi^0\gamma \to \gamma\ \pi^+\pi^-\ \gamma\gamma\ \gamma\gamma\gamma$?



1620

EXERCISES

Exercises Preparation

Preparation/hints in tutorials/thailand2017/README

- **FIRST setup environment: source ~/workshop/build/PandaRoot_trunk.../config.sh**

- To have some data for tutorial macros, do one of the following

```
a) . ./tut_runall.sh 1000   # sim/reco 1000 events pbarp -> J/psi pi+ pi-
b) cp data/signal_p*root .  # preproduced default data
```

- Macros named 'tut_ana...C' are stubs and should be completed by you.

- At places marked with ' #### EXERCISE: ...' some code needs to be added;

```
Run macros (default or different input data) with
> root -l tut_ana...C                 # signal_pid.root, signal_par.root
> root -l 'tut_ana...C(0,"mydata")'   # mydata_pid.root, mydata_par.root
```

- If getting stuck, sample solutions are in the subfolder 'solution'

```
Run solution macros directly (default or different input data) with
> root -l solution/tut_ana...C
> root -l 'solution/tut_ana...C(0,"mydata")'
```

Exercises Suggestions

- Rho Tutorial website:
  http://panda-wiki.gsi.de/cgi-bin/view/Computing/PandaRootRhoTutorial
- Take a look to `tutorials/thailand2017/README`
  `Exercise: #1`

1. Write macro to reconstruct (combinatorics) one/some of the channels in ./data

2. Compare results with nested loop to RhoCandList combinatorics