New Analysis Tutorial
incl. PID & MC truth match

6. 3. 2012

K. Götzen, GSI

Analysis Tutorial

- New Tutorial in PANDA Wiki (linked from Computing->Pandaroot)

http://panda-wiki.gsi.de/cgi-bin/view/Computing/PandaRootTutFeb2012

Tutorial on PandaRoot Simulation and Analysis (Feb. 2012)

Preface/Requirements

This tutorial aims to demonstrate, how simulation and analysis can be performed with the PandaRoot framework. The example channel is

p pbar -> Psi(2S)-> J/Psi (-> e+e-) pi+ pi-

The analysis part will cover

- Data access
- Particle Identification
- Combinatorics
- MC truth match
- Fitting with 4 constraint (4C), vertex constraint, mass constraint

The tutorial is based on **svn revision 14705 of pandaroot/trunk** . All macros and necessary files can be found under **trunk/tutorials/feb12** .

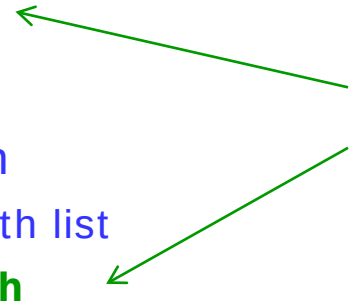
Topics

1. Simulating the Signal Events

- 1.1. Event Generation
- 1.2. Simulation, Digitization, Reconstruction and Particle Identification

2. Analysis of Signal Events

- 2.1. Data Access
- 2.2. Particle Identification
  - 2.2.1. Set Mass Hypothesis only
  - 2.2.2. Various PID Criteria
  - **2.2.3. PID Algorithms**
- 2.3. Combinatorics
- 2.4. Monte Carlo Truth Match
  - 2.4.1. Accessing the MC truth list
  - **2.4.2. Full MC truth match**
- 2.5. Fitting
  - 2.5.1. Vertex Constraint
  - 2.5.2. 4-Constraint
  - 2.5.3. Mass constraint



will say a bit more about it

Data Access/PID

- Data access via PndAnalysis

```
FairRunAna *fRun = new FairRunAna();
fRun->SetInputFile("mymicro.root");
fRun->AddFriend("mysim.root");

PndAnalysis *evr= new PndAnalysis();
...
while (evr->GetEvent())
{
  evr->FillList(eplus,  "ElectronLoosePlus",  "PidAlgoEmcBayes");
  evr->FillList(eminus, "ElectronLooseMinus", "PidAlgoEmcBayes");
  evr->FillList(mcList, "McTruth");
  ...
    TCandidate *mc = mcList[eplus[i].GetMcIdx()];
}
```

- Features:
  - Simple access to reco'd candidates and McTruth objects
  - New: Different (and realistic!) PID algorithms available

PID

Extended PID concept (interfaced by R. Kliemt)

- PID probabilities are computed by various algorithms *PidAlgoEmcBayes*, *PidAlgoDrc*, *PidAlgoMvd*, ...

- PndAnaPidCombiner
  - combines *on demand* probabilities from various algo‘s by computing product of all P_i *(i=algo‘s)*
  - copies resulting probabilities to TCandidate/TCandList

- PndAnaPidSelector
  - selects particles based on these probabilities

- PndAnalysis::FillList is a short-cut to this funtionallity via

```
evr.FillList(..., "PidAlgoEmcBayes;PidAlgoDrc");
```

PID

Extended PID concept (interfaced by R. Kliemt)

	Event 1			*Event 1*			...
	TRK 1	**...**	**TRK m1**	**TRK 1**	**...**	**TRK m2**	
Algo 1	P_e, ... ,P_p	...	P_e, ... ,P_p	P_e, ... ,P_p	...	P_e, ... ,P_p	
Algo 2	P_e, ... ,P_p	...	P_e, ... ,P_p	P_e, ... ,P_p	...	P_e, ... ,P_p	
...	P_e, ... ,P_p	...	P_e, ... ,P_p	P_e, ... ,P_p	...	P_e, ... ,P_p	
Algo n	P_e, ... ,P_p	...	P_e, ... ,P_p	P_e, ... ,P_p	...	P_e, ... ,P_p	

PndAnaPidCombiner
Multiply

	Event 1			*Event 1*			...
	TRK 1	**...**	**TRK m1**	**TRK 1**	**...**	**TRK m2**	
TCandidate	P_e, ... ,P_p	...	P_e, ... ,P_p	P_e, ... ,P_p	...	P_e, ... ,P_p	

PndAnaPidSelector
Cut on

Stand-alone usage of Selector

- Selector can be used independent of PndAnalysis::FillList
- Say, you have a list of charged particles and want to identify loose kaons with PidAlgoDrc & PidAlgoMvd

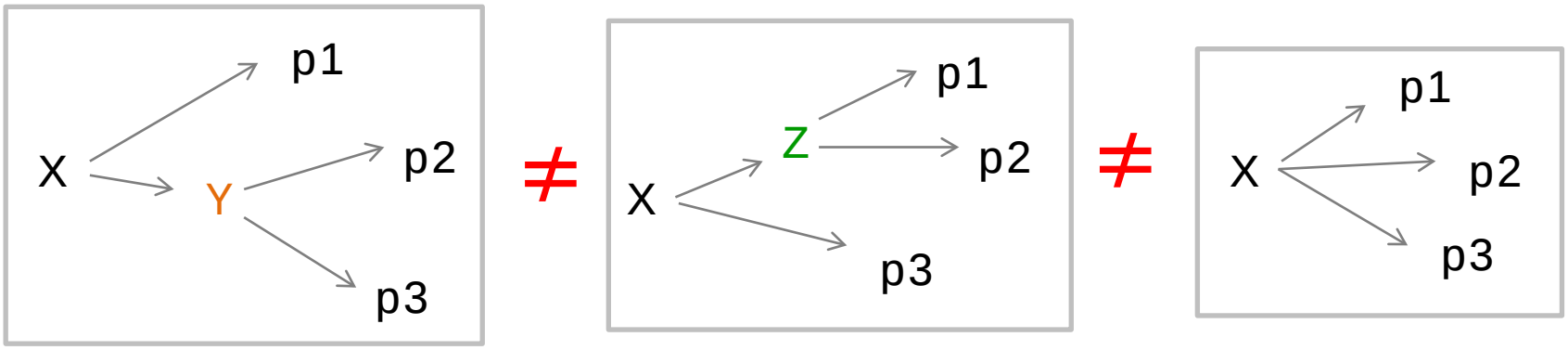
```
...
PndAnalysis *evr=new PndAnalysis();

TCandList charged, kaonLoose;

PndAnaPidSelector kaonSel(„KaonSelector“);
kaonSel.SetSelection(„KaonLoose“);            // set selection criterion

PndAnaPidCombiner pidComb(„PidCombiner“);
pidComb.SetTcaNames(„PidAlgoDrc;PidAlgoMvd“); // set algo‘s

while (evr->GetEvent())
{
  evr->FillList(charged, „Charged");         // start w/ charged candidates

  pidComb.Apply(charged);                   // copy P to candidates
  kaonSelector.Select(charged, kaonLoose);  // select kaons from charged
...
}
```

Mc Truth match

- Mc Truth for final state particles via TCandidate::GetMcIdx

```
evr->FillList(eplus,  "ElectronLoosePlus",  "PidAlgoEmcBayes");
evr->FillList(mcList, "McTruth");

TCandidate mc = mcList[eplus[i].GetMcIdx()];
```

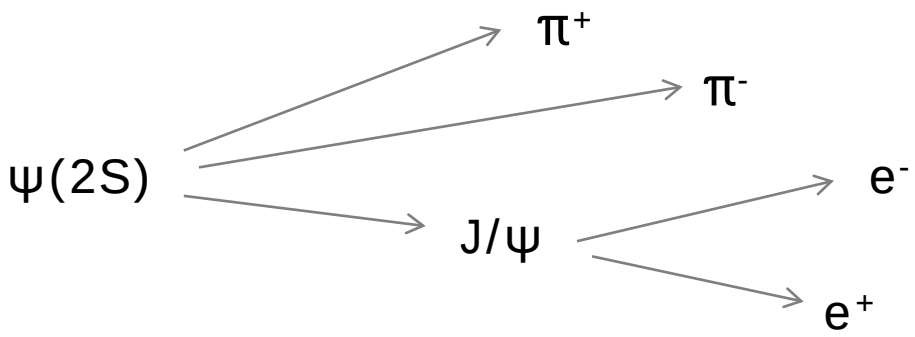
- Physics analysis might require a full mc truth match for decay trees for efficiency calculation, because



Requires PndEvtGenDirect with SetStoreTree!

Example: psi(2S) -> J/psi (e+ e-) pi+ pi-

- We want to reconstruct the following decay




```
evr->FillList( eplus,   „ElectronLoosePlus",  "PidAlgoEmcBayes");
evr->FillList( eminus,  „ElectronLooseMinus", "PidAlgoEmcBayes");
evr->FillList( piplus,  „PionLoosePlus",      "PidAlgoEmcBayes");
evr->FillList( piminus, „PionLooseMinus",     "PidAlgoEmcBayes");

evr->FillList( mcList,  "McTruth");

jpsi.Combine( eplus, eminus );
psi2s.Combine( jpsi, piplus, piminus );
```

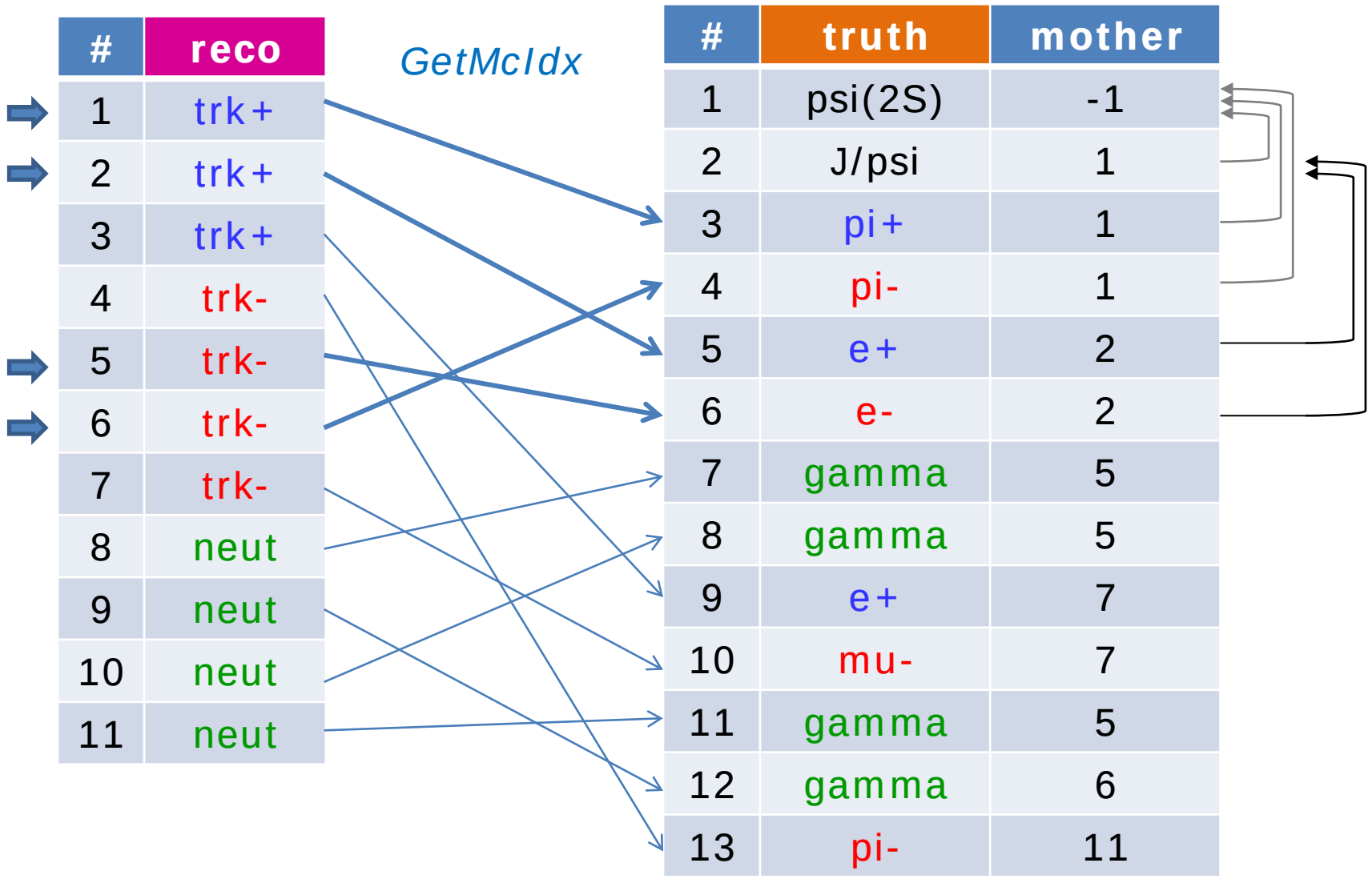
Example: psi(2S) -> J/psi (e+ e-) pi+ pi-

#	reco
1	trk+
2	trk+
3	trk+
4	trk-
5	trk-
6	trk-
7	trk-
8	neut
9	neut
10	neut
11	neut

GetMcIdx

#	truth	mother
1	psi(2S)	-1
2	J/psi	1
3	pi+	1
4	pi-	1
5	e+	2
6	e-	2
7	gamma	5
8	gamma	5
9	e+	7
10	mu-	7
11	gamma	5
12	gamma	6
13	pi-	11

Example: psi(2S) -> J/psi (e+ e-) pi+ pi-

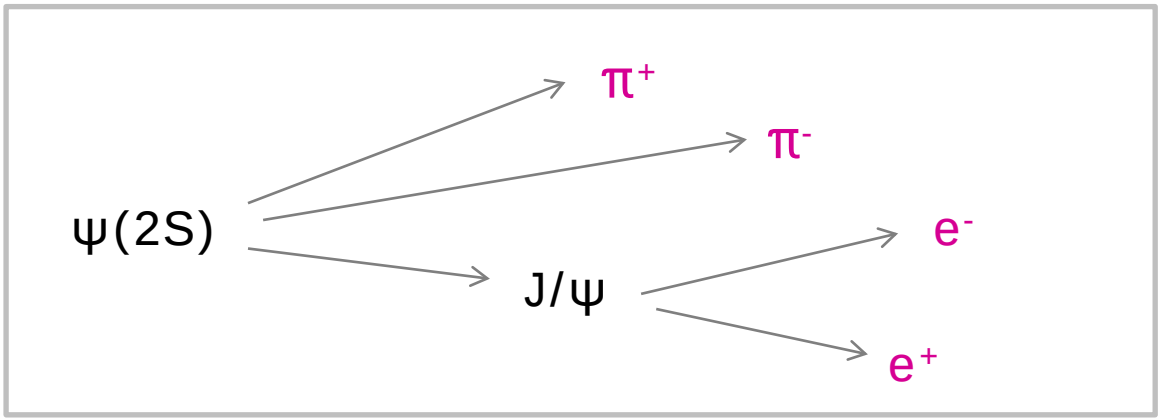


#	reco
1	trk+
2	trk+
3	trk+
4	trk-
5	trk-
6	trk-
7	trk-
8	neut
9	neut
10	neut
11	neut

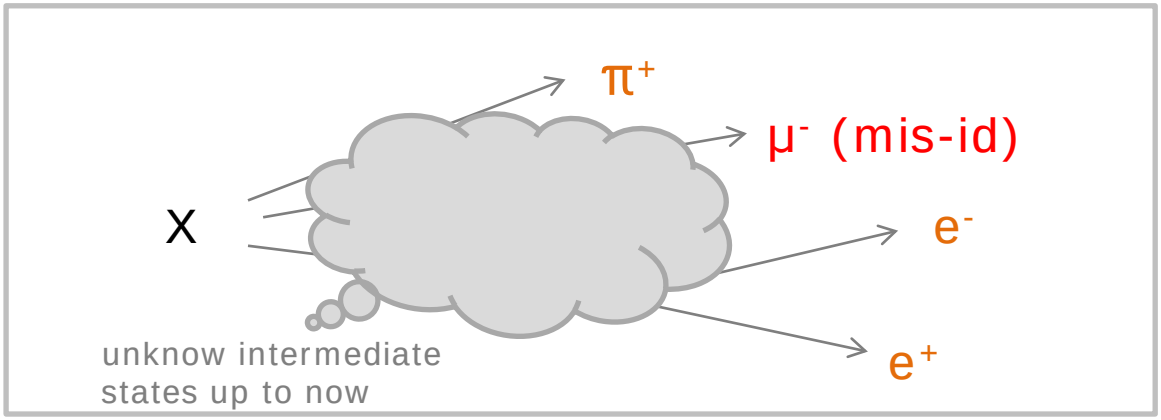
#	truth	mother
1	psi(2S)	-1
2	J/psi	1
3	pi+	1
4	pi-	1
5	e+	2
6	e-	2
7	gamma	5
8	gamma	5
9	e+	7
10	mu-	7
11	gamma	5
12	gamma	6
13	pi-	11

PndMcTruthMatch

- **Checklist** for full tree match:
  1. truth objects of final states have the correct PID types

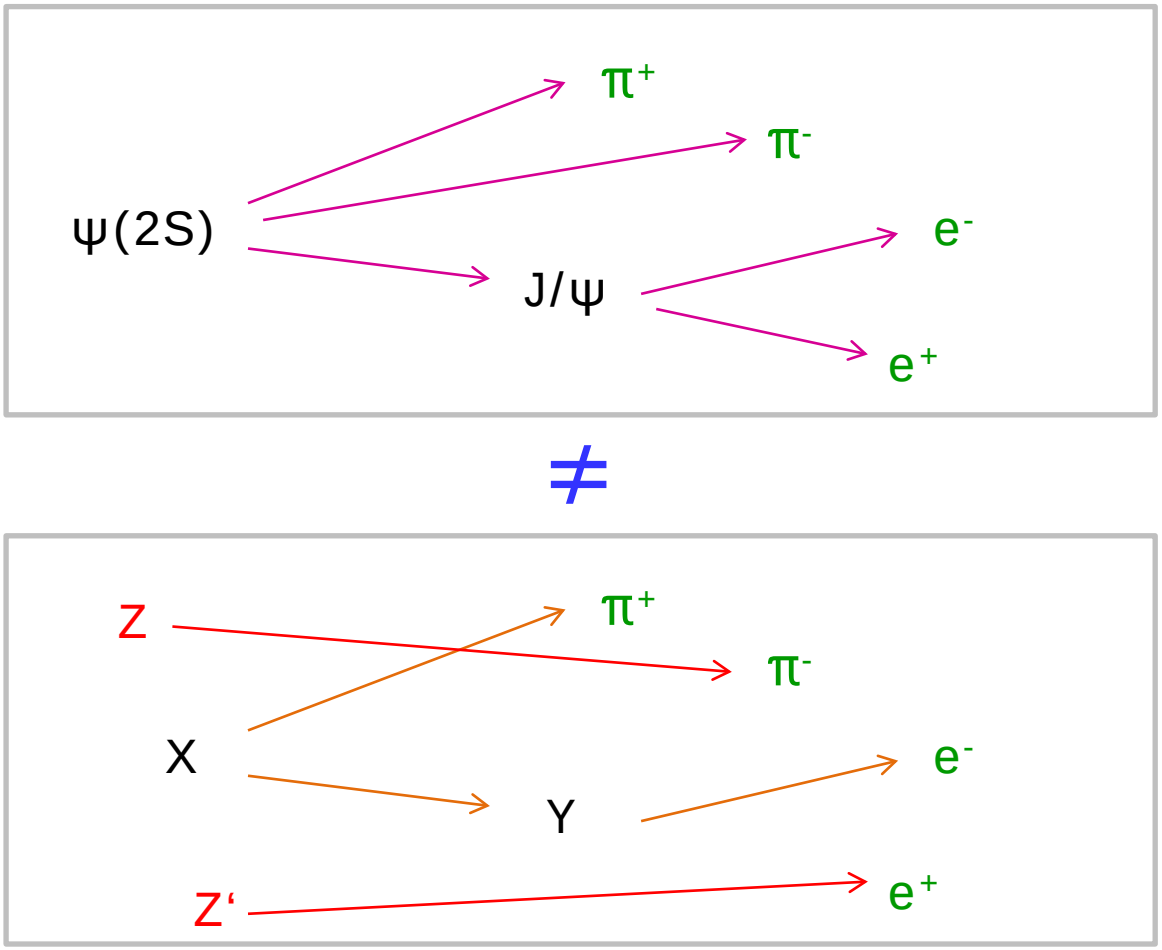


≠



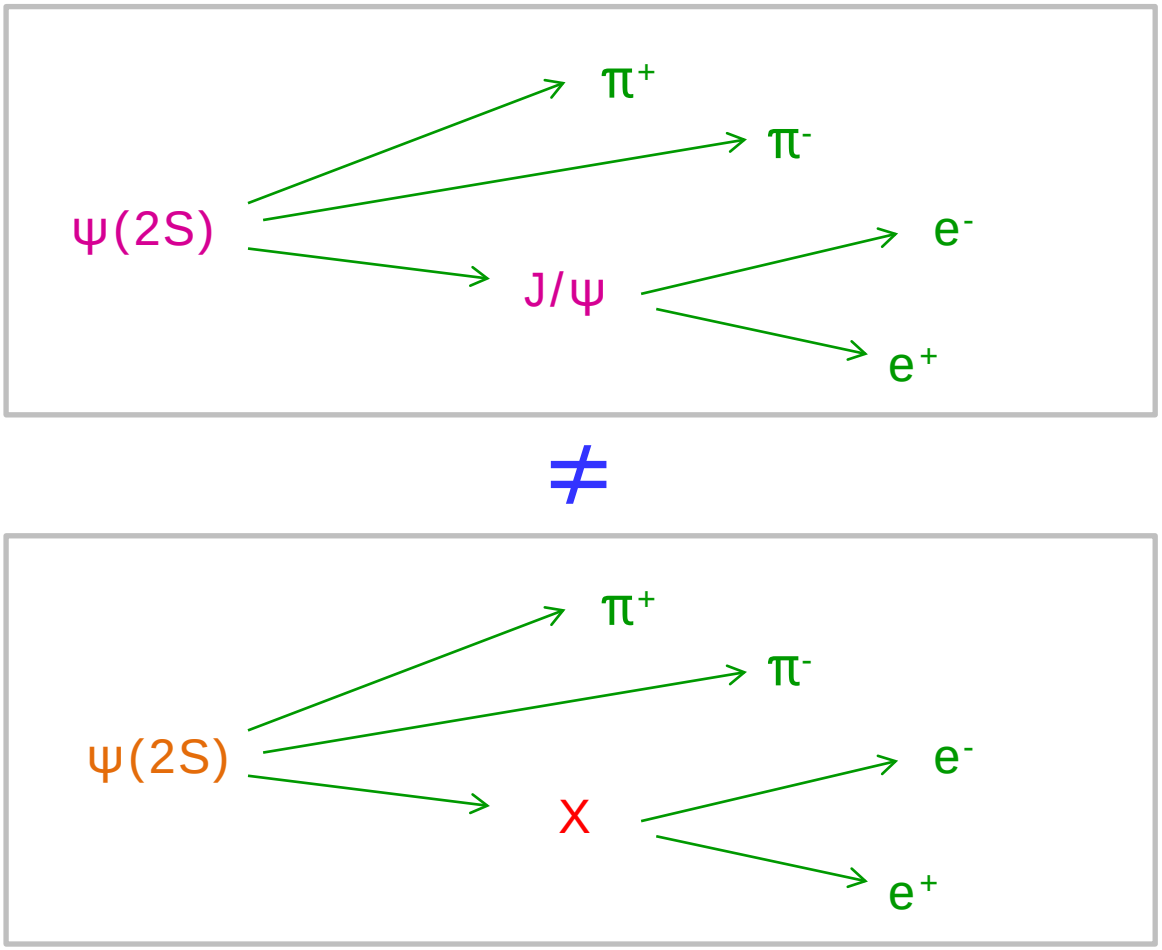
PndMcTruthMatch

- **Checklist** for full tree match:
  2. truth object of final states have the same mother



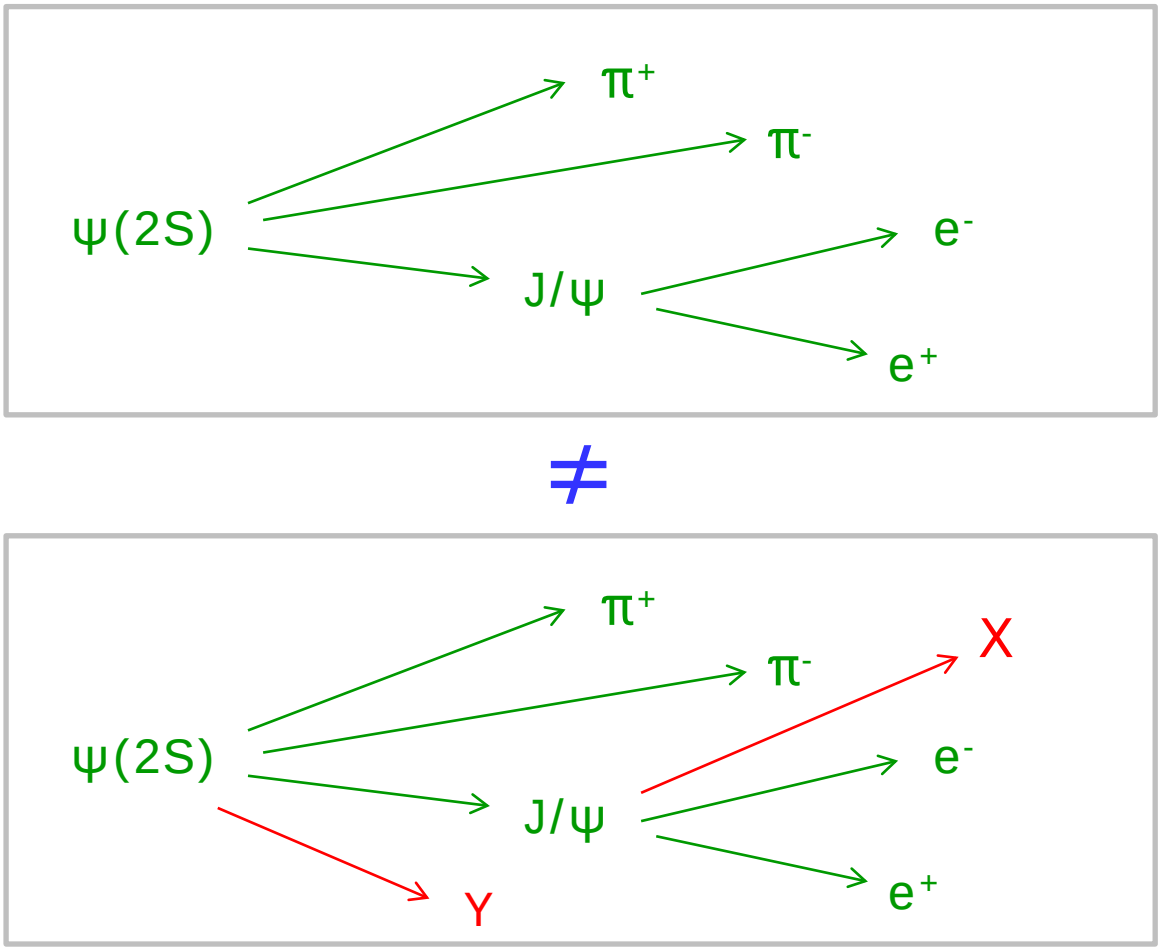
PndMcTruthMatch

- **Checklist** for full tree match:
  3. mother has required type



PndMcTruthMatch

- **Checklist** for full tree match:
  4. mother has correct number of daughters



Usage of PndMcTruthMatch

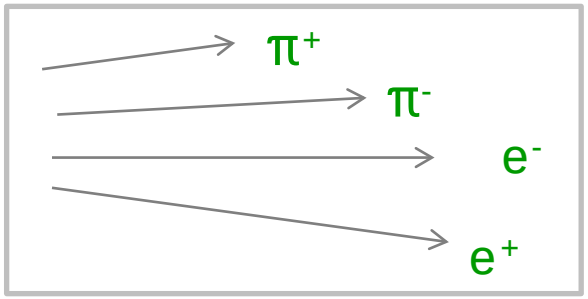
- Only one PndMcTruthMatch object is needed
- For matching, composite candidates have to have type set
- List with Mc truth particles is needed as second parameter

```
PndMcTruthMatch mcm;

while (evr->GetEvent())
{
  evr->FillList(eplus, "ElectronPlus");
  ...
  evr->FillList(mct, "McTruth");     // mc truth list needed for match

  jpsi.Combine(eplus, eminus);
  mcm.SetType(jpsi, "J/psi");       // set type for J/psi (names like EvtGen)

  psi2s.Combine(jpsi, piplus, piminus);
  mcm.SetType(psi2s, "psi(2S)");             // set type for psi(2S)

  bool match = mcm.MctMatch(psi[0], mct);    // perform the match

  TCandidate mcPsi;
  if (match) mcPsi = mct[psi(0).GetMcIdx()]; // access truth of composites!
```

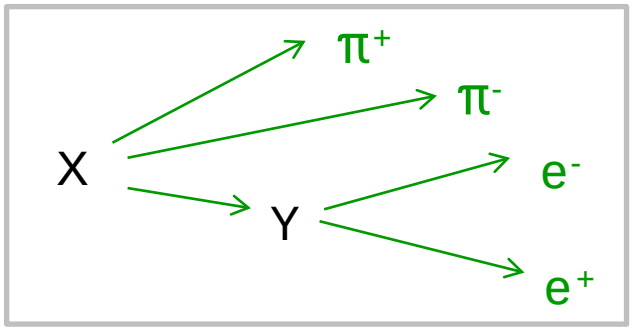
Different levels of matching

- Three different match levels are available via
  `mcm.MctMatch( psi[0], mct, level );`

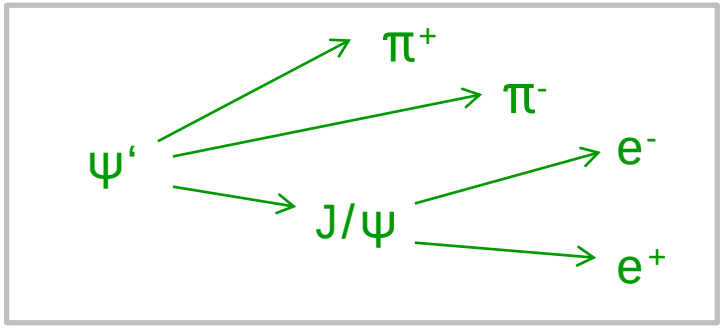
- Level = 0 only looks for correct PID



- Level = 1 matches topology in addition



- Level = 2 (default) is full match



Summary

- New Tutorial is available
  - Plan: Keep functioning (+ extend it) as standard tutorial
- *New Feature:* Extended (true) PID concept
  - Easy use with PndAnalysis::FillList
  - Stand-alone use possible
- *New Feature:* Monte Carlo Truth Match
  - simple to use
  - should work for all kinds of complicated decay trees
  - allows access to non-final-state MC truth resonances