Particle Identification

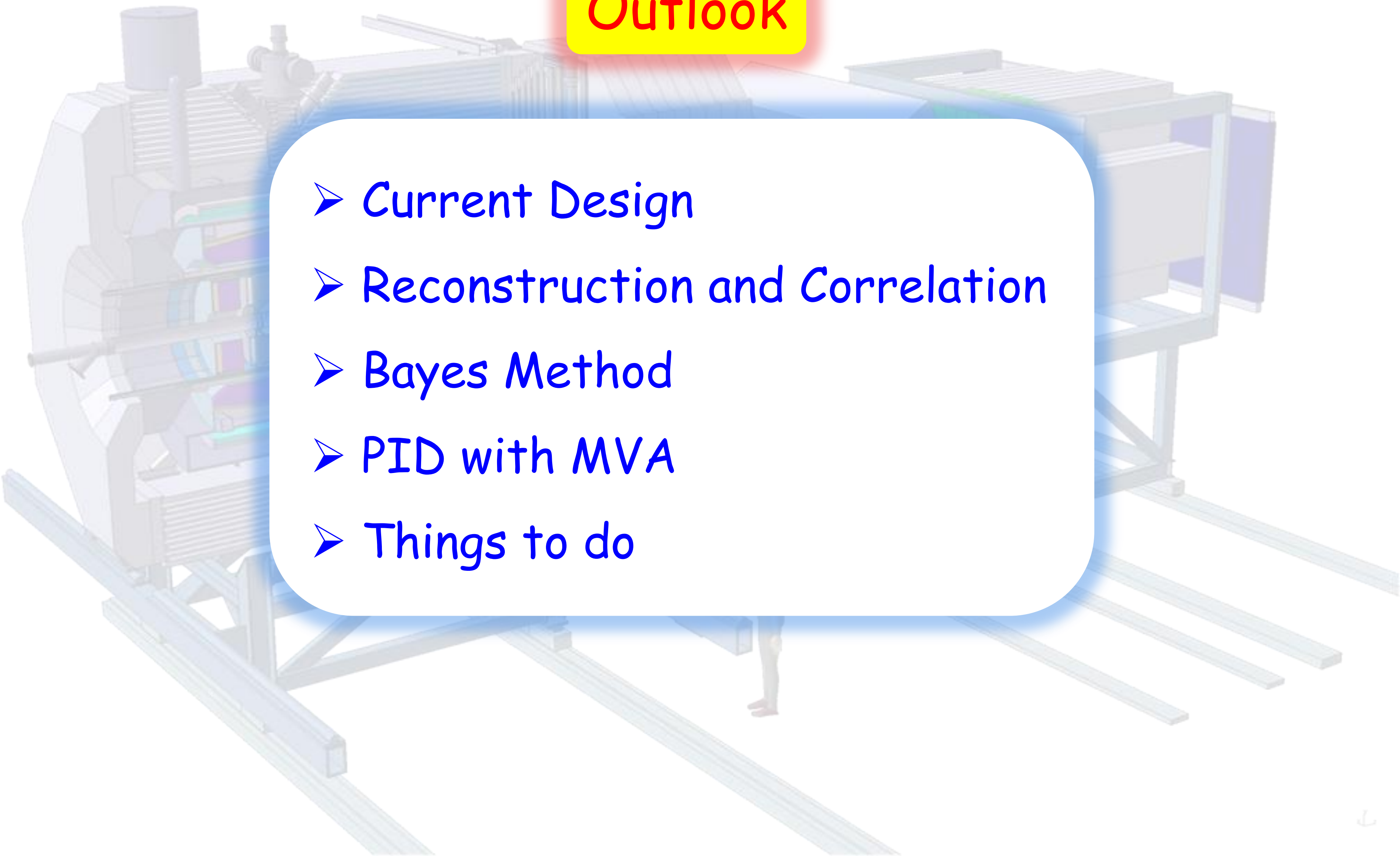
Stefano Spataro

Panda Computing Week
Torino, 23-27 July 2012

UNIVERSITÀ DEGLI STUDI DI TORINO
ALMA UNIVERSITAS TAURINENSIS

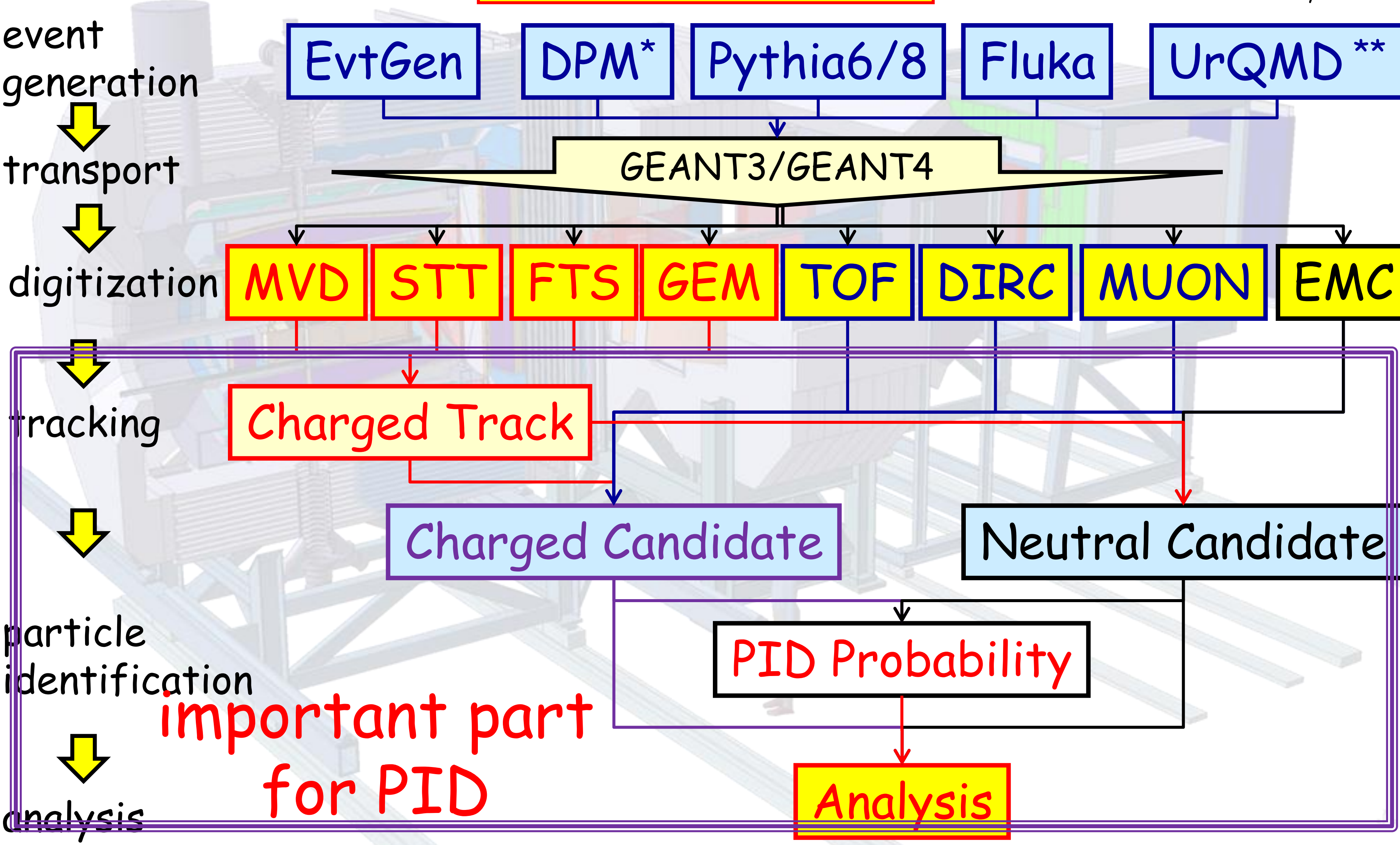
ISTITUTO NAZIONALE DI FISICA NUCLEARE
Sezione di Torino

Outlook



- Current Design
- Reconstruction and Correlation
- Bayes Method
- PID with MVA
- Things to do

Panda Data Flow



* Dual parton Model
** Ultra Relativistic Quantum Molecular Dynamics

PID Data Flow

PndTrack (CT)	PndTrack (CT+GENFIT)	PndTrack (FT)	PndTrack (FT+GENFIT)

PndPidCorrelator
- ✓ extrapolation to PID detectors
- ✓ retrieval PID information

PndPidCandidate (PidChargedCand)

PndPidCandidate (PidNeutralCand)

- ➢ Track parameters
- ➢ PID raw values
- ➢ Detector indexes

PndPidAssociatorTask
- ✓ probability/pdf

PndPidProbability
Bayes, MVA, HardCuts,…

➢ Probability/pdf for all the PID hypothesis

PndPidCandidate
PidChargedCand
PidNeutralCand

- Momentum
- Time-of-flight
- EMC energy
- EMC shower shape
- Cherenkov angle
- MVD dE/dx
- # Muon Layers
- …

PndPidProbability
MVD dE/dx
- P(e), P(π), P(K), …
- Candidate Index

PndPidProbability
DIRC θ_{ch}
- P(e), P(π), P(K), …
- Candidate Index

PndPidProbability
STT dE/dx
- P(e), P(π), P(K), …
- Candidate Index

PndPidProbability
Muon HardCuts
- P(e), P(π), P(K), …
- Candidate Index

PndPidProbability
MVA
- P(e), P(π), P(K), …
- Candidate Index

PndAnaPidCombiner → **Analysis**

Flexible way to switch between different algorithms

PndPidCandidate

PidChargedCand
- Momentum
- STT dE/dx
- EMC energy
- EMC shower shape
- Cherenkov angle
- MVD dE/dx
- # Muon Layers
- …

PidNeutralCand
- Momentum
- STT dE/dx
- EMC energy
- EMC shower shape
- Cherenkov angle
- MVD dE/dx
- # Muon Layers
- …

PndPidProbability

EMC SS
- P(e)
- P(π)
- P(μ)
- P(K)
- P(p)
- …

DIRC
- P(e)
- P(π)
- P(μ)
- P(K)
- P(p)
- …

MVD
- P(e)
- P(π)
- P(μ)
- P(K)
- P(p)
- …

…
- P(e)
- P(π)
- P(μ)
- P(K)
- P(p)
- …

PndAnaPidCombiner

```
analysis->FillList(pip, "PionLoose","BEMC;BMVD;BDIRC")
```

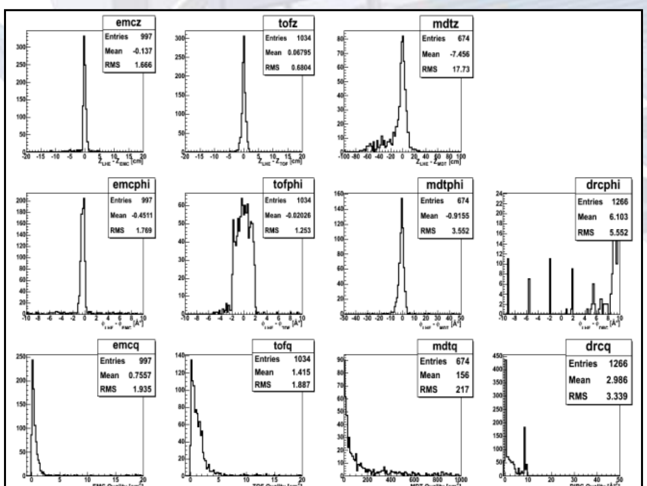
candidate list — **selection criterium** — **algorithms to merge**

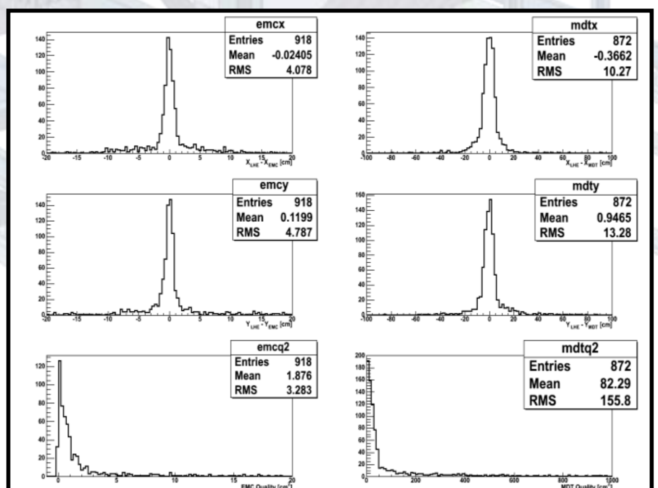
K. Götzen, R.Kliemt

Correlation: PndPidCorrelator

- correlates tracks to PID detectors → PidChargedCand
- finds not correlated EMC clusters → PidNeutralCand
- constructs PID informations

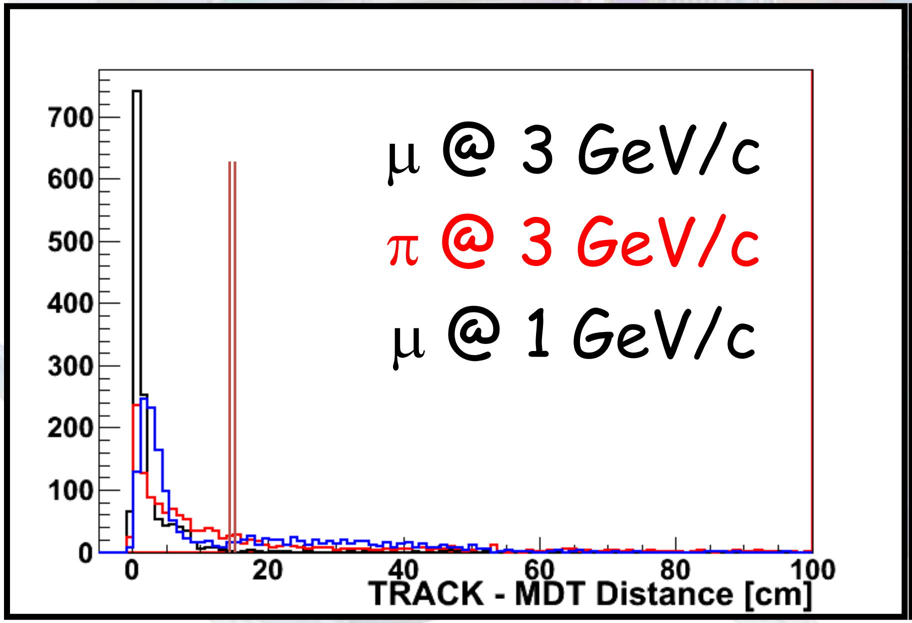
Track propagation → **GEANE** Propagate to PCA



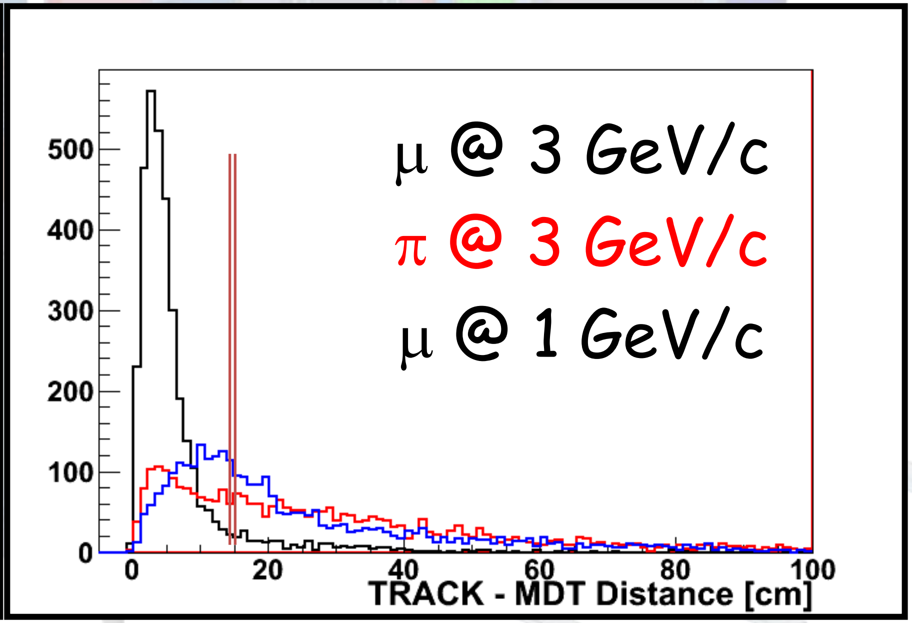


Extrapolation Residuals

ENDCAP + MF



BARREL



Momentum dependence!!

```
PndPidCorrelator* corr = new PndPidCorrelator();
 //corr->SetVerbose();
 corr->SetInputBranch("SttMvdGemGenTrack");
 corr->SetInputIDBranch("SttMvdGemGenTrackID");
 corr->SetInputBranch2("FtsIdealGenTrack");
 corr->SetInputIDBranch2("FtsIdealGenTrackID");
 //corr->SetDebugMode(kTRUE);
 //corr->SetFast(kTRUE);
 fRun->AddTask(corr);
```

Barrel tracking

Forward Tracking

No propagation, only tracks

macro/pid/run_pid_stt.C

PndPidCorrelatori.cxx::GetDrcInfo(...)

```
//______________________________________________________________________________
Bool_t PndPidCorrelator::GetDrcInfo(FairTrackParH* helix, PndPidCandidate* pidCand) {
  if ((helix->GetMomentum().Theta()*TMath::RadToDeg())<20.) return kFALSE;
  FairGeanePro *fProDrc = new FairGeanePro();
  if (!fCorrErrorProp) fProDrc->PropagateOnlyParameters();
  //---
  …
  for (Int_t dd = 0; dd<drcEntries; dd++)
    {
      drcHit = (PndDrcHit*)fDrcHit->At(dd);
      if ( fIdeal && ( ((PndDrcBarPoint*)fDrcPoint->At(drcHit->GetRefIndex()))->GetTrackID() !=pidCand->GetMcIndex()) ) continue;
      drcHit->Position(drcPos);

      if (fGeanePro) // Overwrites vertex if Geane is used
        {
          fProDrc->PropagateToVolume("DrcBase",0,1);
          vertex.SetXYZ(-10000, -10000, -10000); // reset vertex
          FairTrackParH *fRes= new FairTrackParH();
          Bool_t rc =  fProDrc->Propagate(helix, fRes, fPidHyp*pidCand->GetCharge());
          if (!rc) continue;
          vertex.SetXYZ(fRes->GetX(), fRes->GetY(), 0.);
          drcGLength = fProDrc->GetLengthAtPCA();
        }

      Float_t dphi = vertex.DeltaPhi(drcPos);
      Float_t dist = dphi * dphi;

      if ( drcQuality > dist)
        {
          drcIndex = dd;
          drcQuality = dist;
          drcThetaC = drcHit->GetThetaC();
          drcThetaCErr = drcHit->GetErrThetaC();
          drcPhot = 0; // ** to be filled **
        }
    }

  if ((drcQuality<fCorrPar->GetDrcCut()) || (fIdeal && drcIndex!=-1))
{
      pidCand->SetDrcQuality(drcQuality);
      pidCand->SetDrcThetaC(drcThetaC);
      pidCand->SetDrcThetaCErr(drcThetaCErr);
      pidCand->SetDrcNumberOfPhotons(drcPhot);
      pidCand->SetDrcIndex(drcIndex);
    }
  return kTRUE;
}
```

Implementation: Probability and Classifiers

pid/PidClassifier

PndPidXxxAssociatorTask

pid/PidData

PndPidCandidate

pid/PidData

PndPidProbability

- GetElectronPidProb()
- GetMuonPidProb()
- GetPionPidProb()
- GetKaonPidProb()
- GetProtonPidProb()

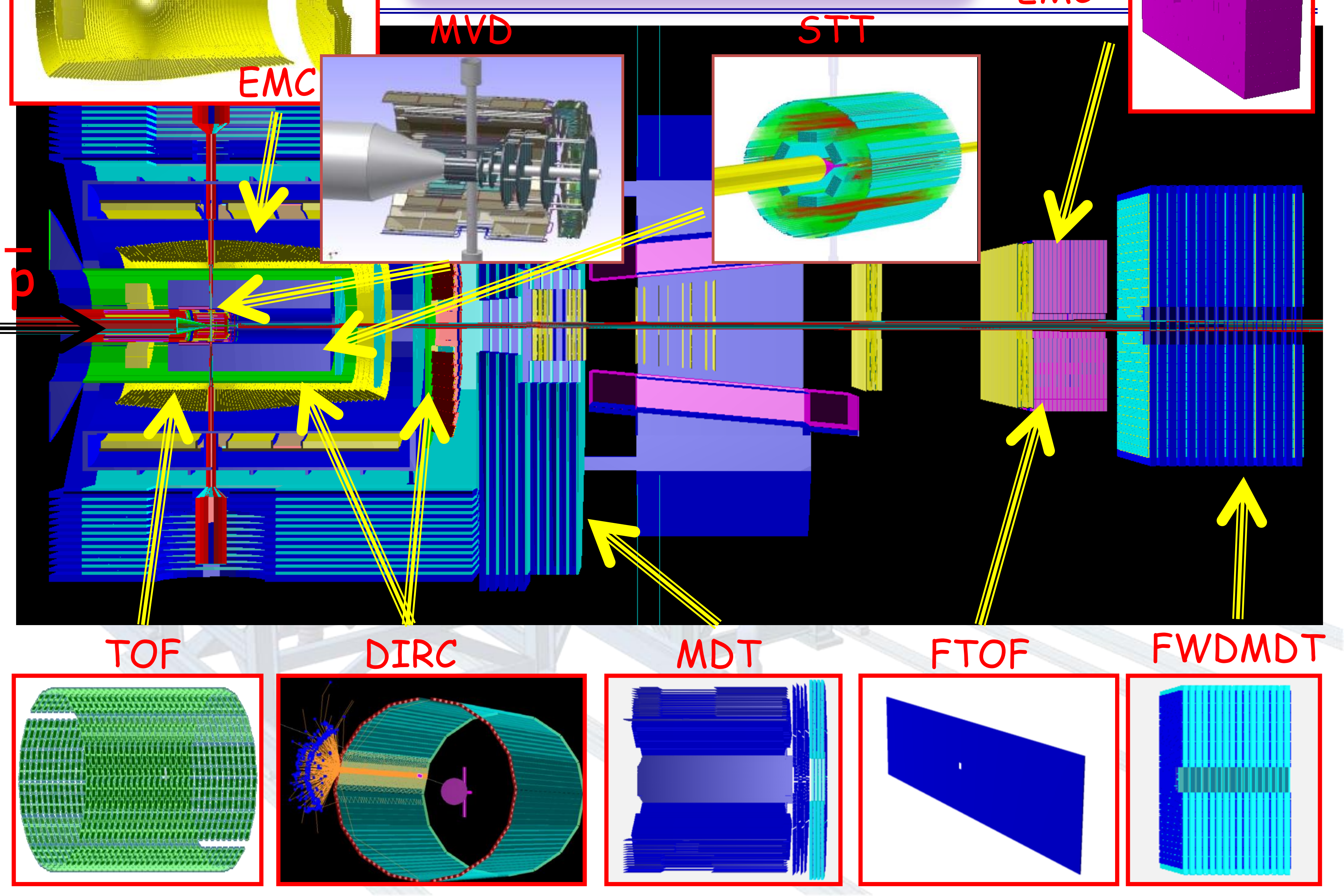
PndPidIdealAssociatorTask

associates candidates with MonteCarlo PID

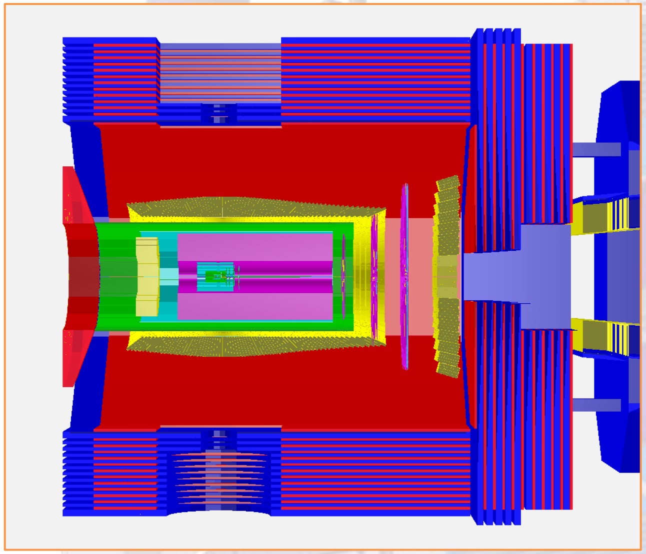
```
PndPidIdealAssociatorTask *assMC= new PndPidIdealAssociatorTask();
fRun->AddTask(assMC);

PndPidMvdAssociatorTask *assMvd= new PndPidMvdAssociatorTask();
fRun->AddTask(assMvd);

PndPidMdtHCAssociatorTask *assMdt= new PndPidMdtHCAssociatorTask();
fRun->AddTask(assMdt);

PndPidDrcAssociatorTask *assDrc= new PndPidDrcAssociatorTask();
fRun->AddTask(assDrc);

PndPidDiscAssociatorTask *assDisc= new PndPidDiscAssociatorTask();
fRun->AddTask(assDisc);

PndPidSttAssociatorTask *assStt= new PndPidSttAssociatorTask();
fRun->AddTask(assStt);

PndPidEmcBayesAssociatorTask *assEMC= new PndPidEmcBayesAssociatorTask();
fRun->AddTask(assEMC);
```

macro/pid/run_pid_stt.C

Particle Identification



Requirements for Particle Identification



different detectors for PID

covering

different momentum/angle ranges

MVD, TPC/STT, Cherenkov, EMC, MDT…

- handling of different PID signals (dE/dx, θ_C, EMC shower, …)
- combining several PID detectors to improve identification
- if one detector does not contribute to PID, it should not decrease the identification perfomances

Requirements for Particle Identification (II)

- handling of different PID signals (dE/dx, θ_C, EMC shower, …)
- combining several PID detectors to improve identification
- if one detector does not contribute to PID, it should not decrease the identification perfomances

- PID procedure should be as much as possible automatic
- PID depends also on analysis

we need to separate

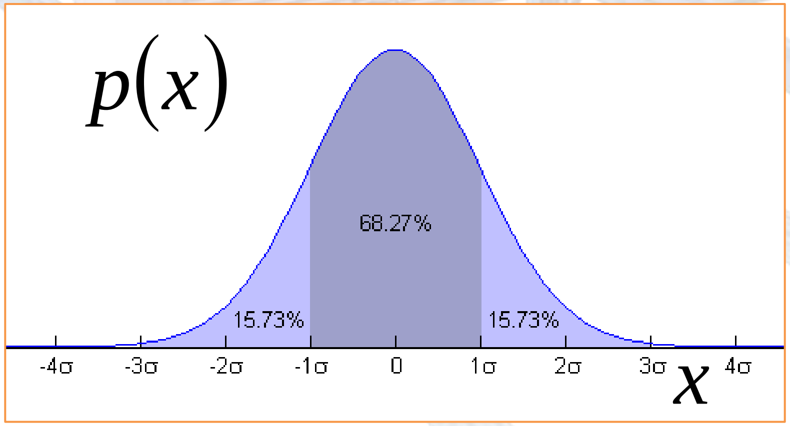
- Detector response (i.e. resolution)
- Event/track selection (analysis)

Probability Density Function

a function that describes the relative likelihood for a random variable to occur at a given point in the observation space (Wikipedia)

i.e. Gaussian distribution

Probability Density Function



probability to find a variable in a given range [a,b]

$$P[a,b] = \frac{\int_a^b p(x)dx}{\int_{-\infty}^{+\infty} p(x)dx}$$

Probability Density Function - II

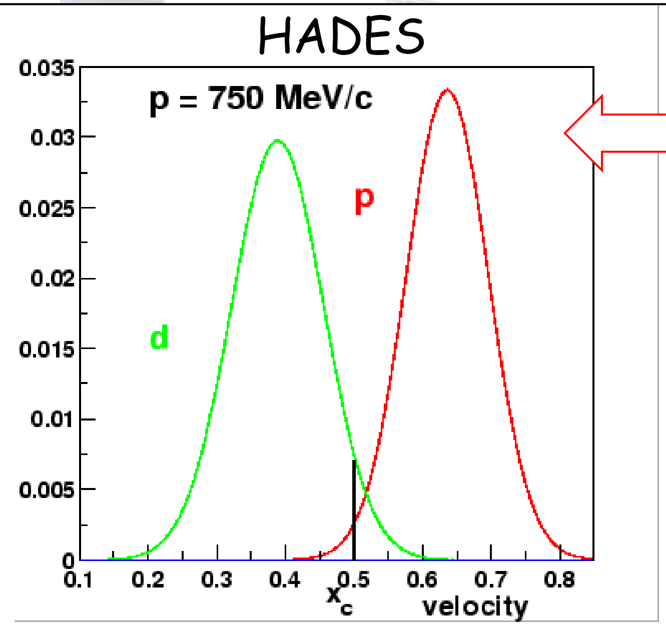


Fig.1 Probability density functions of velocities for protons and deuterons at particle momentum 750 MeV/c

gaussian distributions

For each particle hypothesis calculation of (normalized) pdf

- from simulation
- from experimental data

x – signal (p, dE/dx, θ_C...)

h – particle hyp (e, μ, π, K, p)

$$p(\vec{x}, h)$$

depends on detector response

The Bayes Theorem

If many detectors/algorithms constributing to PID

Global Likelihood

$$L(\vec{x}\,|\,h)=\prod_k p_k(\vec{x}\,|\,h)$$

k = MVD dE/dx, DRC θ_C...

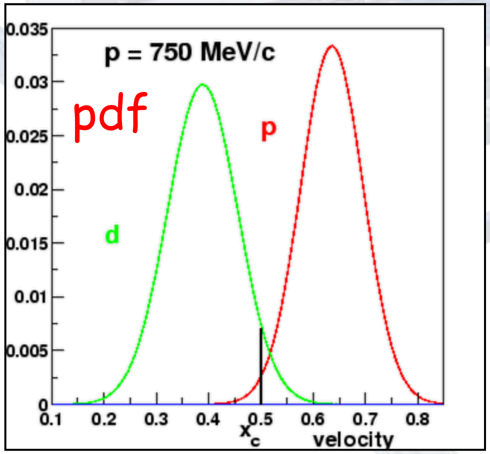
Probability that a given track with given params x corresponds to particle type h

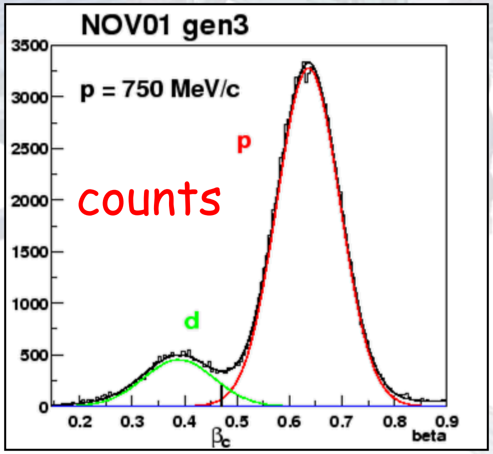
$$P(\vec{x}\,|\,h)=\frac{L(\vec{x}\,|\,h)\times P(h)}{\sum_{h=e,\mu,\pi,K,p} L(\vec{x}\,|\,h)\times P(h)}$$

The Bayes Theorem - II

$$P(\vec{x} \mid h) = \frac{L(\vec{x} \mid h) \times P(h)}{\sum_{h=e,\mu,\pi,K,p} L(\vec{x} \mid h) \times P(h)}$$

$P(h)$ *apriori* probability to find the particle kind h in the detector





P(h) depends only on track/event selection

Algorithms: Bayesian Method

Global Likelihood

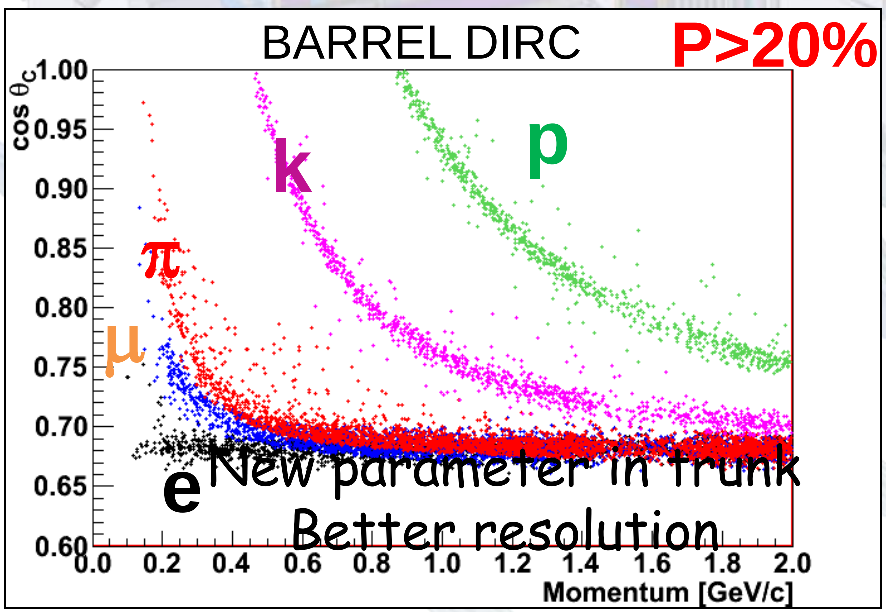
$$L(\vec{x}\,|\,h) = \prod_k p_k(\vec{x}\,|\,h)$$

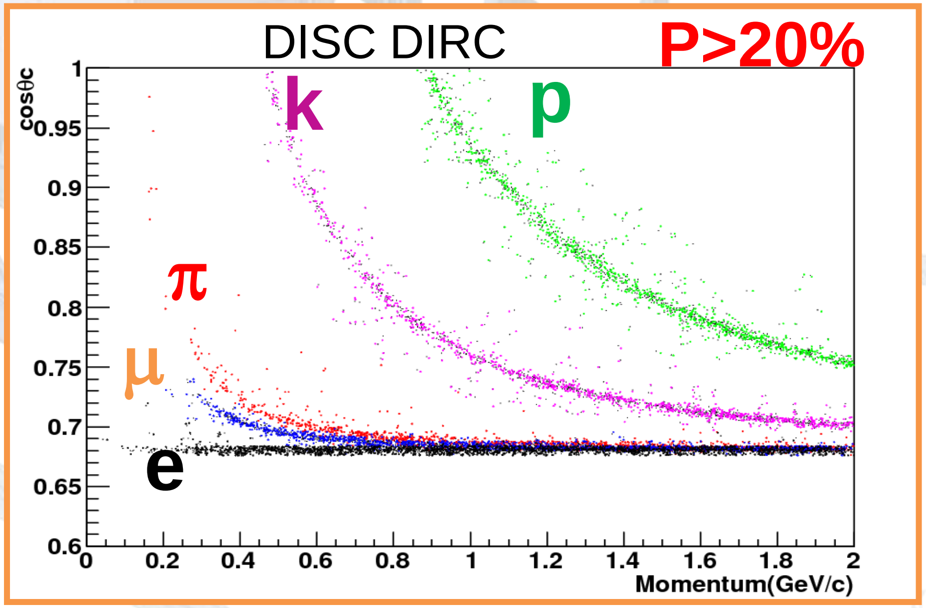
algorithms

$$P(\vec{x}\,|\,h) = \frac{L(\vec{x}\,|\,h)\times P(h)}{\sum_{h=e,\mu,\pi,K,p} L(\vec{x}\,|\,h)\times P(h)}$$

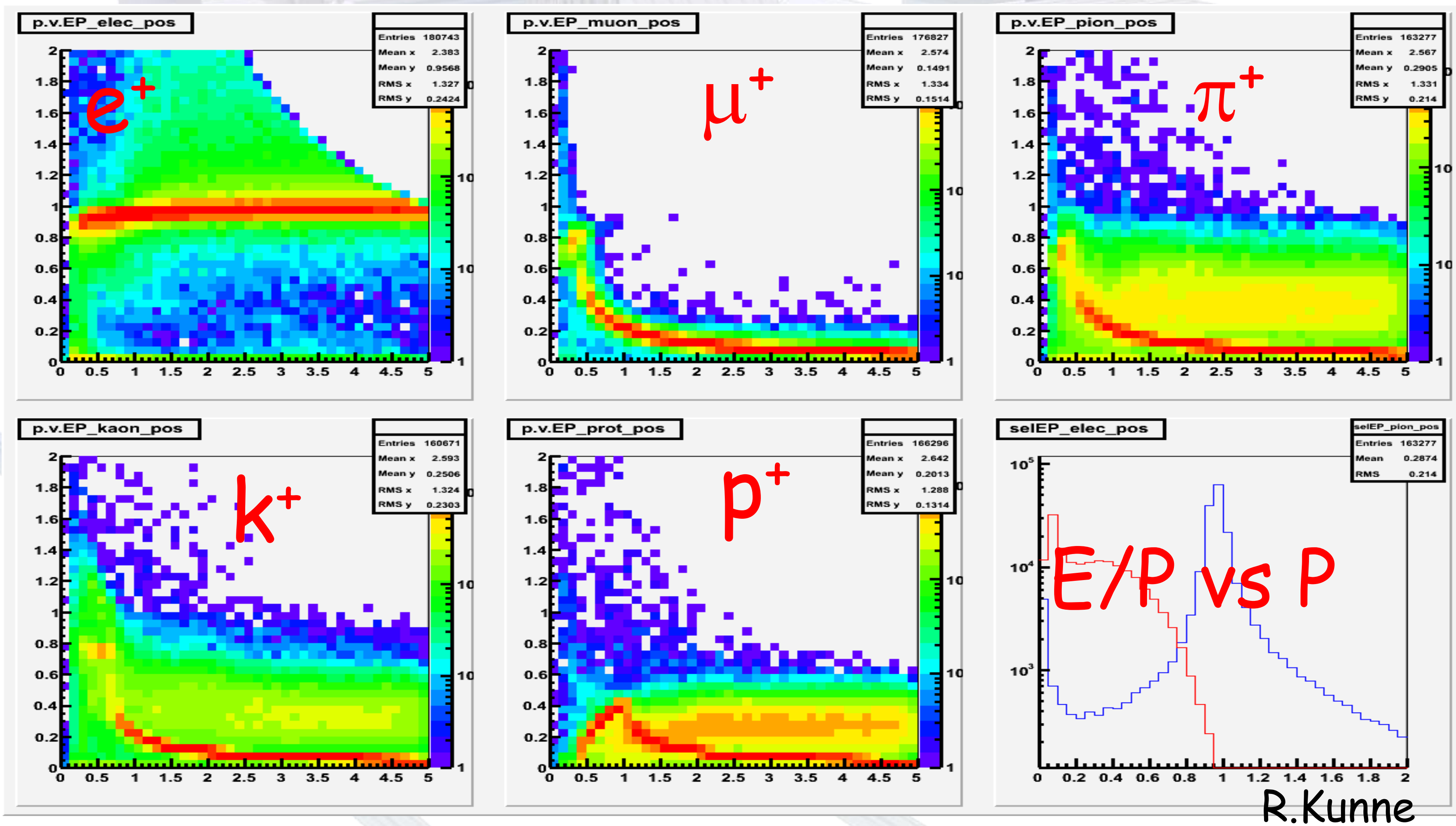
probability

particle flux

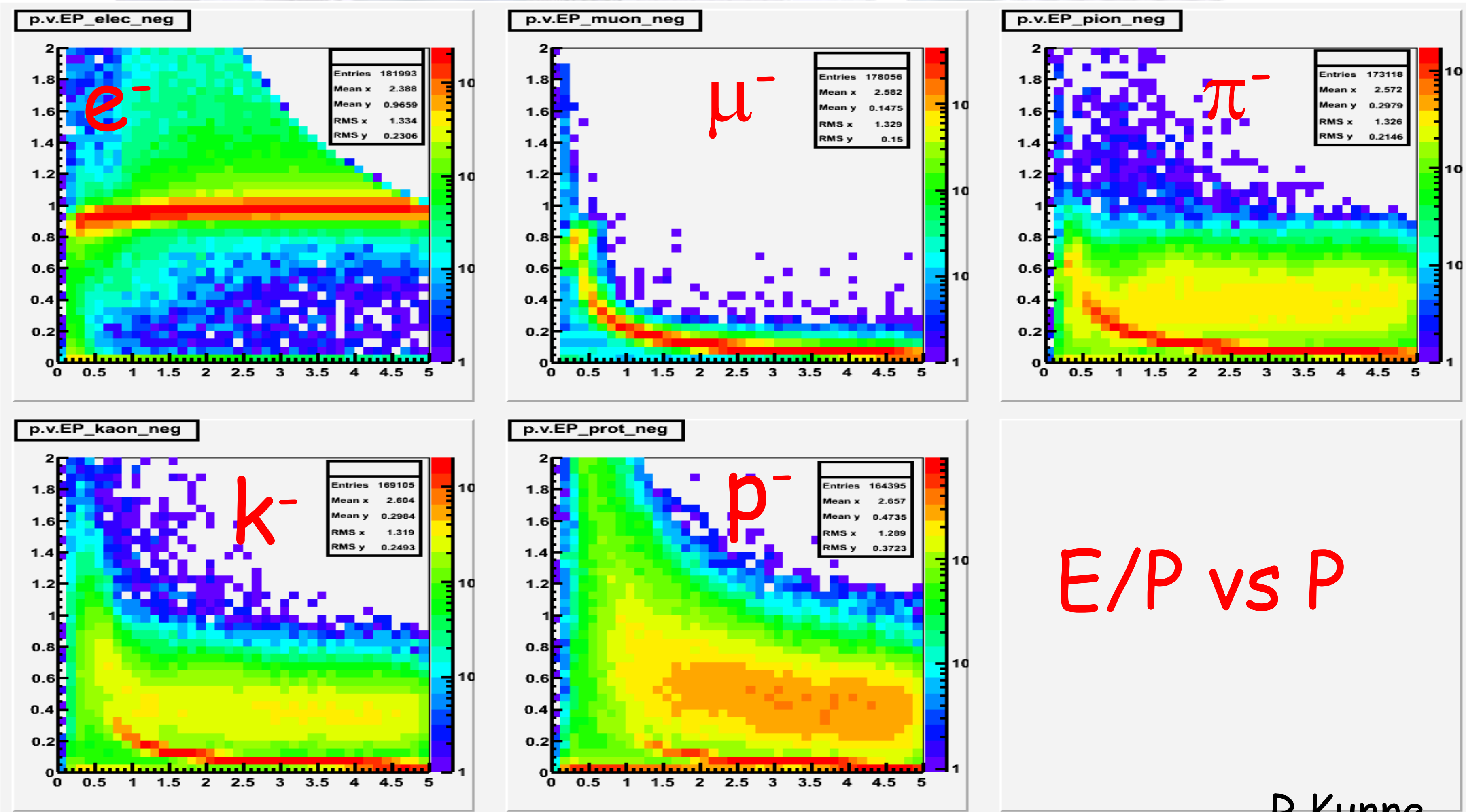




Particle Identification with EMC

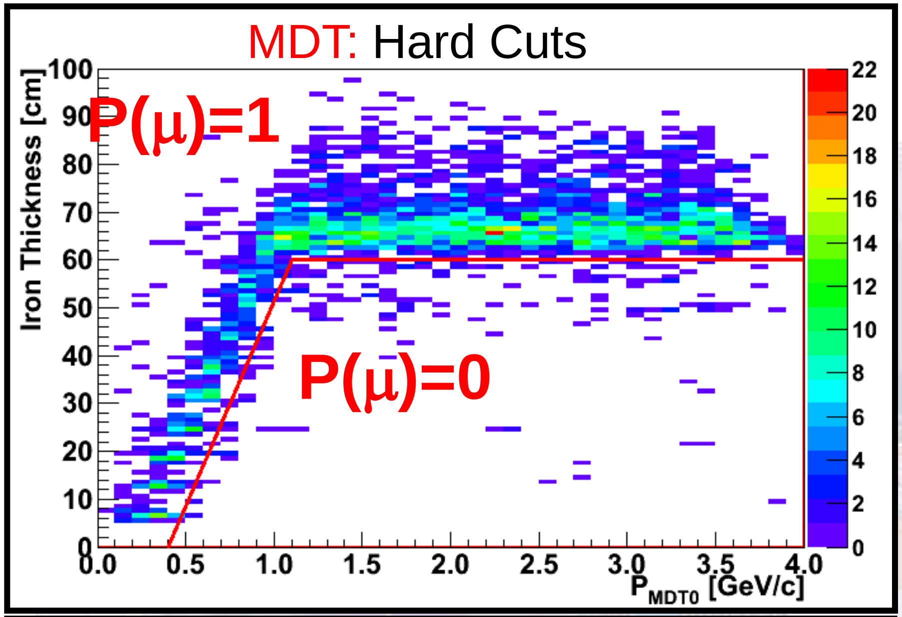


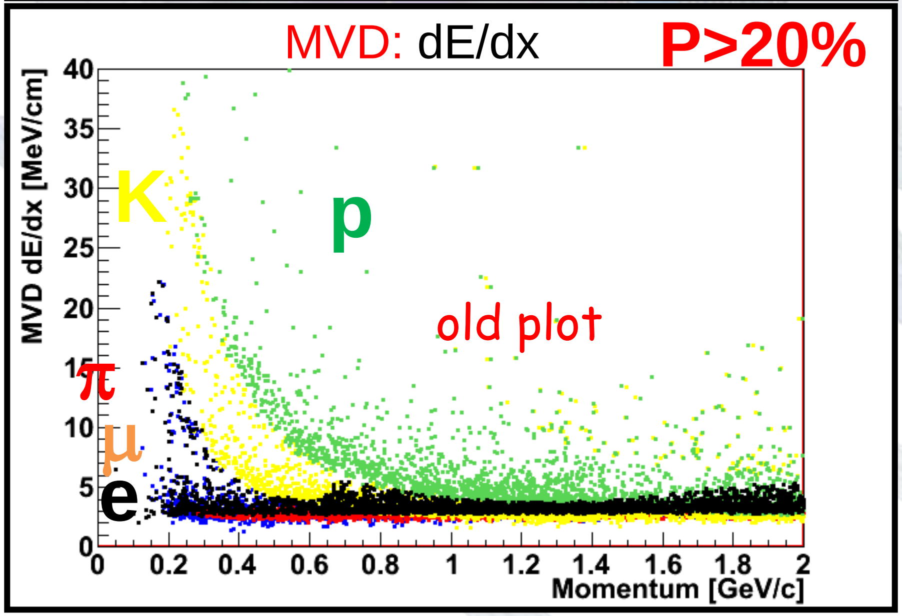
Particle Identification with EMC

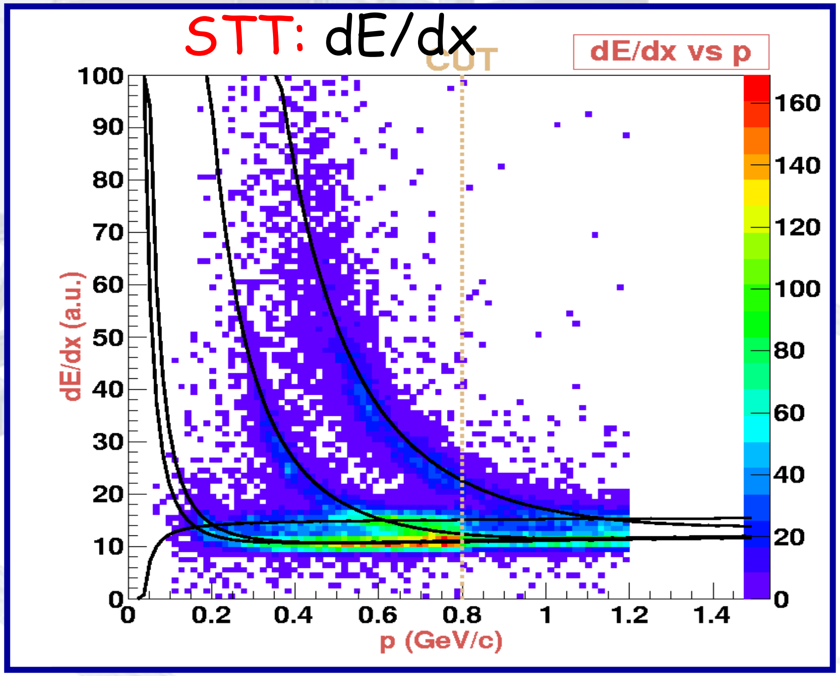


Particle Identification

Algorithms
Bayesian Method/Hard Cuts

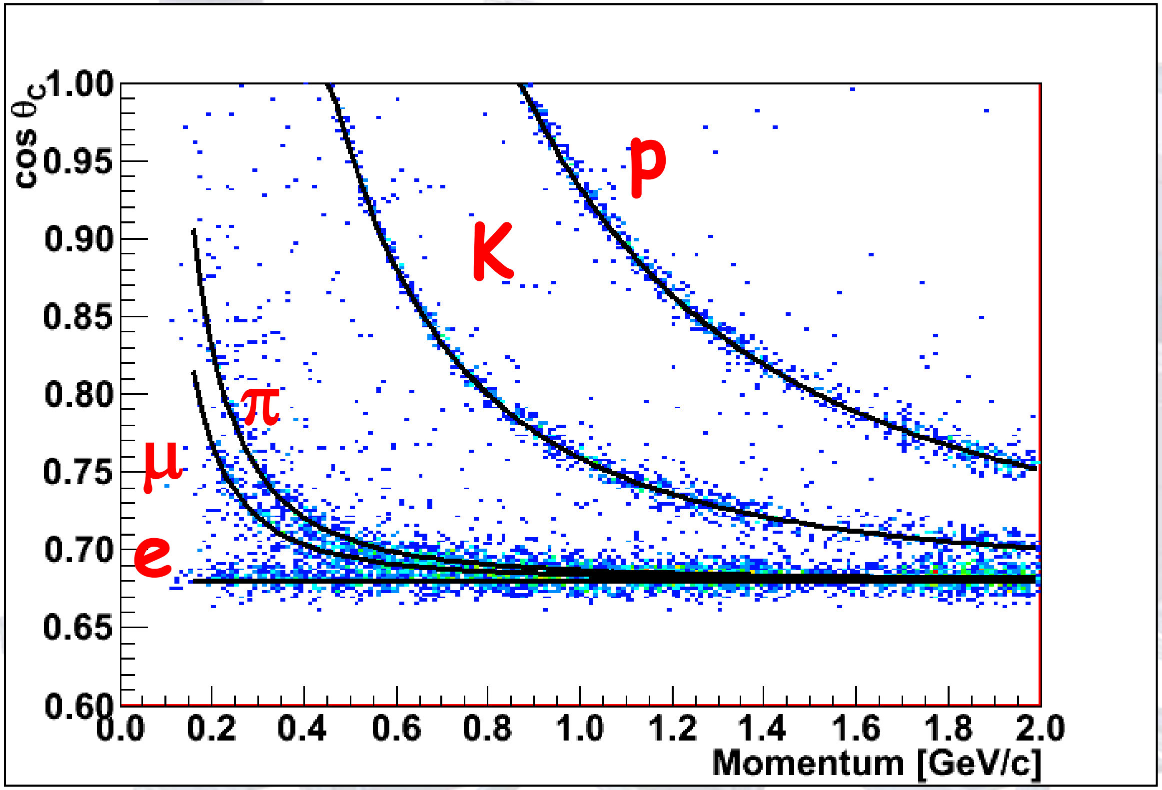






Particle Identification

DIRC: PndPidDrcAssociatorTask (myself)



2000 e, μ, π, K, p

p [0.2, 2.0] GeV/c

θ [20°, 120°]

ϕ [0°, 360°]

pdf: Gaus

center

$\cos\theta_C = 1/n\beta$

sigma

parametrization

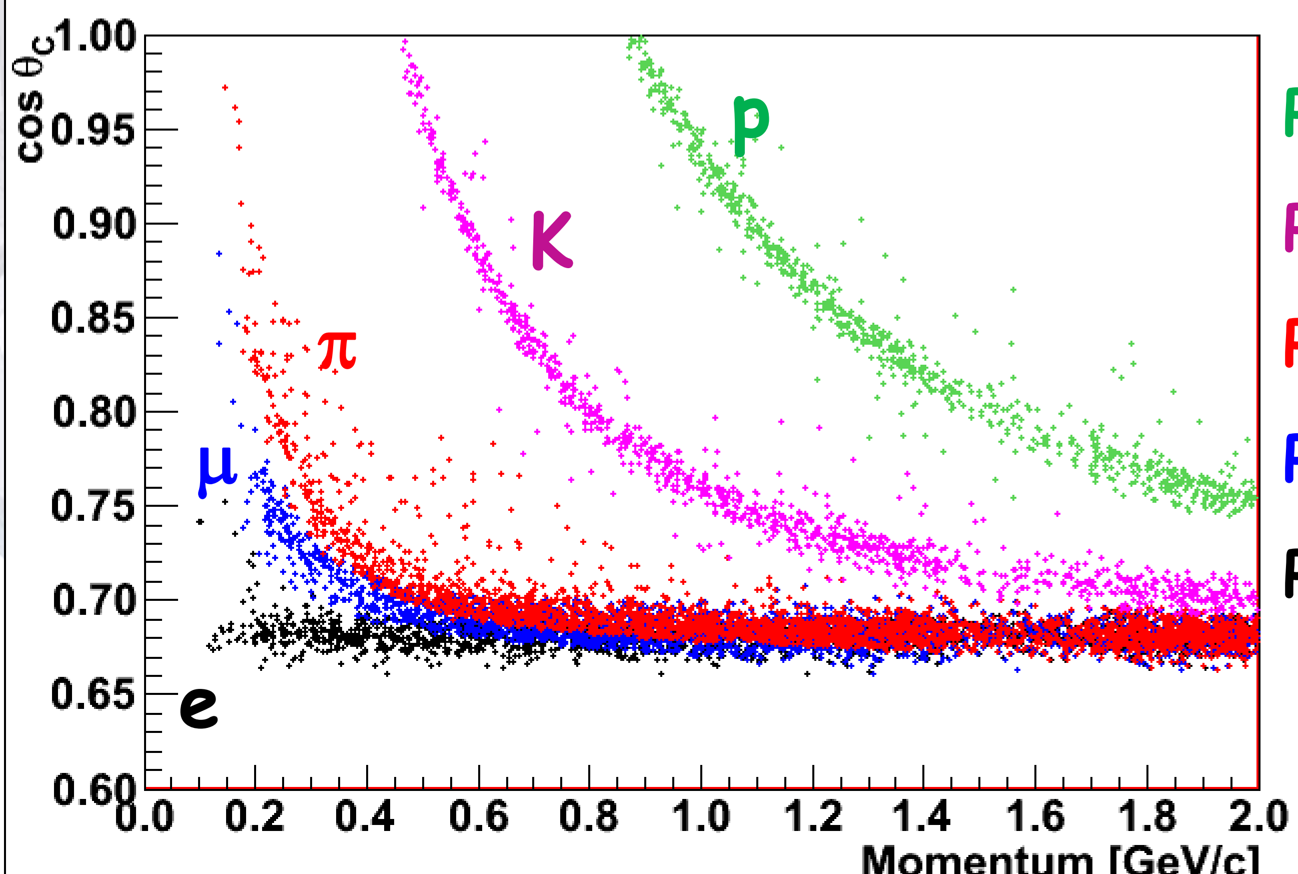
$\sigma(e) = \sigma(\mu) = \sigma(\pi) = 0.006$

$\sigma(K) = \sigma(p) = 0.005$

Particle Identification with DIRC

PndPidProbability::GetXxxProb()
PndPidProbability::GetXxxPdf()

Xxx
Electron, Muon, Pion, Kaon, Proton

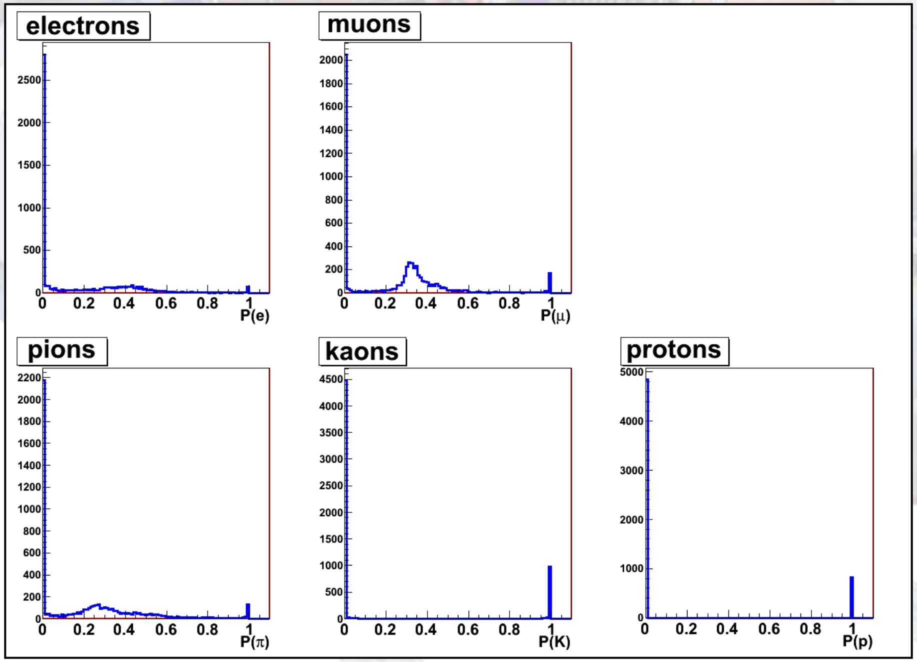


P(p)>0.2
P(K)>0.2
P(π)>0.2
P(μ)>0.2
P(e)>0.2

Combined Particle Identification

macro/pid/pid_check.C

```
PndPidProbability *drc = (PndPidProbability *)drc_array->At(ii);
PndPidProbability *mvd = (PndPidProbability *)mvd_array->At(ii);
PndPidProbability *combo = (*drc) * (*mvd);
```



combining
different algorithms

simple multiplication

Particle Flux

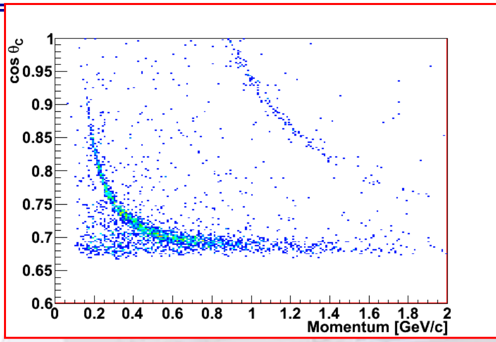
$$P(\vec{x} \mid h) = \frac{L(\vec{x} \mid h) \times P(h)}{\sum_{h=e,\mu,\pi,K,p} L(\vec{x} \mid h) \times P(h)}$$

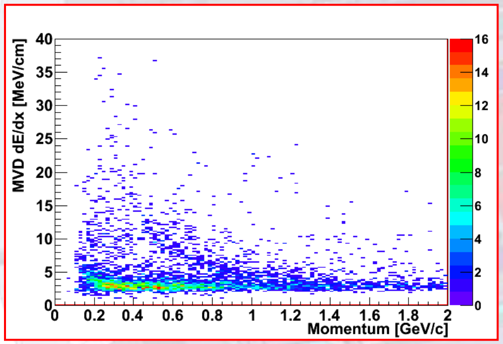
P(h) depends only on track/event selection

PndPidProbability::GetXxxProb(PndPidProbability* flux)

PndPidProbability::GetXxxProb() → default

PndPidProbability* flux = new PndPidProbability(1,1,1,1,1))





Particle Flux - DPM

2000 events @ 6 GeV/c

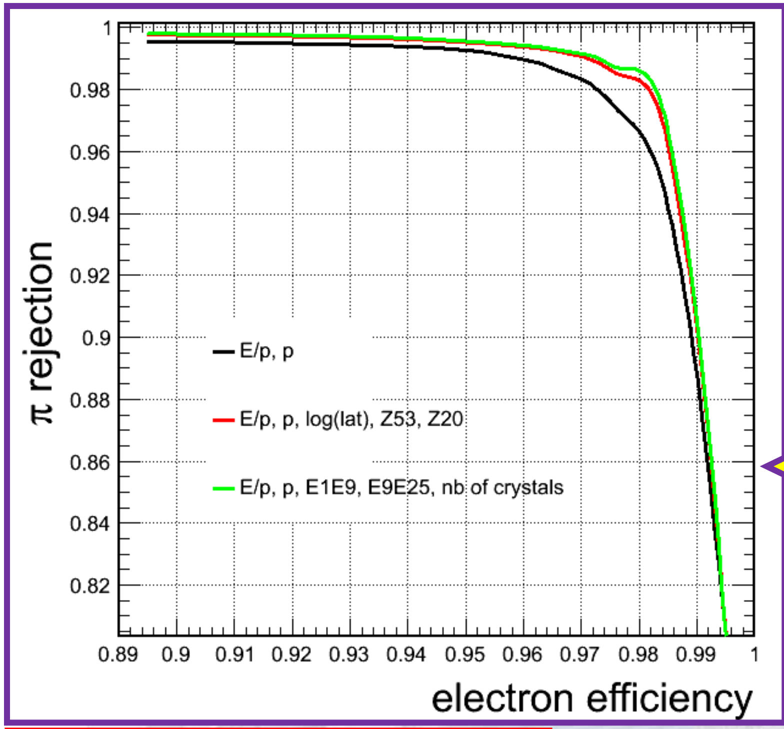
no elastic

particle yields
primary + secondary

	-	+
e	239	237
μ	114	101
π	2282	2375
K	42	35
p	517	1052

```
PndPidProbability *flux = new PndPidProbability(239+237,114+101,2282+2375,35+42,517+1052)
```

MultiVariate Particle Identification



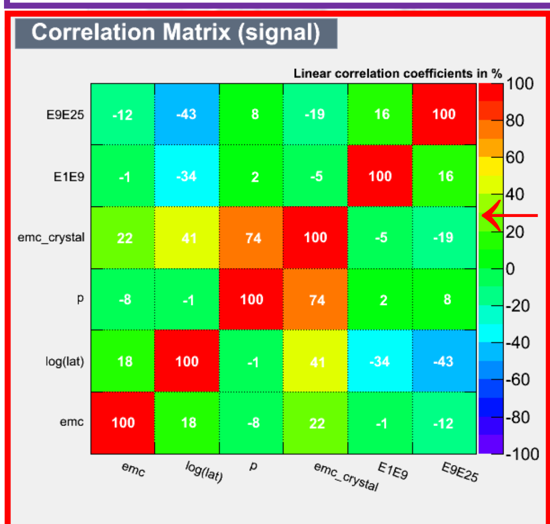
implementation of TMVA methods

- EMC shower shape analysis

e/π separation in EMC

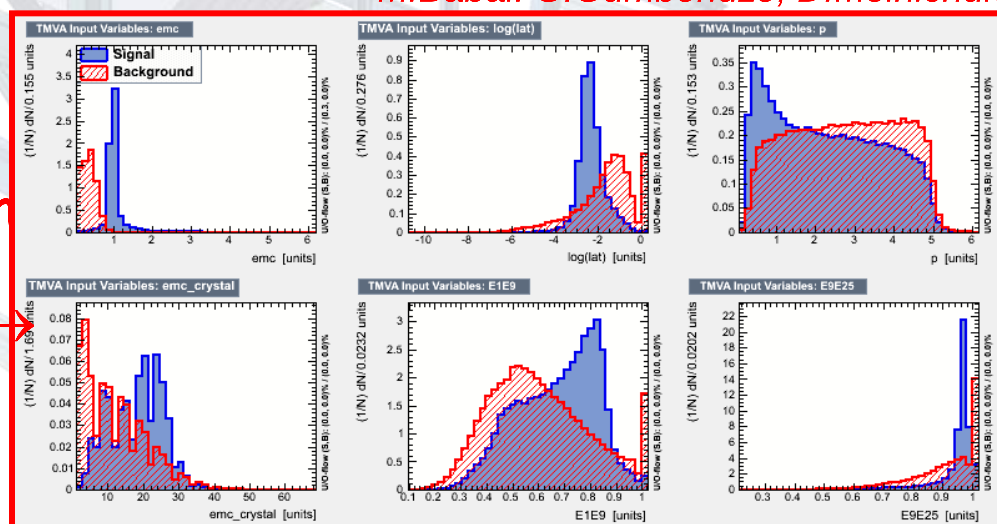
preliminary tests for muons (MDT)

M.Babai. G.Gumberidze, D.Melnichuk



Correlation

Variables



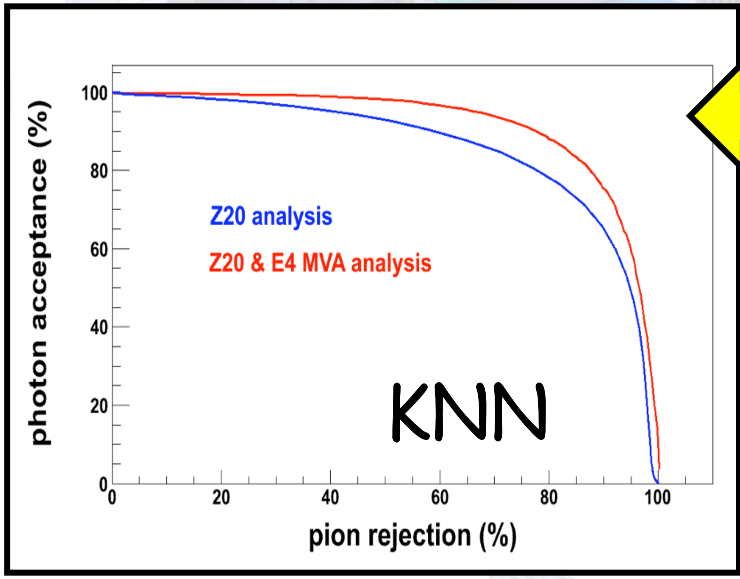
Multivariate Particle Identification

implementation of TMVA methods

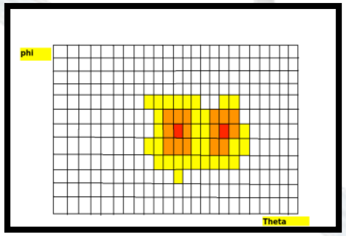
well promising

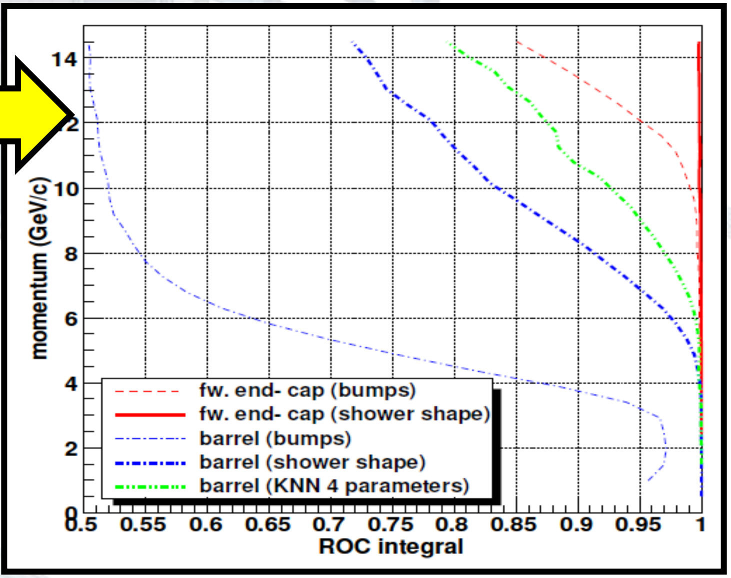
BUT

never put inside PandaRoot



γ/π° separation in EMC





Things to do: Reconstruction

- Validate momentum reconstruction for different particle kinds
- Validate momentum reconstruction for different particle hypothesis (kalman)
- Study and improve track correlation (momentum-wise?)

Things to do: Reconstruction

- Improve and make faster the correlation
- Proper set of correlation windows, as a function of momentum?
- Study and improve the anti-correlation for neutrals
- Correlation with forward PID detector is almost missing
- ...

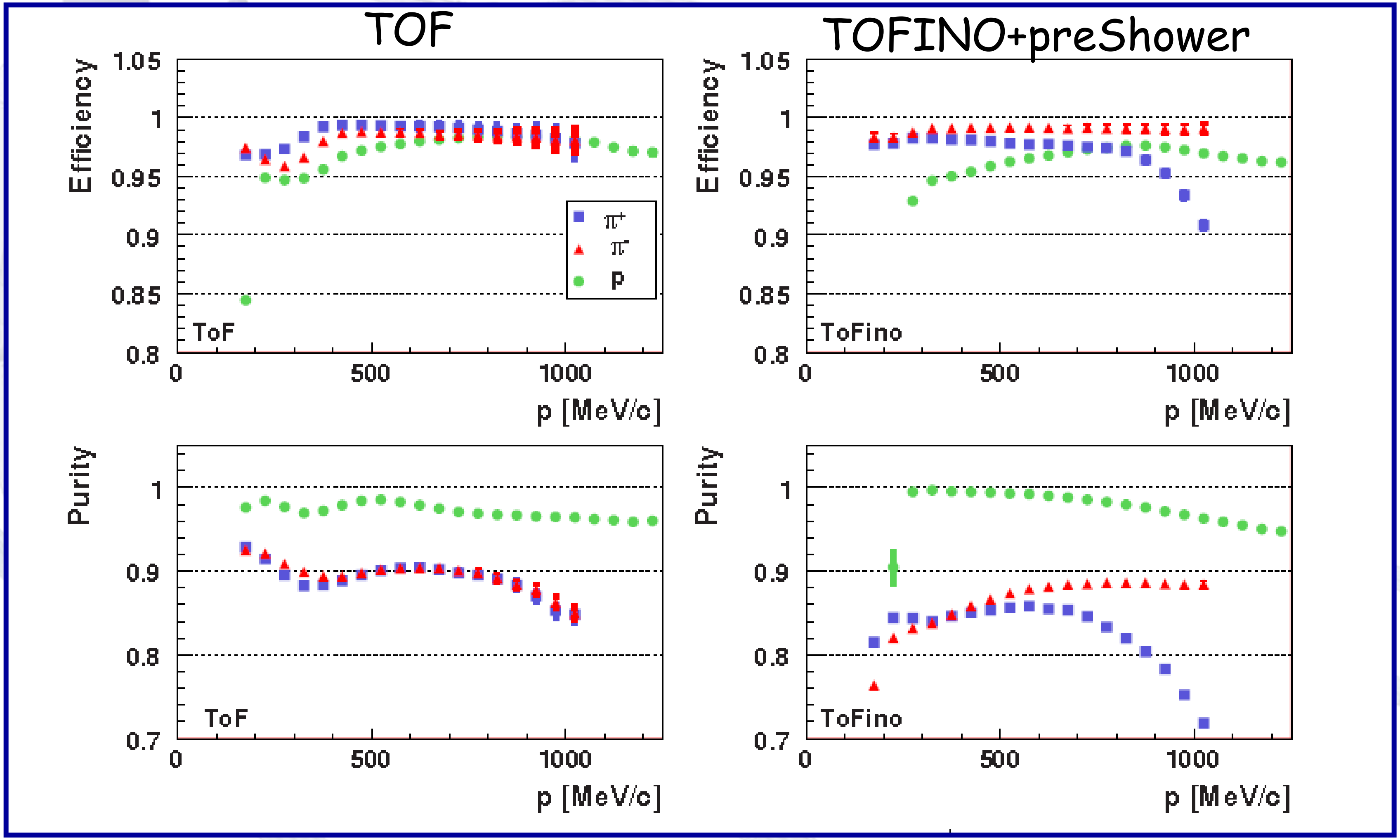
Things to do: Algorithms

- Evaluate efficiency and purity for each algorithm (standard macro)
- Evaluate perfomances by combining different algorithms (global PID)
- Global MVA analysis
- Influence of Geant3/Geant4
- …

Things to do: Algorithms

- Forward PID completely missing
- EMC MVA method for π° identification?
- MDT Muon identification with MVA
- MDT neutral particle identification?
- Use PID for analysis!!!!
- …

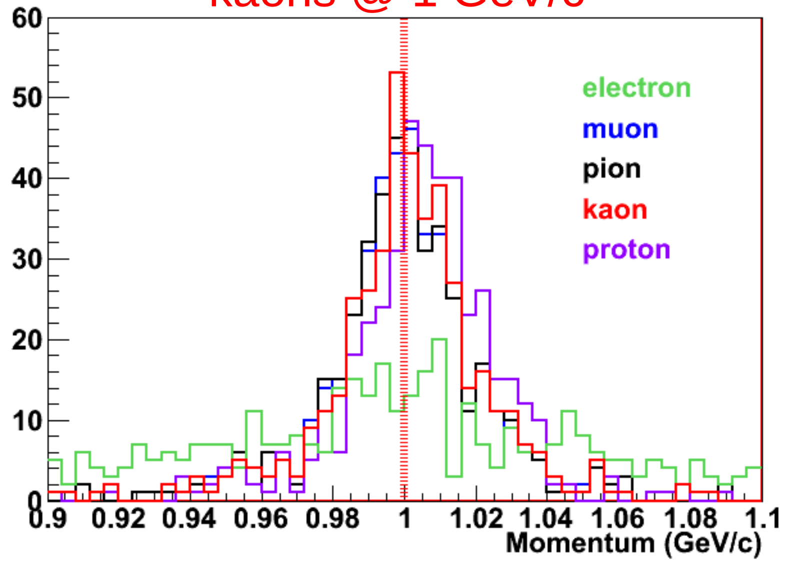
How to validate: Efficiency and Purity



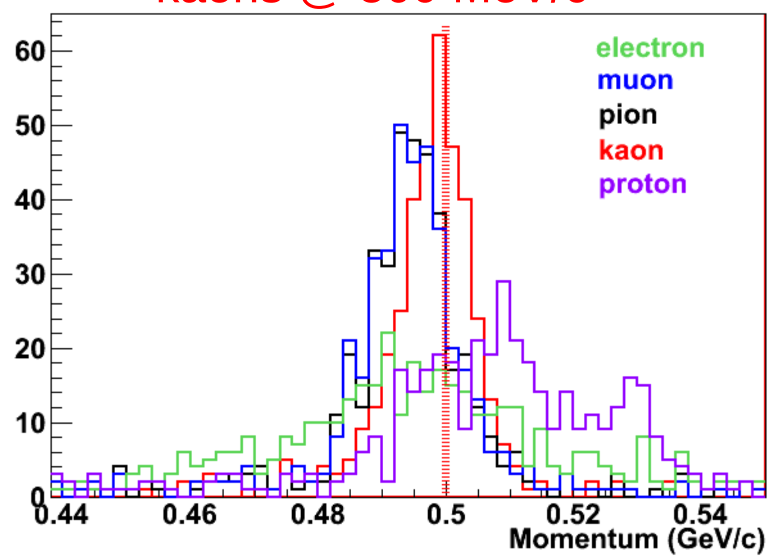
HADES: EPJA40(2009)45

Multi-Hypothesis Particle Fit

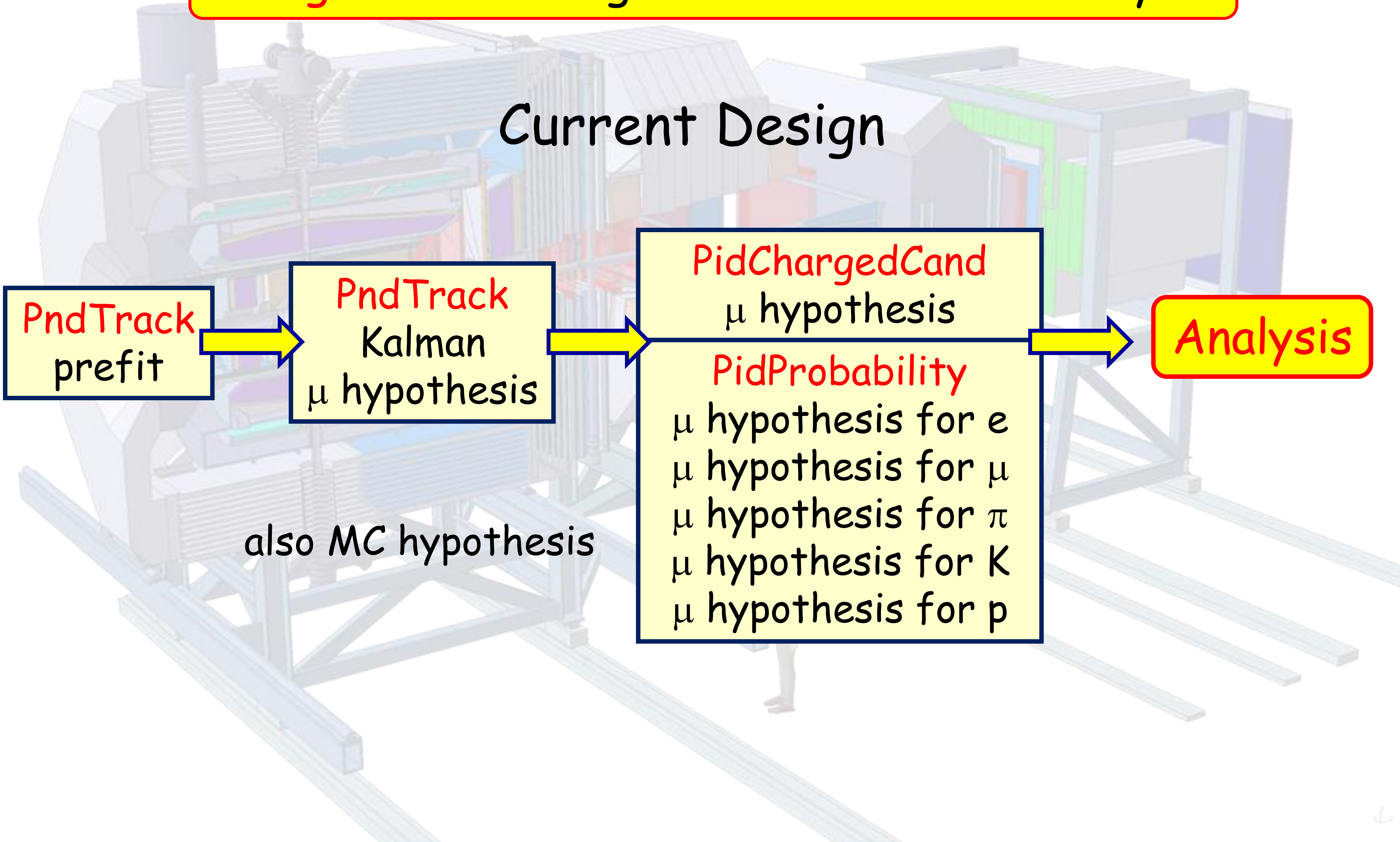
kaons @ 1 GeV/c



kaons @ 500 MeV/c



Things to do: Design and interface to analysis



Current Design

PndTrack prefit → PndTrack Kalman μ hypothesis → PidChargedCand μ hypothesis / PidProbability μ hypothesis for e, μ hypothesis for μ, μ hypothesis for π, μ hypothesis for K, μ hypothesis for p → Analysis

also MC hypothesis

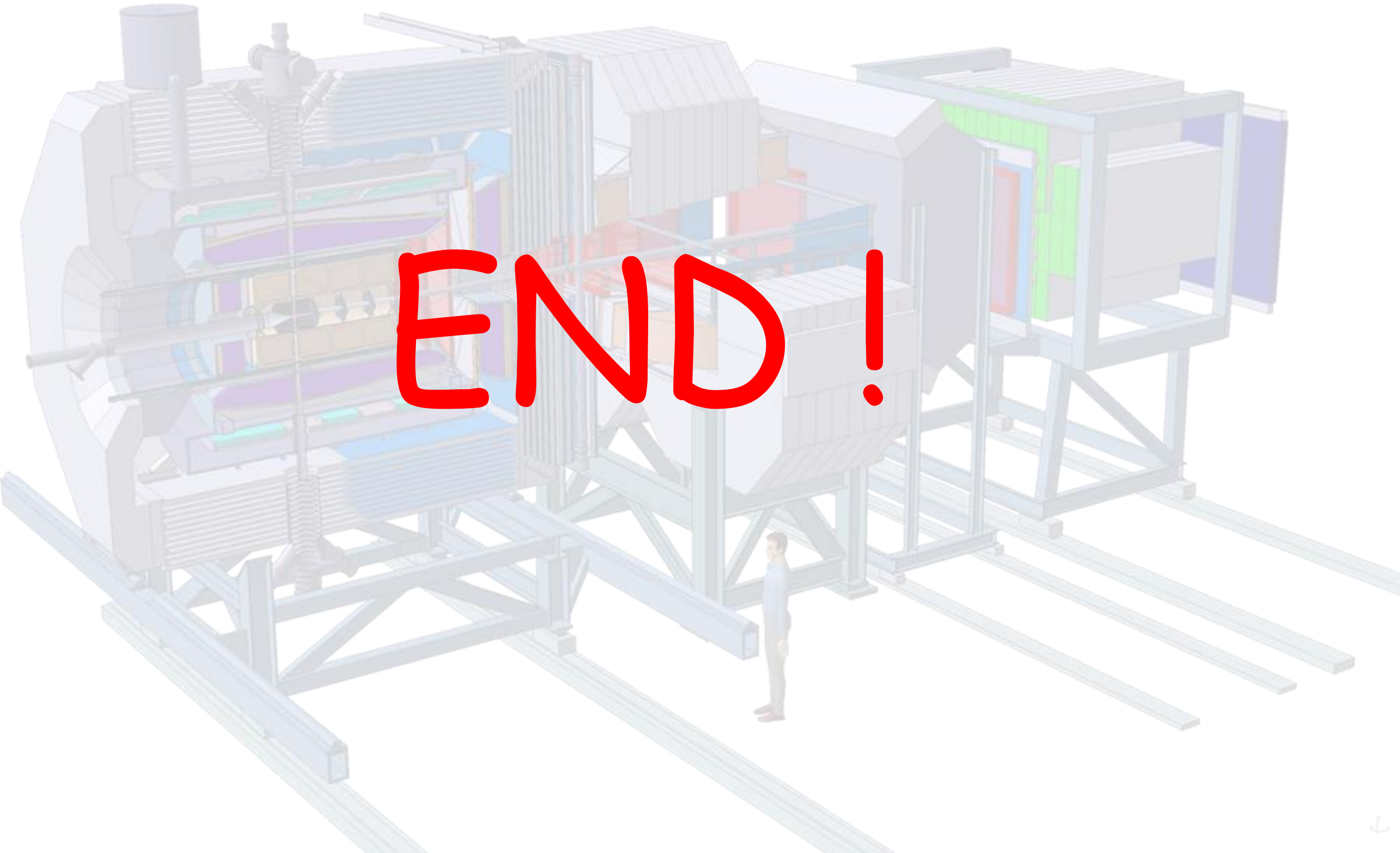
Things to do: Design and interface to analysis

New Design ?

PndTrack prefit →

PndTrack e hypothesis	→	PidChargedCand e hypothesis	PidProbability e
PndTrack μ hypothesis	→	PidChargedCand μ hypothesis	PidProbability μ
PndTrack π hypothesis	→	PidChargedCand π hypothesis	PidProbability π
PndTrack K hypothesis	→	PidChargedCand K hypothesis	PidProbability K
PndTrack p hypothesis	→	PidChargedCand p hypothesis	PidProbability p

ANALYSIS


END !