Bayesian Approach for Particle Identification in Panda(Root)

Stefano Spataro

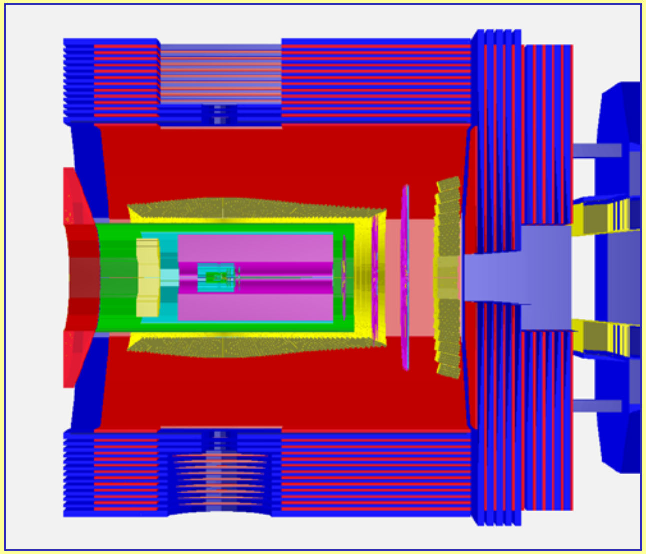
UNIVERSITÀ DEGLI STUDI DI TORINO
ALMA UNIVERSITAS TAURINENSIS

ISTITUTO NAZIONALE DI FISICA NUCLEARE
Sezione di Torino

Overview

- Requirements for Particle Identification
- Probability Density Functions
- The Bayesian Theorem
- Current Implementation in PandaRoot
- Many things to do

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

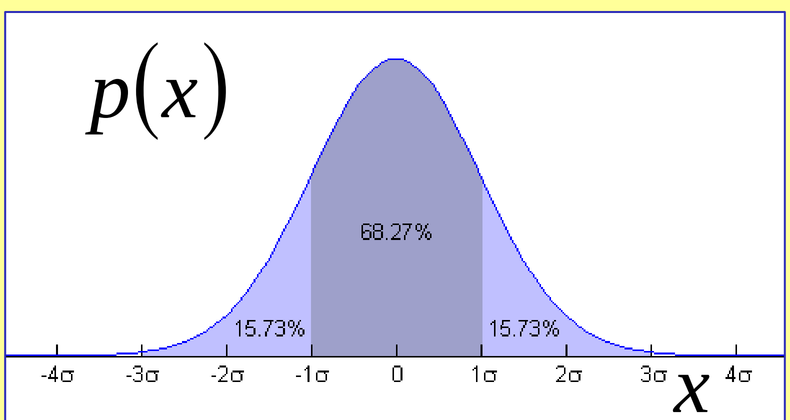
- Detector response (i.e. resolution)
- Event/track selection (analysis)

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

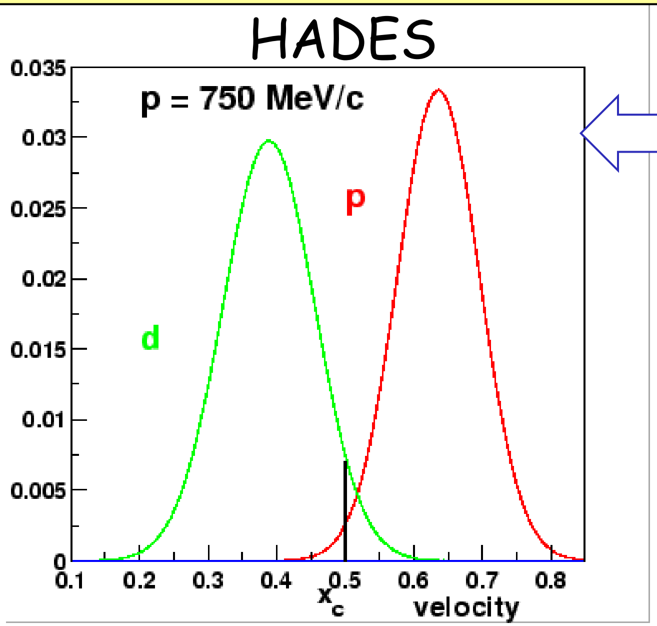


Fig.1 Probability density functions of velocities for protons and deuterons at particle momentum 750 MeV/c

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/algorythms constributing to PID

Global Likelihood

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

k = MVD dE/dx, DRC θ_C...

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

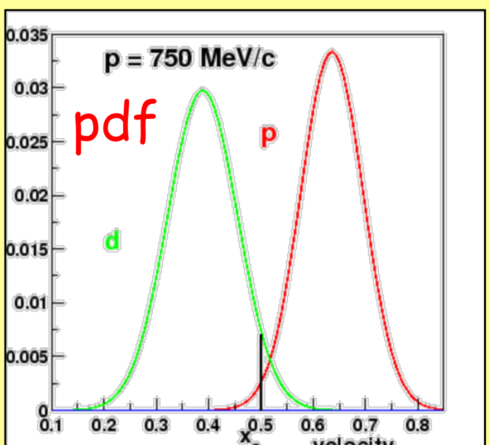
$$P(\vec{x} \mid h) = \frac{L(\vec{x} \mid h) \times P(h)}{\sum_{h=e,\mu,\pi,K,p} L(\vec{x} \mid h) \times P(h)}$$

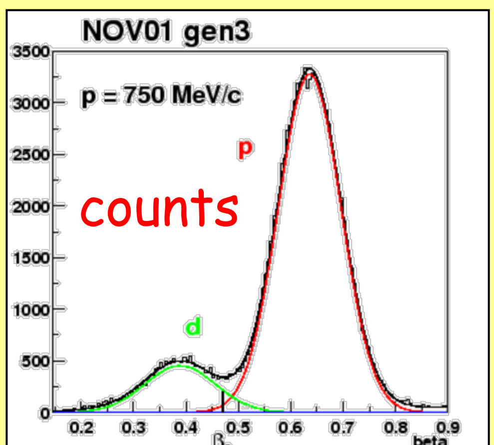
The Bayes Theorem - II

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

$$P(h)$$

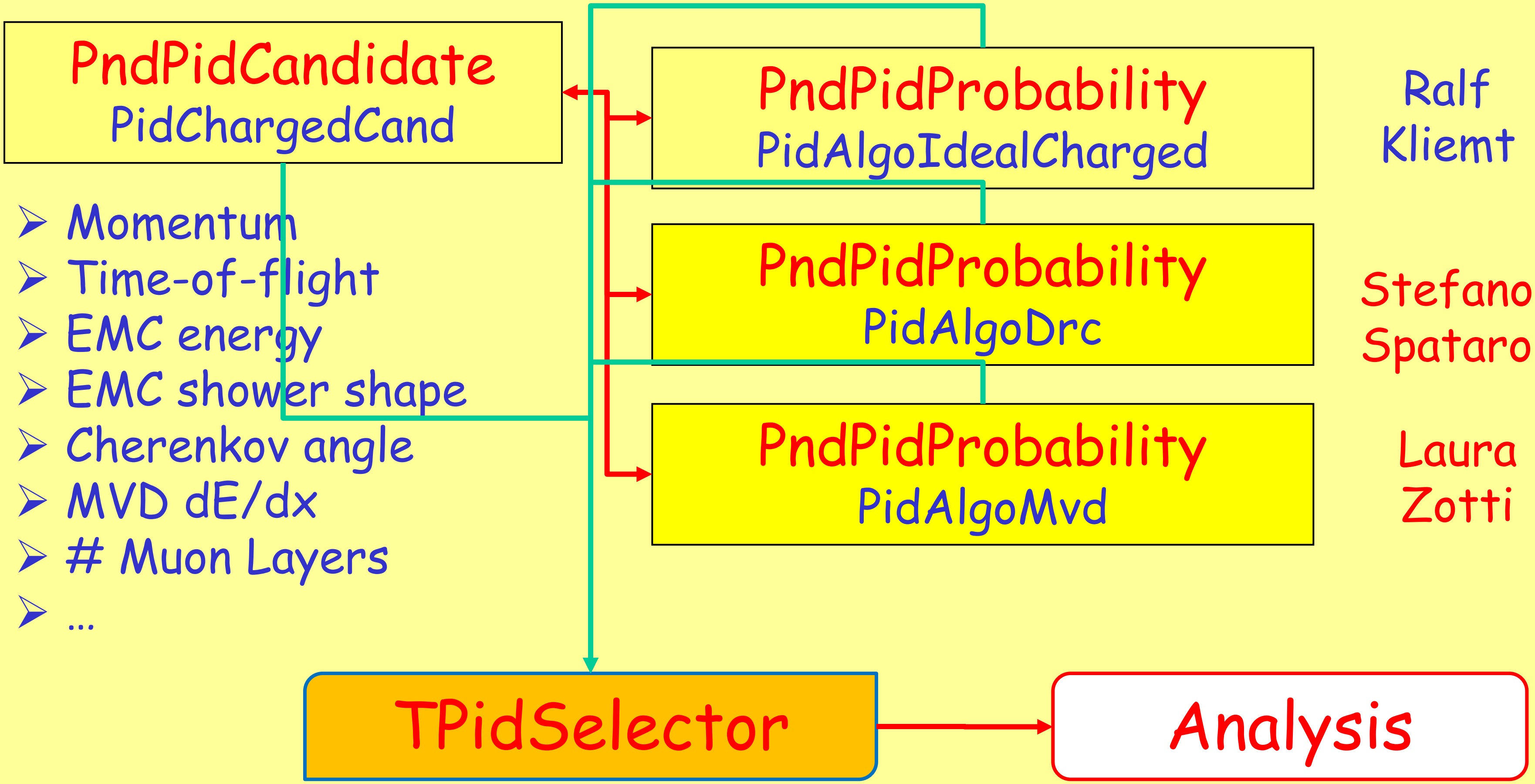
apriori probability to find the particle kind h in the detector





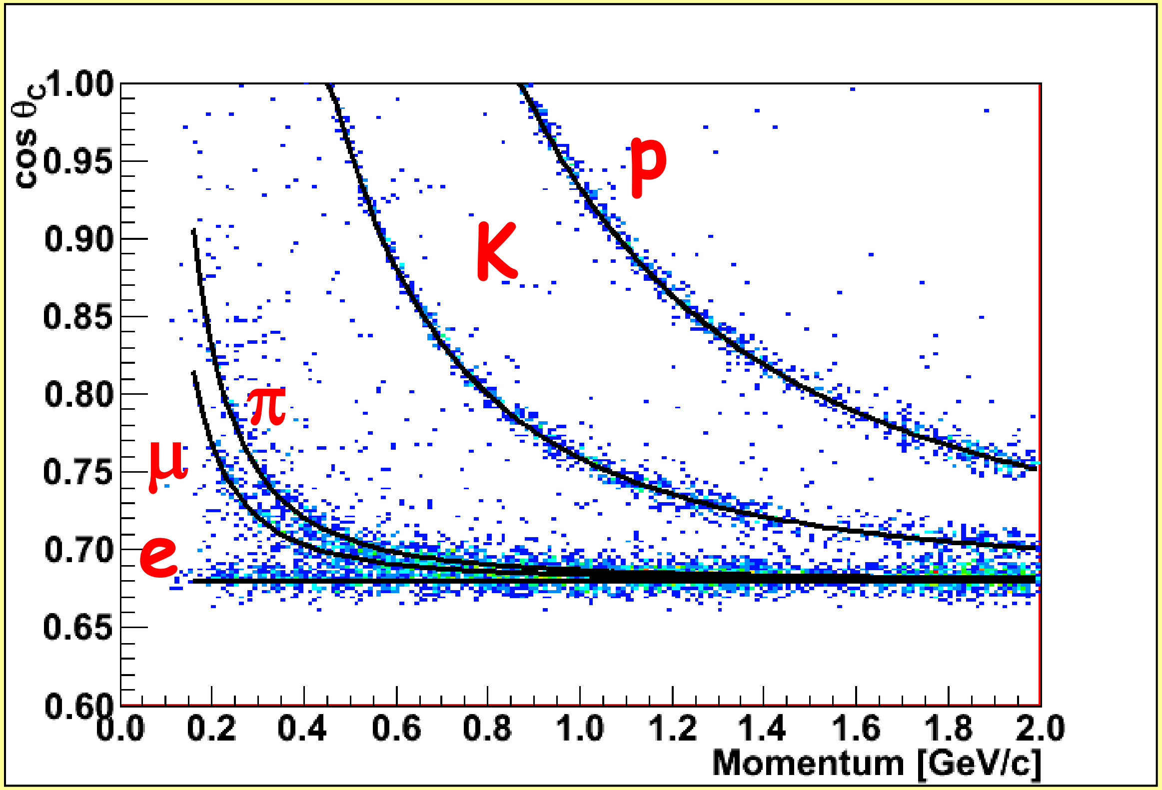
P(h) depends only on track/event selection

Current Implementation in PandaRoot



The first two PID algorythms (apart from MC)

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

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

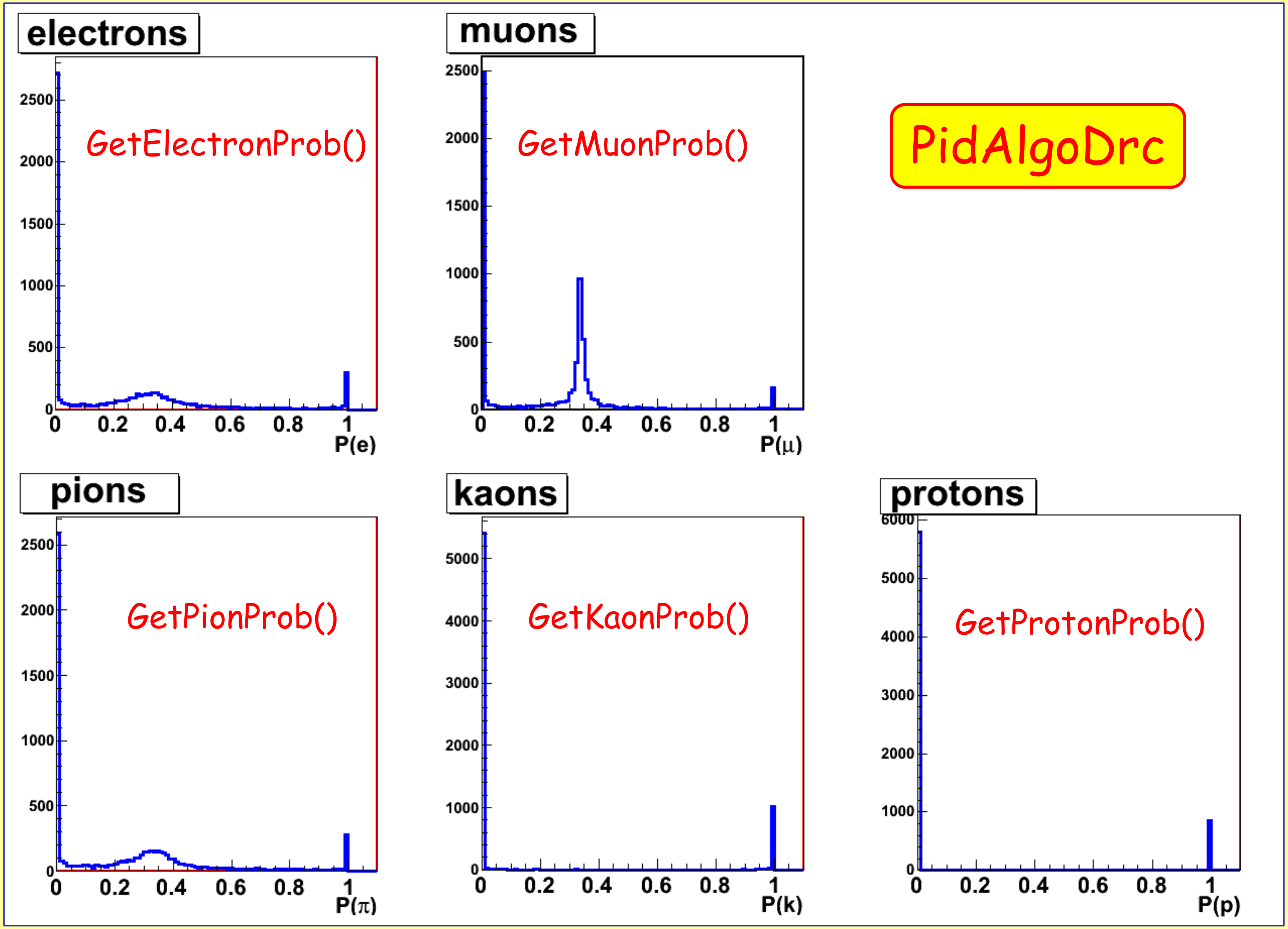
sigma

parametrization

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

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

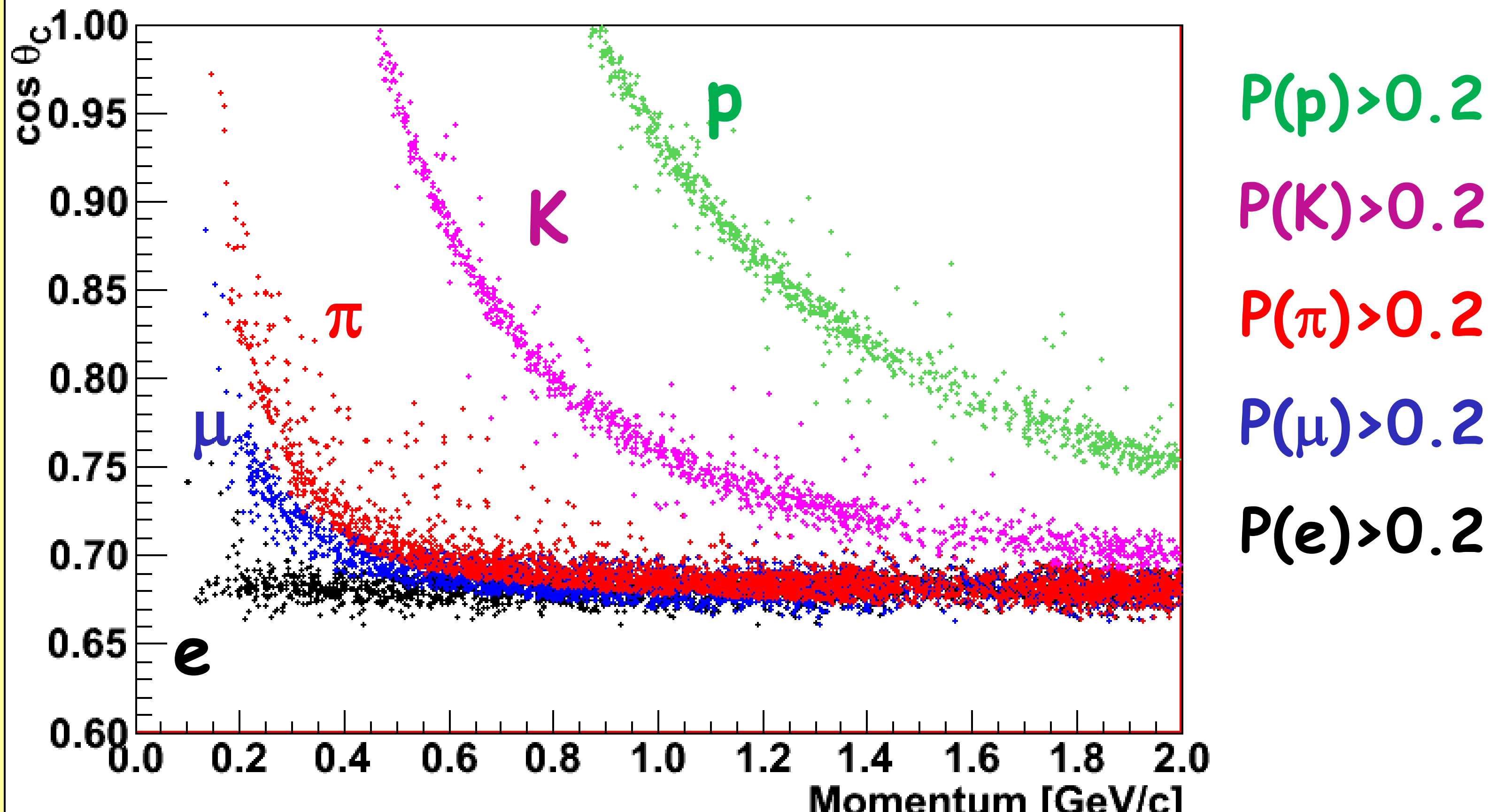
Bayesian Approach for Particle Identification in Panda(Root)



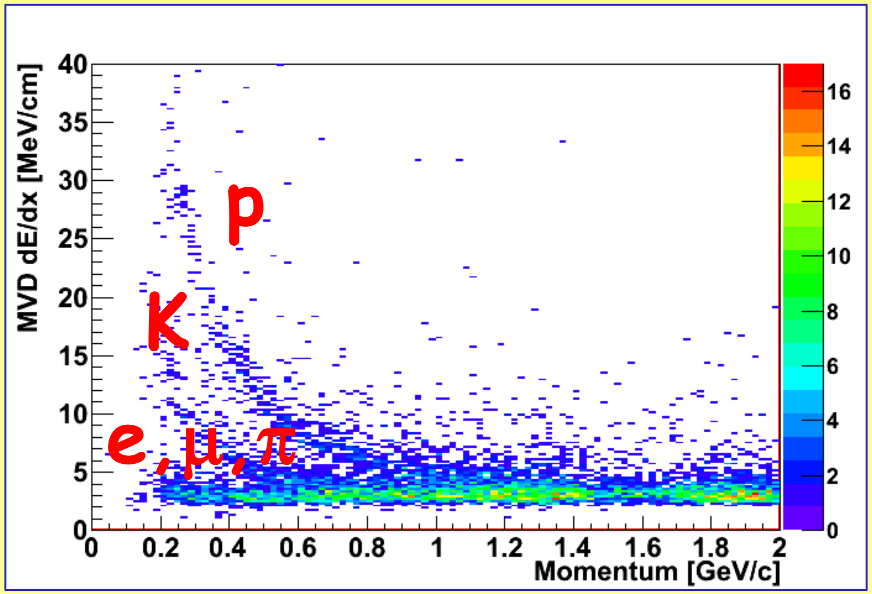
Particle Identification with DIRC

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

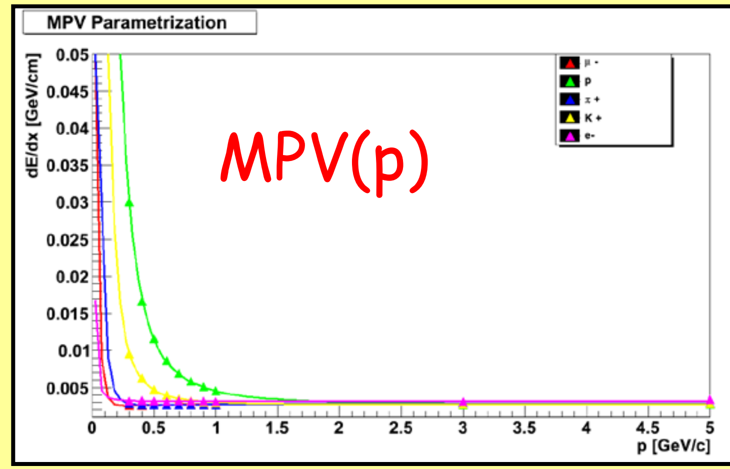
Xxx
Electron, Muon, Pion, Kaon, Proton

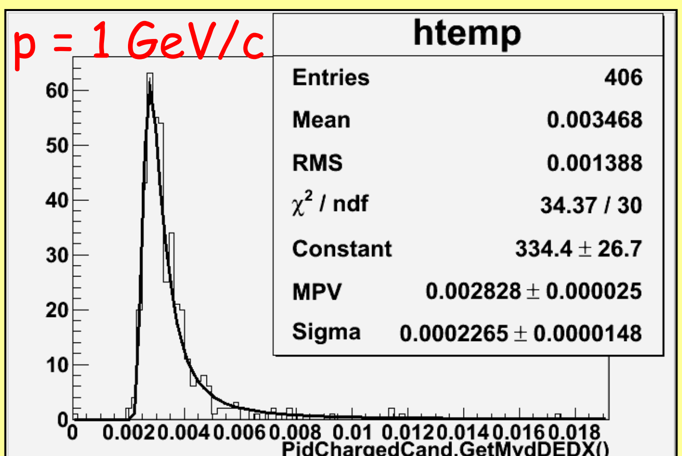


MVD: PndPidMvdAssociatorTask (Laura Zotti)

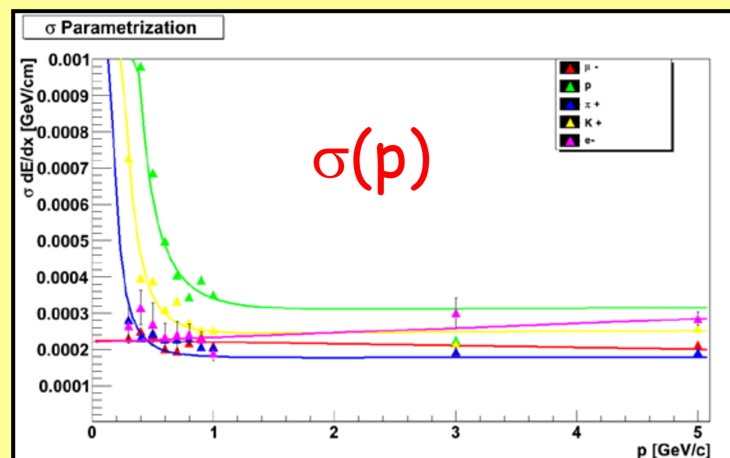


pdf: Landau

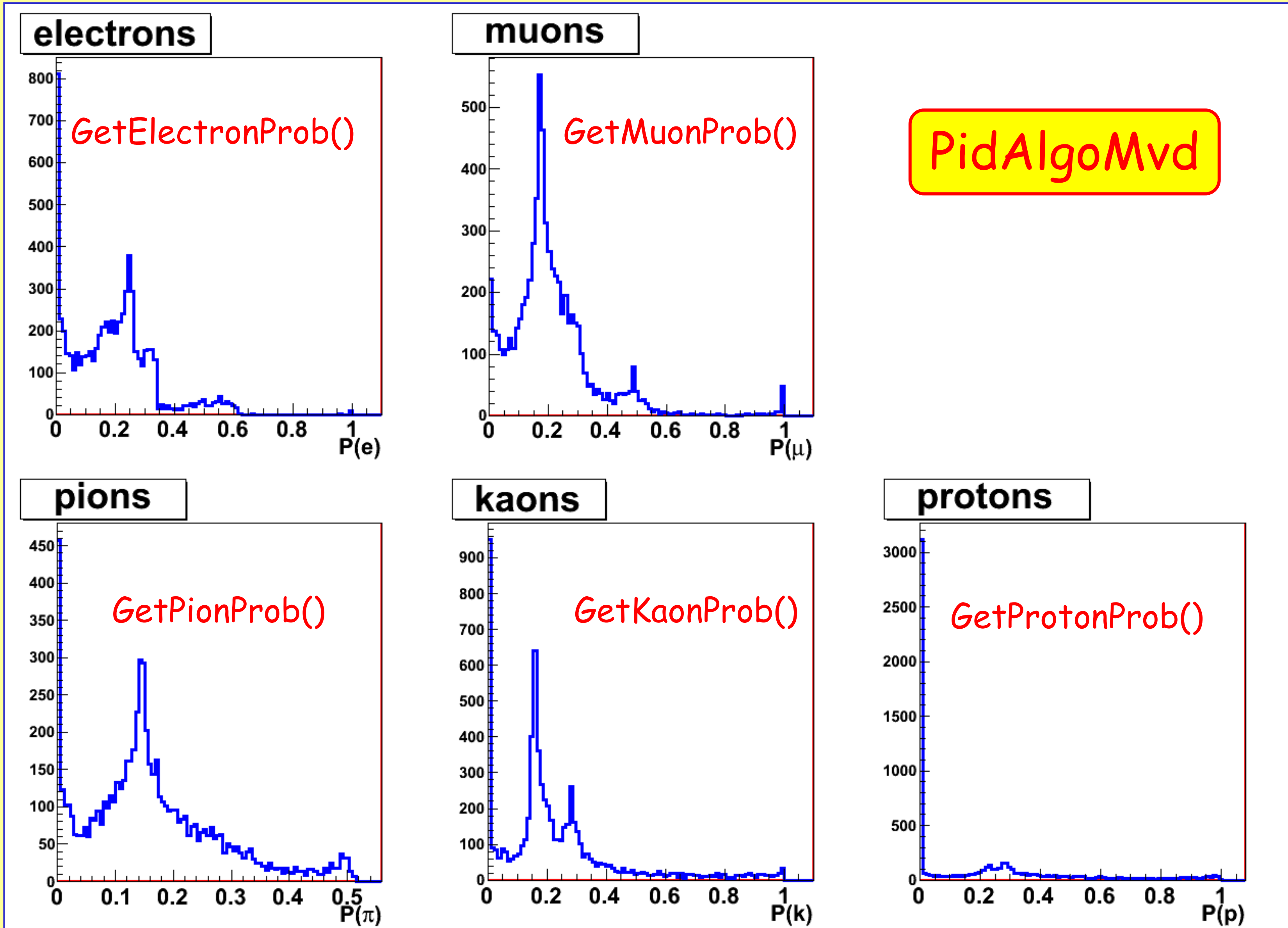




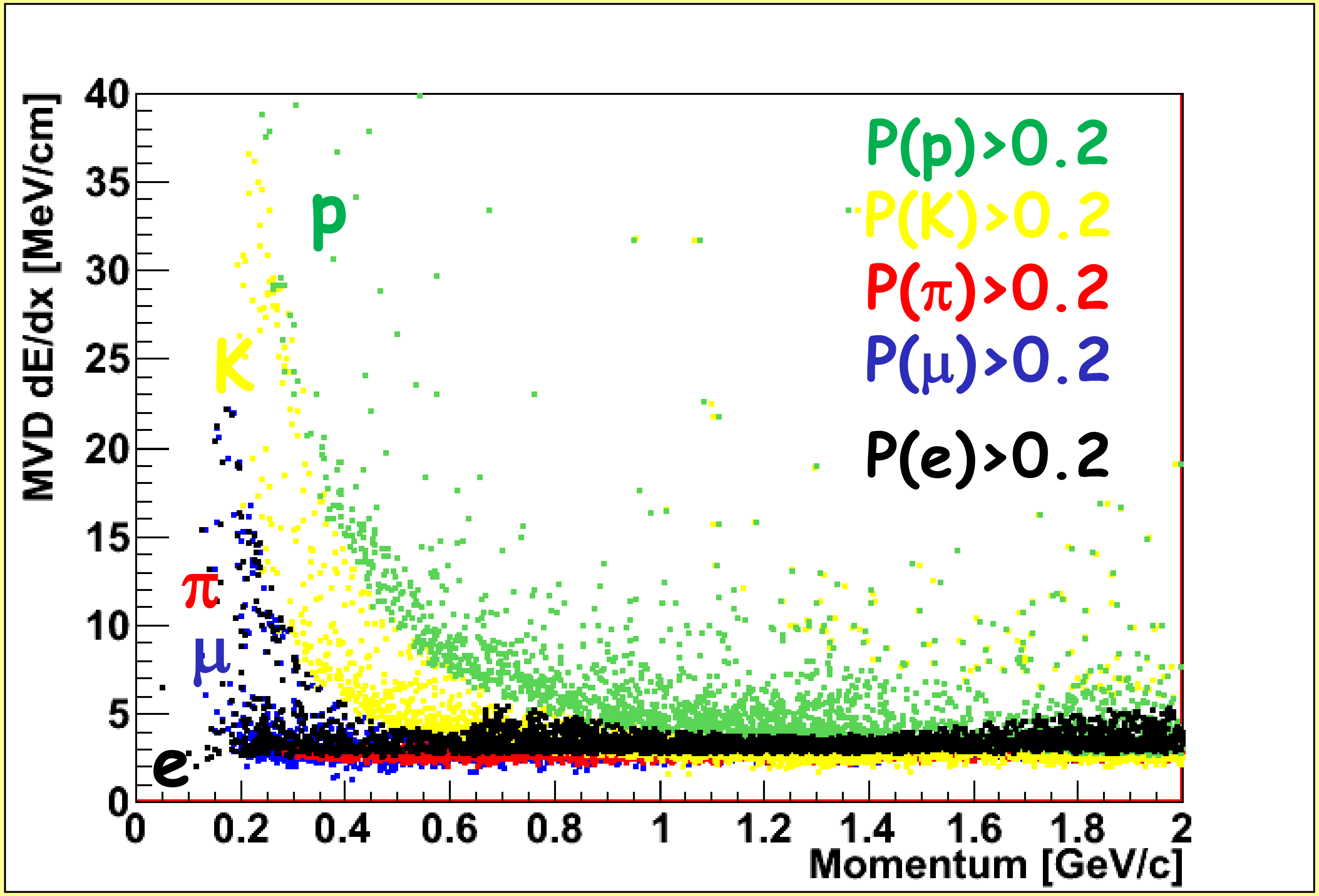
MVD dE/dx Landau



Bayesian Approach for Particle Identification in Panda(Root)



Particle Identification with MVD

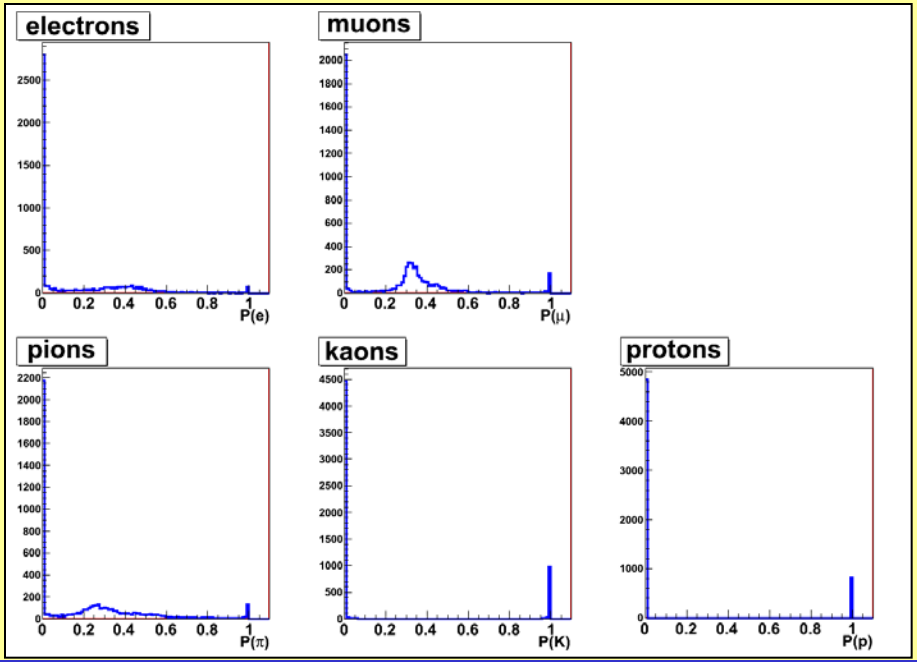


furter details in MVD session, Laura Zotti talk

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 algorythms

simple multiplication

Particle Flux

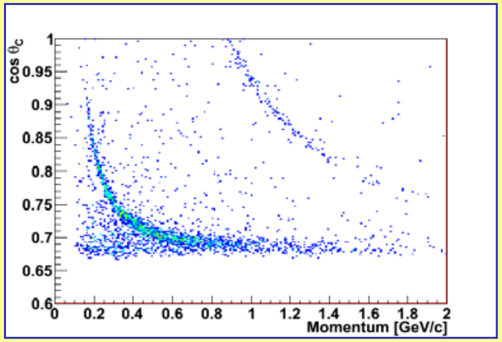
$$P(\vec{x}\,|\,h)=\frac{L(\vec{x}\,|\,h)\times P(h)}{\sum_{h=e,\mu,\pi,K,p} L(\vec{x}\,|\,h)\times P(h)}$$

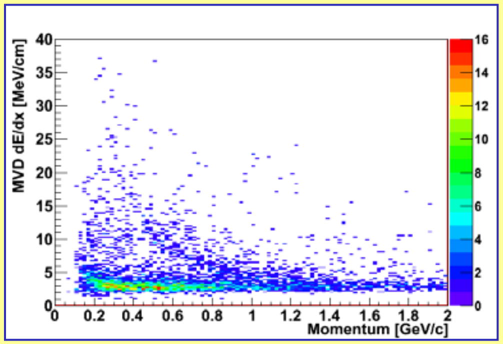
P(h) depends only on track/event selection

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)
```

Things to do

- More PID algorythms are needed (call for manpower)
  TPC, STT, MDT, EMC, …
- Automatic calculation of efficiency/purity (myself)

- Charged and Neutral candidates should be unified
- Smarter way to handle PidProbability (maps?)
- Probability for positive and negative particles?
- Flux (momentum/theta-wise?)