PID
(from a user's point of view)

Ale Gaz

Belle II Physics Week
Valencia, November 29th 2022

Outline

- **What this talk is NOT about**;
- **Introduction**:
  - what is PID (for)?
  - do you need PID for your analysis?
  - how can you use PID effectively? What should you be careful about?
- **PID at Belle II**:
  - global / binary likelihood ratios;
  - weighted likelihood(s);
  - machine learning approaches;
- **PID performance and corrections:**
  - why do you need them?
  - using the "old style" PID tables;
  - using the Systematics Framework.

What this talk in NOT about

You might be a bit disappointed by my presentation today because:

- I will not cover in detail how the Belle II sub-detectors contribute to PID (in fact this is the subject of Samo’s talk yesterday);
- I will not give you “magic numbers” that you can paste right away into the systematics table of your B2Note;
- I will not give you simple and universal recommendations, because eventually the choices one has to make (and the uncertainties one will get) depend very strongly on your analysis;

My goal today is to give you some guidance on how to use existing tools, so that you can take advantage of them to improve the sensitivity of your analysis and you are able to estimate the corrections and uncertainties by yourselves.

Introduction

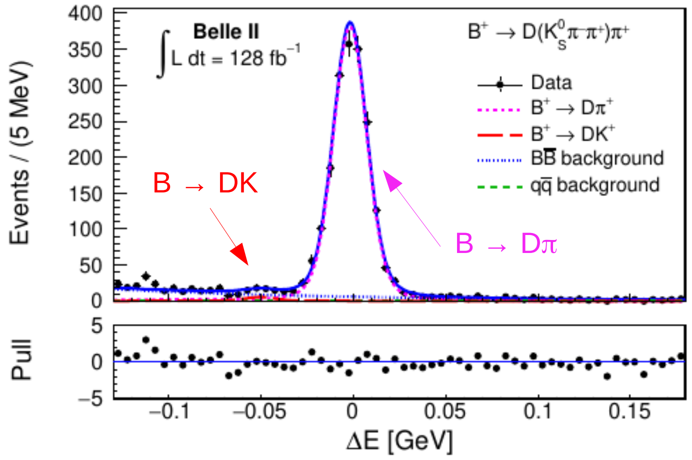
What is PID?

- At Belle II we produce 6 types of **stable** charged particles: electrons, muons, pions, kaons, protons, and deuterons (which I will mostly disregard in this talk);

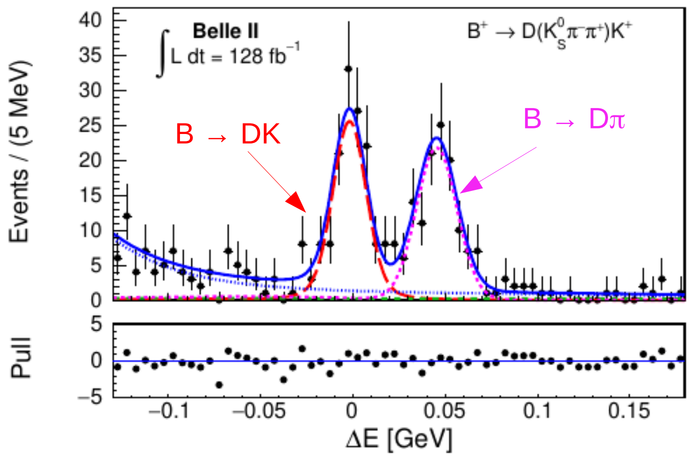
- The task of PID is to distinguish among these different kinds of particles:

- In practice, the identification will never be *perfect*
  - the **efficiency** (probability that the particle I want to select is actually selected) will be < 100%;
  - the **mis-identification rate** (probability that a particle that I do NOT want to select is actually selected) will be > 0%.

π enhanced



K enhanced



JHEP **2022**, 63 (2022)

Is PID relevant for Belle II?

- **Yes, it is!** Flavor physics is (almost) all about distinguishing among final states that are accessible to the same mother particle;

- Sometimes PID is the only handle that allows you to separate between very similar final states, e.g.:
  - ➔ $B^0 \rightarrow K^{*0}\gamma \,/\, \rho^0\gamma$;
  - ➔ $D^0 \rightarrow \pi^+\pi^- \,/\, \mu^+\mu^-$;
  - ➔ $\tau^+ \rightarrow e^+ \nu\ \ \nu,\ \mu^+\ \nu\ \ \nu,\ \pi^+\ \ \nu$;

- Very precise LFUV measurements require a very good control over the efficiencies and the background contaminations (e's and μ's are easy to distinguish, but π's can fake both!);

- PID plays a very important role in the B and Charm Flavor Taggers. A drop or improvement in the PID performance will have a sizable impact on the tagging performance.

Is PID relevant for you?

- **Not necessarily!** It ultimately depends on your analysis;

- There are quite a few cases in which you don't want to use PID:
    1) Your S/N ratio is already very high: applying PID will lower your efficiency without any significant benefit from background reduction.
       Example: we can select clean samples of $K_S \rightarrow \pi^+\pi^-$ by cutting on the flight length significance, applying PID won't help much against a background that comes mostly from random combinations of π's;
    2) Your main backgrounds have the same final state particles as your signal.
       Example: PID won't help you distinguishing $\phi \rightarrow K^+K^-$ from $f_0 \rightarrow K^+K^-$;
    3) Your backgrounds are easy to deal with (because of their shape and/or they can be studied using data control samples).
       Example: continuum background in many $B \rightarrow$ hadronic analyses.

- You should also be aware that PID comes with systematic uncertainties: the benefits from using it should at the very least outweigh its drawbacks.

What is the recommended PID cut?

- Again, **it depends on your analysis**, we cannot give a one-fits-all recommendation;
- Typically one optimizes the selection using the figure of merit:

$$\frac{S}{\sqrt{S+B}}$$

- But this is not the only option:
  - ➔ you might have a background source that will bring in a large systematic uncertainty (because it is not well known/simulated, …) in this case you want to cut harder and increase your S/N ratio;
  - ➔ your background is harmless, so you might live with relatively low S/N ratio and enjoy a larger signal efficiency.

What are the potential pitfalls?

- Analyses are designed on a number of (explicit or implicit) assumptions;
- Our Monte Carlo is built on many assumptions (about particle decays, hadronization models, particle/detector interactions, performance, …);

- Always check your assumptions!

- Have a look at the data (not the signal region!) sooner rather than later:
  - ➔ can you find a control sample reasonably similar to your signal?
  - ➔ are your sidebands reasonably well populated?

- You might discover that some backgrounds are not well simulated, or even not included at all in our Monte Carlo. So you would need to go back and re-optimize your (PID) selection.

PID at Belle II

Analysis level inputs

- All BelleII sub-detectors (with the exception of the PXD) contribute to PID;
- Each of them provides the “likelihood”:

$$\mathcal{L}_\alpha^d$$

where the d index runs over the subdetectors, and α over the particle types;

Example:

the quantity $\mathcal{L}_\mu^{KLM}$ is related to the probability that the particle under study is a muon, based on the KLM information

- In `basf2`, you can access the (positive) log likelihood given by a specific subdetector for a particular particle type by asking something like:

$$\log(\mathcal{L}_\mu^{KLM}) = LL_\mu^{KLM} =$$

$$= \texttt{pidLogLikelihoodValueExpert(13, KLM)}$$

mcPDG (Lund code) of the particle I am interested in

Example

- Let’s generate a ParticleGun particle and see what the BelleII subdetectors think about it:
- The tracking tells us:

  p = 2.00354 Gev/c   cosθ = -0.47389   phi = -0.57255

- The PID log likelihoods, for each subdetector and each hypotesis are:

	e	μ	π	K	p	d
SVD	-2.54876	-1.15602	-1.15602	-1.21668	-1.48521	-4.60517
CDC	-2.85408	-0.08678	-0.00840	-0.23215	-0.02462	-8.27275
TOP	-538.954	-538.949	-538.950	-538.103	-536.000	-534.300
ARICH	-	-	-	-	-	-
ECL	-4.89060	-4.14807	0.10629	0.06672	0.42025	0.16023
KLM	-6.40549	-1.64323	-4.19844	-1.89742	-4.49254	-4.88627

So, what is this particle?

Example

	e	μ	π	K	p	d
SVD	-2.54876	**-1.15602**	**-1.15602**	-1.21668	-1.48521	-4.60517
CDC	-2.85408	-0.08678	**-0.00840**	-0.23215	-0.02462	-8.27275
TOP	-538.954	-538.949	-538.950	**-538.103**	-536.000	-534.300
ARICH	-	-	-	-	-	-
ECL	-4.89060	-4.14807	0.10629	0.06672	**0.42025**	0.16023
KLM	-6.40549	**-1.64323**	-4.19844	-1.89742	-4.49254	-4.88627

- I highlighted the hypotheses for which each sub-detector gives the highest log-likelihood;
- The SVD is undecided between μ and π, the CDC prefers the π, the TOP thinks it's a K, for the ECL it's a p, and for the KLM it's a μ;

(I admit I chose this track because it looks somewhat problematic)

How do we summarize this information and get a more definitive answer?

Example

- To extract the full information about a particle, we can multiply the likelihoods:

$$\mathcal{L}_\alpha = \mathcal{L}_\alpha^{SVD} \cdot \mathcal{L}_\alpha^{CDC} \cdot \mathcal{L}_\alpha^{TOP} \cdot \mathcal{L}_\alpha^{ARICH} \cdot \mathcal{L}_\alpha^{ECL} \cdot \mathcal{L}_\alpha^{KLM}$$

  or equivalently sum the log-likelihoods:

$$\log(\mathcal{L}_\alpha) = LL_\alpha = LL_\alpha^{SVD} + LL_\alpha^{CDC} + LL_\alpha^{TOP} + LL_\alpha^{ARICH} + LL_\alpha^{ECL} + LL_\alpha^{KLM}$$

	e	μ	π	K	p	d
SVD	-2.54876	-1.15602	-1.15602	-1.21668	-1.48521	-4.60517
CDC	-2.85408	-0.08678	-0.00840	-0.23215	-0.02462	-8.27275
TOP	-538.954	-538.949	-538.950	-538.103	-536.000	-534.300
ARICH	-	-	-	-	-	-
ECL	-4.89060	-4.14807	0.10629	0.06672	0.42025	0.16023
KLM	-6.40549	-1.64323	-4.19844	-1.89742	-4.49254	-4.88627
Sum	-555.653	-545.983	-544.207	**-541.383**	-541.582	-551.904

The K hypothesis gets the highest (i.e. less negative) score.
The p hypothesis follows closely

Likelihood Ratios – binary PID

- How do we compare e.g. the K hypothesis vs the p?
- A standard and popular approach is to take the lihelihood ratio:

$$P(K\ vs\ p) = \frac{\mathcal{L}_K}{\mathcal{L}_K + \mathcal{L}_p}$$

which is equivalent to:

$$P(K\ vs\ p) = \frac{1}{1 + e^{[\log(\mathcal{L}_p) - \log(\mathcal{L}_K)]}} = \frac{1}{1 + e^{\Delta LL_{pK}}}$$

- The quantity P(K vs p) is bound to be in [0, 1], so it can be interpreted as a probability;

- Going back to the example we have been discussing:

LL_K = -541.383
LL_p = -541.582
ΔLL_{pK} = -0.199

which gives:

P(K vs p) = 0.5496
P(p vs K) = 0.4504

so, yes, the K hypothesis is slightly favored, but not by much

Likelihood Ratios – binary PID

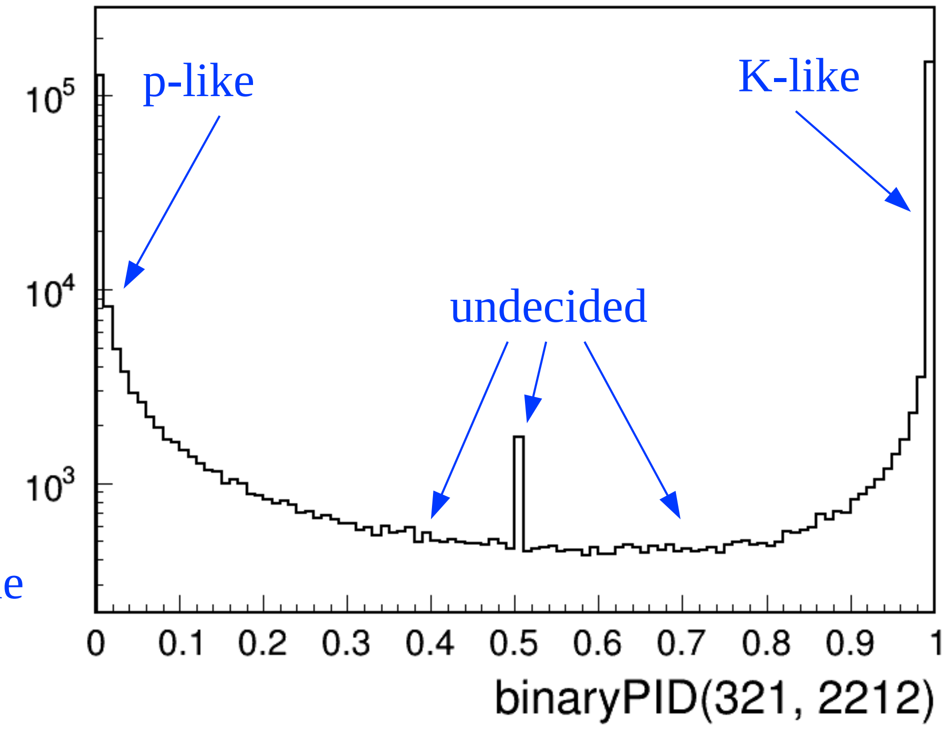
- Obviously you are not expected to compute all this by hand, `basf2` will immediately provide the ***binary*** likelihood ratio we have just defined:

$$P(K\ vs\ p) = \texttt{binaryPID(321, 2212)}$$

mcPDG of the particle whose L goes in the numerator

mcPDG of the particle whose L goes in the denominator

- If we plot P(K vs p), we obtain something like:



(this is a ~random sample of tracks, that contains also e's, μ's and π's)

Likelihood Ratios – global PID

- Binary PID is great if you have a specific source of background that you want to suppress;
- In more generic cases, you want to compare a particle hypothesis (e.g. kaon) against all others, thus switching to the ***global*** likelihood ratio:

$$P(K) = \frac{\mathcal{L}_K}{\mathcal{L}_K + (\mathcal{L}_e + \mathcal{L}_\mu + \mathcal{L}_\pi + \mathcal{L}_p + \mathcal{L}_d)}$$

which is equivalent to:

$$P(K) = \frac{1}{1 + e^{\Delta LL_{eK}} + e^{\Delta LL_{\mu K}} + e^{\Delta LL_{\pi K}} + e^{\Delta LL_{pK}} + e^{\Delta LL_{dK}}}$$

- Crunching the numbers of the example track, we get:

$P(e) = 3.36 \times 10^{-7}$
$P(\mu) = 0.00531$
$P(\pi) = 0.03142$
$P(K) = 0.52965$
$P(p) = 0.43361$
$P(d) = 1.43 \times 10^{-5}$

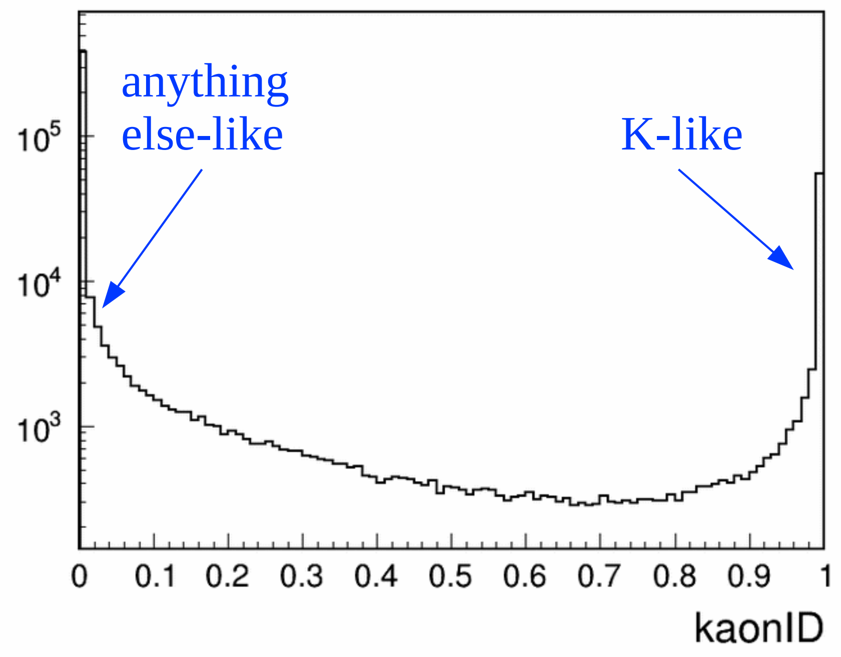
Likelihood ratios – global PID

- The global likelihood ratios I just defined are the **default PID variables**, that you can access from `basf2`.

  In other words, by default all sub-detectors are considered, and a particle hypothesis is checked against all others;

```
electronID
muonID
pionID
kaonID
protonID
deuteronID
```

- The distribution of e.g. kaonID is like:



(this is the same sample of tracks I used for the binary K vs p example)

Likelihood ratios – comments

Q: Does P(α) = 0.8 mean that, in a small interval centered around P(α) = 0.8, 80% of the particles are α, and the remaining 20% is accounted for by all the other species?

Likelihood ratios – comments

Q: Does P(α) = 0.8 mean that, in a small interval centered around P(α) = 0.8, 80% of the particles are α, and the remaining 20% is accounted for by all the other species?

A: No (or at least not necessarily). One key point to remember is that the various particles are not produced with the same abundance (the most abundant ones are the pions). The PID framework does not know anything about it.

If you want to figure out what to expect after your PID selection you will have to use the **efficiency** and **fake rates** that we will define later on.

Likelihood ratios – comments

Q: Given that they include all the subdetectors, is it guaranteed that the global PID variables provide the best PID information?

Likelihood ratios – comments

Q: Given that they include all the subdetectors, is it guaranteed that the global PID variables provide the best PID information?

A: I am afraid the answer is no, for several reasons:

1) the combination of the likelihoods assumes that the inputs are **uncorrelated** (typically a good assumption, but …);
2) sub-detectors can have "**blind spots**": the global PID ignores the fact that there are regions of the phase space for which a sub-detector are not very reliable (think of the points where the dE/dx bands cross);
3) $\Delta LL\alpha = x$ does not correspond to the same separation power for all sub-detectors. I will come back to this very soon;
4) background conditions affect sub-detectors in different ways;
5) ...

Going beyond likelihood ratios

- In principle, with the little knowledge on PID inputs I already covered, you could build your own likelihood ratio, excluding some sub-detectors and concentrating only on some particle hypotheses, so that it is most useful to your analysis;

- This would not be practical, though, so we (the PID group) are trying to provide some standard tools that you can use out of the box. I will briefly mention:
  1) Re-weighted likelihood;
  2) Multivariate approaches, including also other variables

     (well established in leptonID, coming soon for hadronID)

Re-weighted likelihood ratios

- This has been explored by the PNNL group in BELLE2-NOTE-TE-2021-027;
- In the default approach, we combine information from 6 sub-detectors about 6 particle hypothesis in:

$$P(\alpha) = \frac{\exp\left(\sum_d \log \mathcal{L}_\alpha^d\right)}{\sum_p \exp\left(\sum_d \log \mathcal{L}_p^d\right)}$$

Where d runs over the sub-detectors and p over the particle hypotheses.

This is just the same definition of the global likelihood ratio I gave above.

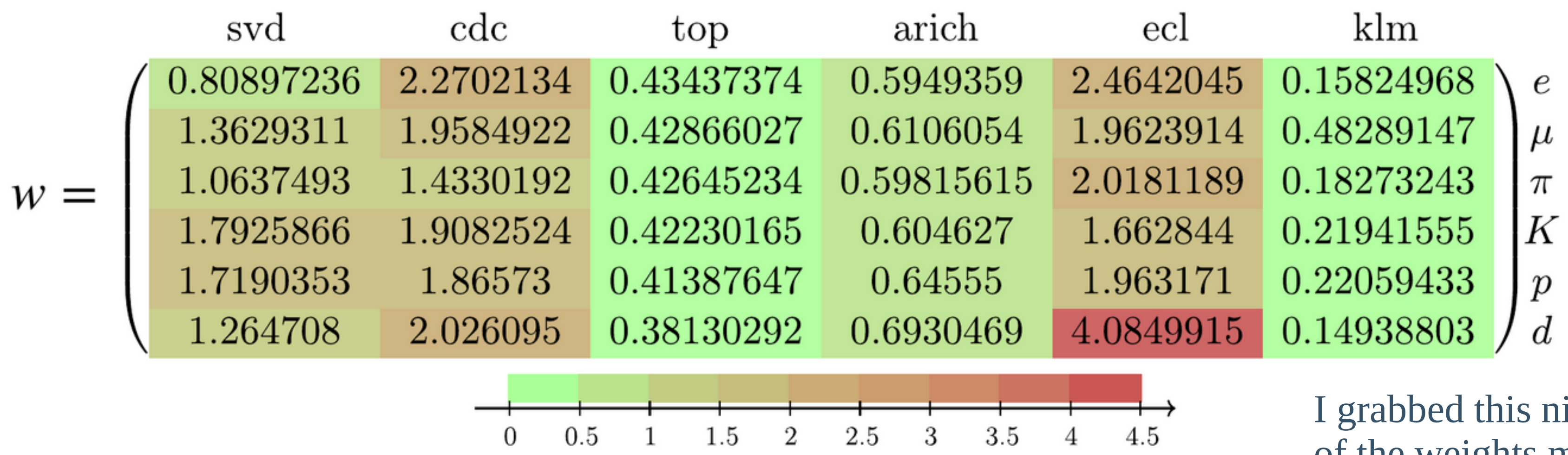
- Idea: multiply each of the 36 inputs by a $w_{p,d}$ weight and optimize the weights so that the separation power is optimal:

$$P(\alpha) = \frac{\exp\left(\sum_d w_{\alpha,d} \log \mathcal{L}_\alpha^d\right)}{\sum_p \exp\left(\sum_d w_{p,d} \log \mathcal{L}_p^d\right)}$$

If you set all $w_{p,d}$ weights to 1, you jump back to the default case

- Let me skip all the details of the optimization (that I encourage you to look by yourselves) and jump to the results!

Re-weighted likelihood



$$w = \begin{array}{ccccccc} \text{svd} & \text{cdc} & \text{top} & \text{arich} & \text{ecl} & \text{klm} & \\ 0.80897236 & 2.2702134 & 0.43437374 & 0.5949359 & 2.4642045 & 0.15824968 & e \\ 1.3629311 & 1.9584922 & 0.42866027 & 0.6106054 & 1.9623914 & 0.48289147 & \mu \\ 1.0637493 & 1.4330192 & 0.42645234 & 0.59815615 & 2.0181189 & 0.18273243 & \pi \\ 1.7925866 & 1.9082524 & 0.42230165 & 0.604627 & 1.662844 & 0.21941555 & K \\ 1.7190353 & 1.86573 & 0.41387647 & 0.64555 & 1.963171 & 0.22059433 & p \\ 1.264708 & 2.026095 & 0.38130292 & 0.6930469 & 4.0849915 & 0.14938803 & d \end{array}$$

I grabbed this nice representation of the weights matrix from one of Geraldine's talks

The above weights matrix might be puzzling to some of you (for good reasons), but:

- it significantly improves the performance, in vast areas of the phase space;
- it is easy to reproduce (targeting e.g. the momentum region that is relevant for your analysis). Yo Sato-san implemented the software infrastructure in `basf2` and provided some examples, see e.g. this talk.

Multivariate approaches

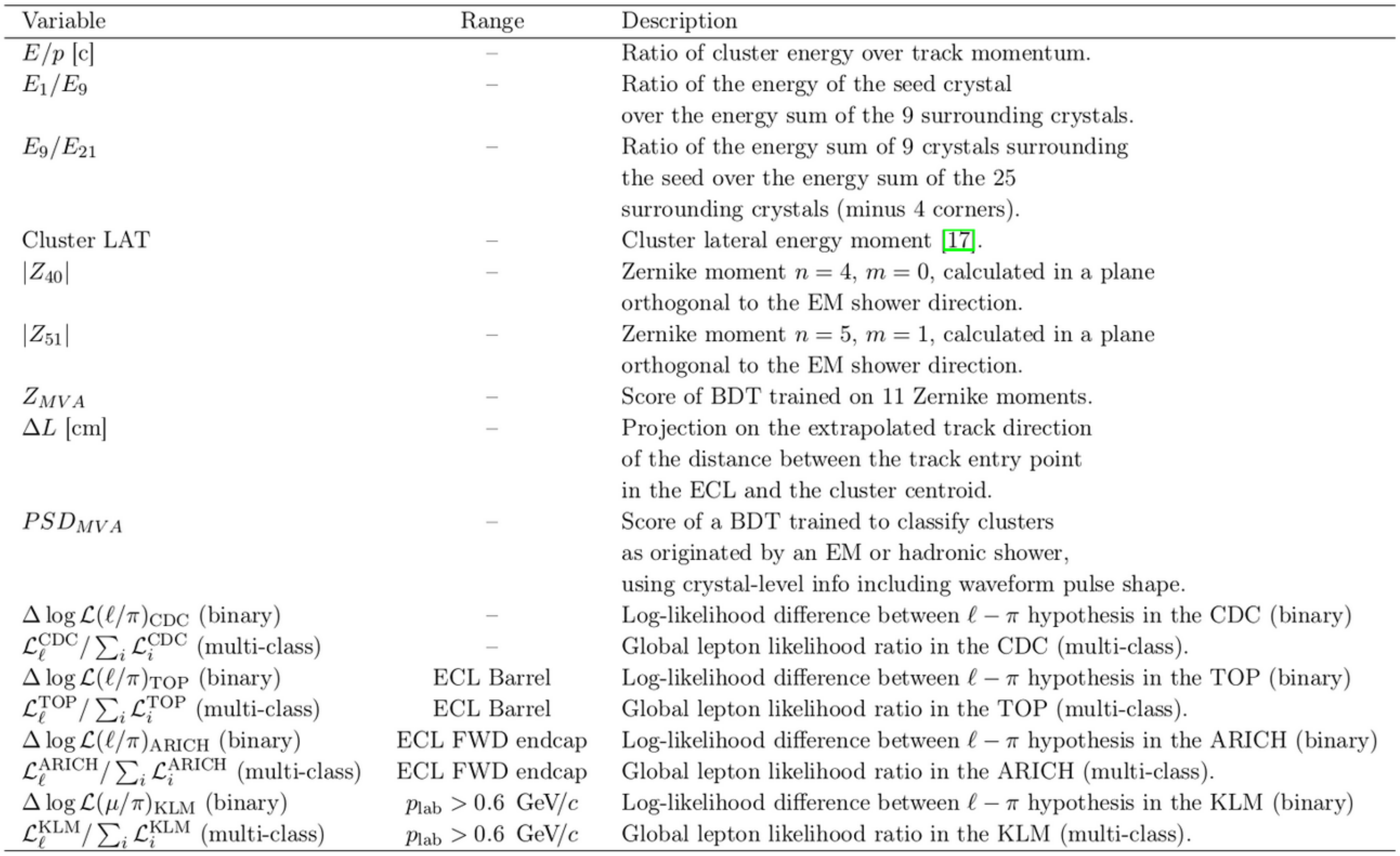
- One other approach is to use multivariate approaches based on machine learning, like Neural Networks or BDT's;

- Advantages:
  - ➔ you can use a large number of variables and include anything that provides some discimination (no matter how small);
  - ➔ these tools are typically good at handling correlations and finding the optimal choice in each corner of the phase space;
  - ➔ if the variable "time" (or a proxy) is included, they can also track the evolution of the detector;
  - ➔ we do not need to model analytical pdf's by ourselves, the tool will take care of that.

- Disadvantages:
  - ➔ they need to be re-trained at every release change;
  - ➔ typically they are "black boxes", so it is not easy to check what's going on under the hood. (But we test them on independent control samples, so we know that they are reliable).

Multivariate approaches – leptonID

We have been using BDT’s for a few years now, to have better lepton ID, especially at low momentum, using as input:

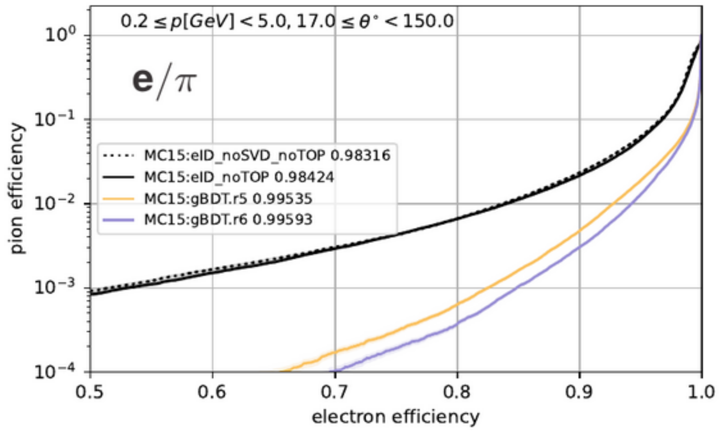


Variable	Range	Description
E/p [c]	–	Ratio of cluster energy over track momentum.
E_1/E_9	–	Ratio of the energy of the seed crystal over the energy sum of the 9 surrounding crystals.
E_9/E_{21}	–	Ratio of the energy sum of 9 crystals surrounding the seed over the energy sum of the 25 surrounding crystals (minus 4 corners).
Cluster LAT	–	Cluster lateral energy moment [17].
$\lvert Z_{40}\rvert$	–	Zernike moment $n = 4$, $m = 0$, calculated in a plane orthogonal to the EM shower direction.
$\lvert Z_{51}\rvert$	–	Zernike moment $n = 5$, $m = 1$, calculated in a plane orthogonal to the EM shower direction.
Z_{MVA}	–	Score of BDT trained on 11 Zernike moments.
ΔL [cm]	–	Projection on the extrapolated track direction of the distance between the track entry point in the ECL and the cluster centroid.
PSD_{MVA}	–	Score of a BDT trained to classify clusters as originated by an EM or hadronic shower, using crystal-level info including waveform pulse shape.
$\Delta\log\mathcal{L}(\ell/\pi)_{\mathrm{CDC}}$ (binary)	–	Log-likelihood difference between $\ell - \pi$ hypothesis in the CDC (binary)
$\mathcal{L}_\ell^{\mathrm{CDC}}/\sum_i \mathcal{L}_i^{\mathrm{CDC}}$ (multi-class)	–	Global lepton likelihood ratio in the CDC (multi-class).
$\Delta\log\mathcal{L}(\ell/\pi)_{\mathrm{TOP}}$ (binary)	ECL Barrel	Log-likelihood difference between $\ell - \pi$ hypothesis in the TOP (binary)
$\mathcal{L}_\ell^{\mathrm{TOP}}/\sum_i \mathcal{L}_i^{\mathrm{TOP}}$ (multi-class)	ECL Barrel	Global lepton likelihood ratio in the TOP (multi-class).
$\Delta\log\mathcal{L}(\ell/\pi)_{\mathrm{ARICH}}$ (binary)	ECL FWD endcap	Log-likelihood difference between $\ell - \pi$ hypothesis in the ARICH (binary)
$\mathcal{L}_\ell^{\mathrm{ARICH}}/\sum_i \mathcal{L}_i^{\mathrm{ARICH}}$ (multi-class)	ECL FWD endcap	Global lepton likelihood ratio in the ARICH (multi-class).
$\Delta\log\mathcal{L}(\mu/\pi)_{\mathrm{KLM}}$ (binary)	$p_{\mathrm{lab}} > 0.6$ GeV/c	Log-likelihood difference between $\ell - \pi$ hypothesis in the KLM (binary)
$\mathcal{L}_\ell^{\mathrm{KLM}}/\sum_i \mathcal{L}_i^{\mathrm{KLM}}$ (multi-class)	$p_{\mathrm{lab}} > 0.6$ GeV/c	Global lepton likelihood ratio in the KLM (multi-class).

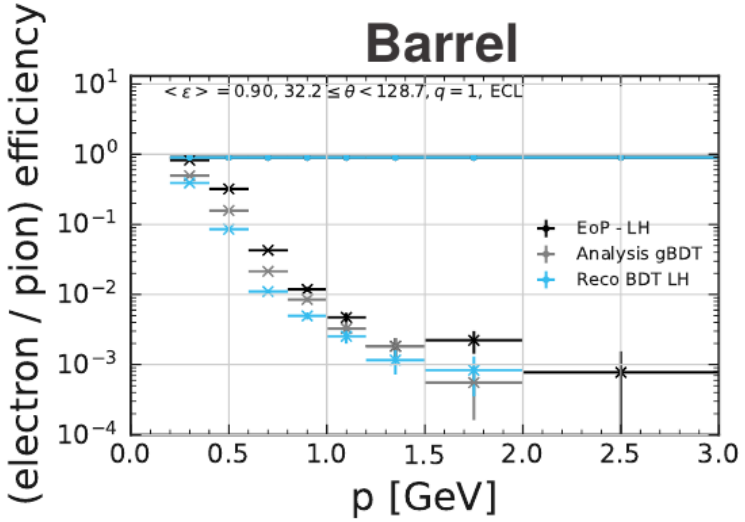
From BELLE2-NOTE-TE-2021-011

Multivariate approaches – leptonID

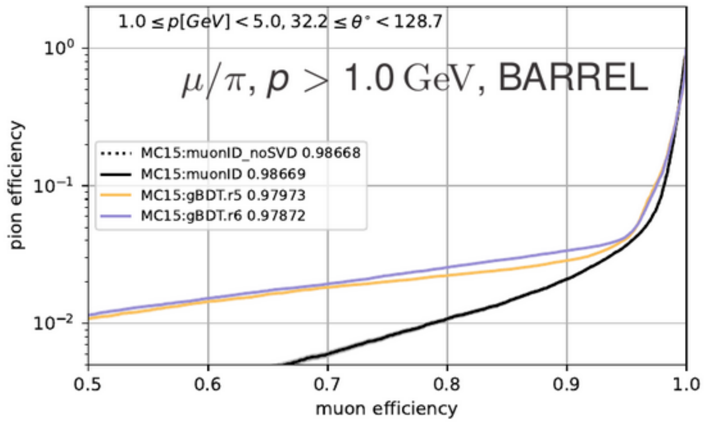
Already available, BDT's using analysis level information:

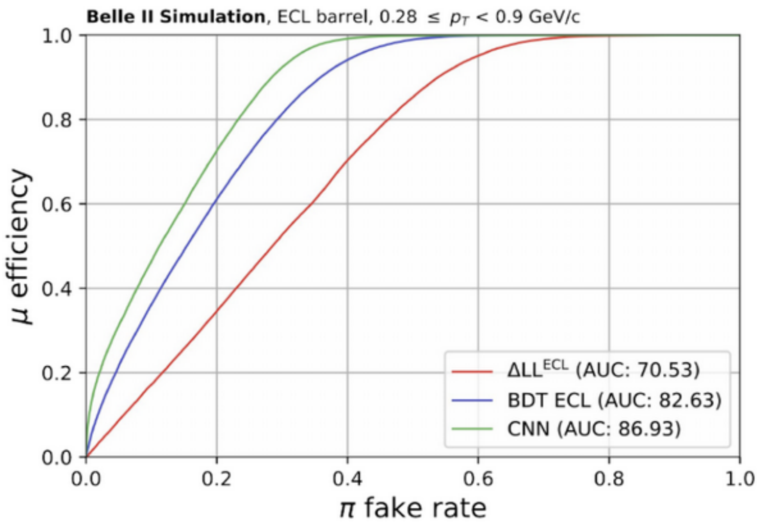


Coming soon, BDT's and CNN's using also reco level information:



In general, BDT's give better performance, but it is not always the case: please check what is best for your analysis!

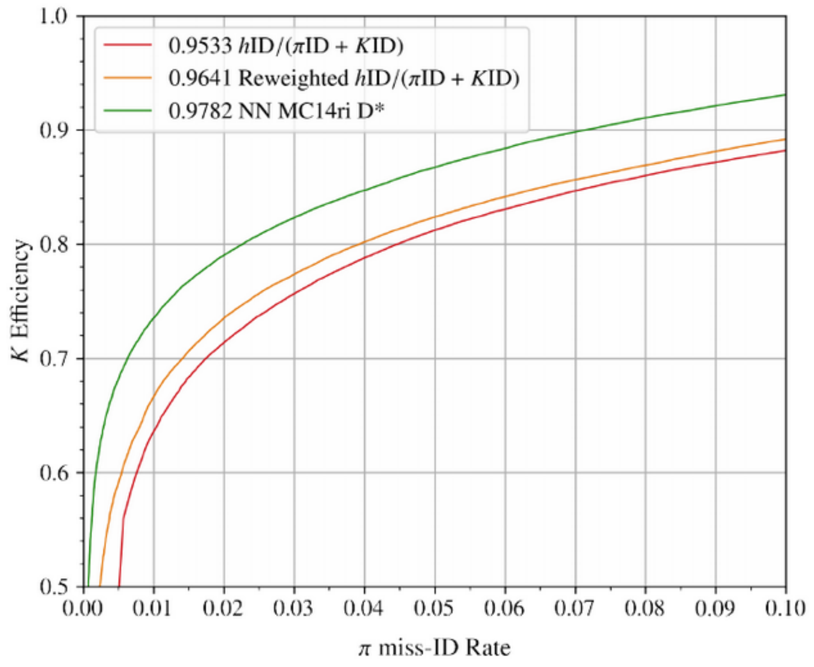


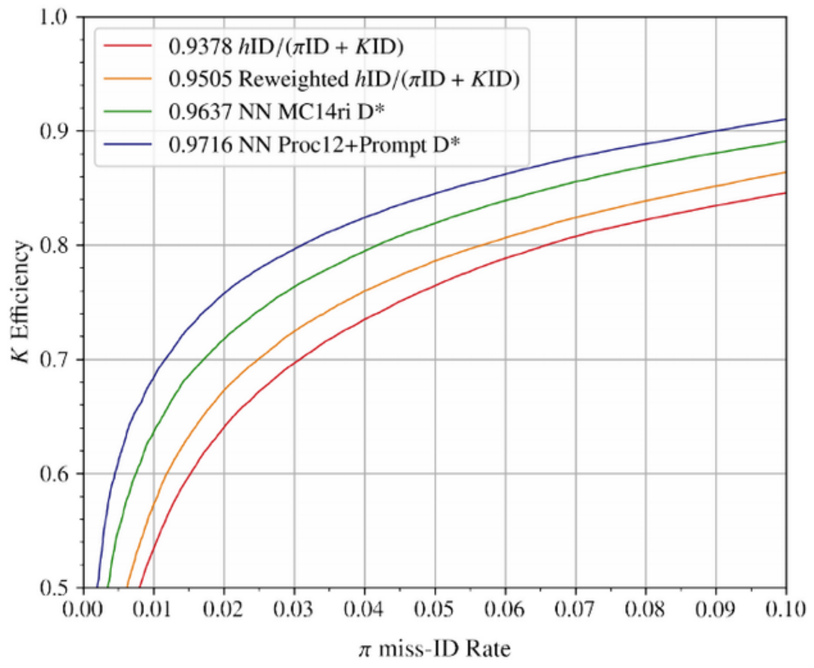


For more information, please see Marcel's and Anja's talks.

Multivariate approaches – hadronID

- A similar effort is ongoing to improve hadronID (in particular K/π separation);
- Neural Network combining analysis level information;
- We can actually train on real data, and improve the separation power:





For more details, please see Stefan's talk

PID performance and corrections

Why measuring PID performance

- I think that we agree that having good PID is likely to improve your analysis, now we want to ask ourselves whether we need to know precisely how good it is;

- In some cases (e.g. measurement of the mass of a resonance, of its lifetime, ...), you can use PID to improve the S/N ratio and just be happy with it, you main result will not depend significantly on the PID performance;

- In other cases it is more important:
  - ➔ you want to measure a branching ratio and take the reconstruction efficiency from the MC. If PID in data is different than in MC and you ignore this difference, you will end up with a bias;
  - ➔ you want to measure a charge asymmetry. PID efficiencies are slightly different for positive and negative particles, if you ignore this you might create a bias;

- In general we want to know exactly how PID works in data and in MC, and we want to know the uncertainty associated to these measurements.

Efficiency and fake rates

Definitions:

- **Efficiency**: probability that the particle α is correctly identified by the PID selection (intended to select α and reject the others);

- **β mis-ID probability (or fake rate)**: probability that the particle β passes the criteria meant to select α particles.

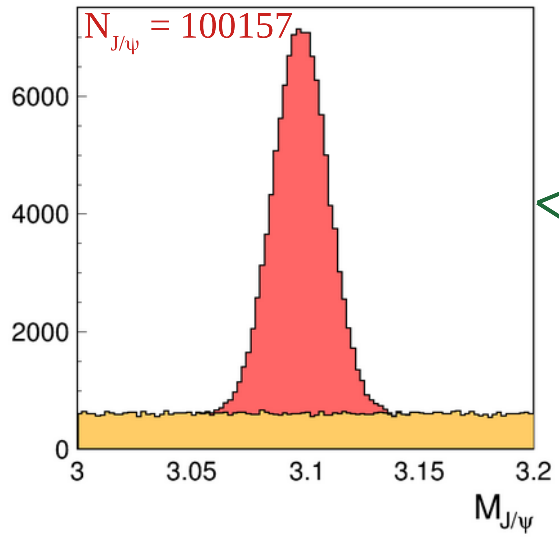
Knowing your efficiencies and fake rates is key to understand your overall signal (backgrounds) selection efficiencies.

Measuring efficiencies and fake rates

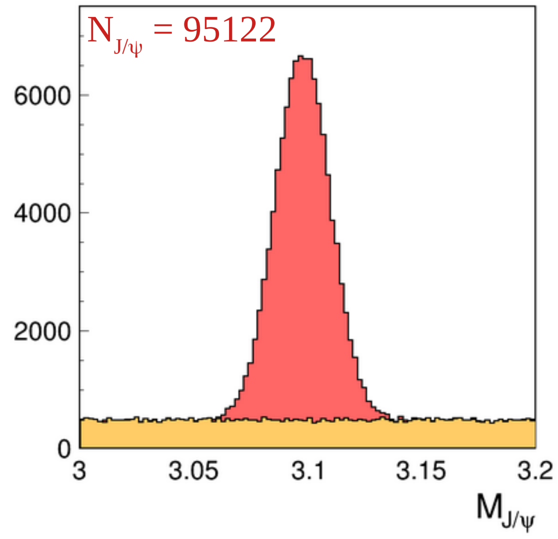
- We measure efficiencies and fake rates using control samples that are very clean, that is we know what particles they are before applying any PID selection;

- Example: $J/\psi \rightarrow \mu^+\mu^-$ with the tag-and-probe method:
  1) we select pairs of tracks with invariant mass ~3 GeV/c^2;
  2) we require that e.g. the negative track (tag) is consistent with being a muon;
  3) we do not make any PID requirement on the other track (probe);

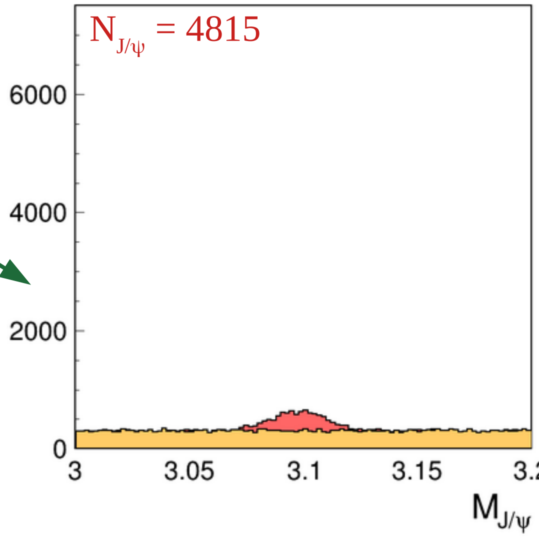
No PID selection on positive track



$P(\mu) > x$



$P(\pi) > y$



From this we derive that:

The μ **efficiency** (for a cut at x) is **~95%**

The $\mu \rightarrow \pi$ **mis-ID** probability (for a cut at y) is **~5%**

Control samples used at Belle II

We can use several control samples for all the particles we are interested in:

- e: $e^+e^- \rightarrow e^+e^-\gamma$, $e^+e^- \rightarrow e^+e^-e^+e^-$, $J/\psi \rightarrow e^+e^-$
- μ: $e^+e^- \rightarrow \mu^+\mu^-\gamma$, $e^+e^- \rightarrow e^+e^-\mu^+\mu^-$, $J/\psi \rightarrow \mu^+\mu^-$
- π: $K^0_S \rightarrow \pi^+\pi^-$, $D^0 \rightarrow K^-\pi^+$, $\Lambda^0 \rightarrow p\,\pi^-$, $\tau^+ \rightarrow \pi^+\pi^-\pi^+\nu$
- K: $D^0 \rightarrow K^-\pi^+$, $D^0 \rightarrow K^-\pi^+\pi^0$, $D_s^+ \rightarrow \phi(K^+K^-)\pi^+$, $\tau^+ \rightarrow K^+K^-\pi^+\nu$
- p: $\Lambda^0 \rightarrow p\,\pi^-$

Getting PID efficiencies in data/MC

You are welcome to select your own control sample, but we (the PID group):

1) centrally produce ntuples for all the maintained control samples;
2) data and MC efficiency tables (to be discontinued);
3) a software tool that allow you to use the centrally produced ntuples to match your specific analysis needs: the Systematics Framework (which is going to become the standard)

PID correction tables

- The PID tables are simple text files that list the probabilities for a certain particle to pass some (common) PID selection;

- The tables:
  - are in bins of (p, θ/cosθ);
  - are produced separately for positive and negative particles;
  - may combine different control samples;
  - report the statistical uncertainty on the efficiencies and the data/MC ratios;
  - report our recommended systematic uncertainty.

PID correction tables – an example

Let's take a look at the muonID correction tables, from:

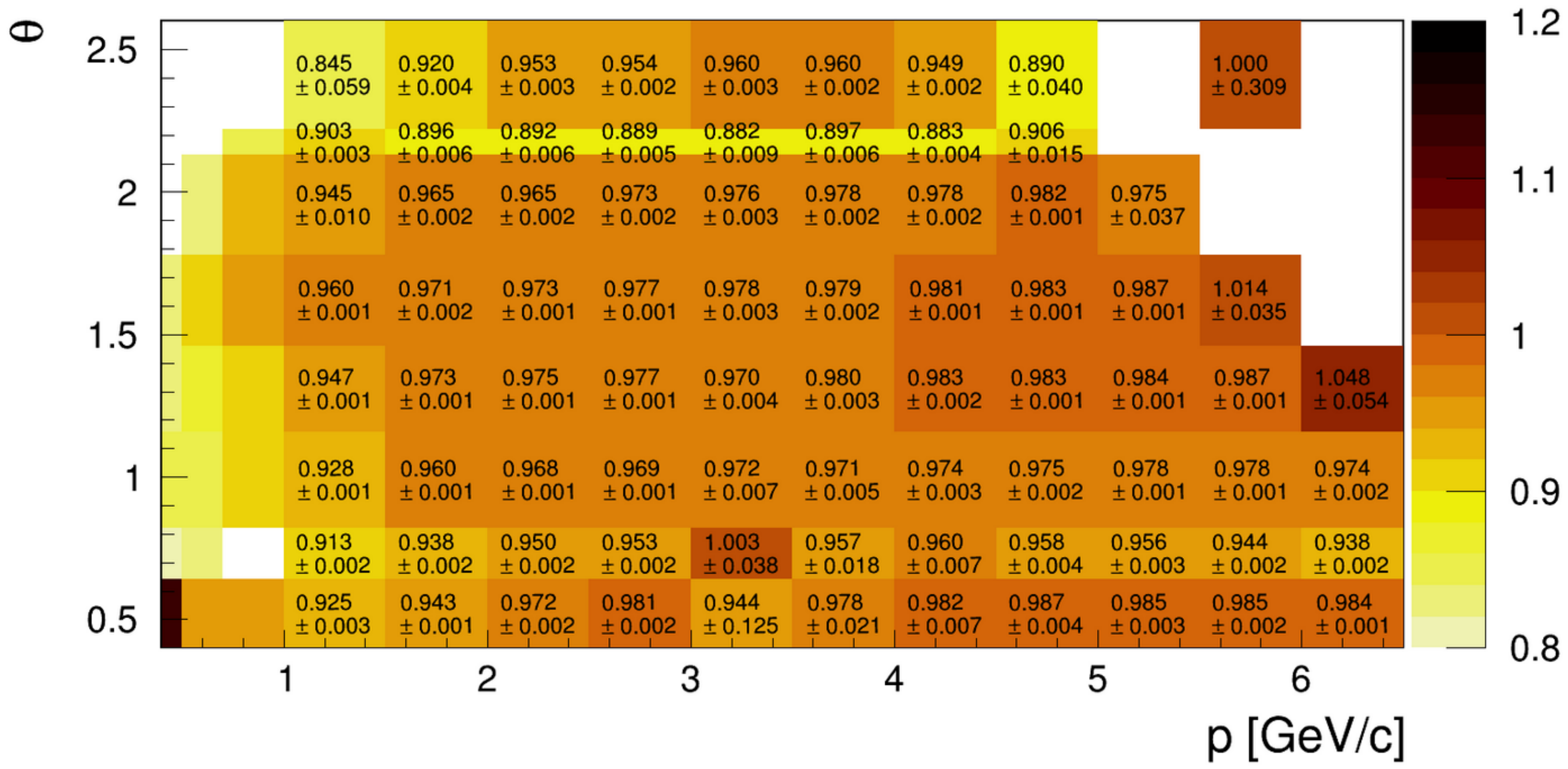
```
/gpfs/group/belle2/users/mmilesi/perf/PID/methods/post_ichep_2022/proc12prompt/MC14ri_a/v3/efficiency
```

```
channel,n_in_comb,is_final_comb,is_best_available,n_duplicates,is_unique,is_outlier,is_abnormal,is_sumqu
ad,is_stat_clipped_dn,is_sys_clipped_dn,is_statsys_clipped_dn,variable,charge,p_min,p_max,theta_min,thet
a_max,working_point,threshold,data_MC_ratio,data_MC_uncertainty_stat_dn,data_MC_uncertainty_stat_up,data
_MC_uncertainty_sys_dn,data_MC_uncertainty_sys_up,data_MC_uncertainty_statsys_dn,data_MC_uncertainty_sta
tsys_up,rel_data_MC_uncertainty_stat_dn,rel_data_MC_uncertainty_stat_up,rel_data_MC_uncertainty_sys_dn,r
el_data_MC_uncertainty_sys_up,rel_data_MC_uncertainty_statsys_dn,rel_data_MC_uncertainty_statsys_up,data
_MC_DISTsys_dn,data_MC_DISTsys_up

jpsill_VS_eell_VS_mumugamma,3,True,True,0,True,False,False,True,False,False,False,binaryMuonID_noSVD_pi,
+,1.0,1.5,0.4,0.64,FixedThresh05,0.500000000000000,0.94402831,0.00052499,0.00052333,0.00296079,0.0028723
6,0.00300697,0.00291964,0.05561142,0.05543582,0.3136336,0.30426618,0.31852577,0.30927502,0.0,0.0

jpsill_VS_eell_VS_mumugamma,3,True,True,0,True,False,False,True,False,False,False,binaryMuonID_noSVD_pi,
-,1.0,1.5,0.4,0.64,FixedThresh05,0.500000000000000,0.94026456,0.00051998,0.00051844,0.00301595,0.0028742
8,0.00306044,0.00292067,0.05530109,0.05513755,0.32075499,0.30568883,0.32548728,0.31062165,0.0,0.0

jpsill_VS_eell_VS_mumugamma,3,True,True,0,True,False,False,True,False,False,False,binaryMuonID_noSVD_pi,
+,1.0,1.5,0.64,0.82,FixedThresh05,0.500000000000000,0.93683487,0.0004638,0.00046287,0.00178229,0.0016815
8,0.00184164,0.00174412,0.04950675,0.04940813,0.19024544,0.17949574,0.1965814,0.18617165,0.0,0.0

[...]
```

PID correction tables – an example

Let's plot one specific example:

Particle:	μ^+
Variable:	muonID_noSVD
Threshold:	0.9

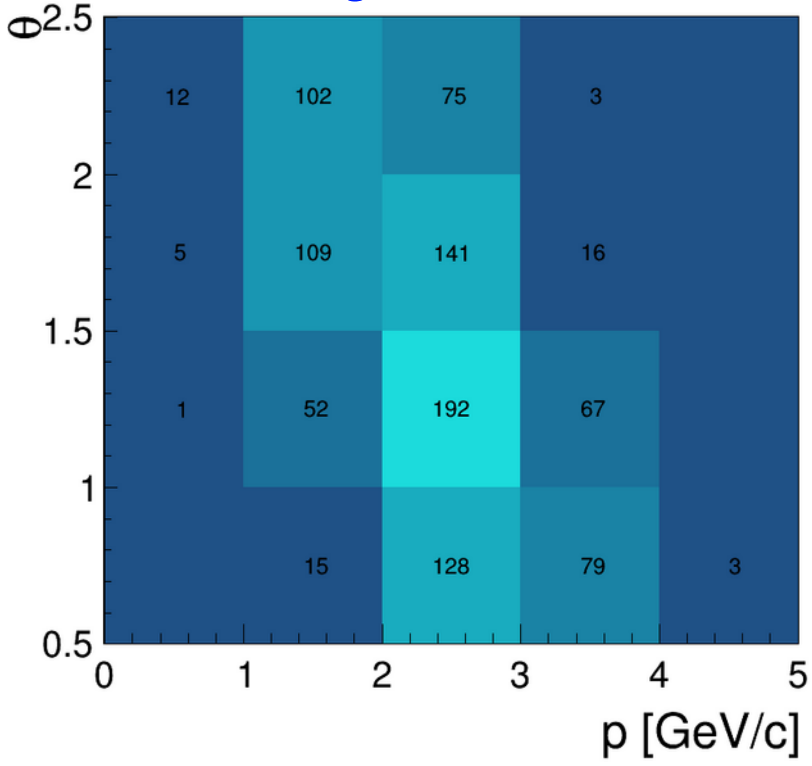


The plot shows the data/MC ratio, as measured on the official control sample(s), for μ^+ passing the selection muonID_noSVD > 0.9
The uncertainty plotted is the sum in quadrature of stat. and syst. uncertainties.

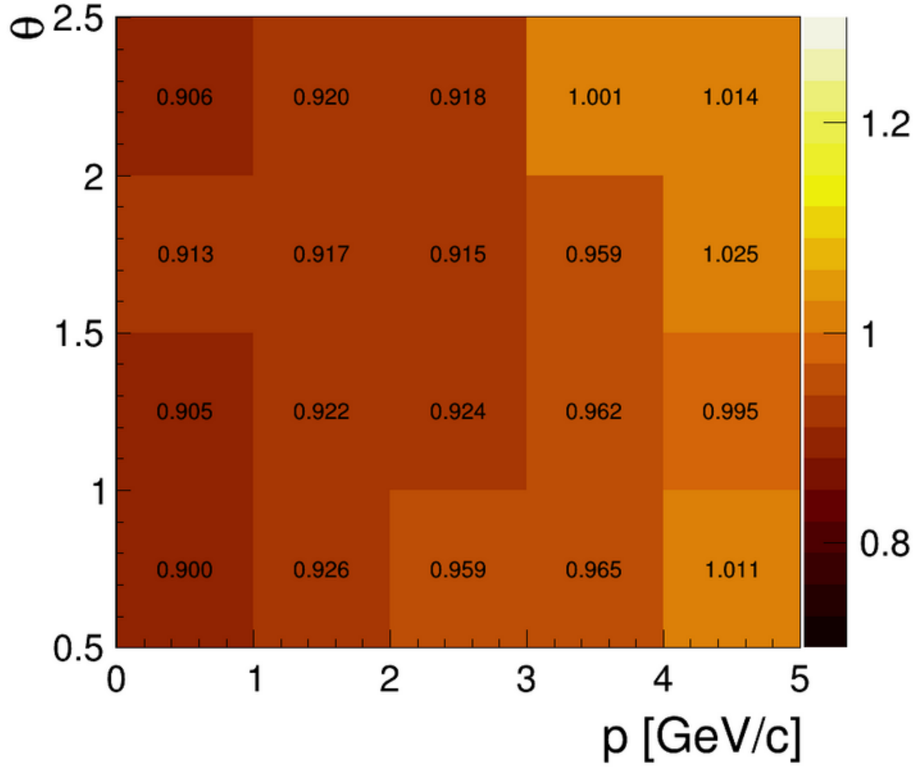
PID tables at work

- Suppose we are looking for $D^+ \rightarrow K^+ \pi^0$. The MC predicts that, with the integrated luminosity we have and taking into account all the selection efficiencies, we expect 1000 signal events, with the (p, θ) spectrum shown on the left;
- The data/MC correction table for our K PID selection is shown on the right;

Signal model



kaonID data/MC correction table



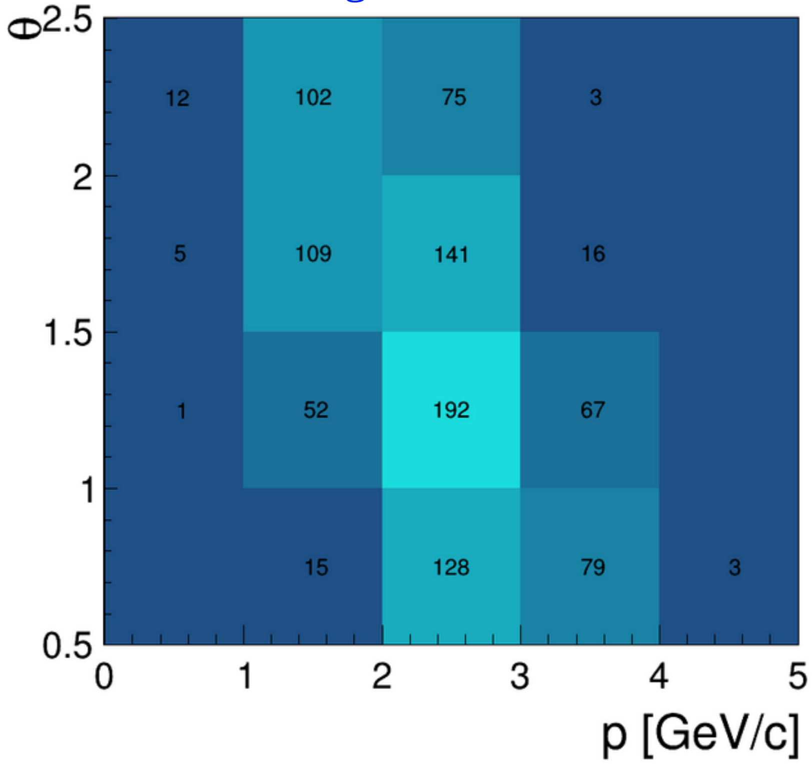
What do we expect to find in the data?

PID tables at work

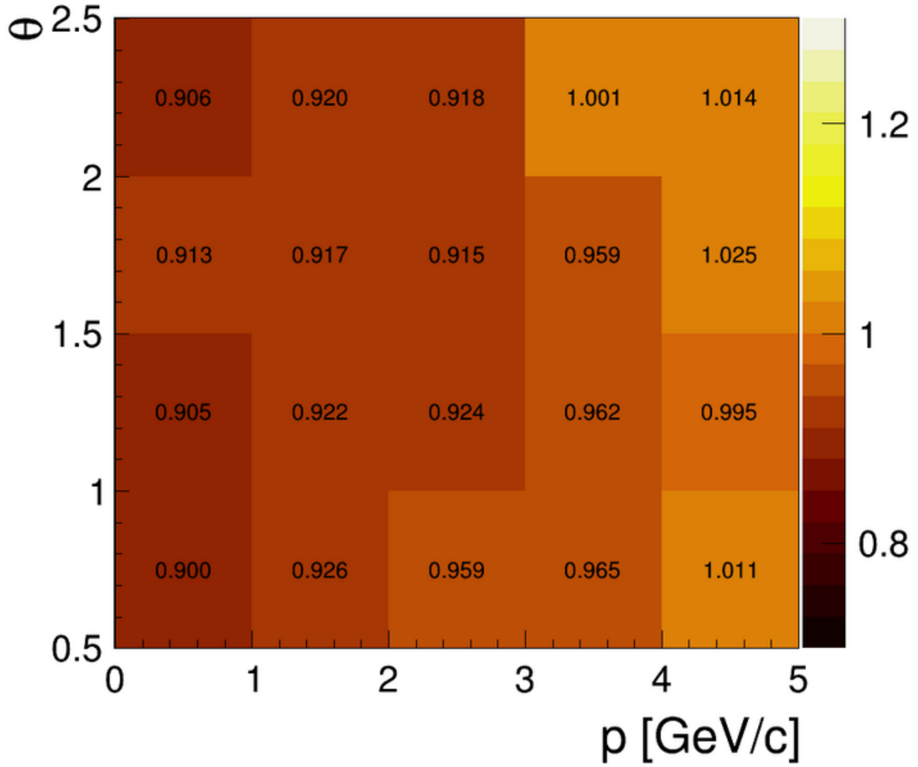
- What we have to compute (for every bin of the distribution) is:

$$N_{exp}^{data} = \frac{\varepsilon^{data}}{\varepsilon^{MC}} \cdot N_{exp}^{MC}$$

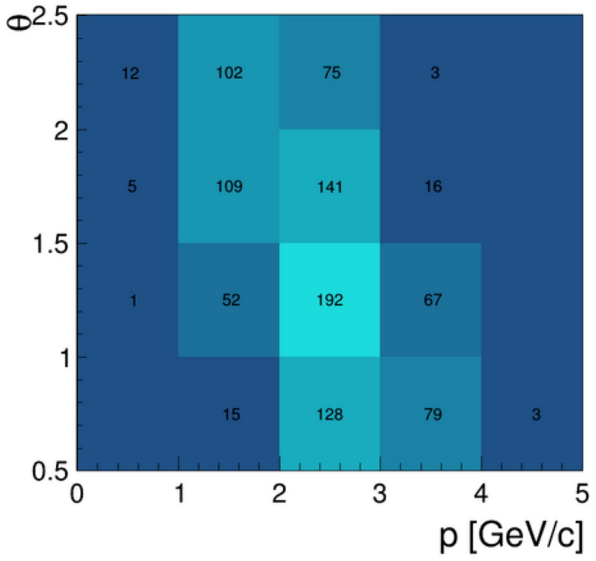
Signal model



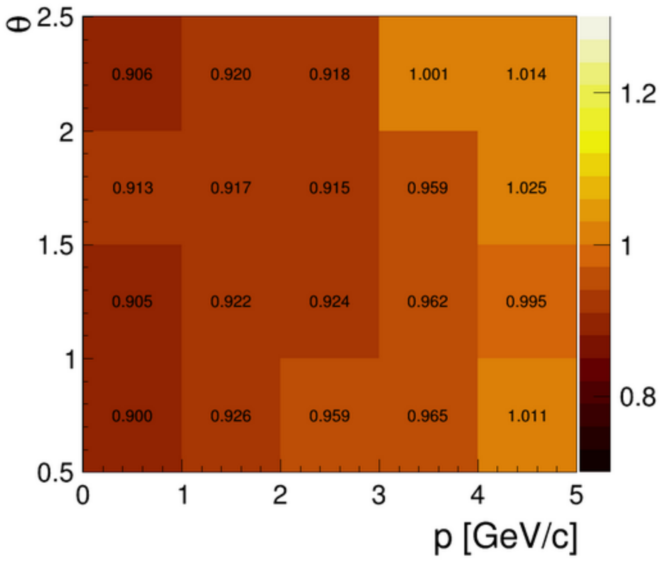
kaonID data/MC correction table



PID tables at work

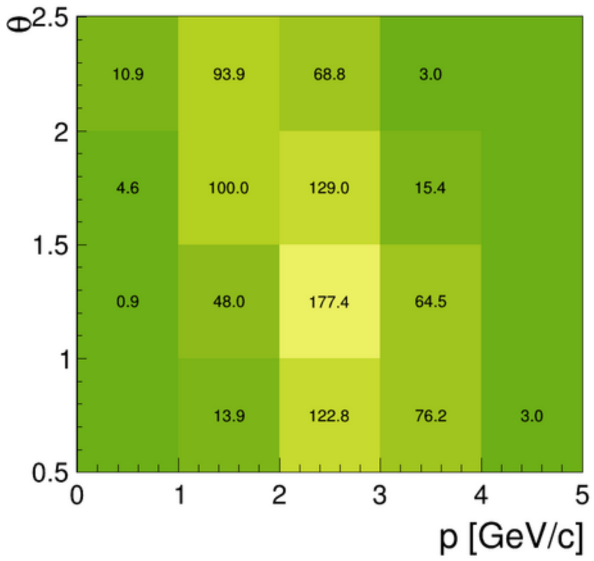


x



=

Corrected signal model



- After applying the PID corrections, we expect 932.2 signal events, because in this example kaonID is less efficient in data than in the MC;
- Also the (p, θ) spectum is slightly distorted wrt the original distribution, because the data/MC efficiency ratio is not uniform across the phase space.

NB: this is a completely dummy example, with numbers I just made up.

PID tables at work – some comments

- What I discussed above is a very simplified situation;

- In real life you will have to:
  - deal with reconstruction efficiencies, not so much with signal events;
  - multiply the correction factors for more than one tracks in the final state, associating to each the correct bin;
  - distinguish between positive and negative tracks;
  - propagate the uncertainties (those that come from the statistics of your signal MC, from the other efficiencies in your analysis, and from the PID tables).

PID tables – basf2 helpers

- Above I described the "by hand" method of applying the PID corrections, which you can apply to the ntuples you already have;

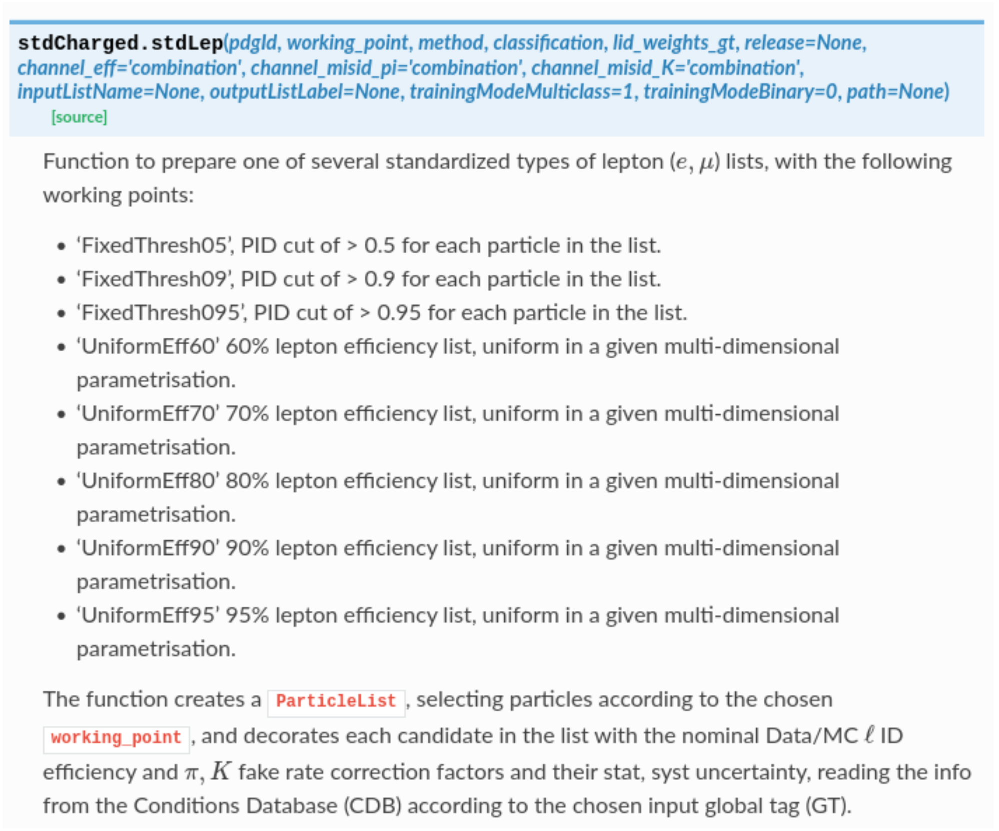
For leptonID only:

- If you still have to produce your ntuples, and the PID correction tables are already available from the database (check the Lepton ID confluence page), you can use the basf2 helpers, that will automatically apply the corrections.

  You can take a look at the example steering file:

  B2A908-ApplyLIDWeights.py

  to see how they work.



```
stdCharged.stdLep(pdgId, working_point, method, classification, lid_weights_gt, release=None,
channel_eff='combination', channel_misid_pi='combination', channel_misid_K='combination',
inputListName=None, outputListLabel=None, trainingModeMulticlass=1, trainingModeBinary=0, path=None)
    [source]
```

Function to prepare one of several standardized types of lepton (e, μ) lists, with the following working points:

- 'FixedThresh05', PID cut of > 0.5 for each particle in the list.
- 'FixedThresh09', PID cut of > 0.9 for each particle in the list.
- 'FixedThresh095', PID cut of > 0.95 for each particle in the list.
- 'UniformEff60' 60% lepton efficiency list, uniform in a given multi-dimensional parametrisation.
- 'UniformEff70' 70% lepton efficiency list, uniform in a given multi-dimensional parametrisation.
- 'UniformEff80' 80% lepton efficiency list, uniform in a given multi-dimensional parametrisation.
- 'UniformEff90' 90% lepton efficiency list, uniform in a given multi-dimensional parametrisation.
- 'UniformEff95' 95% lepton efficiency list, uniform in a given multi-dimensional parametrisation.

The function creates a `ParticleList`, selecting particles according to the chosen `working_point`, and decorates each candidate in the list with the nominal Data/MC ℓ ID efficiency and π, K fake rate correction factors and their stat, syst uncertainty, reading the info from the Conditions Database (CDB) according to the chosen input global tag (GT).

- For hadronID we do not plan to have them, for the reason I will now explain.

Shortcomings of the PID tables

- PID tables do a great job in “standard” cases. However, as the number and complexity of the analyses grow, this approach starts showing its limitations;

- Suppose that:
  - you do not use all the data for a specific campaign;
  - you use many different thresholds (because of the optimization strategy you chose);
  - you cut on some variables that are correlated to PID performance (e.g. *isolation score*, *timeSinceLastInjection*, *nCDCHits*, ...);
  - you need a specific binning or want to bin on variables other than p, θ;
  - you do anything non-standard;
  - …

- One would have to centrally produce an infinite number of tables...

The Systematics Framework

- In recent times we have started using this tool (mostly developed by Sviat), beginning with the hadron ID samples;
- The SF is documented at:

  https://syscorrfw.readthedocs.io/en/latest/index.html

- Advantages of the SF:
  - ➔ you can use the centrally produced control samples to produce the correction tables that suit your needs;
  - ➔ it uses sWeights, so you do not have to fit any distribution (this is done by the tool already);
  - ➔ if what you need is “standard”, scripts and examples are available, you can produce what you need in less than one hour;
  - ➔ if what you need is “non-standard”, there is ample margin for customization.

- The D^*, Λ^0, and K_S models are already available. More will come soon!

Using the SF

- Let’s look at the the kaonID efficiency for kaonID > 0.9 as a function of the momentum, for bucket36;
- I will take:

`/group/belle2/dataprod/Systematics/systematic_corrections_framework/scripts/efficiency_table.py`

and pass it a json file that looks like:

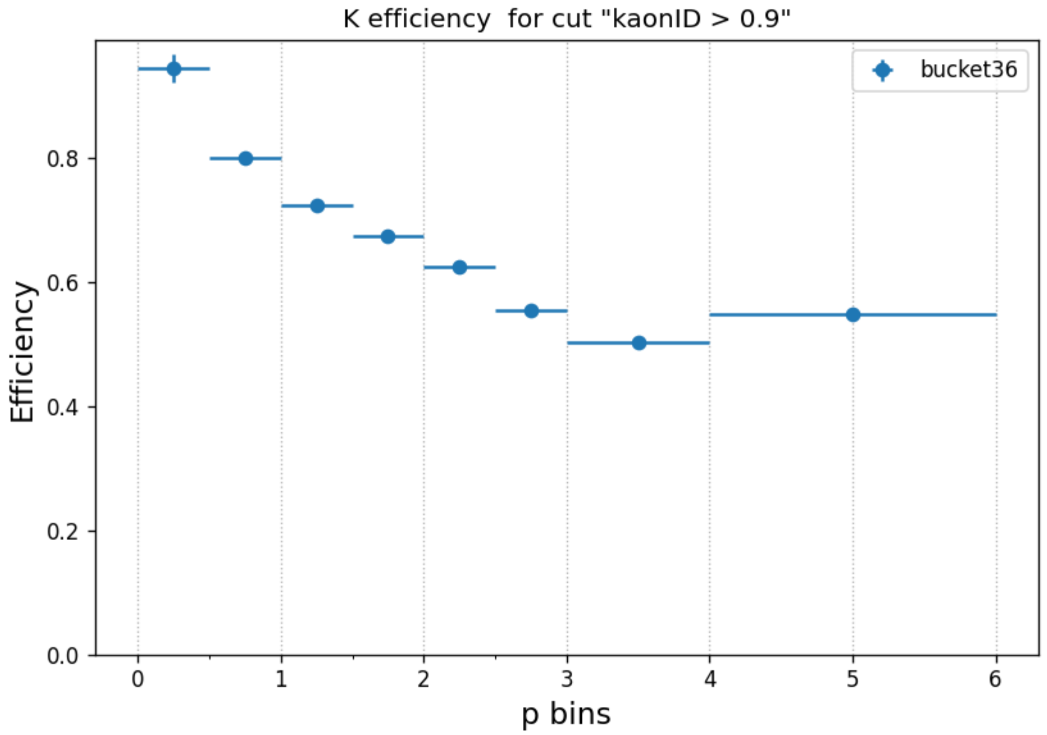
```
{
        "display_plots": true,
        "save_plots": "kaon_efficiency_bucket36.png",
        "cut": "kaonID > 0.9",
        "particle_type": "K",
        "model_names": ["Dst"],
        "data_query": ["bucket36"],
        "precut": "nCDCHits > 20",
        "track_variables": ["p"],
        "binning": [ [0.0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 4.0, 6.0] ],
        "output": "kaon_efficiency_bucket36.json",
        "use_percent": false
}
```

you can specify your selection cuts

and also your binning

Using the SF

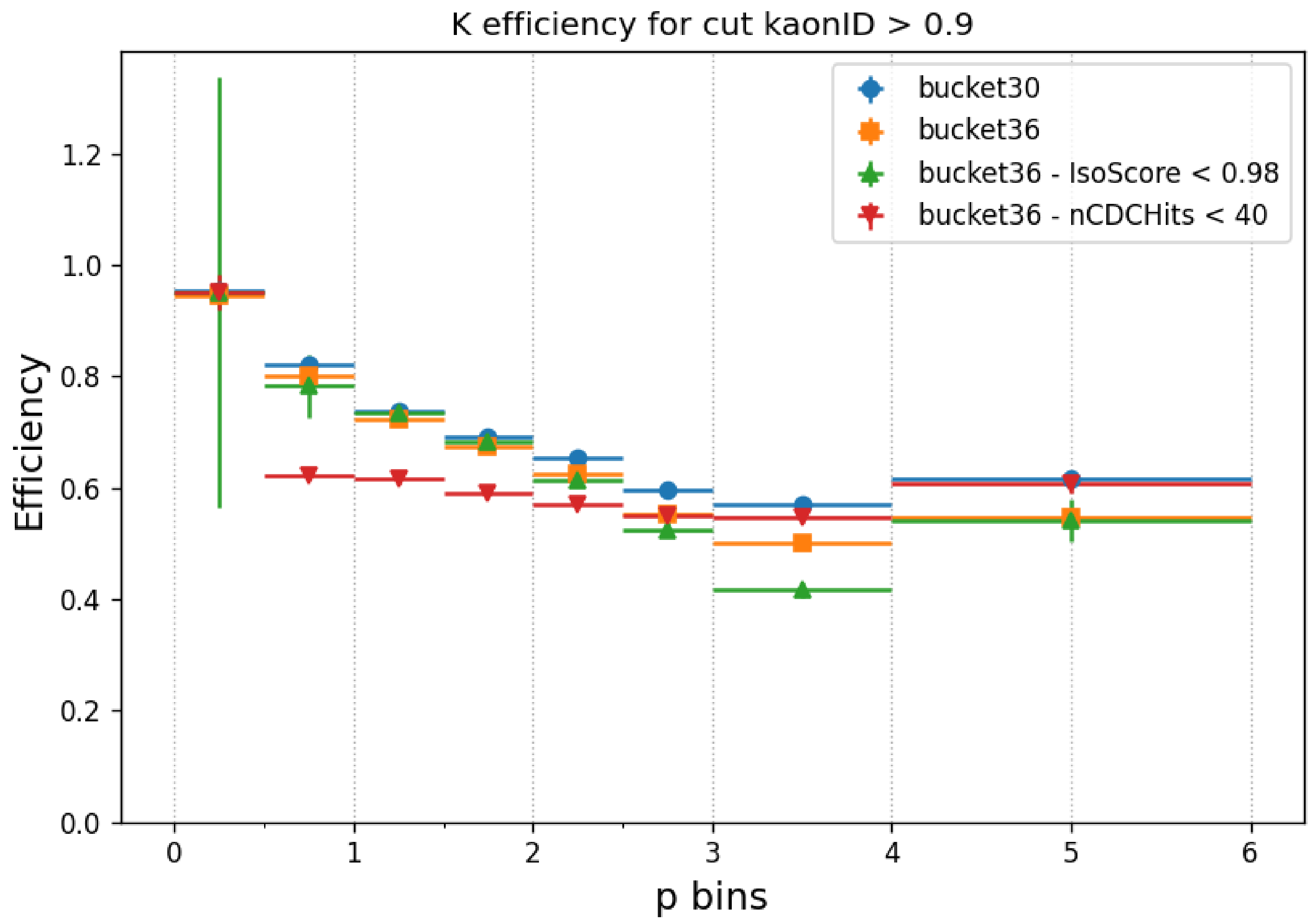
- As output, I get a json file and a plot, that look like:



```
{
    "efficiency_table": [
        0.9447301371322525,
        0.8011868787651882,
        0.7246955232336442,
        0.6751531728809061,
        0.625145684211008 7,
        0.5545854810063586,
        0.5024453742794931,
        0.548246657011927
    ],
    "bin_variables": [
        "p"
    ],
    "stat_error": [
        0.008473513956011317,
        0.0038989983184021656,
        0.003837312798146121,
        0.003831735881289033,
        0.004008309262378364,
        0.004494764509379299,
        0.004399615437385293,
        0.009028968348707865
    ],
    "total_syst_error": [
        0.020355428701701594,
        0.003784864441073177,
        0.0019675983847400325,
        0.001713641524671683,
        0.0014640717092725406,
        0.001355346520599 21,
        0.001070342769574073,
        0.0002854923821761224
    ],
    …
```

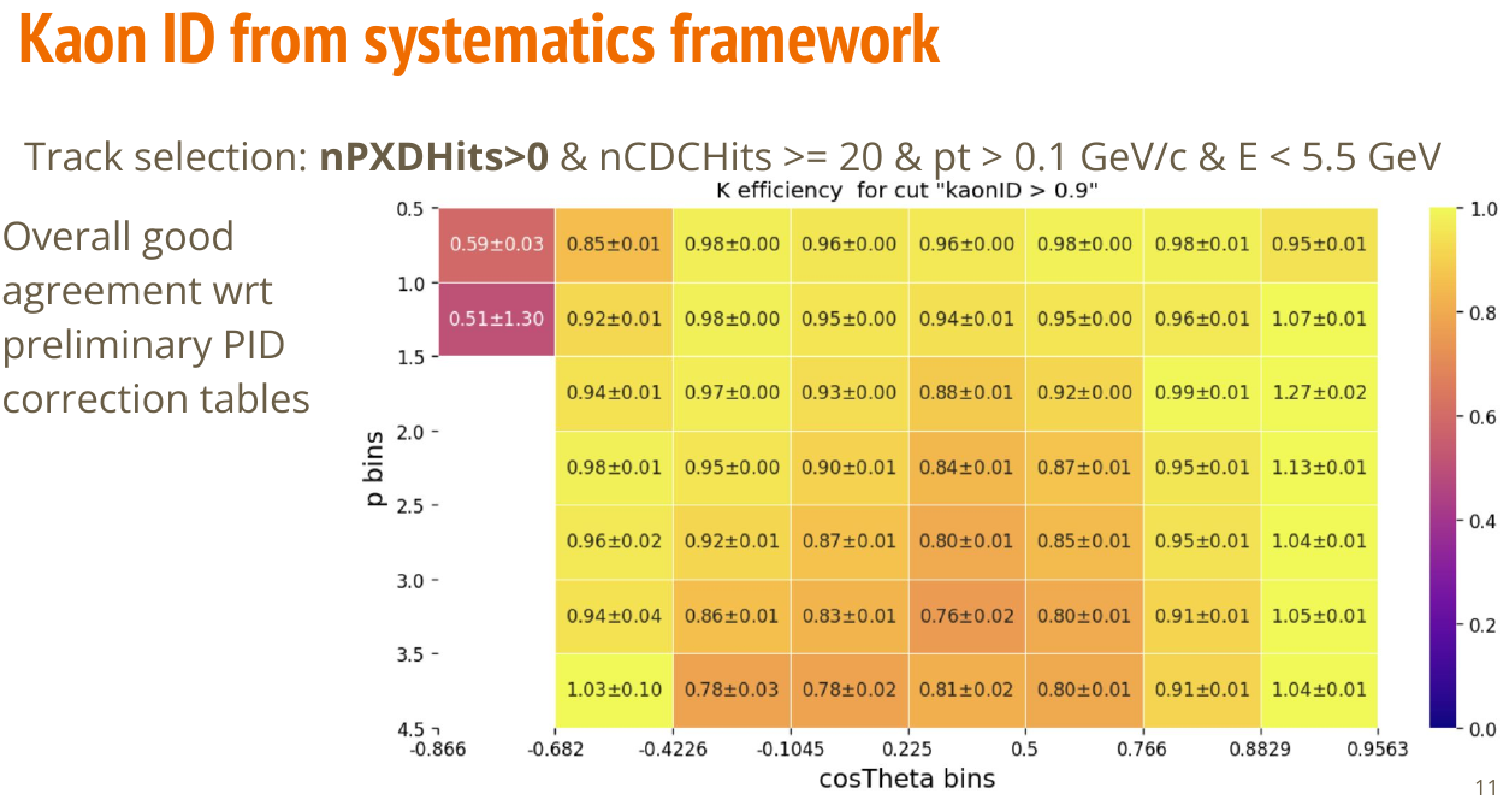
Using the SF

- You can compare different data periods and different tracks selections and see if it matters for PID:



Producing data/MC weights

This is already used by analysts shooting for publication on the Moriond 2023 data set:



Kaon ID from systematics framework

Track selection: **nPXDHits>0** & nCDCHits >= 20 & pt > 0.1 GeV/c & E < 5.5 GeV

Overall good agreement wrt preliminary PID correction tables

K efficiency for cut "kaonID > 0.9"

0.59±0.03	0.85±0.01	0.98±0.00	0.96±0.00	0.96±0.00	0.98±0.00	0.98±0.01	0.95±0.01
0.51±1.30	0.92±0.01	0.98±0.00	0.95±0.00	0.94±0.01	0.95±0.00	0.96±0.01	1.07±0.01
	0.94±0.01	0.97±0.00	0.93±0.00	0.88±0.01	0.92±0.00	0.99±0.01	1.27±0.02
	0.98±0.01	0.95±0.00	0.90±0.01	0.84±0.01	0.87±0.01	0.95±0.01	1.13±0.01
	0.96±0.02	0.92±0.01	0.87±0.01	0.80±0.01	0.85±0.01	0.95±0.01	1.04±0.01
	0.94±0.04	0.86±0.01	0.83±0.01	0.76±0.02	0.80±0.01	0.91±0.01	1.05±0.01
	1.03±0.10	0.78±0.03	0.78±0.02	0.81±0.02	0.80±0.01	0.91±0.01	1.04±0.01

E. Ganiev et al., Search for B $\rightarrow$ K ν $\bar{\nu}$

Producing data/MC weights

- You can produce your own data/MC weight tables with the script:

`/group/belle2/dataprod/Systematics/systematic_corrections_framework/scripts/weight_table.py`

and a json file similar to:

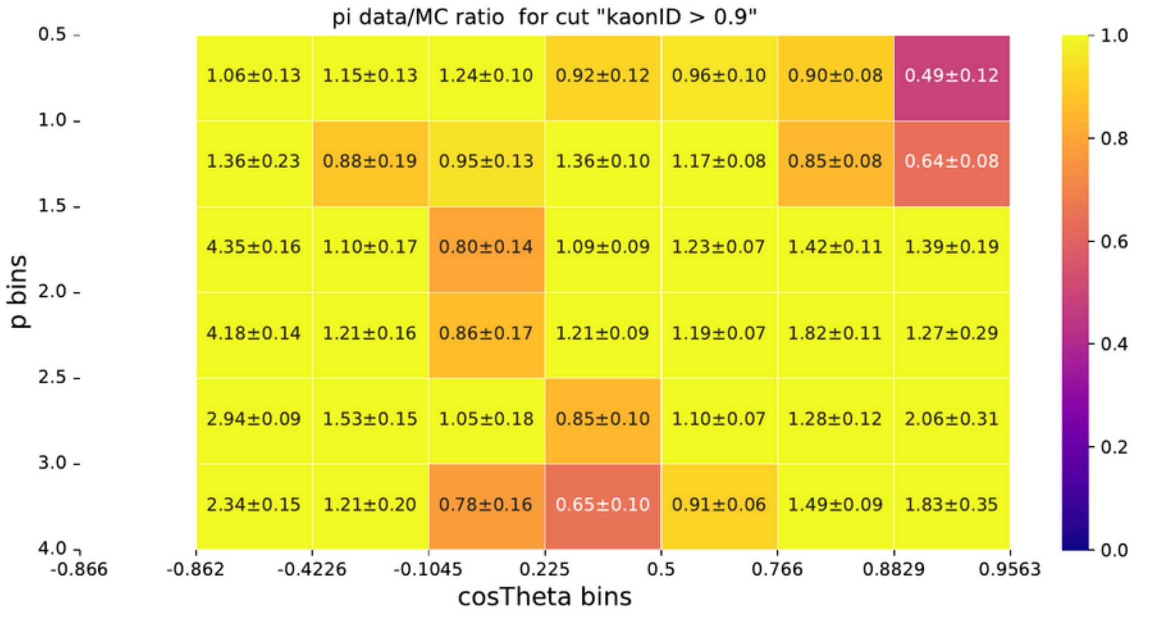
```
{
    "display_plots": true,
    "save_plots": "weights_pions_faking_kaons_proc13_chunk2.png",
    "cut": "kaonID > 0.9",
    "particle_type": "pi",
    "model_names": ["Dst"],
    "data_query": ["proc13_chunk2"],
    "mc_query": ["MC15ri_1"],
    "track_variables": ["p", "cosTheta"],
    "precut": "nCDCHits > 20",
    "event_variables": null,
    "binning": [ [0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 4.0],
                 [-0.866, -0.862, -0.4226, -0.1045, 0.225, 0.5, 0.766, 0.8829, 0.9563] ],
    "output": "weights_pions_faking_kaons_proc13_chunk2.json",
    "use_percent": false
}
```

here we are looking at the π faking K probability

the ratios will be between proc13_chunk2 and MC15ri

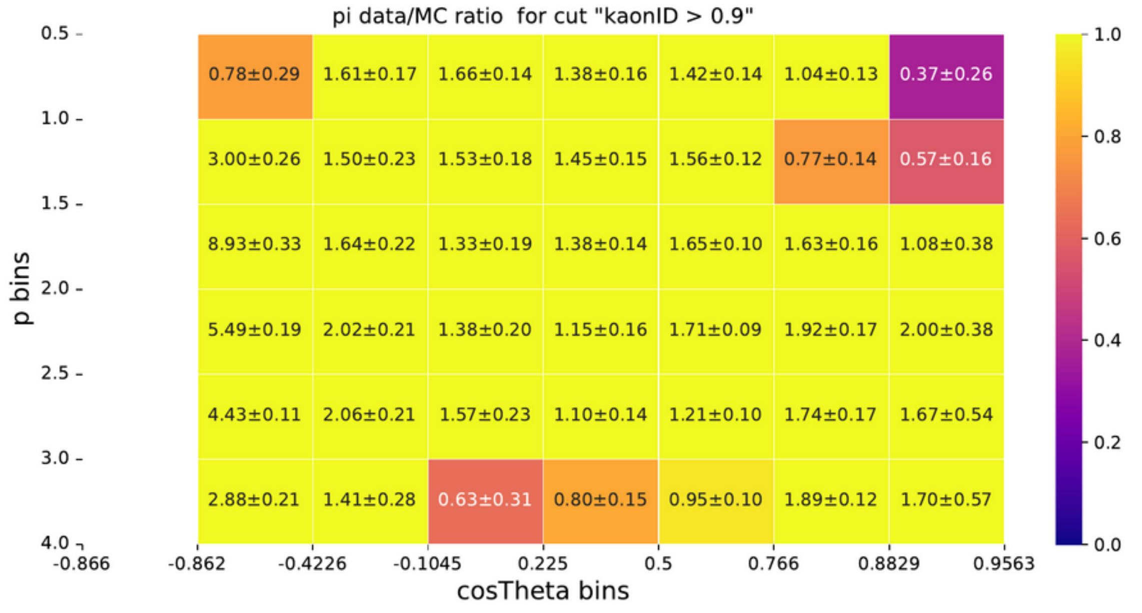
Producing data/MC weights

- Results:



1.06±0.13	1.15±0.13	1.24±0.10	0.92±0.12	0.96±0.10	0.90±0.08	0.49±0.12
1.36±0.23	0.88±0.19	0.95±0.13	1.36±0.10	1.17±0.08	0.85±0.08	0.64±0.08
4.35±0.16	1.10±0.17	0.80±0.14	1.09±0.09	1.23±0.07	1.42±0.11	1.39±0.19
4.18±0.14	1.21±0.16	0.86±0.17	1.21±0.09	1.19±0.07	1.82±0.11	1.27±0.29
2.94±0.09	1.53±0.15	1.05±0.18	0.85±0.10	1.10±0.07	1.28±0.12	2.06±0.31
2.34±0.15	1.21±0.20	0.78±0.16	0.65±0.10	0.91±0.06	1.49±0.09	1.83±0.35

proc13_chunk2



0.78±0.29	1.61±0.17	1.66±0.14	1.38±0.16	1.42±0.14	1.04±0.13	0.37±0.26
3.00±0.26	1.50±0.23	1.53±0.18	1.45±0.15	1.56±0.12	0.77±0.14	0.57±0.16
8.93±0.33	1.64±0.22	1.33±0.19	1.38±0.14	1.65±0.10	1.63±0.16	1.08±0.38
5.49±0.19	2.02±0.21	1.38±0.20	1.15±0.16	1.71±0.09	1.92±0.17	2.00±0.38
4.43±0.11	2.06±0.21	1.57±0.23	1.10±0.14	1.21±0.10	1.74±0.17	1.67±0.54
2.88±0.21	1.41±0.28	0.63±0.31	0.80±0.15	0.95±0.10	1.89±0.12	1.70±0.57

bucket36

Future SF developments

- More functionality is being added, mostly for the experienced and/or adventurous users;

1) You can perform your own kernel density estimation (KDE) fit to the distribution of your PID output variable (that you can later re-sample for later use):

   https://syscorrfw.readthedocs.io/en/latest/fit_pdf_tutorial.html

2) You can correct your MC PID "on the fly" while running basf2 (similar to what is currently done by the basf2 helpers for the lepton ID tables), but with better granularity:

   https://syscorrfw.readthedocs.io/en/latest/pid_resample_tutorial.html

Systematic uncertainties

- As always, this is a tricky subject and we cannot give a one-fits-all recipe;

- The tables and SF give you some systematics that are due to:
  - the quality of the fit (signal modeling, background uncertainties);
  - the consistency across different control samples in case there are more than one;
  - impact of trigger in selecting the control sample;
  - …

- There are others that might be specific to your analyis:
  - the track multiplicity is very different from that of our control samples;
  - our control samples are not fully adequate to represent your final state for other reasons;

- We have now better tools and variables (e.g. the track isolation score, that was recently implemented by Marco) than can assist you.

Final remarks

- Physics analysis is fun, but it is also a complicated business;
- There are many assumptions that you make when designing an analysis: please question the validity of as many as you can!

- PID can greatly help improving your analysis, we provide many variables and tools that will assist you increasing the significance of your signal (or improving your upper limit), but these should be used wisely;

- We (PID group) can certainly improve on many things, but we definitely need your help:
  - ➔ we cannot anticipate all your needs;
  - ➔ we learn a lot when we interact with analysts on real life cases!

- So please, don't be shy, get in touch with us if you think we can help and/or if you have interesting ideas!