Quick Refresher

- Automatically load libraries and provide information

```
$ eic-smear
Using eic-smear version: 1.1.1
Using these eic-smear libraries :
/Users/kkauder/software/lib/libeicsmear.dylib
/Users/kkauder/software/lib/libeicsmeardetectors.dylib
eic-smear [0] BuildTree("pythia.txt",".", -1, "log.txt")
```

- Load as usual, or with a convenient shortcut function

```
eic-smear [1] .L SmearMatrixDetector_0_1.cxx
eic-smear [2] auto d = BuildMatrixDetector_0_1();
# or
eic-smear [1] auto d = BuildByName("matrix")
```

- eic-smear may seg-fault (ROOT issue). Just load by hand.

```
root [0] ] gSystem->Load("libeicsmear"); gSystem->Load("libeicsmeardetectors");
```
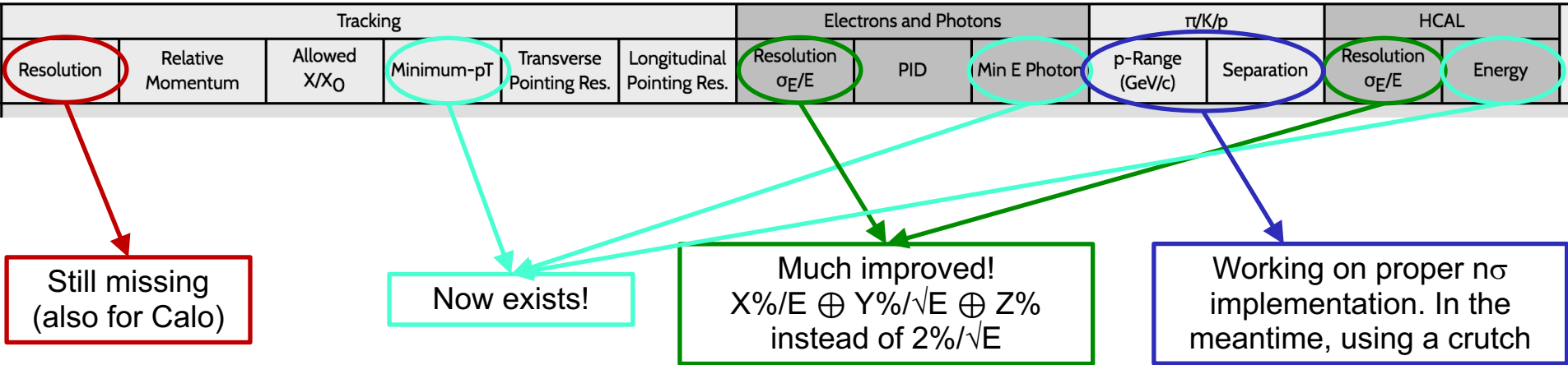
- Available on RCF, cvmfs, … or eic-smear, eicsmeardetectors

Detector Matrix Status

- **Detector Matrix**: Big picture view, evolving in the process, with links to details
    - Original plan was a series of releases at https://physdiv.jlab.org/DetectorMatrix/ with a versioned history
    - So far, only v0.1 has been released by the DWG conveners – delicate balance between "needed" and "possible"
- Matrix v0.2 is being finalized – but likely won't be released before next YR meeting in mid November
- I received a **private** preview, in order to have Fast Simulation ready ASAP after release

Improvements

- New details and better resolutions

Tracking						Electrons and Photons			π/K/p		HCAL	
Resolution	Relative Momentum	Allowed X/X_0	Minimum-pT	Transverse Pointing Res.	Longitudinal Pointing Res.	Resolution σ_E/E	PID	Min E Photon	p-Range (GeV/c)	Separation	Resolution σ_E/E	Energy

Still missing (also for Calo)

Now exists!

Much improved! X%/E ⊕ Y%/√E ⊕ Z% instead of 2%/√E

Working on proper nσ implementation. In the meantime, using a crutch

- Note: Versioning seems to have fallen by the wayside for now. Online version is in a superposition of old and new…

Implementation Done

- Acceptance and resolution ex.

```
// 2%/E  + 12%/sqrtE + 2%
// A        B          C
Smear::Acceptance::Zone EmcalFwdZone(ThetaFromEta ( 3.5 ),ThetaFromEta ( 1 ),
                                     0., TMath::TwoPi(), // phi
                                     0.05, TMath::Infinity() ); // E
Smear::Device EmcalFwd(Smear::kE, "sqrt( pow ( 0.02,2 ) + pow( 0.12,2)*E + pow ( 0.02*E,2 ) )");
```

*Note: These formulas can be much more complex functions of P, θ, φ, E, …
for more concrete concepts*

- Perfect PID where 3σ is assumed. Crude but hopefully useful.

```
// p < 6 GeV, also in principle p> 100 MeV from tracker but that's not clarified yet
Smear::Acceptance::Zone PidBarrelZone(ThetaFromEta(1),ThetaFromEta(-1),
                                      0., TMath::TwoPi(), // phi
                                      0., TMath::Infinity(), // E
                                      0., 6 ); // p

Smear::PerfectID PidBarrel;
```

Tracking Preview

- Ernst has made public the [tracking part](#) and asked me officially to implement it for evaluation
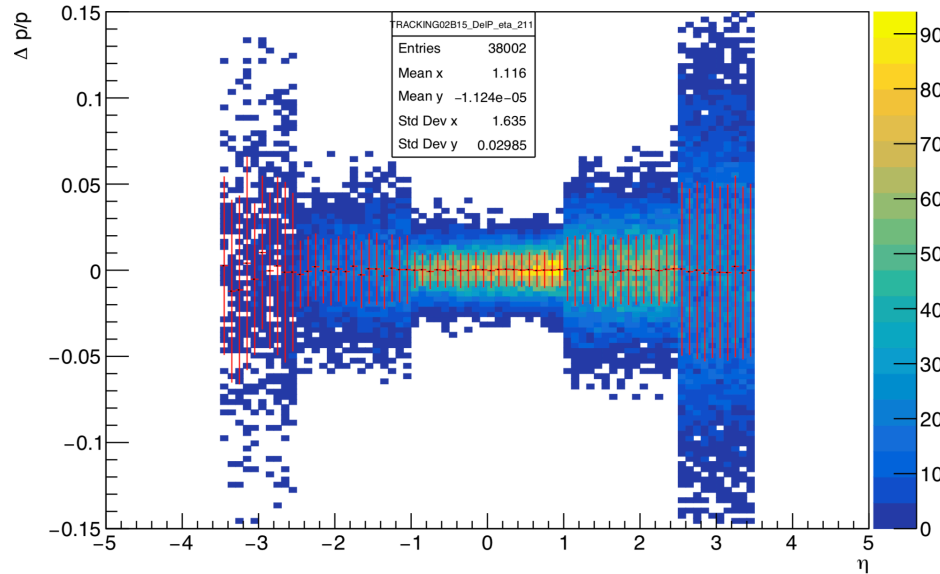
- Done. Available on RCF and github

  SmearTrackingPreview_0_2_B1_5T.cxx
  and …3T.cxx
  short names:
  TRACKING02B15
  TRACKING02B3

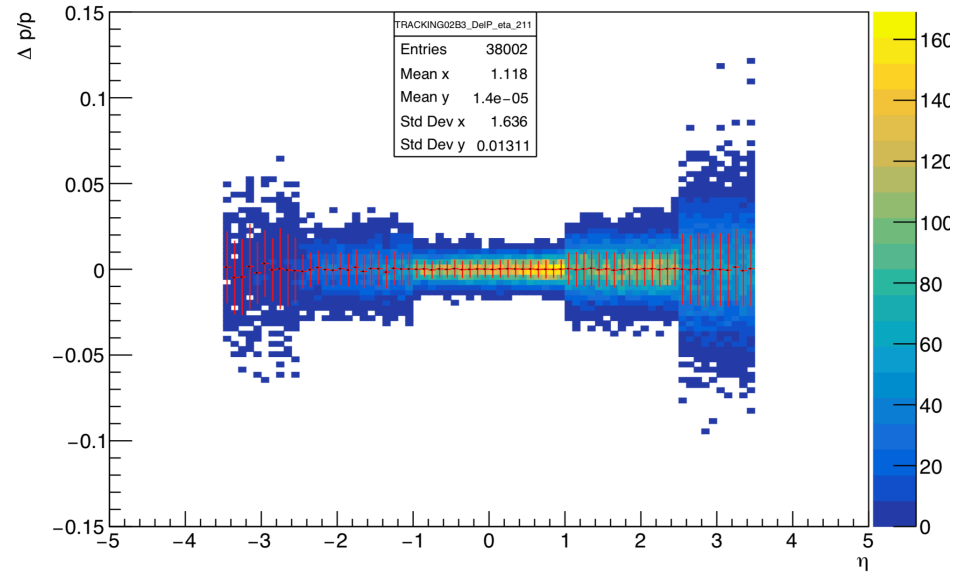|		Tracking (B = 1.5 T)			
Resolution	Relative momentum	Allowed X/X₀	minimum-pT	Trasverse pointing res.	Longitudinal pointing res.
					not accessible
					reduce
	$\sigma p/p \sim 0.2\% \times p \oplus 5\%$				
	$\sigma p/p \sim 0.04\% \times p \oplus 2\%$	~5% or less X	to be determined	dca(xy) ~ 40/pT μm ⊕ 10 μm	dca(z) ~ 100/pT μm ⊕ 20 μm
	$\sigma p/p \sim 0.04\% \times p \oplus 1\%$		100 MeV/c with 50% acceptance (similar for pi and K)	dca(xy) ~ 30/pT μm ⊕ 5 μm	dca(z) ~ 30/pT μm ⊕ 5 μm
	$\sigma p/p \sim 0.04\% \times p \oplus 2\%$		to be determined	dca(xy) ~ 40/pT μm ⊕ 10 μm	dca(z) ~ 100/pT μm ⊕ 20 μm
	$\sigma p/p \sim 0.2\% \times p \oplus 5\%$				

Charged pions

B = 1.5T
TRACKING02B15_DelP_eta_211

B = 3T
TRACKING02B3_DelP_eta_211

- Integrated over P, so dominated by constant term.
  These terms increased by factors of 2-10
  → **significantly worse** than in Matrix v0.1

Next

Before release or shortly thereafter hope to include:

- **Far Forward detectors**. Exist and can be plugged in immediately if approved

- **Low-Q² tagger**. I have histos from Jarda, but need to update for…

- **Efficiency:** Some technical issues (ask me for details), coming soon.

  Note: This will still not be sufficient information → currently ignored

minimum-pT
100 MeV/c with 50% acceptance (similar for pi and K)

- **Evaluation of tracking** has started in Jets/HF

- **DELPHES** implementation is being prepared. Hope to put a volunteer on cross-validation.

- Some more version control of the matrix…

Also next?

In various stages:

- Better/more **realistic nσ PID**. On-going discussion with PID group to incorporate 7 concepts at https://gitlab.com/preghenella/pid
  I have code and we agree more or less on the interface
  → Need their buy-in and contribution! I cannot write it for them

- **Angular resolution**
  - Calo code exists, only needs feedback and approval on a few parameters.
  - Tracking: Would be nice to have at least "1mrad or better"
  - Advice on how to improve communication is appreciated