libShower

Manual
V0.1
frank.meisel@lhep.unibe.ch

Description

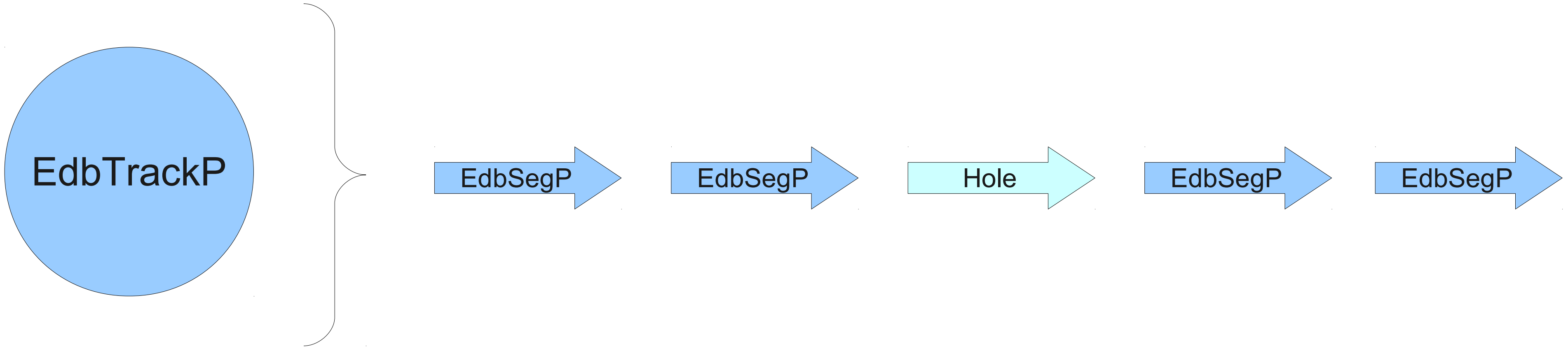
- **This manual is intended to make the user familiar on the way showers are dealt with.**
- **The user should not bother about details.**
- **To increase the performance, the user may give additional information on the date to obtain a better result.**

Overview

- Shower for Event classification.
- Shower characteristics
- Shower implementation in FEDRA
-

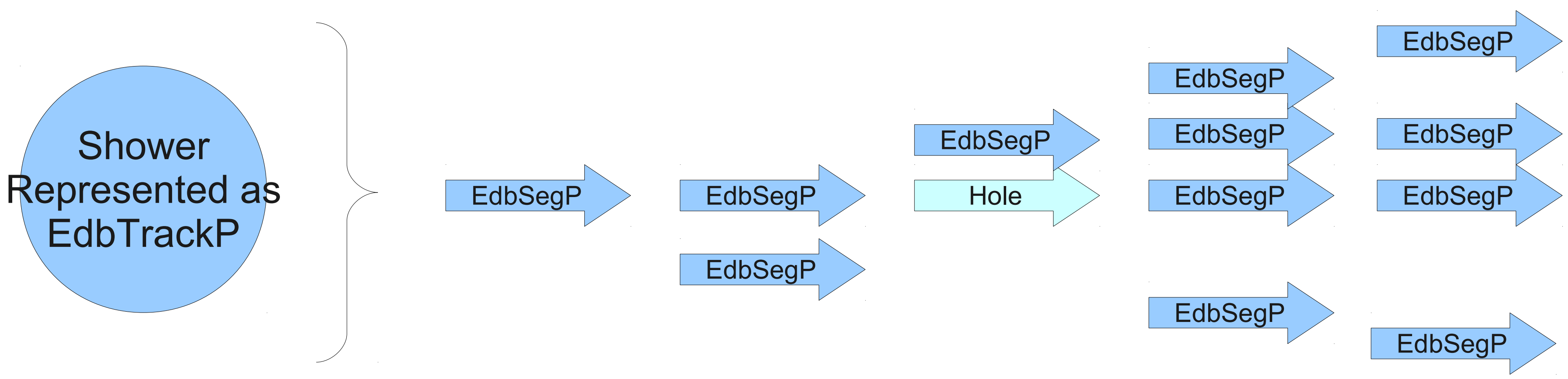
Object description

- A track in fedra
  - One Segment Per Plate
  - N(), Npl(), N0, Theta(), XYZ, P()



Object description

- A shower in fedra
  - Can have *More* than One Segment Per Plate
  - N(), Npl(), N0, Theta(), XYZ, P()
  - Plus additional information: *longitudinal&transversal* Profile!



Object description

Due to backward compabilities we have to use this way.
This is a pain in the back... :-)

- Using EdbTrackP as object class for Shower is ok, but we lose physical information.
- Compensated by infos in EdbShowerAlgE...
  - May be put later in a extra *EdbShowerP* class...
- Up to now: how is a shower represented/stored?
- We have Shower.root which uses a TTree to store the variables. Technically equivalent to do storage like EdbTrackP, but it requires complex I/O conversions to get from
  - TTree entry ↔ EdbTrackP

What is to be done for showers?

- In case we have scanback done, vertex found, scanforth done (hadr. re-interaction), we need for the full description of the event some other information: showers?
- Why are showers for the event classification important?
- Electrons: nu_e from beam; tau->e;
- Photons: present in almost any event, mainly through the decay of:
- Pi0: Find two photons, match them in case of more

Finding showers

- Showers are either attached to vertex directly or indirectly:
  - Electrons: start to shower direct from the vertex without flight length; IP to vertex rather small; DeltaZ to vertex very small.
  - Photons: fly – then showering happens; therefore IP to vertex bigger than attached tracks; DeltaZ to vertex can be large (mean flight length ca10plates).
- Finding a electron shower can be easier than photon shower (directly attached track or BT)

Properties of Showers

- We need to be sure of the following things.
- We have to find the shower (start)
- We have to reconstruct (collect all the tracks) it having as much of it reconstructed as possibles (lowest loss as possible).
- After shower(s) found and reconstructed we need the main properties:
  - Energy
  - ID
  - 1ry-2ry vtx attachment

Properties of the `Shower.root` file.
a) basic quantities

Name in `Shower.root`	Description	Analogy in EdbTrackP
number_eventb/I	MC event number	->MCEvt()
sizeb	Number of Basetracks	N()
nfilmb[]	Plate ID of BT[i]	??(to be looked up)
lengthfilmb	Number of Plates Crossed	Npl()
x,y,z -b[]	X,Y,Z Coordinates of BT[i]	->X(),Y(),Z()
tx,ty -b[]	TX,TY Coordinates of BT[i]	->TX(),TY()
plateb -b[]	??(to be looked up)	??(to be looked up)
Numbereventb[]	Numbereventb[]	Numbereventb[]
showerID/I	showerID/I	showerID/I
isizeb/I	isizeb/I	isizeb/I
ntrace1simu[sizeb]/I	ntrace1simu[sizeb]/I	ntrace1simu[sizeb]/I
ntrace2simu[sizeb]/I	ntrace2simu[sizeb]/I	ntrace2simu[sizeb]/I
ntrace3simu[sizeb]/I	ntrace3simu[sizeb]/I	ntrace3simu[sizeb]/I
Ntrace4simu[sizeb]/I	Ntrace4simu[sizeb]/I	Ntrace4simu[sizeb]/I
chi2btkb[sizeb]/F	chi2btkb[sizeb]/F	chi2btkb[sizeb]/F

Properties of the `Shower.root` file. b) derived quantities

Name in `Shower.root`	**Description**	**Analogy in EdbTrackP**
number_eventb/I	number_eventb/I	number_eventb/I
sizeb	sizeb	sizeb
nfilmb[]	nfilmb[]	nfilmb[]
lengthfilmb	lengthfilmb	lengthfilmb
x,y,z -b[]	x,y,z -b[]	x,y,z -b[]
tx,ty -b[]	tx,ty -b[]	tx,ty -b[]
plateb -b[]	plateb -b[]	plateb -b[]
Numbereventb[]	Numbereventb[]	Numbereventb[]
showerID/I	showerID/I	showerID/I
isizeb/I	isizeb/I	isizeb/I
ntrace1simu[sizeb]/I	ntrace1simu[sizeb]/I	ntrace1simu[sizeb]/I
ntrace2simu[sizeb]/I	ntrace2simu[sizeb]/I	ntrace2simu[sizeb]/I
ntrace3simu[sizeb]/I	ntrace3simu[sizeb]/I	ntrace3simu[sizeb]/I
Ntrace4simu[sizeb]/I	Ntrace4simu[sizeb]/I	Ntrace4simu[sizeb]/I
chi2btkb[sizeb]/F	chi2btkb[sizeb]/F	chi2btkb[sizeb]/F

Reconstruction: Mode A

- Choose Initiator Basetracks:

```
ShowerInstance = new EdbShowerRec();
ShowerInstance-> SetInBTArray(TObjArray* InBTArray);
```

- Set the PVR object on which the reconstruction shall operate:

```
ShowerInstance-> SetEdbPVRec(EdbPVRec* Ali);
```

- Start Reconstruction

```
ShowerInstance-> Execute();
```

- Retrieve Reconstructed Shower Array

```
ShowerInstance-> GetRecoShowerArray();
```

Reconstruction: Mode B Frederics Algo Implementation

- To be filled and explained...:
- void recdown(int num, int MAXPLATE, int DATA, int Ncand, double* x0, double* y0, double* z0, double* tx0, double* ty0, double* chi20, int* W0, double* P0, int* Flag0, int* plate0, int* id0, int* TRid, double* Esim, int piece2, int piece2par)

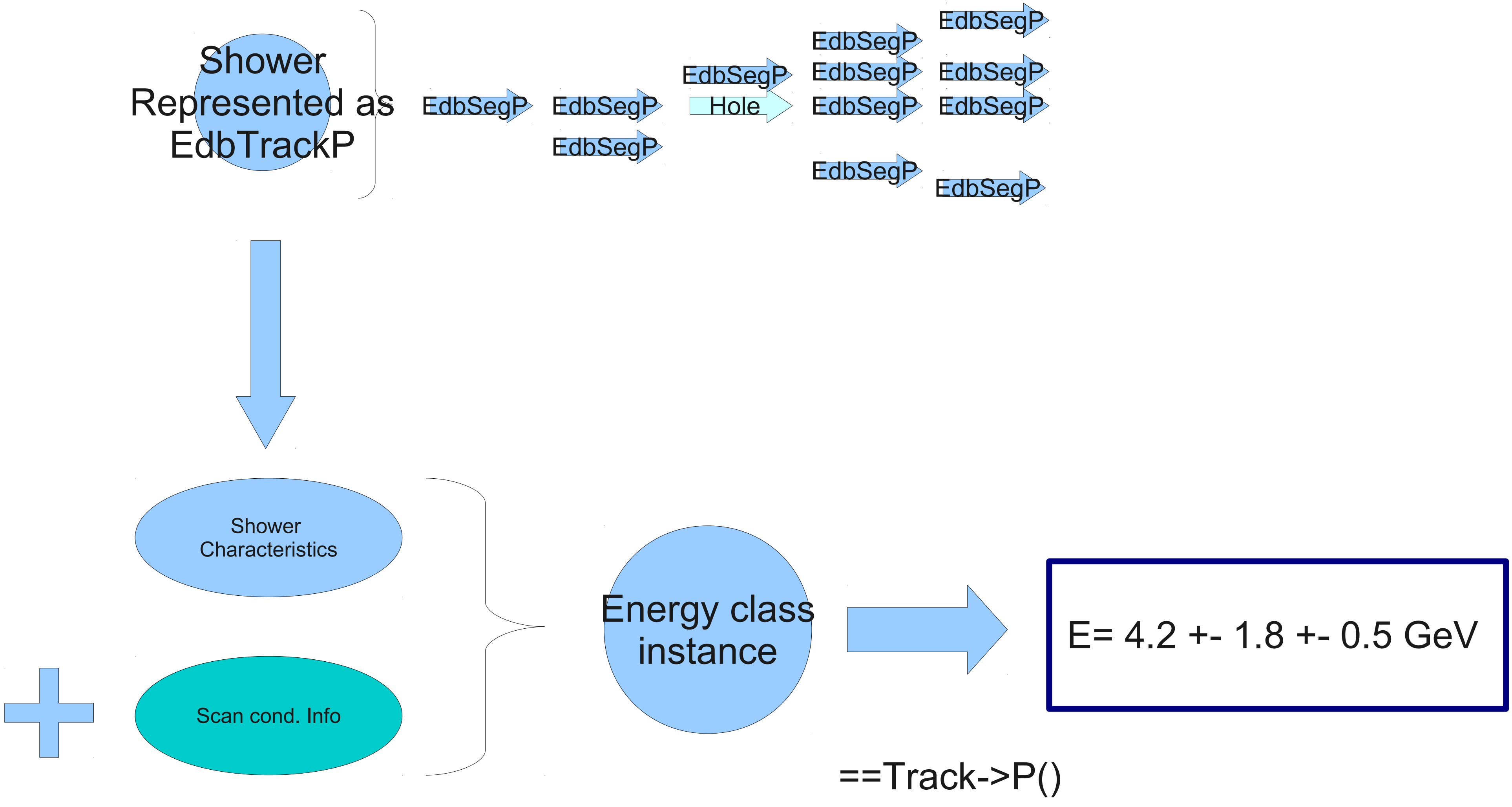
Shower Energy Measurement

- The Shower Energy is measured using:
  - Basic Properties of the Shower
  - Conditions from Brick Scanning

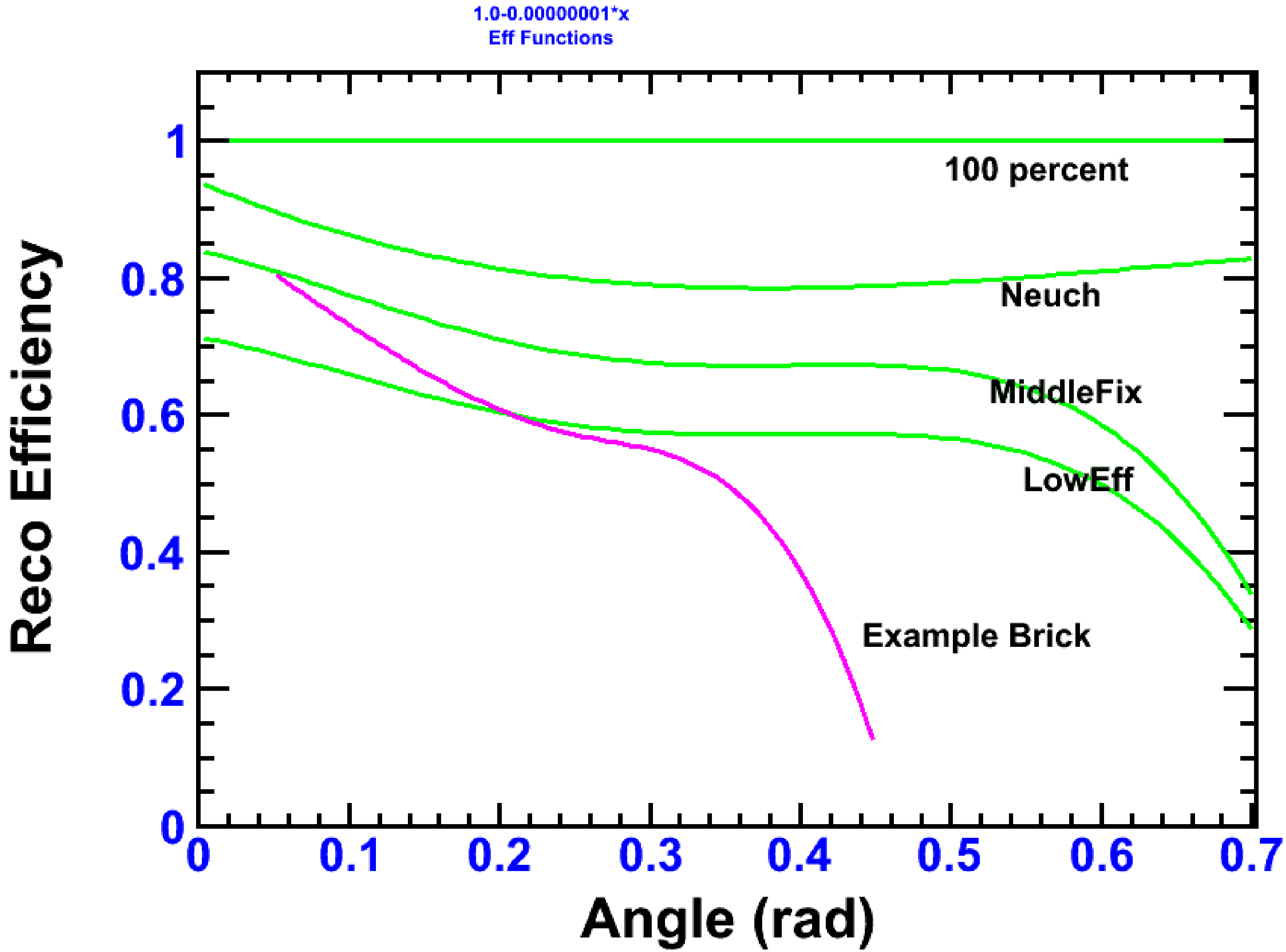
- Shower Properties
  - Shape
  - Density
  - Angle

- Scan Conditions
  - Reco Efficiency
  - BG Level

Block diagram: energy



Implemented ScanEfficiencies



Example Brick: Bern 144320, Mic4

Energy: Technical

- Get Reconstructed Shower Array
- Start Energy Estimation
- Retrieve Reconstructed Shower Array, with P() filled.

```
// Instantate ShowerAlgorithmEnergy Class
EdbShowerAlgESimple* ShowerAlgEnergyInstance = new EdbShowerAlgESimple();

// Run Shower Energy Algorithm on all shower/tracks
ShowerAlgEnergyInstance->DoRun(RecoShowerArray);

// All tracks contain now in P() the estimated energy.
```

Energy: stat. Results: sigma(E)/E gamma

Energy/NPl	0.5	0.75	1	1.5	2	3	4	6	8	16	32	64
10	0.68	0.64	0.63	0.62	0.6	0.54	0.49	0.45	0.45	0.43	0.33	0.33
12	0.61	0.56	0.54	0.54	0.52	0.47	0.44	0.4	0.4	0.41	0.32	0.38
14	0.59	0.51	0.49	0.49	0.48	0.45	0.42	0.38	0.36	0.41	0.31	0.37
16	0.59	0.51	0.47	0.42	0.4	0.38	0.36	0.32	0.32	0.37	0.29	0.31
18	0.56	0.47	0.44	0.42	0.4	0.38	0.36	0.31	0.3	0.36	0.28	0.39
20	0.7	0.55	0.48	0.39	0.36	0.34	0.32	0.29	0.27	0.32	0.26	0.37
23	0.57	0.49	0.45	0.39	0.37	0.33	0.31	0.27	0.25	0.29	0.25	0.31
26	0.64	0.52	0.46	0.4	0.35	0.3	0.29	0.25	0.22	0.25	0.22	0.25
29	0.55	0.47	0.44	0.38	0.34	0.29	0.28	0.24	0.21	0.23	0.2	0.24
32	0.58	0.49	0.44	0.38	0.34	0.3	0.27	0.23	0.21	0.21	0.19	0.23
35	0.59	0.49	0.44	0.38	0.35	0.3	0.27	0.23	0.2	0.2	0.18	0.21
40	0.58	0.48	0.44	0.39	0.35	0.29	0.27	0.23	0.2	0.18	0.17	0.21
45	0.58	0.48	0.45	0.39	0.35	0.29	0.27	0.23	0.2	0.18	0.16	0.19

Energy: stat. Results: sigma(E)/E ele

Energy/NPl	0.5	0.75	1	1.5	2	3	4	6	8	16	32	64
10	0.94	0.92	0.91	0.86	0.78	0.68	0.62	0.55	0.52	0.45	0.34	0.25
12	0.74	0.73	0.72	0.69	0.65	0.58	0.53	0.49	0.47	0.44	0.34	0.27
14	0.72	0.64	0.62	0.6	0.58	0.52	0.48	0.44	0.44	0.43	0.34	0.29
16	0.67	0.6	0.55	0.5	0.49	0.45	0.42	0.39	0.39	0.4	0.32	0.32
18	0.61	0.55	0.51	0.47	0.46	0.42	0.4	0.37	0.36	0.4	0.31	0.36
20	0.56	0.52	0.48	0.43	0.4	0.39	0.38	0.34	0.33	0.37	0.29	0.27
23	0.57	0.51	0.46	0.41	0.37	0.35	0.34	0.31	0.29	0.33	0.27	0.36
26	0.6	0.52	0.46	0.39	0.35	0.33	0.32	0.29	0.26	0.29	0.25	0.31
29	0.63	0.54	0.47	0.38	0.34	0.31	0.3	0.27	0.25	0.27	0.23	0.28
32	0.64	0.54	0.47	0.39	0.34	0.3	0.29	0.26	0.24	0.25	0.21	0.28
35	0.65	0.54	0.47	0.39	0.34	0.3	0.28	0.26	0.23	0.23	0.2	0.26
40	0.62	0.53	0.46	0.38	0.34	0.3	0.28	0.26	0.23	0.21	0.19	0.22
45	0.62	0.53	0.46	0.38	0.34	0.3	0.28	0.25	0.23	0.21	0.18	0.23

Interpolation

- Interpolate energy statistic uncertanties with ROOT Tspline3 class (between energies)
- (No interpolation done between plate binnings.)

More here : SYSTEMATICS

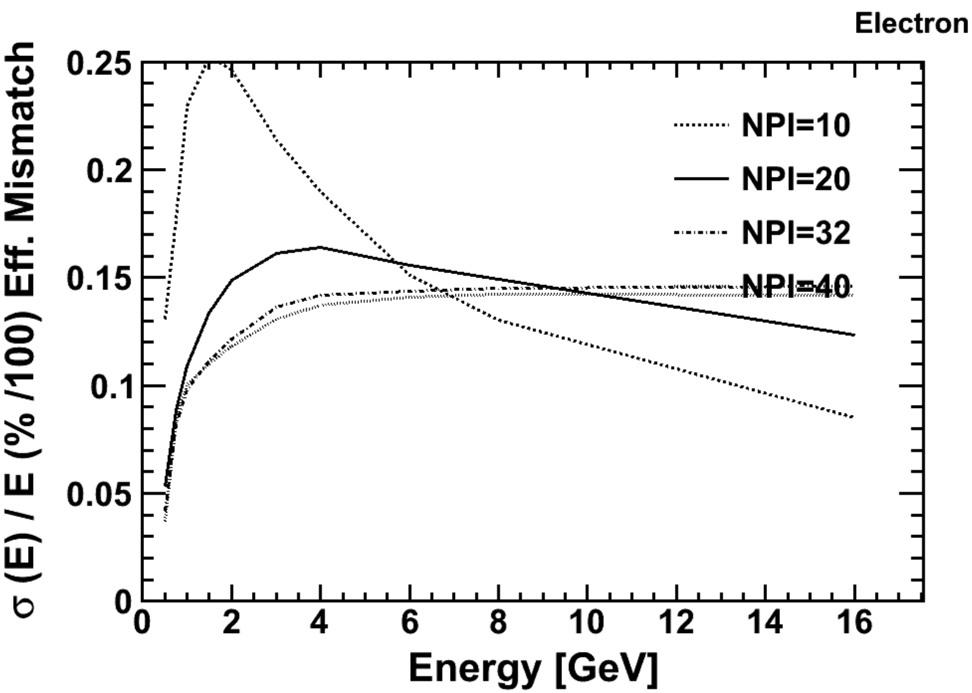
- Calculating energy gives you stat. AND sys. On the screen. Saved is then the quadratic sum.
  - (Attention: at the moment (svn1056) in fedra there is no additional variable to store momentum/energy error of a track/shower. Uncertainty is now ***only*** written in shower.root file.)
- Different sources have been investigated.
- See the following table for an overview.
  - (Details will be written up in the thesis, also many plots and tables; *explanations will follow here*)

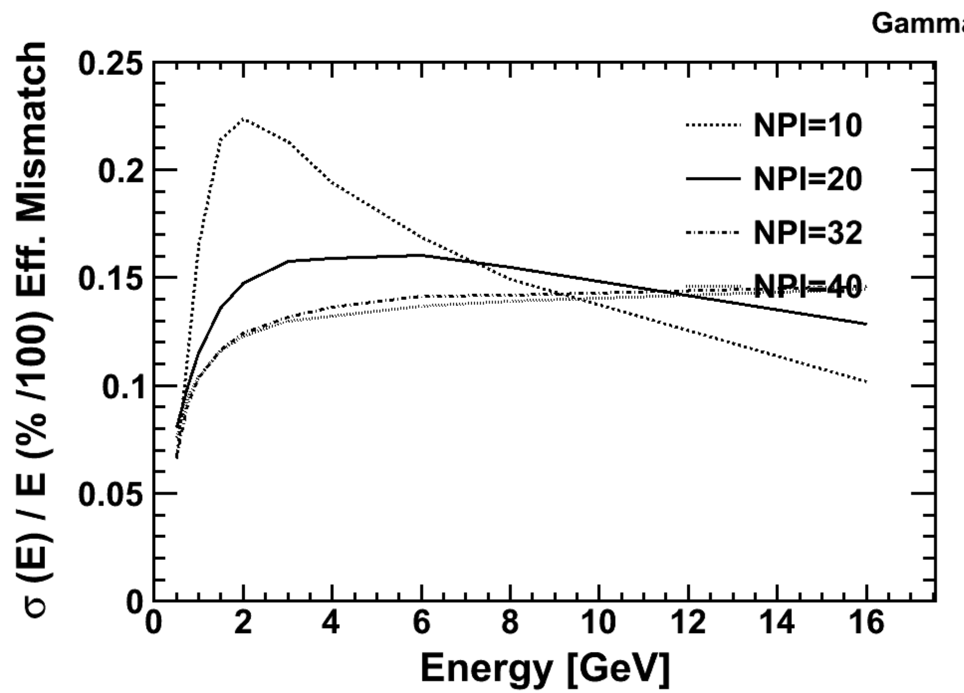
	DEPENDS ON ENERGY ??	DEPENDS ON NPLATES ??	E/G IMPACT ??	PRIORITY	Sample Resolution/E gamma @ 1GeV@20Pl	Sample Resolution/E Electron @ 4GeV@20Pl
FROM THE SCANNING METHOD						
EFFICIENCY MISMATCH	YES	YES	NO	HIGH	0.12	0.16
BG LEVEL	YES	YES	YES	HIGH	?	?
NPL, Before,After match	YES	YES	NO	MIDDLE	0.026	0.081
DISALIGNMENT	YES	YES	NO	(SKIP)	-	-
FROM THE RECO METHOD						
E/G MISMATCH	YES	YES	NO	LOW	0.04	0.03
Shower Alg						
FROM THE Multivariate METHOD						
N TRAININGS CYLCES	YES	YES	NO	LOW	-	-
N TRAININGS EVENTS	YES	YES	NO	LOW	-	-
TYPE TRAININGS FILE	YES	YES	NO	MIDDLE	-	-
Cycles&Events(File fixed)	YES	YES	NO	LOW	0.012	0.0068
FROM THE MONTE CARLO METHOD						
QUALITY CUT	-	-	-	LOW	?	?
SHOWER PARAMETRISATION	-	-	-	(SKIP)	-	-
ESTIMATED SUM SYST. (+linear, i.e. Overestimated a bit)					<0.33	<0.4
STATISTICAL					0.48	0.38

E: Systematics

- Show uncertainties from different sources.
- Electrons left, Photons right.
- (Tables with full valueset will be put later)
-
- Take Care of different Y-axis scaling!

E: Systematics: Eff mismatch

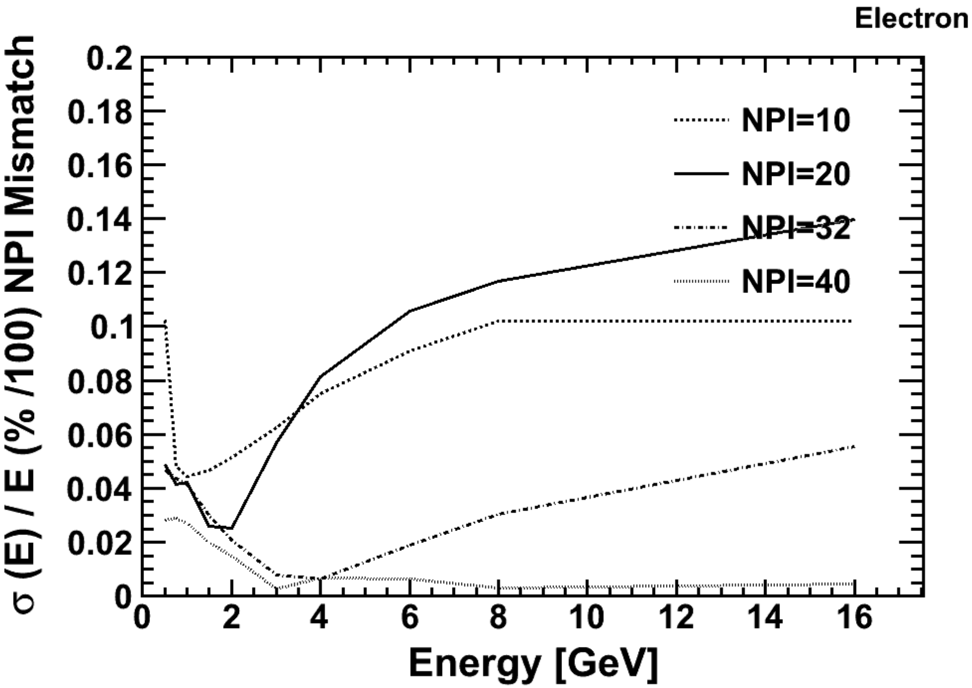


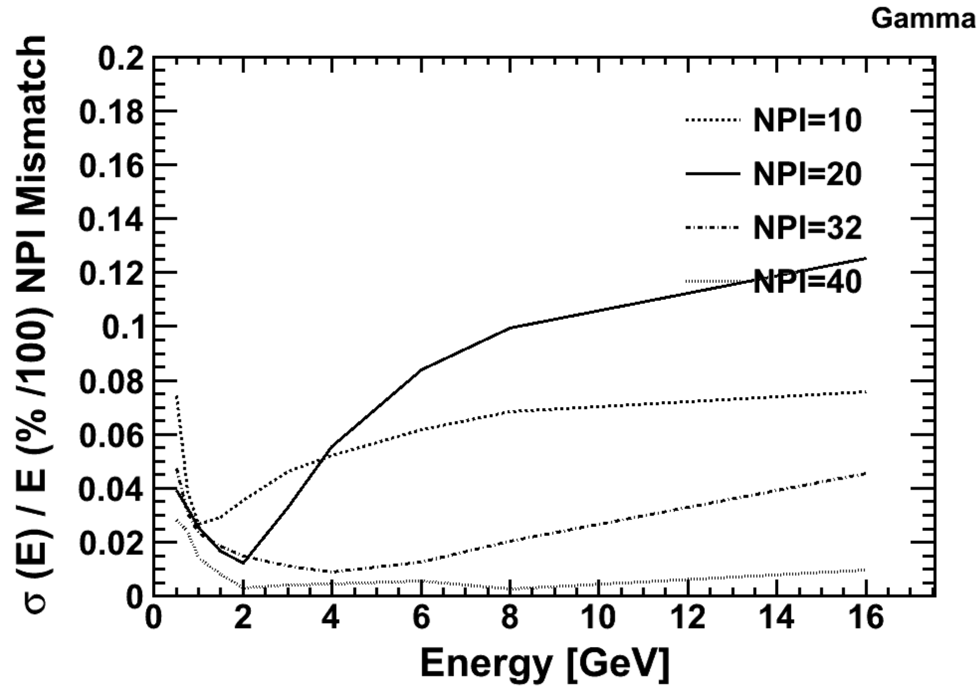


Energy/NPl	0.5	0.75	1	1.5	2	3	4	6	8	16	32	64
10	0.13	0.18	0.23	0.25	0.25	0.21	0.19	0.15	0.13	0.085	0.051	0.028
12	0.099	0.13	0.18	0.21	0.22	0.2	0.18	0.15	0.14	0.096	0.059	0.032
14	0.054	0.11	0.15	0.18	0.19	0.18	0.17	0.15	0.14	0.098	0.054	0.025
16	0.052	0.099	0.13	0.16	0.18	0.18	0.17	0.16	0.14	0.11	0.059	0.027
18	0.05	0.088	0.11	0.14	0.16	0.17	0.17	0.16	0.15	0.11	0.059	0.025
20	0.053	0.088	0.11	0.13	0.15	0.16	0.16	0.16	0.15	0.12	0.062	0.022
23	0.053	0.084	0.1	0.12	0.13	0.15	0.16	0.15	0.15	0.13	0.07	0.029
26	0.04	0.079	0.1	0.12	0.13	0.14	0.15	0.15	0.15	0.14	0.058	0.017
29	0.053	0.088	0.11	0.12	0.12	0.14	0.14	0.15	0.15	0.14	0.064	0.018
32	0.042	0.082	0.098	0.11	0.12	0.14	0.14	0.14	0.15	0.15	0.068	0.029
35	0.036	0.083	0.1	0.11	0.12	0.13	0.14	0.14	0.14	0.14	0.068	0.023
40	0.037	0.084	0.1	0.11	0.12	0.13	0.14	0.14	0.14	0.14	0.072	0.02
45	0.022	0.068	0.091	0.11	0.12	0.13	0.14	0.14	0.14	0.14	0.067	0.023

Energy/NPl	0.5	0.75	1	1.5	2	3	4	6	8	16	32	64
10	0.066	0.11	0.16	0.21	0.22	0.21	0.19	0.17	0.15	0.1	0.061	0.03
12	0.063	0.11	0.15	0.18	0.19	0.19	0.18	0.16	0.15	0.11	0.069	0.036
14	0.071	0.099	0.13	0.17	0.18	0.18	0.18	0.17	0.15	0.11	0.066	0.033
16	0.064	0.1	0.13	0.15	0.17	0.17	0.17	0.16	0.15	0.12	0.068	0.028
18	0.059	0.093	0.11	0.14	0.15	0.16	0.16	0.16	0.16	0.12	0.071	0.029
20	0.081	0.099	0.12	0.14	0.15	0.16	0.16	0.16	0.15	0.13	0.076	0.036
23	0.071	0.092	0.11	0.12	0.13	0.15	0.15	0.15	0.15	0.13	0.073	0.031
26	0.079	0.099	0.11	0.12	0.13	0.14	0.14	0.15	0.15	0.14	0.063	0.022
29	0.089	0.1	0.11	0.12	0.13	0.14	0.14	0.14	0.14	0.14	0.064	0.017
32	0.067	0.091	0.1	0.12	0.12	0.13	0.14	0.14	0.14	0.15	0.069	0.023
35	0.079	0.091	0.1	0.12	0.13	0.13	0.13	0.14	0.14	0.14	0.067	0.021
40	0.076	0.094	0.1	0.12	0.12	0.13	0.13	0.14	0.14	0.14	0.059	0.026
45	0.072	0.096	0.11	0.12	0.13	0.13	0.13	0.14	0.14	0.14	0.057	0.011

E: Systematics: NPL mismatch



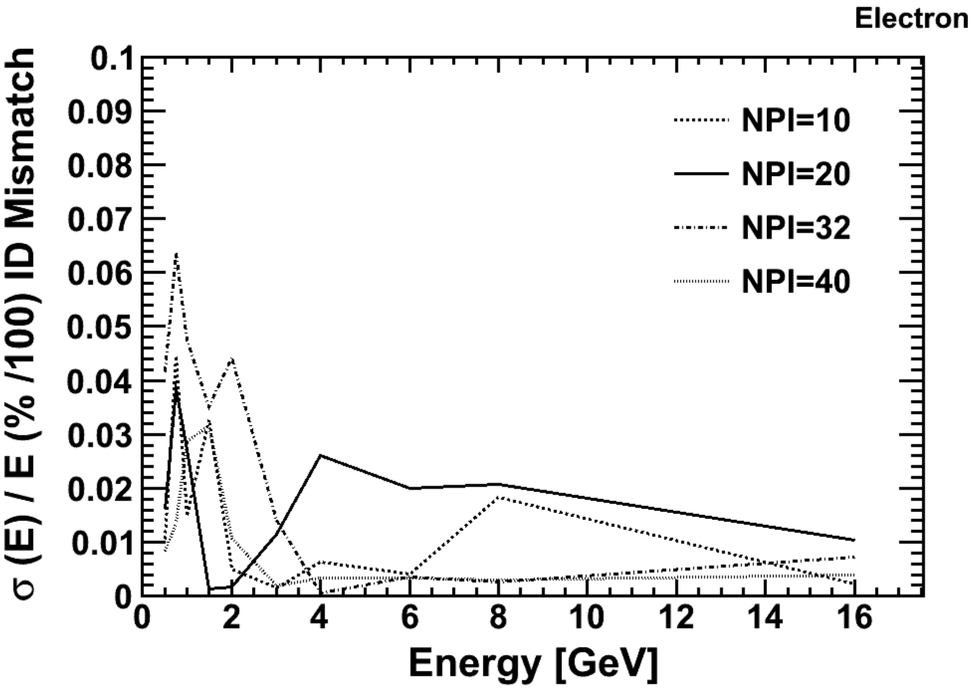


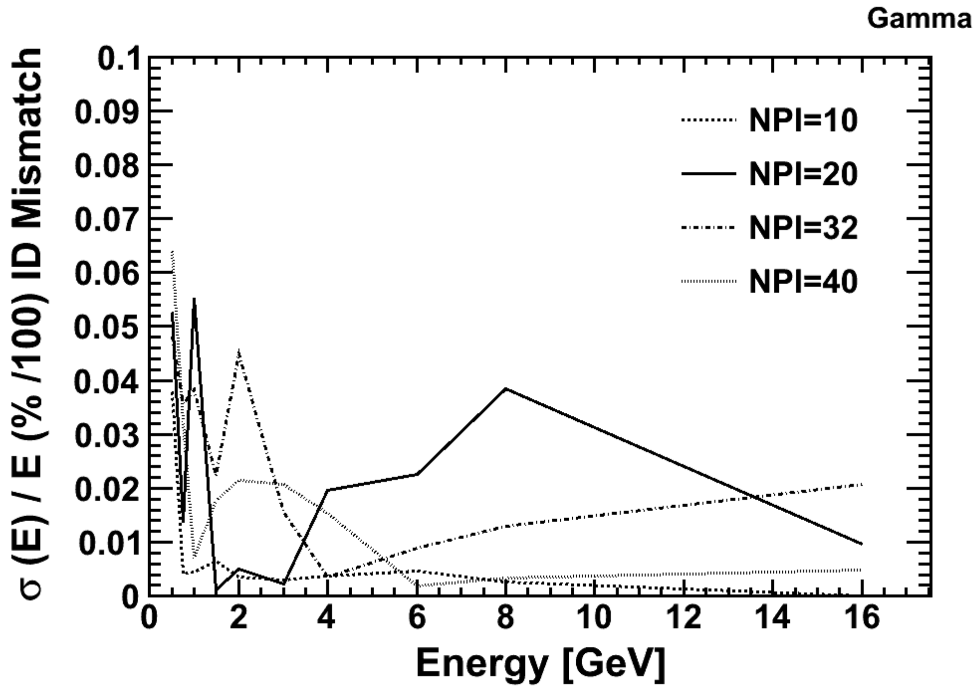
Energy/NPl	0.5	0.75	1	1.5	2	3	4	6	8	16
10	0.1	0.053	0.055	0.067	0.084	0.11	0.14	0.19	0.21	0.18
12	0.067	0.032	0.063	0.11	0.15	0.18	0.21	0.24	0.26	0.24
14	0.049	0.033	0.044	0.087	0.12	0.16	0.19	0.23	0.25	0.24
16	0.06	0.036	0.033	0.052	0.085	0.13	0.15	0.17	0.19	0.2
18	0.054	0.036	0.027	0.027	0.056	0.098	0.12	0.15	0.17	0.2
20	0.058	0.046	0.044	0.029	0.035	0.073	0.1	0.14	0.15	0.18
23	0.054	0.048	0.043	0.025	0.014	0.034	0.057	0.087	0.1	0.15
26	0.079	0.069	0.062	0.042	0.024	0.015	0.031	0.06	0.077	0.12
29	0.044	0.049	0.047	0.031	0.019	0.0051	0.015	0.041	0.058	0.098
32	0.051	0.046	0.043	0.031	0.022	0.0082	0.0073	0.023	0.037	0.069
35	0.12	0.095	0.077	0.05	0.033	0.017	0.0089	0.013	0.023	0.052
40	0.035	0.032	0.03	0.023	0.018	0.0085	0.0083	0.0064	0.0066	0.021
45	0.1	0.08	0.065	0.045	0.031	0.02	0.014	0.0088	0.0064	0.0095

l@lhe

Energy/NPl	0.5	0.75	1	1.5	2	3	4	6	8	16
10	0.074	0.04	0.03	0.042	0.056	0.078	0.092	0.12	0.13	0.15
12	0.038	0.019	0.024	0.065	0.098	0.14	0.17	0.2	0.22	0.23
14	0.042	0.032	0.02	0.039	0.069	0.12	0.15	0.19	0.22	0.23
16	0.031	0.022	0.015	0.022	0.045	0.09	0.12	0.15	0.17	0.19
18	0.044	0.035	0.022	0.011	0.024	0.062	0.089	0.12	0.14	0.17
20	0.045	0.038	0.03	0.018	0.014	0.042	0.07	0.11	0.13	0.16
23	0.062	0.038	0.026	0.016	0.012	0.019	0.035	0.062	0.08	0.13
26	0.052	0.039	0.025	0.017	0.014	0.0096	0.018	0.041	0.058	0.1
29	0.085	0.051	0.037	0.028	0.022	0.012	0.01	0.026	0.04	0.081
32	0.049	0.032	0.025	0.02	0.016	0.011	0.0091	0.015	0.025	0.054
35	0.082	0.059	0.045	0.031	0.022	0.014	0.0091	0.0074	0.016	0.04
40	0.066	0.051	0.04	0.027	0.02	0.012	0.0073	0.0056	0.0042	0.018
45	0.18	0.13	0.1	0.066	0.049	0.031	0.021	0.01	0.0045	0.0094

E: Systematics: ID mismatch

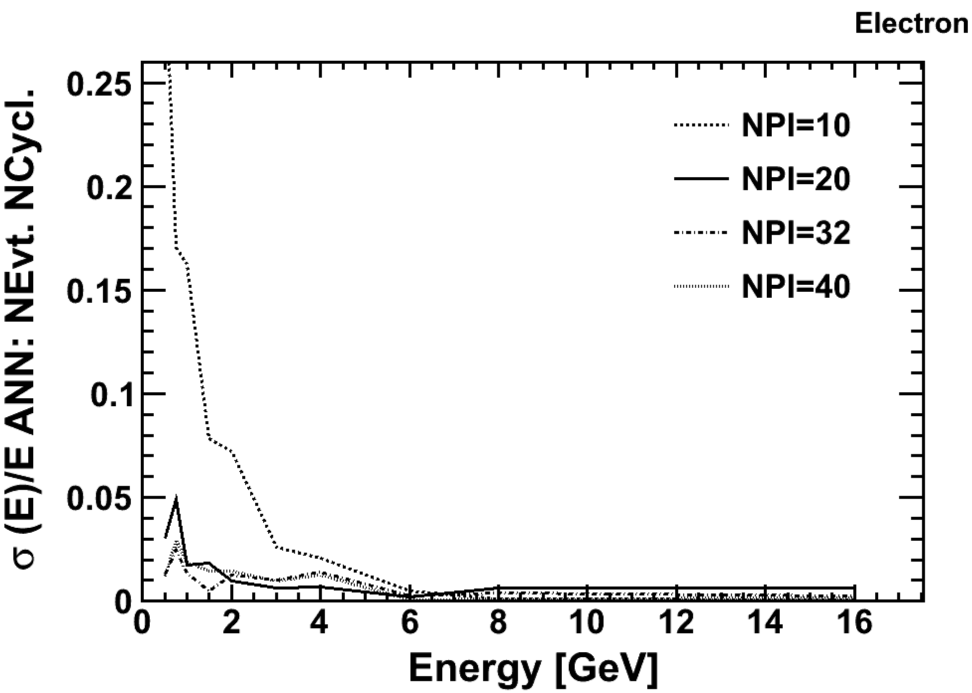


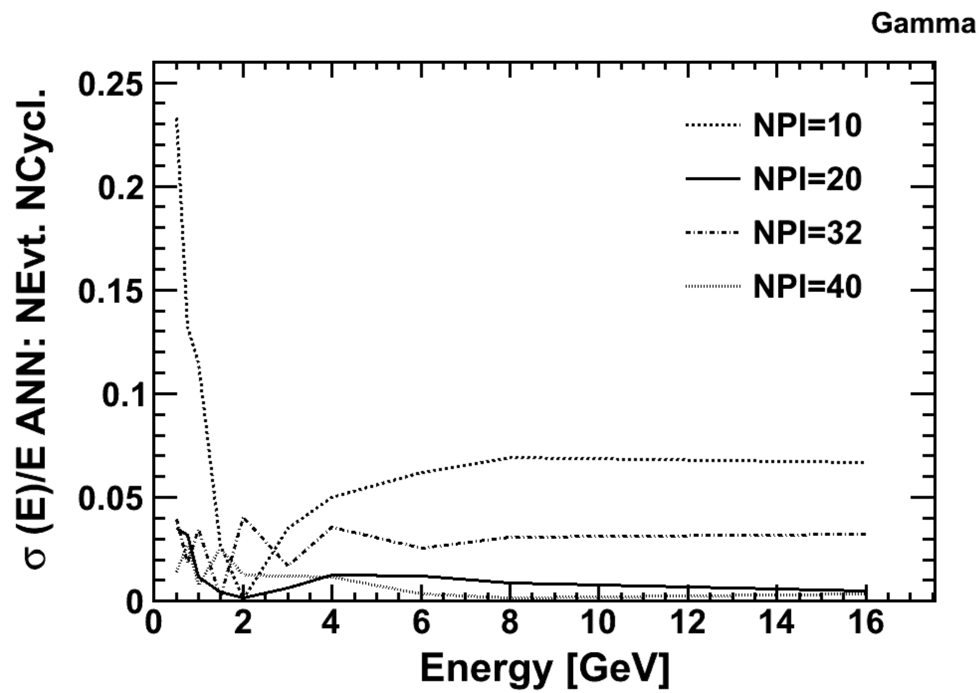


Energy/NPl	0.5	0.75	1	1.5	2	3	4	6	8	16
10	0.011	0.045	0.015	0.033	0.0049	0.0016	0.0063	0.0041	0.018	0.0022
12	0.048	0.022	0.011	0.051	0.086	0.091	0.089	0.011	0.073	0.05
14	0.13	0.032	0.008	0.029	0.061	0.079	0.08	0.044	0.067	0.034
16	0.042	0.043	0.012	0.0012	0.023	0.04	0.043	0.031	0.012	0.023
18	0.048	0.061	0.041	0.005	0.0051	0.018	0.034	0.045	0.0024	0.0073
20	0.016	0.038	0.027	0.0013	0.0017	0.012	0.026	0.02	0.021	0.01
23	0.058	0.0043	0.035	0.0052	0.017	0.0039	0.0071	0.021	0.016	0.029
26	0.025	0.062	0.055	0.045	0.037	0.0016	0.0016	0.015	0.018	0.024
29	0.043	0.043	0.022	0.014	0.028	0.004	0.00087	0.0032	0.0033	0.003
32	0.042	0.064	0.047	0.035	0.044	0.014	0.00063	0.0036	0.0027	0.0073
35	0.03	0.02	0.014	0.033	0.025	0.0099	0.0055	0.0054	0.00013	0.011
40	0.0083	0.014	0.029	0.032	0.011	0.0019	0.0034	0.0033	0.003	0.0039
45	0.059	0.039	0.035	0.034	0.038	0.022	0.0045	0.0037	0.0055	0.0048

Energy/NPl	0.5	0.75	1	1.5	2	3	4	6	8	16
10	0.038	0.0041	0.0045	0.0066	0.0036	0.003	0.0037	0.0047	0.0026	5.5e-05
12	0.0052	0.034	0.011	0.077	0.092	0.11	0.093	0.019	0.088	0.062
14	0.034	0.029	0.032	0.019	0.049	0.076	0.084	0.037	0.066	0.044
16	0.062	0.053	0.0023	2.7e-05	0.02	0.021	0.035	0.039	0.016	0.031
18	0.046	0.056	0.05	0.0046	0.013	0.02	0.038	0.035	0.0086	0.0043
20	0.053	0.014	0.055	0.0012	0.0051	0.0022	0.02	0.023	0.038	0.0096
23	0.044	0.043	0.025	0.0066	0.023	0.0031	0.0055	0.021	0.024	0.028
26	0.055	0.076	0.058	0.061	0.039	0.004	0.0036	0.0066	0.024	0.026
29	0.033	0.032	0.029	0.017	0.018	0.003	0.0027	0.00024	0.0086	0.0096
32	0.048	0.036	0.038	0.022	0.045	0.016	0.0035	0.0088	0.013	0.021
35	0.03	0.027	0.025	0.022	0.037	0.01	0.012	5.2e-05	0.0091	0.013
40	0.064	0.032	0.007	0.018	0.021	0.021	0.015	0.0019	0.0034	0.0049
45	0.035	0.044	0.029	0.042	0.035	0.025	0.0025	0.0032	0.0054	0.0032

E: Systematics: ANN uncertainty





Energy/NPl	0.5	0.75	1	1.5	2	3	4	6	8	16
10	0.32	0.17	0.16	0.078	0.072	0.026	0.021	0.0051	0.00096	0.00072
12	0.1	0.005	0.05	0.012	0.03	0.017	0.021	0.013	0.014	0.012
14	0.0032	0.018	0.0067	0.021	0.019	0.016	0.018	0.0082	0.0074	0.0053
16	0.055	0.036	0.026	0.012	0.022	0.012	0.031	0.024	0.025	0.027
18	0.013	0.037	0.0082	0.011	0.0098	0.0019	0.0092	0.00087	0.0019	0.015
20	0.03	0.049	0.018	0.018	0.0095	0.0065	0.0068	0.002	0.0063	0.0063
23	0.013	0.0092	0.0087	0.0052	0.011	0.0017	0.015	0.0072	0.0087	0.0039
26	0.0082	0.014	0.0074	0.0026	0.0072	0.0029	0.013	0.001	0.00095	0.0072
29	0.012	0.026	0.015	0.011	0.0073	0.0078	0.013	0.0017	0.0022	0.0081
32	0.013	0.025	0.014	0.0051	0.012	0.01	0.014	0.0024	0.0039	0.0024
35	0.0049	0.057	0.0087	0.012	0.0018	0.0049	0.0053	0.00089	0.00055	0.00029
40	0.012	0.029	0.019	0.014	0.015	0.0099	0.012	0.00064	0.0011	0.0014
45	0.0015	0.042	0.013	0.015	0.011	0.01	0.009	8.9e-05	0.00088	0.00085

Energy/NPl	0.5	0.75	1	1.5	2	3	4	6	8	16
10	0.23	0.13	0.11	0.027	0.0012	0.035	0.05	0.062	0.069	0.067
12	0.19	0.092	0.11	0.076	0.11	0.087	0.1	0.083	0.079	0.064
14	0.099	0.39	0.048	0.37	0.091	0.16	0.12	0.077	0.061	0.065
16	0.12	0.039	0.036	0.011	0.038	0.038	0.057	0.051	0.055	0.059
18	0.077	0.06	0.022	0.021	0.041	0.069	0.097	0.098	0.096	0.097
20	0.035	0.032	0.012	0.0046	0.0013	0.0063	0.013	0.012	0.0087	0.0047
23	0.012	0.32	0.03	0.26	0.017	0.18	0.01	0.12	0.12	0.12
26	0.044	0.023	0.013	0.018	0.011	0.0057	0.029	0.032	0.035	0.042
29	0.029	0.013	0.013	0.0012	0.015	0.0057	0.021	0.014	0.013	0.021
32	0.039	0.018	0.035	0.0021	0.04	0.017	0.035	0.026	0.031	0.032
35	0.022	0.018	0.008	0.02	0.019	0.009	0.013	0.001	0.00036	0.001
40	0.014	0.027	0.0076	0.025	0.013	0.012	0.012	0.0033	0.0016	0.0033
45	0.023	0.023	0.0065	0.0091	0.015	0.00047	0.013	0.0014	0.0049	0.0038

E: Systematics: Addition

- As we know, systematic sources are neither totally independent nor totally dependent...
- But to derive an analytical form for the covariance elements is even more impossible in experimental cases (no analytical function to describe the dependencies)
- To calculate ALL variations by means of MC estimates would require tooo much computing power.
- So tradeoff here: take the sources listed in table and add them
  - *Q*uadratically: sigma^sys_tot= *Q*SUM(sigma^sys_i)
  - *L*inear: sigma^sys_tot= *L*SUM(sigma^sys_i)
- „True“ error will be in between....
- The more systematic sources considered, the larger the error we get.
- We think that the most important we have investigated...

Shower ID

- Problem:
  - One single Shower can originate from either photons, electrons or ($_{\text{charged}}$) pions

- Solution:
  - Shower ID class, which does a differentiation!

- Technicalities:
  - The ID class determines Shower from its originating particle

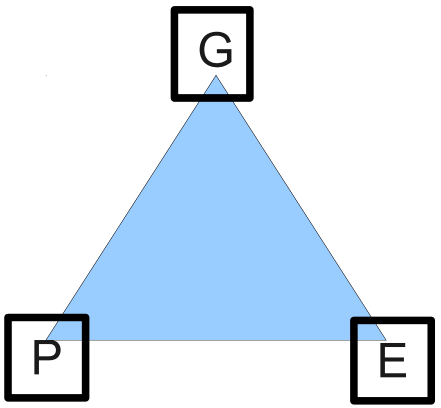
Shower ID

- Instead of assigning the ANN one number for
  - Gamma, Electron, Pion(+-)
- We do decions based on „X vs Y“, i.e.
  - Gamma vs Electron
  - Electron vs Pion
- Because the ANN is better in discriminating two-by-two variable discrimination.
- In continuation with former implementations, we save these results in eProb1,eProb90_X_vs_Y

Eprob1: bg rejection at 1%

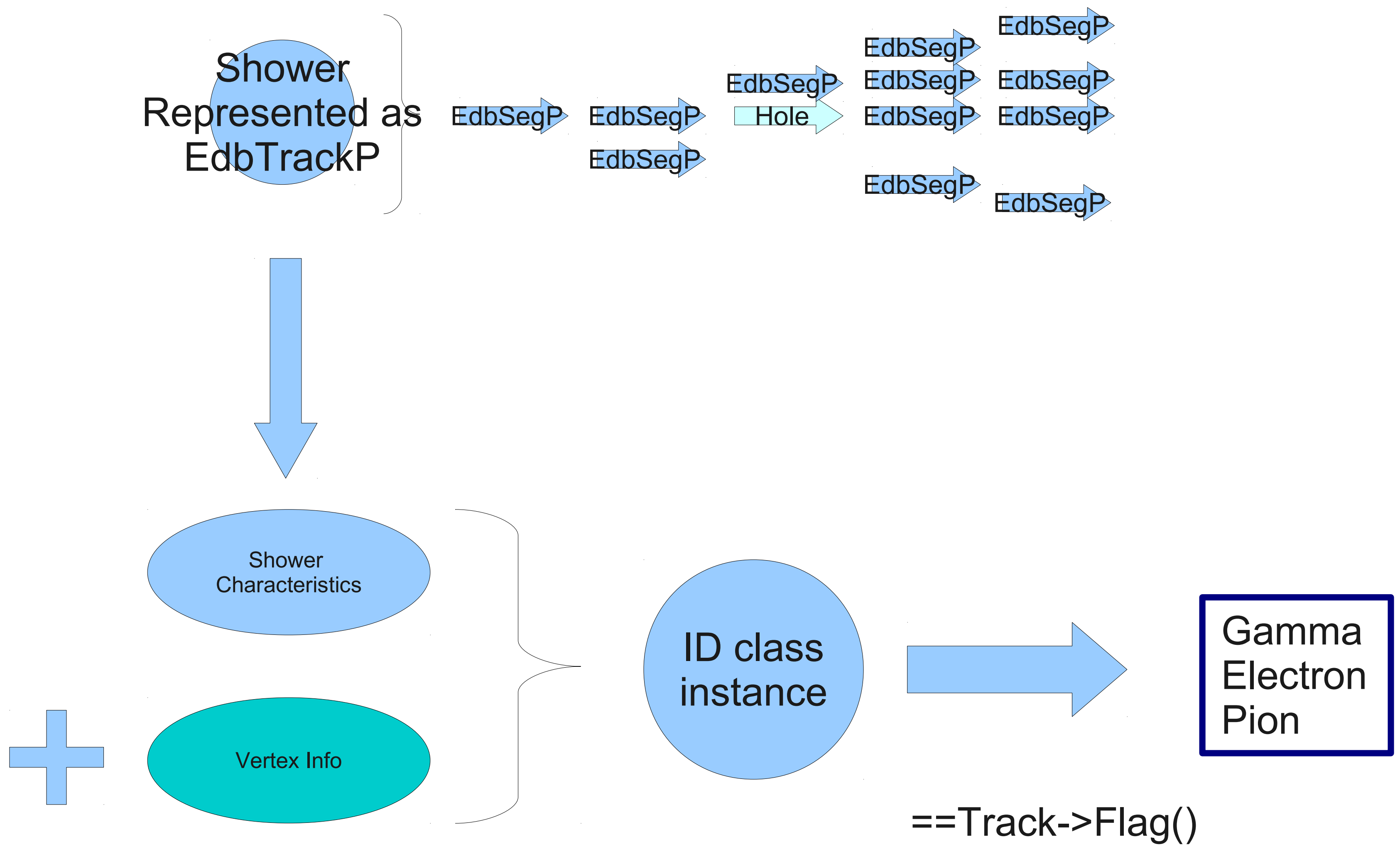
Eprob90: signal efficiency 90%

Shower ID



- Two type of variables:
  - SHID: 0(G),1(E),2(P)
  - SPID: 0,1,2
  - So we can assign a unique
  - ID for each combination:
  - Example:
    - gamma_vs_e: SHID=0,SPID=1
    - e_vs_pi: SHID=1,SPID=2
  - Always put 0 if it belongs to Type 0 (SHID)
  - Always put 1 if it belongs to Type 1 (SPID)

Block diagram: ID



More here...

The Cleaning of High BG level...

- We try to keep a constant BT density level of not more than 40BT/mm2
- To get this, we have the EdbPVRec patterns object, which we use for checks:
- We calculate BT density plate by plate and can adapt the Quality Cut plate by plate accordingly.

EdbPVRQuality

- This class does BG cleaning... How it works:
- EdbPVRQuality* Qualityobject = new EdbPVRQuality(gAli);
  - gAli is the filled EdbPVRec object. Usually this implicates the data coming from cp.root files.
  - LinkedTracks.root EdbPVRec object usually is already very „clean“, i.e. that linked tracks are often „good“ (physically) objects.
- Qualityobject->Print();
- Qualityobject->Help();