Calibration of DCV(Run 81)

HongMin KIM

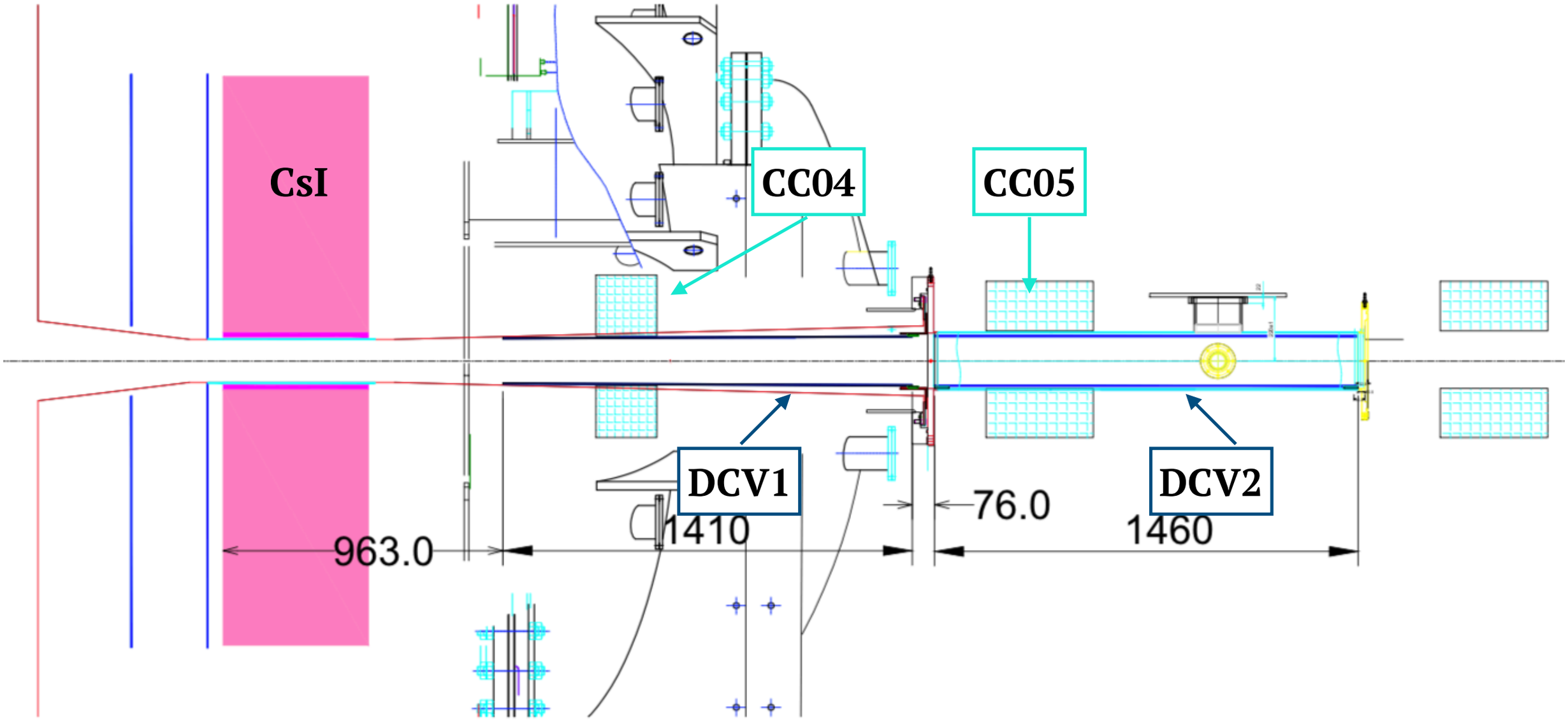
/home/had/hmkim/work/hmkim/run81/final_cal

Run Information

run number	Start time	Stop time	Run Type	Note	L1 Raw	L1 Live	L1 Accepted	L1 Accept rate	L2 Accepted	SemiOnline Plots
30776	2019-03-01 01:30:52	2019-03-01 02:22:09	COSMIC	Cosmic_PS_CSI_PartOfVetos	0	0	0	0.00	4046983	link
30777	2019-03-01 02:23:45	2019-03-01 03:15:06	COSMIC	Cosmic_PS_CSI_PartOfVetos	0	0	0	0.00	4464411	link
30778	2019-03-01 03:16:01	2019-03-01 04:06:19	COSMIC	Cosmic_PS_CSI_PartOfVetos	0	0	0	0.00	4627232	link
30779	2019-03-01 04:07:14	2019-03-01 04:58:45	COSMIC	Cosmic_PS_CSI_PartOfVetos	0	0	0	0.00	4614097	link
30780	2019-03-01 04:59:39	2019-03-01 05:49:59	COSMIC	Cosmic_PS_CSI_PartOfVetos	0	0	0	0.00	4619830	link
30781	2019-03-01 05:50:55	2019-03-01 06:41:13	COSMIC	Cosmic_PS_CSI_PartOfVetos	0	0	0	0.00	4619750	link
30782	2019-03-01 06:42:06	2019-03-01 07:32:45	COSMIC	Cosmic_PS_CSI_PartOfVetos	0	0	0	0.00	334772	link
30783	2019-03-01 07:33:39	2019-03-01 07:44:16	COSMIC	Cosmic_PS_CSI_PartOfVetos	0	0	0	0.00	64274	link
30784	2019-03-01 07:45:34	2019-03-01 07:56:45	COSMIC	Cosmic_PS_CSI_PartOfVetos	0	0	0	0.00	0	link
30785	2019-03-01 07:57:49	2019-03-01 08:08:58	COSMIC	Cosmic_PS_CSI_PartOfVetos	0	0	0	0.00	990560	link
30786	2019-03-01 08:10:00	2019-03-01 09:00:48	COSMIC	Cosmic_PS_CSI_PartOfVetos	0	0	0	0.00	4828488	link
30787	2019-03-01 09:01:42	2019-03-01 09:10:48	COSMIC	Cosmic_PS_CSI_PartOfVetos	0	0	0	0.00	859356	link
Total				0	0	0		34069753		

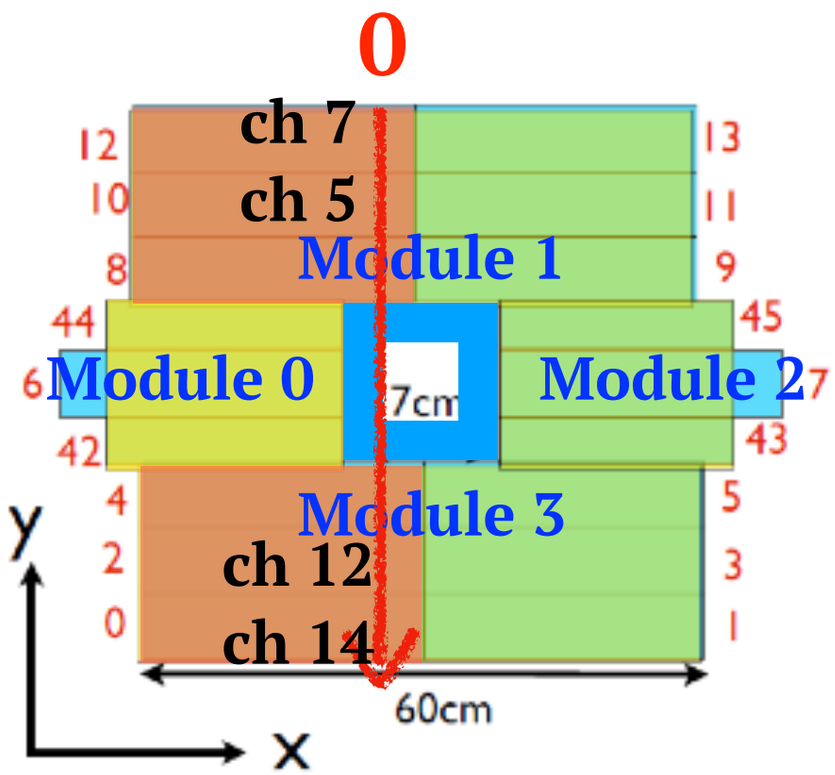
/home/had/hmkim/work/hmkim/run81/final_cal/root/run%d.root, RunID

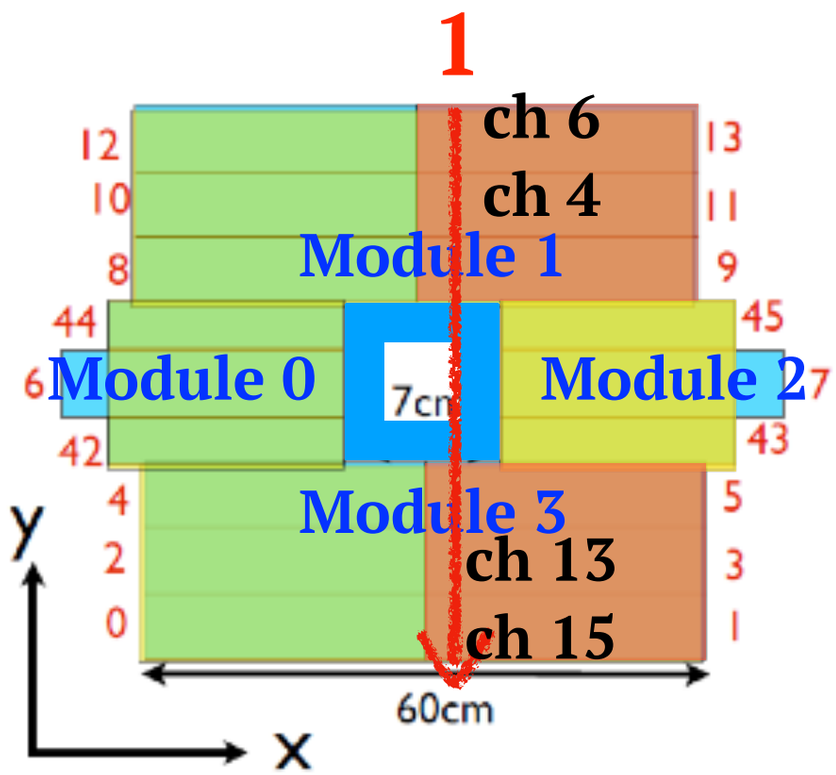
CC04, CC05 : Cosmic trigger of DCV



Assign the flag number for tracking the cosmic ray

/home/had/hmkim/work/hmkim/run81/final_cal/dcv1_calib0.C

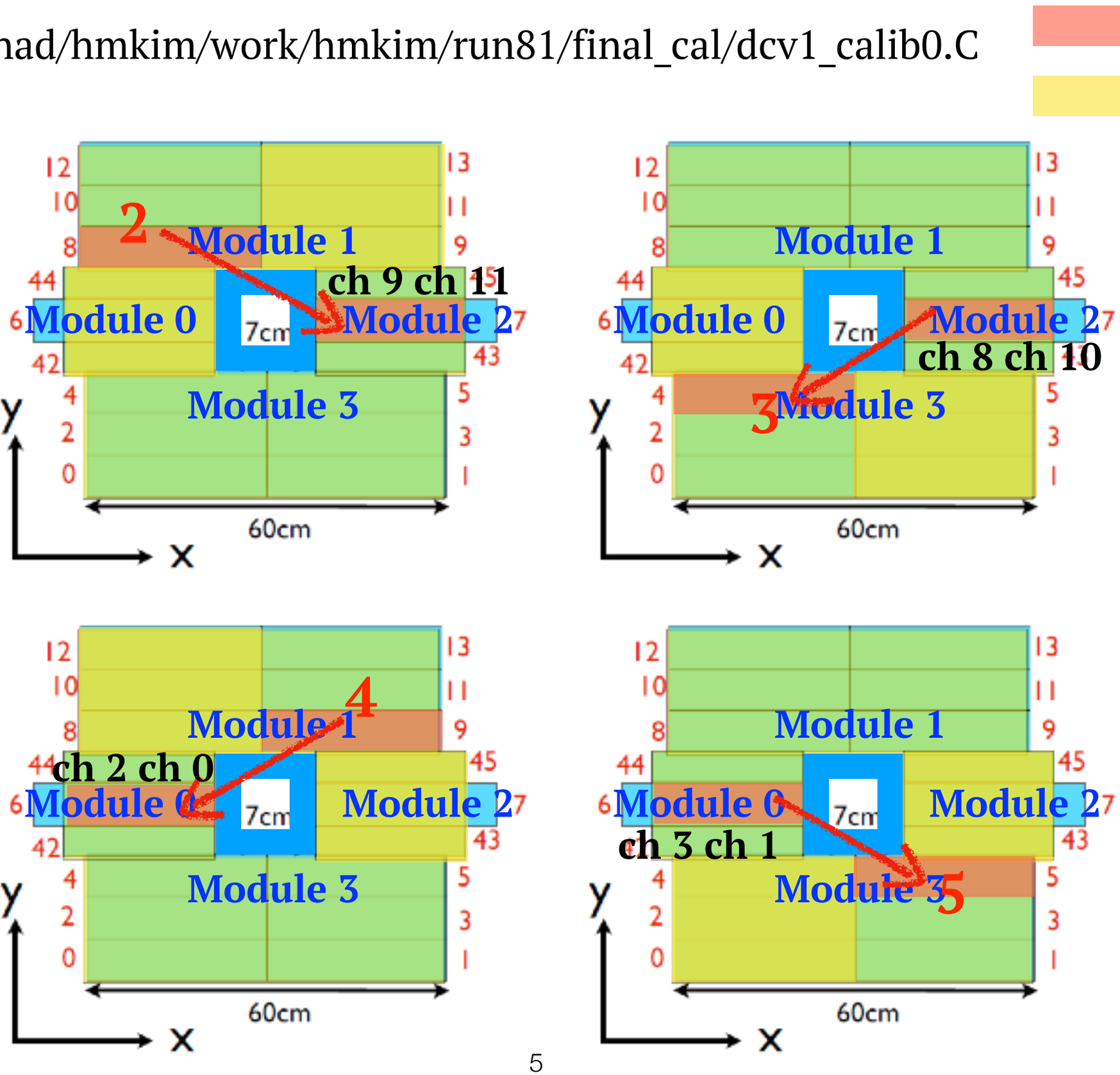




- First I got the CC04_thre from the MPV × 0.7 of CC04Ene distribution.
- Hit : CC04Ene > CC04_thre / Veto : CC04Ene < CC04_thre

Assign the flag number for tracking the cosmic ray

/home/had/hmkim/work/hmkim/run81/final_cal/dcv1_calib0.C

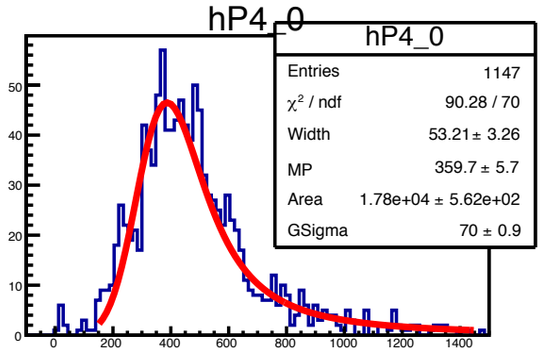


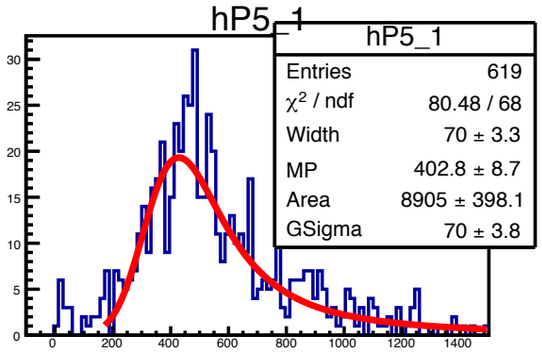
Assign the flag number for tracking the cosmic ray

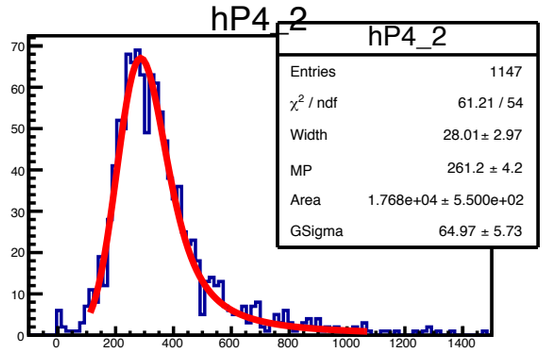
Flag Number	Hit (CC04Ene > CC04_thre(CC04MPV * 0.7))	Veto (CC04Ene < CC04_thre(CC04MPV * 0.7))	DCV1 Channel
0	(0 && 2 && 4 && 8 && 10 &&12) || (14 && 16 && 18 && 22 && 24 && 26) || (28 && 30 && 32 && 36 && 38 && 40)	6, 20, 34, 42, 44, 46, 48, 50, 52, 54 ,56	5, 7, 12, 14
1	(1 && 3 && 5 && 9 && 11 && 13) || (15 && 17 && 19 && 23 && 25 && 27) || (29 && 31 && 33 && 37 && 39 && 41)	7, 21, 35, 43, 45, 47, 49, 51, 53, 55, 57	4, 6, 13, 15
2	(8 || 22 || 36) && (7 || 21 || 35)	6, 20, 34, 42, 44, 46, 48, 50, 52, 54, 56, 9, 23, 37, 11, 25, 39, 13, 27, 41	9, 11
3	(4 || 18 || 32) && (7 || 21 || 35)	6, 20, 34, 42, 44, 46, 48, 50, 52, 54, 56, 1, 15, 29, 3, 17, 31, 5, 19, 33	8, 10
4	(9 || 23 || 37) && (6 || 20 || 34)	7, 21, 35, 43, 45, 47, 49, 51, 53, 55, 57, 8, 22, 36, 10, 24, 38, 12, 26, 40	0, 2
5	(5 || 19 || 33) && (6 || 20 || 34)	7, 21, 35, 43, 45, 47, 49, 51, 53, 55, 57, 0, 14, 28, 2, 16, 30, 4, 18, 32	1, 3

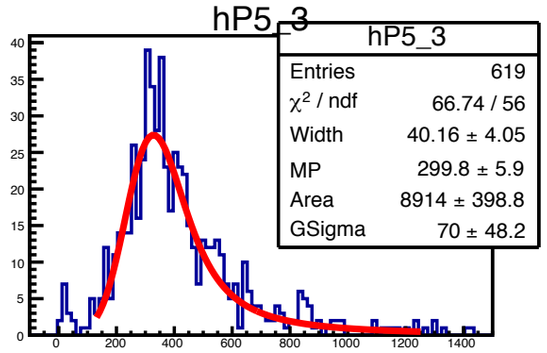
DCVPeak distribution, hP(FlagNumber)_(DCV ch)

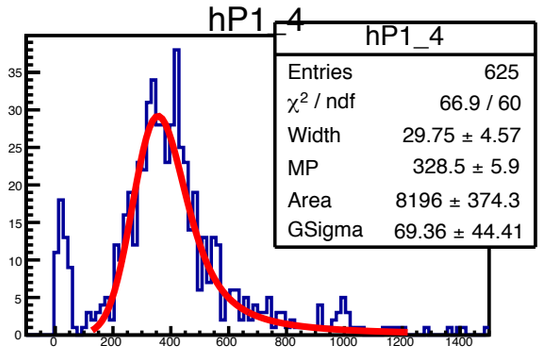
Convoluted Landau + Gauss Fit → Get the MP value

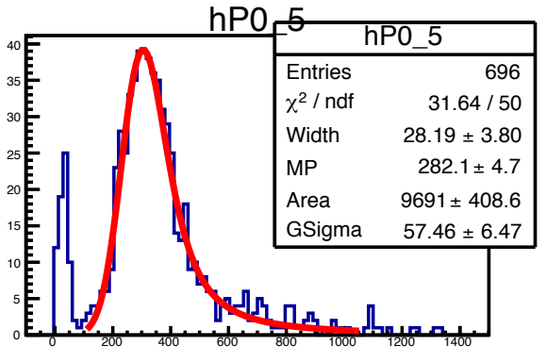


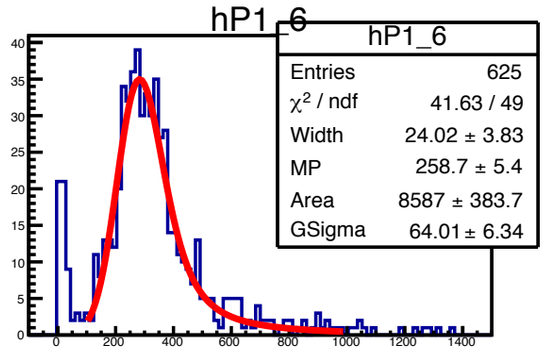


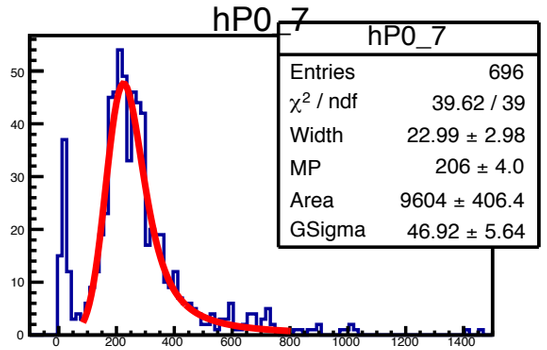


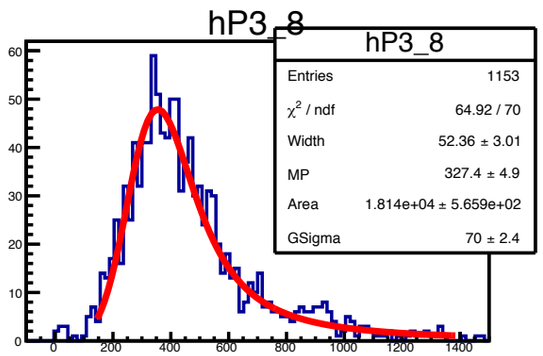


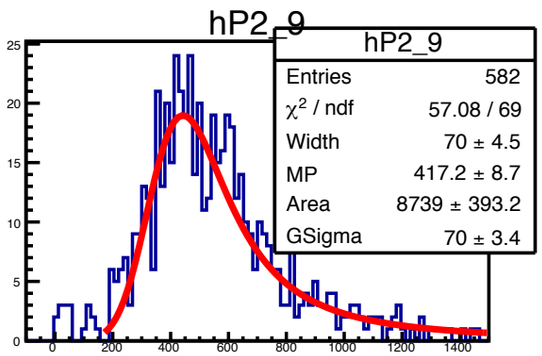


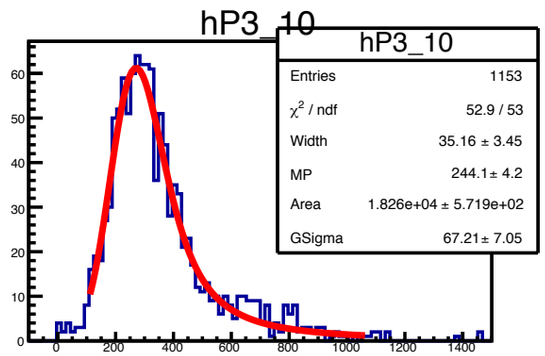


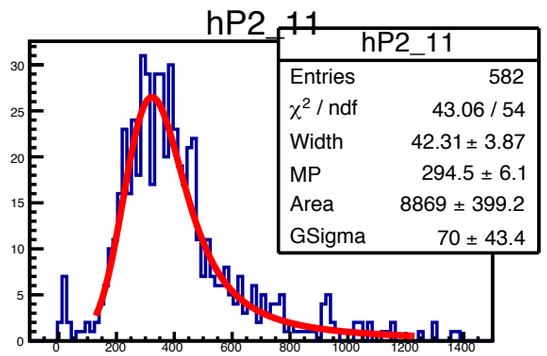


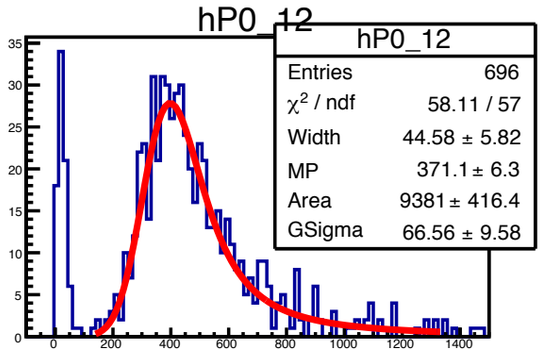


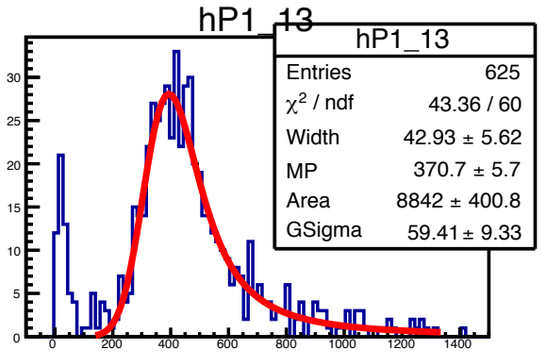


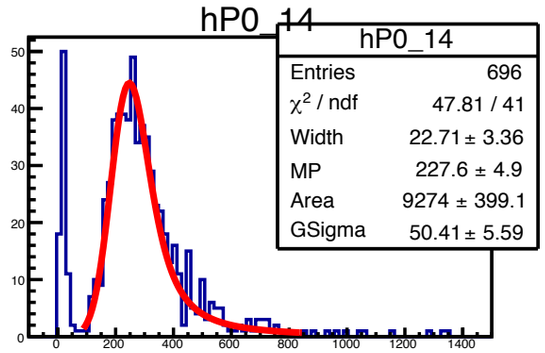


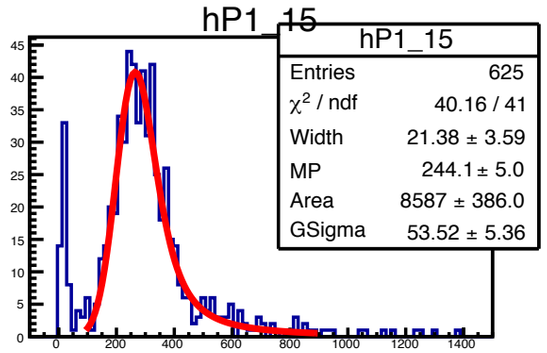












Events

/home/had/hmkim/work/hmkim/run81/final_cal/dcv1_calib0.C → dcv1_output0.root

DCVPeak

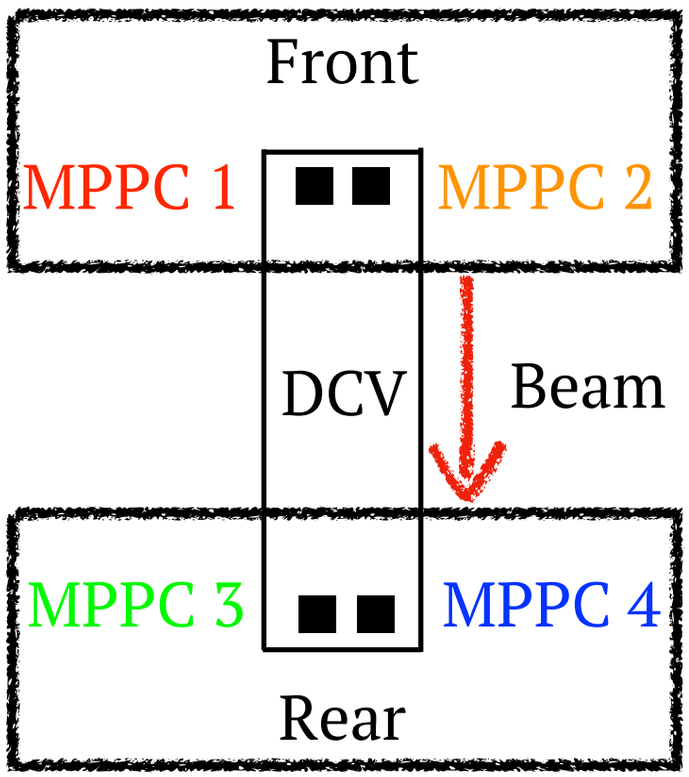
/home/had/hmkim/work/hmkim/run81/final_cal/dcv1_calib1.C

gain(MP value from hP)			
359.7	402.8	261.2	299.8
328.5	282.1	258.7	206
327.4	417.2	244.1	294.5
371.1	370.7	227.6	244.1

DCVcalib = DCVPeak / gain

(Define 'DCVcalib' for calculating the DCVEne)

hE(FlagNumber)_(PlotNumber) → Fill(DCVcalib[2*ch] + DCVcalib[2*ch + 1])



- DCVPeak is normalized to DCVcalib by being divided gain factor.
- For the energy deposit of one cosmic ray, 4 MPPCs are shared the light.
- First, I summed the 2 MPPCs at the Front and Rear.

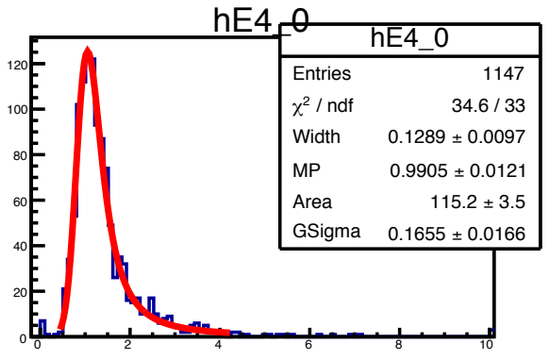
hE(FlagNumber)_(PlotNumber)

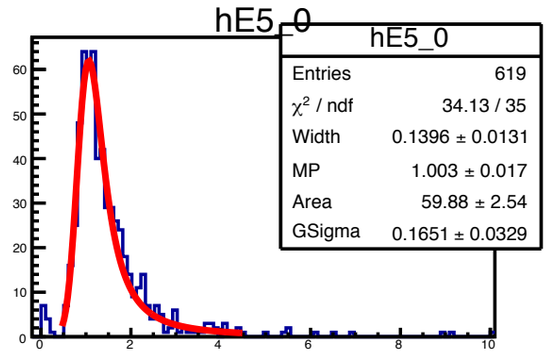
PlotNumber	DCV ch	PlotNumber	DCV ch
0	ch 0 + ch 1	4	ch 8 + ch 9
1	ch 2 + ch 3	5	ch 10 + ch 11
2	ch 4 + ch 5	6	ch 12 + ch 13
3	ch 6 + ch 7	7	ch 14 + ch 15

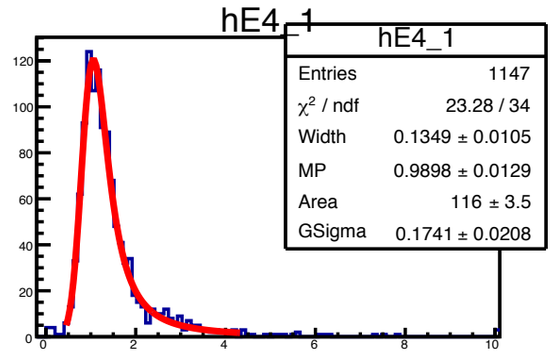
ch 0 : hE4_0	ch 1 : hE5_0	ch 2 : hE4_1	ch 3 : hE5_1
ch 4 : hE1_2	ch 5 : hE0_2	ch 6 : hE1_3	ch 7 : hE0_3
ch 8 : hE3_4	ch 9 : hE2_4	ch 10 : hE3_5	ch 11 : hE2_5
ch 12 : hE0_6	ch 13 : hE1_6	ch 14 : hE0_7	ch 15 : hE1_7

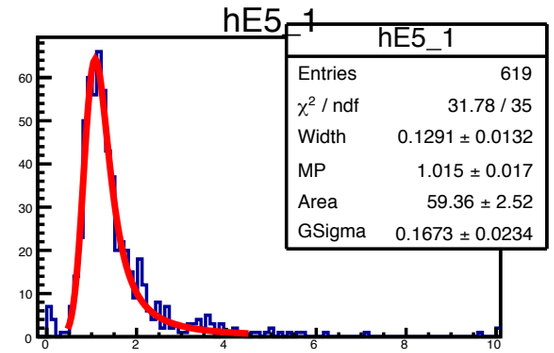
DCVcalib distribution, hE(FlagNumber)_(PlotNumber)

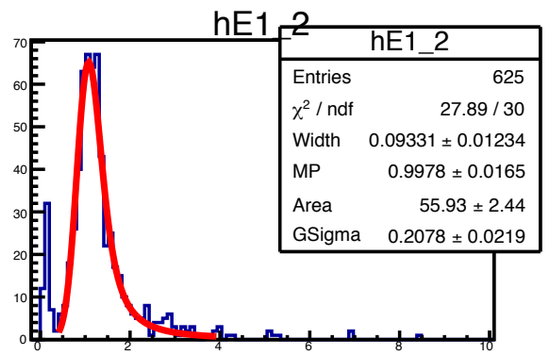
Convoluted Landau + Gauss Fit → Get the MP value

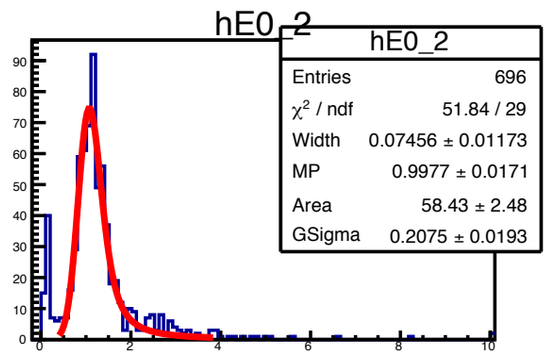


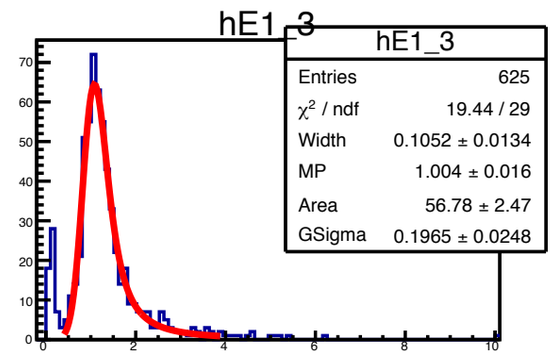


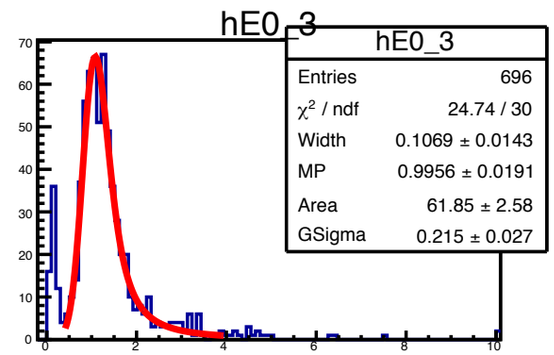


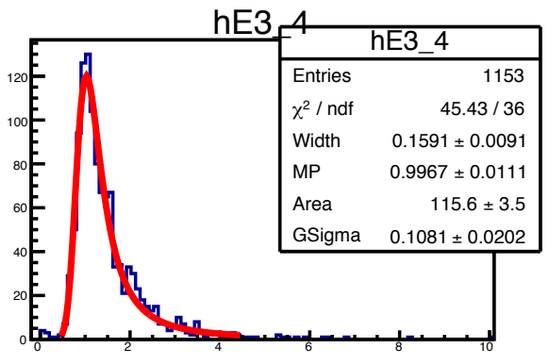


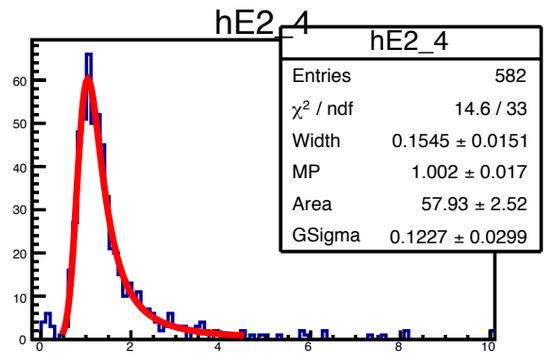


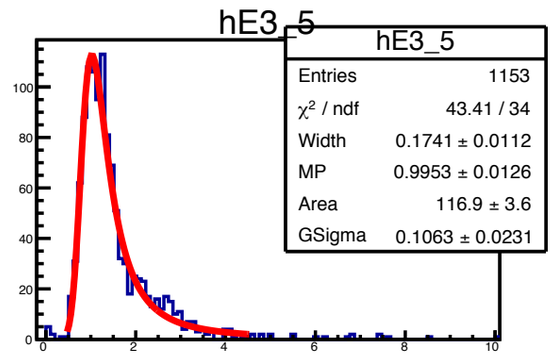


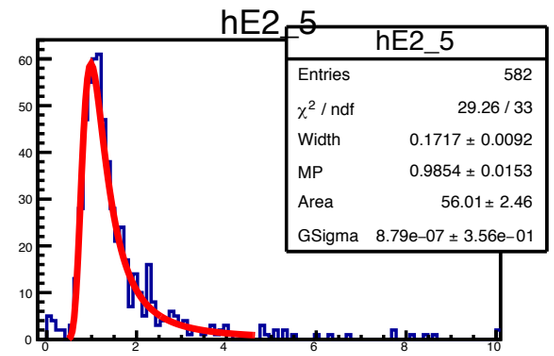


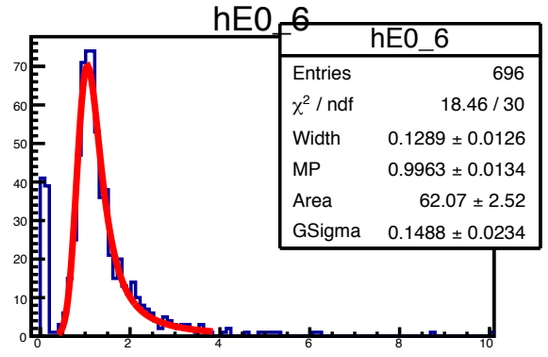


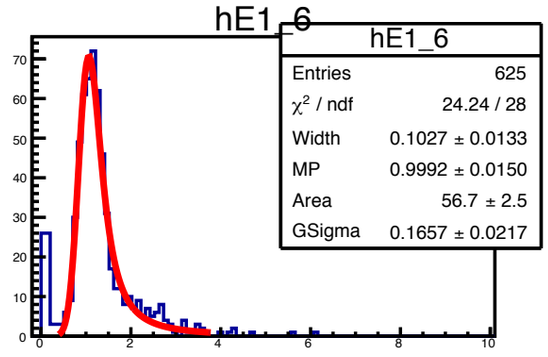


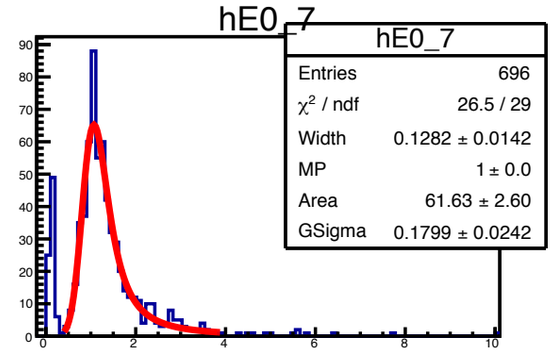


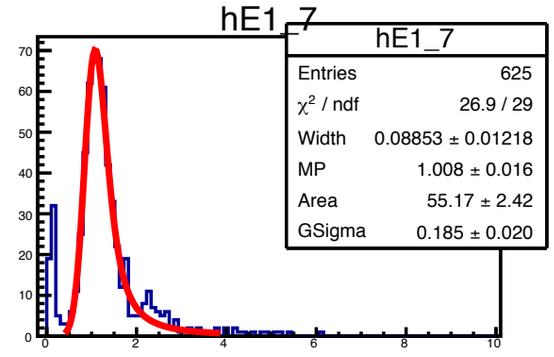












Events

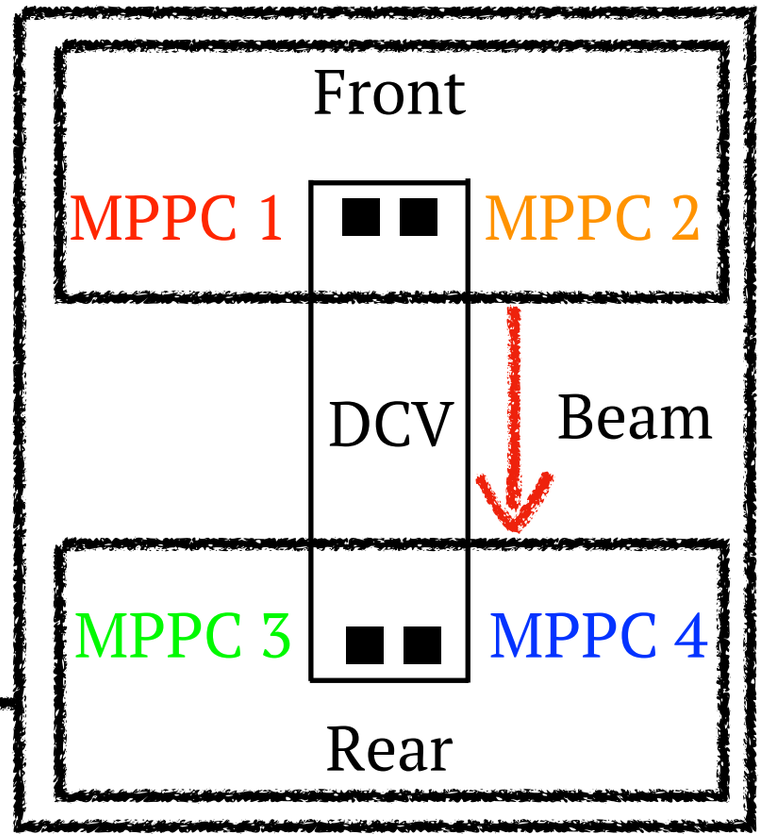
/home/had/hmkim/work/hmkim/run81/final_cal/dcv1_calib1.C → dcv1_output1.root

DCVcalib[MeV]

/home/had/hmkim/work/hmkim/run81/final_cal/dcv1_calib2.C

gain(MP value from hP)			
359.7	402.8	261.2	299.8
328.5	282.1	258.7	206
327.4	417.2	244.1	294.5
371.1	370.7	227.6	244.1

norm(MP value from hE)			
1.291	1.349	1.273	1.325
1.179	1.202	1.185	1.181
1.316	1.324	1.334	1.338
1.178	1.195	1.167	1.182



DCVcalib = DCVPeak / (gain*norm)

(Define 'DCVcalib' for calculating the DCVEne)

hES(FlagNumber)_(PlotNumber)
→ Fill(DCVcalib[4*ch] + DCVcalib[4*ch+1]
+DCVcalib[4*ch+2]+DCVcalib[4*ch+3])

- DCVPeak is normalized to DCVcalib by being divided gain and norm factor.
- For the energy deposit of one cosmic ray, 4 MPPCs are shared the light.
- Second, I summed the 4 MPPCs in the same module.

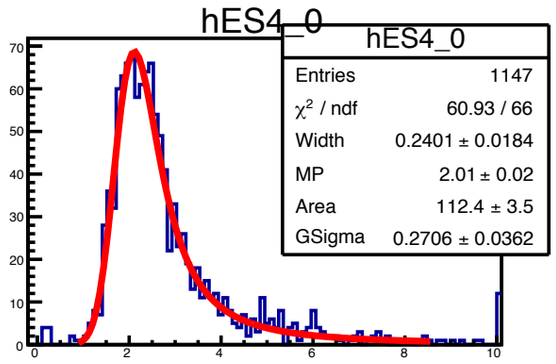
hES(FlagNumber)_(PlotNumber)

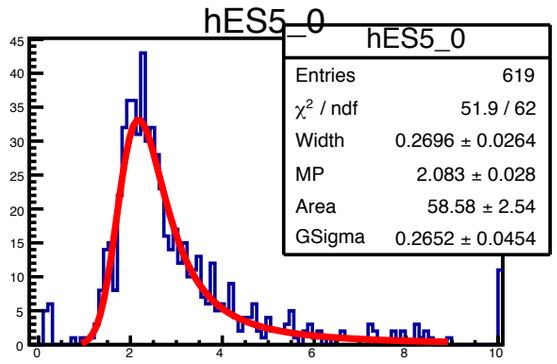
PlotNumber	DCV ch
0	ch 0 + ch 1 + ch 2 + ch 3
1	ch 4 + ch 5 + ch 6 + ch 7
2	ch 8 + ch 9 + ch 10 + ch 11
3	ch 12 + ch 13 + ch 14 + ch 15

ch 0 : hES4_0	ch 1 : hES5_0	ch 2 : hES4_0	ch 3 : hES5_0
ch 4 : hES1_1	ch 5 : hES0_1	ch 6 : hES1_1	ch 7 : hES0_1
ch 8 : hES3_2	ch 9 : hES2_2	ch 10 : hES3_2	ch 11 : hES2_2
ch 12 : hES0_3	ch 13 : hES1_3	ch 14 : hES0_3	ch 15 : hES1_3

DCVcalib distribution, hES(FlagNumber)_(PlotNumber)

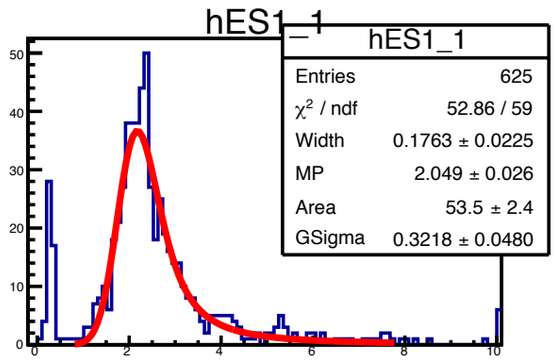
ch 0+1+2+3

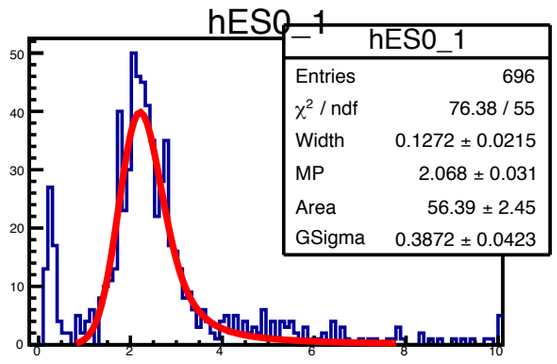




ch 0+1+2+3

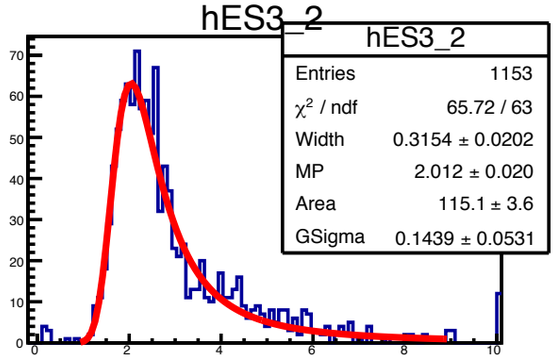
ch 4+5+6+7

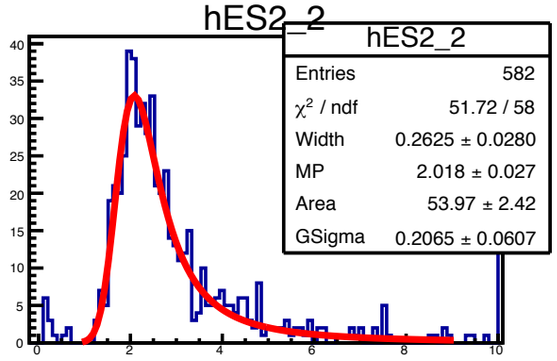




ch 4+5+6+7

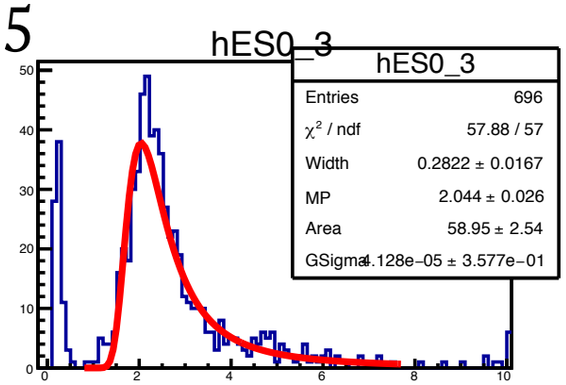
ch 8+9+10+11

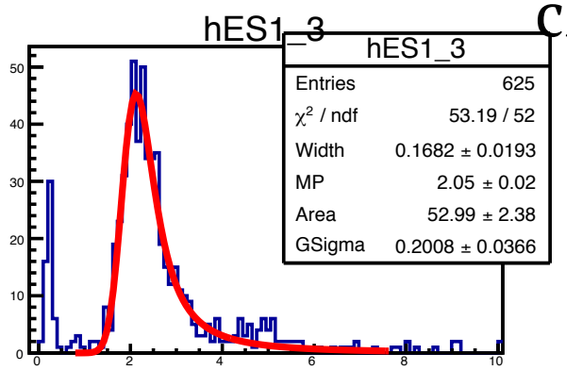




ch 8+9+10+11

ch 12+13+14+15





ch 12+13+14+15

Events

DCVcalib[MeV]

/home/had/hmkim/work/hmkim/run81/final_cal/dcv1_calib2.C → dcv1_output2.root

/home/had/hmkim/work/hmkim/run81/final_cal/dcv1_calib3.C

gain(MP value from hP)			
359.7	402.8	261.2	299.8
328.5	282.1	258.7	206
327.4	417.2	244.1	294.5
371.1	370.7	227.6	244.1

norm(MP value from hE)			
1.291	1.349	1.273	1.325
1.179	1.202	1.185	1.181
1.316	1.324	1.334	1.338
1.178	1.195	1.167	1.182

norm_module(MP value from hES)			
2.01	2.083	2.01	2.083
2.049	2.068	2.049	2.068
2.012	2.018	2.012	2.018
2.044	2.05	2.044	2.05

↓

DCVcalib = DCVPeak / (gain*norm*norm_module)

(Define ‘DCVcalib’ for calculating the DCVEne)

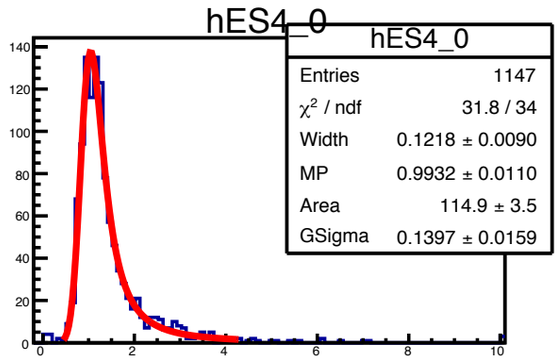
↓

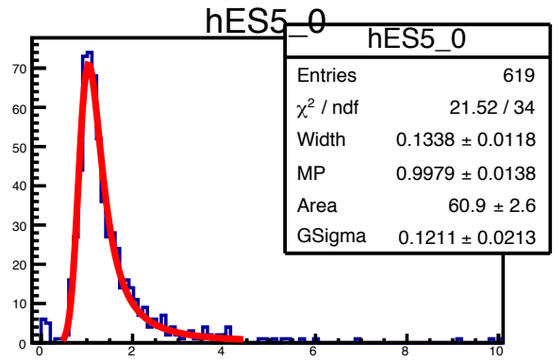
hES(FlagNumber)_(PlotNumber)
→ Fill(DCVcalib[4*ch] + DCVcalib[4*ch+1]
+DCVcalib[4*ch+2]+DCVcalib[4*ch+3])

- DCVPeak is normalized to DCVcalib by being divided gain and norm factor and norm_module factor

DCVcalib distribution, hES(FlagNumber)_(PlotNumber)

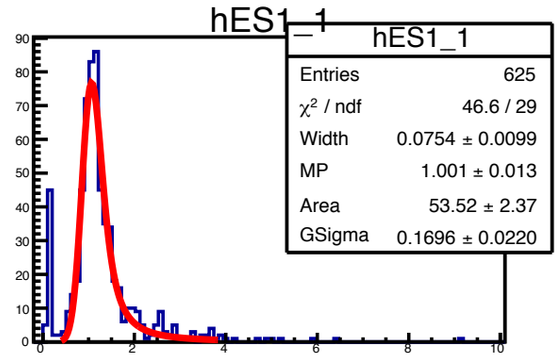
ch 0+1+2+3

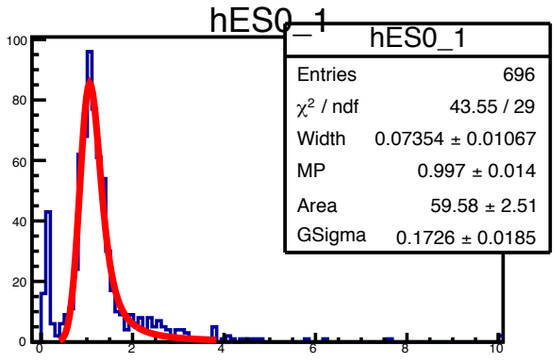




ch 0+1+2+3

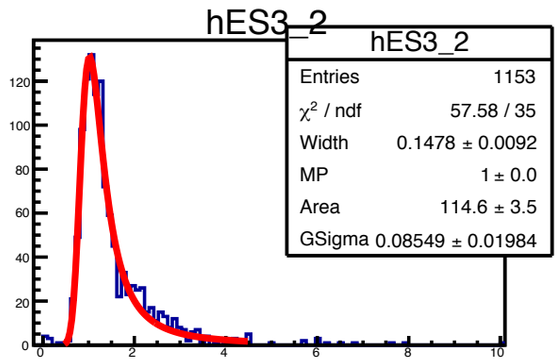
ch 4+5+6+7

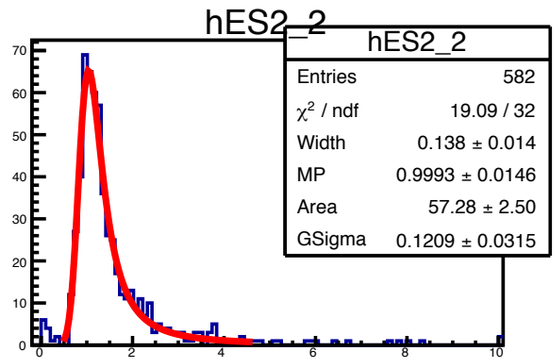




ch 4+5+6+7

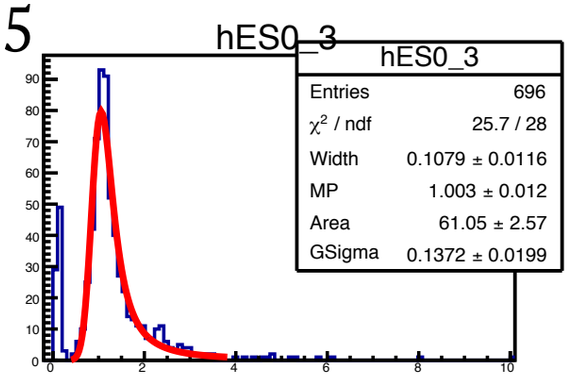
ch 8+9+10+11

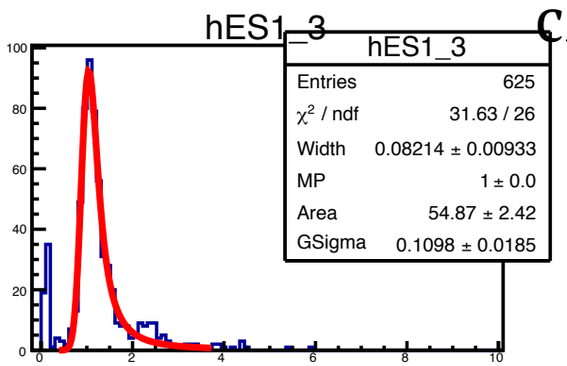




ch 8+9+10+11

ch 12+13+14+15





ch 12+13+14+15

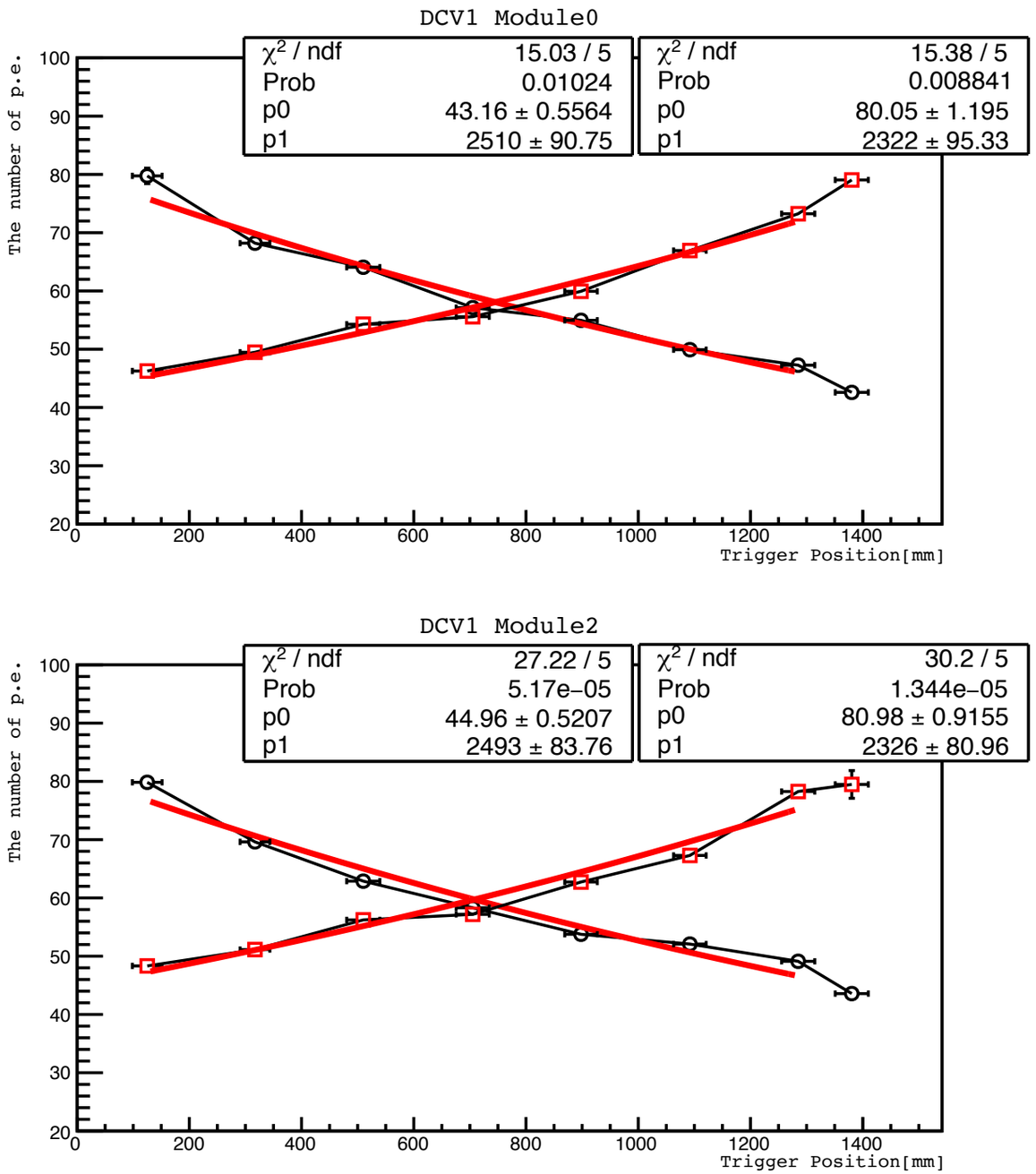
Events

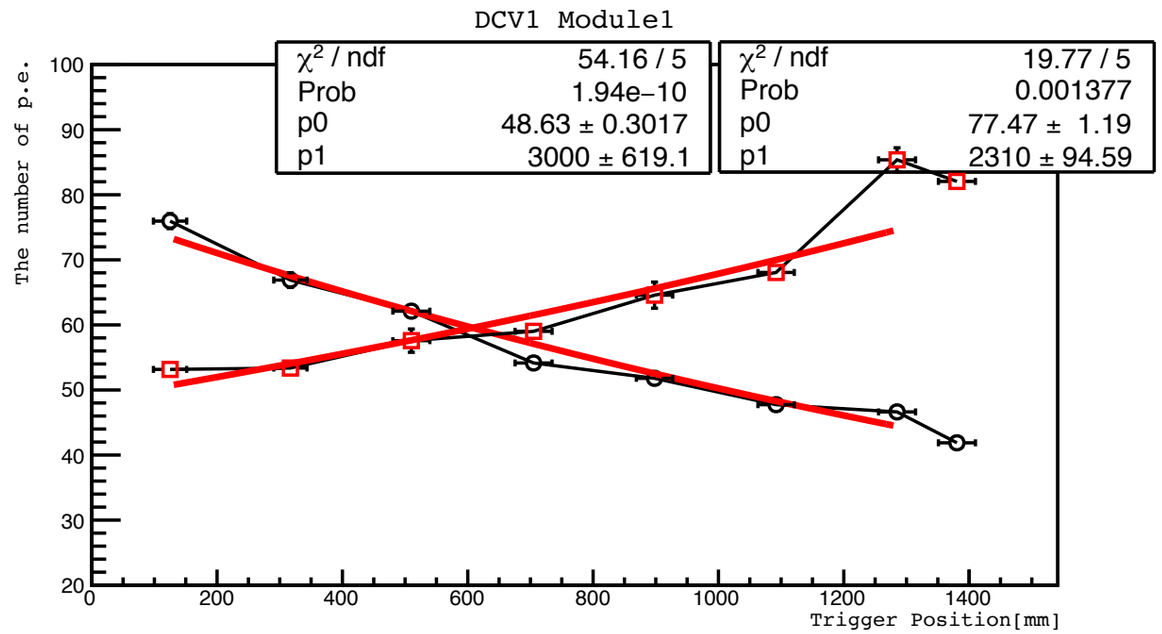
DCVcalib[MeV]

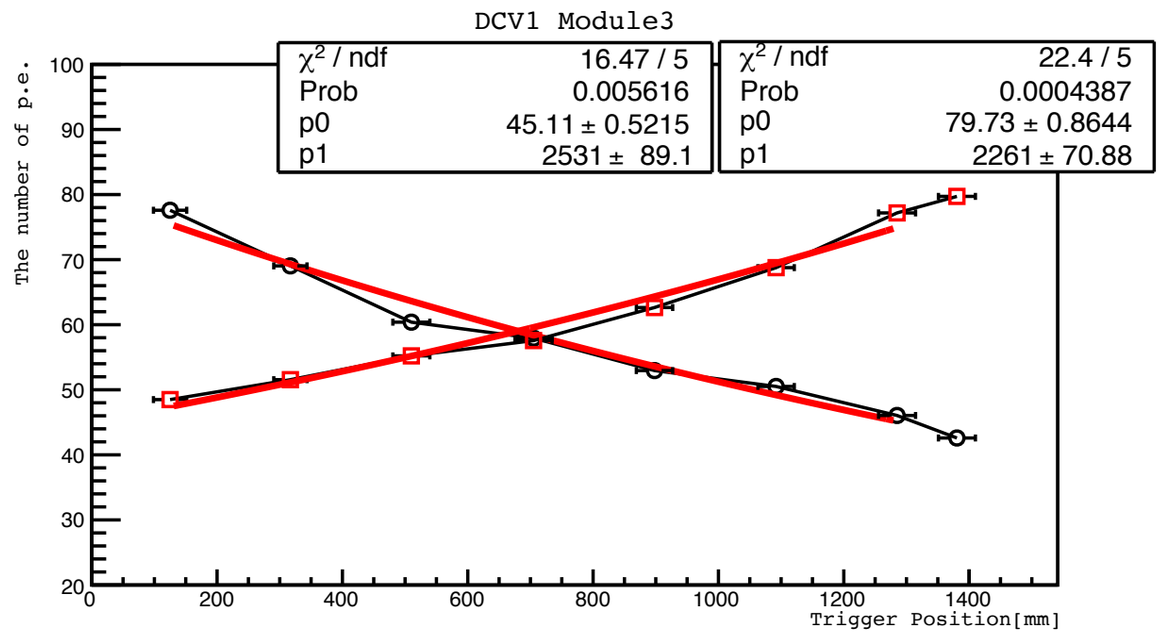
/home/had/hmkim/work/hmkim/run81/final_cal/dcv1_calib3.C → dcv1_output3.root

Attenuation effect

- fitting function = [0] * exp (±x / [1])
- fitting range 128 mm ~ 1285 mm

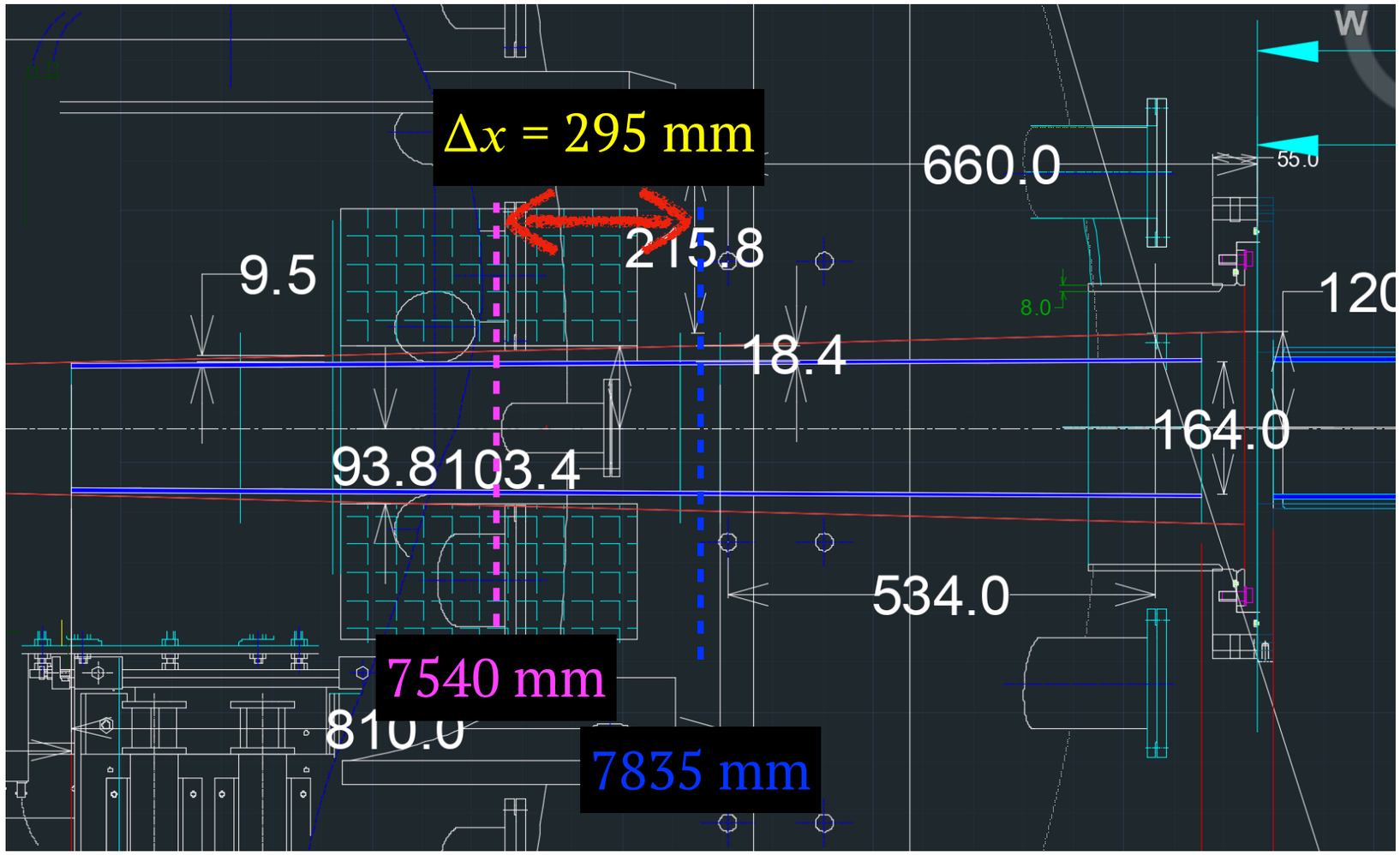


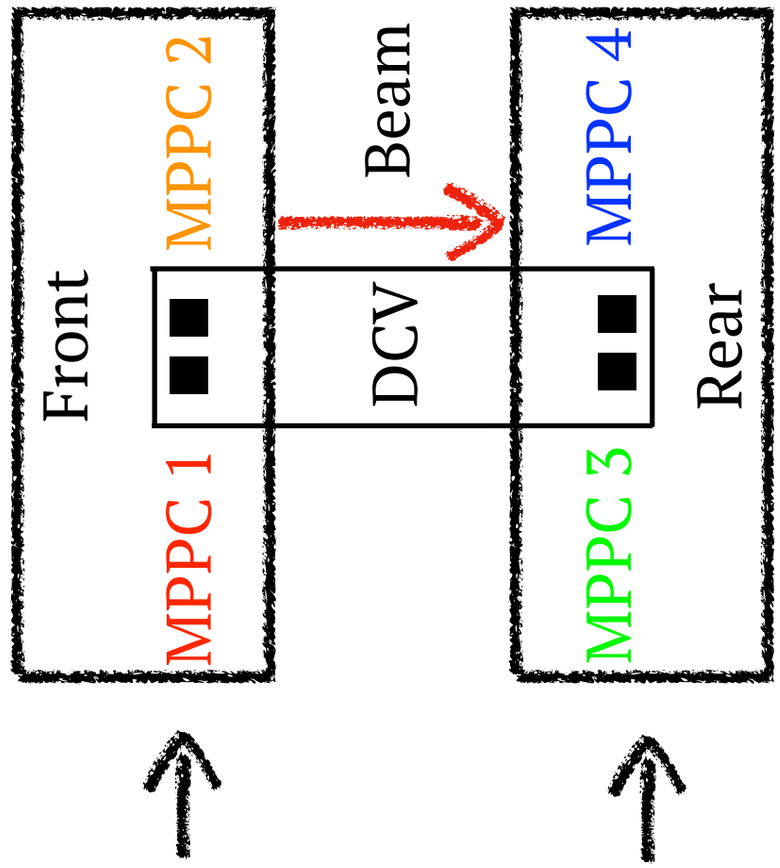




Attenuation Length λ_{avg} = **2469 mm**

Calculation of attenuation factor





lambda_dcv1 = 2469; // Attenuation Length of DCV1
pos_DCV1 = 7835; // Center position of DCV1
pos_CC04 = 7540; // Center position of CC04

att1_f = exp((pos_DCV1 - pos_CC04)/lambda_dcv1);
att1_r = exp(-(pos_DCV1 - pos_CC04)/lambda_dcv1);

att1_f =1.12691

att1_r =0.88738

/home/had/hmkim/work/hmkim/run81/final_cal/dcv1_calib3.C

att			
1.12691	1.12691	0.88738	0.88738
1.12691	1.12691	0.88738	0.88738
1.12691	1.12691	0.88738	0.88738
1.12691	1.12691	0.88738	0.88738

↓

DCVcalib = DCVPeak / (gain*norm*norm_module)
DCVcalib_att = DCVcalib * att

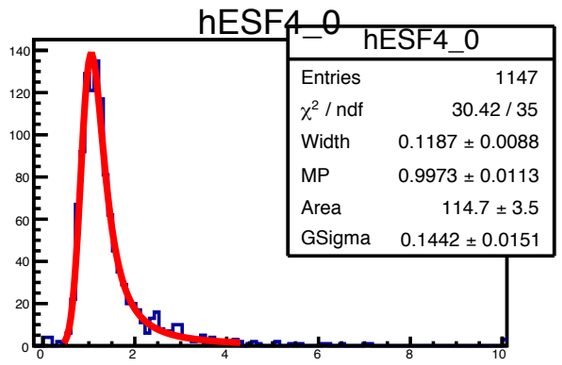
(Define ‘DCVcalib’ for calculating the DCVEne
and ‘DCVcalib_att’ for including attenuation effect)

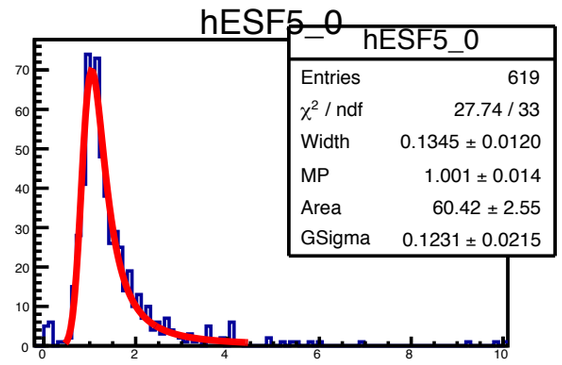
↓

hESF(FlagNumber)_(PlotNumber)
→ Fill(DCVcalib_att[4*ch] + DCVcalib_att[4*ch+1]
+DCVcalib_att[4*ch+2]+DCVcalib_att[4*ch+3])

DCVcalib distribution including att. effect, hESF(FlagNum)_(PlotNum)

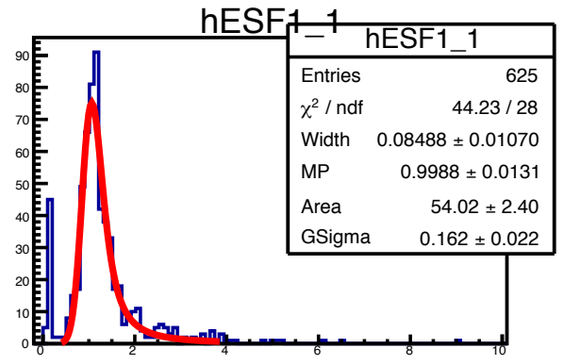
ch 0+1+2+3

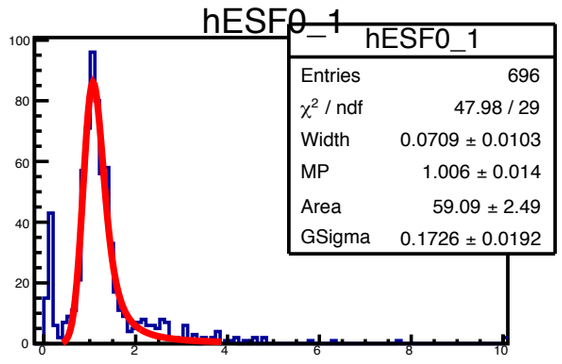




ch 0+1+2+3

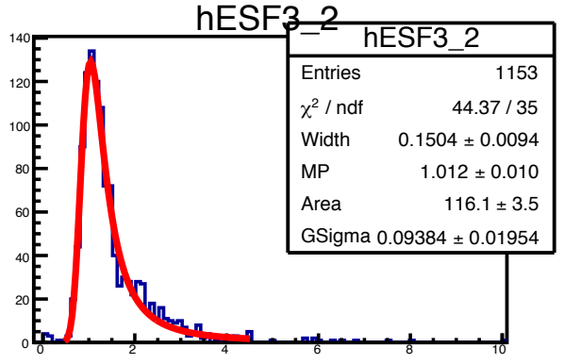
ch 4+5+6+7

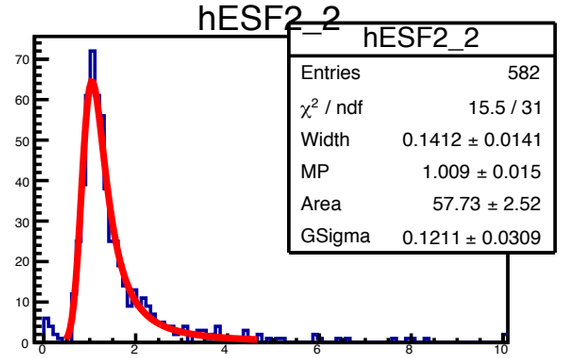




ch 4+5+6+7

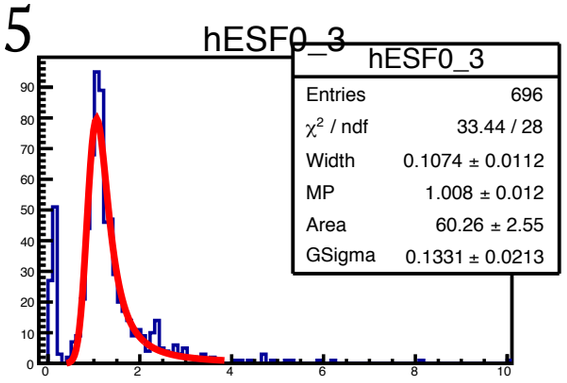
ch 8+9+10+11

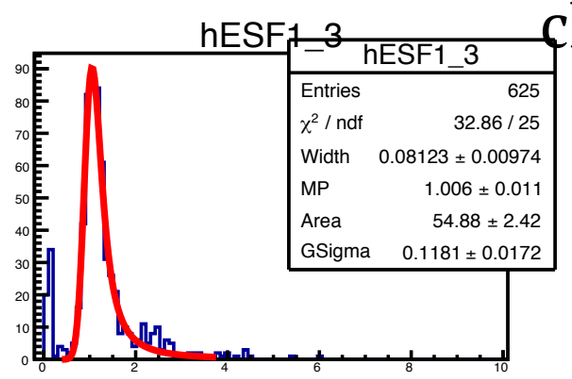




ch 8+9+10+11

ch 12+13+14+15





ch 12+13+14+15

Events

DCVcalib_att[MeV]

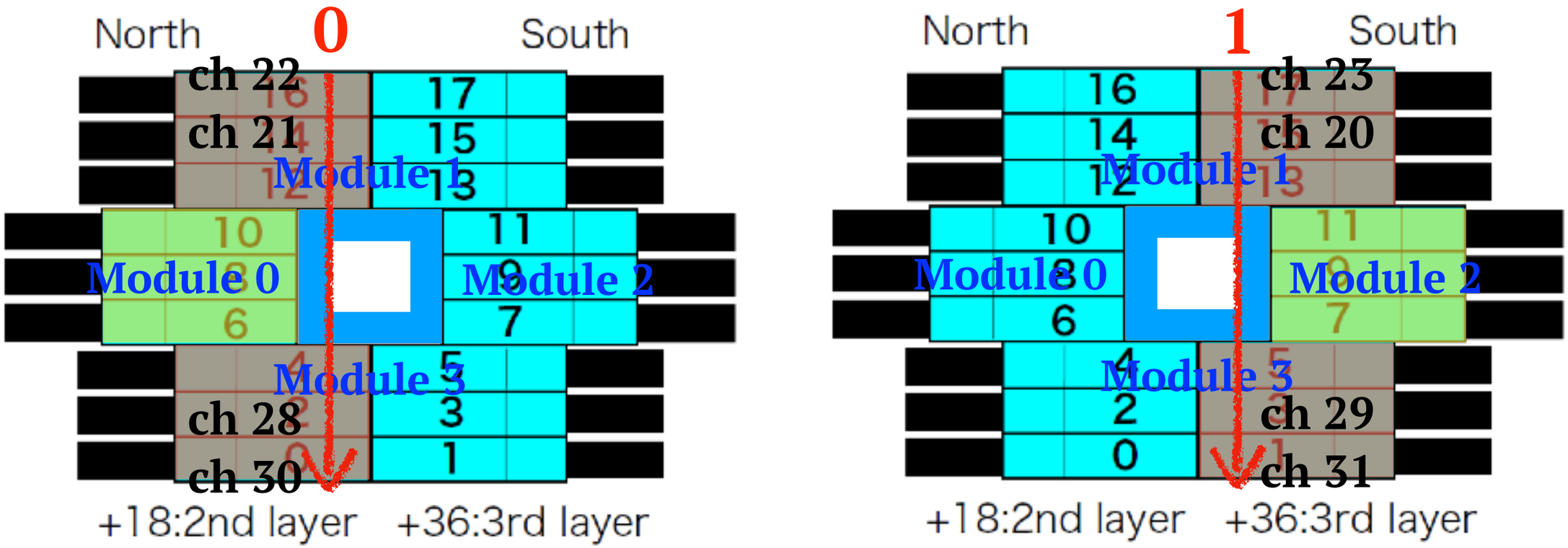
/home/had/hmkim/work/hmkim/run81/final_cal/dcv1_calib3.C → dcv1_output3.root

Assign the flag number for tracking the cosmic ray

/home/had/hmkim/work/hmkim/run81/final_cal/dcv1_calib0.C

Hit

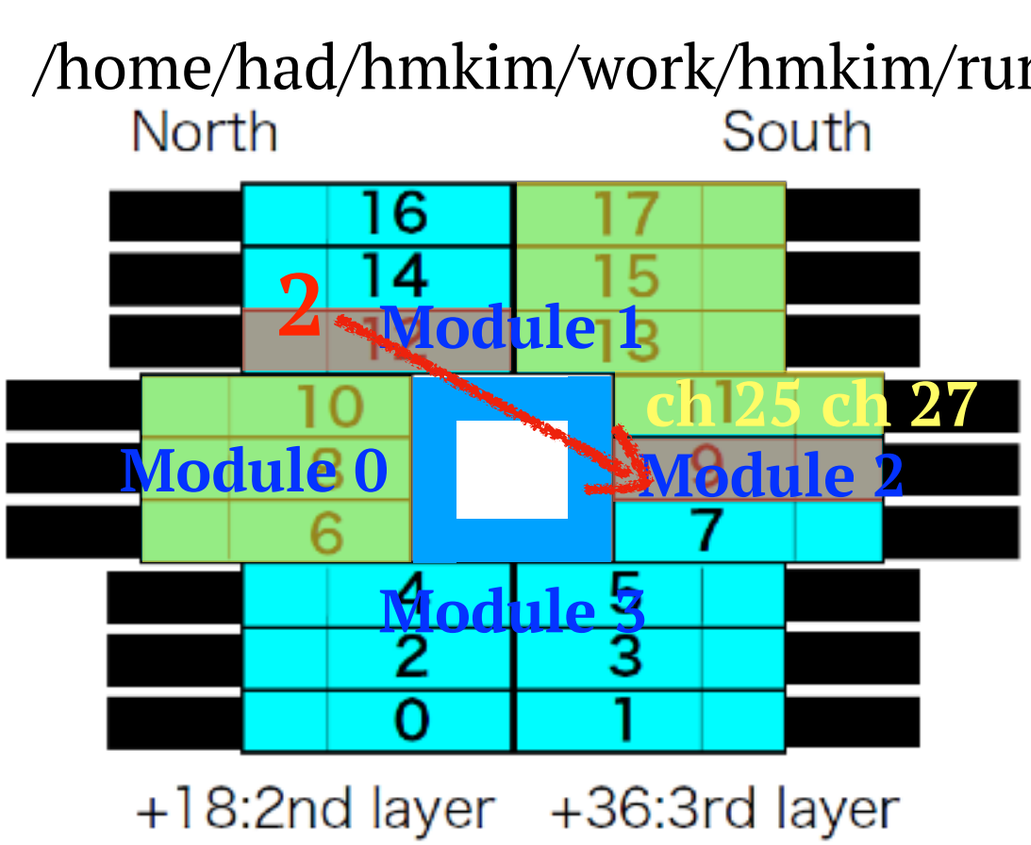
Veto

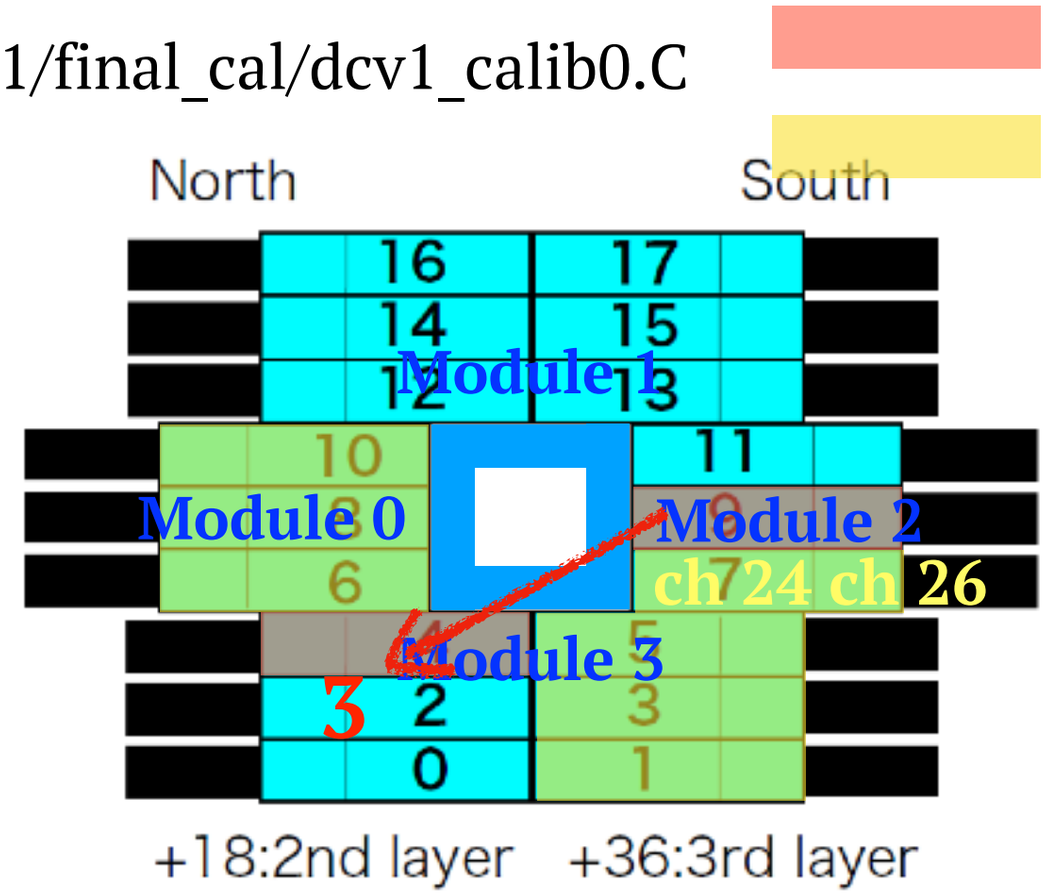


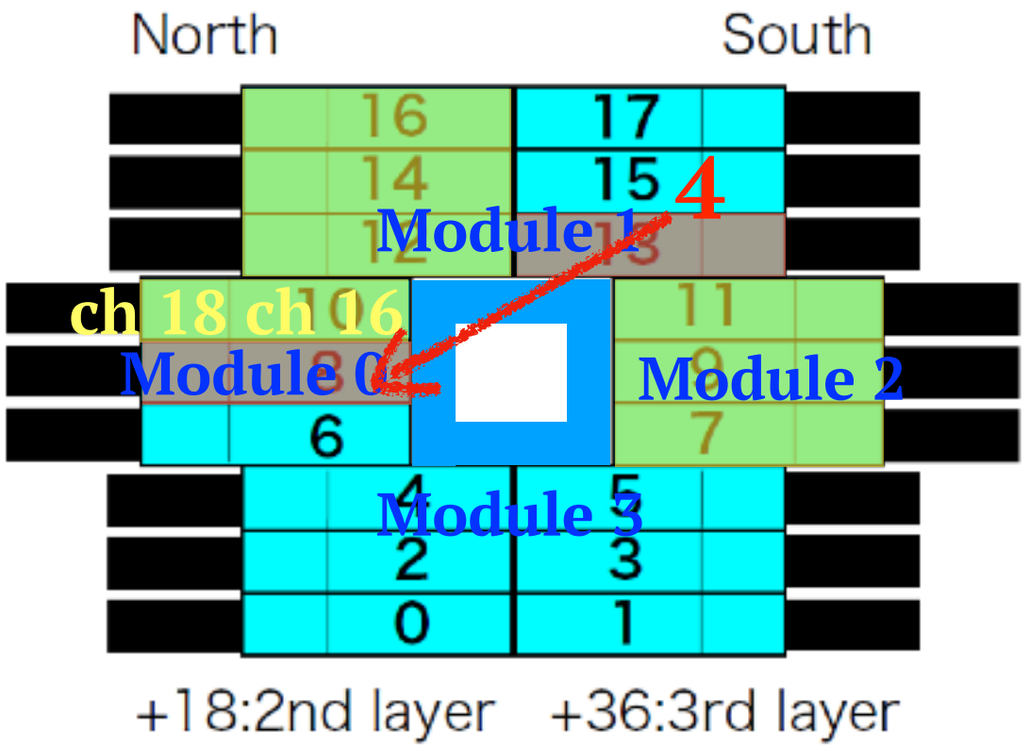
- First I got the CC05_thre from the MPV × 0.7 of CC05Ene distribution.
- Hit : CC05Ene > CC05_thre / Veto : CC05Ene < CC05_thre

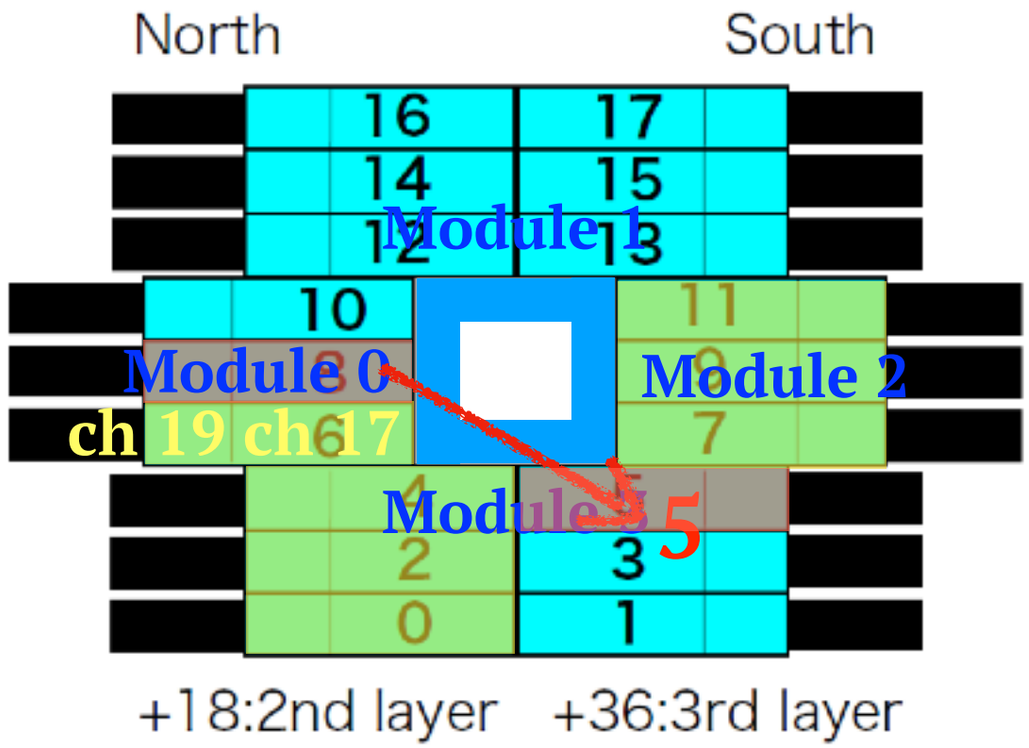
Assign the flag number for tracking the cosmic ray

/home/had/hmkim/work/hmkim/run81/final_cal/dcv1_calib0.C







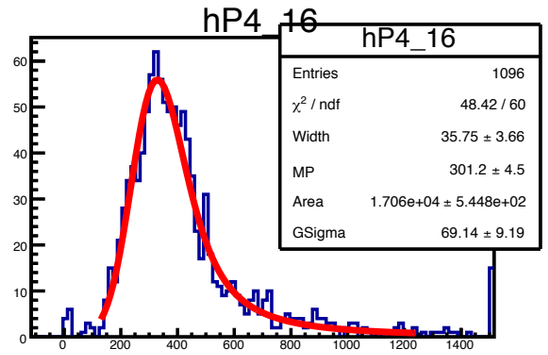


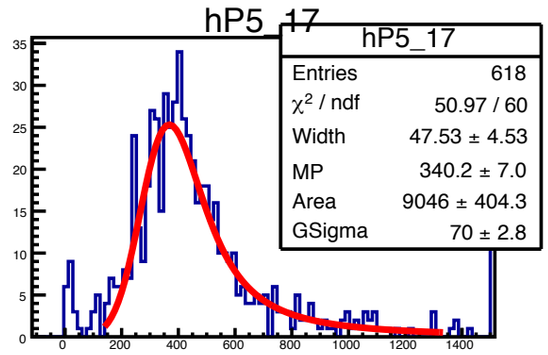
Assign the flag number for tracking the cosmic ray

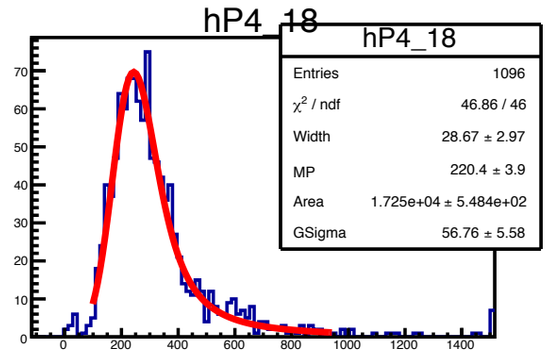
Flag Number	Hit (CC05Ene > CC05_thre(CC05MPV*0.7)	Veto (CC05Ene < CC05_thre(CC05MPV*0.7))	DCV2 Channel
0	(0 && 2 && 4 && 12 && 14 &&16) || (18 && 20 && 22 && 30 && 32 && 34) || (36 && 38 && 40 && 48 && 50 && 52)	6, 8, 10, 24, 26, 28, 42, 44, 46	21, 23, 28, 30
1	(1 && 3 && 5 && 13 && 15 && 17) || (19 && 21 && 23 && 31 && 33 && 35) || (37 && 39 && 41 && 49 && 51 && 53)	7, 9, 11, 25, 27, 29, 43, 45, 47	20, 22, 29, 31
2	(12 || 30 || 48) && (9 || 27 || 45)	6, 8, 10, 24, 26, 28, 42, 44, 46, 13, 15, 17, 31, 33, 35, 49, 51, 53, 11, 29, 47	25, 27
3	(4 || 22 || 40) && (9 || 27 || 45)	6, 8, 10, 24, 26, 28, 42, 44, 46, 1, 3, 5, 19, 21, 23, 37, 39, 41, 7, 25, 43	24, 26
4	(13 || 31 || 49) && (8 || 26 || 44)	7, 9, 11, 25, 27, 29, 43, 45, 47, 12, 14, 16, 30, 32, 34, 48, 50, 52, 10, 28, 46	16, 18
5	(5 || 23 || 41) && (8 || 26 || 44)	7, 9, 11, 25, 27, 29, 43, 45, 47, 0, 2, 4, 18, 20, 22, 36, 38, 40, 6, 24, 42	17, 19

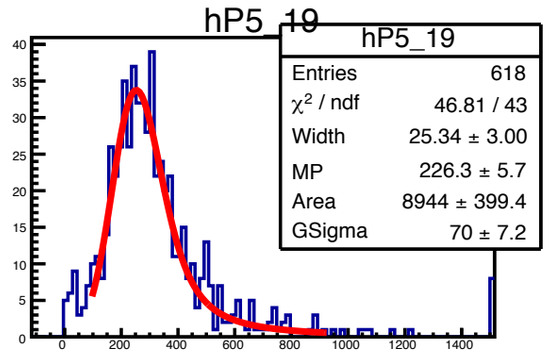
DCVPeak distribution, hP(FlagNumber)_(DCV ch)

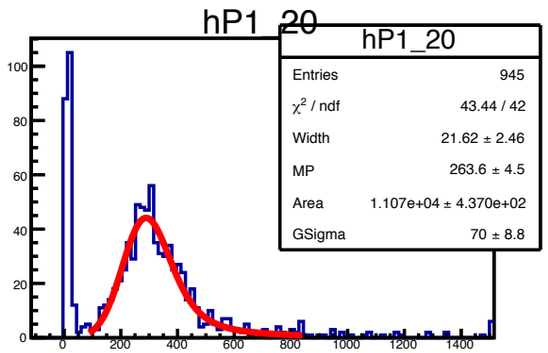
Convoluted Landau + Gauss Fit → Get the MP value

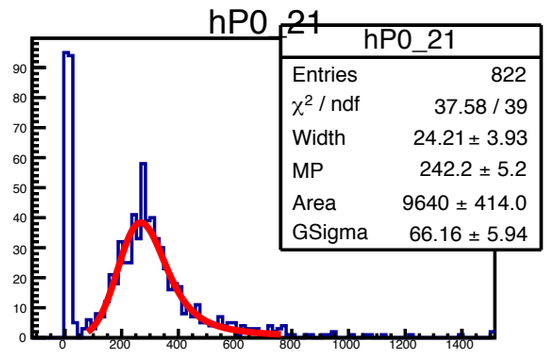


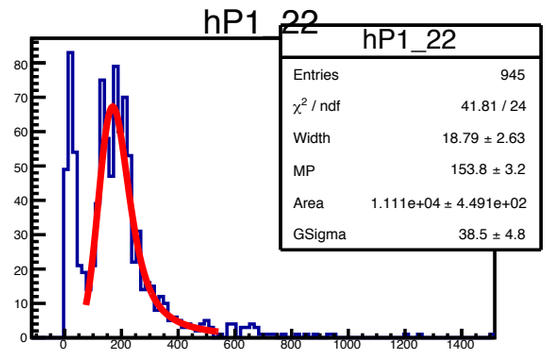


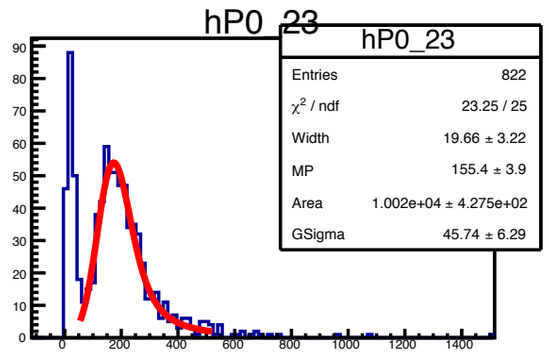


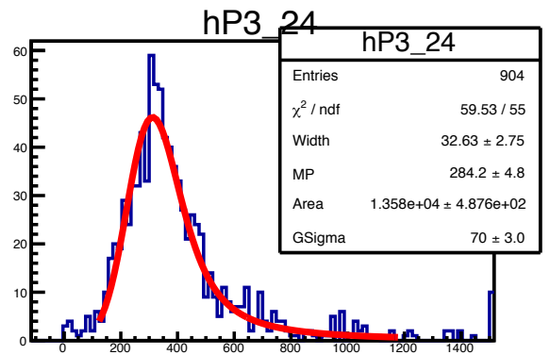


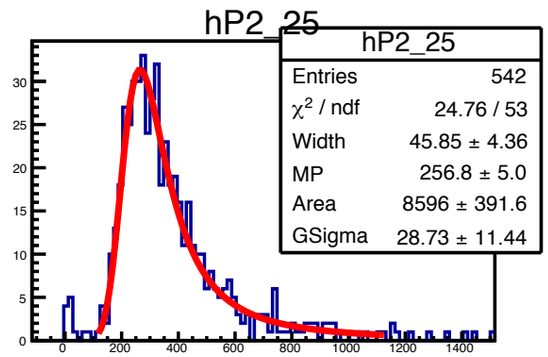


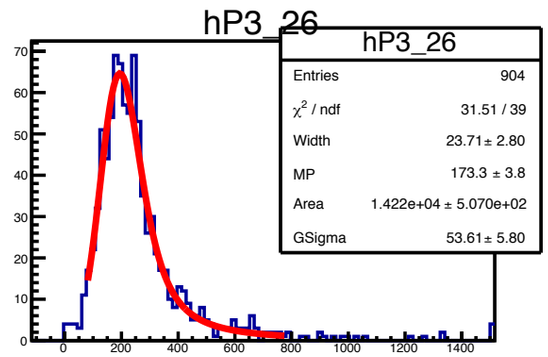


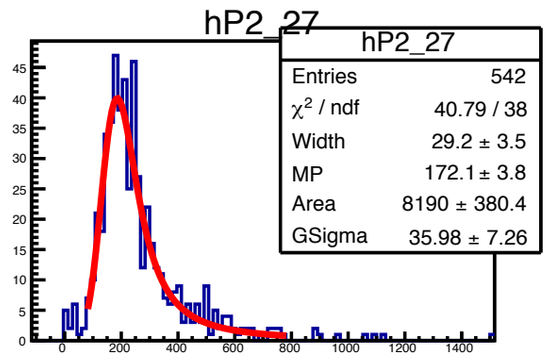


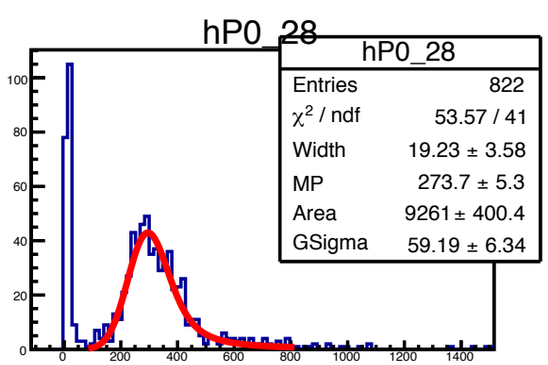


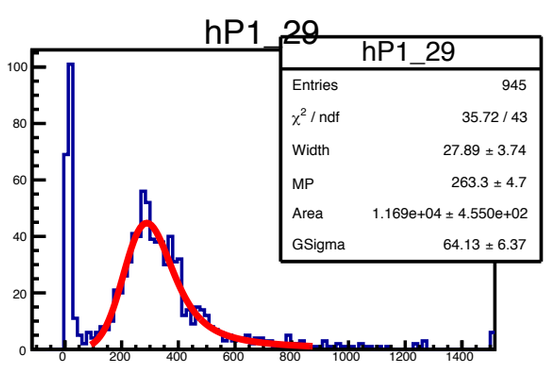


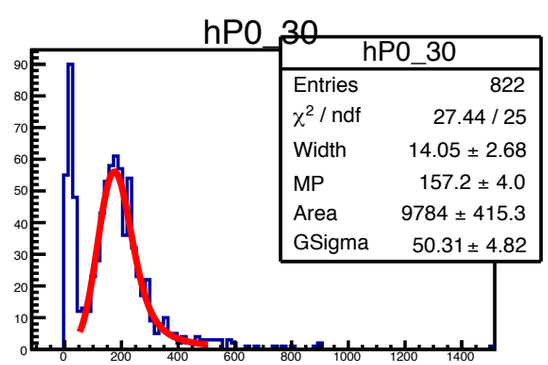


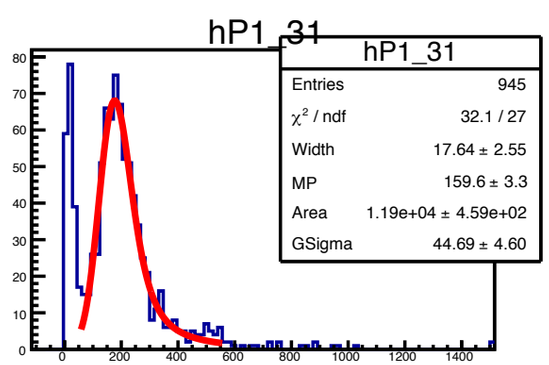












Events

/home/had/hmkim/work/hmkim/run81/final_cal/dcv2_calib0.C → dcv2_output0.root

DCVPeak

/home/had/hmkim/work/hmkim/run81/final_cal/dcv2_calib1.C

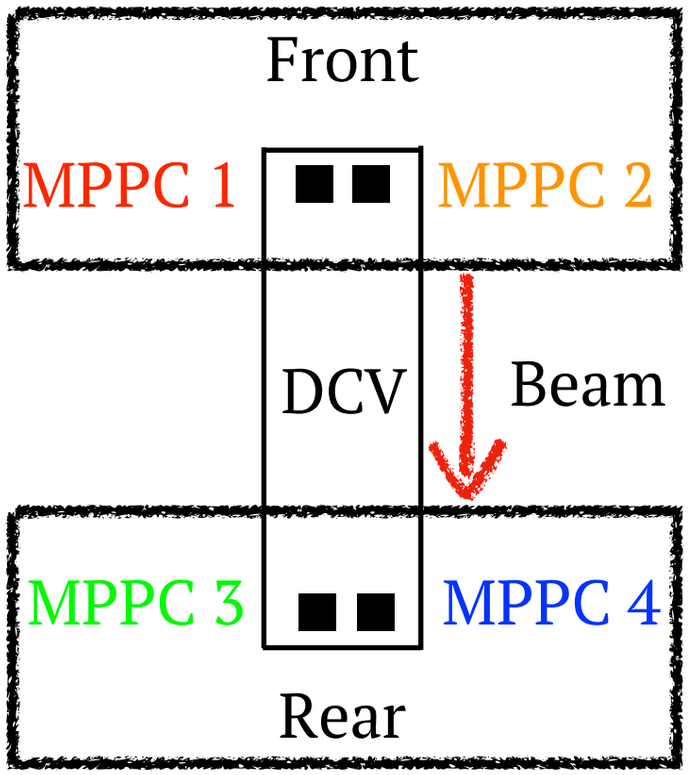
gain(MP value from hP)			
301.2	340.2	220.4	226.3
263.6	242.2	153.8	155.4
284.2	256.8	173.3	172.1
273.7	263.3	157.2	159.6

↓

DCVcalib = DCVPeak / gain

↓ (Define 'DCVcalib' for calculating the DCVEne)

hE(FlagNumber)_(PlotNumber) → Fill(DCVcalib[2*ch] + DCVcalib[2*ch + 1])



- DCVPeak is normalized to DCVcalib by being divided gain factor.
- For the energy deposit of one cosmic ray, 4 MPPCs are shared the light.
- First, I summed the 2 MPPCs at the Front and Rear.

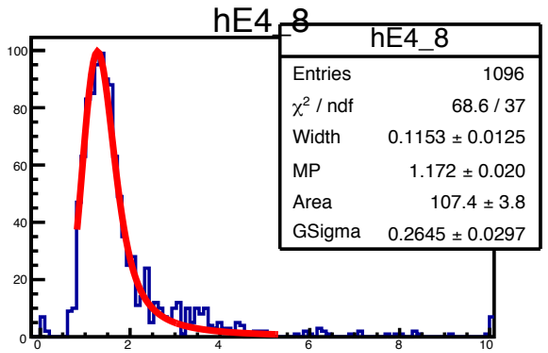
hE(FlagNumber)_(PlotNumber)

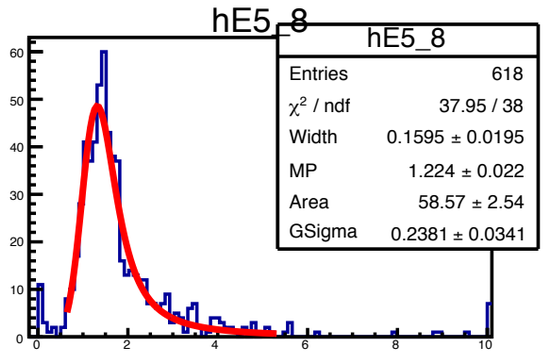
PlotNumber	DCV ch	PlotNumber	DCV ch
8	ch 16 + ch 17	12	ch 24 + ch 25
9	ch 18 + ch 19	13	ch 26 + ch 27
10	ch 20 + ch 21	14	ch 28 + ch 29
11	ch 22 + ch 23	15	ch 30 + ch 31

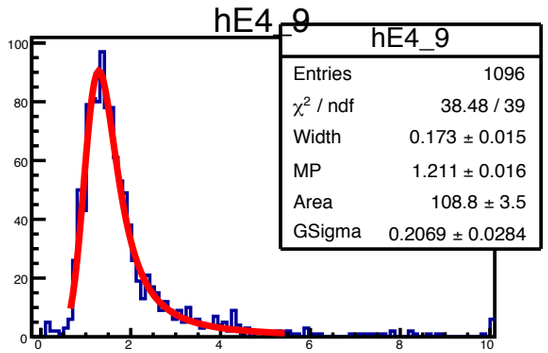
ch 16 : hE4_8	ch 17 : hE5_8	ch 18 : hE4_9	ch 19 : hE5_9
ch 20 : hE1_10	ch 21 : hE0_10	ch 22 : hE1_11	ch 23 : hE0_11
ch 24 : hE3_12	ch 25 : hE2_12	ch 26 : hE3_13	ch 27 : hE2_13
ch 28 : hE0_14	ch 29 : hE1_14	ch 30 : hE0_15	ch 31 : hE1_15

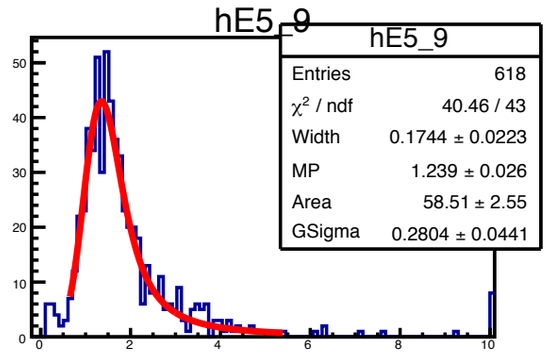
DCVcalib distribution, hE(FlagNumber)_(PlotNumber)

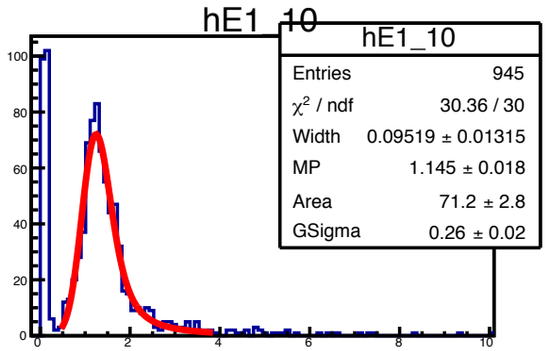
Convoluted Landau + Gauss Fit → Get the MP value

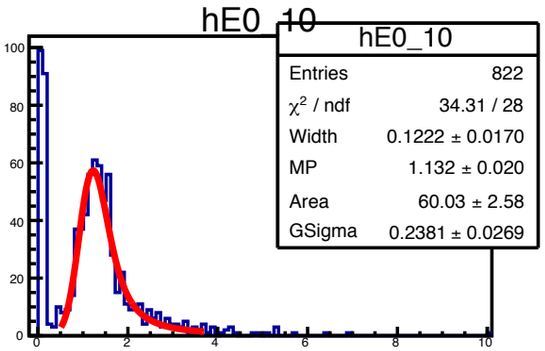


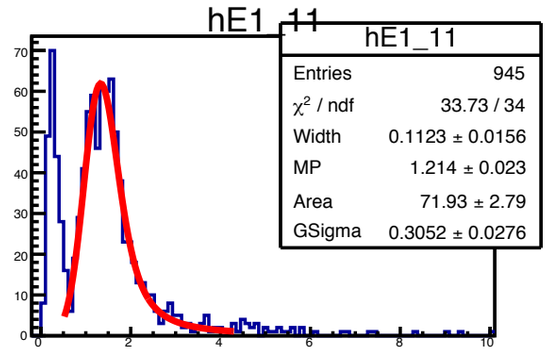


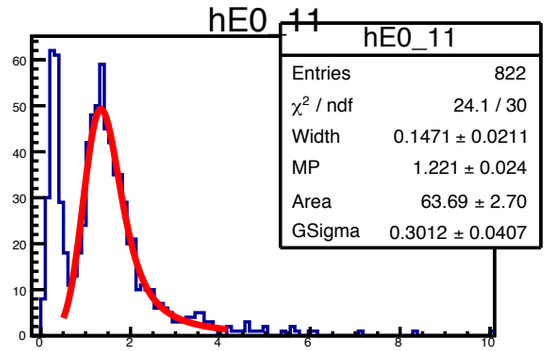


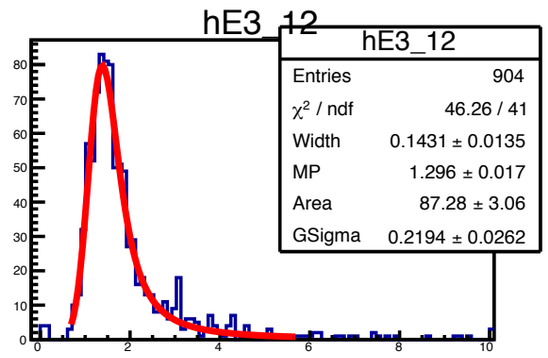


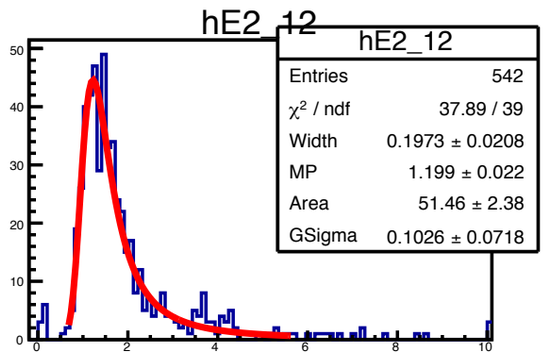


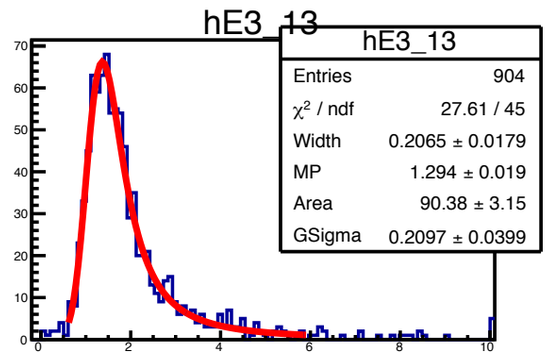


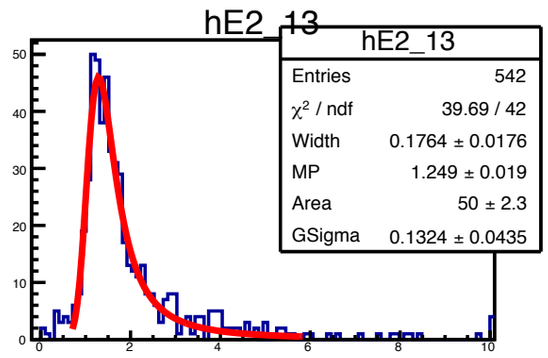


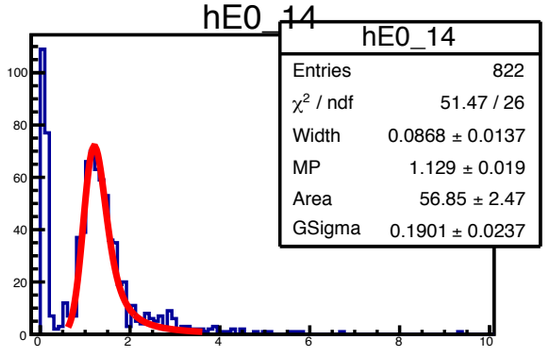


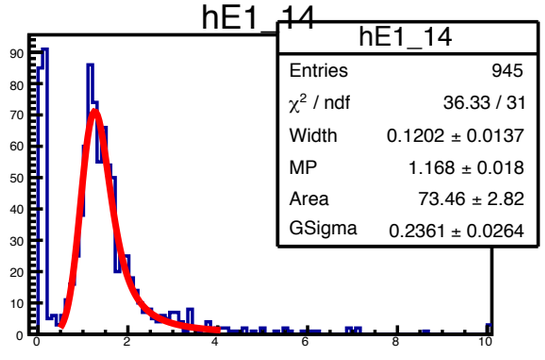


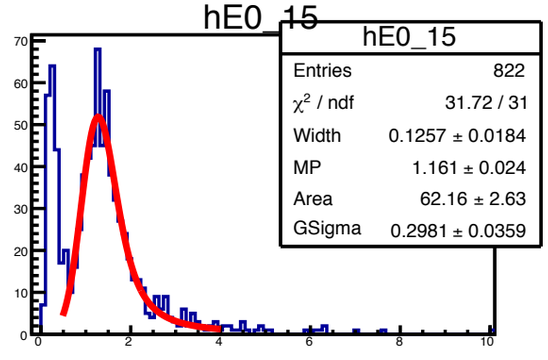


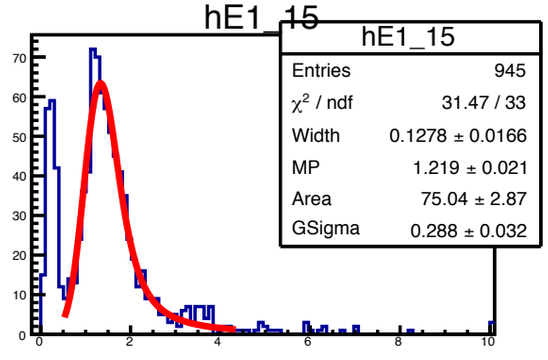












Events

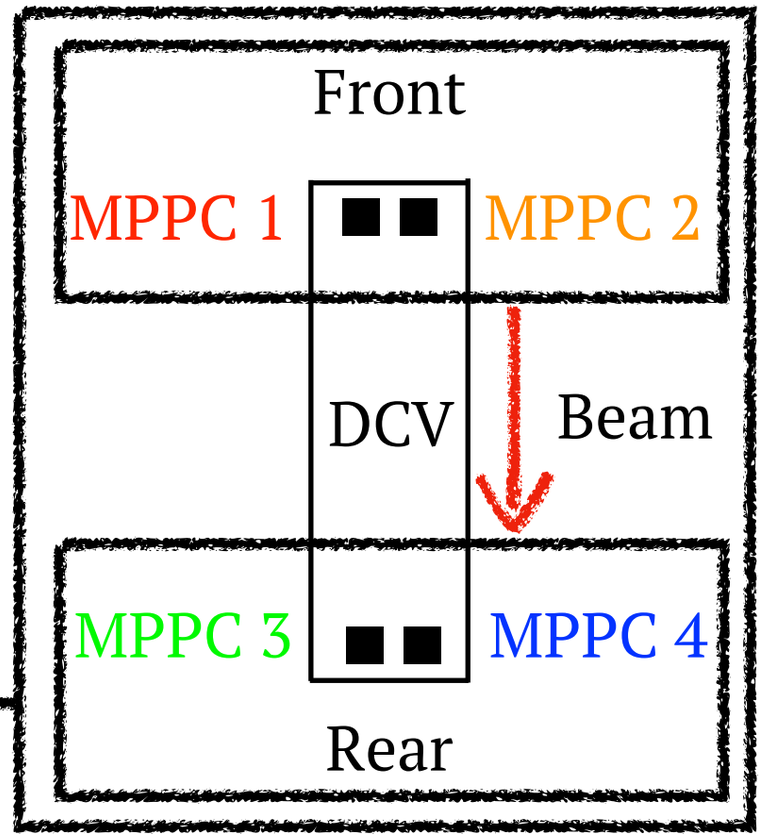
/home/had/hmkim/work/hmkim/run81/final_cal/dcv2_calib1.C → dcv2_output1.root

DCVcalib[MeV]

/home/had/hmkim/work/hmkim/run81/final_cal/dcv2_calib2.C

gain(MP value from hP)			
301.2	340.2	220.4	226.3
263.6	242.2	153.8	155.4
284.2	256.8	173.3	172.1
273.7	263.3	157.2	159.6

norm(MP value from hE)			
1.172	1.224	1.211	1.239
1.145	1.132	1.214	1.221
1.296	1.199	1.294	1.249
1.129	1.168	1.161	1.219



DCVcalib = DCVPeak / (gain*norm)

(Define 'DCVcalib' for calculating the DCVEne)

hES(FlagNumber)_(PlotNumber)
→ Fill(DCVcalib[4*ch] + DCVcalib[4*ch+1]
+DCVcalib[4*ch+2]+DCVcalib[4*ch+3])

- DCVPeak is normalized to DCVcalib by being divided gain and norm factor.
- For the energy deposit of one cosmic ray, 4 MPPCs are shared the light.
- Second, I summed the 4 MPPCs in the same module.

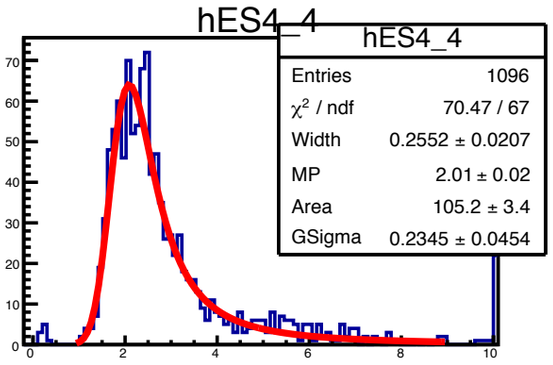
hES(FlagNumber)_(PlotNumber)

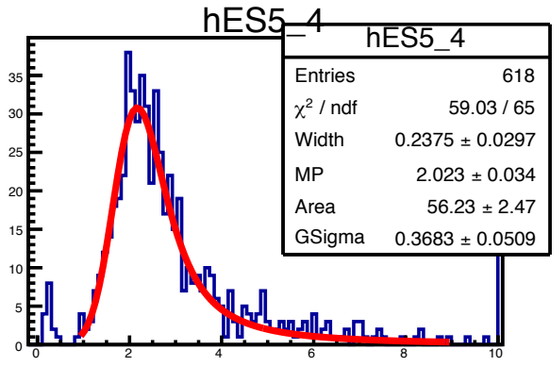
PlotNumber	DCV ch
4	ch 16 + ch 17 + ch 18 + ch 19
5	ch 20 + ch 21 + ch 22 + ch 23
6	ch 24 + ch 25 + ch 26 + ch 27
7	ch 28 + ch 29 + ch 30 + ch 31

ch 16 : hES4_4	ch 17 : hES5_4	ch 18 : hES4_4	ch 19 : hES5_4
ch 20 : hES1_5	ch 21 : hES0_5	ch 22 : hES1_5	ch 23 : hES0_5
ch 24 : hES3_6	ch 25 : hES2_6	ch 26 : hES3_6	ch 27 : hES2_6
ch 28 : hES0_7	ch 29 : hES1_7	ch 30 : hES0_7	ch 31 : hES1_7

DCVcalib distribution, hES(FlagNumber)_(PlotNumber)

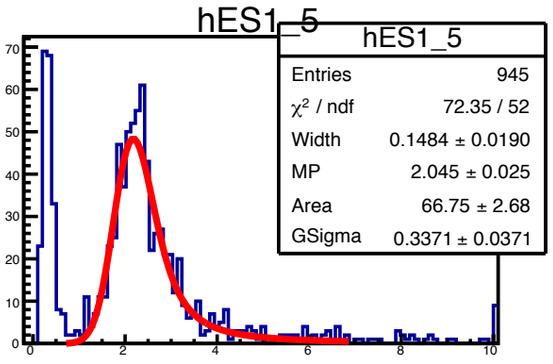
ch 16+17+18+19

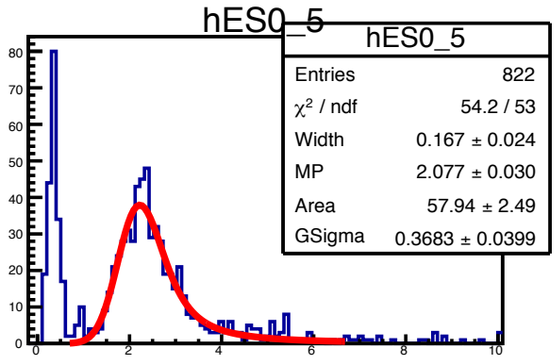




ch 16+17+18+19

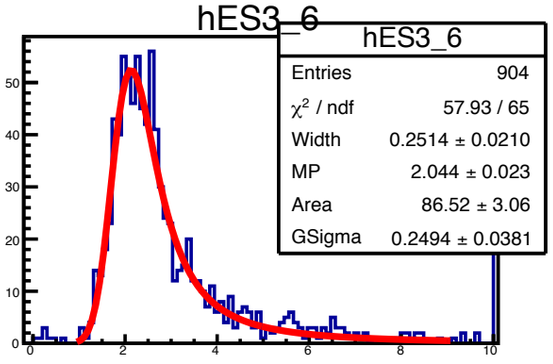
ch 20+21+22+23

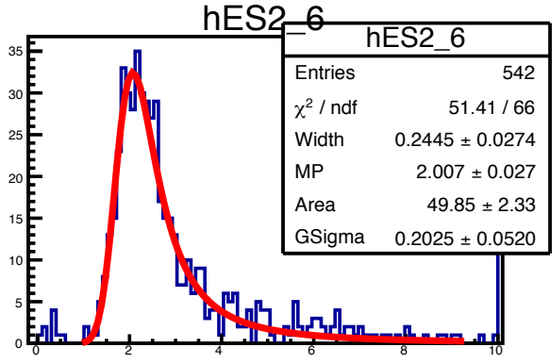




ch 20+21+22+23

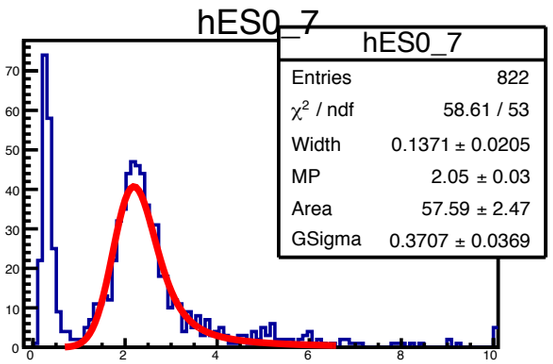
ch 24+25+26+27

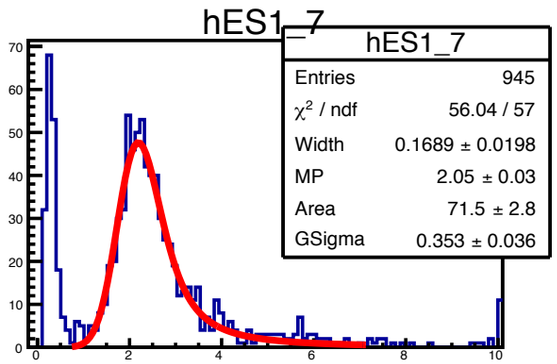




ch 24+25+26+27

ch 28+29+30+31





ch 28+29+30+31

Events

DCVcalib[MeV]

/home/had/hmkim/work/hmkim/run81/final_cal/dcv2_calib2.C → dcv2_output2.root

/home/had/hmkim/work/hmkim/run81/final_cal/dcv1_calib3.C

gain(MP value from hP)			
301.2	340.2	220.4	226.3
263.6	242.2	153.8	155.4
284.2	256.8	173.3	172.1
273.7	263.3	157.2	159.6

norm(MP value from hE)			
1.172	1.224	1.211	1.239
1.145	1.132	1.214	1.221
1.296	1.199	1.294	1.249
1.129	1.168	1.161	1.219

norm_module(MP value from hES)			
2.01	2.023	2.01	2.023
2.045	2.077	2.045	2.077
2.044	2.007	2.044	2.007
2.05	2.05	2.05	2.05

↓

DCVcalib = DCVPeak / (gain*norm*norm_module)

(Define ‘DCVcalib’ for calculating the DCVEne)

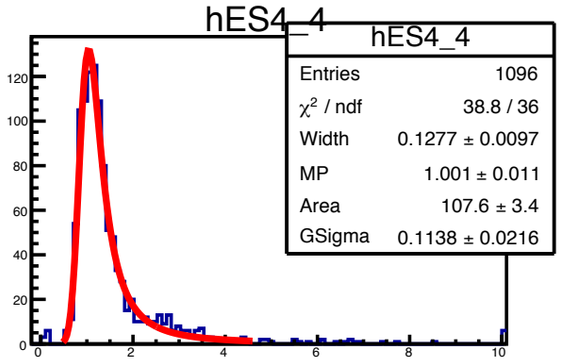
↓

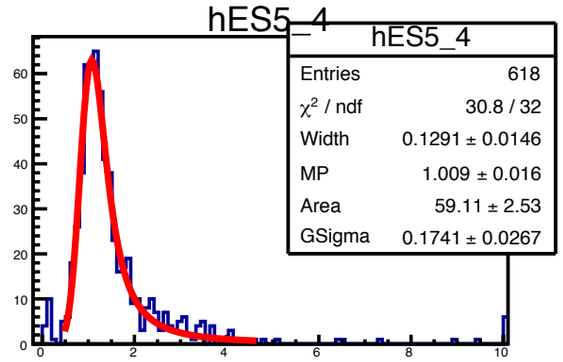
hES(FlagNumber)_(PlotNumber)
→ Fill(DCVcalib[4*ch] + DCVcalib[4*ch+1]
+DCVcalib[4*ch+2]+DCVcalib[4*ch+3])

- DCVPeak is normalized to DCVcalib by being divided gain and norm factor and norm_module factor

DCVcalib distribution, hES(FlagNumber)_(PlotNumber)

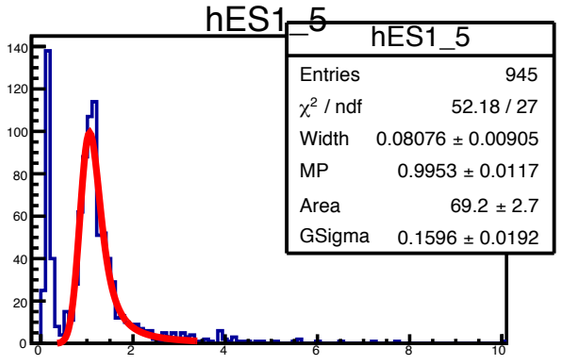
ch 16+17+18+19

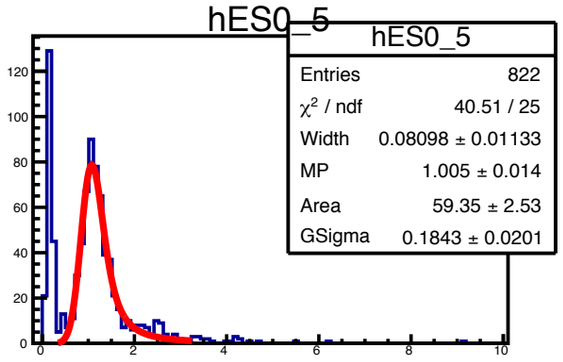




ch 16+17+18+19

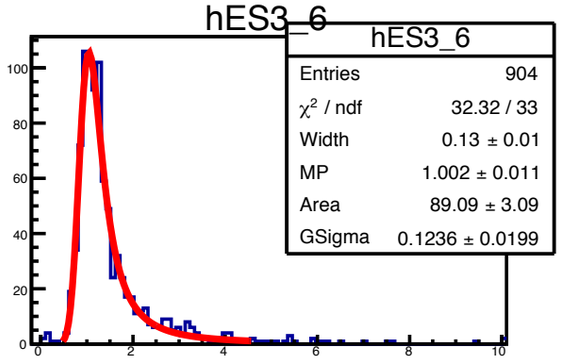
ch 20+21+22+23

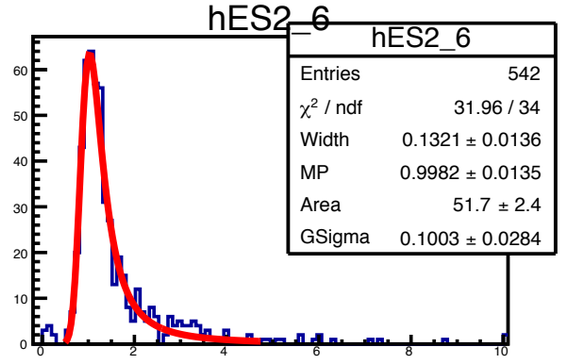




ch 20+21+22+23

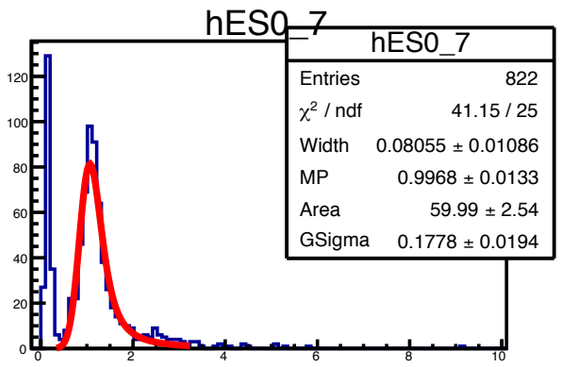
ch 24+25+26+27

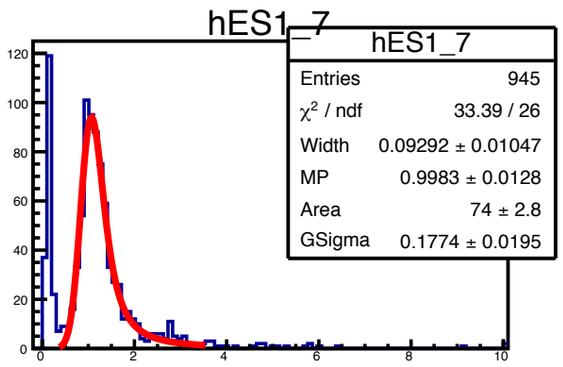




ch 24+25+26+27

ch 28+29+30+31





ch 28+29+30+31

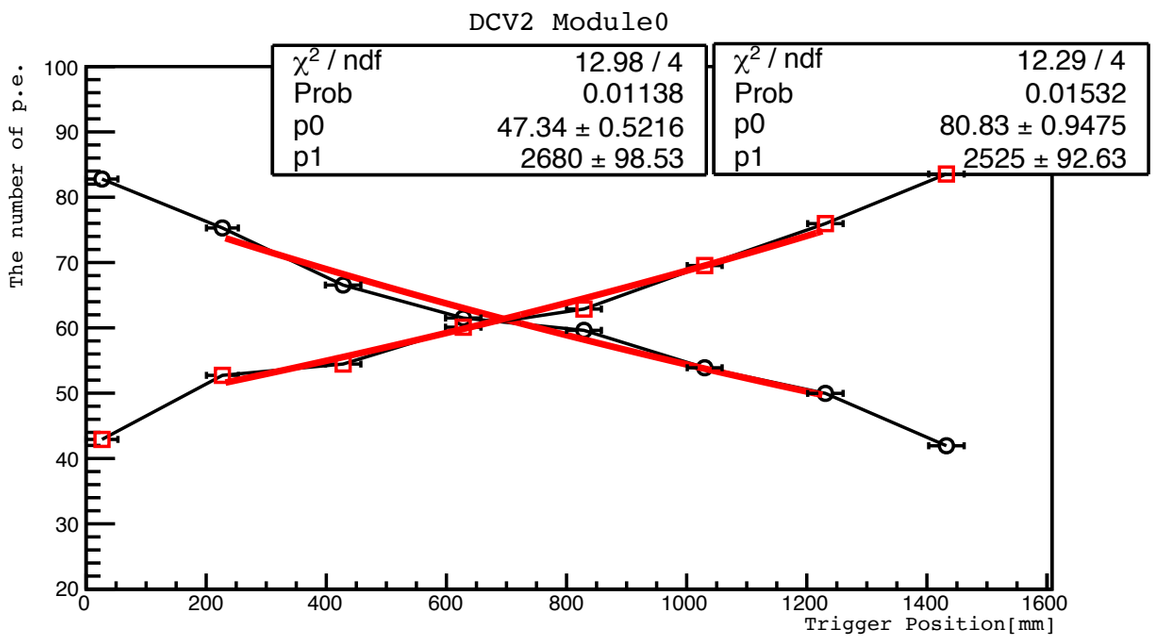
Events

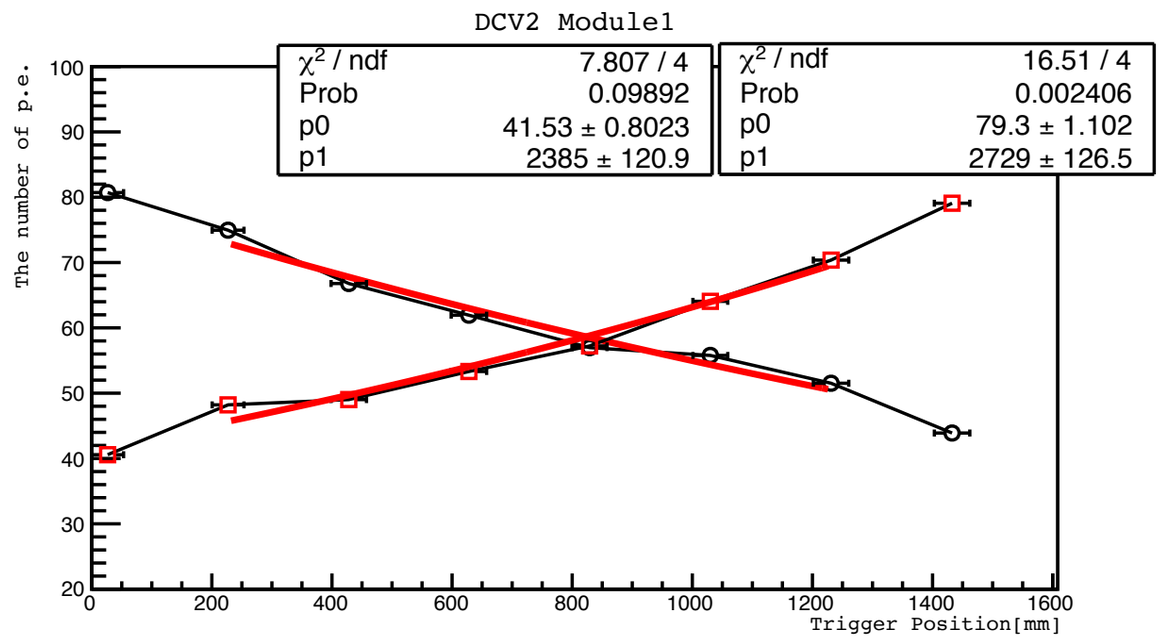
DCVcalib[MeV]

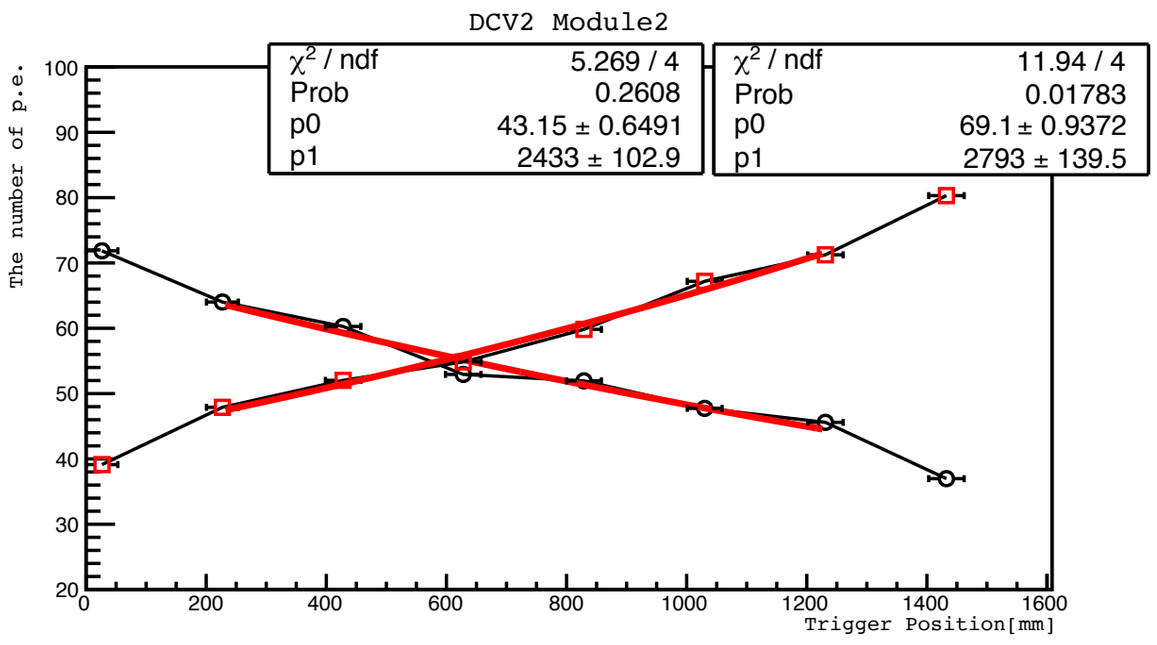
/home/had/hmkim/work/hmkim/run81/final_cal/dcv2_calib3.C → dcv2_output3.root

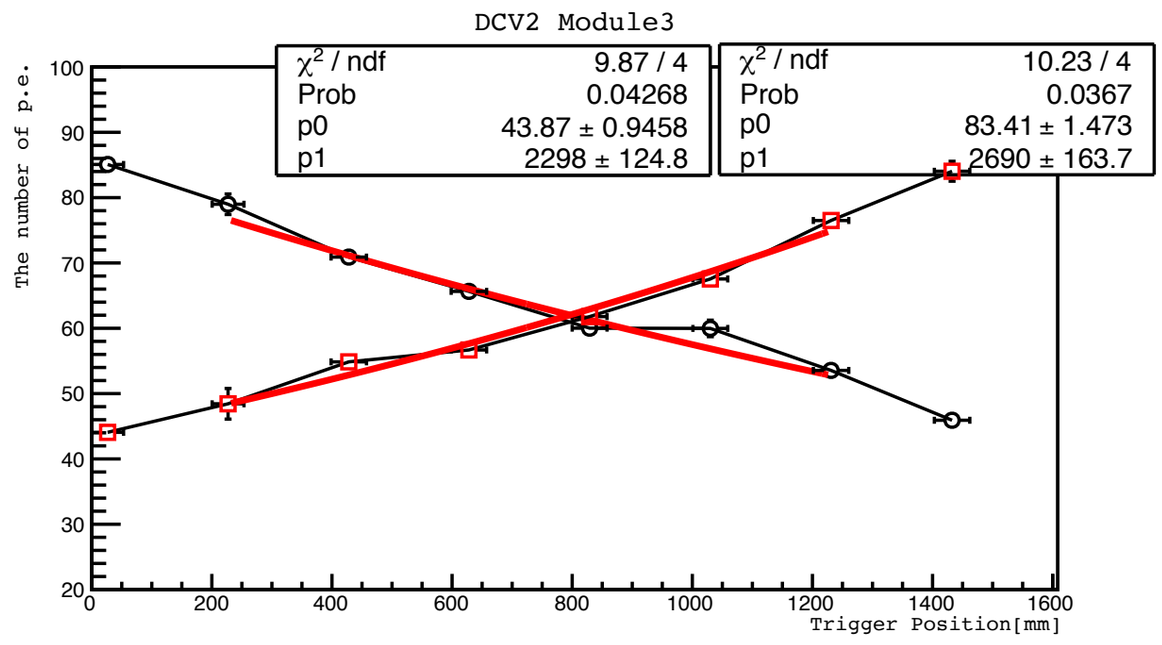
Attenuation effect

- fitting function = [0] * exp(±x / [1])
- fitting range 227 mm ~ 1231 mm



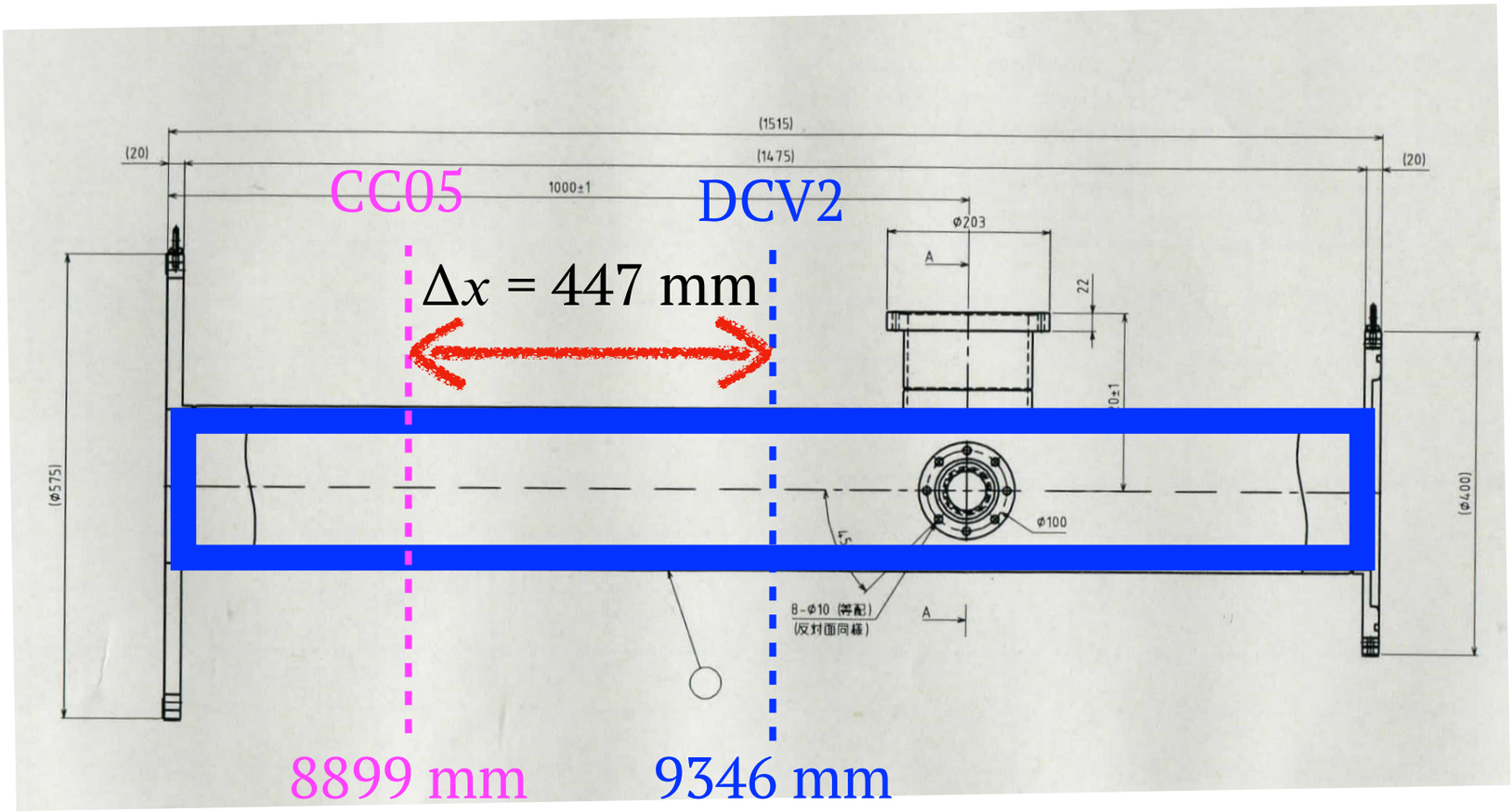


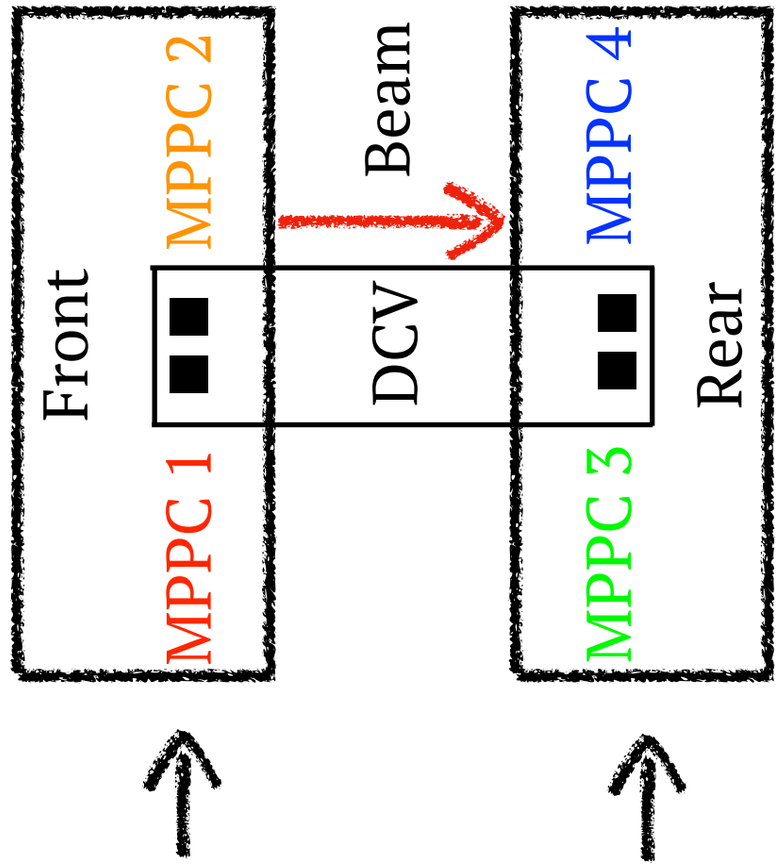




Attenuation Length λ_{avg} = **2567 mm**

Calculation of attenuation factor





att2_f
=1.19021

att2_r
=0.840185

```
lambda_dcv2 = 2567; // Attenuation Length of DCV2
pos_DCV2 = 9346; // Center position of DCV2
pos_CC05 = 8899; // Center position of CC05

att2_f = exp((pos_DCV2 - pos_CC05)/lambda_dcv2);
att2_r = exp(-(pos_DCV2 - pos_CC05)/lambda_dcv2);
```

/home/had/hmkim/work/hmkim/run81/final_cal/dcv2_calib3.C

att			
1.19021	1.19021	0.840185	0.840185
1.19021	1.19021	0.840185	0.840185
1.19021	1.19021	0.840185	0.840185
1.19021	1.19021	0.840185	0.840185

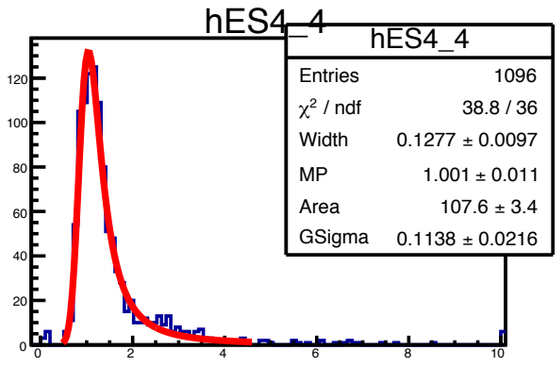
DCVcalib = DCVPeak / (gain*norm*norm_module)
DCVcalib_att = DCVcalib * att

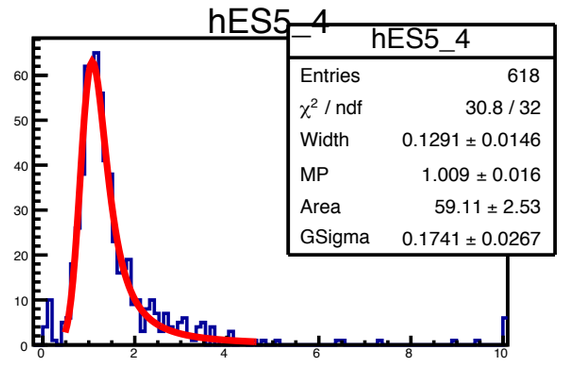
(Define ‘DCVcalib’ for calculating the DCVEne
and ‘DCVcalib_att’ for including attenuation effect)

hESF(FlagNumber)_(PlotNumber)
→ Fill(DCVcalib_att[4*ch] + DCVcalib_att[4*ch+1]
+DCVcalib_att[4*ch+2]+DCVcalib_att[4*ch+3])

DCVcalib distribution including att. effect, hESF(FlagNum)_(PlotNum)

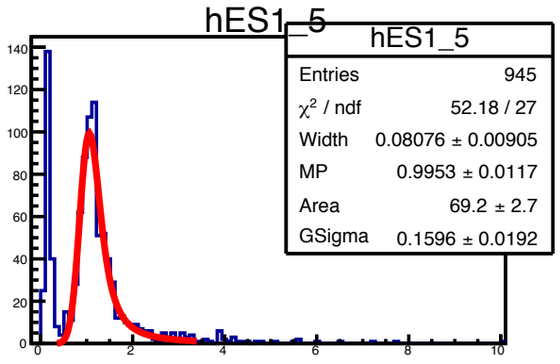
ch 16+17+18+19

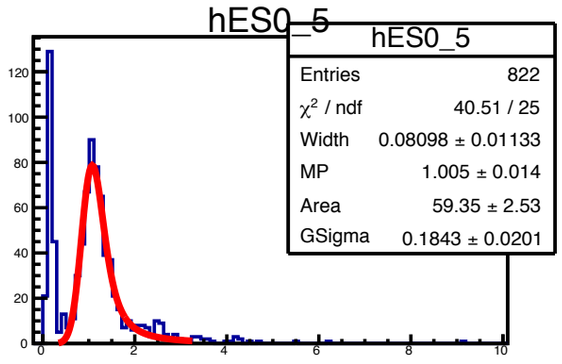




ch 16+17+18+19

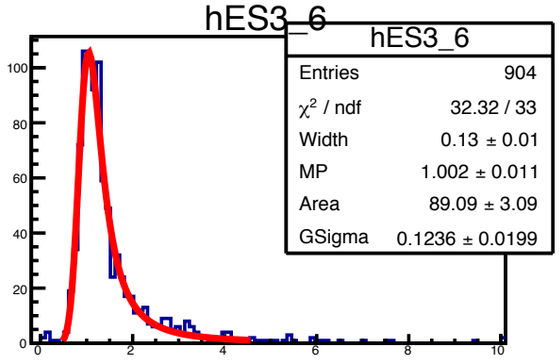
ch 20+21+22+23

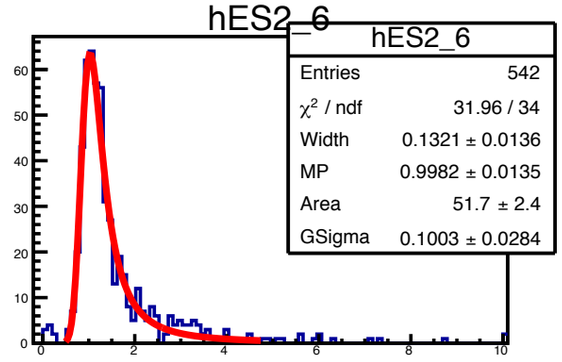




ch 20+21+22+23

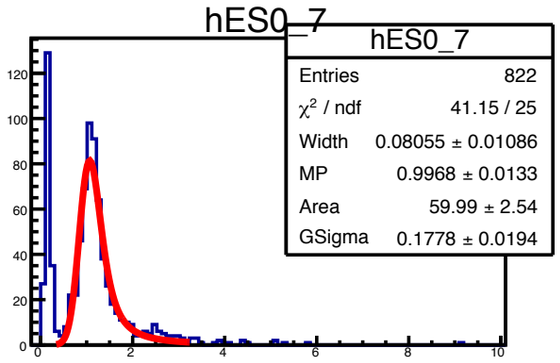
ch 24+25+26+27

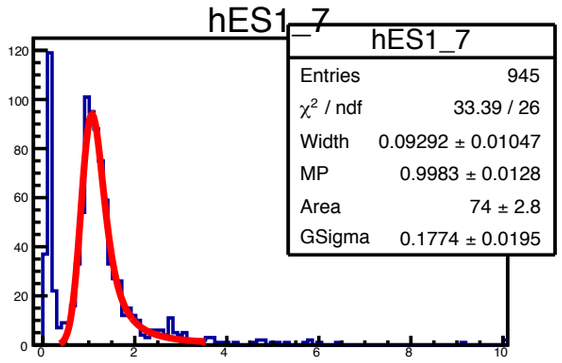




ch 24+25+26+27

ch 28+29+30+31





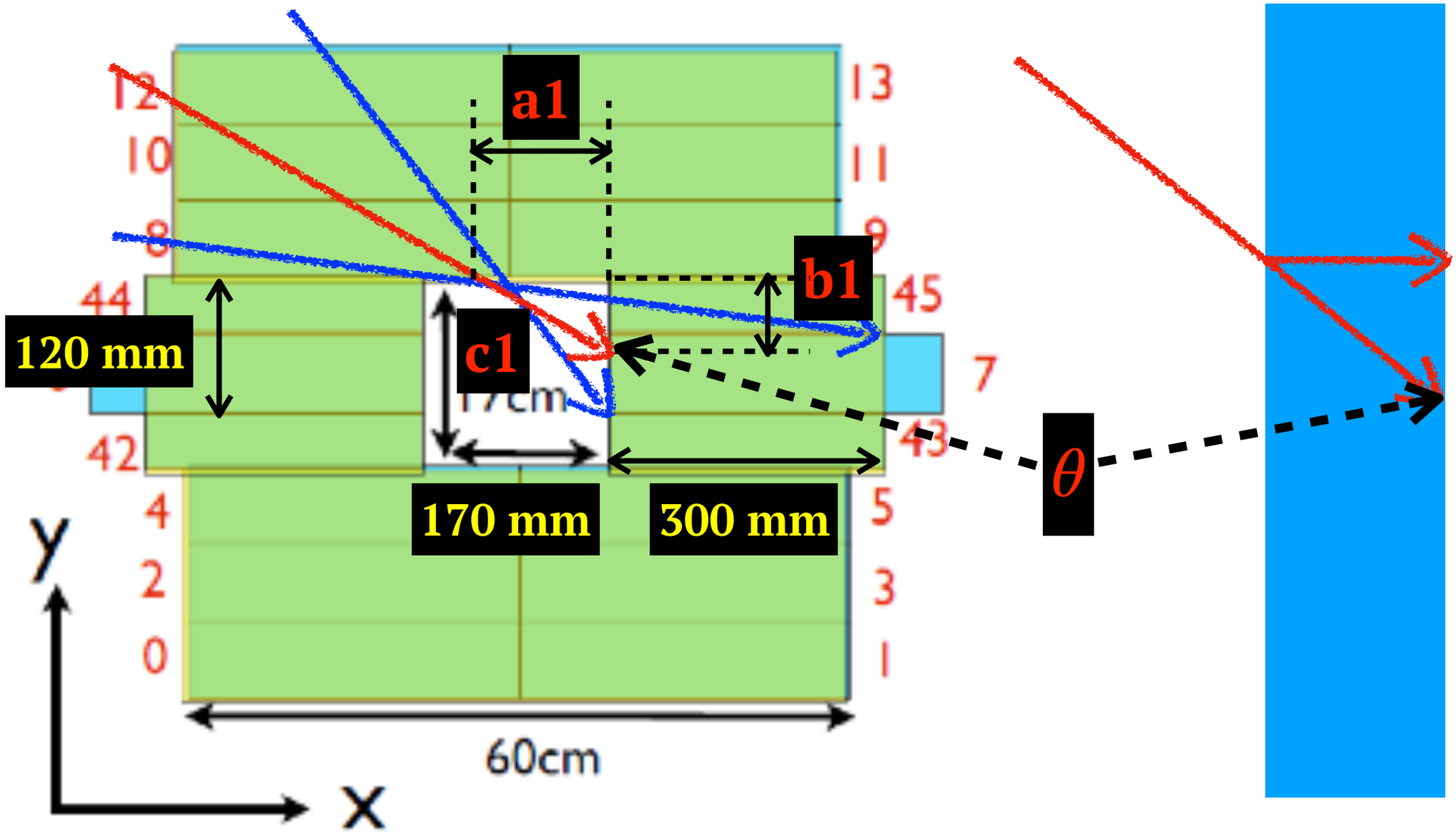
ch 28+29+30+31

Events

DCVcalib_att[MeV]

/home/had/hmkim/work/hmkim/run81/final_cal/dcv2_calib3.C → dcv2_output3.root

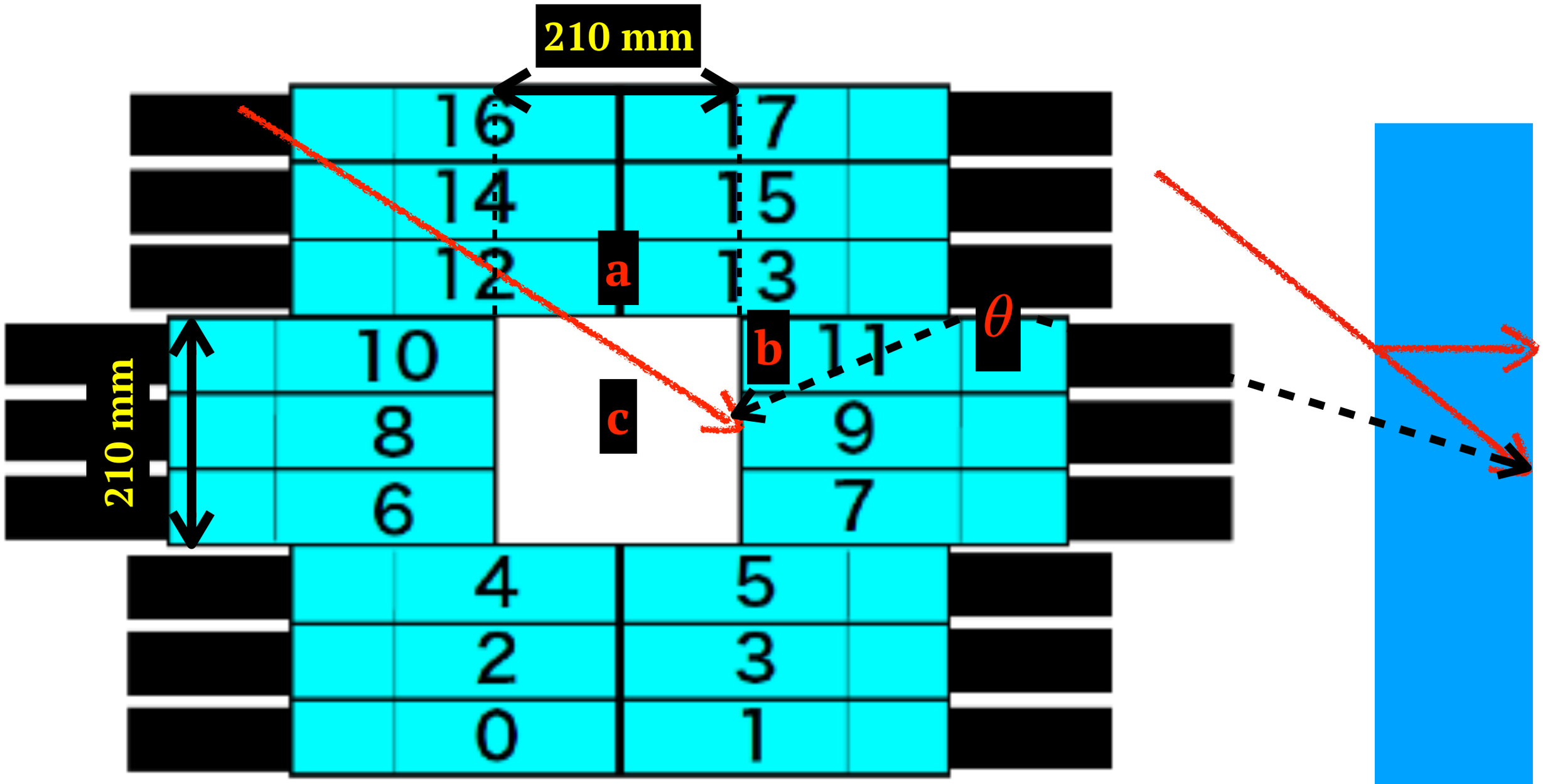
Path length(DCV1)




```
float a1 = 170*3/4;
float x = 5*17/317;
float b1 = (x+120)/2;
float c1 = sqrt((a1*a1)+(b1*b1));
float sin1 = a1/c1;
```

sin1 : 0.904172

Path length(DCV2)




```
float a2 = 210*3/4;
float b2 = 210/2;
float c2 = sqrt((a2*a2)+(b2*b2));

float sin2 = a2/c2;
```

sin2 : 0.831235

Total calculation the calibration factors including path length

/home/had/hmkim/work/hmkim/run81/final_cal/out_cal/make_calib_factor.C

For DCV1

gain(MP value from hP)			
359.7	402.8	261.2	299.8
328.5	282.1	258.7	206
327.4	417.2	244.1	294.5
371.1	370.7	227.6	244.1

(for MPPC Single ch)

norm(MP value from hE)			
1.291	1.349	1.273	1.325
1.179	1.202	1.185	1.181
1.316	1.324	1.334	1.338
1.178	1.195	1.167	1.182

(for Front(2 ch) and Rear(2 ch))

norm_module(MP value from hES)			
2.01	2.083	2.01	2.083
2.049	2.068	2.049	2.068
2.012	2.018	2.012	2.018
2.044	2.05	2.044	2.05

(for Module(4 ch))

att			
1.12691	1.12691	0.88738	0.88738
1.12691	1.12691	0.88738	0.88738
1.12691	1.12691	0.88738	0.88738
1.12691	1.12691	0.88738	0.88738

path_length			
0.904172	0.904172	0.904172	0.904172
1	1	1	1
0.904172	0.904172	0.904172	0.904172
1	1	1	1

calib_factor = att/(gain * norm * norm_module*path_length)

Total calculation the calibration factors including path length

/home/had/hmkim/work/hmkim/run81/final_cal/out_cal/make_calib_factor.C

For DCV2

gain(MP value from hP)			
301.2	340.2	220.4	226.3
263.6	242.2	153.8	155.4
284.2	256.8	173.3	172.1
273.7	263.3	157.2	159.6

(for MPPC Single ch)

norm(MP value from hE)			
1.172	1.224	1.211	1.239
1.145	1.132	1.214	1.221
1.296	1.199	1.294	1.249
1.129	1.168	1.161	1.219

(for Front(2 ch) and Rear(2 ch))

norm_module(MP value from hES)			
2.01	2.023	2.01	2.023
2.045	2.077	2.045	2.077
2.044	2.007	2.044	2.007
2.05	2.05	2.05	2.05

(for Module(4 ch))

att			
1.19021	1.19021	0.840185	0.840185
1.19021	1.19021	0.840185	0.840185
1.19021	1.19021	0.840185	0.840185
1.19021	1.19021	0.840185	0.840185

path_length			
0.0.831235	0.0.831235	0.0.831235	0.0.831235
1	1	1	1
0.0.831235	0.0.831235	0.0.831235	0.0.831235
1	1	1	1

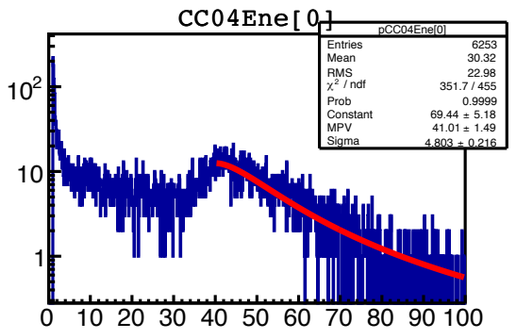
calib_factor = att/(gain * norm * norm_module*path_length)

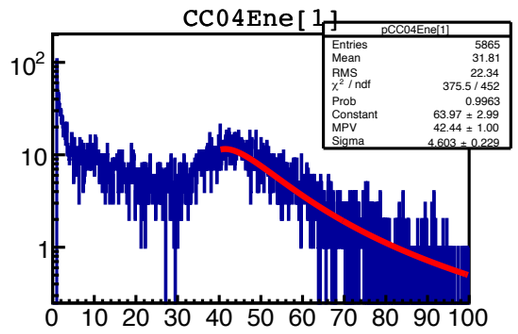
DCV	Mod.	Ch.	Calib_Factor	DCV	Mod.	Ch.	Calib_Factor
1	0	0	**0.00133529**	2	0	16	**0.00201801**
		1	**0.00110115**			17	**0.00169977**
		2	**0.00146846**			18	**0.00188408**
		3	**0.0011861**			19	**0.00178197**
	1	4	**0.00142003**		1	20	**0.00192833**
		5	**0.00160706**			21	**0.0020901**
		6	**0.00141271**			22	**0.00220042**
		7	**0.00176377**			23	**0.00213192**
	2	8	**0.00143773**		2	24	**0.00190192**
		9	**0.00111811**			25	**0.00231707**
		10	**0.00149798**			26	**0.00220515**
		11	**0.00123423**			27	**0.00234294**
	3	12	**0.00126117**		3	28	**0.0018789**
		13	**0.00124093**			29	**0.00188789**
		14	**0.0016345**			30	**0.00224562**
		15	**0.00150027**			31	**0.00210661**

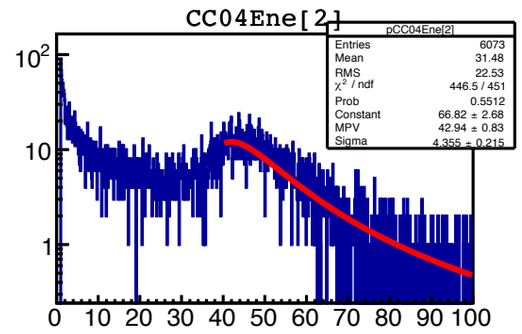
BACK UP

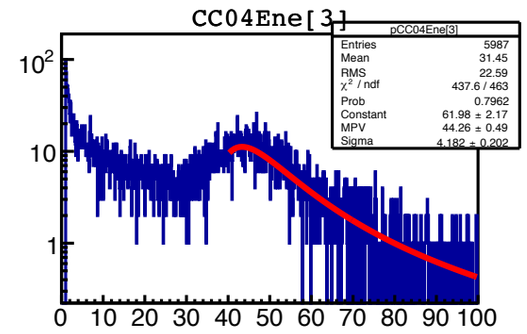
Setting the CC04Ene threshold

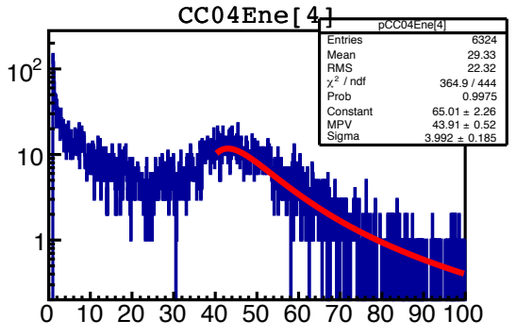
/home/had/hmkim/work/hmkim/run81/CC04thre.C → CC04thre_30776_30787.root
Fitting Function : Landau, CC04thre = CC04MPV × 0.7

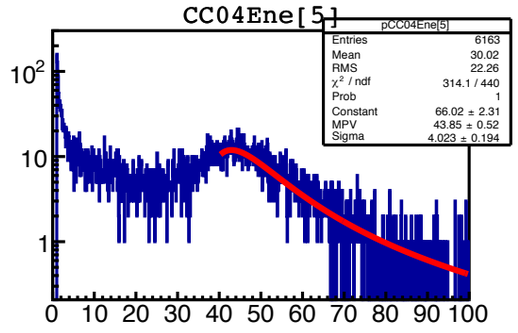


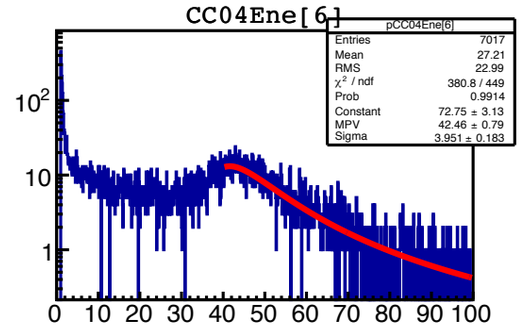


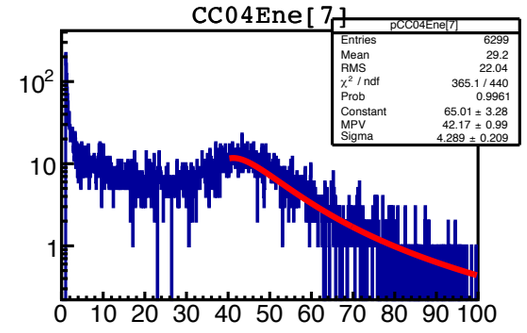


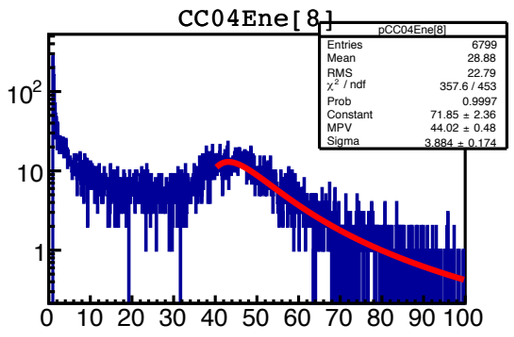


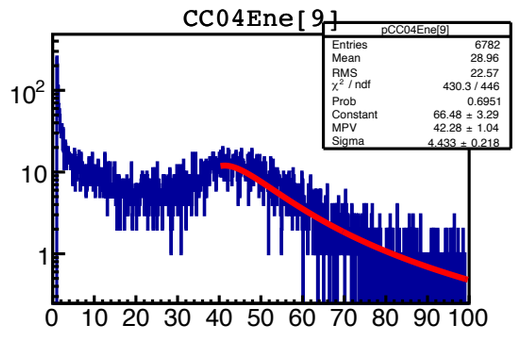


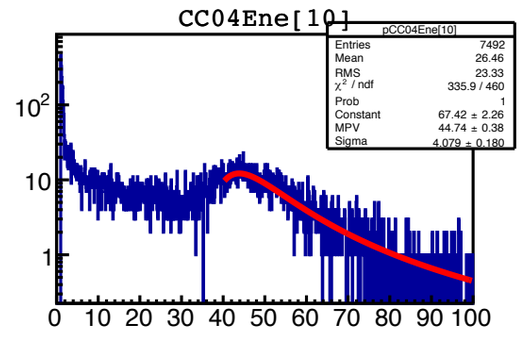


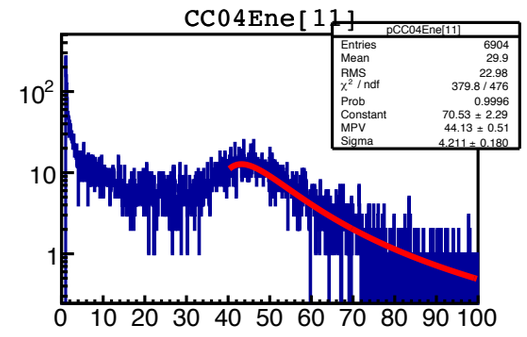


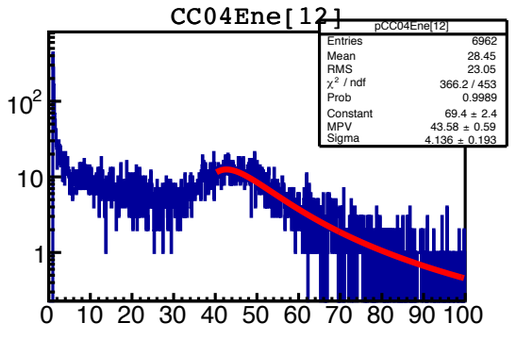


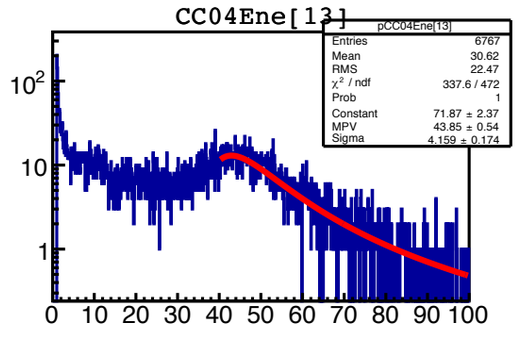


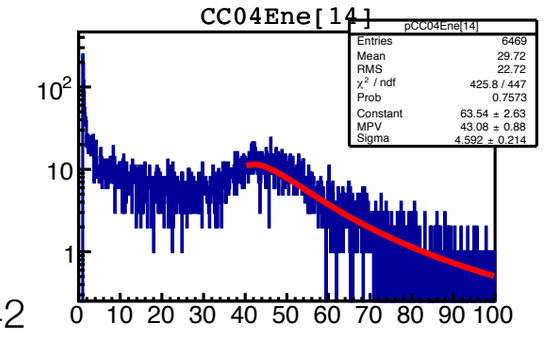


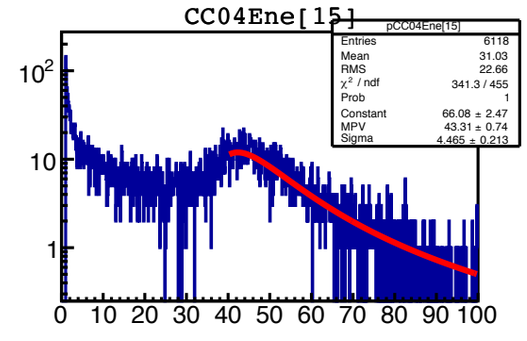








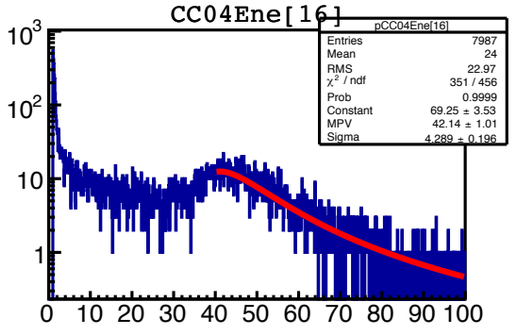


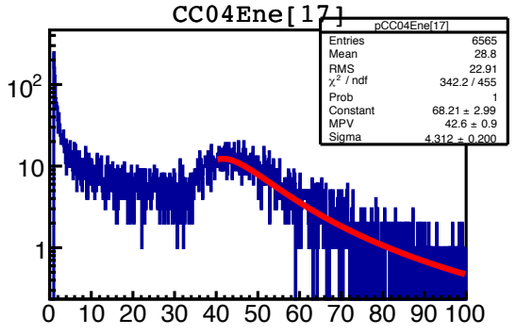


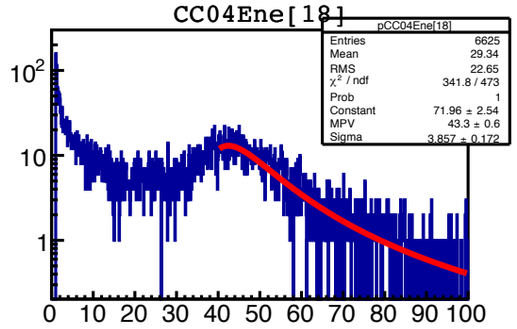
Setting the CC04Ene threshold

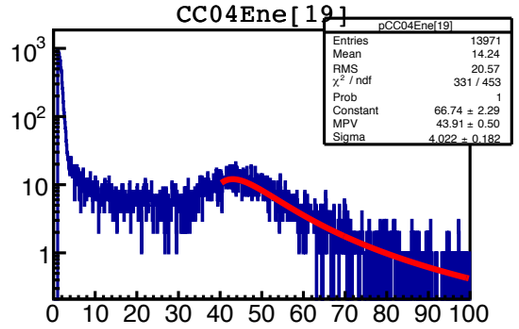
/home/had/hmkim/work/hmkim/run81/CC04thre.C → CC04thre_30776_30787.root

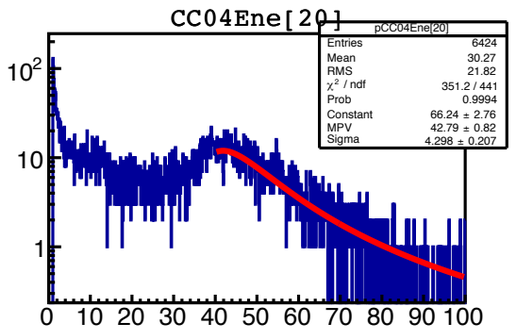
Fitting Function : Landau, CC04thre = CC04MPV × 0.7

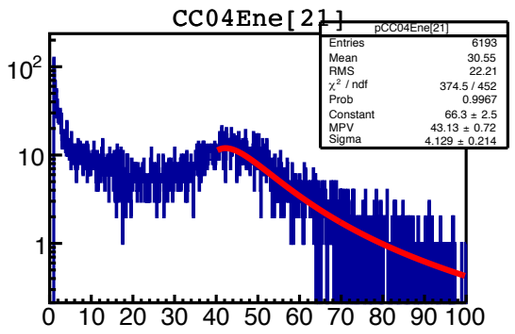


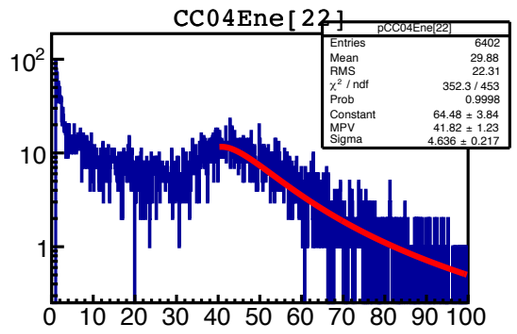


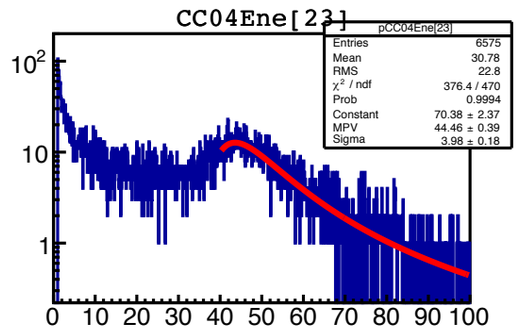


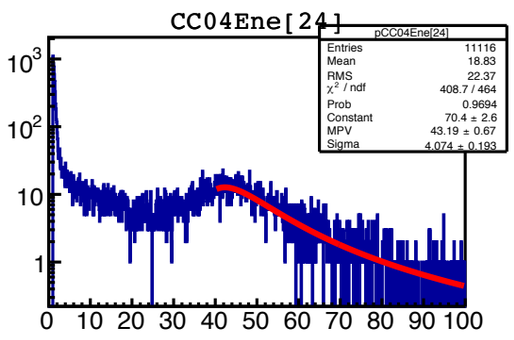


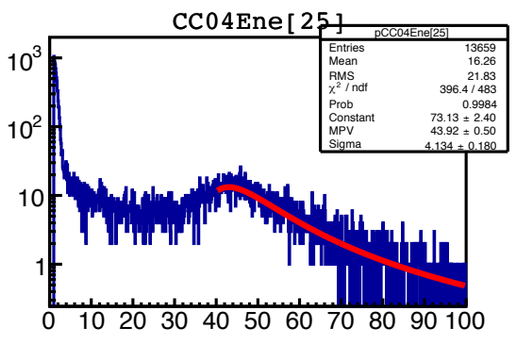


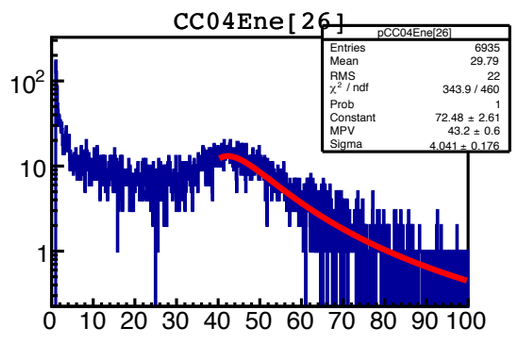


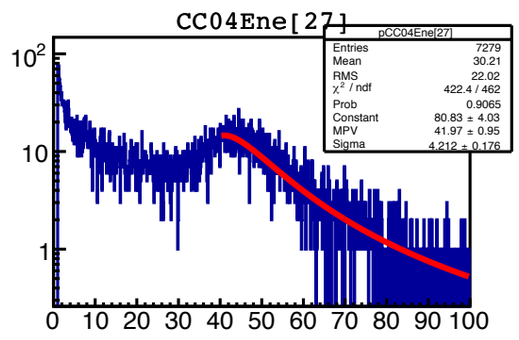


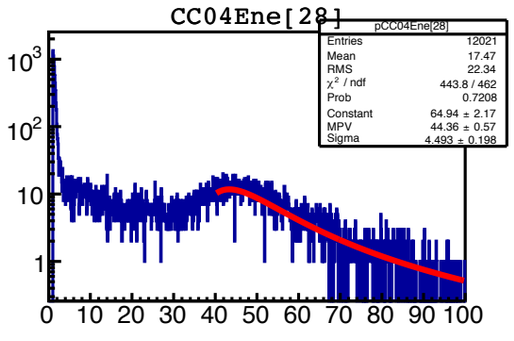


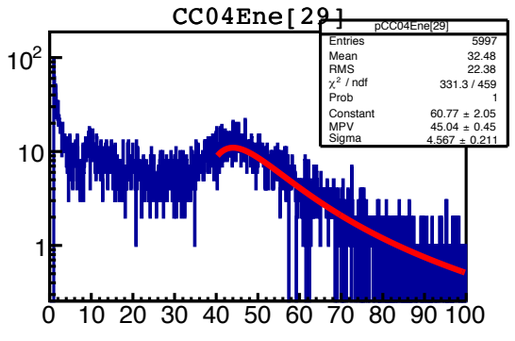


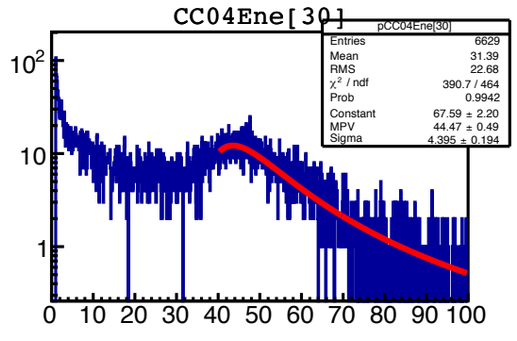


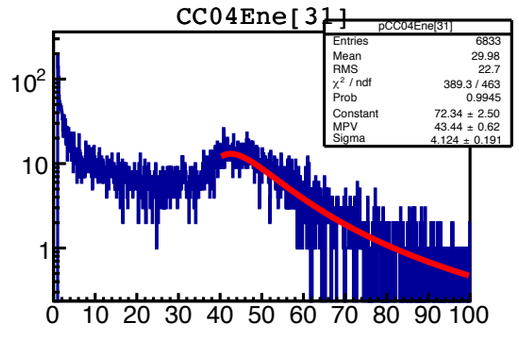






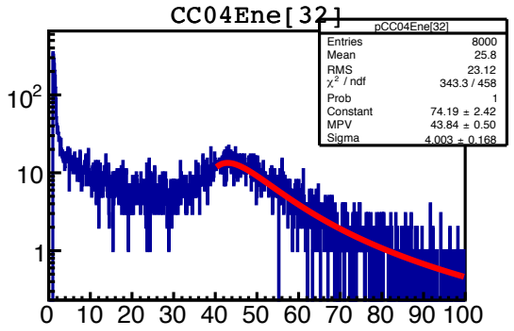


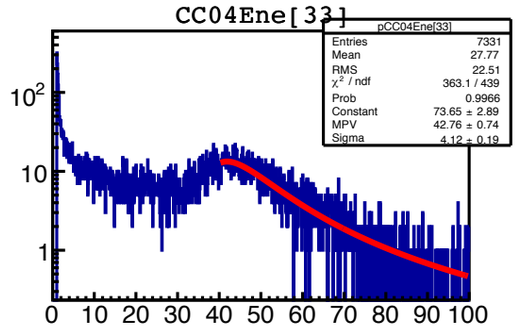


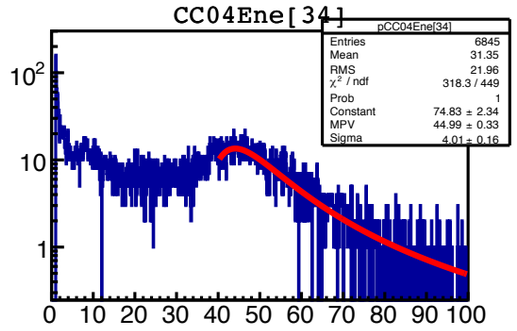


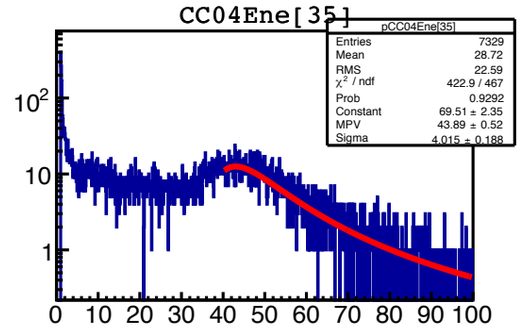
Setting the CC04Ene threshold

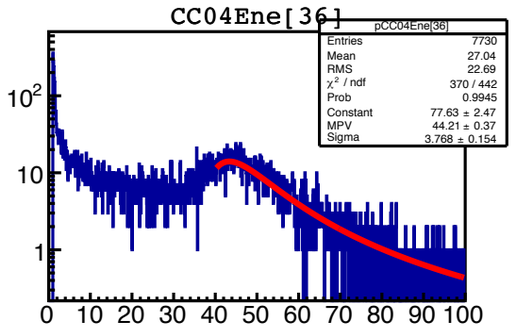
/home/had/hmkim/work/hmkim/run81/CC04thre.C → CC04thre_30776_30787.root
Fitting Function : Landau, CC04thre = CC04MPV × 0.7

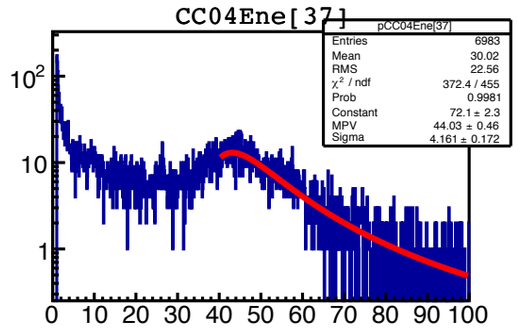


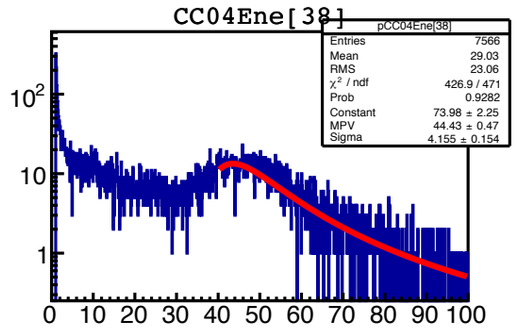


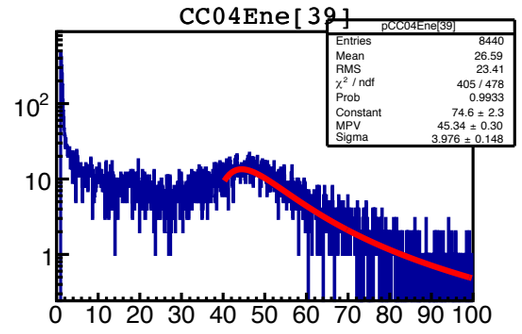


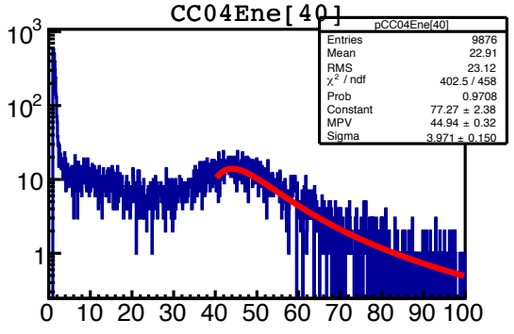


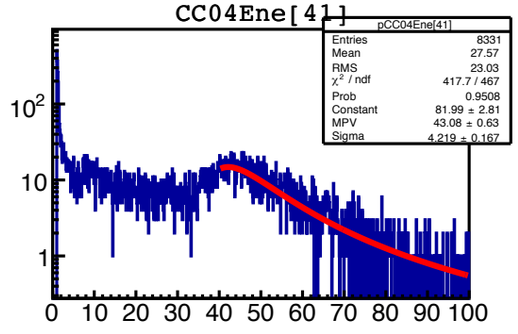


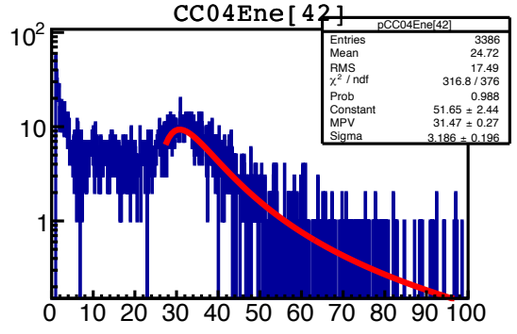


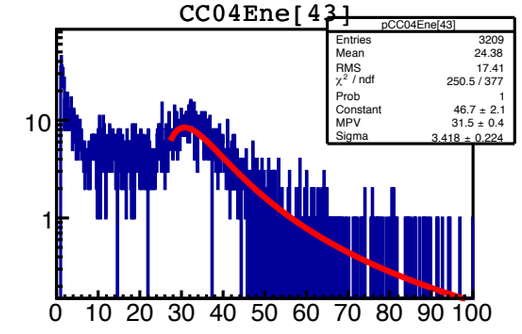


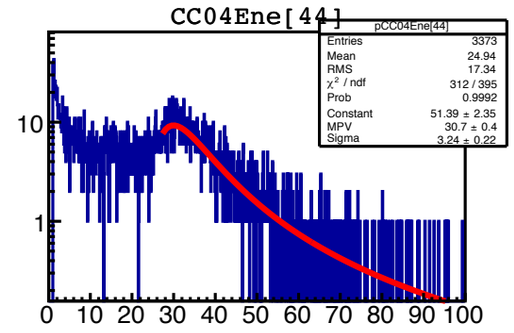


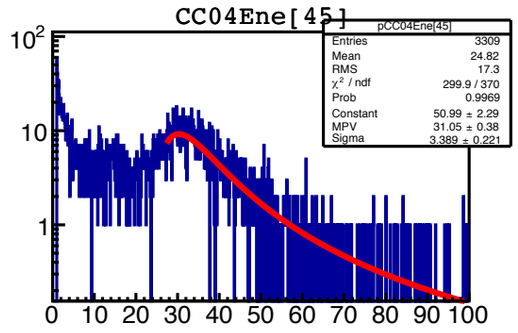


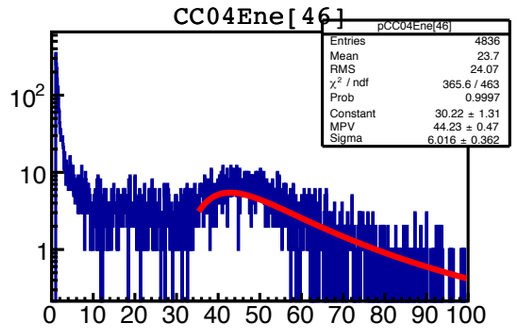


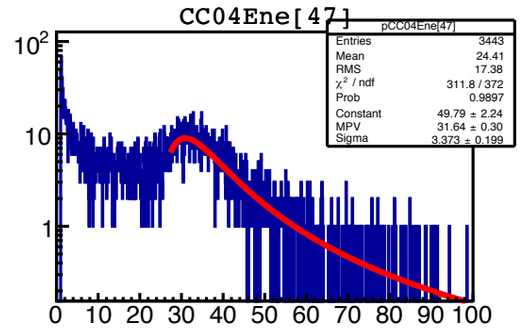






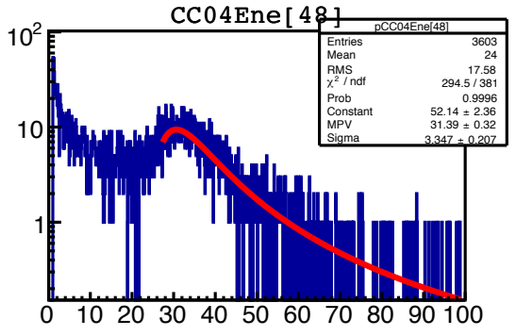


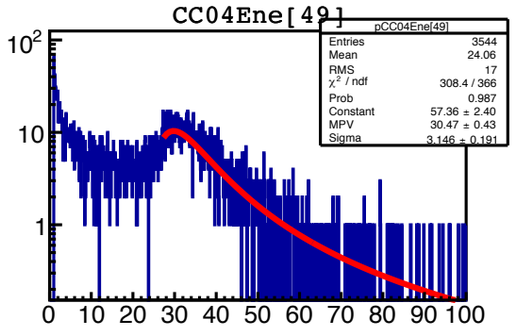


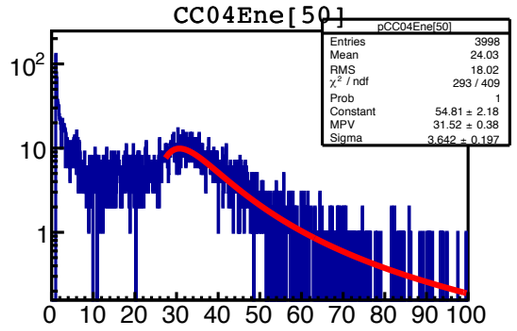


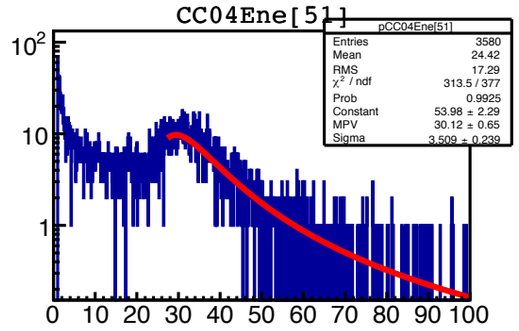
Setting the CC04Ene threshold

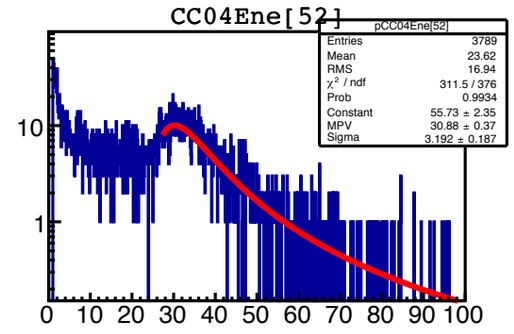
/home/had/hmkim/work/hmkim/run81/CC04thre.C → CC04thre_30776_30787.root
Fitting Function : Landau, CC04thre = CC04MPV × 0.7

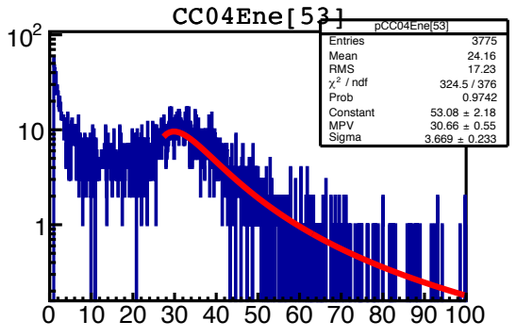


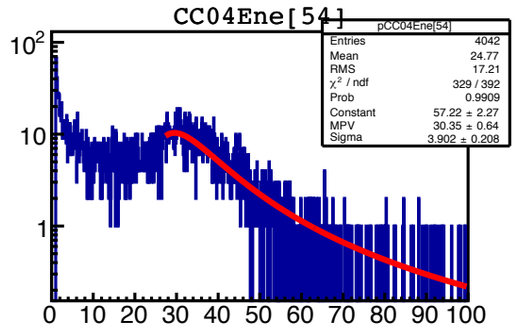


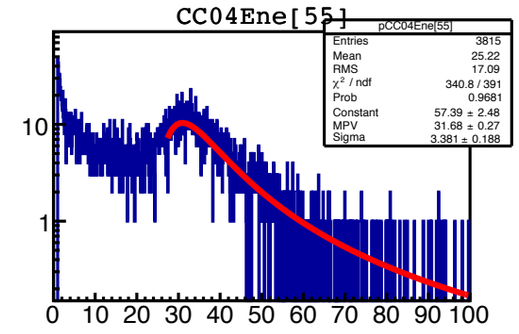


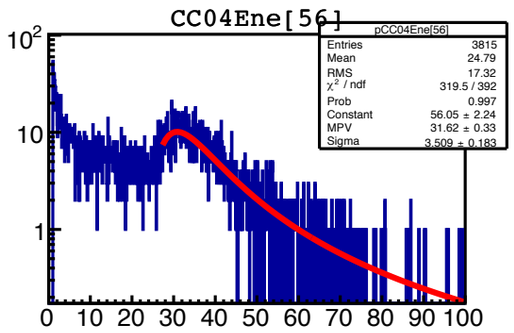


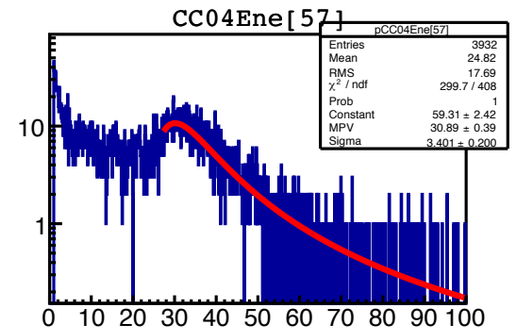


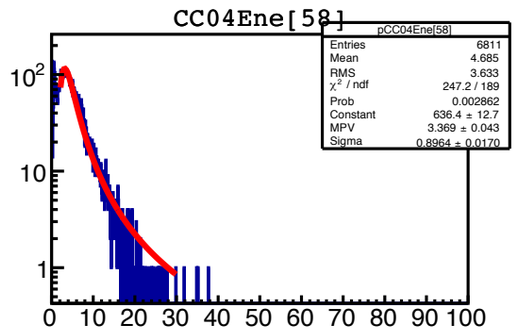


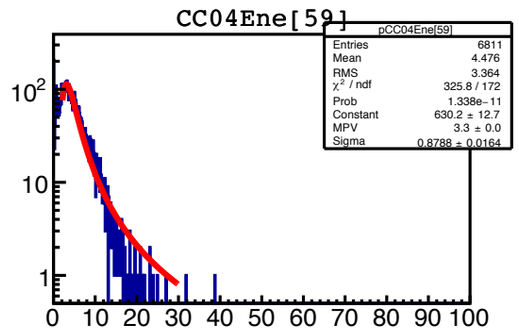


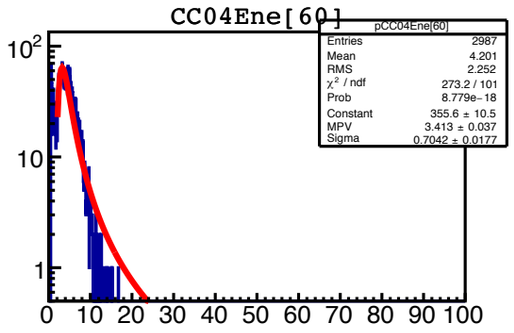


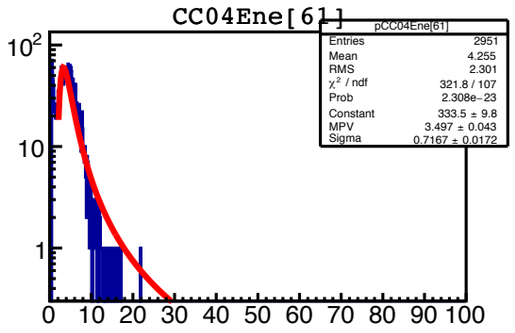


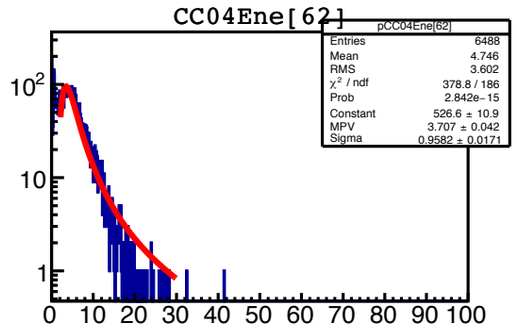


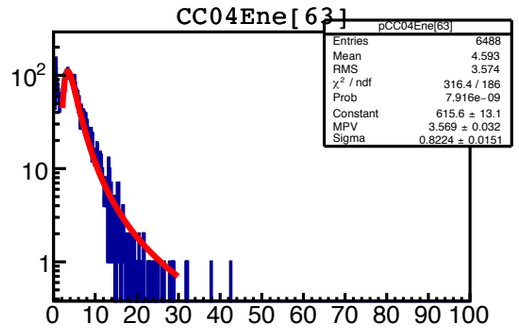






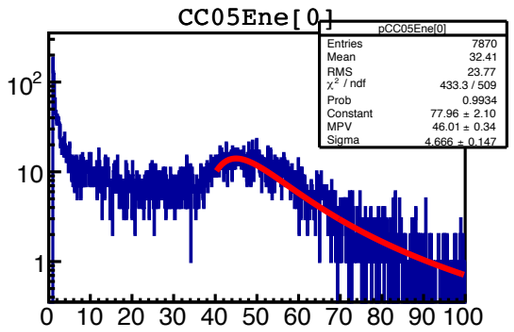


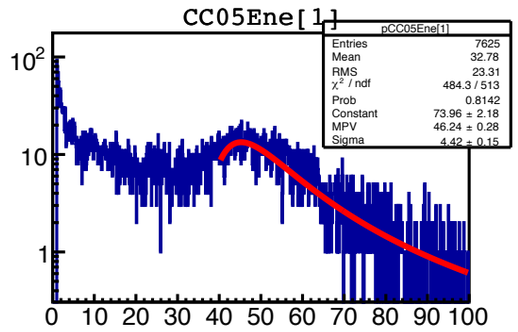


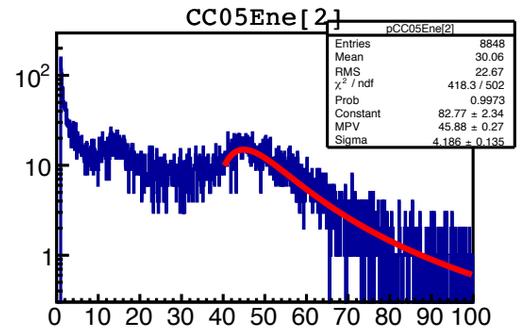


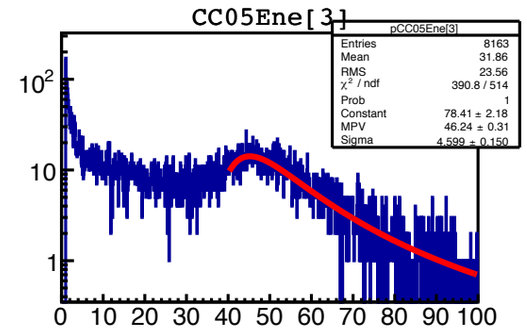
Setting the CC05Ene threshold

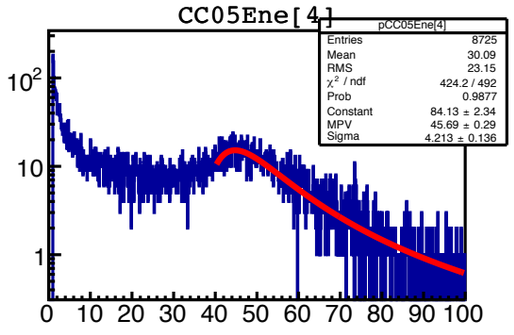
/home/had/hmkim/work/hmkim/run81/CC05thre.C → CC05thre_30776_30787.root
Fitting Function : Landau, CC05thre = CC05MPV × 0.7

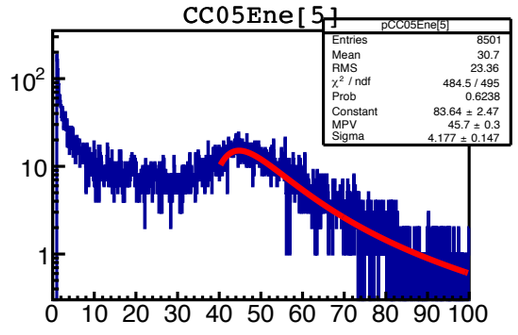


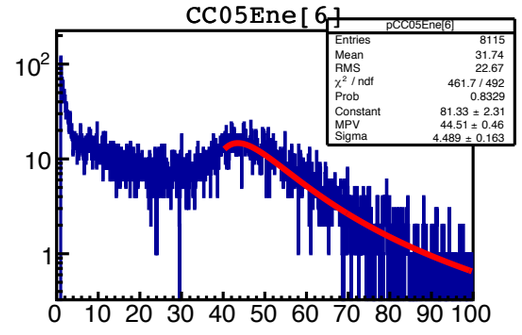


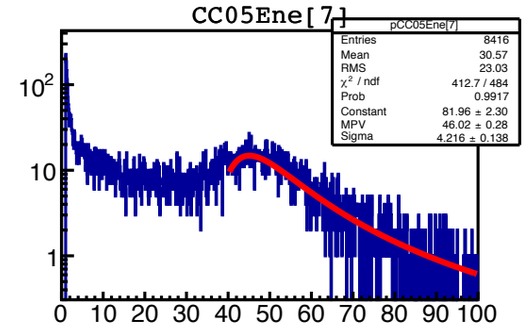


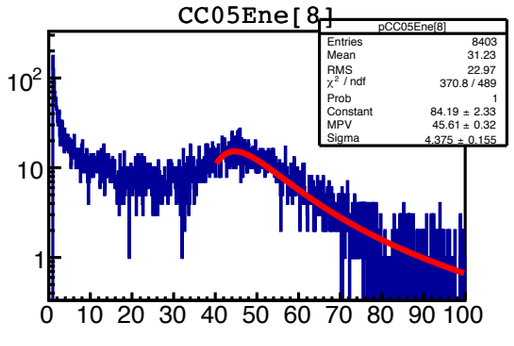


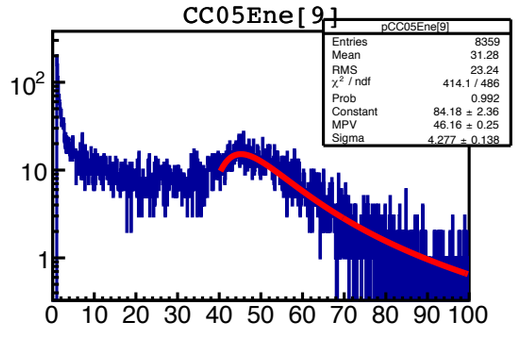


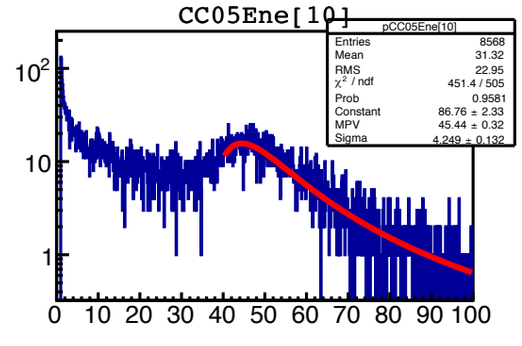


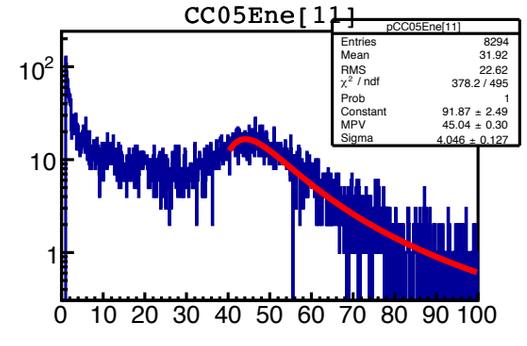


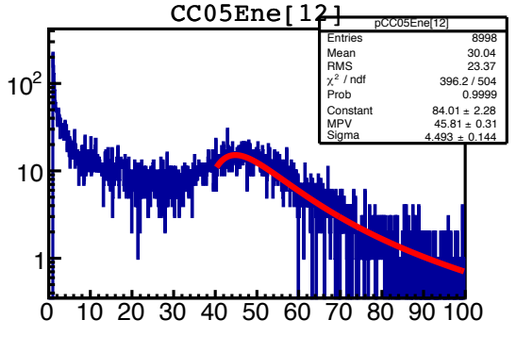


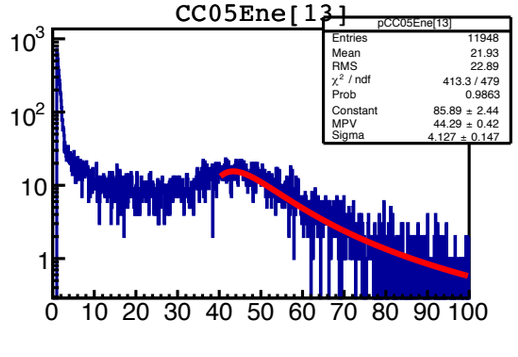


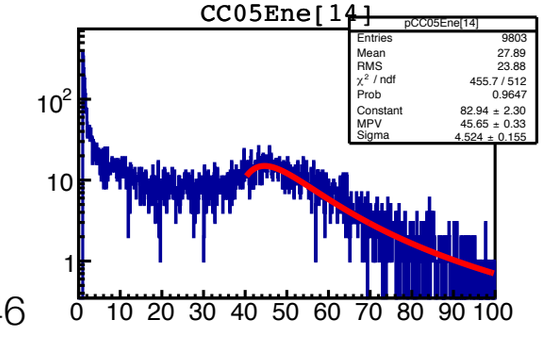


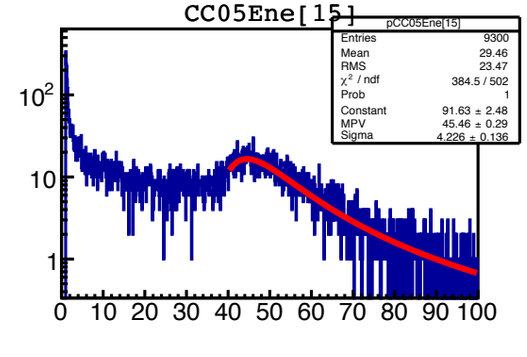








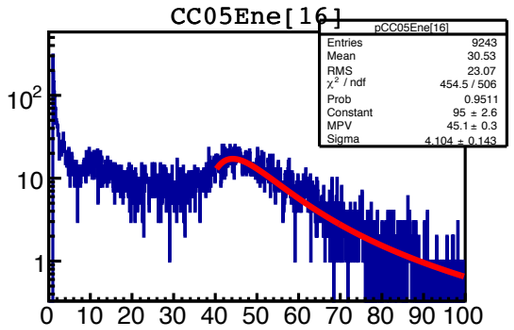


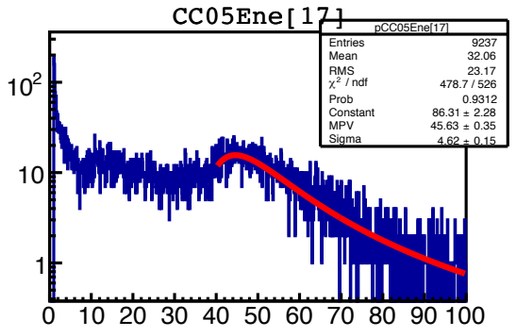


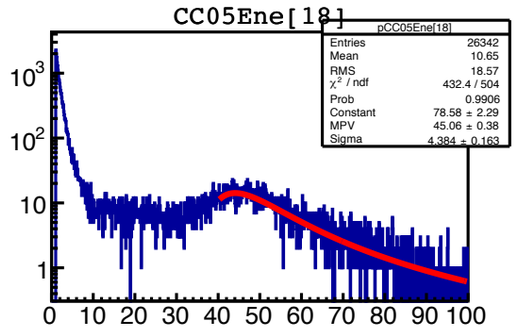
Setting the CC05Ene threshold

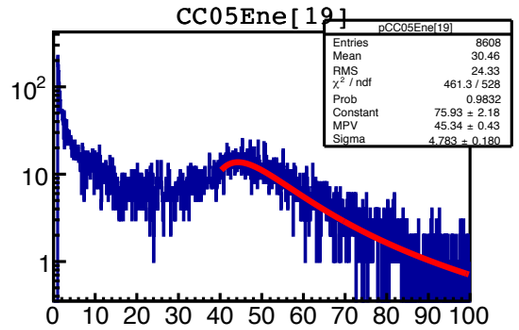
/home/had/hmkim/work/hmkim/run81/CC05thre.C → CC05thre_30776_30787.root

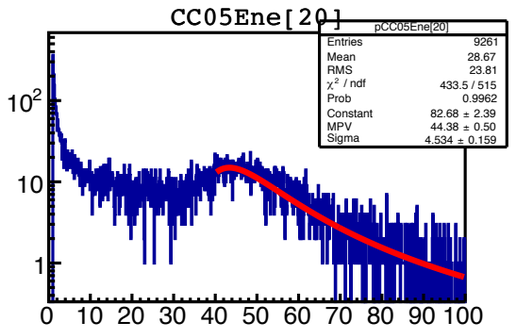
Fitting Function : Landau, CC05thre = CC05MPV × 0.7

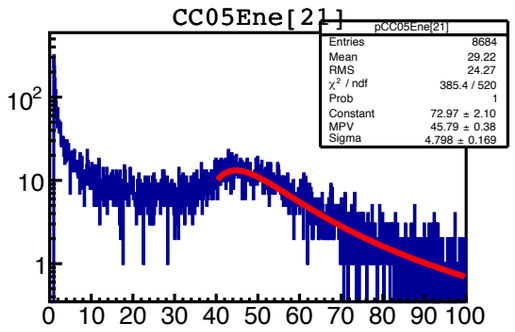


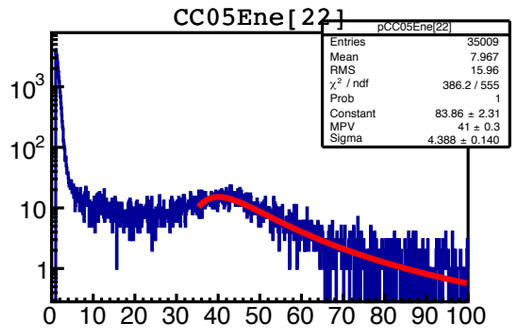


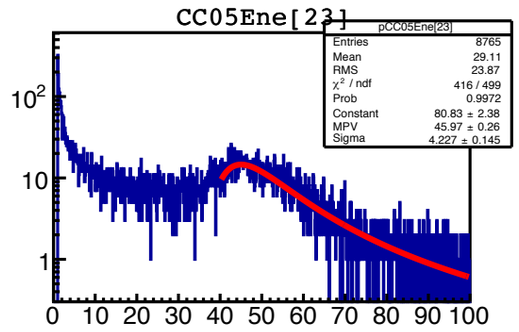


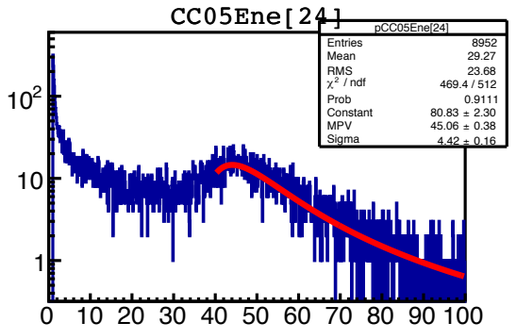


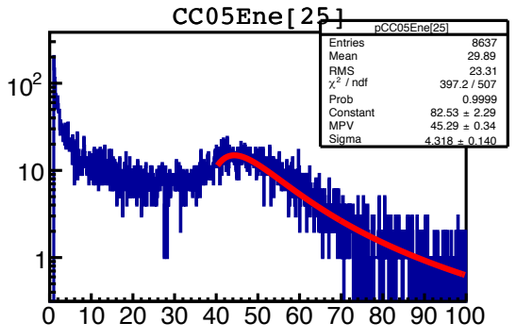


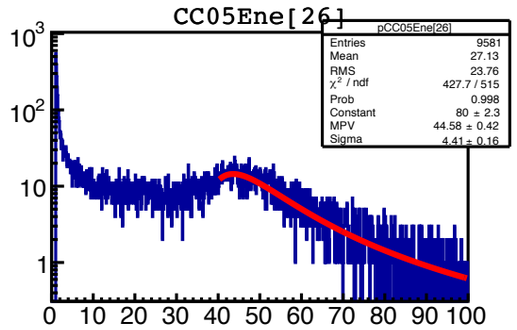


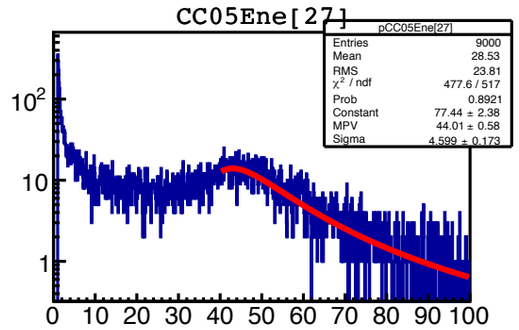


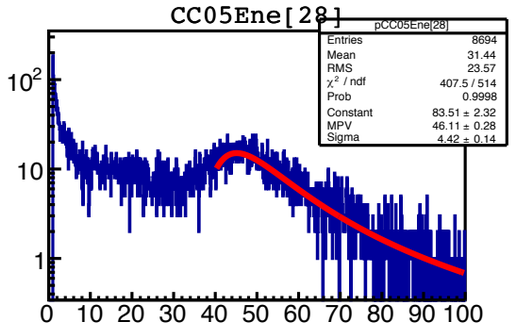


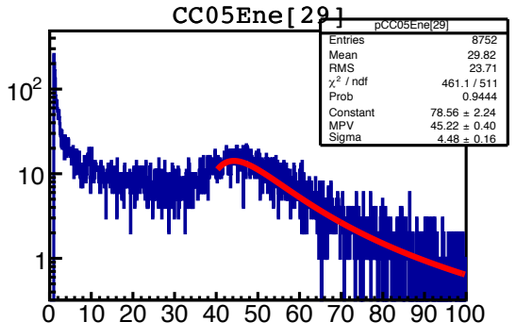


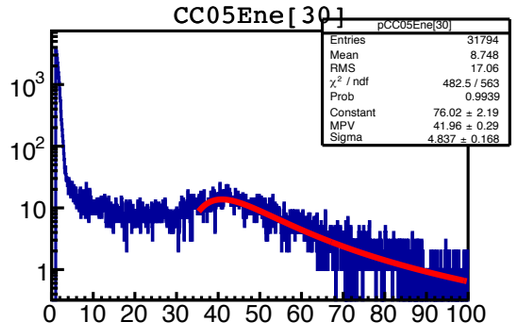


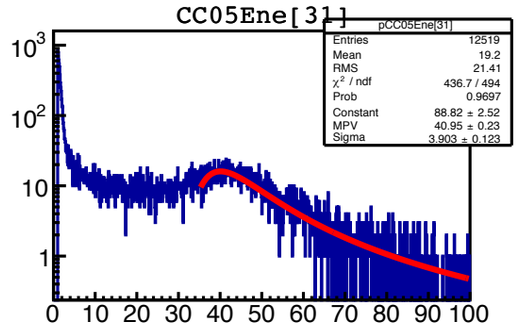








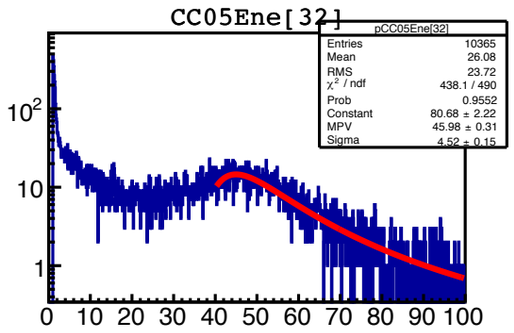


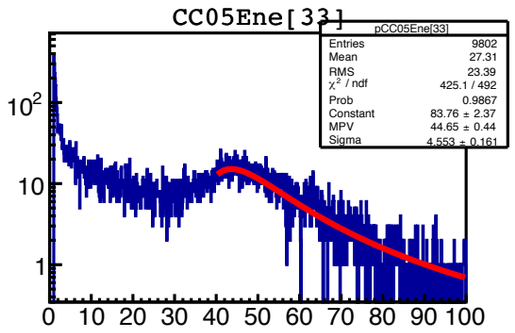


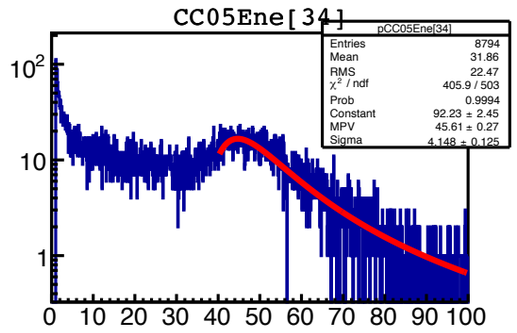
Setting the CC05Ene threshold

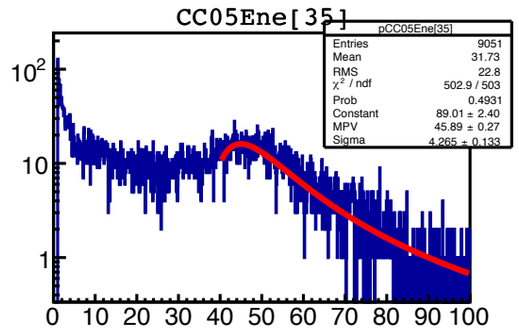
/home/had/hmkim/work/hmkim/run81/CC05thre.C → CC05thre_30776_30787.root

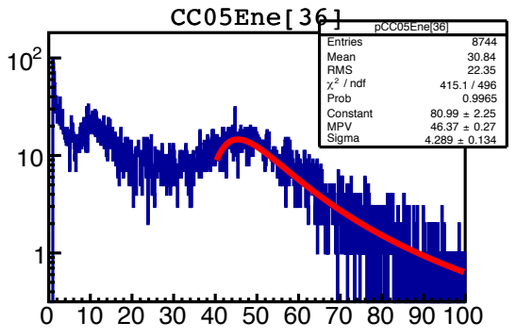
Fitting Function : Landau, CC05thre = CC05MPV × 0.7

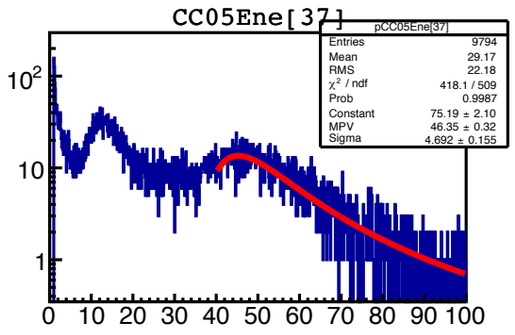


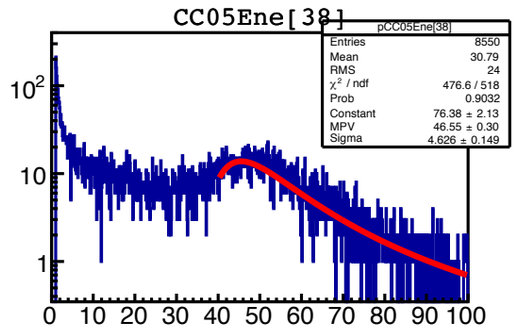


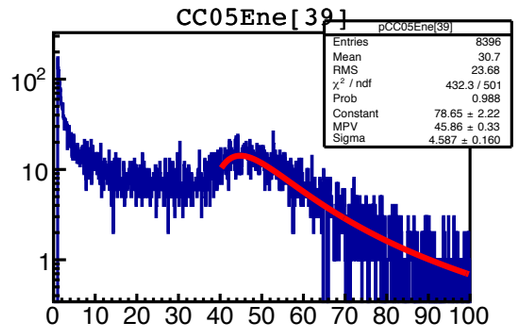


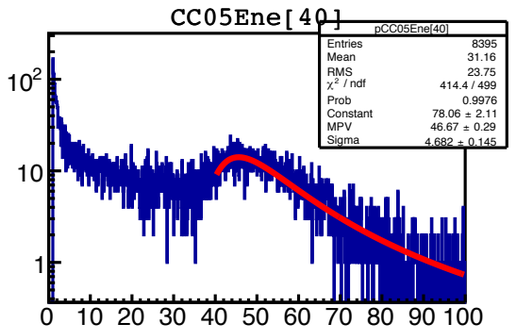


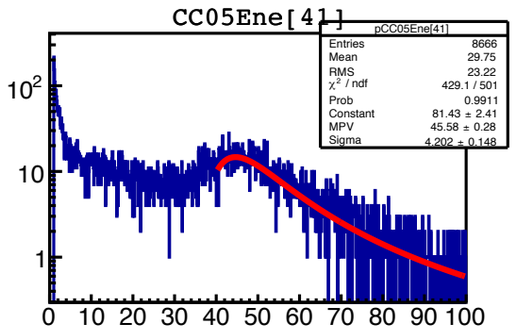


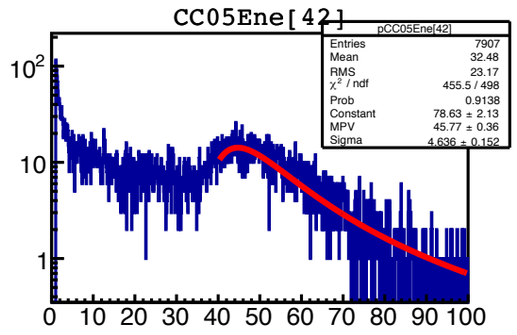


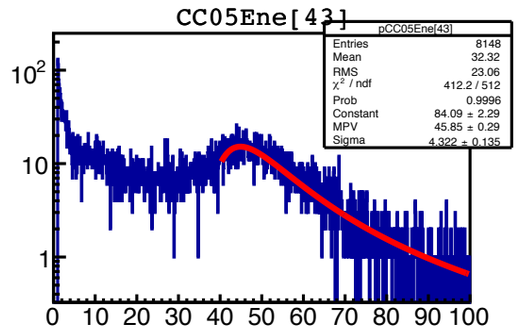


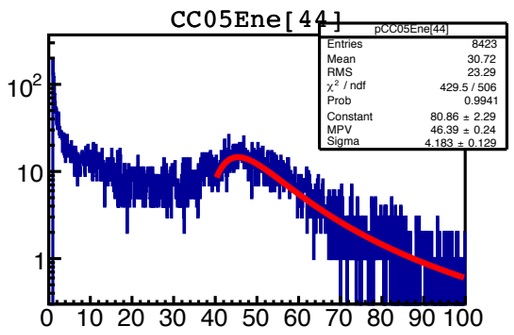


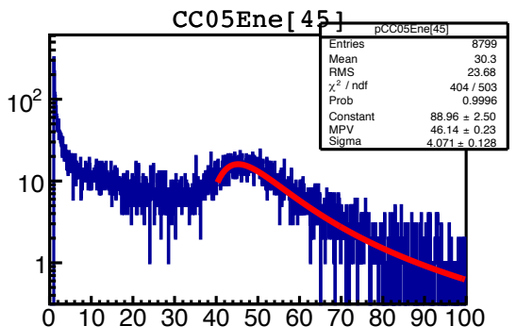


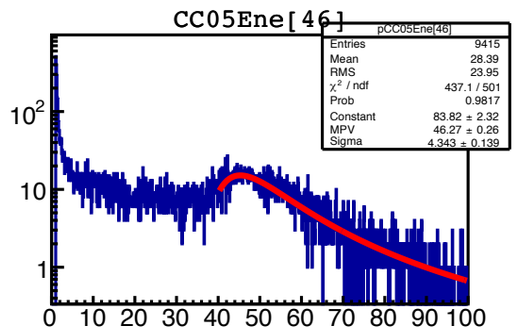


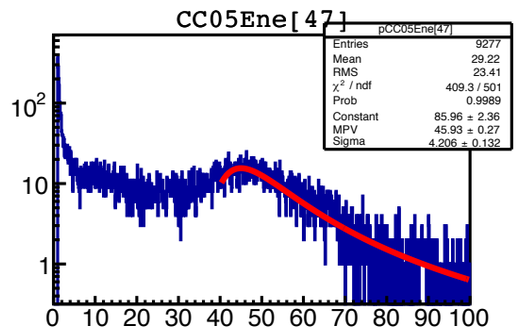








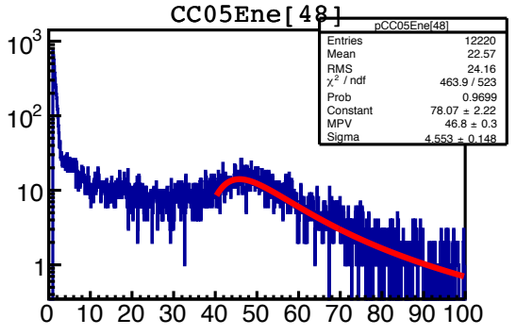


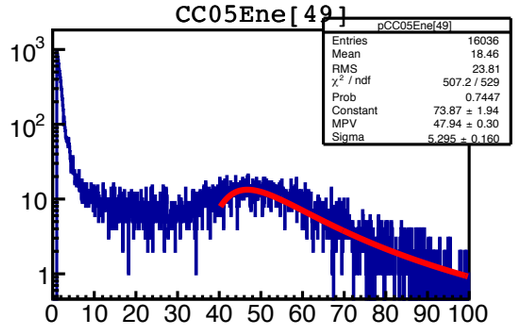


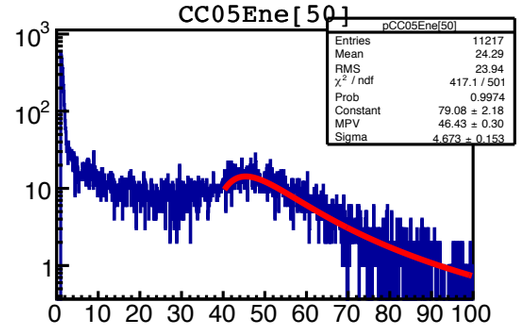
Setting the CC05Ene threshold

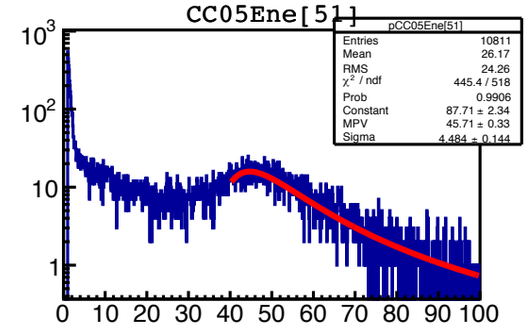
/home/had/hmkim/work/hmkim/run81/CC05thre.C → CC05thre_30776_30787.root

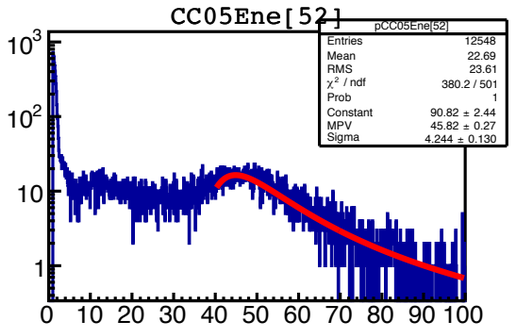
Fitting Function : Landau, CC05thre = CC05MPV × 0.7

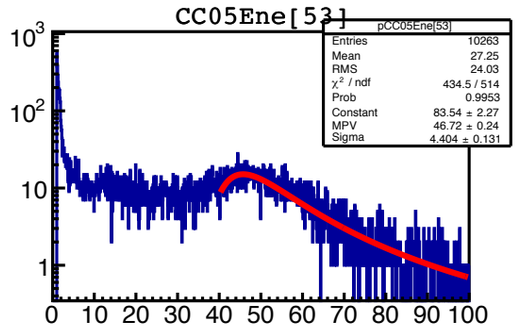


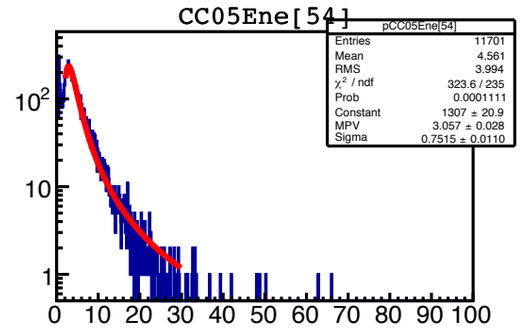


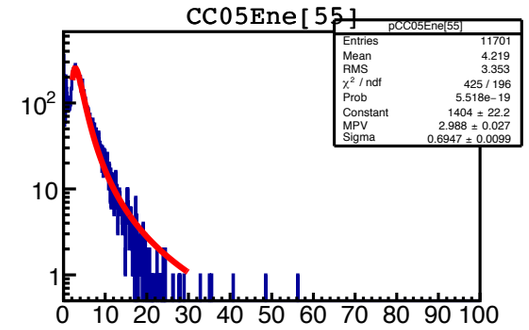


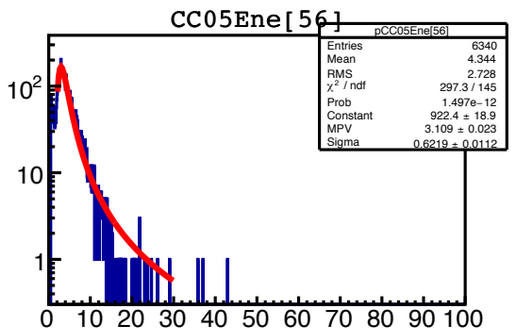


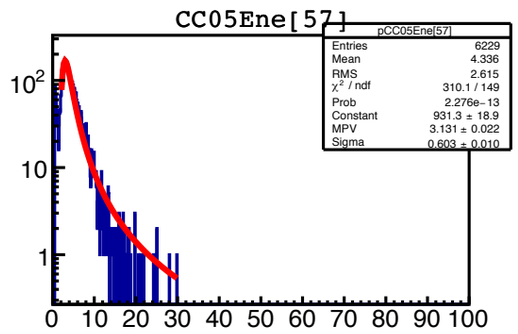


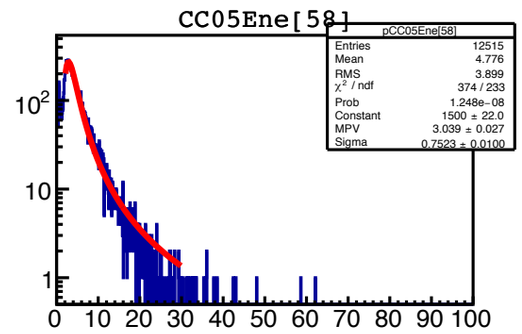


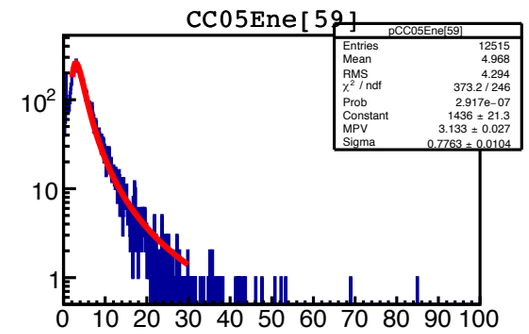












/home/had/hmkim/work/hmkim/run81/lim_ana/dcv1_calib1.C

```
float gain04[32]={359.7,402.8,261.2,299.8, //DCV1 module0
                  328.5,282.1,258.7,206.,  //DCV1 module1
                  327.4,417.2,244.1,294.5, //DCV1 module2
                  371.1,370.7,227.6,244.1, //DCV1 module3
                  1.,1.,1.,1.,1.,1.,1.,1.,   // DCV2 module 0 & 1
                  1.,1.,1.,1.,1.,1.,1.,1. }; // DCV2 module 2 & 3
```

```
for(int id=0;id<DCVNumber;id++){
  int ich=DCVModID[id];
  DCVcalib[ich]=DCVPeak[id]/gain04[ich];
}
```

```
for(int ich=0;ich<16;ich++){
  if(TrFlag04[0]==1) hE0[ich]->Fill(DCVcalib[2*ich]+DCVcalib[2*ich+1]);
  if(TrFlag04[1]==1) hE1[ich]->Fill(DCVcalib[2*ich]+DCVcalib[2*ich+1]);
  if(TrFlag04[2]==1) hE2[ich]->Fill(DCVcalib[2*ich]+DCVcalib[2*ich+1]);
  if(TrFlag04[3]==1) hE3[ich]->Fill(DCVcalib[2*ich]+DCVcalib[2*ich+1]);
  if(TrFlag04[4]==1) hE4[ich]->Fill(DCVcalib[2*ich]+DCVcalib[2*ich+1]);
  if(TrFlag04[5]==1) hE5[ich]->Fill(DCVcalib[2*ich]+DCVcalib[2*ich+1]);
}
```

/home/had/hmkim/work/hmkim/run81/final_cal/dcv1_calib2.C

```
float norm04[32] = {1.291,1.349,1.273,1.325,
                    1.179,1.202,1.185,1.181,
                    1.316,1.324,1.334,1.338,
                    1.178,1.195,1.167,1.182,
                    1.,1.,1.,1.,1.,1.,1.,1.,
                    1.,1.,1.,1.,1.,1.,1.,1. };
```

```
for(int id=0;id<DCVNumber;id++){
  int ich=DCVModID[id];
  DCVcalib[ich]=DCVPeak[id]/(gain04[ich]*norm04[ich]);
}
```

```
for(int ich=0;ich<8;ich++){
  if(TrFlag04[0]==1) hES0[ich]->Fill(DCVcalib[4*ich]+DCVcalib[4*ich+1]+DCVcalib[4*ich+3]+DCVcalib[4*ich+4]);
  if(TrFlag04[1]==1) hES1[ich]->Fill(DCVcalib[4*ich]+DCVcalib[4*ich+1]+DCVcalib[4*ich+3]+DCVcalib[4*ich+4]);
  if(TrFlag04[2]==1) hES2[ich]->Fill(DCVcalib[4*ich]+DCVcalib[4*ich+1]+DCVcalib[4*ich+3]+DCVcalib[4*ich+4]);
  if(TrFlag04[3]==1) hES3[ich]->Fill(DCVcalib[4*ich]+DCVcalib[4*ich+1]+DCVcalib[4*ich+3]+DCVcalib[4*ich+4]);
  if(TrFlag04[4]==1) hES4[ich]->Fill(DCVcalib[4*ich]+DCVcalib[4*ich+1]+DCVcalib[4*ich+3]+DCVcalib[4*ich+4]);
  if(TrFlag04[5]==1) hES5[ich]->Fill(DCVcalib[4*ich]+DCVcalib[4*ich+1]+DCVcalib[4*ich+3]+DCVcalib[4*ich+4]);
}
```

/home/had/hmkim/work/hmkim/run81/final_cal/dcv1_calib3.C

```
float norm_module04[32] = {2.01,2.083,2.01,2.083,
                           2.049,2.068,2.049,2.068,
                           2.012,2.018,2.012,2.018,
                           2.044,2.05,2.044,2.05,
                           1.,1.,1.,1.,1.,1.,1.,1.,
                           1.,1.,1.,1.,1.,1.,1.,1. };
```

```
for(int id=0;id<DCVNumber;id++){
  int ich=DCVModID[id];
  DCVcalib[ich]=(DCVPeak[id]/(gain04[ich]*norm04[ich]*norm_module04[ich]);
}
```

```
for(int ich=0;ich<8;ich++){
  if(TrFlag04[0]==1) hES0[ich]->Fill(DCVcalib[4*ich]+DCVcalib[4*ich+1]+DCVcalib[4*ich+2]+DCVcalib[4*ich+3]);
  if(TrFlag04[1]==1) hES1[ich]->Fill(DCVcalib[4*ich]+DCVcalib[4*ich+1]+DCVcalib[4*ich+2]+DCVcalib[4*ich+3]);
  if(TrFlag04[2]==1) hES2[ich]->Fill(DCVcalib[4*ich]+DCVcalib[4*ich+1]+DCVcalib[4*ich+2]+DCVcalib[4*ich+3]);
  if(TrFlag04[3]==1) hES3[ich]->Fill(DCVcalib[4*ich]+DCVcalib[4*ich+1]+DCVcalib[4*ich+2]+DCVcalib[4*ich+3]);
  if(TrFlag04[4]==1) hES4[ich]->Fill(DCVcalib[4*ich]+DCVcalib[4*ich+1]+DCVcalib[4*ich+2]+DCVcalib[4*ich+3]);
  if(TrFlag04[5]==1) hES5[ich]->Fill(DCVcalib[4*ich]+DCVcalib[4*ich+1]+DCVcalib[4*ich+2]+DCVcalib[4*ich+3]);
}
```

/home/had/hmkim/work/hmkim/run81/final_cal/dcv2_calib1.C

```
float gain05[32] = {1.,1.,1.,1.,1.,1.,1.,1.,
                    1.,1.,1.,1.,1.,1.,1.,1.,
                    301.2,340.2,220.4,226.3,
                    263.6,242.2,153.8,155.4,
                    284.2,256.8,173.3,172.1,
                    273.7,263.3,157.2,159.6 };
```

```
for(int id=0;id<DCVNumber;id++){
  int ich=DCVModID[id];
  DCVcalib[ich]=DCVPeak[id]/gain05[ich];
}
```

```
for(int ich=0;ich<16;ich++){
  if(TrFlag05[0]==1) hE0[ich]->Fill(DCVcalib[2*ich]+DCVcalib[2*ich+1]);
  if(TrFlag05[1]==1) hE1[ich]->Fill(DCVcalib[2*ich]+DCVcalib[2*ich+1]);
  if(TrFlag05[2]==1) hE2[ich]->Fill(DCVcalib[2*ich]+DCVcalib[2*ich+1]);
  if(TrFlag05[3]==1) hE3[ich]->Fill(DCVcalib[2*ich]+DCVcalib[2*ich+1]);
  if(TrFlag05[4]==1) hE4[ich]->Fill(DCVcalib[2*ich]+DCVcalib[2*ich+1]);
  if(TrFlag05[5]==1) hE5[ich]->Fill(DCVcalib[2*ich]+DCVcalib[2*ich+1]);
}
```

/home/had/hmkim/work/hmkim/run81/final_cal/dcv2_calib2.C

```
float norm05[32] = {1.,1.,1.,1.,1.,1.,1.,1.,
                    1.,1.,1.,1.,1.,1.,1.,1.,
                    1.172,1.224,1.211,1.239,
                    1.145,1.132,1.214,1.221,
                    1.296,1.199,1.294,1.249,
                    1.129,1.168,1.161,1.219 };
```

```
for(int id=0;id<DCVNumber;id++){
  int ich=DCVModID[id];
  DCVcalib[ich]=DCVPeak[id]/(gain05[ich]*norm05[ich]);
}
```

```
for(int ich=0;ich<8;ich++){
  if(TrFlag05[0]==1) hES0[ich]->Fill(DCVcalib[4*ich]+DCVcalib[4*ich+1]+DCVcalib[4*ich+3]+DCVcalib[4*ich+4]);
  if(TrFlag05[1]==1) hES1[ich]->Fill(DCVcalib[4*ich]+DCVcalib[4*ich+1]+DCVcalib[4*ich+3]+DCVcalib[4*ich+4]);
  if(TrFlag05[2]==1) hES2[ich]->Fill(DCVcalib[4*ich]+DCVcalib[4*ich+1]+DCVcalib[4*ich+3]+DCVcalib[4*ich+4]);
  if(TrFlag05[3]==1) hES3[ich]->Fill(DCVcalib[4*ich]+DCVcalib[4*ich+1]+DCVcalib[4*ich+3]+DCVcalib[4*ich+4]);
  if(TrFlag05[4]==1) hES4[ich]->Fill(DCVcalib[4*ich]+DCVcalib[4*ich+1]+DCVcalib[4*ich+3]+DCVcalib[4*ich+4]);
  if(TrFlag05[5]==1) hES5[ich]->Fill(DCVcalib[4*ich]+DCVcalib[4*ich+1]+DCVcalib[4*ich+3]+DCVcalib[4*ich+4]);
}
```

/home/had/hmkim/work/hmkim/run81/final_cal/dcv2_calib3.C

```
float norm_module05[32] = {1.,1.,1.,1.,1.,1.,1.,1.,
                           1.,1.,1.,1.,1.,1.,1.,1.,
                           2.01,2.023,2.01,2.023,
                           2.045,2.077,2.045,2.077,
                           2.044,2.007,2.044,2.007,
                           2.05,2.05,2.05,2.05      };
```

```
for(int id=0;id<DCVNumber;id++){
  int ich=DCVModID[id];
  DCVcalib[ich]=(DCVPeak[id]/(gain05[ich]*norm05[ich]*norm_module05[ich]);
}
```

```
for(int ich=0;ich<8;ich++){
  if(TrFlag05[0]==1) hES0[ich]->Fill(DCVcalib[4*ich]+DCVcalib[4*ich+1]+DCVcalib[4*ich+2]+DCVcalib[4*ich+3]);
  if(TrFlag05[1]==1) hES1[ich]->Fill(DCVcalib[4*ich]+DCVcalib[4*ich+1]+DCVcalib[4*ich+2]+DCVcalib[4*ich+3]);
  if(TrFlag05[2]==1) hES2[ich]->Fill(DCVcalib[4*ich]+DCVcalib[4*ich+1]+DCVcalib[4*ich+2]+DCVcalib[4*ich+3]);
  if(TrFlag05[3]==1) hES3[ich]->Fill(DCVcalib[4*ich]+DCVcalib[4*ich+1]+DCVcalib[4*ich+2]+DCVcalib[4*ich+3]);
  if(TrFlag05[4]==1) hES4[ich]->Fill(DCVcalib[4*ich]+DCVcalib[4*ich+1]+DCVcalib[4*ich+2]+DCVcalib[4*ich+3]);
  if(TrFlag05[5]==1) hES5[ich]->Fill(DCVcalib[4*ich]+DCVcalib[4*ich+1]+DCVcalib[4*ich+2]+DCVcalib[4*ich+3]);
}
```

Total calculation the calibration factors including path length

/home/had/hmkim/work/hmkim/run81/final_cal/out_cal/make_calib_factor.C

```
float gain[32] = {359.7,402.8,261.2,299.8,
                  328.5,282.1,258.7,206.,
                  327.4,417.2,244.1,294.5,
                  371.1,370.7,227.6,244.1,
                  301.2,340.2,220.4,226.3,
                  263.6,242.2,153.8,155.4,
                  284.2,256.8,173.3,172.1,
                  273.7,263.3,157.2,159.6 };

float norm[32] = {1.291,1.349,1.273,1.325,
                  1.179,1.202,1.185,1.181,
                  1.316,1.324,1.334,1.338,
                  1.178,1.195,1.167,1.182,
                  1.172,1.224,1.211,1.239,
                  1.145,1.132,1.214,1.221,
                  1.296,1.199,1.294,1.249,
                  1.129,1.168,1.161,1.219 };

float norm_module[32] = {2.01,2.083,2.01,2.083,
                         2.049,2.068,2.049,2.068,
                         2.012,2.018,2.012,2.018,
                         2.044,2.05,2.044,2.05,
                         2.01,2.023,2.01,2.023,
                         2.045,2.077,2.045,2.077,
                         2.044,2.007,2.044,2.007,
                         2.05,2.05,2.05,2.05      };
```

```
float lambda_dcv1 = 2469; // Attenuation Length of DCV1
float pos_DCV1 = 7835; // Center position of DCV1
float pos_CC04 = 7540; // Center position of CC04
float att1_f = exp((pos_DCV1 - pos_CC04)/lambda_dcv1);
float att1_r = exp(-(pos_DCV1 - pos_CC04)/lambda_dcv1);

float lambda_dcv2 = 2567; // Attenuation Length of DCV2
float pos_DCV2 = 9346; // Center position of DCV2
float pos_CC05 = 8899; // Center position of CC05
float att2_f = exp((pos_DCV2 - pos_CC05)/lambda_dcv2);
float att2_r = exp(-(pos_DCV2 - pos_CC05)/lambda_dcv2);

float att[32] = {att1_f,att1_f,att1_r,att1_r,
                 att1_f,att1_f,att1_r,att1_r,
                 att1_f,att1_f,att1_r,att1_r,
                 att1_f,att1_f,att1_r,att1_r,
                 att2_f,att2_f,att2_r,att2_r,
                 att2_f,att2_f,att2_r,att2_r,
                 att2_f,att2_f,att2_r,att2_r,
                 att2_f,att2_f,att2_r,att2_r };
```

```
float a1 = 170*3/4;
float x = 5*17/317;
float b1 = (x+120)/2;
float c1 = sqrt((a1*a1)+(b1*b1));
float sin1 = a1/c1;

float a2 = 210*3/4;
float b2 = 210/2;
float c2 = sqrt((a2*a2)+(b2*b2));
float sin2 = a2/c2;

float path_length[32] =
      {sin1,sin1,sin1,sin1,
       1.,1.,1.,1.,
       sin1,sin1,sin1,sin1,
       1.,1.,1.,1.,
       sin2,sin2,sin2,sin2,
       1.,1.,1.,1.,
       sin2,sin2,sin2,sin2,
       1.,1.,1.,1. };
```

```
float calib_factor_wo_path[32];
float calib_factor[32];

for(int i=0;i<32;i++){
  calib_factor_wo_path[i] = att[i]/(gain[i] * norm[i] * norm_module[i]);
  calib_factor[i] = att[i]*path_length[i]/(gain[i] * norm[i] * norm_module[i]);
}
```

```
DCV Ch : 0, Calib_Factor_wo_path : 0.00120733, Calib_Factor : 0.00133529
DCV Ch : 1, Calib_Factor_wo_path : 0.000995634, Calib_Factor : 0.00110115
DCV Ch : 2, Calib_Factor_wo_path : 0.00132774, Calib_Factor : 0.00146846
DCV Ch : 3, Calib_Factor_wo_path : 0.00107244, Calib_Factor : 0.0011861
DCV Ch : 4, Calib_Factor_wo_path : 0.00142003, Calib_Factor : 0.00142003
DCV Ch : 5, Calib_Factor_wo_path : 0.00160706, Calib_Factor : 0.00160706
DCV Ch : 6, Calib_Factor_wo_path : 0.00141271, Calib_Factor : 0.00141271
DCV Ch : 7, Calib_Factor_wo_path : 0.00176377, Calib_Factor : 0.00176377
DCV Ch : 8, Calib_Factor_wo_path : 0.00129995, Calib_Factor : 0.00143773
DCV Ch : 9, Calib_Factor_wo_path : 0.00101097, Calib_Factor : 0.00111811
DCV Ch : 10, Calib_Factor_wo_path : 0.00135444, Calib_Factor : 0.00149798
DCV Ch : 11, Calib_Factor_wo_path : 0.00111596, Calib_Factor : 0.00123423
DCV Ch : 12, Calib_Factor_wo_path : 0.00126117, Calib_Factor : 0.00126117
DCV Ch : 13, Calib_Factor_wo_path : 0.00124093, Calib_Factor : 0.00124093
DCV Ch : 14, Calib_Factor_wo_path : 0.0016345, Calib_Factor : 0.0016345
DCV Ch : 15, Calib_Factor_wo_path : 0.00150027, Calib_Factor : 0.00150027
DCV Ch : 16, Calib_Factor_wo_path : 0.00167744, Calib_Factor : 0.00201801
DCV Ch : 17, Calib_Factor_wo_path : 0.00141291, Calib_Factor : 0.00169977
DCV Ch : 18, Calib_Factor_wo_path : 0.00156611, Calib_Factor : 0.00188408
DCV Ch : 19, Calib_Factor_wo_path : 0.00148123, Calib_Factor : 0.00178197
DCV Ch : 20, Calib_Factor_wo_path : 0.00192833, Calib_Factor : 0.00192833
DCV Ch : 21, Calib_Factor_wo_path : 0.0020901, Calib_Factor : 0.0020901
DCV Ch : 22, Calib_Factor_wo_path : 0.00220042, Calib_Factor : 0.00220042
DCV Ch : 23, Calib_Factor_wo_path : 0.00213192, Calib_Factor : 0.00213192
DCV Ch : 24, Calib_Factor_wo_path : 0.00158094, Calib_Factor : 0.00190192
DCV Ch : 25, Calib_Factor_wo_path : 0.00192603, Calib_Factor : 0.00231707
DCV Ch : 26, Calib_Factor_wo_path : 0.00183299, Calib_Factor : 0.00220515
DCV Ch : 27, Calib_Factor_wo_path : 0.00194753, Calib_Factor : 0.00234294
DCV Ch : 28, Calib_Factor_wo_path : 0.0018789, Calib_Factor : 0.0018789
DCV Ch : 29, Calib_Factor_wo_path : 0.00188789, Calib_Factor : 0.00188789
DCV Ch : 30, Calib_Factor_wo_path : 0.00224562, Calib_Factor : 0.00224562
DCV Ch : 31, Calib_Factor_wo_path : 0.00210661, Calib_Factor : 0.00210661
```

Other cosmic run periods in Run81

Period1

30910	2019-03-04 11:01:48	2019-03-04 11:52:34	COSMIC	Cosmic_PS_CSI_PartOfVetos
30911	2019-03-04 11:59:52	2019-03-04 12:50:36	COSMIC	Cosmic_PS_CSI_PartOfVetos
31011	2019-03-07 11:33:10	2019-03-07 12:23:33	COSMIC	Cosmic_PS_CSI_PartOfVetos
31012	2019-03-07 12:24:49	2019-03-07 13:15:11	COSMIC	Cosmic_PS_CSI_PartOfVetos
31014	2019-03-07 14:04:26	2019-03-07 14:54:44	COSMIC	Cosmic_PS_CSI_PartOfVetos
31015	2019-03-07 14:56:51	2019-03-07 15:47:46	COSMIC	Cosmic_PS_CSI_PartOfVetos
31016	2019-03-07 15:48:50	2019-03-07 16:19:17	COSMIC	Cosmic_PS_CSI_PartOfVetos
31017	2019-03-07 16:26:43	2019-03-07 17:17:25	COSMIC	Cosmic_PS_CSI_PartOfVetos
31018	2019-03-07 17:23:24	2019-03-07 18:14:09	COSMIC	Cosmic_PS_CSI_PartOfVetos
31019	2019-03-07 18:40:48	2019-03-07 19:12:06	COSMIC	Cosmic_PS_CSI_PartOfVetos
31020	2019-03-07 19:15:40	2019-03-07 20:06:07	COSMIC	Cosmic_PS_CSI_PartOfVetos
31021	2019-03-07 20:08:42	2019-03-07 20:59:09	COSMIC	Cosmic_PS_CSI_PartOfVetos
31022	2019-03-07 21:01:44	2019-03-07 21:52:05	COSMIC	Cosmic_PS_CSI_PartOfVetos
31023	2019-03-07 21:53:35	2019-03-07 22:44:07	COSMIC	Cosmic_PS_CSI_PartOfVetos
31024	2019-03-07 22:46:08	2019-03-07 23:36:26	COSMIC	Cosmic_PS_CSI_PartOfVetos
31025	2019-03-07 23:37:56	2019-03-08 00:28:28	COSMIC	Cosmic_PS_CSI_PartOfVetos

Other cosmic run periods in Run81

31026	2019-03-08 00:29:30	2019-03-08 01:19:48	COSMIC	Cosmic_PS_CSI_PartOfVetos
31027	2019-03-08 01:21:04	2019-03-08 02:11:44	COSMIC	Cosmic_PS_CSI_PartOfVetos
31028	2019-03-08 02:17:55	2019-03-08 03:08:46	COSMIC	Cosmic_PS_CSI_PartOfVetos
31029	2019-03-08 03:12:30	2019-03-08 04:02:53	COSMIC	Cosmic_PS_CSI_PartOfVetos
31030	2019-03-08 04:04:07	2019-03-08 04:54:25	COSMIC	Cosmic_PS_CSI_PartOfVetos
31031	2019-03-08 05:03:02	2019-03-08 05:21:45	COSMIC	Cosmic_PS_CSI_PartOfVetos
31032	2019-03-08 07:34:32	2019-03-08 08:25:08	COSMIC	Cosmic_PS_CSI_PartOfVetos
31033	2019-03-08 08:26:48	2019-03-08 09:17:16	COSMIC	Cosmic_PS_CSI_PartOfVetos
31034	2019-03-08 09:18:26	2019-03-08 10:01:48	COSMIC	Cosmic_PS_CSI_PartOfVetos
31035	2019-03-08 10:05:34	2019-03-08 10:21:00	COSMIC	Cosmic_PS_CSI_PartOfVetos

Other cosmic run periods in Run81

Period2

31240	2019-03-14 15:04:25	2019-03-14 15:55:03	COSMIC	Cosmic_TightPS
31241	2019-03-14 15:56:02	2019-03-14 16:46:41	COSMIC	Cosmic_TightPS
31242	2019-03-14 16:47:40	2019-03-14 17:38:07	COSMIC	Cosmic_TightPS
31243	2019-03-14 17:39:07	2019-03-14 18:29:27	COSMIC	Cosmic_TightPS
31244	2019-03-14 18:30:26	2019-03-14 19:20:47	COSMIC	Cosmic_TightPS
31245	2019-03-14 19:21:41	2019-03-14 20:12:01	COSMIC	Cosmic_TightPS
31246	2019-03-14 20:13:56	2019-03-14 21:04:51	COSMIC	Cosmic_TightPS
31247	2019-03-14 21:05:46	2019-03-14 21:56:05	COSMIC	Cosmic_TightPS
31364	2019-03-18 09:32:34	2019-03-18 10:23:26	COSMIC	Cosmic_TightPS
31373	2019-03-18 22:33:46	2019-03-18 22:36:06	COSMIC	Cosmic_TightPS

Other cosmic run periods in Run81

Period3

31374	2019-03-18 23:56:46	2019-03-19 00:47:41	COSMIC	Cosmic_TightPS_IBTrig
31375	2019-03-19 04:31:42	2019-03-19 05:22:22	COSMIC	Cosmic_TightPS_IBTrig
31376	2019-03-19 05:39:13	2019-03-19 06:29:48	COSMIC	Cosmic_TightPS_IBTrig
31377	2019-03-19 06:30:58	2019-03-19 07:21:20	COSMIC	Cosmic_TightPS_IBTrig
31378	2019-03-19 07:22:23	2019-03-19 08:12:46	COSMIC	Cosmic_TightPS_IBTrig
31379	2019-03-19 08:13:59	2019-03-19 09:04:24	COSMIC	Cosmic_TightPS_IBTrig
31380	2019-03-19 09:05:28	2019-03-19 09:55:50	COSMIC	Cosmic_TightPS_IBTrig
31381	2019-03-19 09:57:15	2019-03-19 10:48:16	COSMIC	Cosmic_TightPS_IBTrig
31382	2019-03-19 10:49:10	2019-03-19 11:39:48	COSMIC	Cosmic_TightPS_IBTrig
31383	2019-03-19 11:40:58	2019-03-19 12:31:20	COSMIC	Cosmic_TightPS_IBTrig
31384	2019-03-19 12:32:23	2019-03-19 13:22:58	COSMIC	Cosmic_TightPS_IBTrig
31385	2019-03-19 13:23:57	2019-03-19 14:14:18	COSMIC	Cosmic_TightPS_IBTrig
31386	2019-03-19 14:15:30	2019-03-19 15:06:08	COSMIC	Cosmic_TightPS_IBTrig
31387	2019-03-19 15:07:17	2019-03-19 15:57:46	COSMIC	Cosmic_TightPS_IBTrig
31388	2019-03-19 15:58:48	2019-03-19 16:49:12	COSMIC	Cosmic_TightPS_IBTrig
31389	2019-03-19 16:53:35	2019-03-19 17:44:15	COSMIC	Cosmic_TightPS_MBTrig

Other cosmic run periods in Run81

Period4

31422	2019-03-20 20:56:08	2019-03-20 21:46:29	COSMIC	Cosmic_TightPS_MPPCconfig5
⋮				
31446	2019-03-21 17:33:44	2019-03-21 18:24:10	COSMIC	Cosmic_TightPS_MPPCconfig5

Period5

31447	2019-03-21 18:30:47	2019-03-21 18:32:59	COSMIC	Cosmic_TightPS_MPPCconfig10_CC04CC05
⋮				
31506	2019-03-23 20:37:58	2019-03-23 20:38:25	COSMIC	Cosmic_TightPS_MPPCconfig10_CC04CC05