Validation of single-particle test samples with SDHCal and comparison with AHCaL

CIEMAT SDHCAL group meeting 23/10/20

Camilo

GOBIERNO DE ESPAÑA

MINISTERIO DE CIENCIA E INNOVACIÓN

Ciemat

Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas

cfp

CIEMAT física de partículas

Identifying the start of the shower to filter events

We have two options:

- The parameter should be in ilcSoft (maybe stored in the LCTuple may be not (if not we can add it and produce again the ntuples).
- As another option we can try to identify the position where the shower starts using the test beam analysis tools (Hector)
  - What information would be needed?
  - x, y, z or i, j(path) plane of each hit would be enough?
- As you can see in the list of available variables the most promising variables: rcmo(x, y, z)[nrec]
- **As a 3rd option, selecting events without ECAL activity**

List of variables available in the standard LCTuple

```
// Fixed size dimensions of array or collections stored in the TTree if any.

   // Declaration of leaf types
   Int_t           evevt;
   Int_t           evrun;
   Float_t         evwgt;
   Long64_t        evtim;
   Float_t         evsig;
   Float_t         evene;
   Float_t         evpoe;
   Float_t         evpop;
   Int_t           evnch;
   Char_t          evpro[1];   //[evnch]
   Int_t           nmcp;
   Int_t           mcori[807];   //[nmcp]
   Int_t           mcpdg[807];   //[nmcp]
   Int_t           mcgst[807];   //[nmcp]
   Int_t           mcsst[807];   //[nmcp]
   Float_t         mcvtx[807];   //[nmcp]
   Float_t         mcvty[807];   //[nmcp]
   Float_t         mcvtz[807];   //[nmcp]
   Float_t         mcepx[807];   //[nmcp]
   Float_t         mcepy[807];   //[nmcp]
   Float_t         mcepz[807];   //[nmcp]
   Float_t         mcmox[807];   //[nmcp]
   Float_t         mcmoy[807];   //[nmcp]
   Float_t         mcmoz[807];   //[nmcp]
   Float_t         mcmas[807];   //[nmcp]
   Float_t         mcene[807];   //[nmcp]
   Float_t         mccha[807];   //[nmcp]
   Float_t         mctim[807];   //[nmcp]
   Float_t         mcspx[807];   //[nmcp]
   Float_t         mcspy[807];   //[nmcp]
   Float_t         mcspz[807];   //[nmcp]
   Int_t           mccf0[807];   //[nmcp]
   Int_t           mccf1[807];   //[nmcp]
   Int_t           mcpa0[807];   //[nmcp]
   Int_t           mcpa1[807];   //[nmcp]
   Int_t           mcda0[807];   //[nmcp]
   Int_t           mcda1[807];   //[nmcp]
   Int_t           mcda2[807];   //[nmcp]
   Int_t           mcda3[807];   //[nmcp]
   Int_t           mcda4[807];   //[nmcp]
   Int_t           nrec;
   Int_t           rcori[41];   //[nrec]
   Int_t           rccid[41];   //[nrec]
   Int_t           rctyp[41];   //[nrec]
   Float_t         rccov[41][10];   //[nrec]
   Float_t         rcrpx[41];   //[nrec]
   Float_t         rcrpy[41];   //[nrec]
   Float_t         rcrpz[41];   //[nrec]
   Float_t         rcgpi[41];   //[nrec]
   Int_t           rcpiu[41];   //[nrec]
   Int_t           rcnpi[41];   //[nrec]
   Int_t           rcfpi[41];   //[nrec]
   Float_t         rcmox[41];   //[nrec]
   Float_t         rcmoy[41];   //[nrec]
   Float_t         rcmoz[41];   //[nrec]
   Float_t         rcmas[41];   //[nrec]
   Float_t         rcene[41];   //[nrec]
   Float_t         rccha[41];   //[nrec]
   Int_t           rcntr[41];   //[nrec]
   Int_t           rcncl[41];   //[nrec]
   Int_t           rcnrp[41];   //[nrec]
   Int_t           rcftr[41];   //[nrec]
```

No ECAL hits!

```
Int_t           rcftr[41];   //[nrec]
Int_t           rcvts[41];   //[nrec]
Int_t           rcvte[41];   //[nrec]
Int_t           rccom[41];   //[nrec]
Int_t           npid;
Int_t           pityp[212];   //[npid]
Int_t           pipdg[212];   //[npid]
Float_t         pillh[212];   //[npid]
Int_t           pialg[212];   //[npid]
Int_t           ntrk;
Int_t           trori[36];   //[ntrk]
Int_t           trtyp[36];   //[ntrk]
Float_t         trch2[36];   //[ntrk]
Int_t           trndf[36];   //[ntrk]
Float_t         tredx[36];   //[ntrk]
Float_t         trede[36];   //[ntrk]
Float_t         trrih[36];   //[ntrk]
Int_t           trshn[36][12];   //[ntrk]
Int_t           trnts[36];   //[ntrk]
Int_t           trfts[36];   //[ntrk]
Int_t           trsip[36];   //[ntrk]
Int_t           trsfh[36];   //[ntrk]
Int_t           trslh[36];   //[ntrk]
Int_t           trsca[36];   //[ntrk]
Int_t           ntrst;
Int_t           tsloc[144];   //[ntrst]
Float_t         tsdze[144];   //[ntrst]
Float_t         tsphi[144];   //[ntrst]
Float_t         tsome[144];   //[ntrst]
Float_t         tszze[144];   //[ntrst]
Float_t         tstnl[144];   //[ntrst]
Float_t         tscov[144][15];   //[ntrst]
Float_t         tsrpx[144];   //[ntrst]
Float_t         tsrpy[144];   //[ntrst]
Float_t         tsrpz[144];   //[ntrst]
Int_t           nsth;
Int_t           stori[1];   //[nsth]
Int_t           stci0[1];   //[nsth]
Int_t           stci1[1];   //[nsth]
Double_t        stpox[1];   //[nsth]
Double_t        stpoy[1];   //[nsth]
Double_t        stpoz[1];   //[nsth]
Float_t         stedp[1];   //[nsth]
Float_t         sttim[1];   //[nsth]
Float_t         stmox[1];   //[nsth]
Float_t         stmoy[1];   //[nsth]
Float_t         stmoz[1];   //[nsth]
Float_t         stptl[1];   //[nsth]
Int_t           stmcp[1];   //[nsth]
Int_t           nsch;
Int_t           scori[1];   //[nsch]
Int_t           scci0[1];   //[nsch]
Int_t           scci1[1];   //[nsch]
Float_t         scpox[1];   //[nsch]
Float_t         scpoy[1];   //[nsch]
Float_t         scpoz[1];   //[nsch]
Float_t         scene[1];   //[nsch]
Int_t           r2mnrel;
Int_t           r2mf[169];   //[r2mnrel]
Int_t           r2mt[169];   //[r2mnrel]
Float_t         r2mw[169];   //[r2mnrel]
```

The Standard LCTuple xml Marling, no ECAL hits

```
<processor name="MergeSimCaloHits" type="MergeCollections">
  <!--MergeCollections creates a transient subset collection that merges all input collections -->
  <!--Names of all input collections-->
  <parameter name="InputCollections" type="StringVec">
    BeamCalCollection
    EcalBarrelCollection
    EcalEndcapRingCollection
    EcalEndcapsCollection
    HcalBarrelRegCollection
    HcalEndcapRingCollection
    HcalEndcapsCollection
    LumiCalCollection
    YokeBarrelCollection
    YokeEndcapsCollection
    LHCalCollection
  </parameter>
 <!--Optional IDs for input collections - if given id will be added to all objects in merged collections as ext<CollID>()-->
  <parameter name="InputCollectionIDs" type="IntVec">
    25 20 21 29 22 23 30 24 27 31
  </parameter>
  <parameter name="OutputCollection" type="string">SimCalorimeterHits </parameter>
  <!--verbosity level of this processor ("DEBUG0-4,MESSAGE0-4,WARNING0-4,ERROR0-4,SILENT")-->
  <parameter name="Verbosity" type="string">DEBUG </parameter>
 </processor>


<processor name="MergeCaloHits" type="MergeCollections">
  <!--MergeCollections creates a transient subset collection that merges all input collections -->
  <!--Names of all input collections-->
  <parameter name="InputCollections" type="StringVec">
    HcalBarrelCollectionDigi
    HcalEndcapRingCollectionDigi
    HcalEndcapsCollectionDigi
  </parameter>
 <!--Optional IDs for input collections - if given id will be added to all objects in merged collections as ext<CollID>()-->
  <parameter name="InputCollectionIDs" type="IntVec">
    29 22 23
  </parameter>
  <parameter name="OutputCollection" type="string">CaloHits </parameter>
  <!--verbosity level of this processor ("DEBUG0-4,MESSAGE0-4,WARNING0-4,ERROR0-4,SILENT")-->
  <parameter name="Verbosity" type="string">DEBUG </parameter>
 </processor>
```

The Standard LCTuple xml Marling, with ECAL hits

```xml
<processor name="MergeSimCaloHits" type="MergeCollections">
  <!--MergeCollections creates a transient subset collection that merges all input collections -->
  <!--Names of all input collections-->
  <parameter name="InputCollections" type="StringVec">
    BeamCalCollection
    EcalBarrelCollection
    EcalEndcapRingCollection
    EcalEndcapsCollection
    HcalBarrelRegCollection
    HcalEndcapRingCollection
    HcalEndcapsCollection
    LumiCalCollection
    YokeBarrelCollection
    YokeEndcapsCollection
    LHCalCollection
  </parameter>
 <!--Optional IDs for input collections - if given id will be added to all objects in merged collections as ext<CollID>()-->
  <parameter name="InputCollectionIDs" type="IntVec">
    25 20 21 29 22 23 30 24 27 31
  </parameter>
  <parameter name="OutputCollection" type="string">SimCalorimeterHits </parameter>
  <!--verbosity level of this processor ("DEBUG0-4,MESSAGE0-4,WARNING0-4,ERROR0-4,SILENT")-->
  <parameter name="Verbosity" type="string">DEBUG </parameter>
</processor>


<processor name="MergeCaloHits" type="MergeCollections">
  <!--MergeCollections creates a transient subset collection that merges all input collections -->
  <!--Names of all input collections-->
  <parameter name="InputCollections" type="StringVec">
    HcalBarrelCollectionDigi
    HcalEndcapRingCollectionDigi
    HcalEndcapsCollectionDigi
  </parameter>
 <!--Optional IDs for input collections - if given id will be added to all objects in merged collections as ext<CollID>()-->
  <parameter name="InputCollectionIDs" type="IntVec">
    25 20 21 29 22 23
  </parameter>
  <parameter name="OutputCollection" type="string">CaloHits </parameter>
  <!--verbosity level of this processor ("DEBUG0-4,MESSAGE0-4,WARNING0-4,ERROR0-4,SILENT")-->
  <parameter name="Verbosity" type="string">DEBUG </parameter>
</processor>
```

new LCTuple production running in CIEMAT

First test produciton is running in gaeucali...

Backup

Backup

The second test-dataset for the SDHCAL validation and AHCAL comparison

- This presentation is a follow up of our previous report `https://agenda.linearcollider.org/event/8559/`
- Details about the ILD confluence production for the **second** test production with the latest ilcsoft v02-01-02. `https://ild.ngt.ndu.ac.jp/elog/dbd-prod/323`
- We are interested again in K_L^0 particles
- For the first test production we presented results using high level objects (Physiscs objetcts). Now we have a working recipe that give us access to the low level objects (SDHCAL hits).

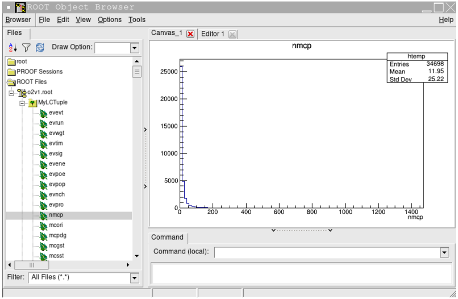
First look at the second test-dataset for the SDHCAL validation and AHCal comparison, K_L^0

- **o2** Energy range: (1,2,5,10,20,30,40,50,60,70,80,90,100,110) GeV. `/ilc/prod/ilc/mc-opt/ild/dst-merged/1-calib/single/ILD_l5_o2_v02_nobg/v02-01-02`
- **o1** Energy range: (1,2,5,10,20,30,40,50,**60**,70,**80,90,100,110**) GeV. (single particle dataset, in blue new datasets wrt first test sample) `/ilc/prod/ilc/mc-opt/ild/dst-merged/1-calib/single/ILD_l5_o1_v02_nobg/v02-01-02`
- We made a full copy of both datasets to our local cluster in CIEMAT dedicated to CALICE/ILD analysis by accessing the dataset via DIRAC.
- Using the same ilcsoft version v02-**01**-02 $\rightarrow$ `/cvmfs/ilc.desy.de/sw/x86_64_gcc82_sl6/v02-01-02/init_ilcsoft.sh` as for the central production we have produced the corresponding LCTuples.
- `/pool/calice3/data/MonteCarlo/sdhcal_validation/second_test_production/o1/dstm`

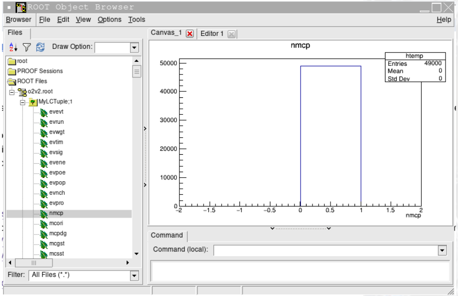
DST-merged datasets look different

First a remark about the LCtuples:

v02-00-02



v02-01-02



The nmcp variable accounts for the number of MC particles in a given event. In the default LCTuple this variable appears always at zero in this second test-production.

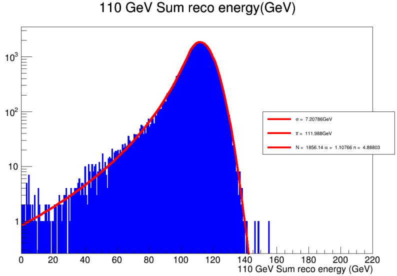
links with all results, please explore yourself:

- first test production
  - http://wwwae.ciemat.es/~carrillo/calice/indexk0o1.html
  - http://wwwae.ciemat.es/~carrillo/calice/indexk0o2.html
- second test production
  - http://wwwae.ciemat.es/~carrillo/calice/indexk0o1v2.html
  - http://wwwae.ciemat.es/~carrillo/calice/indexk0o2v2.html

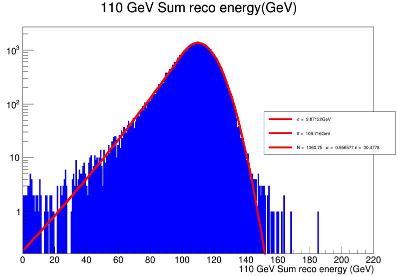
Comparison o1/o2 $\otimes$ $1^{st}/2^{nd}$ Test Production, K_L^0 110 GeV

2^{nd}TP

AHCAL(o1)



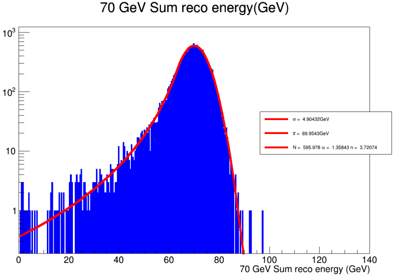
SDHCAL(o2)

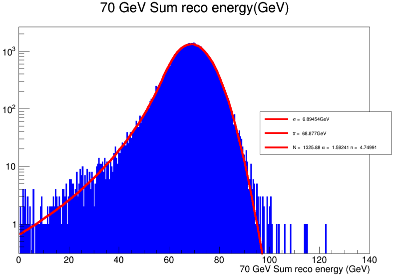


Comparison o1/o2 ⊗ $1^{st}/2^{nd}$ Test Production, K_L^0 70 GeV

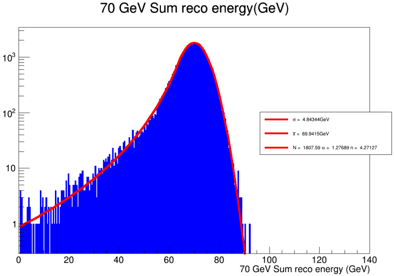
AHCAL(o1) | SDHCAL(o2)

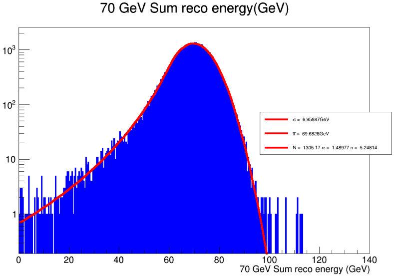
1^{st}TP



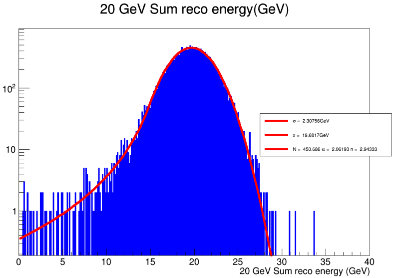
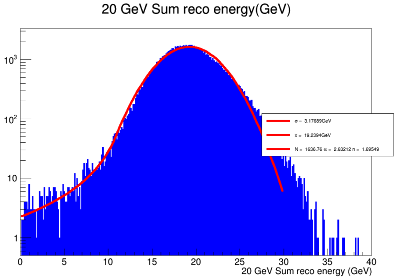
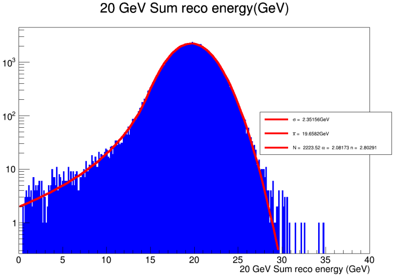
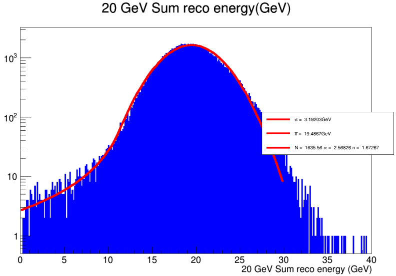


2^{nd}TP





Comparison o1/o2 ⊗ $1^{st}/2^{nd}$ Test Production, K_L^0 20 GeV

	AHCAL(o1)	SDHCAL(o2)
1^{st}TP		
2^{nd}TP		

resolution and discrepancy for o1 and o2, fit results

1^{st}TP

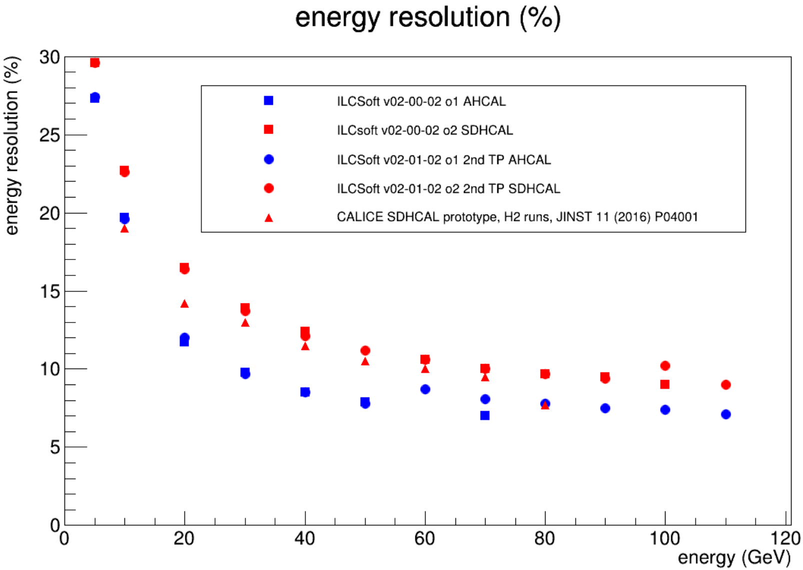
sim p (GeV)	mean (GeV)	sigma (GeV)	resolution (%)	discrepancy(%)
		o1		
1	0.85	0.34	**39.6%**	**15.1%**
2	1.64	0.61	**37.2%**	**18.3%**
5	4.37	1.19	**27.3%**	**12.5%**
10	9.11	1.80	**19.7%**	**8.9%**
20	19.68	2.31	**11.7%**	**1.6%**
30	29.75	2.91	**9.8%**	**0.8%**
40	39.75	3.39	**8.5%**	**0.6%**
50	49.50	3.94	**7.9%**	**1.0%**
70	69.95	4.90	**7.0%**	**0.1%**
		o2		
1	0.79	0.31	**38.6%**	**20.8%**
2	1.48	0.56	**38.2%**	**26.2%**
5	3.86	1.14	**29.6%**	**22.9%**
10	8.28	1.88	**22.7%**	**17.2%**
20	19.24	3.18	**16.5%**	**3.8%**
30	29.51	4.11	**13.9%**	**1.6%**
40	39.27	4.85	**12.4%**	**1.8%**
60	58.95	6.27	**10.6%**	**1.8%**
70	68.88	6.90	**10.0%**	**1.6%**
80	78.77	7.62	**9.7%**	**1.5%**
90	88.45	8.40	**9.5%**	**1.7%**
100	98.50	8.91	**9.0%**	**1.5%**

$$\text{resolution} = \frac{sigma}{mean}, \text{ discrepancy} = \frac{sim\,p - mean}{sim\,p}$$

2^{nd}TP

sim p (GeV)	mean (GeV)	sigma (GeV)	resolution (%)	discrepancy(%)
		o1		
1	0.66	0.23	**34.9%**	**34.0%**
2	1.63	0.62	**37.9%**	**18.5%**
5	4.37	1.19	**27.4%**	**12.7%**
10	9.12	1.79	**19.6%**	**8.9%**
20	19.66	2.35	**12.0%**	**1.7%**
30	29.73	2.90	**9.7%**	**0.9%**
40	39.76	3.37	**8.5%**	**0.6%**
50	49.71	3.90	**7.8%**	**0.6%**
60	59.82	4.33	**8.7%**	**17.2%**
70	69.94	4.84	**8.1%**	**14.5%**
80	80.13	5.46	**7.8%**	**12.6%**
90	90.63	6.02	**7.5%**	**11.0%**
100	101.20	6.71	**7.4%**	**9.4%**
110	112.00	7.21	**7.1%**	**8.0%**
		o2		
1	0.81	0.31	**38.4%**	**19.1%**
2	1.51	0.56	**37.2%**	**24.5%**
5	3.92	1.16	**29.6%**	**21.7%**
10	8.40	1.90	**22.6%**	**16.0%**
20	19.49	3.19	**16.4%**	**2.6%**
30	29.86	4.09	**13.7%**	**0.5%**
40	39.74	4.80	**12.1%**	**0.6%**
50	49.64	5.56	**11.2%**	**0.7%**
60	59.63	6.31	**10.6%**	**0.6%**
70	69.68	6.96	**10.0%**	**0.5%**
80	79.63	7.70	**9.7%**	**0.5%**
90	89.66	8.40	**9.4%**	**0.4%**
100	98.50	10.09	**10.2%**	**1.5%**
110	109.70	9.87	**9.0%**	**0.3%**

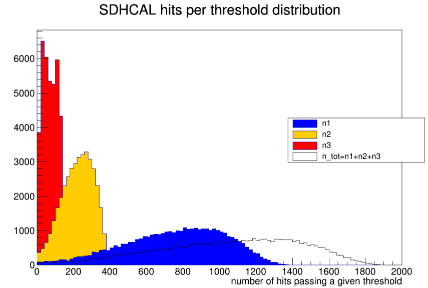
Resolution for the four scenarions: o1/o2 ⊗ $1^{st}/2^{nd}$ TP

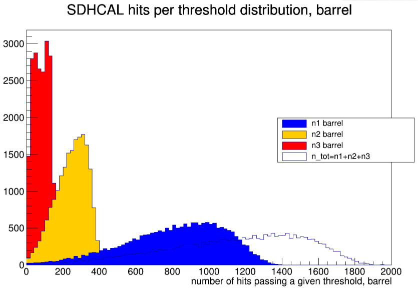


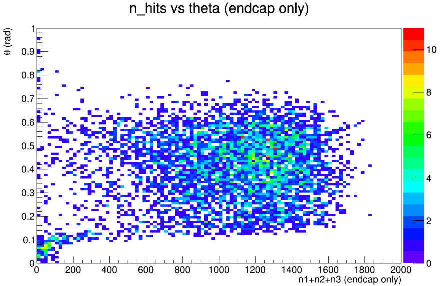
SDHCAL Hit Level Analysis

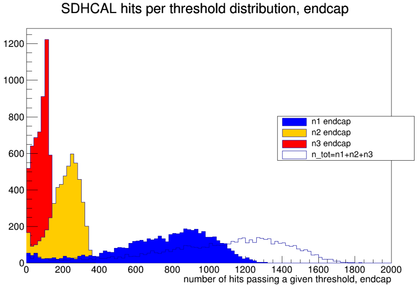
- For the single-hit level analysis, the dst datasets are not enough. The hit information is skimmed.
- An analysis of the rec dataset was needed.
- A copy of the rec dataset to CIEMAT was done.
- `/pool/calice3/data/MonteCarlo/sdhcal_validation/second_test_production/rec/o2v2`
- A customized LCTuple was produced out of rec dataset including the single hit information.
- As a reminder each hit in the SDHCAL tell us if the read energy on a given pad has passed one, two or three pre-set threshold.

SDHCAL Hit Level Analysis 110 GeV

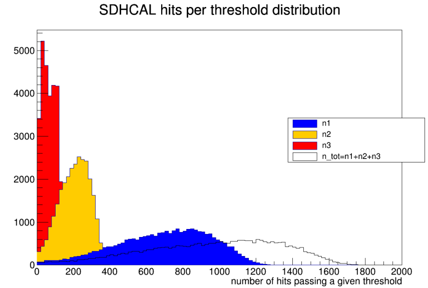


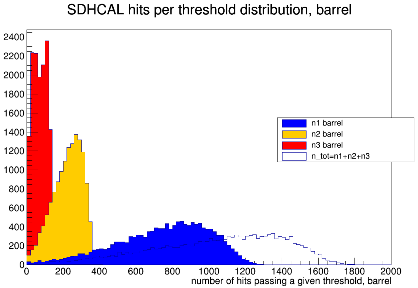


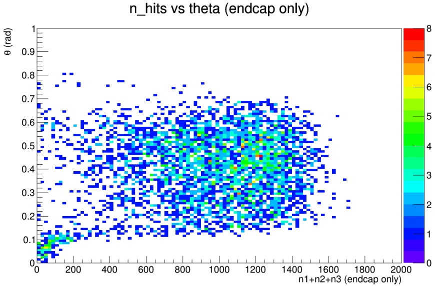


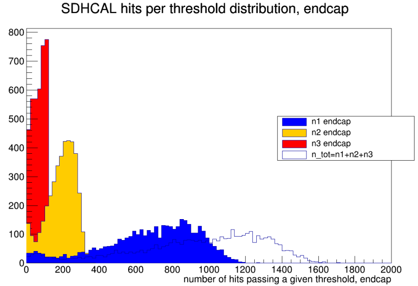


SDHCAL Hit Level Analysis 100 GeV

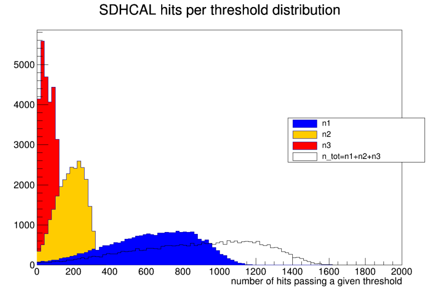


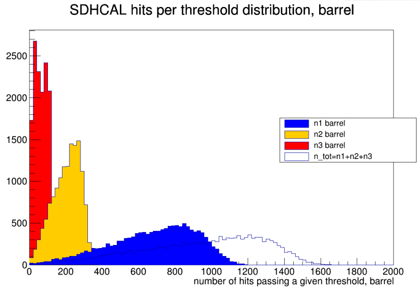


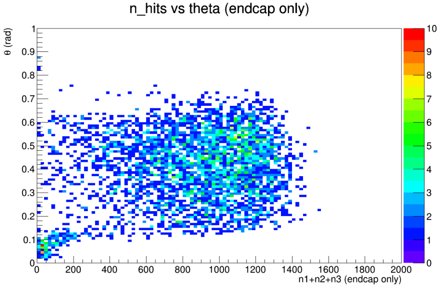


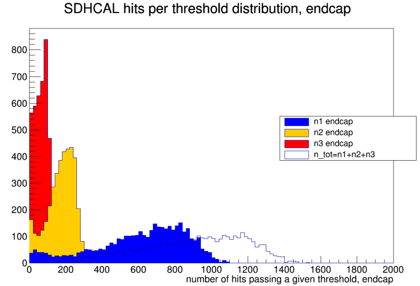


SDHCAL Hit Level Analysis 090 GeV

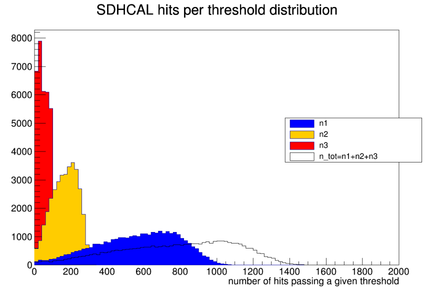


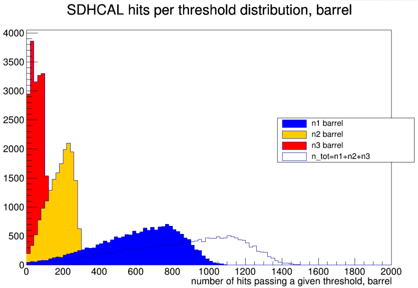


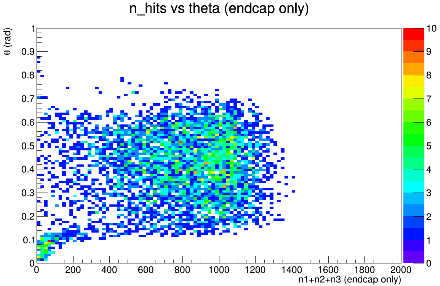


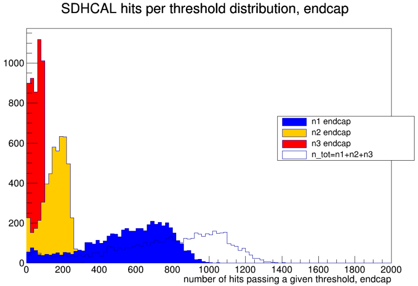


SDHCAL Hit Level Analysis 080 GeV

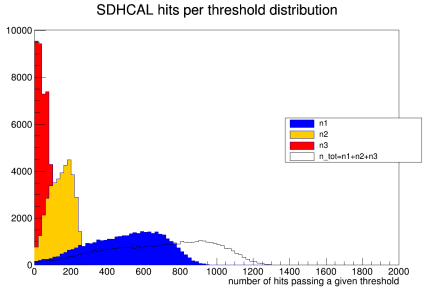


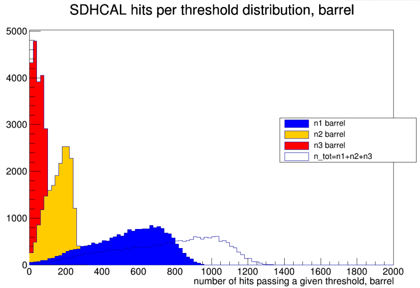


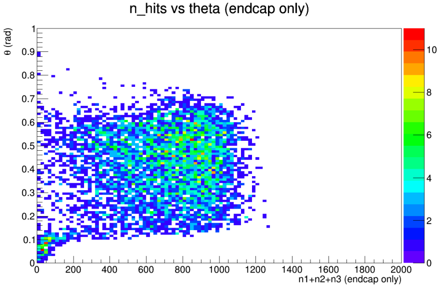


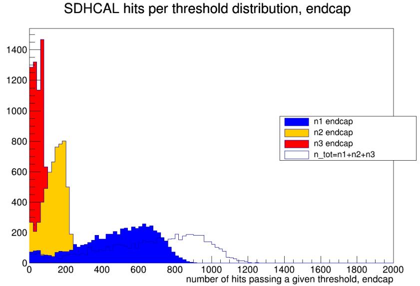


SDHCAL Hit Level Analysis 070 GeV

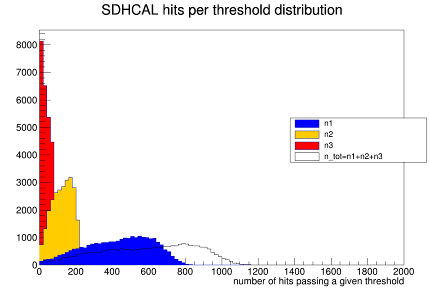


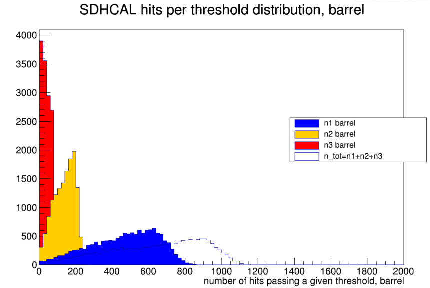


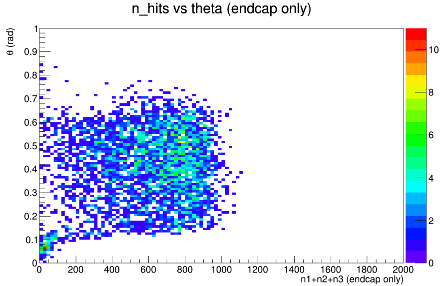


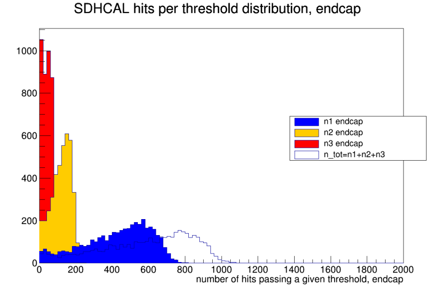


SDHCAL Hit Level Analysis 060 GeV

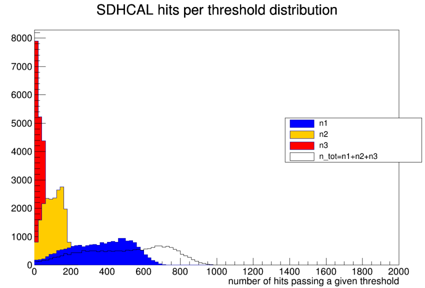


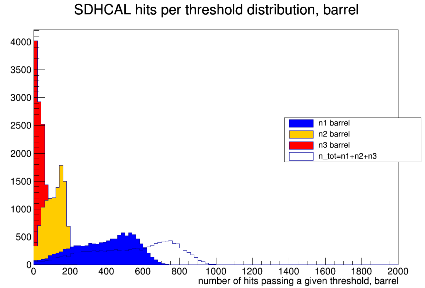


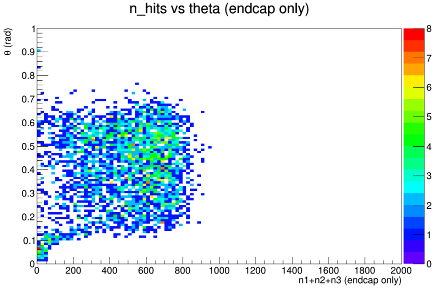


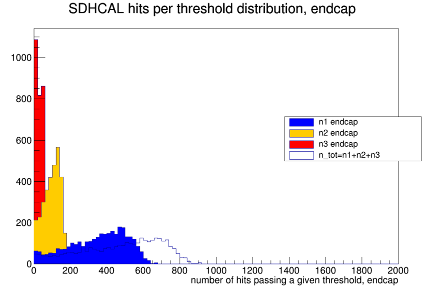


SDHCAL Hit Level Analysis 050 GeV

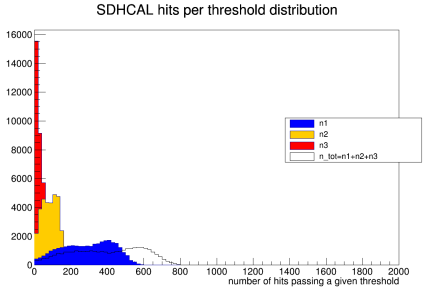


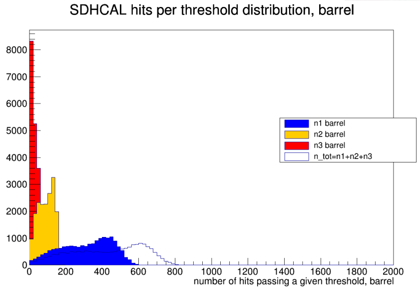


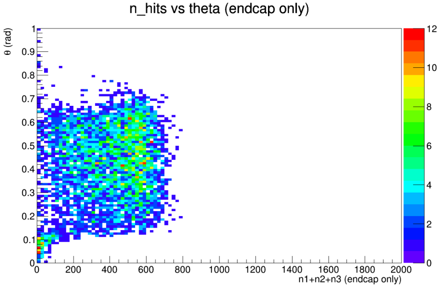


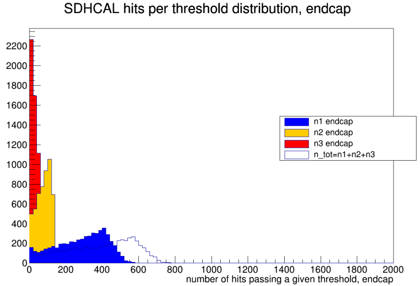


SDHCAL Hit Level Analysis 040 GeV

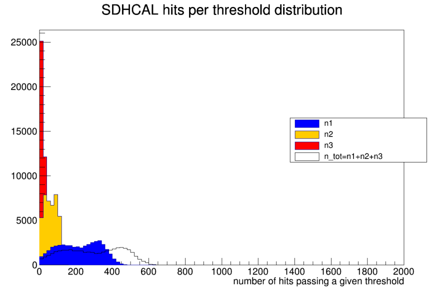


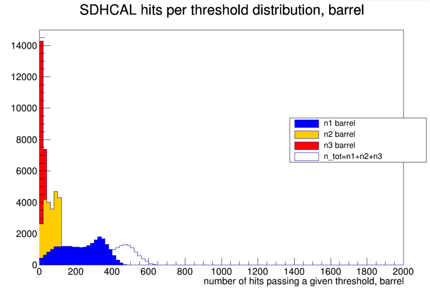


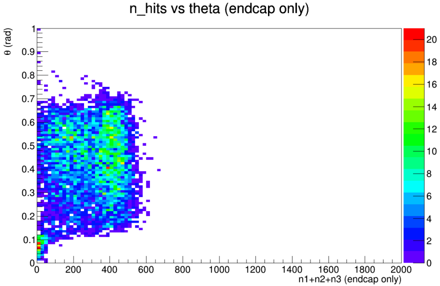


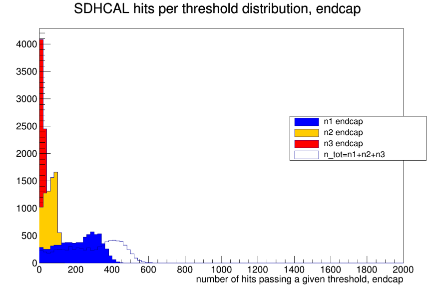


SDHCAL Hit Level Analysis 030 GeV

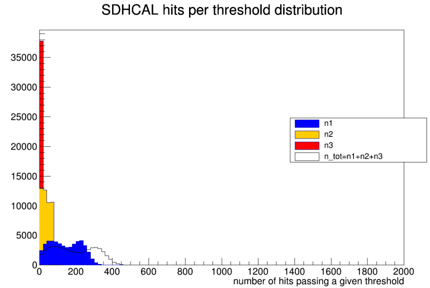


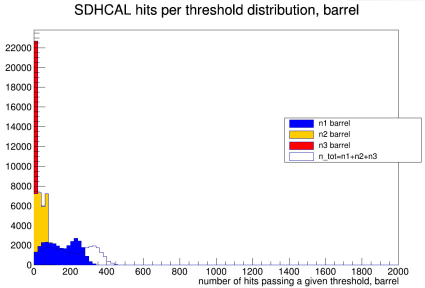


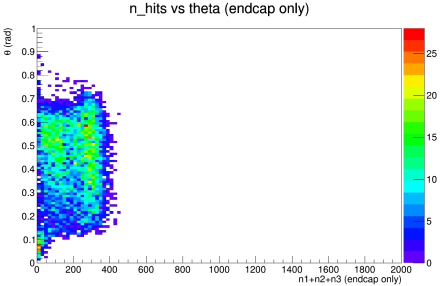


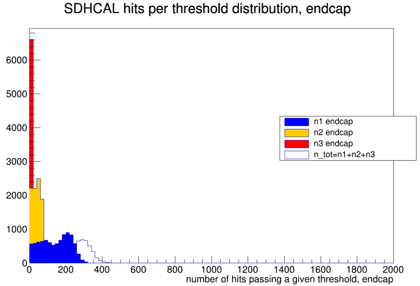


SDHCAL Hit Level Analysis 020 GeV

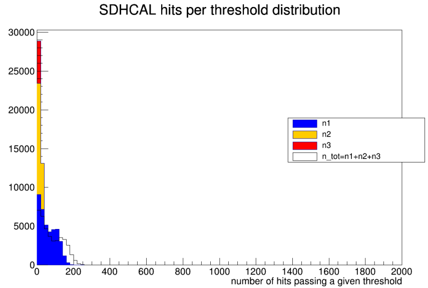


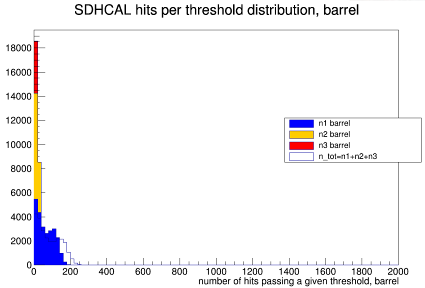


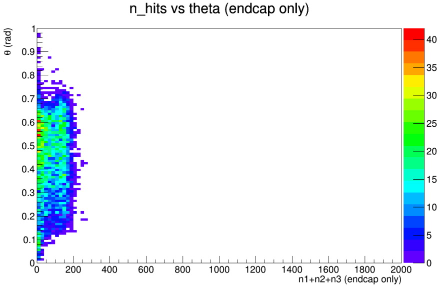


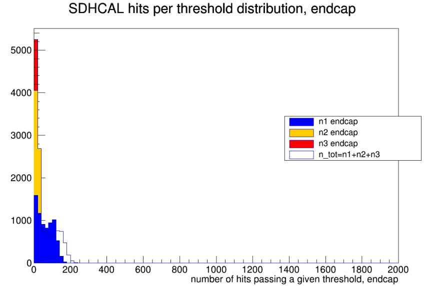


SDHCAL Hit Level Analysis 010 GeV

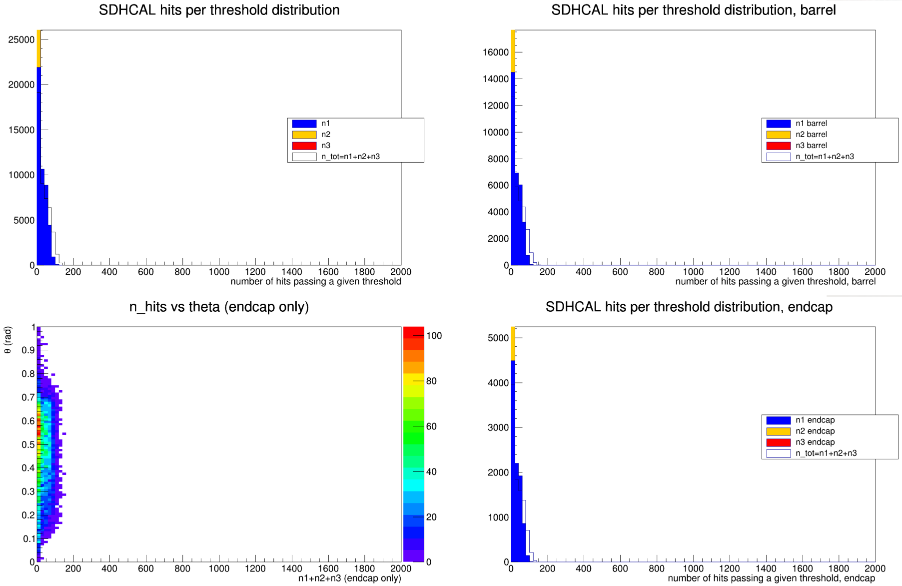




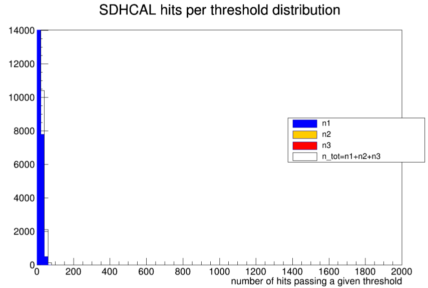


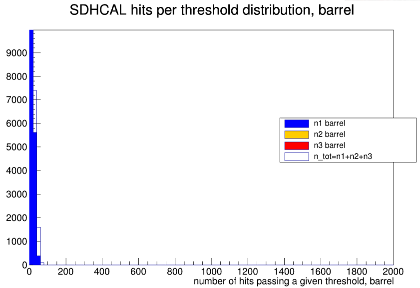


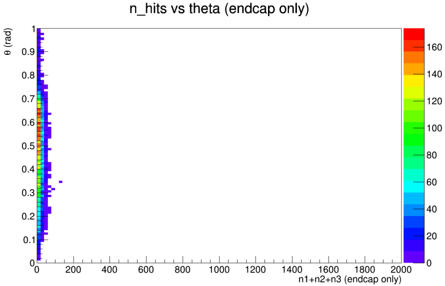
SDHCAL Hit Level Analysis 005 GeV

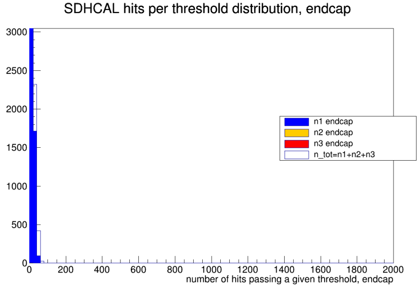


SDHCAL Hit Level Analysis 002 GeV

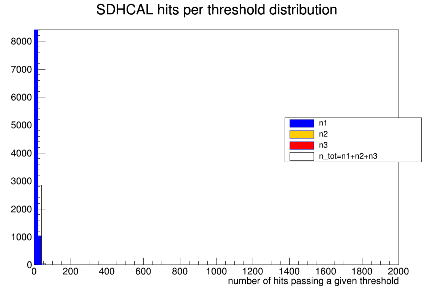


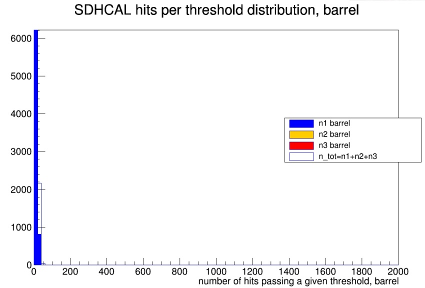


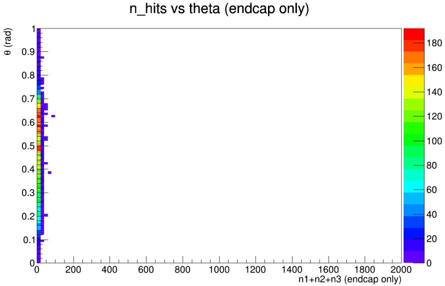


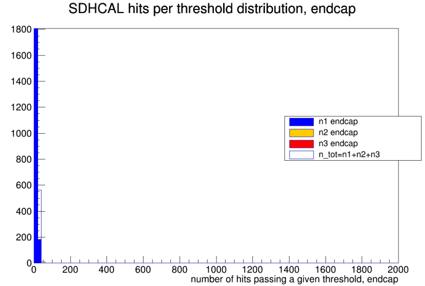


SDHCAL Hit Level Analysis 001 GeV









Conclusions

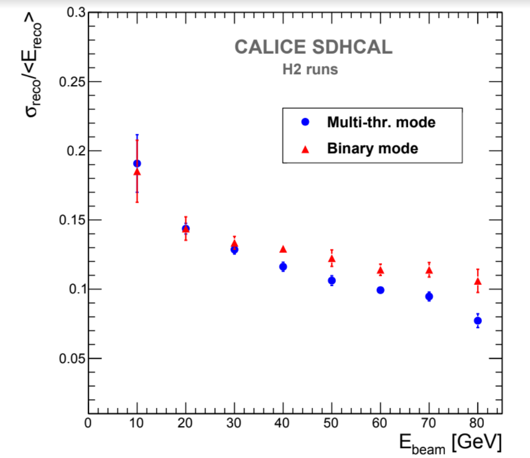
- No relevant difference has been observed with the new test-sample for the SDHCAL/AHCAL performance.
- Next steps:
  - extra variables to check the SDHCAL calibration are under scrutiny.
  - study the SDHCAL local reconstructed objects (cluster performance).
- key point about SDHCAL in ilcsoft[1]:
  - Geant4 physics model used in ilcsoft is QGSP-Bert which is not ideal to simulate SDHCAL.
  - FTF-BIC is the more appropriate for SDHCAL.

[1] https://geant4.web.cern.ch/node/155

Backup

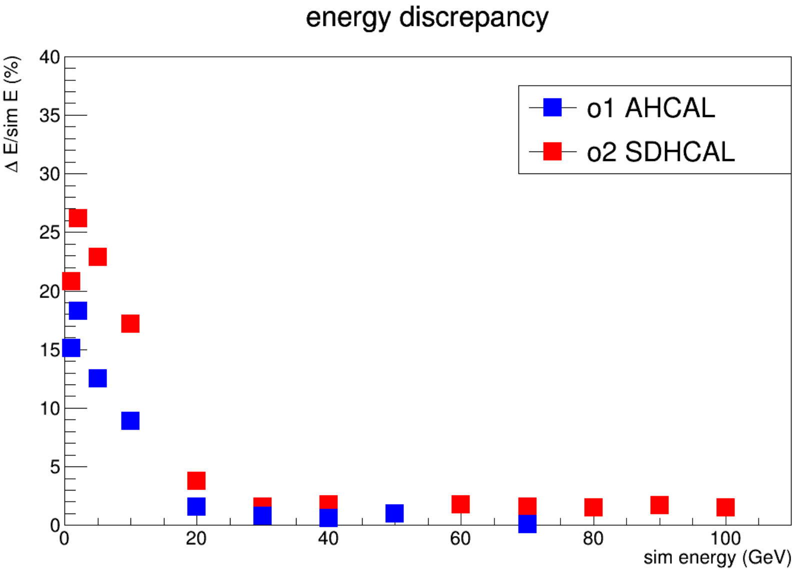
Backup

Only SDHCAL resolution observed in test-beams



CALICE collaboration, First results of the CALICE SDHCAL technological prototype, JINST **11** (2016) P04001.

Comparison for the two scenarions, discrepancy.



Crystalball fit

$$f(x;\alpha,n,\bar{x},\sigma) = N \cdot \begin{cases} \exp(-\frac{(x-\bar{x})^2}{2\sigma^2}), & \text{for } \frac{x-\bar{x}}{\sigma} > -\alpha \\ A \cdot (B - \frac{x-\bar{x}}{\sigma})^{-n}, & \text{for } \frac{x-\bar{x}}{\sigma} \leqslant -\alpha \end{cases}$$

$$A = \left(\frac{n}{|\alpha|}\right)^n \cdot \exp\left(-\frac{|\alpha|^2}{2}\right),$$

$$B = \frac{n}{|\alpha|} - |\alpha|,$$

$$N = \frac{1}{\sigma(C+D)},$$

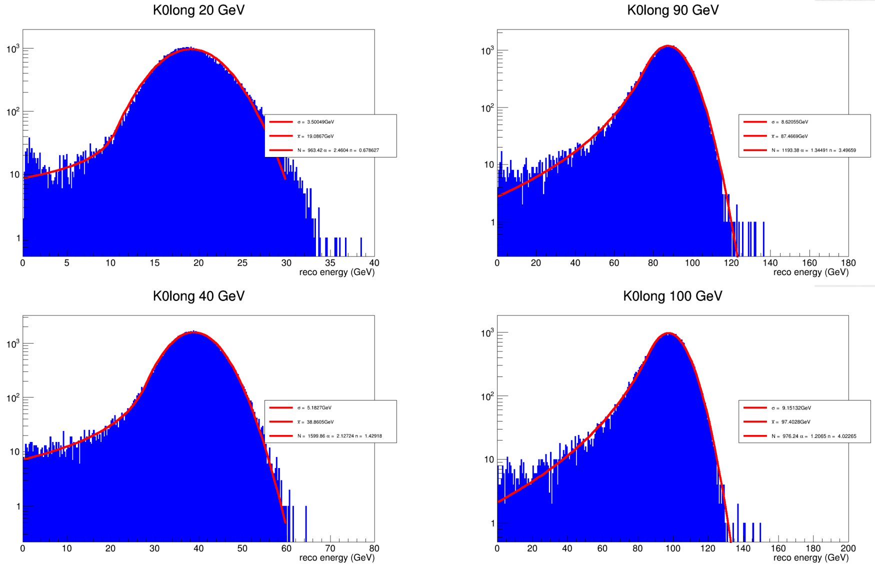
$$C = \frac{n}{|\alpha|} \cdot \frac{1}{n-1} \cdot \exp\left(-\frac{|\alpha|^2}{2}\right),$$

$$D = \sqrt{\frac{\pi}{2}}\left(1 + \operatorname{erf}\left(\frac{|\alpha|}{\sqrt{2}}\right)\right).$$

```
 FCN=342.074 FROM MIGRAD    STATUS=CONVERGED     184 CALLS         185 TOTAL
                     EDM=2.61519e-08    STRATEGY= 1  ERROR MATRIX UNCERTAINTY   0.3 per cent
  EXT PARAMETER                                   STEP         FIRST
  NO.   NAME      VALUE            ERROR          SIZE      DERIVATIVE
   1  N            2.00731e+03   9.14867e+00  -2.32131e-02   2.47481e-05
   2  mean         5.83022e+01   2.70121e-02   2.65898e-05  -4.92050e-03
   3  sigma        6.59899e+00   2.21181e-02   1.22279e-04   1.43070e-02
   4  alpha        1.80238e+00   2.83231e-02   7.01543e-05  -6.05402e-03
   5  n            1.97606e+00   1.01879e-01  -9.94635e-05   1.27298e-03
60 GeV thismax2.1e+03 mean=58 sigma=6.6 error=11%
```

https://en.wikipedia.org/wiki/Crystal_Ball_function

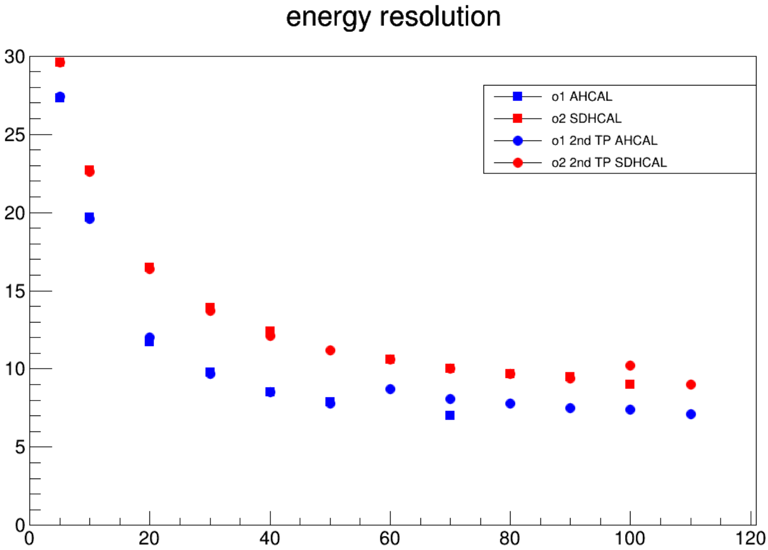
Crystalball fit, K_L^0, o2



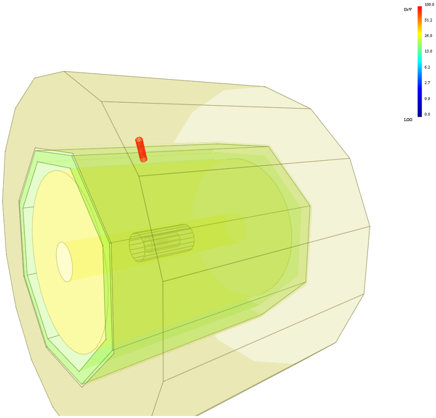
Summary, K_L^0, o2

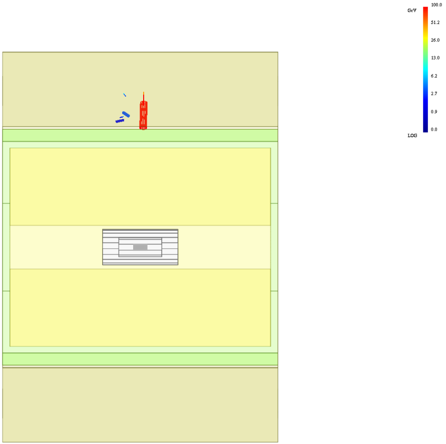
sim energy (GeV)	CB $\bar{x}$ (GeV)	CB σ (GeV)	$\frac{\sigma}{E}$ (%)
1	0.79	0.3	30
2	1.4	0.53	26
5	3.5	1.2	25
10	7.8	2.1	21
20	19	3.5	18
30	29	4.4	15
40	39	5.2	13
60	58	6.6	11
70	68	7.2	10
80	78	7.9	9.8
90	87	8.6	9.6
100	97	9.2	9.2

Summary Resolution, K_L^0



First look at the datasets for the SDHCAL validation, event display K_L^0 110 GeV, energy deposit in SDHCAL





List of variables available in the standard LCTuple

```
// Fixed size dimensions of array or collections stored in the TTree if any.

   // Declaration of leaf types
   Int_t           evevt;
   Int_t           evrun;
   Float_t         evwgt;
   Long64_t        evtim;
   Float_t         evsig;
   Float_t         evene;
   Float_t         evpoe;
   Float_t         evpop;
   Int_t           evnch;
   Char_t          evpro[1];   //[evnch]
   Int_t           nmcp;
   Int_t           mcori[807];   //[nmcp]
   Int_t           mcpdg[807];   //[nmcp]
   Int_t           mcgst[807];   //[nmcp]
   Int_t           mcsst[807];   //[nmcp]
   Float_t         mcvtx[807];   //[nmcp]
   Float_t         mcvty[807];   //[nmcp]
   Float_t         mcvtz[807];   //[nmcp]
   Float_t         mcepx[807];   //[nmcp]
   Float_t         mcepy[807];   //[nmcp]
   Float_t         mcepz[807];   //[nmcp]
   Float_t         mcmox[807];   //[nmcp]
   Float_t         mcmoy[807];   //[nmcp]
   Float_t         mcmoz[807];   //[nmcp]
   Float_t         mcmas[807];   //[nmcp]
   Float_t         mcene[807];   //[nmcp]
   Float_t         mccha[807];   //[nmcp]
   Float_t         mctim[807];   //[nmcp]
   Float_t         mcspx[807];   //[nmcp]
   Float_t         mcspy[807];   //[nmcp]
   Float_t         mcspz[807];   //[nmcp]
   Int_t           mccf0[807];   //[nmcp]
   Int_t           mccf1[807];   //[nmcp]
   Int_t           mcpa0[807];   //[nmcp]
   Int_t           mcpa1[807];   //[nmcp]
   Int_t           mcda0[807];   //[nmcp]
   Int_t           mcda1[807];   //[nmcp]
   Int_t           mcda2[807];   //[nmcp]
   Int_t           mcda3[807];   //[nmcp]
   Int_t           mcda4[807];   //[nmcp]
   Int_t           nrec;
   Int_t           rcori[41];   //[nrec]
   Int_t           rccid[41];   //[nrec]
   Int_t           rctyp[41];   //[nrec]
   Float_t         rccov[41][10];   //[nrec]
   Float_t         rcrpx[41];   //[nrec]
   Float_t         rcrpy[41];   //[nrec]
   Float_t         rcrpz[41];   //[nrec]
   Float_t         rcgpi[41];   //[nrec]
   Int_t           rcpiu[41];   //[nrec]
   Int_t           rcnpi[41];   //[nrec]
   Int_t           rcfpi[41];   //[nrec]
   Float_t         rcmox[41];   //[nrec]
   Float_t         rcmoy[41];   //[nrec]
   Float_t         rcmoz[41];   //[nrec]
   Float_t         rcmas[41];   //[nrec]
   Float_t         rcene[41];   //[nrec]
   Float_t         rccha[41];   //[nrec]
   Int_t           rcntr[41];   //[nrec]
   Int_t           rcncl[41];   //[nrec]
   Int_t           rcnrp[41];   //[nrec]
   Int_t           rcftr[41];   //[nrec]
```

```
   Int_t           rcftr[41];   //[nrec]
   Int_t           rcvts[41];   //[nrec]
   Int_t           rcvte[41];   //[nrec]
   Int_t           rccom[41];   //[nrec]
   Int_t           npid;
   Int_t           pityp[212];   //[npid]
   Int_t           pipdg[212];   //[npid]
   Float_t         pillh[212];   //[npid]
   Int_t           pialg[212];   //[npid]
   Int_t           ntrk;
   Int_t           trori[36];   //[ntrk]
   Int_t           trtyp[36];   //[ntrk]
   Float_t         trch2[36];   //[ntrk]
   Int_t           trndf[36];   //[ntrk]
   Float_t         tredx[36];   //[ntrk]
   Float_t         trede[36];   //[ntrk]
   Float_t         trrih[36];   //[ntrk]
   Int_t           trshn[36][12];   //[ntrk]
   Int_t           trnts[36];   //[ntrk]
   Int_t           trfts[36];   //[ntrk]
   Int_t           trsip[36];   //[ntrk]
   Int_t           trsfh[36];   //[ntrk]
   Int_t           trslh[36];   //[ntrk]
   Int_t           trsca[36];   //[ntrk]
   Int_t           ntrst;
   Int_t           tsloc[144];   //[ntrst]
   Float_t         tsdze[144];   //[ntrst]
   Float_t         tsphi[144];   //[ntrst]
   Float_t         tsome[144];   //[ntrst]
   Float_t         tszze[144];   //[ntrst]
   Float_t         tstnl[144];   //[ntrst]
   Float_t         tscov[144][15];   //[ntrst]
   Float_t         tsrpx[144];   //[ntrst]
   Float_t         tsrpy[144];   //[ntrst]
   Float_t         tsrpz[144];   //[ntrst]
   Int_t           nsth;
   Int_t           stori[1];   //[nsth]
   Int_t           stci0[1];   //[nsth]
   Int_t           stci1[1];   //[nsth]
   Double_t        stpox[1];   //[nsth]
   Double_t        stpoy[1];   //[nsth]
   Double_t        stpoz[1];   //[nsth]
   Float_t         stedp[1];   //[nsth]
   Float_t         sttim[1];   //[nsth]
   Float_t         stmox[1];   //[nsth]
   Float_t         stmoy[1];   //[nsth]
   Float_t         stmoz[1];   //[nsth]
   Float_t         stptl[1];   //[nsth]
   Int_t           stmcp[1];   //[nsth]
   Int_t           nsch;
   Int_t           scori[1];   //[nsch]
   Int_t           scci0[1];   //[nsch]
   Int_t           scci1[1];   //[nsch]
   Float_t         scpox[1];   //[nsch]
   Float_t         scpoy[1];   //[nsch]
   Float_t         scpoz[1];   //[nsch]
   Float_t         scene[1];   //[nsch]
   Int_t           r2mnrel;
   Int_t           r2mf[169];   //[r2mnrel]
   Int_t           r2mt[169];   //[r2mnrel]
   Float_t         r2mw[169];   //[r2mnrel]
```