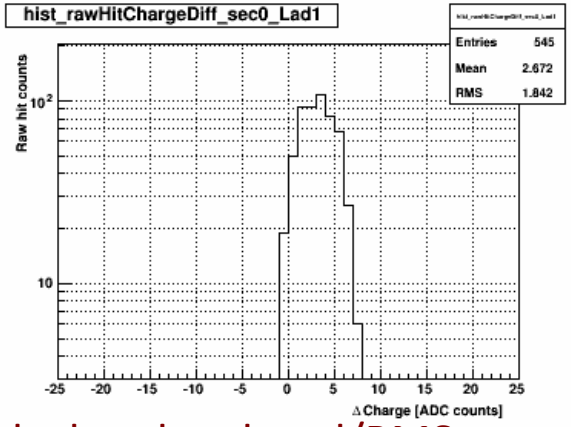
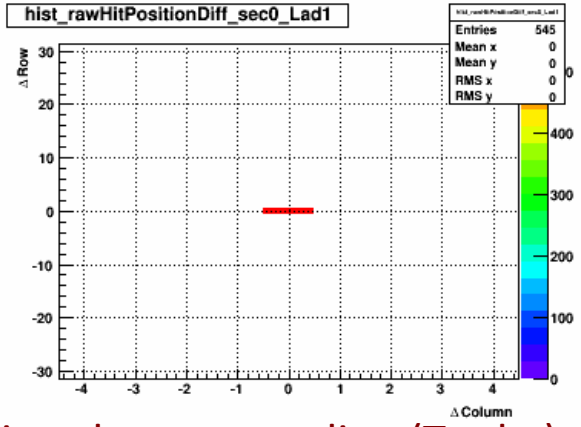
IST ZS/nonZS data check

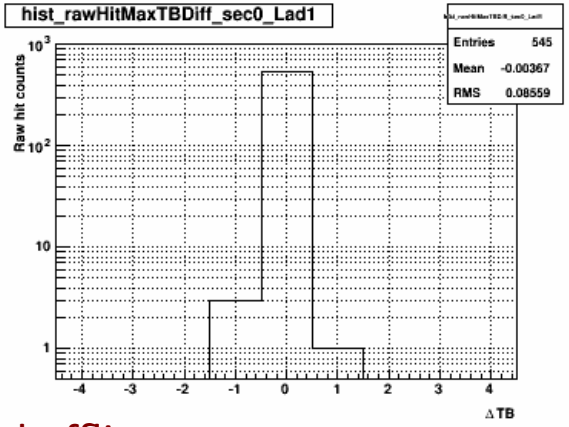
Yaping Wang (UIC)

ZS/non-ZS data comparison (#15040003+#15040005+#15040006)

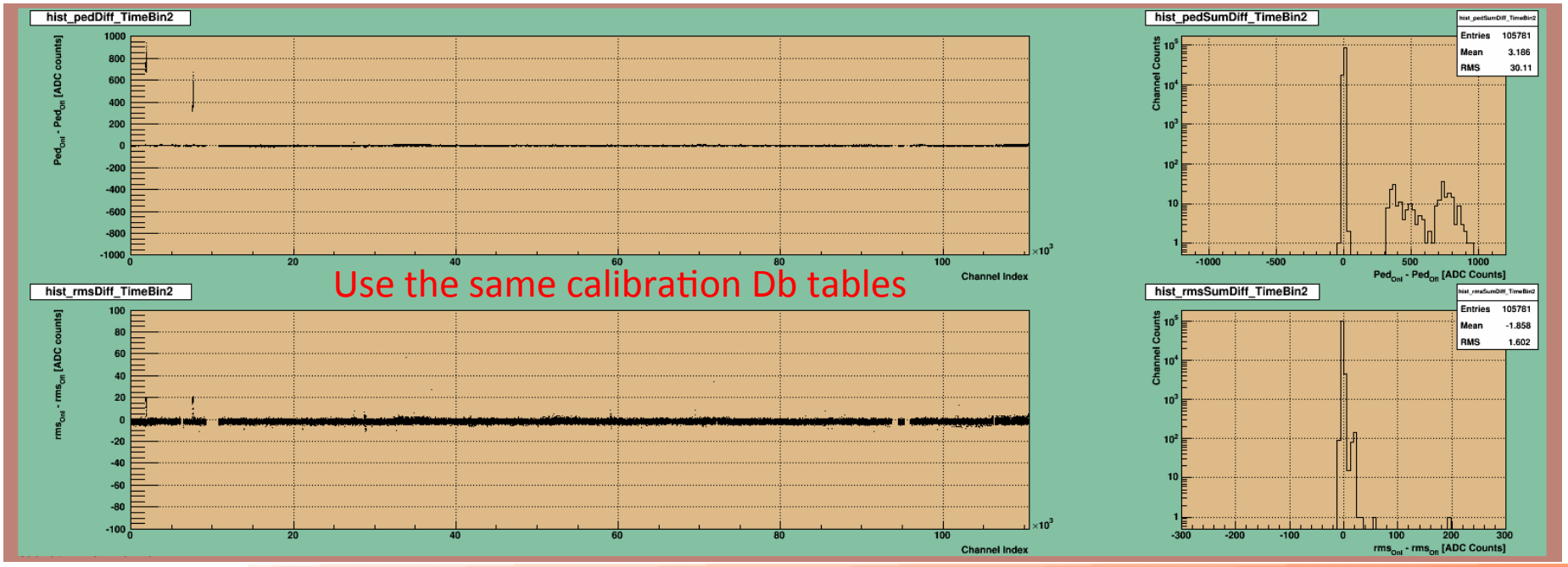
Matched raw hits comparison between ZS and non-ZS data:





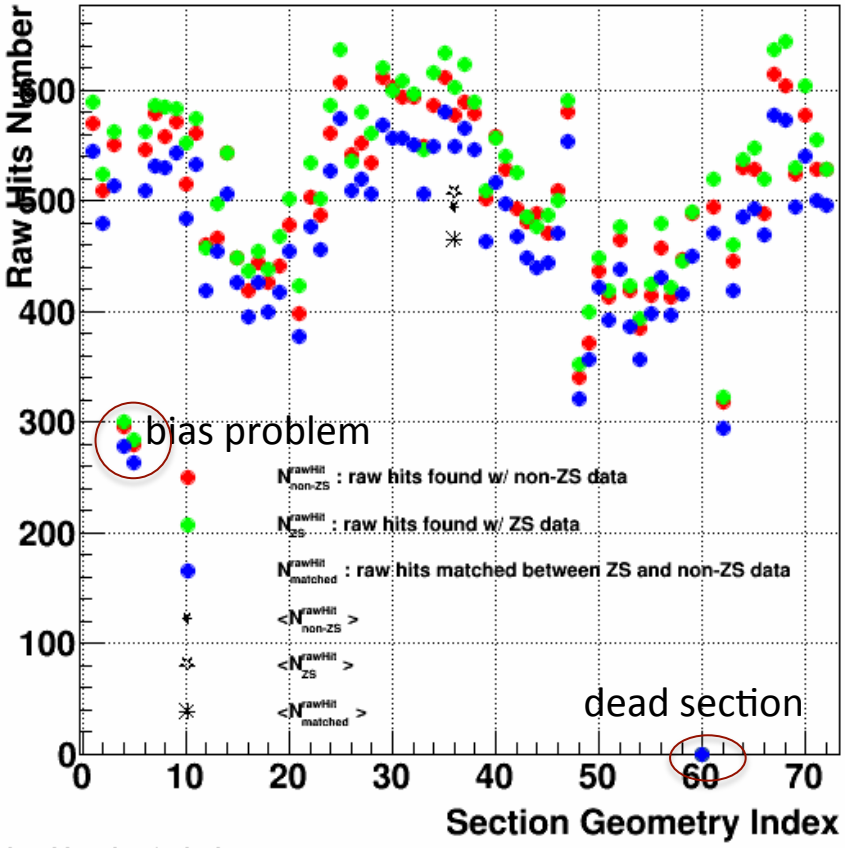


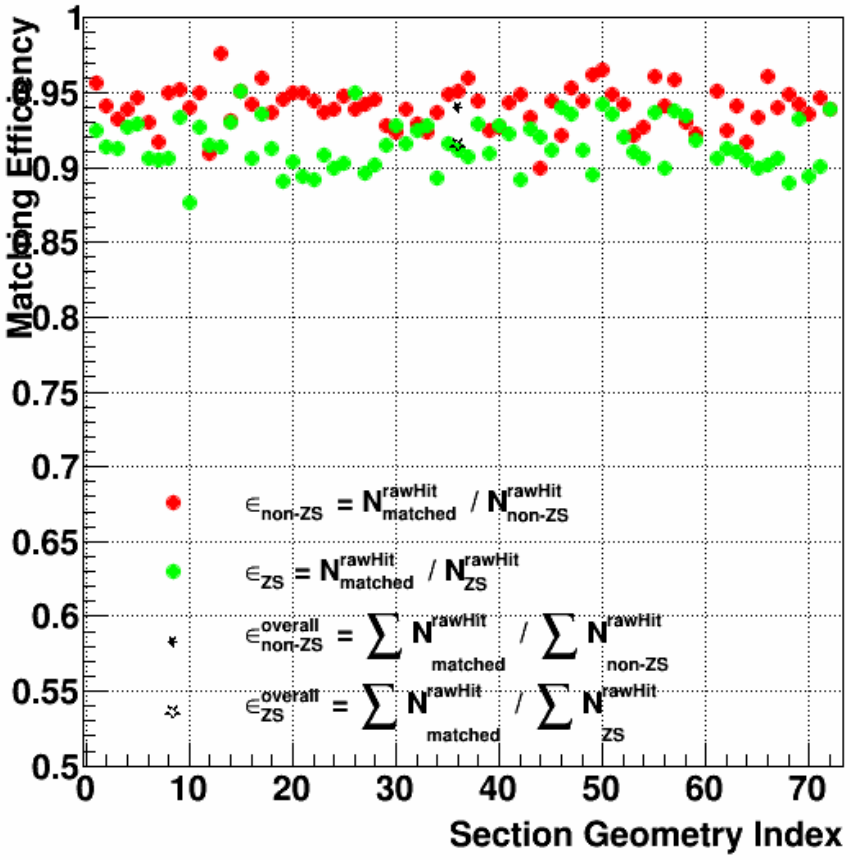
Calculated pedestal/RMS comparison between online (Tonko) and offline:



ZS/non-ZS data comparison (#15040003+#15040005+#15040006)

Found raw hits/matching efficiency vs different position (readout section):

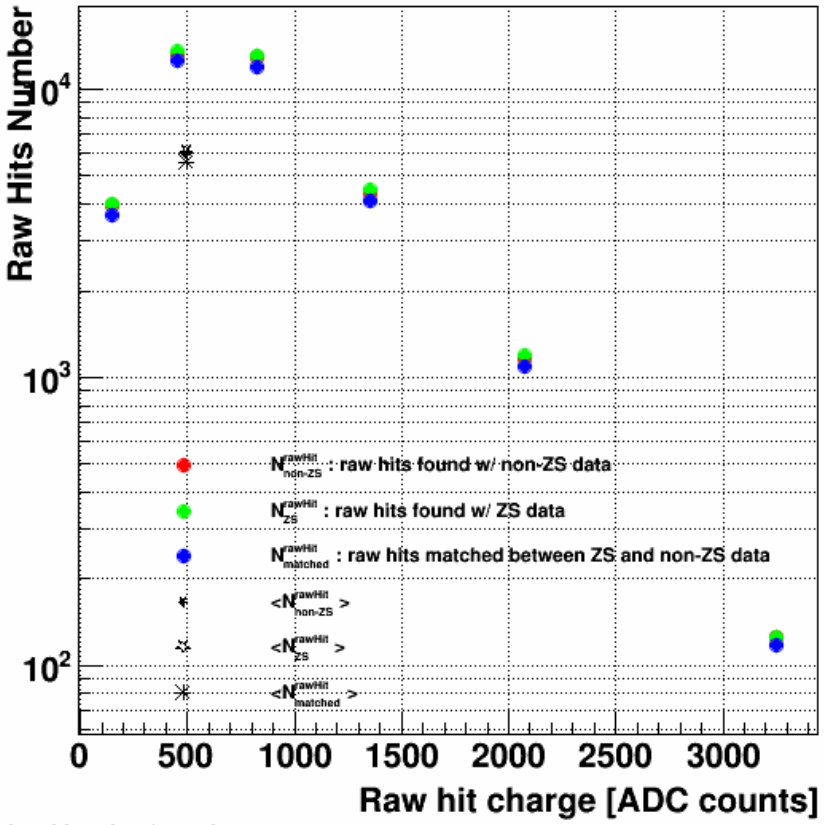


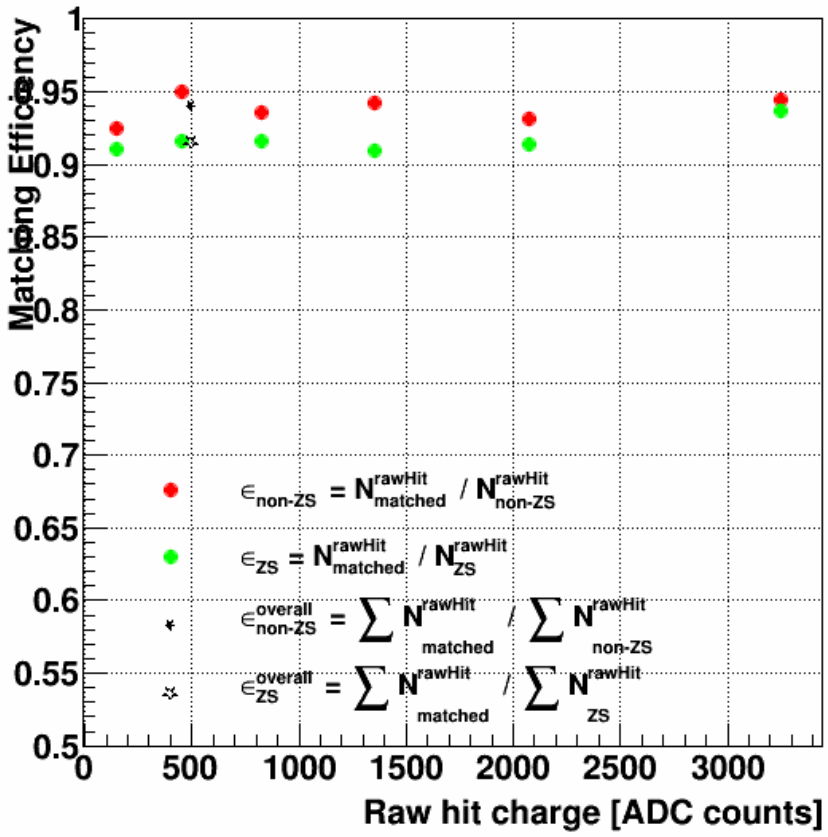


The number of raw hits found in ZS data is a bit bigger than in non-ZS data (raw hits loss in non-ZS data). Because of the difference of the pedestal between online and offline calculation methods?

ZS/non-ZS data comparison (#15040003+#15040005+#15040006)

Found raw hits/matching efficiency vs ADC range:





The number of raw hits found in ZS data is a bit bigger than in non-ZS data (raw hits loss in non-ZS data).

Six ADC ranges defined: (0, 300], (300, 600], (600, 1050], (1050, 1650], (1650, 2500], (2500, 4000]

Pedestal/RMS calculation methods comparison

Online (using pedestal run):

1. Accumulate event by event for each channel each time bin:

```
p->ped[arm][apv][ch][tb] += (float) adc ;
p->rms[arm][apv][ch][tb] += (float) (adc * adc) ;
p->cou[arm][apv][ch][tb]++ ;
```

2. Calculate pedestal/rms for each channel each time bin:

```
pp = ped->ped[arm][apv][ch][t]/(double) ped->cou[arm][apv][ch][t];
rr  = ped->rms[arm][apv][ch][t]/(double) ped->cou[arm][apv][ch][t];
rr = sqrt(rr - pp*pp) ;
ped->ped[arm][apv][ch][t] = pp ;
ped->rms[arm][apv][ch][t] = rr ;
```

3. Average pedestal/RMS over all time bins for each channel:

```
ped += p->ped[arm][apv][c][t] ;
rms += p->rms[arm][apv][c][t] ;
cou_tb++ ;
ped /= cou_tb ;
rms /= cou_tb ;
```

4. Threshold setting for each channel:

3.0

```
p->thr[arm][apv][c] = (u_short) (ped + rms * n_sigma + 0.5) ;
```

5. Do ZS for each channel each time bin (if any adc > p->thr):

```
*d16++ = (short)((float)f[i_save+i].adc - p_thr->ped[arm][apv]
[ch][i] + 0.5);
```

Offline (using non-ZS data of physics run):

1. Fill histograms event by event for each channel each time bin:

```
int code = kIstNumTimeBins * elecId + t;
TH1F* histPed = mHistPedVec[ code ];
histPed->Fill( (float)adc );
```

2. Exclude possible signal entries for each channel each time bin:

```
TH1F *histPed = *mHistPedVecIter;
float meanPed = histPed->GetMean();
float rmsPed  = histPed->GetRMS();
histPed->GetXaxis()->SetRangeUser(meanPed-mPedCut*rmsPed,
                meanPed+mPedCut*rmsPed); // mPedCut = 3.0
```

3. Get Mean/RMS as pedestal/RMS for each channel each time bin:

```
meanPed    = histPed->GetMean();
rmsPed     = histPed->GetRMS();
```

4. Average pedestal/RMS over all time bins for each channel:

```
averagePed += pedestal[TB]/numTimeBins;
averageRms += rms[TB]/numTimeBins;
```

5. Do pedestal subtraction:

```
signalCorrected[channel][timebin]    =
(float)signalUnCorrected[channel][timebin] - mPedVec[elecId];
```