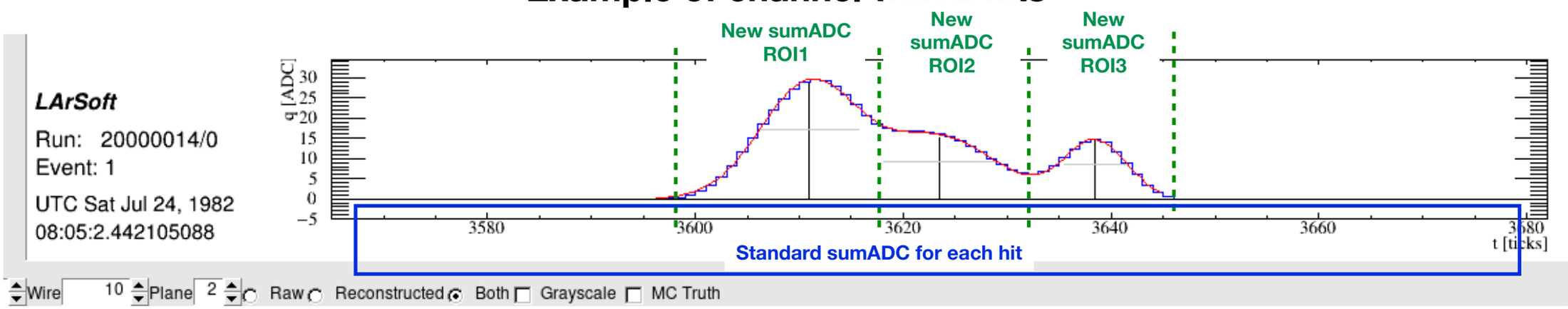
Hit reconstruction issue

A.Meregaglia (LP2I Bordeaux)

Hit reconstruction

- In the energy reconstruction study we noticed that for channels with only one hit the Gaussian integral and the ADC sum are almost equivalent.
- However, for channels with more than one hit, SumADC function integrates the whole ROI therefore several hits on the same channel have differen integral but the same SumADC.
- A quick fix was applied in a custom version of larreco and tested on the same events.
- The sumADC integrates ADC in a range of ± 2 σ around the Gaussian mean.
- In case the integration ranges of two hits overlap a correction is applied in order to avoid ADC double counting, using the rms of the two Gaussians as weights to split the common interval.

Example of channel with 3 hits

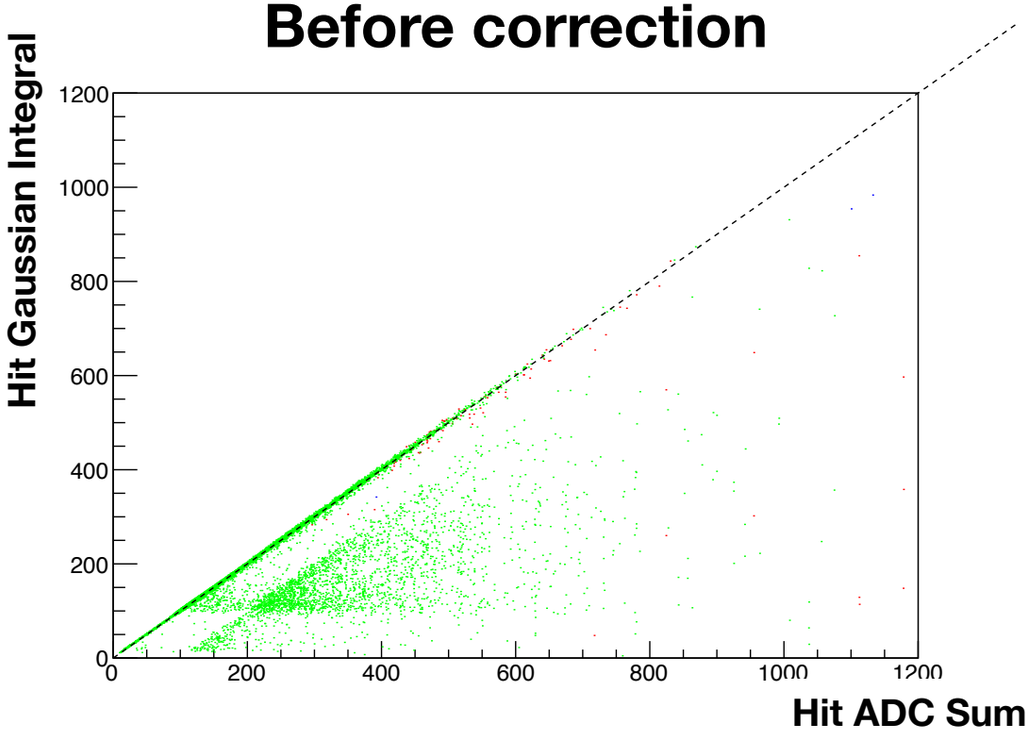


The 3 hits are correctly fitted with different Gaussian integrals, but each one has the same ADC sum of the whole ROI. In the modified version each one has a different sumADC

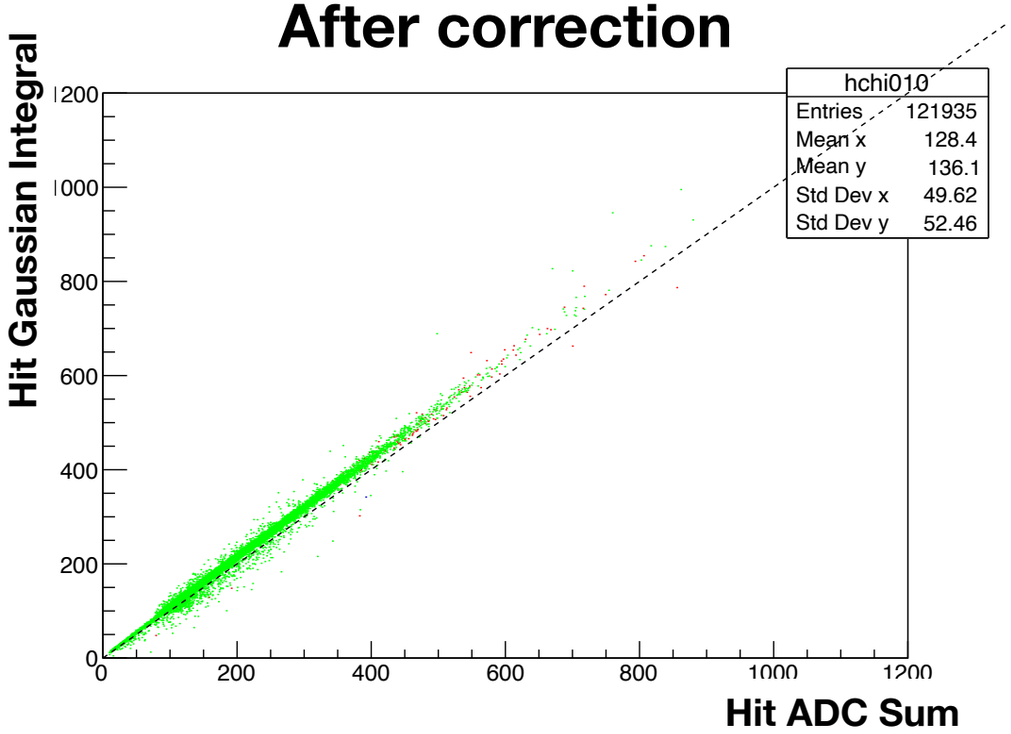
Hits study

1.5 GeV Muon simulation

Before correction



After correction



Line: y=x

Blue: chi2=-1 (0.004%)

Red: chi2>10 (0.1%)

Green: 0<chi2<10 (99.9%)

Quick and dirty algoritm

- All the modifications are done in the *GausHitFinder_Module.cc* file. I can give you my modified file if you want to, but here below I underline the main ideas.

1. I define new some variables to gat info about previous and next hit

```
//ANS define new variables
            float nextpeak(0);
            float prevpeak(0);
            float nextpeakSig(0);
            float prevpeakSig(0);
            float nsigmaADC(2.0);
            float newright(0);
             float newleft(0);

//ANS get prev and next peaks
                  if(numHits==0)
                    {
                      newleft =-9999;
                      newright=9999;
                      nextpeak=0;
                      prevpeak=0;
                      nextpeakSig=0;
                      prevpeakSig=0;
                    }
                  if(numHits<nGausForFit-1)
                    {
                      nextpeak = (peakParamsVec.at(numHits+1)).peakCenter;
                      nextpeakSig = (peakParamsVec.at(numHits+1)).peakSigma;
                   }
                  if(numHits>0)
                    {
                      prevpeak= (peakParamsVec.at(numHits-1)).peakCenter;
                      prevpeakSig = (peakParamsVec.at(numHits-1)).peakSigma;
                   }
```

Quick and dirty algoritm

2. I define limits to compute ROI according to the gaussian fit (2 sigmas but it can be different).

```
  //ANS change limits in ADC counts
              std::vector<float>::const_iterator sumStartItrAns = range.begin() + peakMean -nsigmaADC*peakWidth;
              std::vector<float>::const_iterator sumEndItrAns = range.begin() + peakMean +nsigmaADC*peakWidth;
```

3. Check if there is overlap in the different hit ROI and use sigma as weight to redefine boundaries with some protections.

```
 if(nGausForFit>1)
                  {
                    if(numHits>0)
                      {
                        if((peakMean-nsigmaADC*peakWidth)<(prevpeak+nsigmaADC*prevpeakSig))
                          {
                           float difPeak = peakMean-prevpeak;
                           float weightpeak = prevpeakSig/(prevpeakSig+peakWidth);
                           sumStartItrAns=range.begin() + prevpeak + difPeak*weightpeak;
                           newleft=prevpeak + difPeak*weightpeak;
                          }
                      }
                    if(numHits<nGausForFit-1)
                      {
                        if((peakMean+nsigmaADC*peakWidth)>(nextpeak-nsigmaADC*nextpeakSig))
                          {
                           float difPeak = nextpeak-peakMean;
                           float weightpeak = peakWidth/(nextpeakSig+peakWidth);
                           sumEndItrAns=range.begin() + peakMean + difPeak*weightpeak;
                           newright= peakMean + difPeak*weightpeak;
                          }
                      }
                  }

  //Add some protections
                if(newright-newleft < 0)
                 continue;
                if(sumStartItrAns<sumStartItr)
                  sumStartItrAns=sumStartItr;
                if(sumEndItrAns>sumEndItr)
                  sumEndItrAns=sumEndItr;
```

Quick and dirty algoritm

4. Fill a new ADCSum integral in the readout structure replacing the old one (to avoid modifing my analysis code but of course an additional variable should be created keeping both of them).

```
                double sumADCAns = std::accumulate(sumStartItrAns, sumEndItrAns, 0.);

// ok, now create the hit
                recob::HitCreator hitcreator(
                  *wire,                          // wire reference
                  wid,                            // wire ID
                  startT + roiFirstBinTick,       // start_tick TODO check
                  endT + roiFirstBinTick,         // end_tick TODO check
                  peakWidth,                      // rms
                  peakMean + roiFirstBinTick,     // peak_time
                  peakMeanErr,                    // sigma_peak_time
                  peakAmp,                        // peak_amplitude
                  peakAmpErr,                     // sigma_peak_amplitude
                  charge,                         // hit_integral
                  chargeErr,                      // hit_sigma_integral
                  //                  sumADC,                         // summedADC FIXME
                  sumADCAns,                      // summedADC FIXME
                  nGausForFit,                    // multiplicity
                  numHits,                        // local_index TODO check that the order is correct
                  chi2PerNDF,                     // goodness_of_fit
                  NDF                             // dof
                );
```