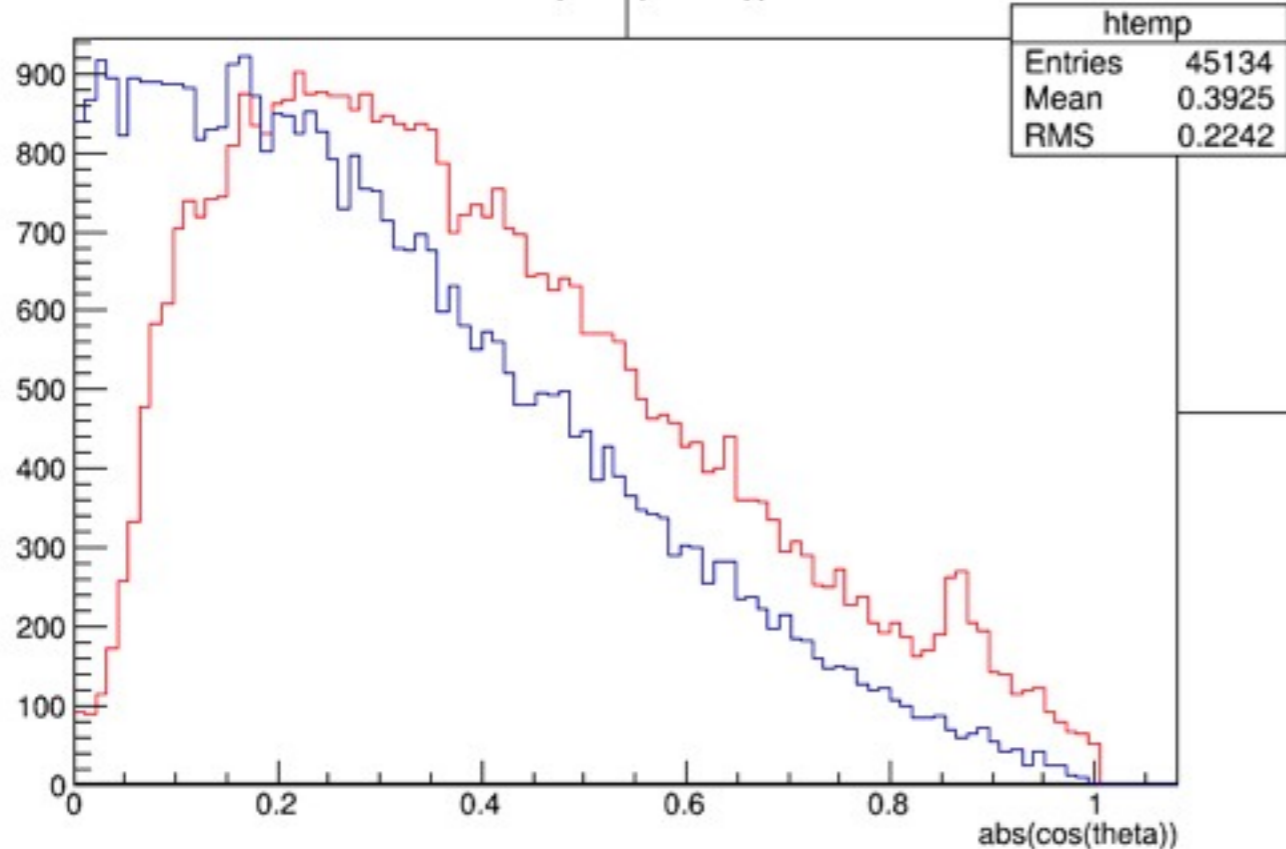


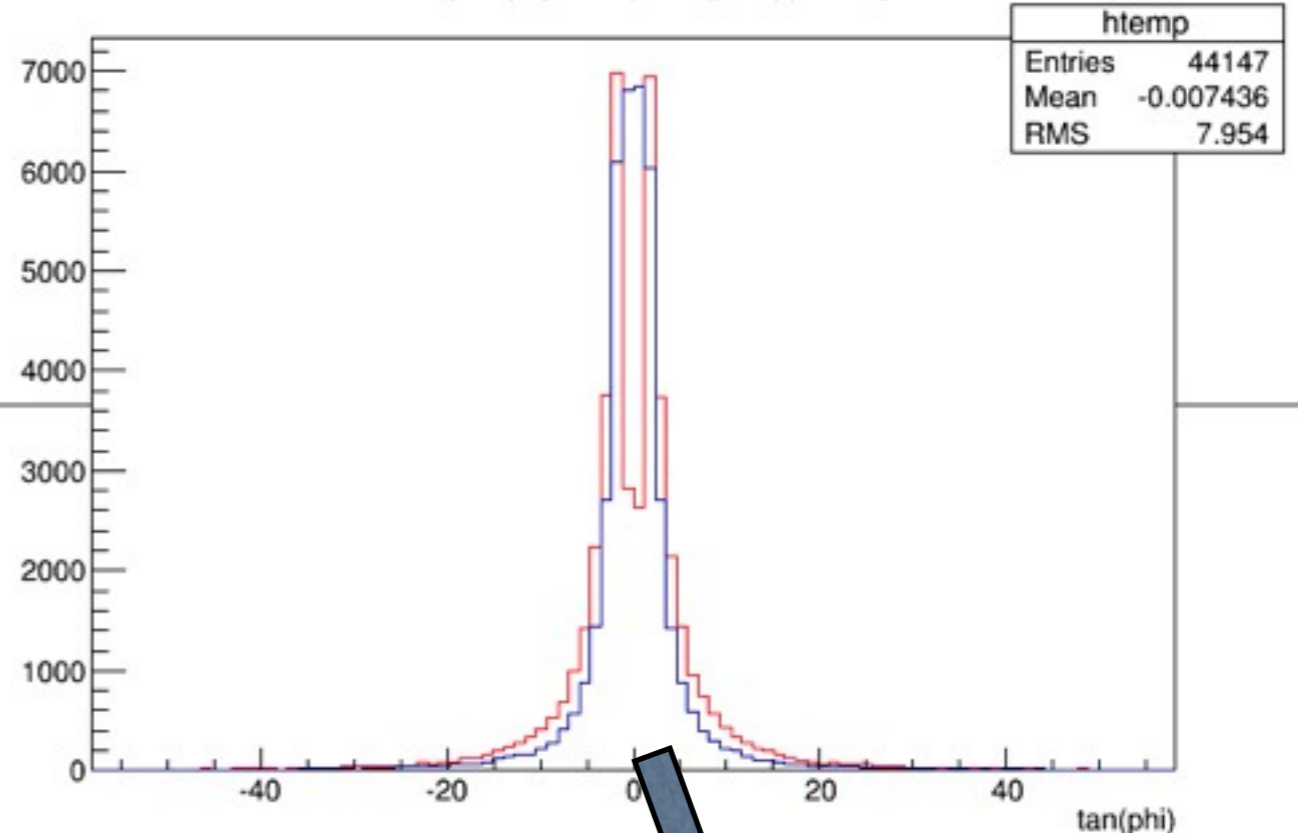
**ClusterCrawler
+Track3DKalmanSPS
+TrackStitching**

Eric, 26-Sep-2013

abs(cos(theta))

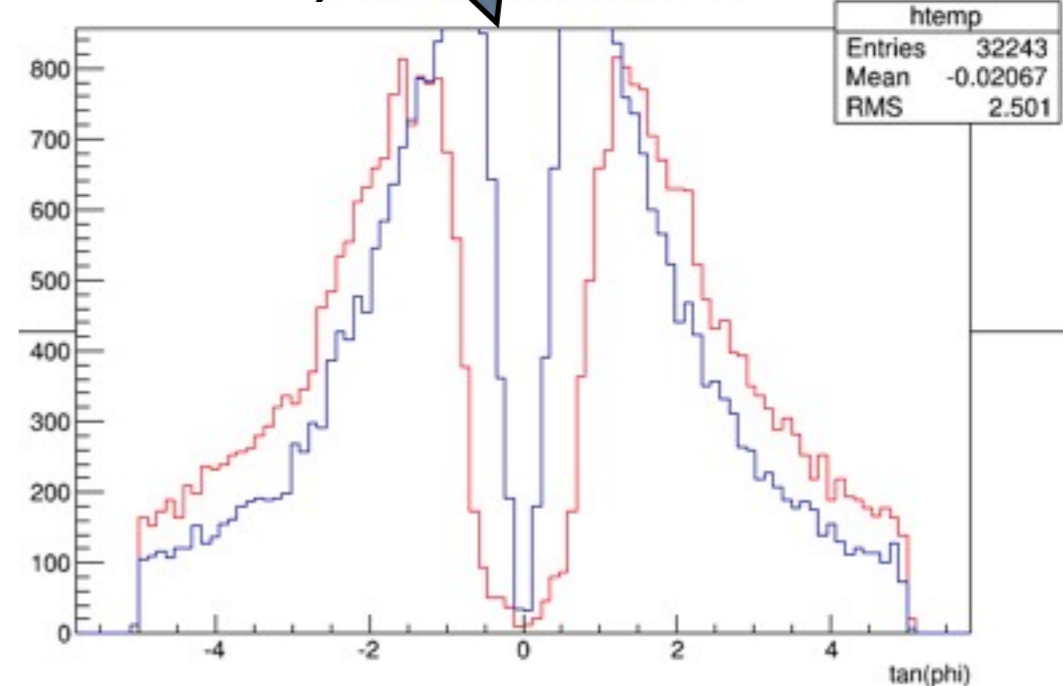


tan(phi) {abs(tan(phi))<50}



(zoomed)

tan(phi) {abs(tan(phi))<5}



“abs” to make sense of vector not matter in truth and reco
Reco:45134 tracks, MC:42216
Red is Reco

“tan” to do the same

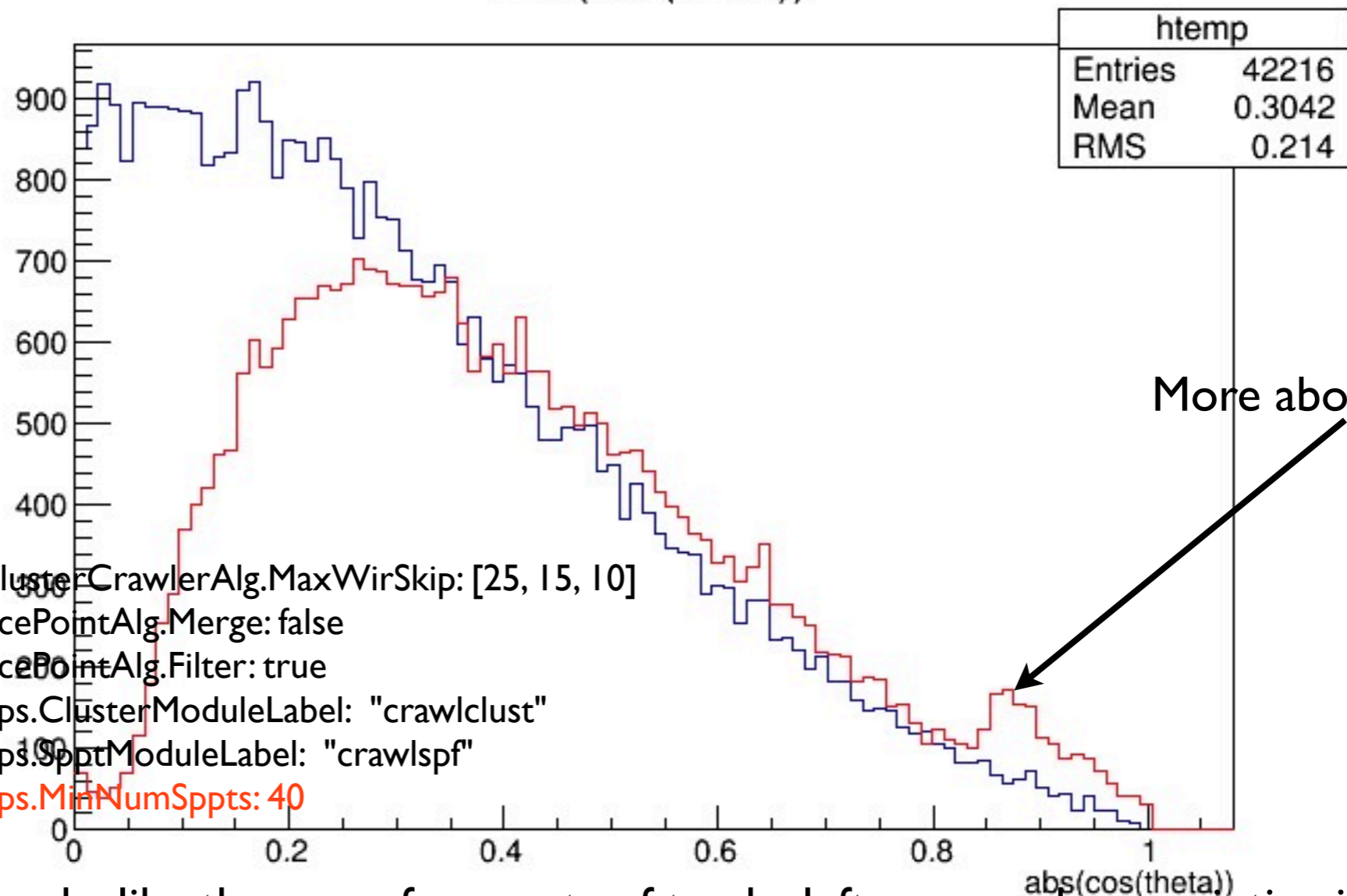
```

physics.producers.crawlclust.ClusterCrawlerAlg.MaxWirSkip: [25, 15, 10]
physics.producers.crawlspf.SpacePointAlg.Merge: false
physics.producers.crawlspf.SpacePointAlg.Filter: true
physics.producers.crawltrkkalsps.ClusterModuleLabel: "crawlclust"
physics.producers.crawltrkkalsps.SpptModuleLabel: "crawlspf"
physics.producers.crawltrkkalsps.MinNumSppts: 20
  
```

Nsppts > 40 |costh| plot

Shows that the long, robust parts of tracks are properly recon'd -- at least non-vertical ones are.

abs(cos(theta))



More about this later

physics.producers.crawlclust.ClusterCrawlerAlg.MaxWirSkip: [25, 15, 10]
physics.producers.crawlspf.SpacePointAlg.Merge: false
physics.producers.crawlspf.SpacePointAlg.Filter: true
physics.producers.crawltrkkalsps.ClusterModuleLabel: "crawlclust"
physics.producers.crawltrkkalsps.SpacePointModuleLabel: "crawlspf"
physics.producers.crawltrkkalsps.MinNumSppts: 40

Looks like there are fragments of tracks left over, perhaps pointing in the right direction, that could be stitched to the robust ones.

What's next

- Looking just at the $|\cos\theta|$ plot I suspect that there are many tracks at that are pointing in roughly the right direction. They are broken up by clustering.
- Remember, `Track3DKalmanSPS` produces a track for every spacepoint clump it's given.
- Every clump originates from time triplets of hits on the three views, with hits considered by cluster. So, if any view breaks up a cluster one gets an extra track.

Thus

- Can I lower the threshold of spacepoints required in a good track?
 - **In the end, the goal is to remove all cosmic Associated Hits. Literally, art::Assns of Track to Hit.**
- Simultaneously stitch together tracks pointing in the same direction
- ergo, TrackStitcher_module now in LArSoft.
- reads in recob::Tracks, puts onto the evt:
 - `vector<Tracks>`
 - `vector<PtrVector<Tracks>>` // component tracks of above

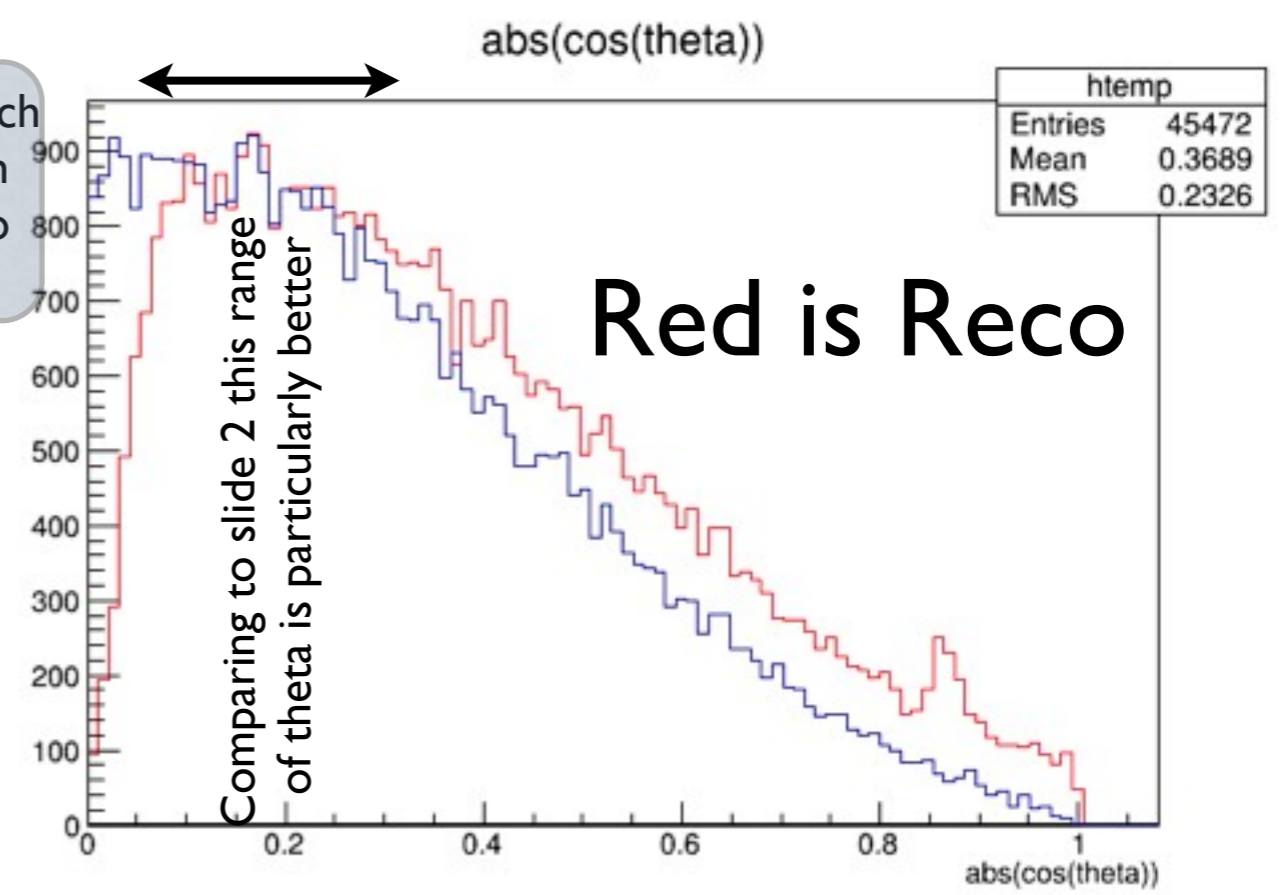
TrackStitcher

```
TimeModule> run: 1 subRun: 0 event: 2304 stitch TrackStitcher 0.00332499
TimeModule> run: 1 subRun: 0 event: 2304 TriggerResults TriggerResultInserter 0.000219107
TimeModule> run: 1 subRun: 0 event: 2304 HackDayTrackAnaKalmanSPS TrackAnaHack 16.5086
TimeEvent> run: 1 subRun: 0 event: 2304 46.7779
TimeModule> run: 1 subRun: 0 event: 2305 caldata CalWireMicroBooNE 12.0602
TimeModule> run: 1 subRun: 0 event: 2305 ffthit FFTHitFinder 22.3602
TimeModule> run: 1 subRun: 0 event: 2305 crawlclust ClusterCrawler 0.430057
TimeModule> run: 1 subRun: 0 event: 2305 crawlspf SpacePointFinder 23.4711
%MSG-w Track3DKalmanSPS: Track3DKalmanSPS:crawltrkkalsps 25-Sep-2013 10:34:53 CDT run: 1 subRun: 0 event: 2305
There are 106 Spacepoint PtrVectors (spacepoint clumps) in this event.
%MSG
TimeModule> run: 1 subRun: 0 event: 2305 crawltrkkalsps Track3DKalmanSPS 8.91103
%MSG-w TrackStitcher.beginning: TrackStitcher:stitch 25-Sep-2013 10:35:02 CDT run: 1 subRun: 0 event: 2305
There are 11 Tracks in this event before stitching.
%MSG
%MSG-w TrackStitcher.end: TrackStitcher:stitch 25-Sep-2013 10:35:02 CDT run: 1 subRun: 0 event: 2305
There are 6 Tracks in this event after stitching.
%MSG
TimeModule> run: 1 subRun: 0 event: 2305 stitch TrackStitcher 0.00455403
TimeModule> run: 1 subRun: 0 event: 2305 TriggerResults TriggerResultInserter 0.000183105
TimeModule> run: 1 subRun: 0 event: 2305 HackDayTrackAnaKalmanSPS TrackAnaHack 11.5709
```

Result of Stitcher

0.98, 10.0 gives similar result, which is nice: we can cut cosmics with good efficiency and not eat into neutrino epts.

```
reco: [ caldata, ffthit, crawlclust, crawlspf, crawltrkkalsps, stitch ]
ana: [ HackDayTrackAnaKalmanSPS ]
...
physics.producers.crawlclust.ClusterCrawlerAlg.MaxWirSkip: [25, 15, 10]
physics.producers.crawlspf.SpacePointAlg.Merge: false
physics.producers.crawlspf.SpacePointAlg.Filter: true
physics.producers.crawltrkkalsps.ClusterModuleLabel: "crawlclust"
physics.producers.crawltrkkalsps.SpptModuleLabel: "crawlspf"
physics.producers.crawltrkkalsps.MinNumSppts: 10 # was 20
#physics.producers.crawltrkkalsps.DistanceU: 5.
physics.producers.stitch.TrackModuleLabel: "crawltrkkalsps"
physics.producers.stitch.SpptModuleLabel: "crawlspf"
physics.producers.stitch.CosAngTolerance: 0.85 # pretty loose
physics.producers.stitch.SpptSepTolerance: 15.0
physics.analyzers.HackDayTrackAnaKalmanSPS.TrackModuleLabel: "stitch"
```



Reco: 45472 tracks, MC: 42216

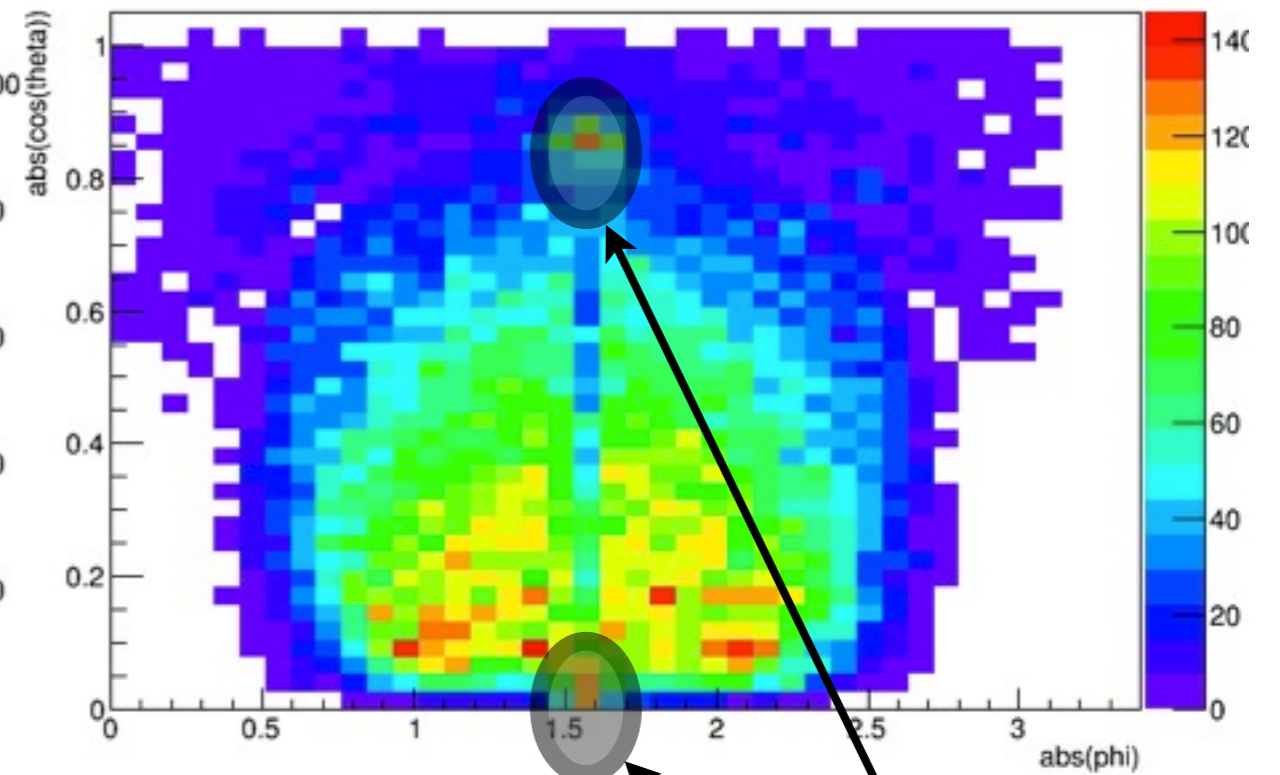
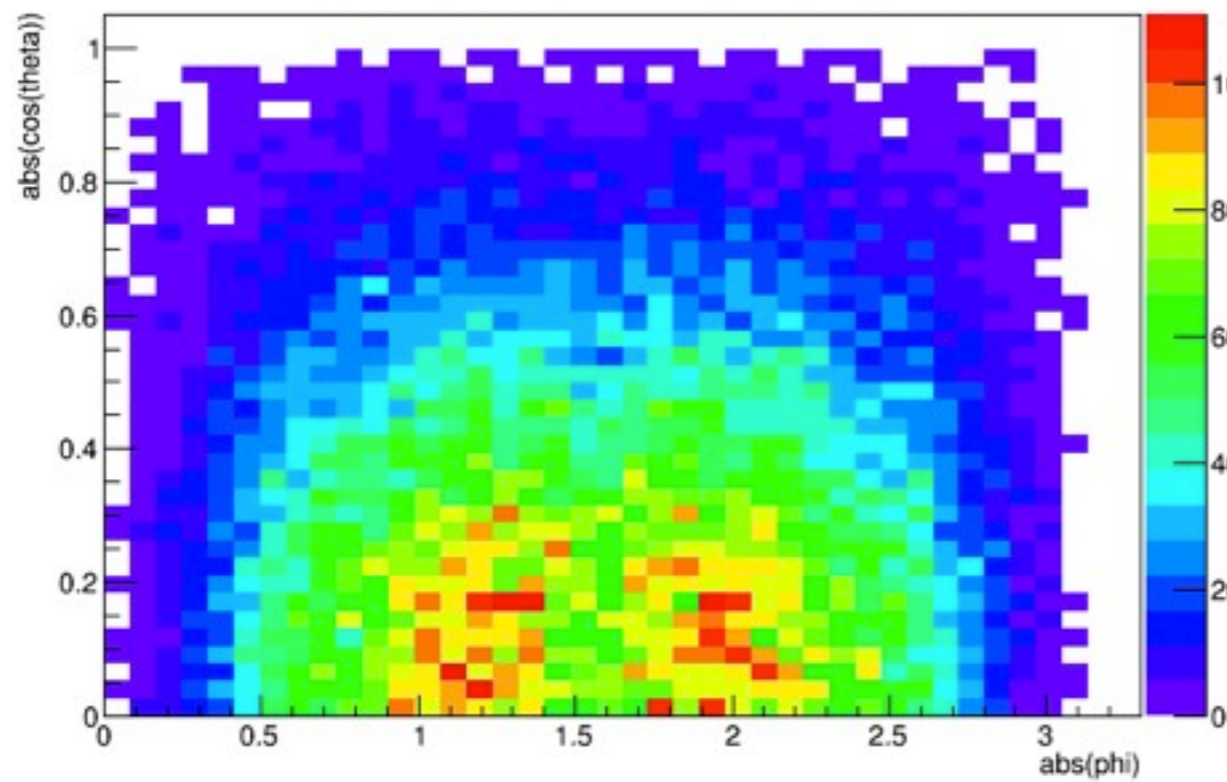
angles

MC

Reco

abs(cos(theta)):abs(phi)

abs(cos(theta)):abs(phi)

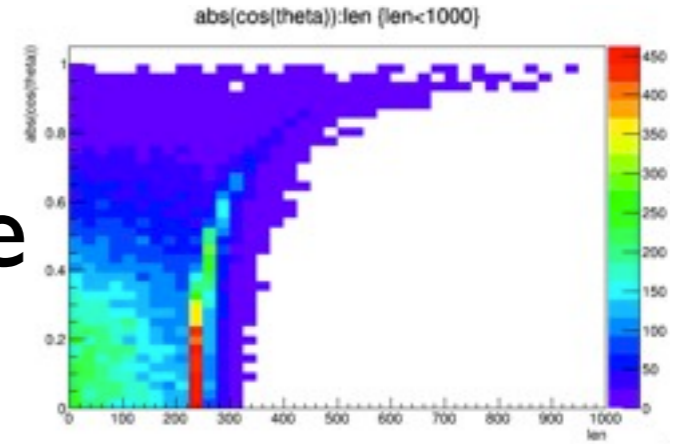


parallel to U,V wire

parallel to Y wire

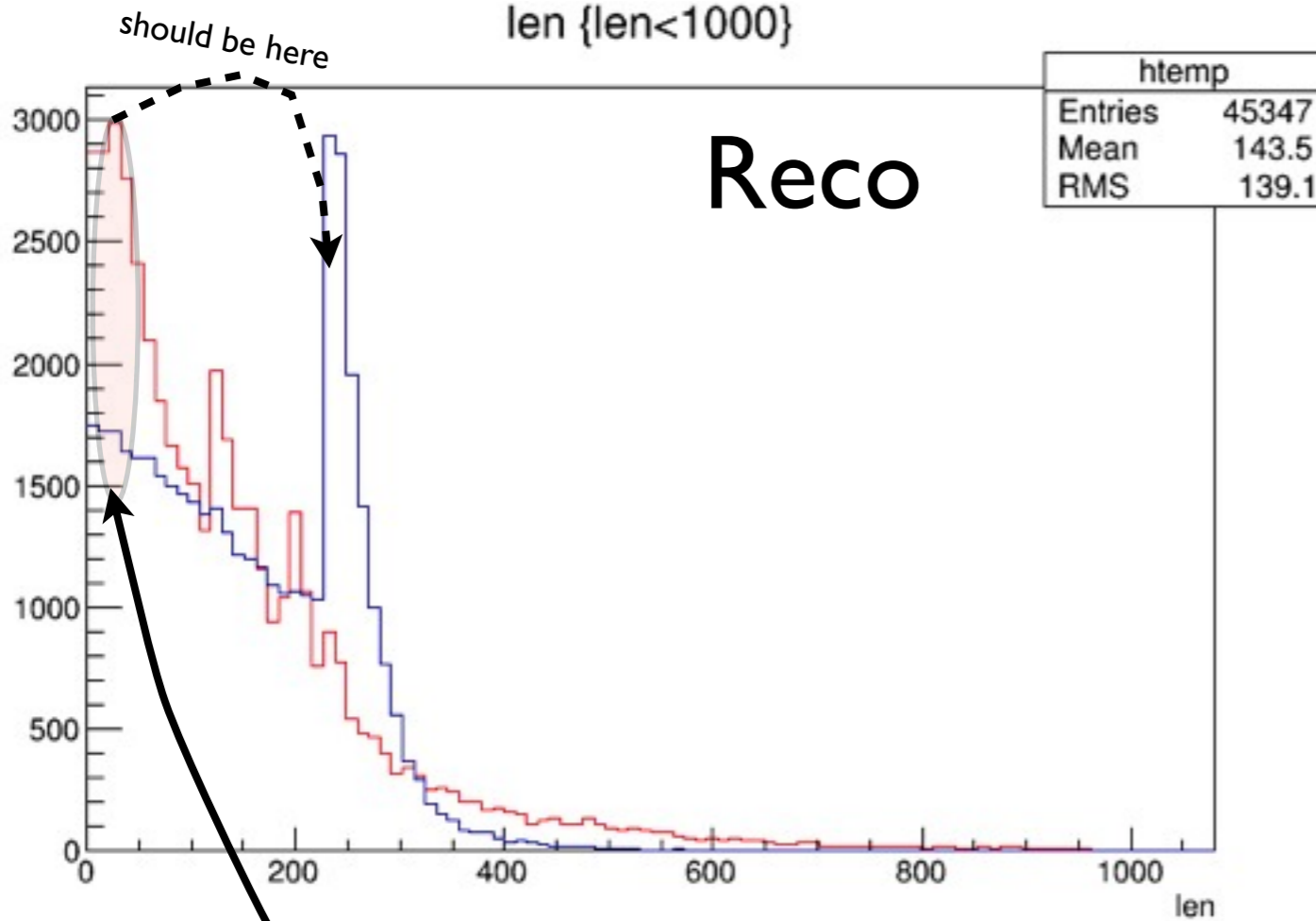
length

True

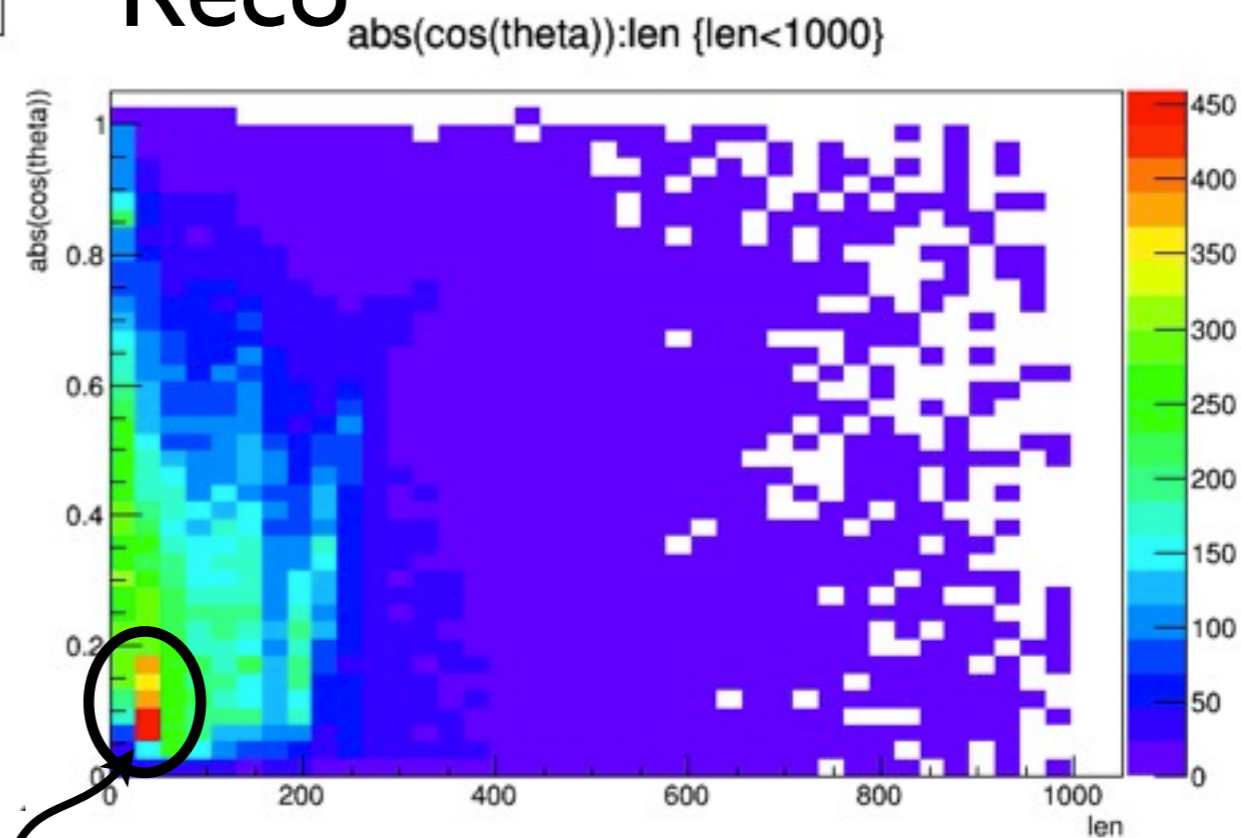


len {len<1000}

Reco



Reco

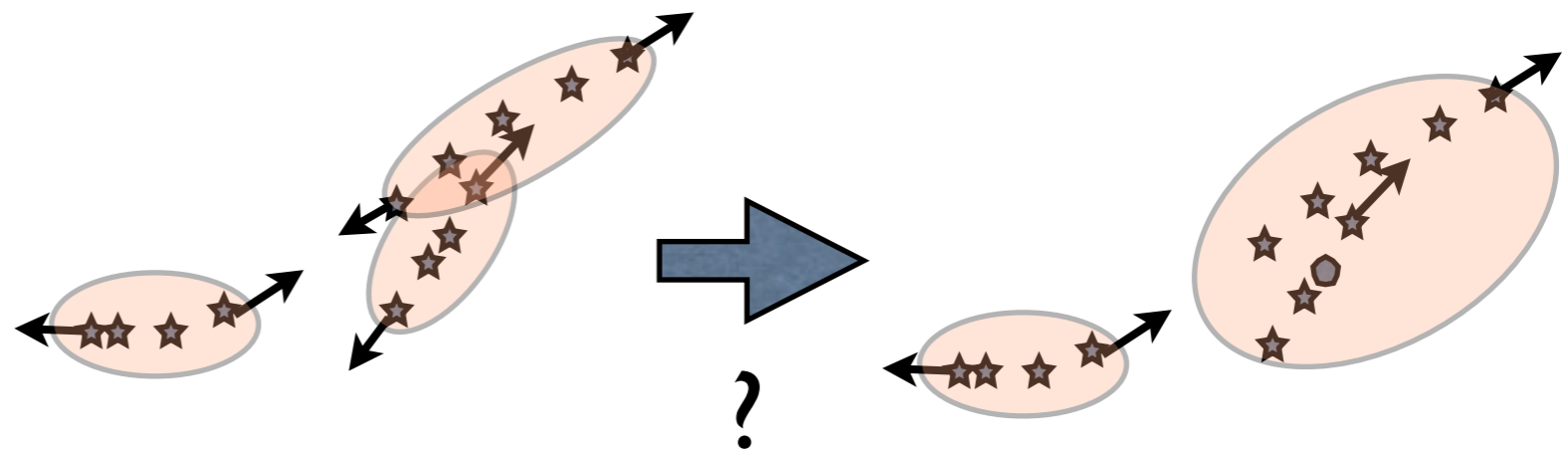


So, although the stitching helps fill in the costh plot, my track lengths are too short for the straight downward going cosmics.

length

- This quantity is the “path integral” from first to last point along track.
- If I’m stitching tracks but not sorting points within them, I can get some nutty results

Stitching requirement is just to match (to within dialed-in tolerances) endpoint separations and pointing directions.

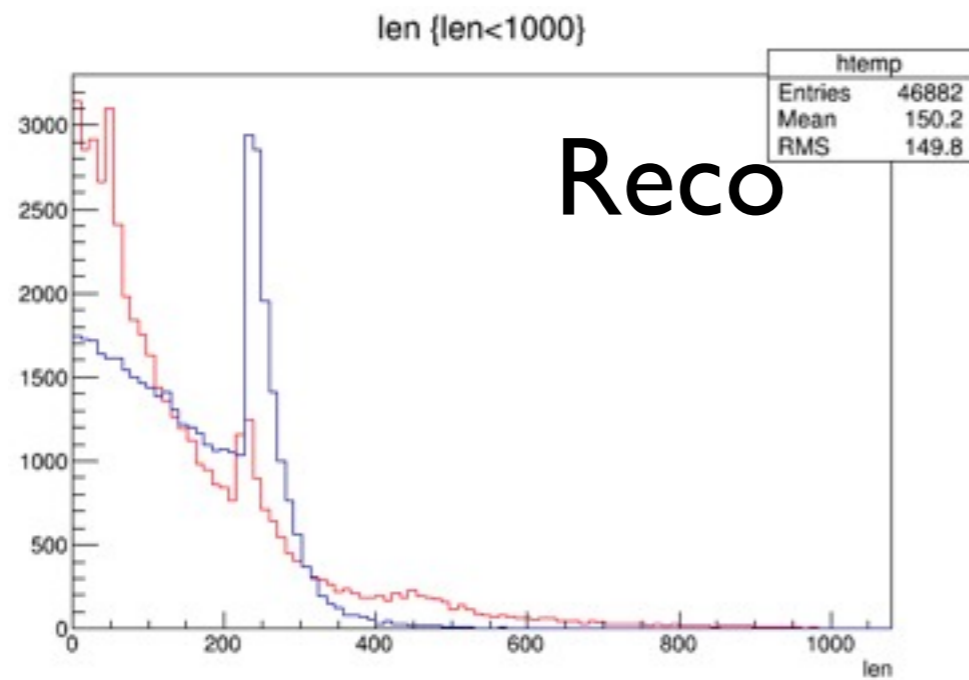


Next

- Ana module to Check the associated Hits of my stitched tracks against the CheatedHits.
- Producer module to exclude those Hits, and recluster. Then Re-recon SpacePoints, Re-recon Track3DKalSPS, ...

- Done!** ● currently Track3DKalSPS sorts spacepoints in z before running a track through 'em. For cosmics I oughta be sorting in y.

sppts sorted in y



Reco

Better

