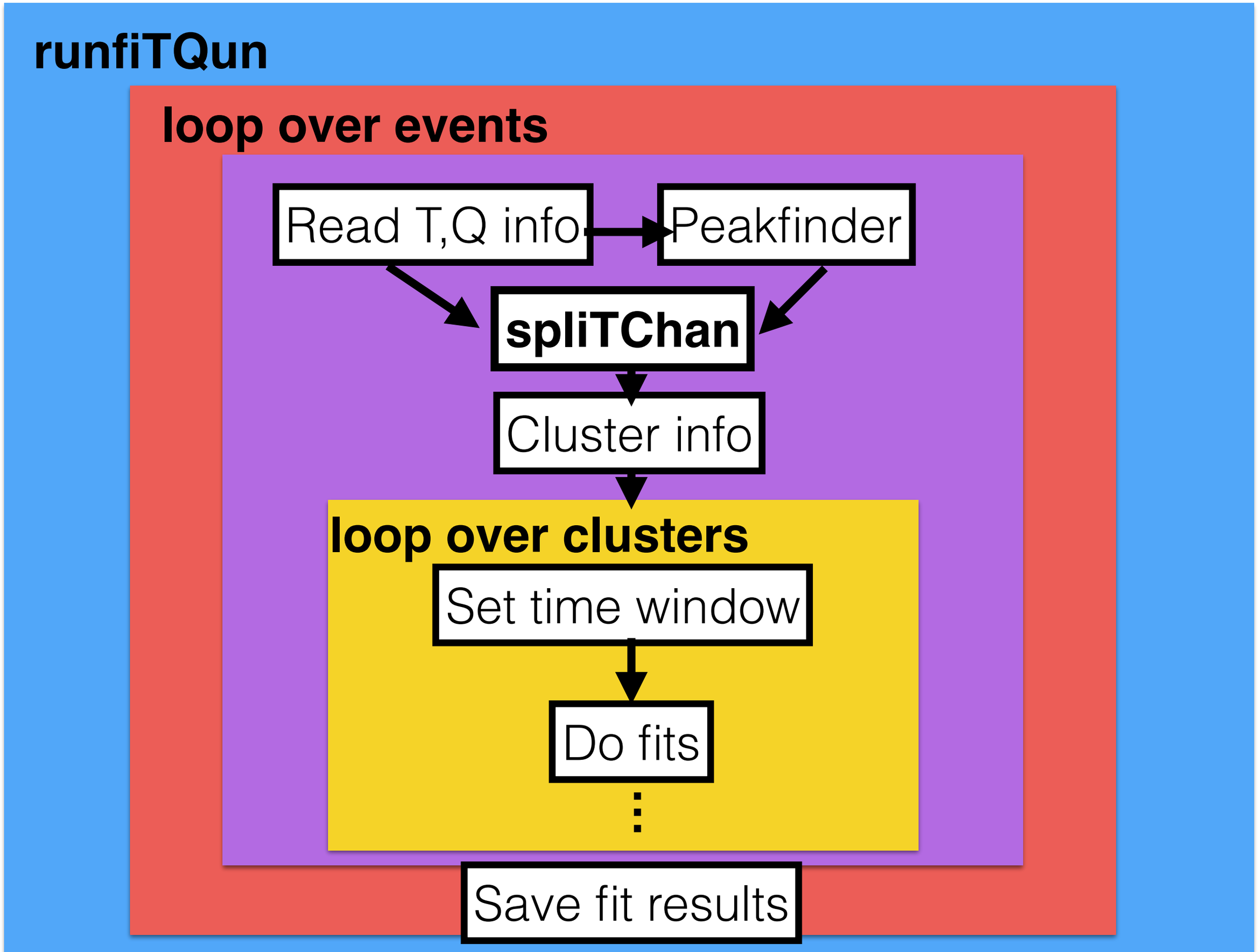
Introduction to spliTChan and Multiple-Segment Muon Fit

Andrew Missert
University of Colorado, Boulder
fiTQun Workshop
Dec. 15, 2014

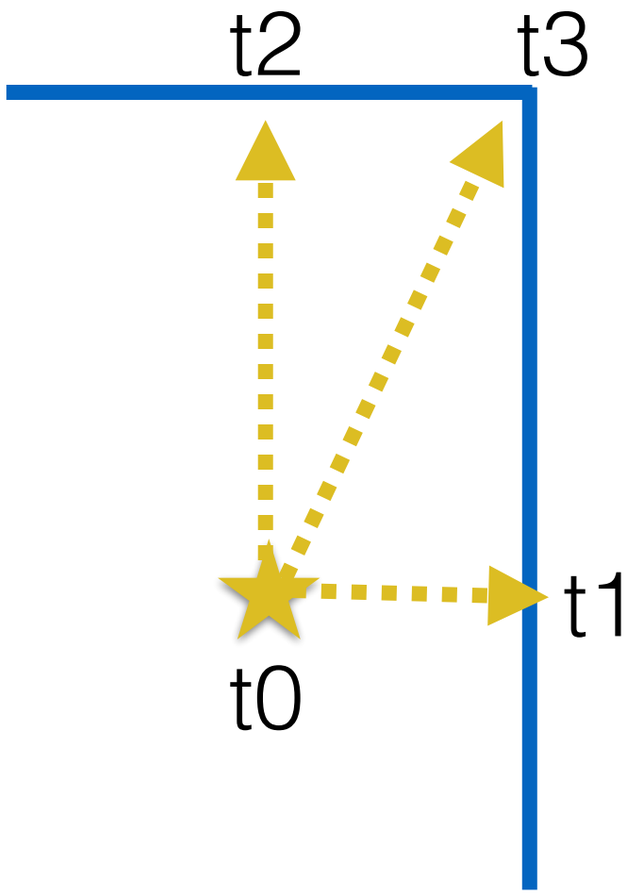
SpliTChan

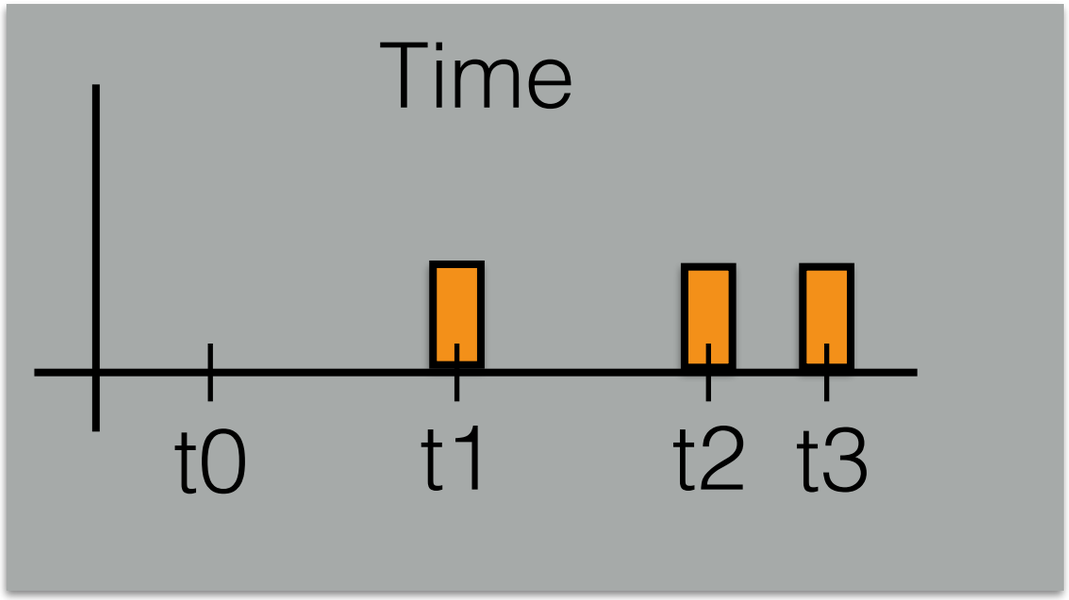
- fiTQun does not use all of the charge (Q) and timing (T) information at once
- Instead, the fits use only the T, Q information within a particular time window
  - this time window and the associated hits are referred to as a “cluster”.
- A particular event may contain one or more clusters (e.g. muon and Michel electron)
- Before any fits are run, we need a program to parse the PMT hit information and find the subverts
- spliTChan is the program performs this task

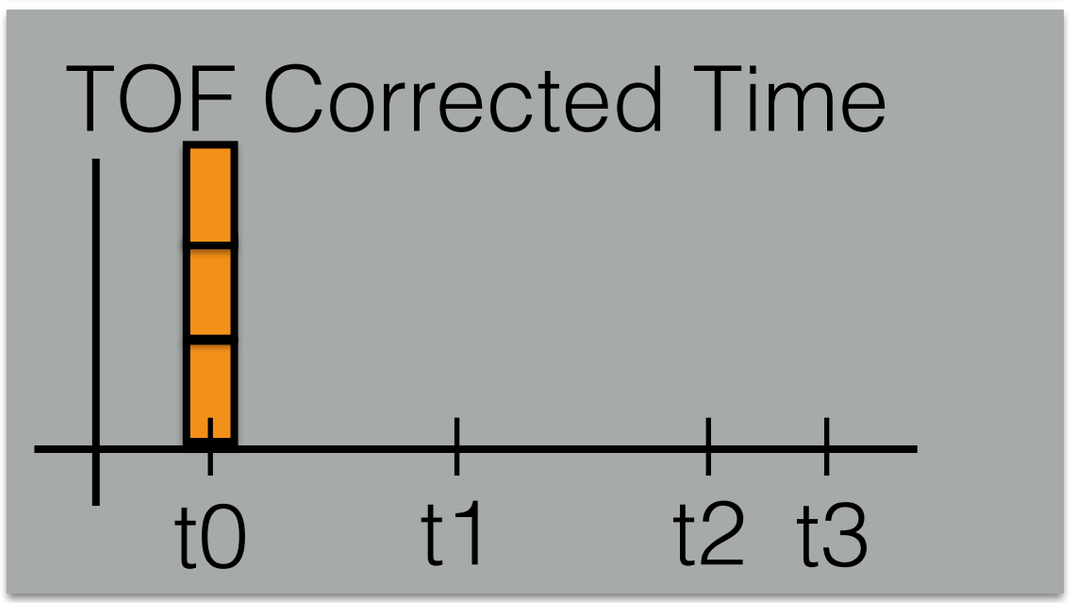

runfiTQun
loop over events
Read T,Q info
Peakfinder
spliTChan
Cluster info
loop over clusters
Set time window
Do fits
Save fit results

SpliTChan Algorithm

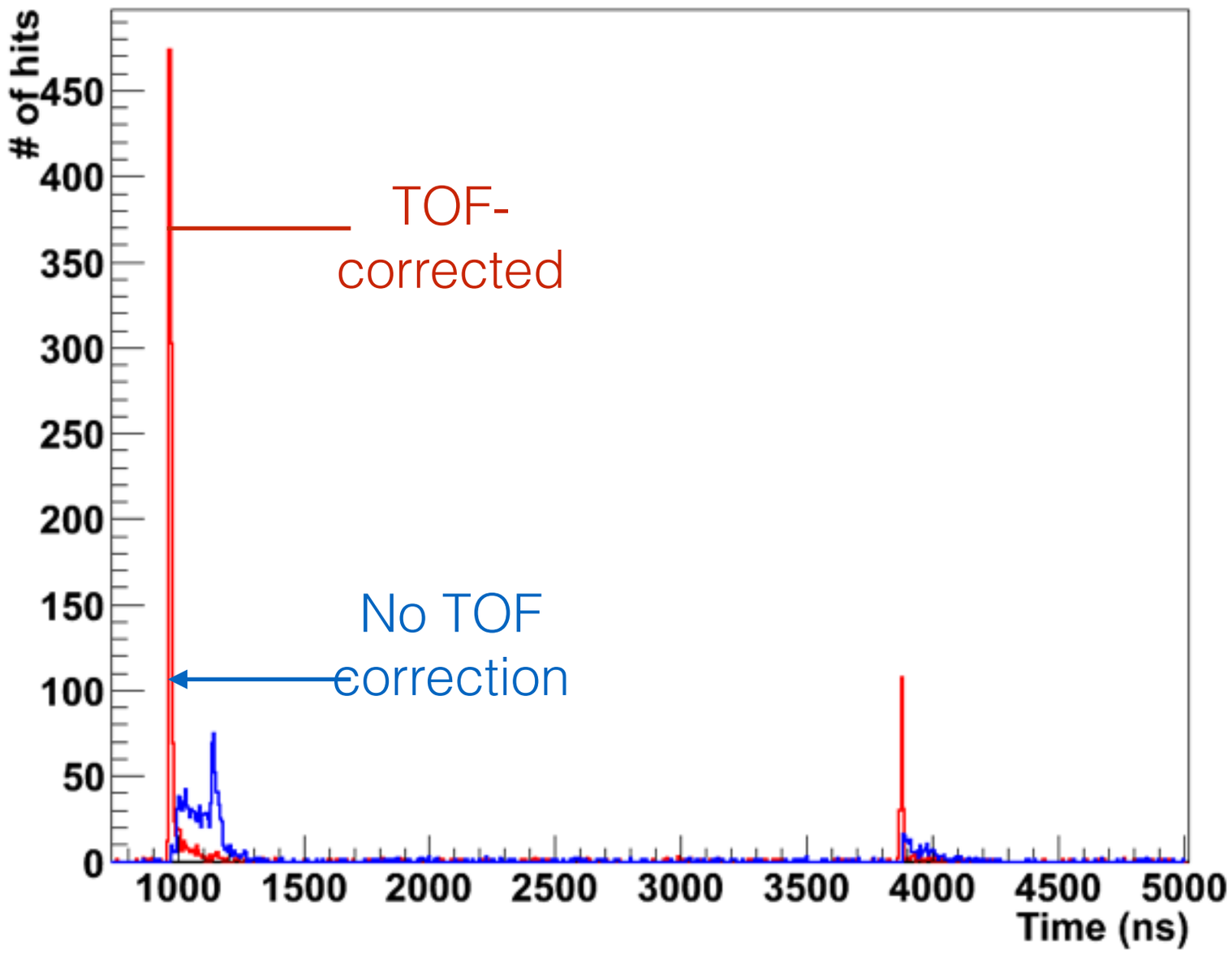
- spliTChan finds clusters use a fixed time window in “TOF-corrected time”





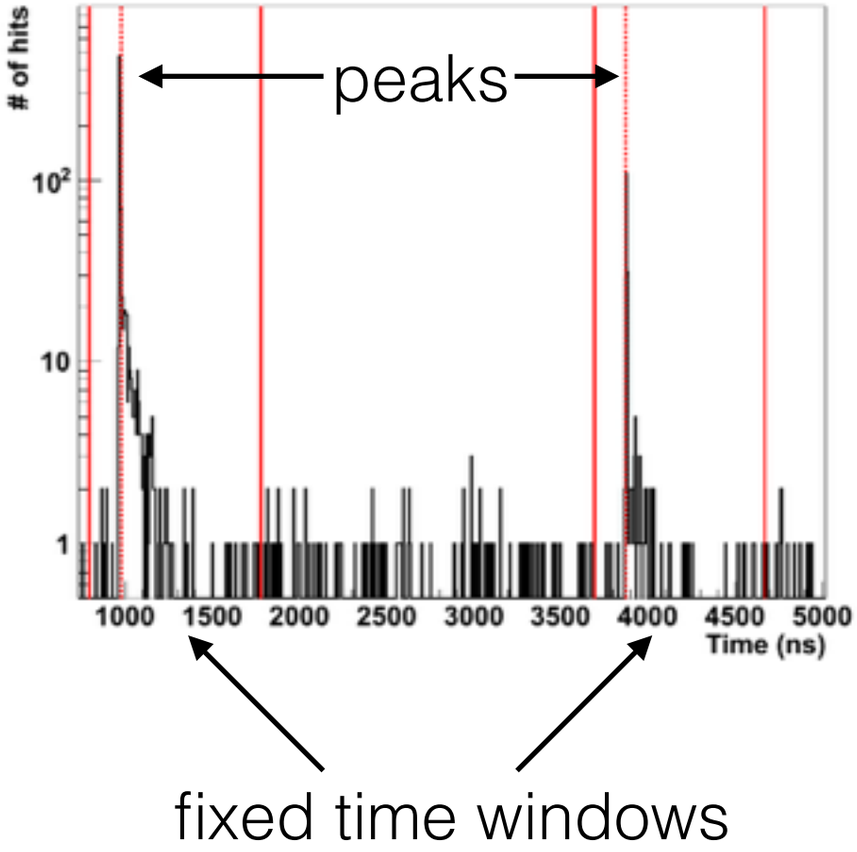


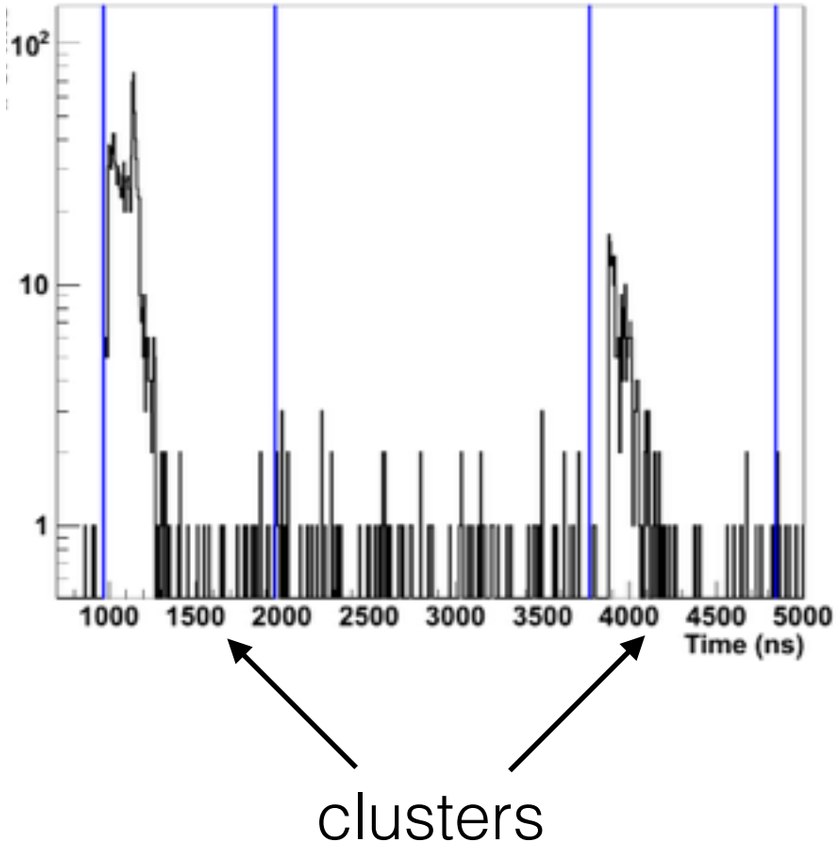
SpliTChan Algorithm



SpliTChan Algorithm

- New algorithm fixes a time window of (t_{peak}-180ns, t_{peak}+800ns) in TOF-corrected space
- hit times within this window and transformed back to actual time
- the min (max) of these actual times determines the start (end) of the cluster





spliTChan Parameters

- Some spliTChan parameters can be set in **spliTCHan.parameters.dat** file

- In general, these should need to be changed

- **spliTChan.UseTOFMethod** (default = 1) Toggles between cluster finding using the TOF method and the old sliding time window method.

- **spliTChan.FixedTBeforePk** (default = 180 ns)

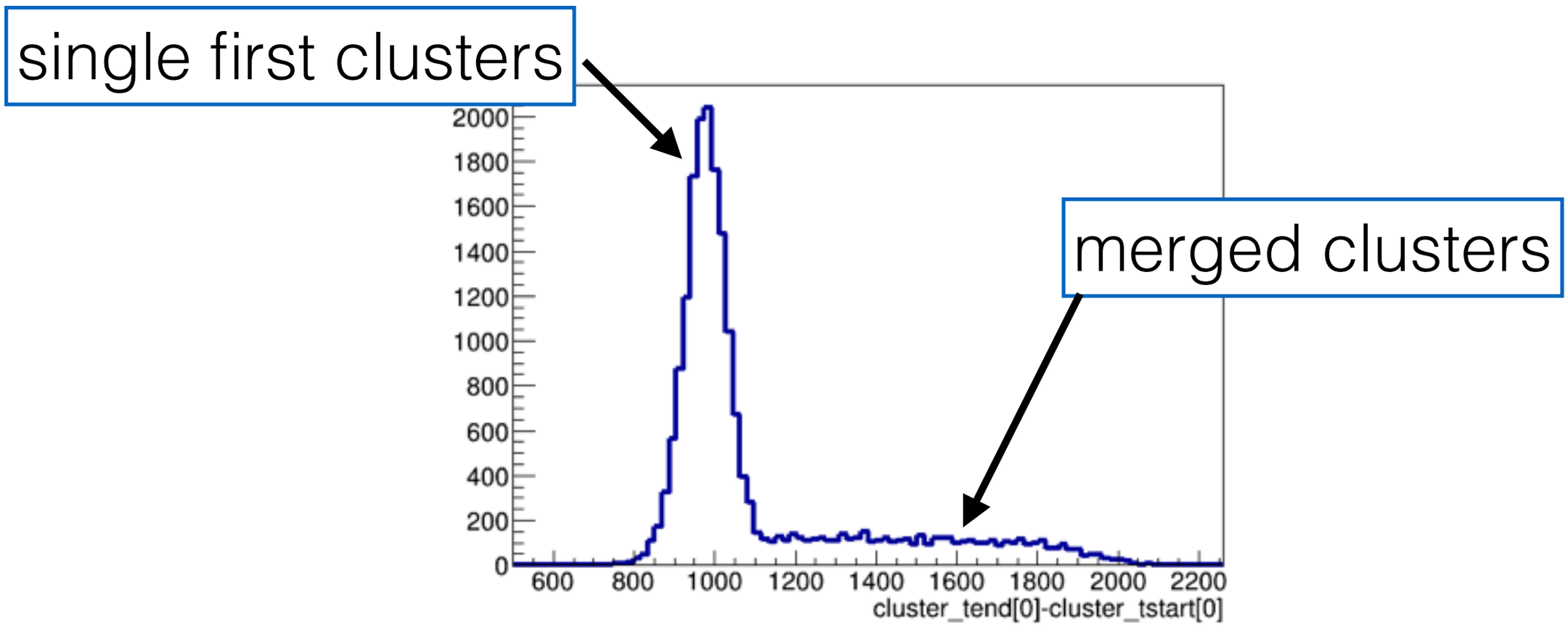
- **spliTChan.FixedTAfterPk** (default = 800 ns)

spliTChan outputs

- **cluster_ncand** - total # of clusters in event. (used to be # of ‘candidate’ clusters, but now all candidate clusters are good clusters)
- **cluster_tstart[i]** - time at beginning of ith cluster
- **cluster_tend[i]** - time at end of ith cluster
- **cluster_nhits[i]** - total # of PMT hits in ith cluster
- **cluster_totq[i]** - total integrated charge in ith cluster
- **cluster_goodflag[i]** - obsolete when using TOF method (should always be 1)
- **cluster_npeaks[i][0]** - total # of Peakfinder peaks in ith cluster

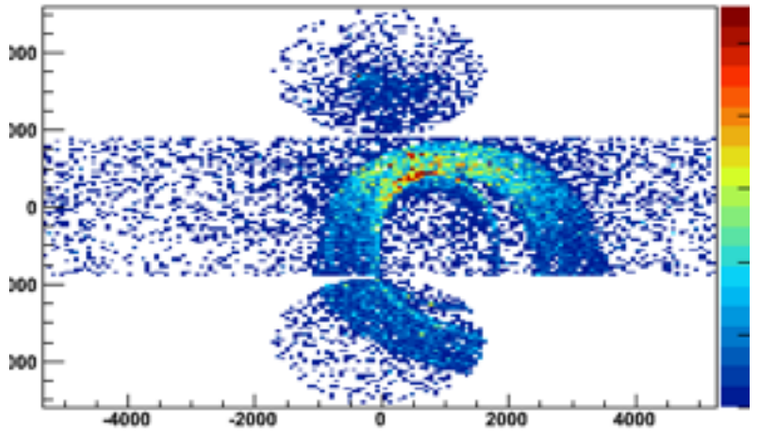
spliTChan outputs

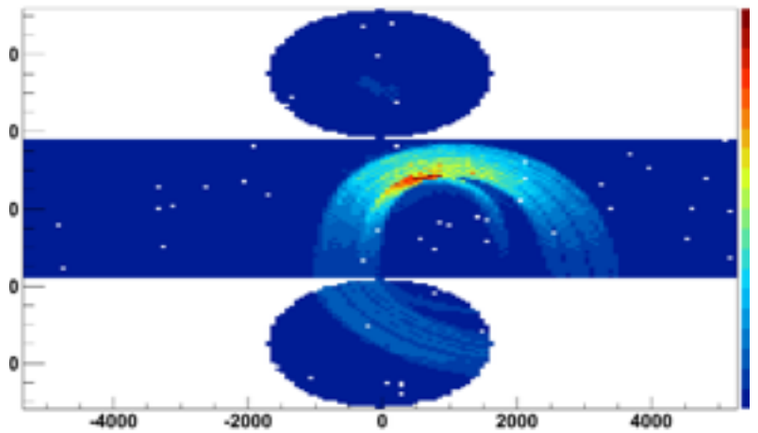
- Draw("cluster_tend[0]-cluster_tstart[0]")
  - Clusters are merged if they overlap

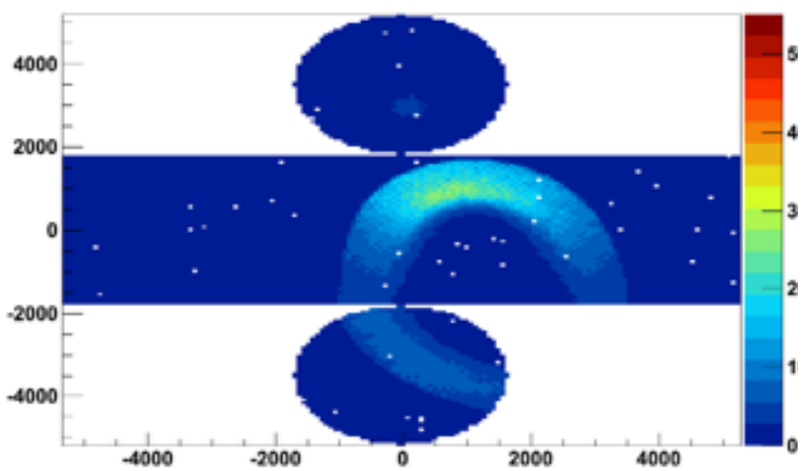


Multiple Segment Muon Fit

- Multiple segment muon fit addresses scattering of muon along its track
- Muon scattering results in charge distribution with distinct rings from different parts of the track
  - This causes the fit to mis-reconstruct the kinematics of the event when using straight tracks
  - Also, some of this structure could be mistaken as a separate ring by the MR fitter
- Solution: fit muon track in segments



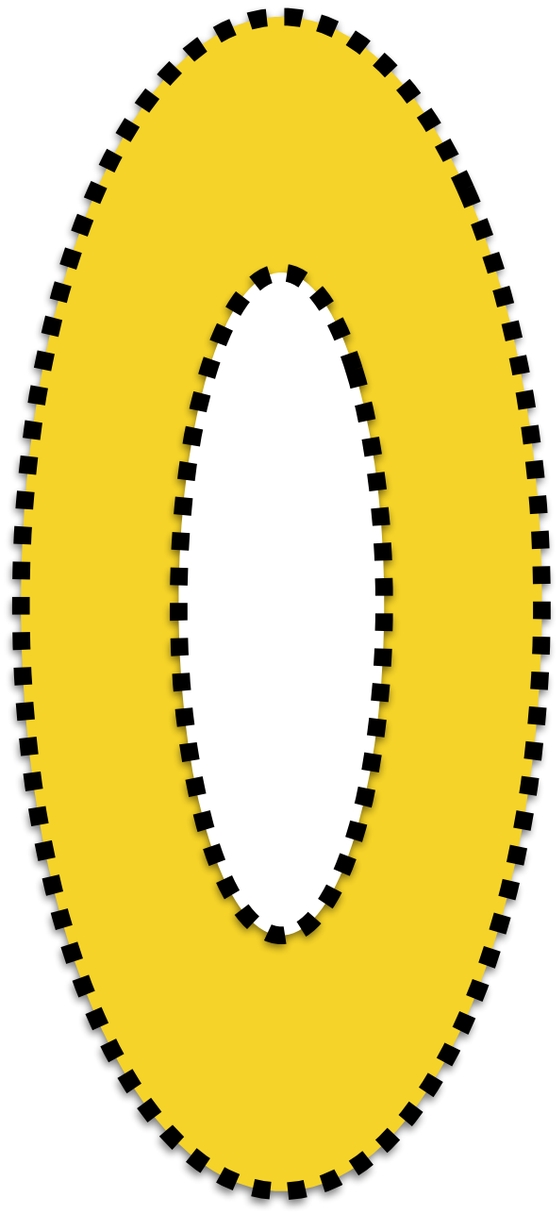




Track Segments in fiTQun

track parameters:

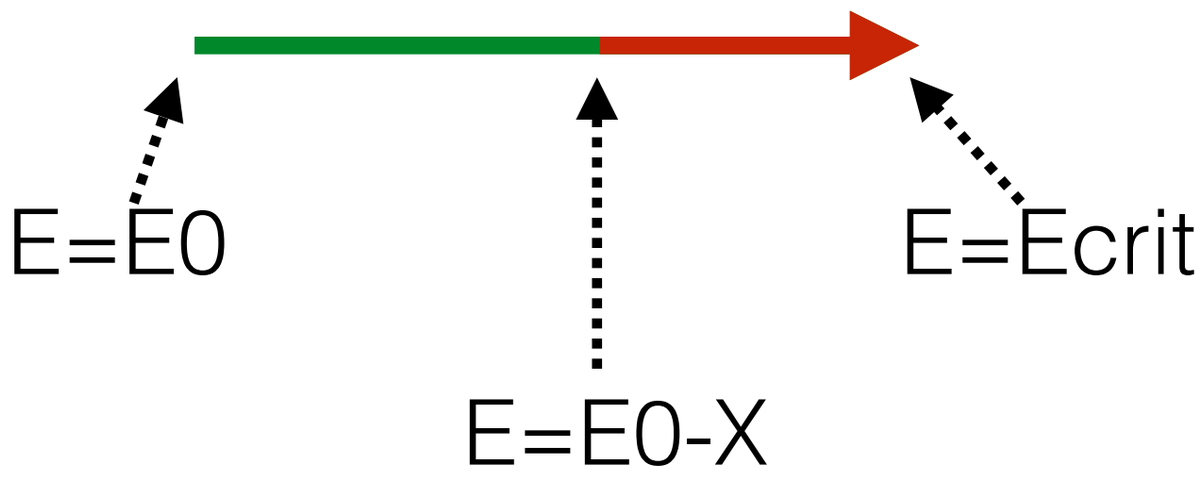
{p,x,y,z,t,θ,Φ,E_{loss}=0}

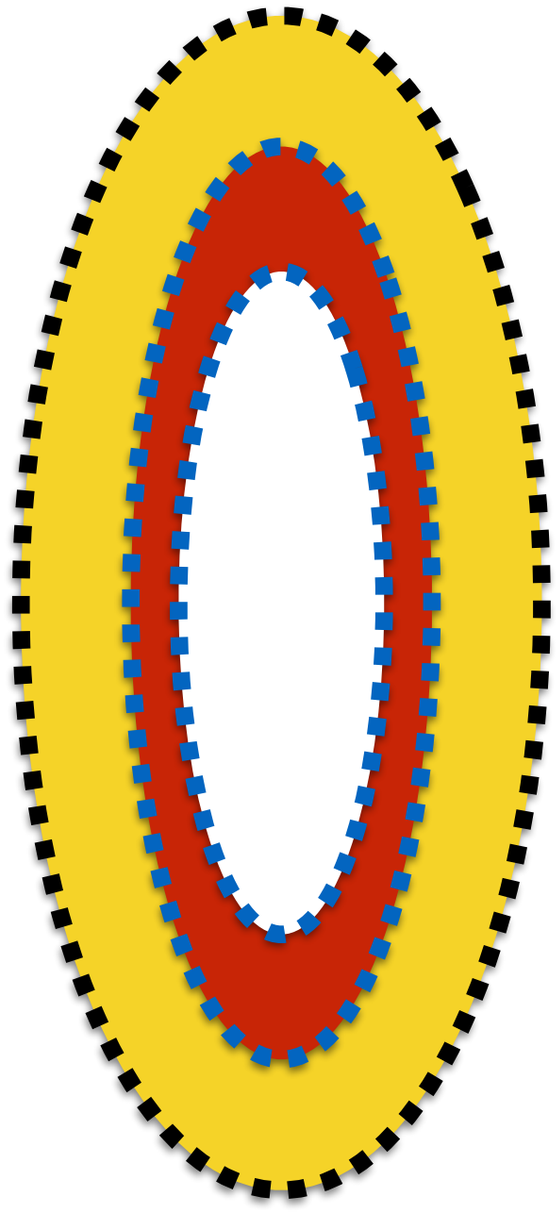


Track Segments in fiTQun

track parameters:

{p,x,y,z,t,θ,Φ,E_{loss}=X}

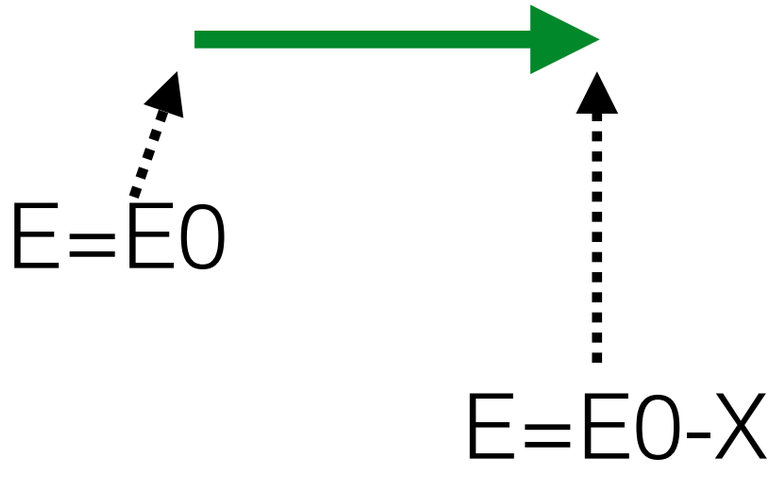


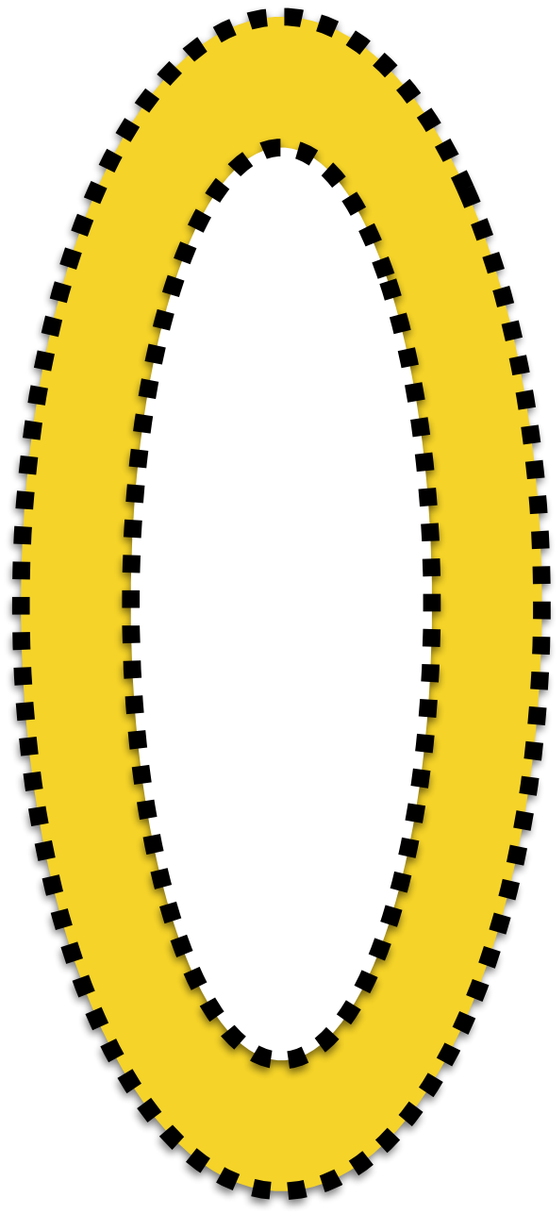


Track Segments in fiTQun

track parameters:

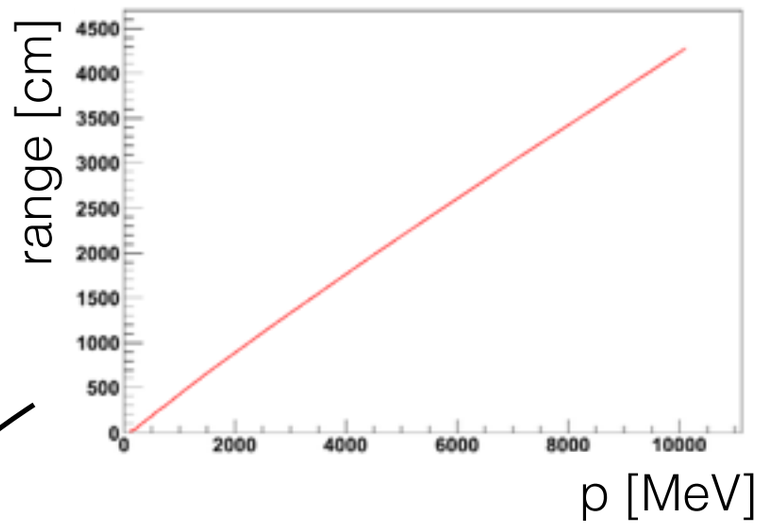
{p,x,y,z,t,θ,Φ,E_{loss}=X}

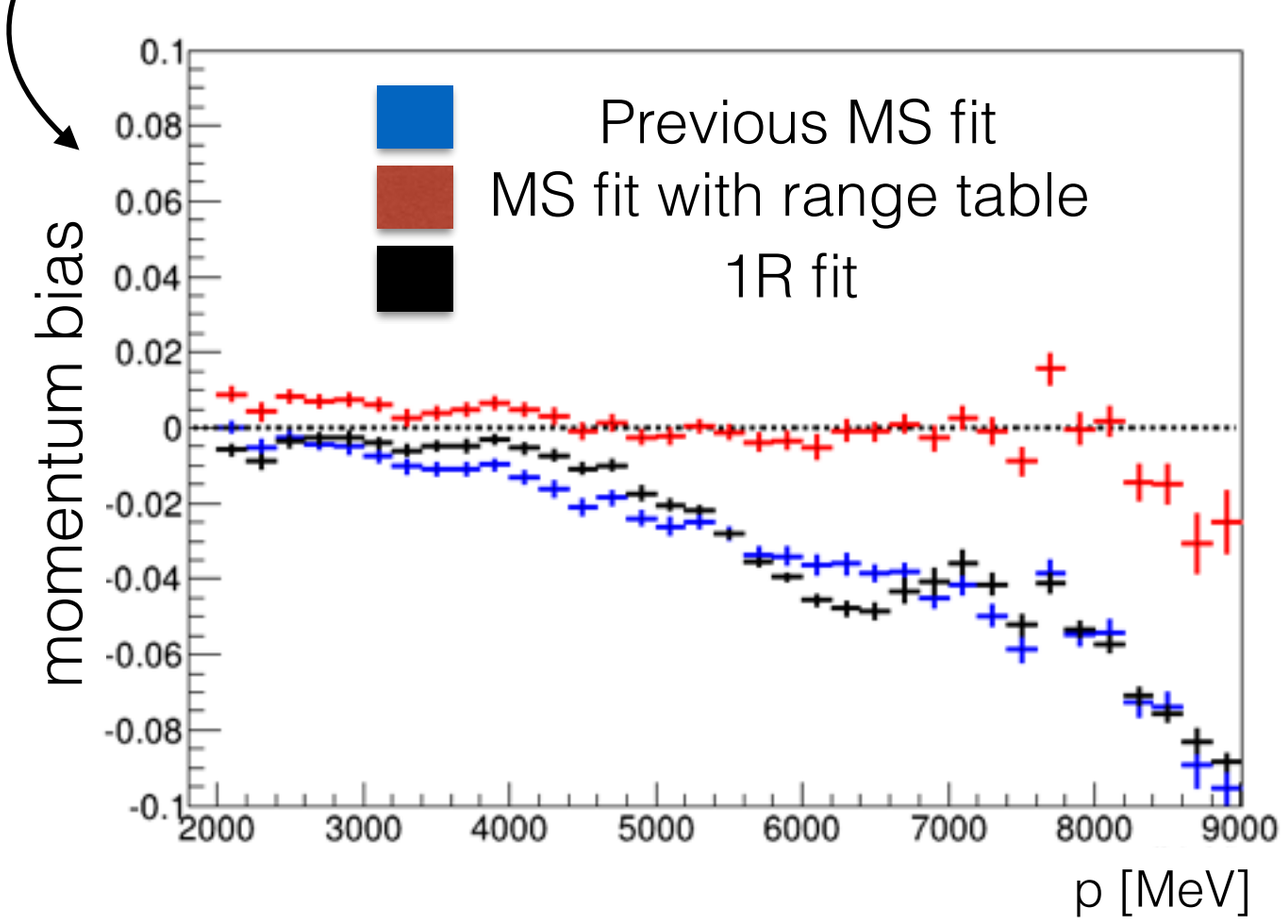




Multi-segment Muon fit

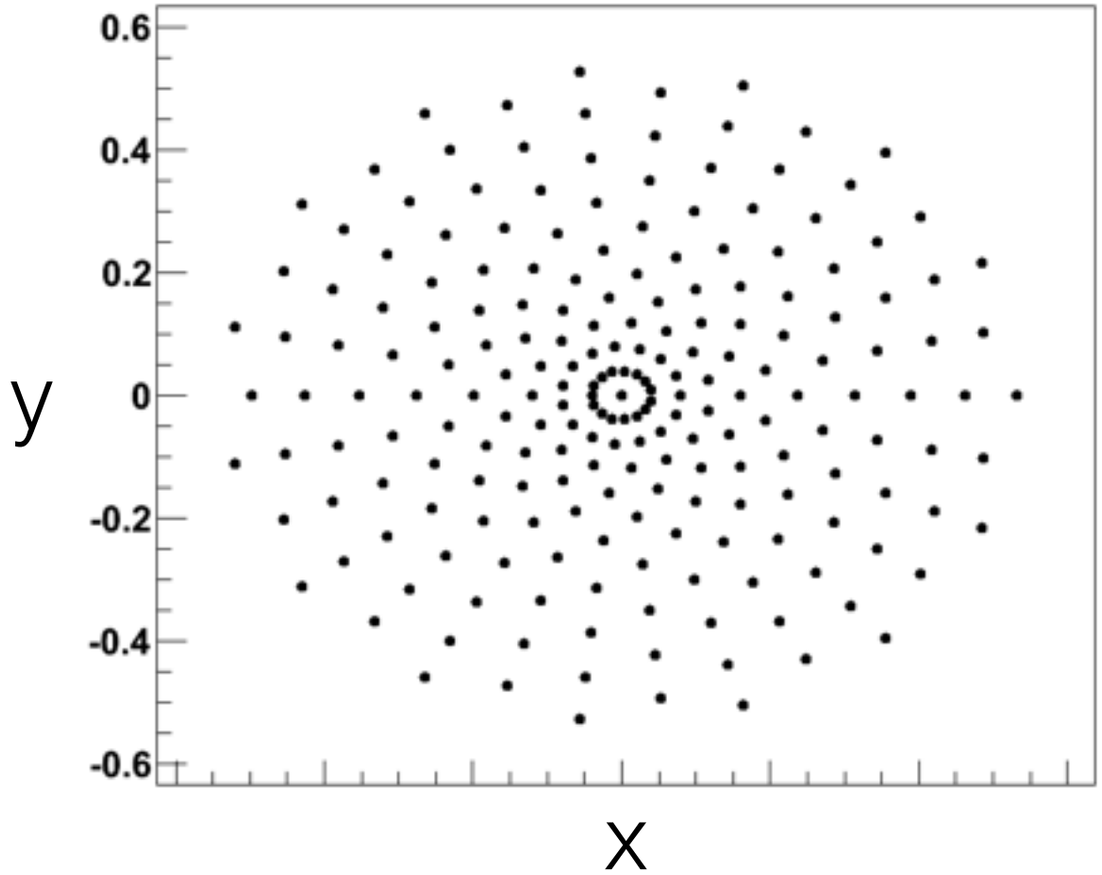
- fiTQun generally uses linear approximation for dE/dx
- This approximation breaks down for the downstream segments which have lower momentum
- For multi-seg fit we use muon range table filled from PDG for segment length, time calculation
- can be found in **gpdgRange.root**
  - Similar improvements if we use this for pions?





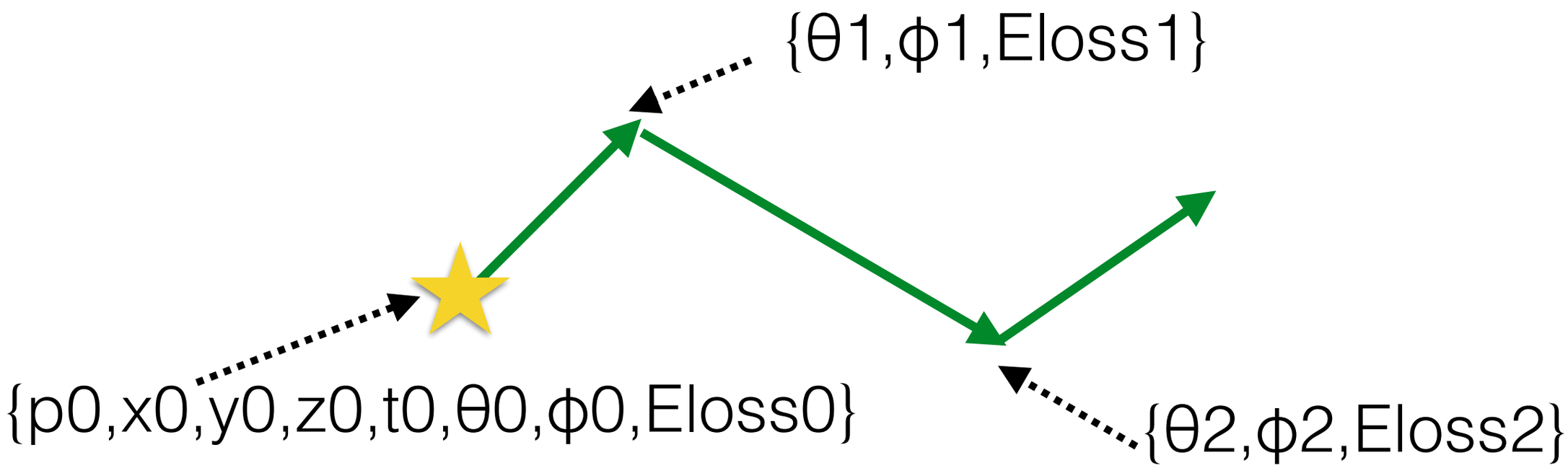
Multi-segment Muon fit

- Initial vertex and direction is seeded from the 1R muon fit
- Course grid search to seed each segment direction
    - Special algorithm to do this (see **SegDirGridSrch** method in fiTQun.cc)



Multi-segment Muon fit

- Initial vertex and direction is seeded from the 1R muon fit
- Course grid search to seed each segment direction
  - Special algorithm to do this (see SegDirGridSrch)
- Simultaneously fit all segments (max 3)
  - can also fit sequentially (max 9)



Multi-segment Muon Fit Parameters

- **fiTQun.DoMSFit** (default = 1) Toggles On/Off the multiple segment muon fits
- **fiTQun.nSegMax** (default = 3) Number of segments to be fitted (max 3 for simultaneous fit, 9 for sequential fit)
- **fiTQun.MSFitMethod** (default = 0) Determines the way the parameters are fitted. Default (0) fits all parameters simultaneously, Set to 1 to fit each segment sequentially, downstream to upstream. Sequential fit can handle more segments
- **fiTQun.MSElossMin** (default = 168 MeV) Parameter that sets the minimum segments lengths. (Fit will not converge if Eloss tend toward zero.)
- **fiTQun.MSThetaRes** (default = 0.05 rad) Determines the number of points checked when seeding segment directions. Smaller values will result in better seeding but require more time to compute. Probably should not be changed
- **fiTQun.MSScattSig** (default = 4) Determines the range of directions searched when seeding the segment directions. Probably should not be changed.

Multi-segment Muon Fit Output

- output variable root name is **fqms***
- **fqmsnfit** - total number of multi segment fits run (up to 2)
- **fqmsnll[i]** - log likelihood of ith multi seg fit
- reconstructed quantities are indexed by both the fit # and the segment #, i.e. **fqmsmom[i][j]** is momentum of the jth segment of the ith fit.

Using Multi Seg Fit

- to compare with the 1R fits, use the 0th segment, i.e. **fqmsmom[0][0]**, **fqmsnll[0][0]**…

- simple analysis of MS fit output:

  - **git init** (in new directory)

  - **git remote add origin https://bitbucket.org/amissert/mstools.git**

  - **git pull origin master**

  - edit **runmstools.c** to point to directory of muon MC files

  - I have also included a tarball of cosmic MC to show improved performance above 2GeV.