'global' alignment using collision data

Wouter Hulsbergen
27-01-2010

- improvements of 'TAlignment' framework
  - TWIKI
  - use of survey data
  - combining different data sets
- field-off alignment
- field-on data
- miscellaneous

twiki with manual for TAlignment

https://twiki.cern.ch/twiki/bin/view/LHCb/TAlignmentManual

- subjects
  - how to choose detector elements that you want to align for
  - how to choose lagrange constraints
  - how to choose survey constraints
  - how to combine alignment info from different data sets
  - some examples
- read also the section '**other initialization issues**'
  - for alignment to work properly it is best if geometry is initialized with a geometry corresponding to the data set that you use
    - e.g. velo half closed and not in current parking position
  - make sure to set EventClockSvc().InitialTime

```
# specify the time of the 'first' event in ns
EventClockSvc().InitialTime = 1260350949785664000
```

using the survey

- what survey information?
  - OT C-frames (dx,dy,dz)
  - OT modules ('nominal +/- 50 micron')
  - IT ladders, layers, (2008 survey), boxes (2009 survey)
  - TT ladders
  - Velo modules, wafers
- alignment procedure can now use these data
  - as a set of **xml files** (survey **uncertainties** hardcoded in alignment code)
  - as single python strings specifying a parameter and its error (see TWIKI)
- in the head of TAlignment

using the survey (II)

- how does it work?
  - position of hit determined by different alignment constants at different level in detector hierarchy, e.g.

$$\Delta x_hit = \Delta x_Cframe + \Delta x_module$$

survey: precision of module in C-frame: 100 micron

survey: precision of C-frame in global frame: 0.5 mm

  - assume survey measures coordinates in LOCAL frame, so with respect to parent
  - assume survey measurements are uncorrelated in this frame
  - now align **both** the parent and its daughters, e.g. an OT Cframe and all its modules **simultaneously**
- alignment procedure finds the **combination** of module and C-frame parameters **that best fit the survey**

using the survey (III)

- why would you do this?
  - make optimally use of available information: useful addition to track information
  - stabilizes alignment
  - non-arbitrary way to fix all weak modes: weak modes are solved so that survey chisquare is minimized
- but what happens if survey is wrong?
  - no big deal. at some point you collect sufficient information in sample that survey measurement is irrelevant
- why would you not want to do this?
  - survey is never perfect: very hard to estimate uncertainties
    - don't take all correlations into account
  - result will depend on size of track sample

combining different data sets

- problem: how to combine information from field-on, field-off, Ted-data, cosmics ?
  - cannot process raw data with same set of options
  - need a way to combine alignment info
- solution in TAlignment
  - use persisted alignment derivatives ('alignderivatives.dat' file)
  - future plan: save these to a file summary record in DST
- 2 ways
  - run a script with different 'alignderivatives.dat' files as input
  - use 'alignderivatives.dat' from data set A as input to alignment with B
    - this would be equivalent to using A's constants plus covariance matrix as input
    - can still run iterations on B
- for instructions, see the TWIKI
- one caveat
  - for the moment this only works if you align same detector elements
    → persisted file only works if exactly same alignables
  - this can be solved, but takes at least 2 full days … not high priority

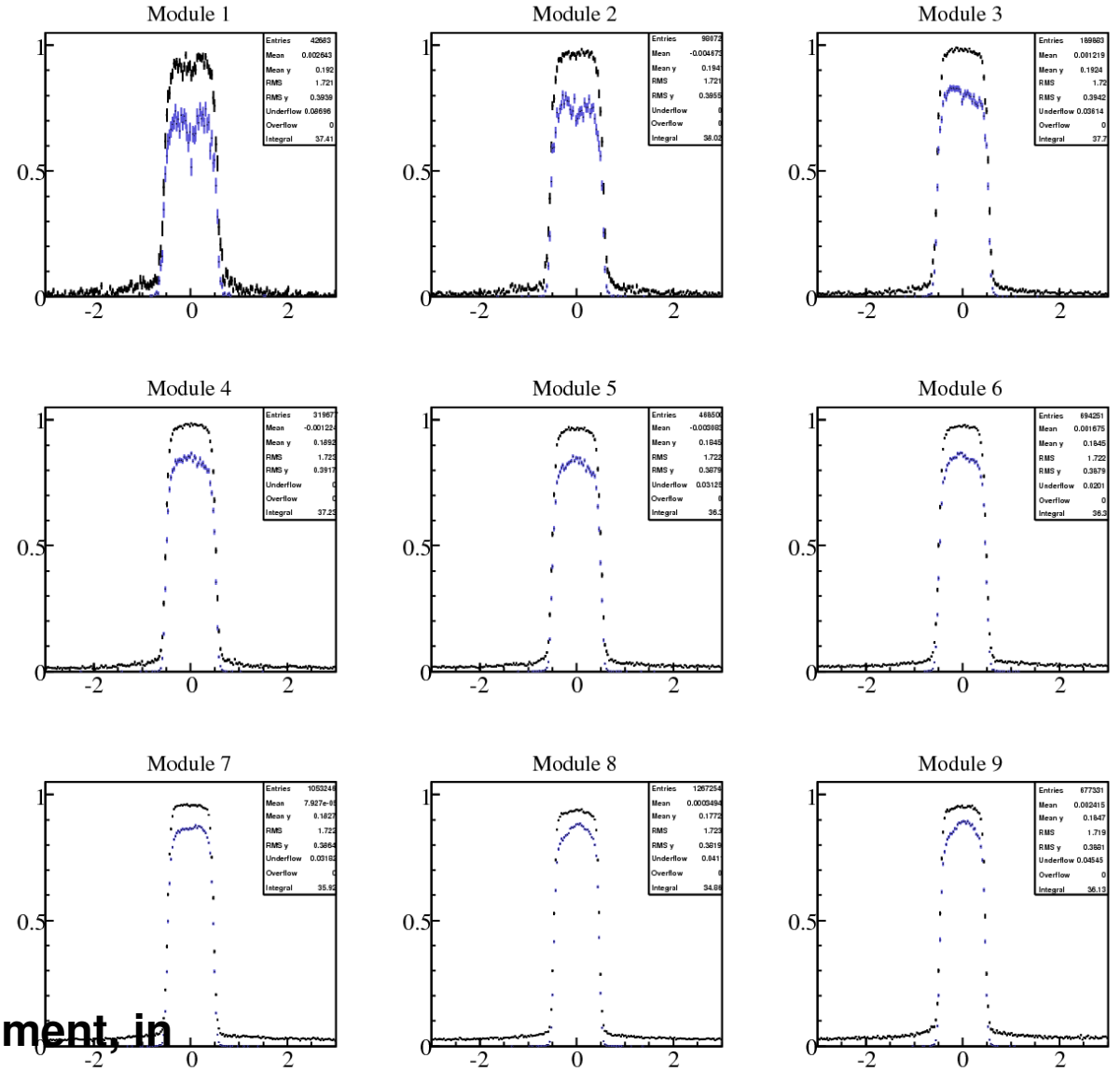
TrackHitAdder

- reconstructed tracks miss hits

OT hit efficiency profiles:

in black: all raw hits

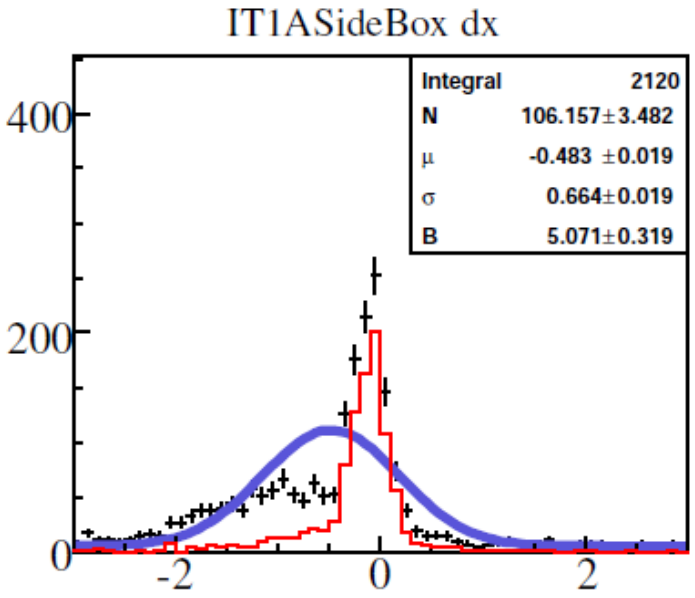
in blue: hits on tracks



claim: BIG problem for alignment, in particular if hits are in overlap regions

TrackHitAdder

- example: OT tracks extrapolated to IT hits



red: hits on tracks
black: all hits

- alignment convergence extremely poor because track finding removes hits
- could loose pattern reco hits, but we don't want to pick up garbage tracks
- solution: add missing hits back on existing, reconstructed tracks

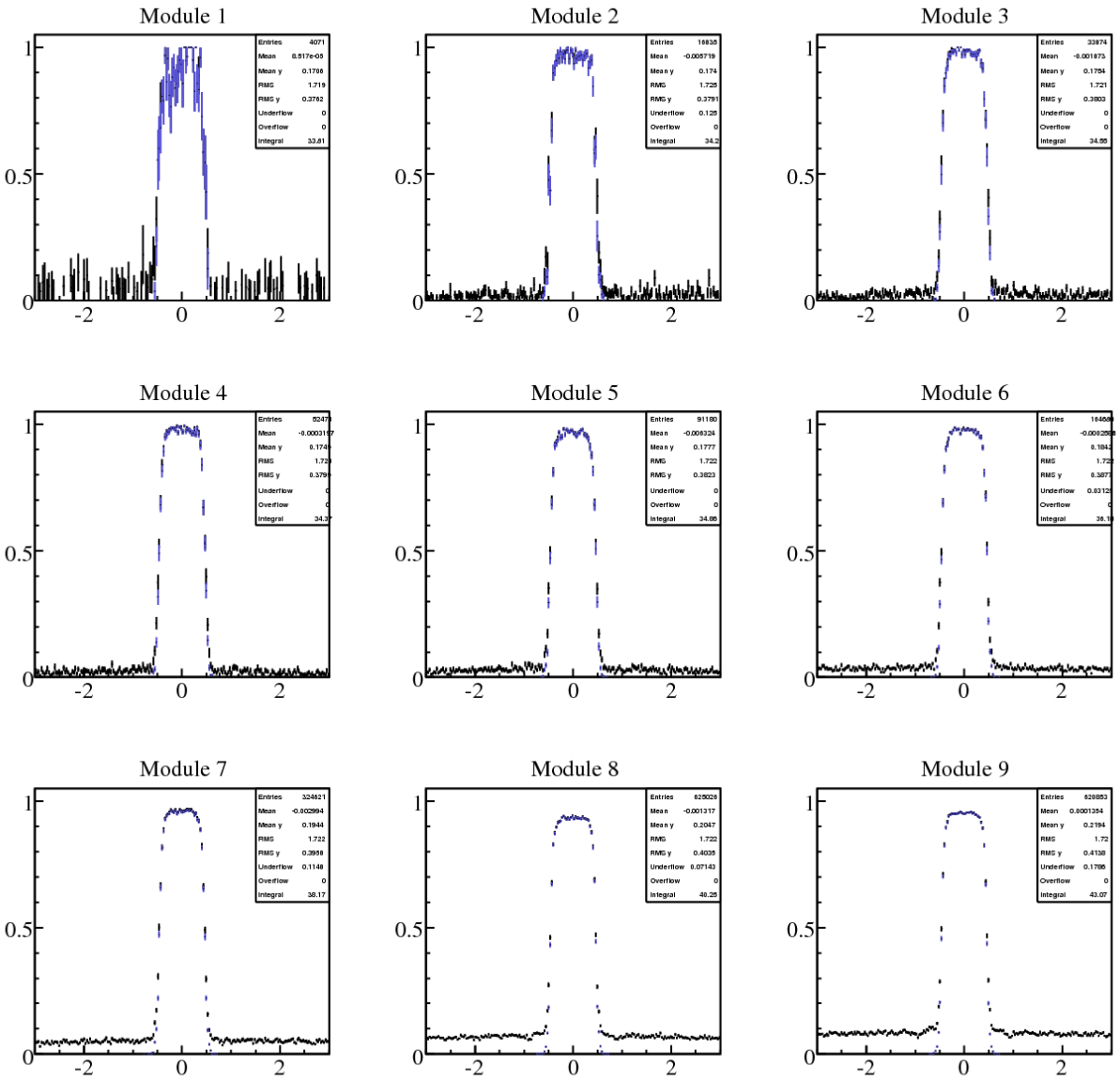
TrackHitAdder

- as you have seen Manuel committed yesterday **TrackHitCollector**
- I wasn't patient enough: have written **TrackHitAdder** algorithm in parallel
  - resides in head of TrackUtils package
  - adds TT, IT and OT hits within given fixed window (0.5, 1 and 2.5 mm)
    - does not add Velo hits, yet
  - only about 200 lines of code
  - about 1ms per event (+ time to refit the tracks)
    - relies on small modifications in TfKernel and TfTools to optimally search hits in window
  - refitting tracks is slow, but … with a few hours of work we can change fit such that we can insert hits
    - saves expensive reinitialization of material and field integrations
  - not extremely sophisticated, but just what I need for Alignment
- future
  - use Manuel's tool … ?

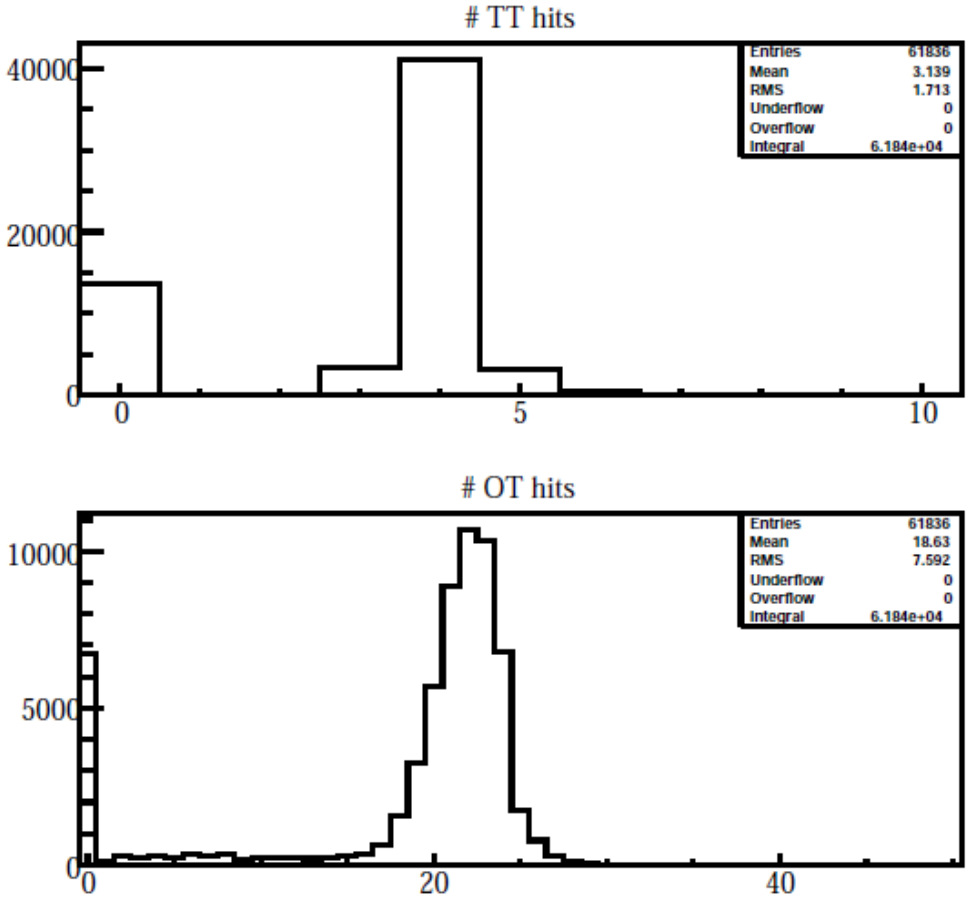
TrackHitAdder sanity check

doesn't miss a single hit

doesn't add anything outside driftcell



more advertisement



this is data

with HitAdder and after alignment, number of TT and OT hits per track is near perfect

(compare this to what is currently on the DST)

- MESSAGE: **don't study calibration, resolution, efficiency without considering the hits omitted by the pattern reco**
- we need a tool like this in the standard tracking sequence

how to deal with 'out-of-time' OT hits?

- one problem: some OT hits have poor drifttime measurement
  - to be understood by OT experts
- since I don't use outlier rejection in fit, loose many tracks in chisquare cut after adding more OT hits
- solution: **ignores driftttime if hit outside driftttime spectrum** (with 1 sigma tolerance)
  - hit without drifttime still useful for alignment
  - implemented in **TrajOTProjector**
  - note: not a cut on the residual. cut really on measured tdc value, after tof/propagation time correction.
  - seems to be quite stable. do we want this by default in track fit?
    - need to do some work to OTMeasurement before I can commit this

magnet off alignment

- stepwise approach
  - velo-ecal
  - velo - OT C-frames
  - velo - OT modules
  - velo - OT – IT boxes
  - velo – OT – IT boxes – TT layers
- main challenge: get the overlaps in IT correct
- on Friday 15th the first database was released to LHCBCOND. see also

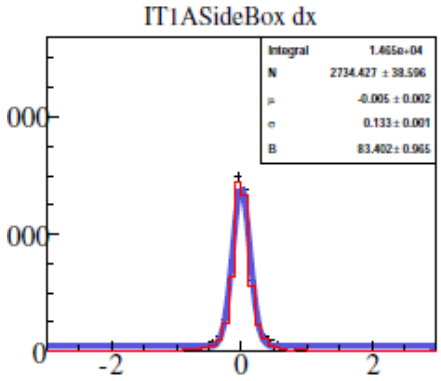
  ~wouter/public/AlignDB/VeloOTTxTyModulesTxITTxTyRzTTTxFieldOff_20100115.db
- some changes since then
  - updated TT internal alignment using field-on data

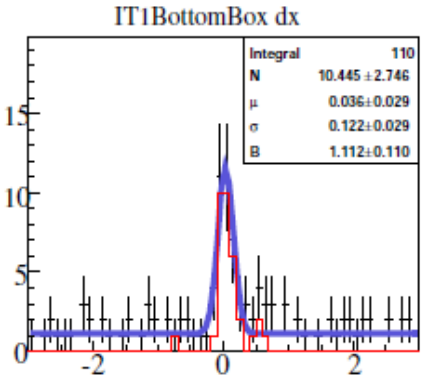
    VeloOTTxTyModulesTxITTxTyRzTTTxFieldOffTTLaddersFieldOn_20100122.db
  - updated OT survey to take into account bowing of the bridge, improved OT module alignment, improved IT overlap

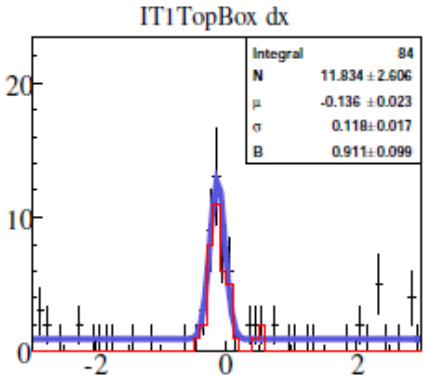
    VeloOTTxTyModulesTxITTxTyRzTTTxFieldOffTTLaddersFieldOn_20100127.db

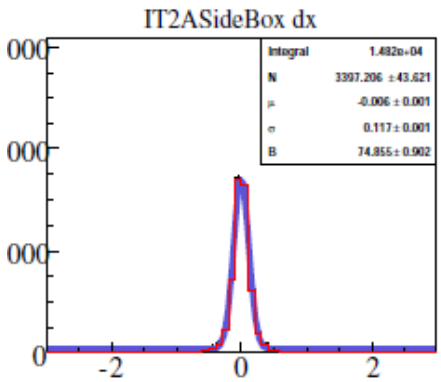
track type

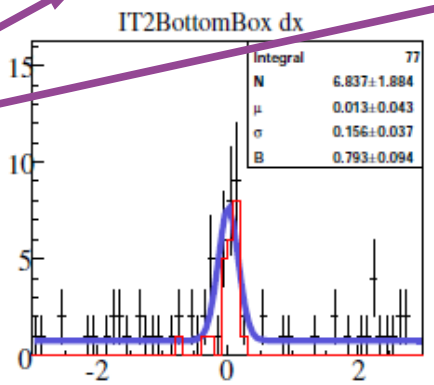
ITASideTrack

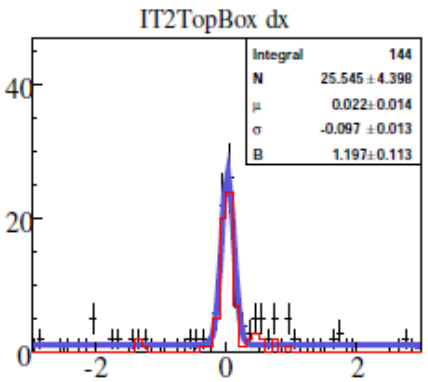








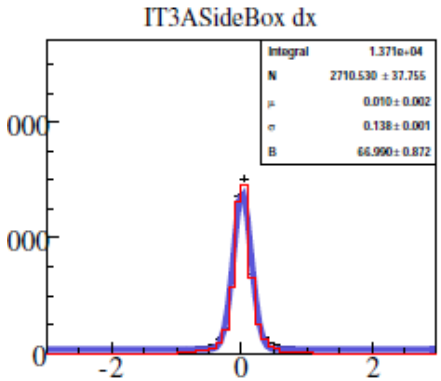


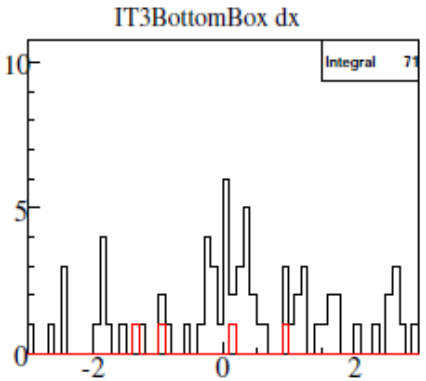


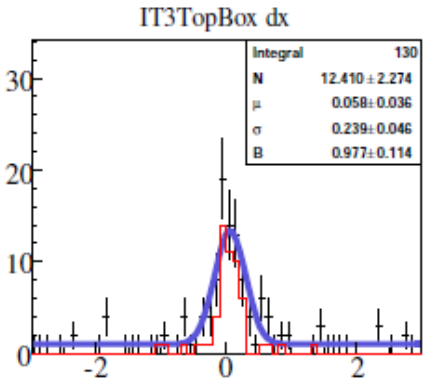
hit types in IT

all hits

hits on this track



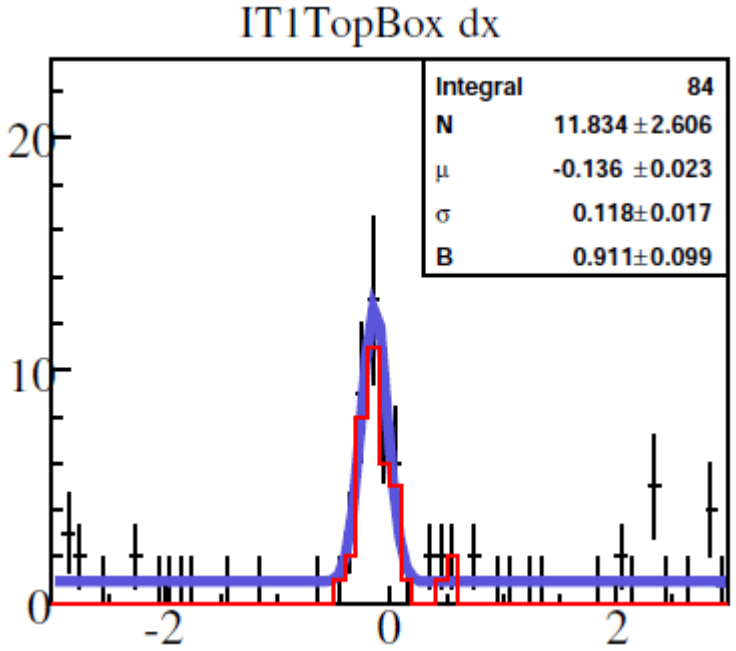




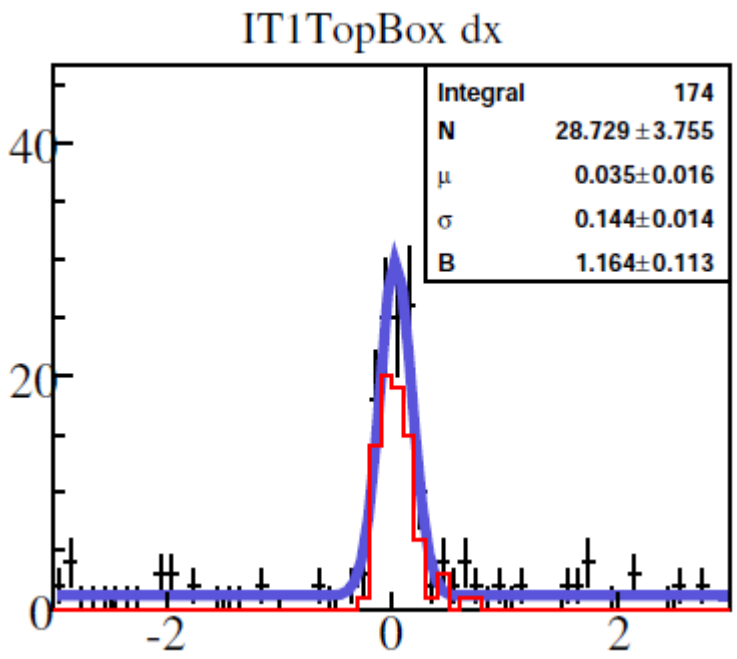
progress in overlaps

- reported last week: not all overlaps right yet
- e.g. residuals of hits in T1 TOP box with respect to A and C-side IT tracks:

A-side



C-side

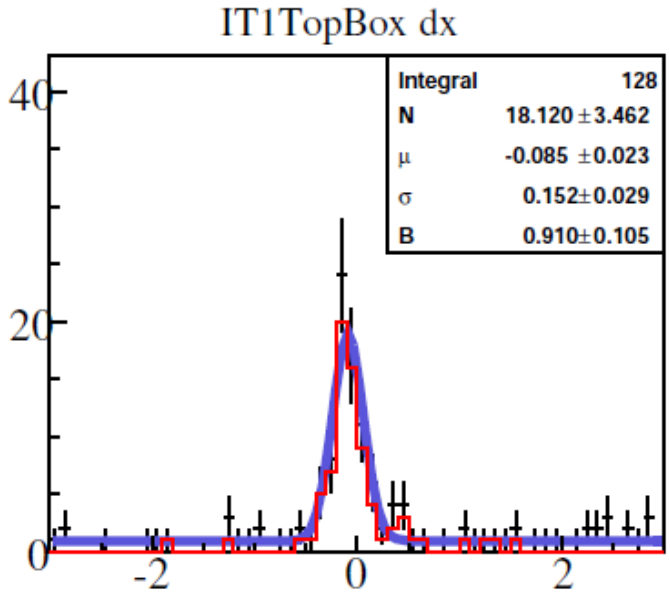


- apparently, A and C side are pulling in opposite directions
  - about 220 micron apart. why don't they just move together?
  - did we not yet align for all relevant degrees of freedom? (left/right velo)
  - or **are we selecting the wrong tracks?**

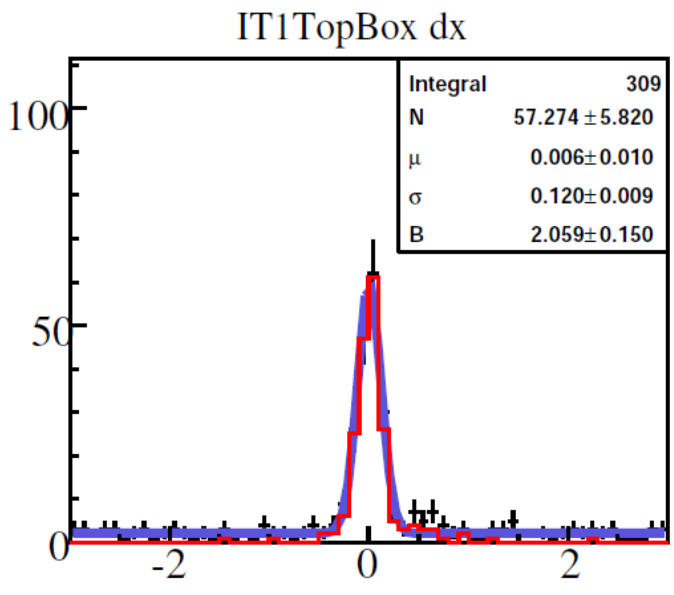
progress in overlaps

- reported last week: not all overlaps right yet
- e.g. residuals of hits in T1 TOP box with respect to A and C-side IT tracks:

A-side



C-side



- using the TrackHitAdder and better track selection bias is halved
  - about 90micron (was 220)
  - about 1.5x more hits in these plots

'reduced' understanding in velo ecal match

- have used different strategies for fixing the 'y' angle of the tracker
- approach 1 (december)
  - fix velo such that tracks point perfectly at calo
  - align the rest of the track wrt velo
- approach 2 (january)
  - use OT/IT survey constraints to fix velo rotation
- preferred approach 2 because IT/OT stay closer to survey
- unfortunately not clear what the value is of the 'global' position survey
- VELO-CALO match now looks worse than ever
  - may need to go back to approach 1.

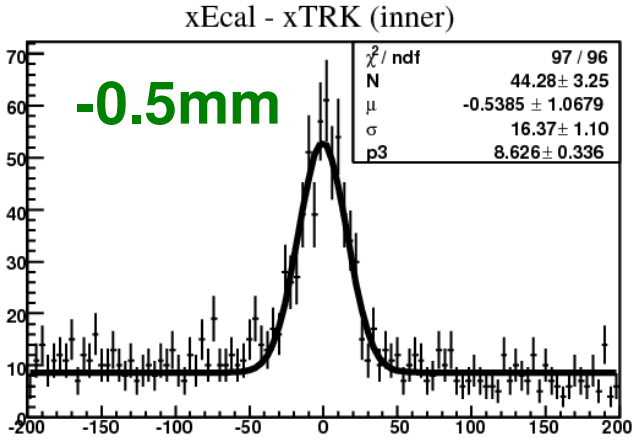
ecal-velo match

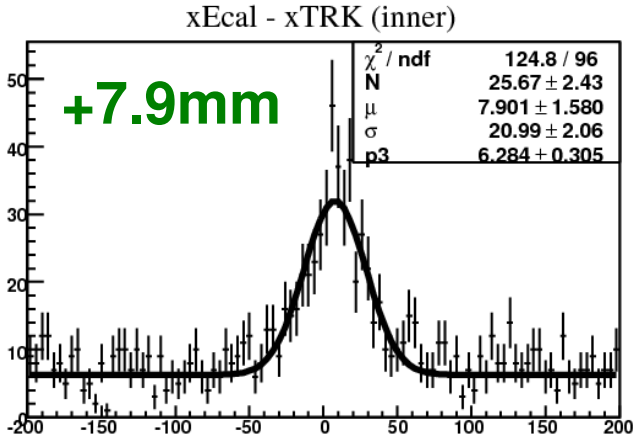
- extrapolate all VELO tracks to ECAL. assume 'MIP' for cluster position

A-SIDE ('LEFT')

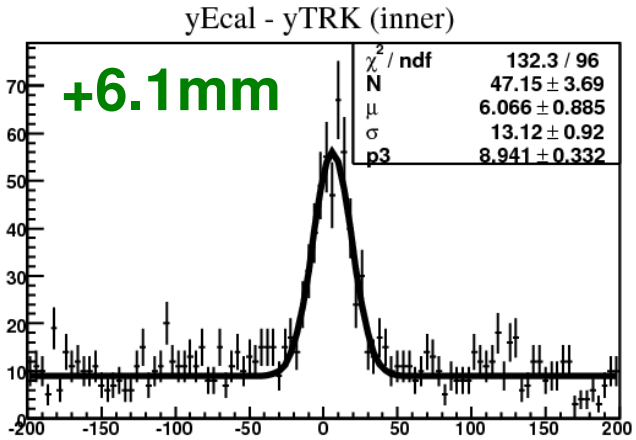
C-SIDE ('RIGHT')

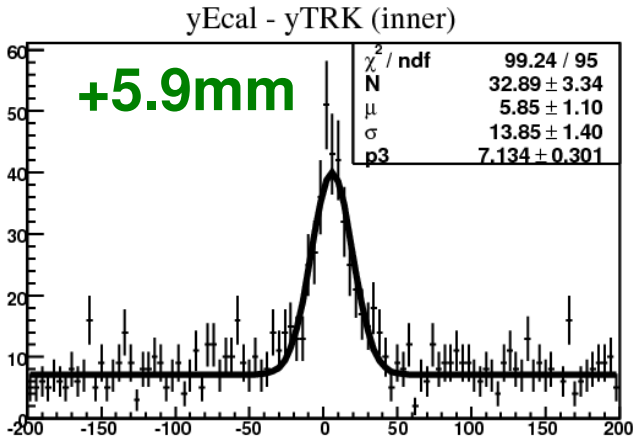
ΔX





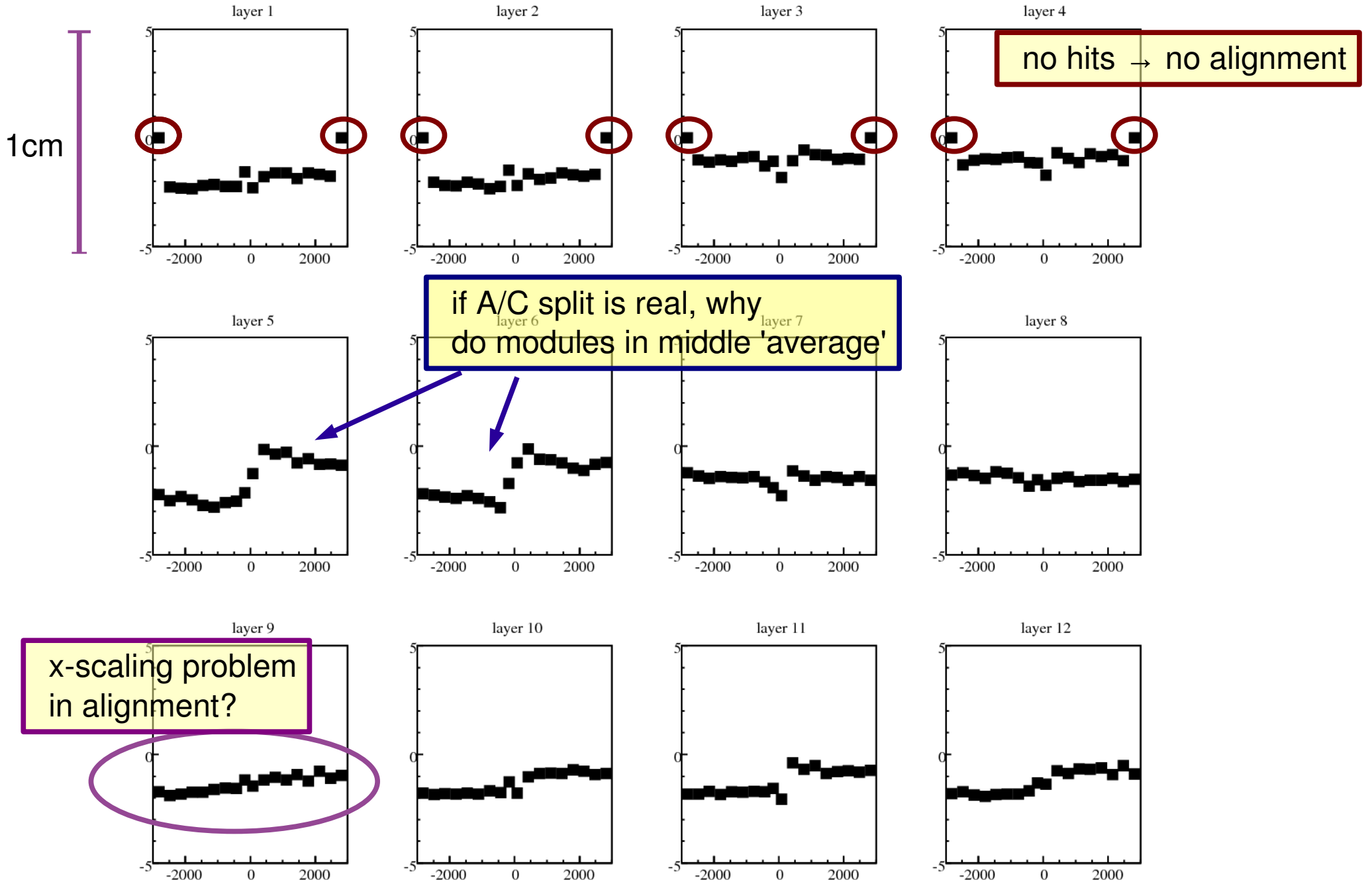
Δy





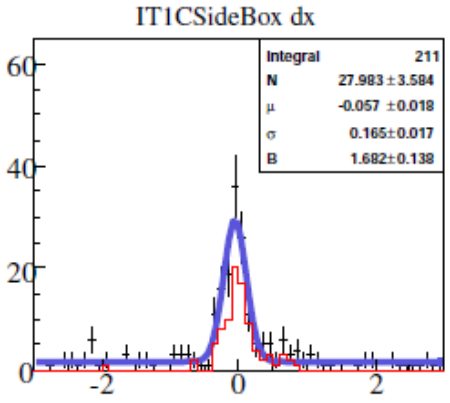
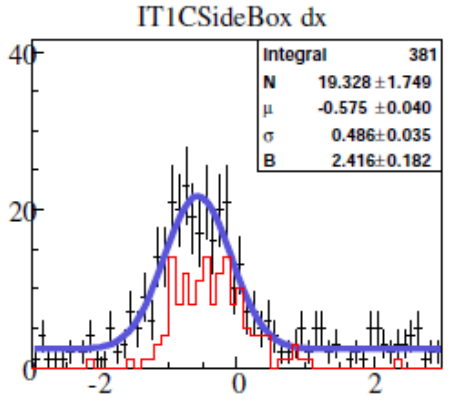
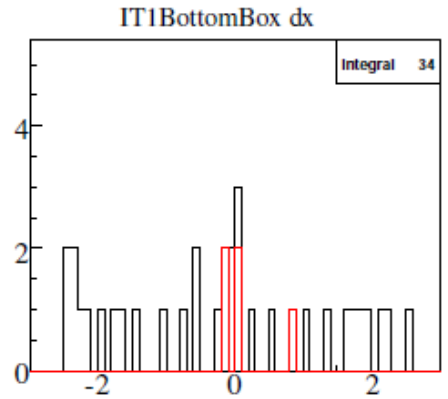
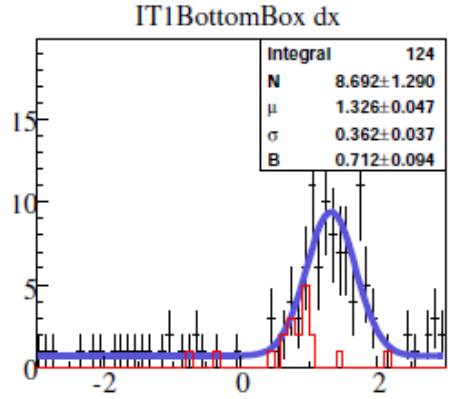
- this looked better last week … before I updated the OT survey

more questions: dX of OT modules including C-frames



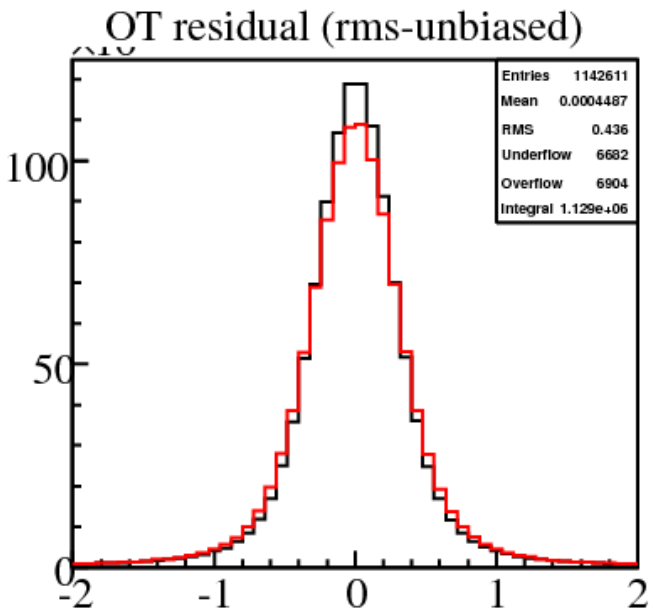
next step: testing this database on field-on

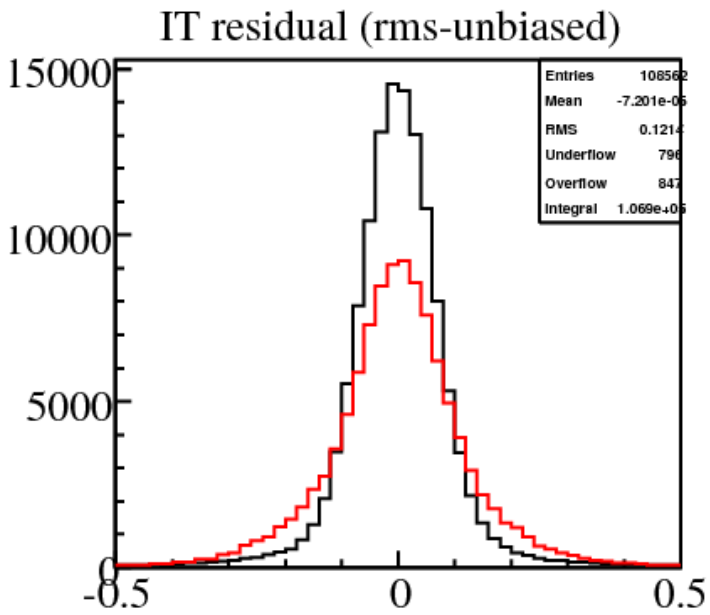
- **IT overlaps** look very strange in field-on data, e.g.

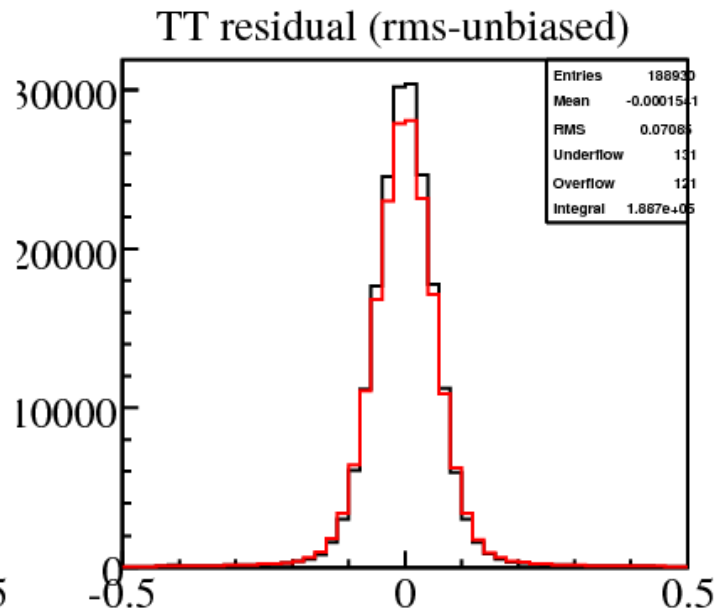
	OFF	ON
ITBottomTrack		
ITCSideTrack		

- also IT-OT overlaps look poor
- interesting: rest of IT stays more or less internally consistent
  - did Bottom and C boxes really moved ~1mm apart?
  - is it possible that support frames rotated around fixation point on rail?

- running alignment on magnet-on data improves situation
  - sufficient statistics to include IT ladder Tx, Rz alignment

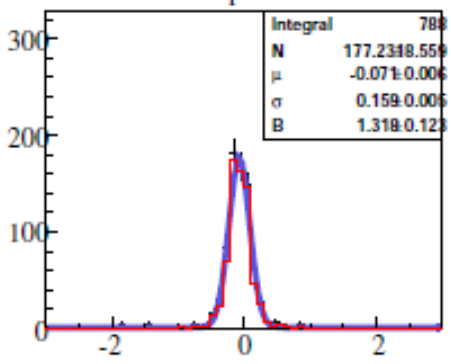




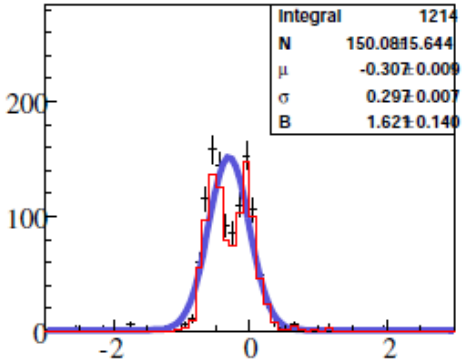


- improves overlaps … but not everywhere
- suspect that this has to do with top box Y shearing
- and this database is VERY BAD for magnet off ...

before



after

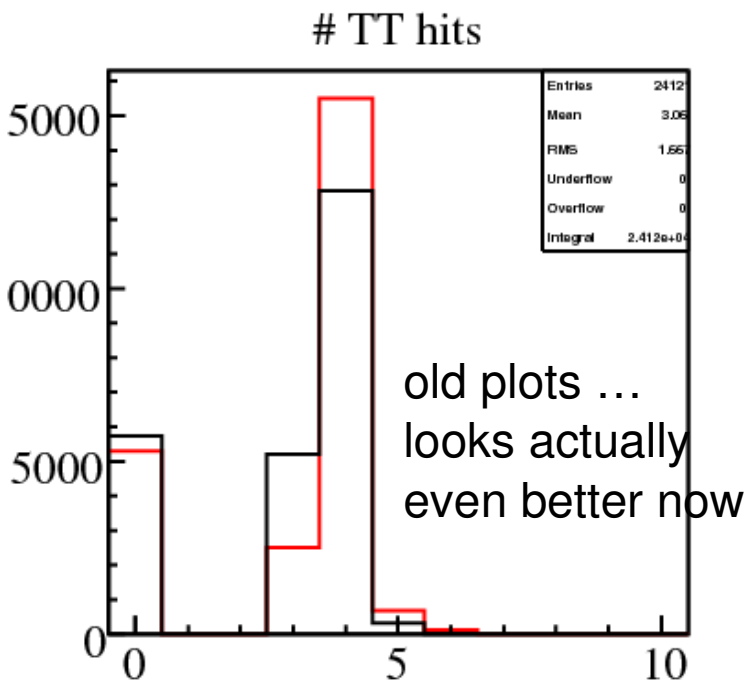


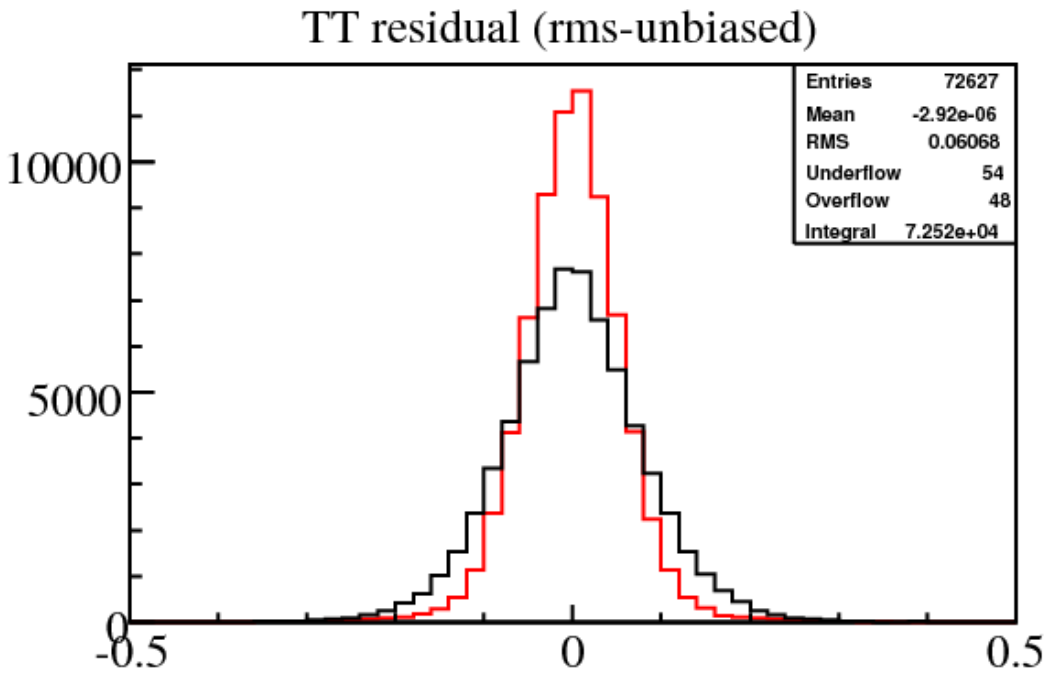
ITCSideTrack

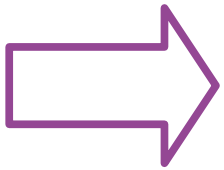
TT ladder alignment

- old news: TT ladder alignment improves TT tracking
  - see also Sascha's and Matt's talks

before module alignment
after module alignment







- more TT hits per track
- more downstream tracks
- much better residuals of TT hits

conclusions

- lot's of new functionality
  - implicitly deal with correlations by aligning different levels in hierarchy simultaneously
  - combine info from different datasets (thouh not actually used yet)
  - TrackHitAdder
- released a first magnet-off alignment
- prepared another alignment that just fixes TT
  - important improvement for physics … see Sascha's talk
  - release now, ot wait until magnet-on alignment better understood?
- magnet off alignment does NOT work very well for magnet on data
  - clearly visible in IT/OT overlaps
  - can partially fix this with re-alignment, but not everything
  - magnet-on database also doesn't work for magnet-off
  - needs a lot more work: need to figure out what moved, if anything
    - or could it be the field? how would you prove that?