Alignment Monitoring

Jim Pivarski

Texas A&M University

23 March, 2007

Trying to include all suggestions

1. DQM-based monitoring upstream of alignment process

2. Sanity checks in AlignmentProducer

3. Geometry validation: have the chambers moved?

4. Validation in reconstructed data: does the resolution improve?

Trying to include all suggestions

1. DQM-based monitoring upstream of alignment process
   - Monitors pre-loaded alignment (whatever is in HLT)
   - Compares with reference/reports an error if something's wrong
   - Most urgent: needs to be in CSA07 (not yet started)

2. Sanity checks in AlignmentProducer

3. Geometry validation: have the chambers moved?

4. Validation in reconstructed data: does the resolution improve?

Trying to include all suggestions

1. DQM-based monitoring upstream of alignment process
   - Monitors pre-loaded alignment (whatever is in HLT)
   - Compares with reference/reports an error if something's wrong
   - Most urgent: needs to be in CSA07 (not yet started)
2. Sanity checks in AlignmentProducer
   - Coverage, convergence, iterative improvement in residuals
   - Needs to be in AlignmentProducer's loop to book new histograms with each iteration
3. Geometry validation: have the chambers moved?
4. Validation in reconstructed data: does the resolution improve?

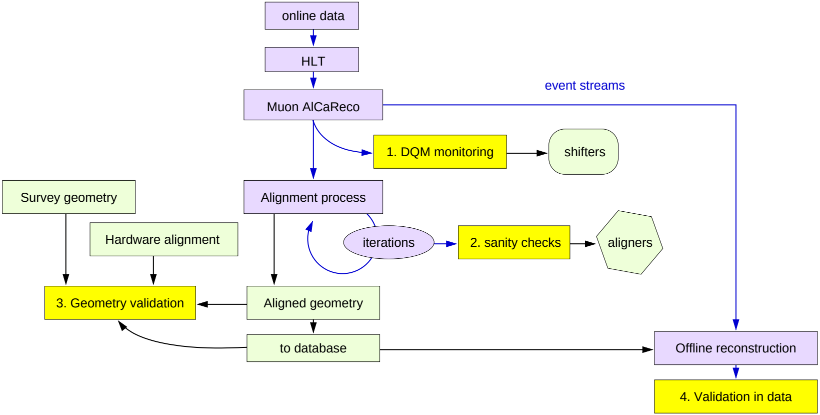
Trying to include all suggestions

1. DQM-based monitoring upstream of alignment process
   - Monitors pre-loaded alignment (whatever is in HLT)
   - Compares with reference/reports an error if something's wrong
   - Most urgent: needs to be in CSA07 (not yet started)
2. Sanity checks in AlignmentProducer
   - Coverage, convergence, iterative improvement in residuals
   - Needs to be in AlignmentProducer's loop to book new histograms with each iteration
3. Geometry validation: have the chambers moved?
   - Reads multiple geometries to look at differences/time dependence
   - Does not loop over tracks; reads geometries *only*
4. Validation in reconstructed data: does the resolution improve?

Trying to include all suggestions

1. DQM-based monitoring upstream of alignment process
   - Monitors pre-loaded alignment (whatever is in HLT)
   - Compares with reference/reports an error if something's wrong
   - Most urgent: needs to be in CSA07 (not yet started)
2. Sanity checks in AlignmentProducer
   - Coverage, convergence, iterative improvement in residuals
   - Needs to be in AlignmentProducer's loop to book new histograms with each iteration
3. Geometry validation: have the chambers moved?
   - Reads multiple geometries to look at differences/time dependence
   - Does not loop over tracks; reads geometries *only*
4. Validation in reconstructed data: does the resolution improve?
   - Confirms alignment in data (e.g. we installed the right geometry)
   - Same functionality as 1: compare to reference, same plots(?)

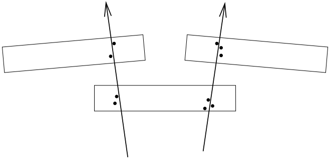
Where these fit into the big picture



- Histogram-filling modules attach to existing event streams; they don't require new loops

Plots that can be attached to any track loop (1, 2, and 4)

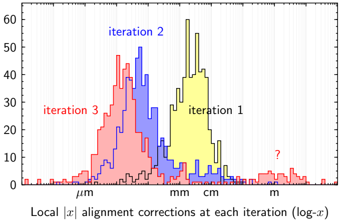
- J/ψ, Υ, Z dimuon mass spectrum
- p_T for selected events
- Residuals versus everything (R, ϕ, Z, chamber-by-chamber?)
- Overlap plots for physically overlapping chambers

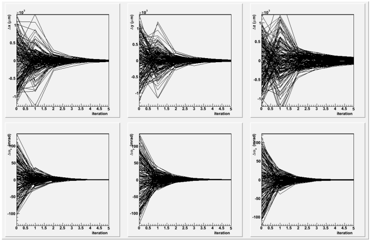


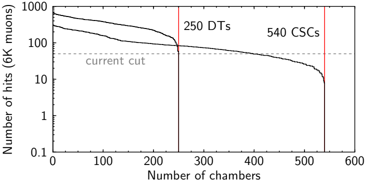
$\text{residual}_{\text{chamber 1}} - \text{residual}_{\text{chamber 2}}$

(track cancels, effectively a "ruler" curved by the $\vec{B}$ field)

Plots specifically for AlignmentProducer iterations



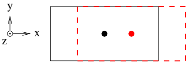




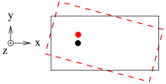
- Is the procedure converging (HIP mostly)?
- Are we missing any chambers (all algos)?
- Histograms need to know which iteration we're on

If there's enough memory in the budget. . .

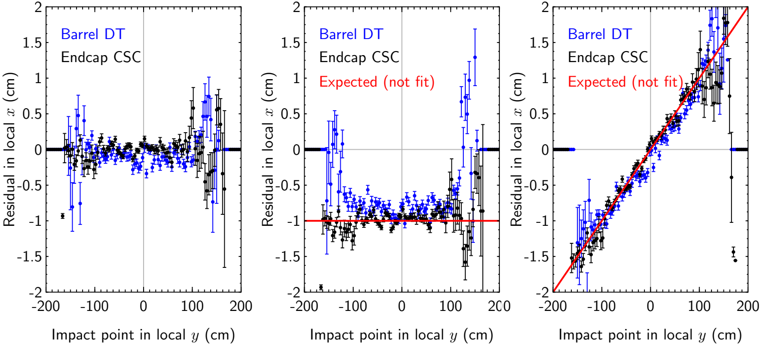
r_x vs x, y for *every* DT/CSC, and r_y vs y for *every* CSC



x: offset in r_x

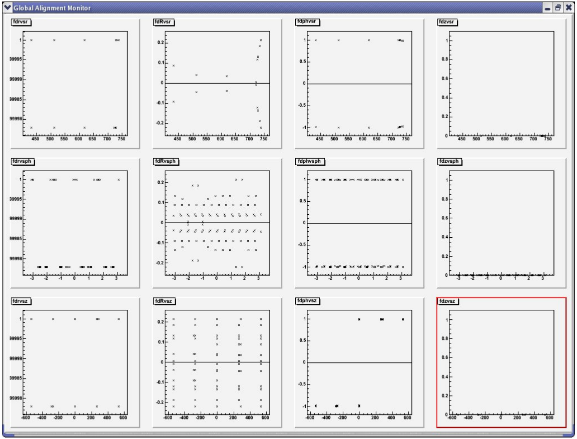


ϕ_z: r_x linear in y



Geometry Validation (early development)

Reads two geometries and takes their difference: $\Delta\vec{p}$



vs. R

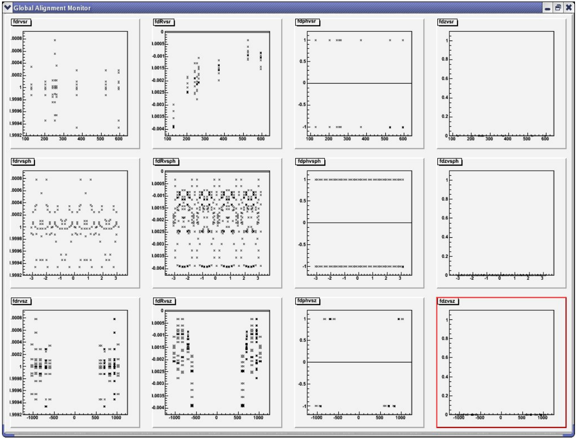
vs. ϕ

vs. Z

$|\Delta\vec{p}|$ $\Delta\vec{p}\cdot\hat{R}$ $\Delta\vec{p}\cdot r\hat{\phi}$ $\Delta\vec{p}\cdot\hat{Z}$ (DT)

Geometry Validation (early development)

Reads two geometries and takes their difference: $\Delta\vec{p}$



vs. R

vs. ϕ

vs. Z

$|\Delta\vec{p}|$ $\Delta\vec{p}\cdot\hat{R}$ $\Delta\vec{p}\cdot r\hat{\phi}$ $\Delta\vec{p}\cdot\hat{Z}$ (CSC)

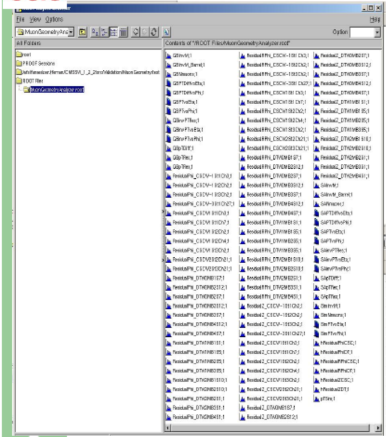
Validation in Reconstructed Data

CSIC

MuonGeometry Validation suite

I F C A

Instituto de Física de Cantabria



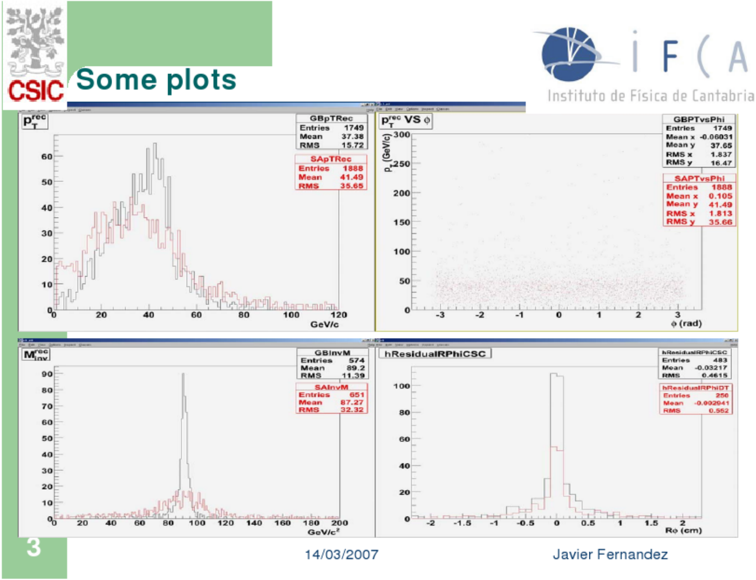
- **1D/2D Plots for SA & GB physics objects: pT, resolution, invariant mass,... vs ϕ & η**
- **Access to simulation hits and comparison**
- **Residuals (ϕ, Rϕ, Z) for individual DT & CSC hitted chambers, summary plot for DT & CSC,...**
- Easy configuration:

```
module myAnalyzer = MuonGeometryAnalyzer {
      untracked string DataType = "RealData"
#      untracked string DataType = "SimData"

      untracked bool  doSAplots = true
      untracked string StandAloneTrackCollectionLabel =
      "standAloneMuons"
      untracked bool  doGBplots = true
      untracked string GlobalMuonTrackCollectionLabel =
      "globalMuons"
      untracked bool  doResplots = true
      untracked string RecHits4DDTCollectionLabel =
      "dt4DSegments"
      untracked string RecHits2DCSCCollectionLabel =
      "cscSegments"
      untracked string rootFileName =
      "MuonGeometryAnalyzer.root"
}
```

2 14/03/2007 Javier Fernandez

Validation in Reconstructed Data



CSIC

Some plots

Instituto de Física de Cantabria

3

14/03/2007

Javier Fernandez

Who will do what?

1. DQM-based monitoring	Javier Fernandez?
2. Sanity checks in AlignmentProducer	Jim Pivarski?
3. Geometry Validation	Dmitry Yakorev, Jim Pivarski
4. Validation with reconstructed tracks	Javier Fernandez

Discussion?