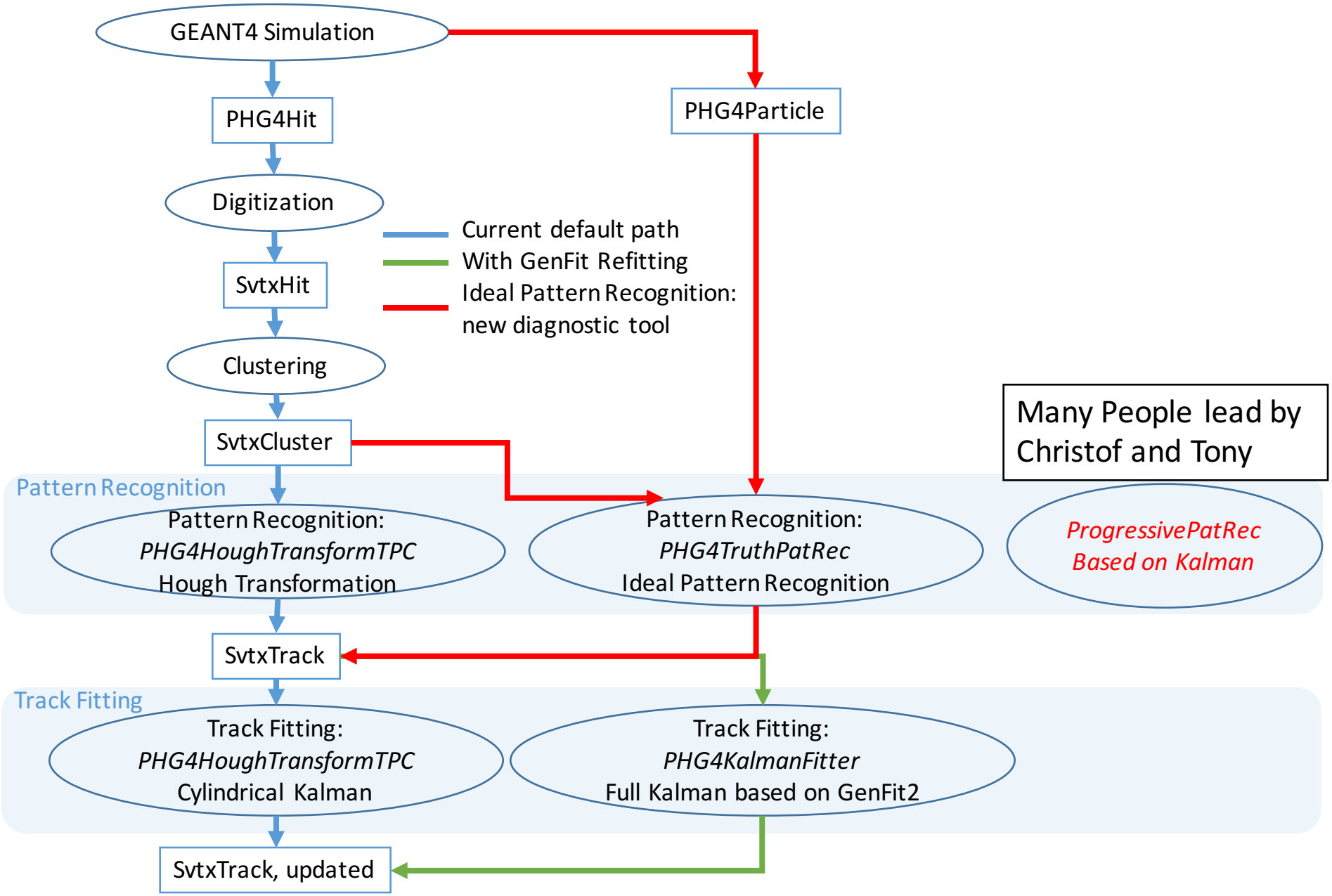
PHGenFit: Track Updater

Jin Huang(BNL), Christof Roland(MIT), Haiwang Yu (NMSU)

sPHENIX tracking



Christof's proposal

Building blocks for a combinatorial track finder (CTF)

CTF Workflow

Blocks needed

Vertexing

Write from scratch based on Pixel triplets?
For the time being -> Cheat V(0,0,0)

Iterate

Clustering

Code exists

Seeding

Use hough to get started
Replace later

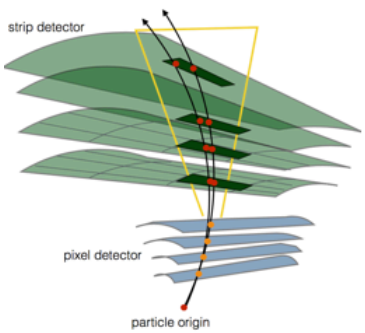
Trajectory Building

Most work goes here
- Detector navigation
- Trajectory book keeping
- Kalman updater

Ambiguity cleaning

Write once we need iterations to optimize for efficiency

Space point seeded track finding



Track fitting

Use GenFit. Done.

Quality selection

Apply offline

3D vertexing (optional)

Code exists

Christof Roland

4

sPhenix Tracking Meeting Mar 10 2017

Track updater with one measurement

PHGenFit::Track

genfit:::Track

vector::<AbsTrackRep>

TVectorD stateSeed_; // 6D: position, omentum TMatrixDSym covSeed_; // 6D

Only affect the starting TrackPoint => trackPointID == 0

vector::<TrackPoint>
= > map< AbsTrackRep*, AbsFitterInfo* >

Where measurements and KalmanFitterInfo resides

maps::<AbsTrackRep*, FitStatus>

Fitting Status: converged? chi2?

new

updateOneMeasurement (
vector<PHGenFit::Measurement*>& measurements,
map<double, PHGenFit::Track*>& incr_chi2s_new_tracks
)

return a map incr_chi2 ⟹ new Track

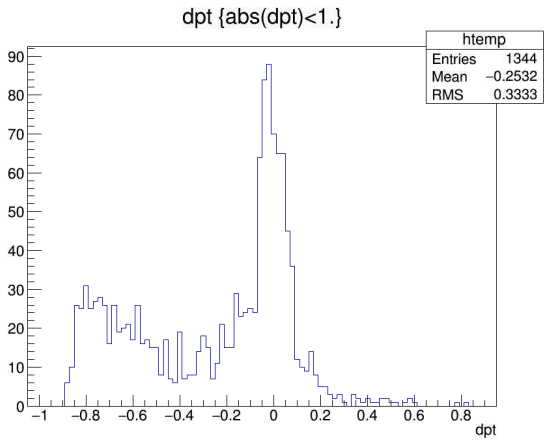
update

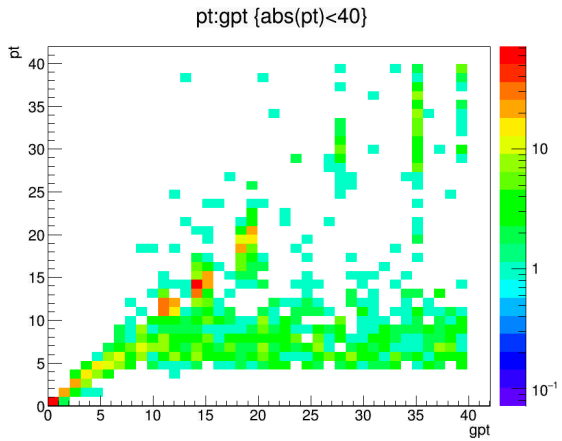
genfit::MeasuredStateOnPlane* extrapolateToCylinder(…)

extrapolation now works with tracks without fitting

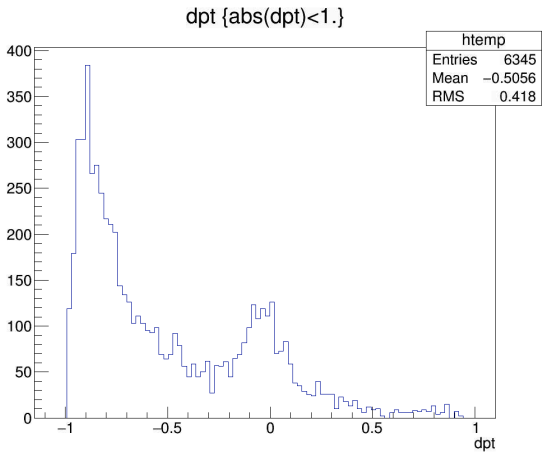
MAPS(0,1,2)+INTT(3,4,5,6)

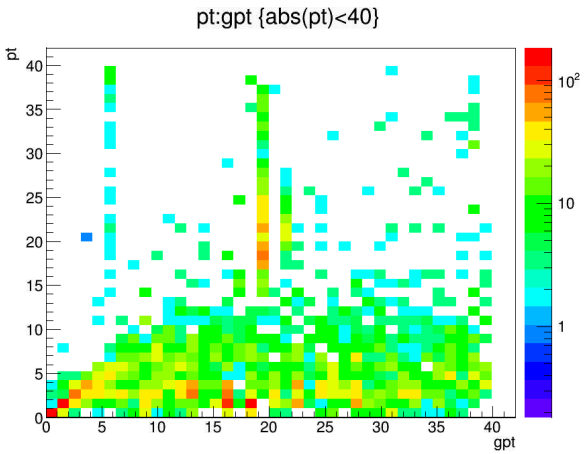
1pion/event





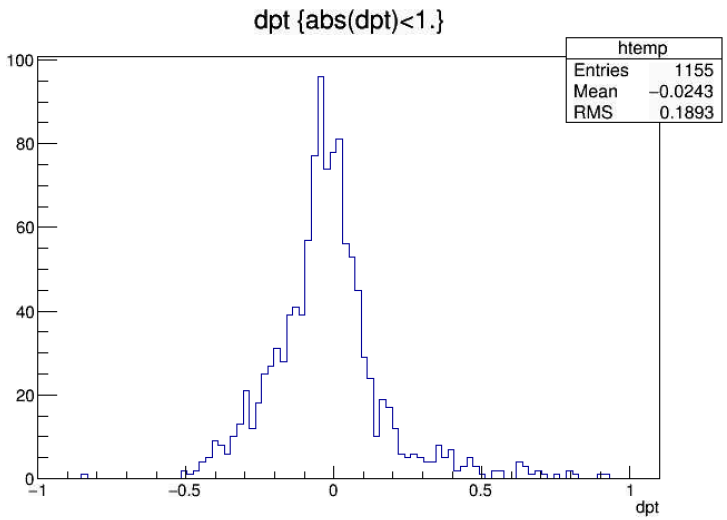
1000pion/event

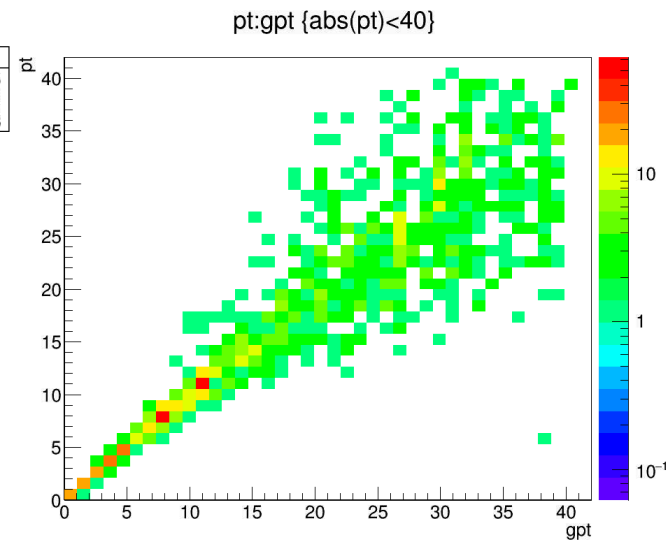




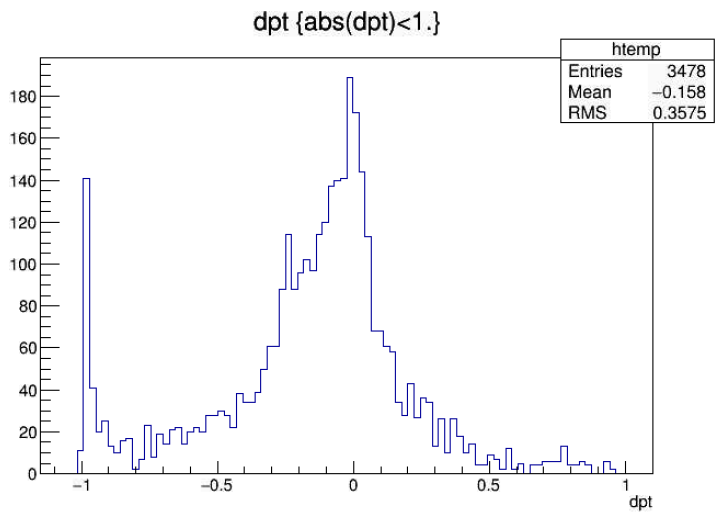
MAPS(0,1,2)+INTT(3,5)+TPC(7)

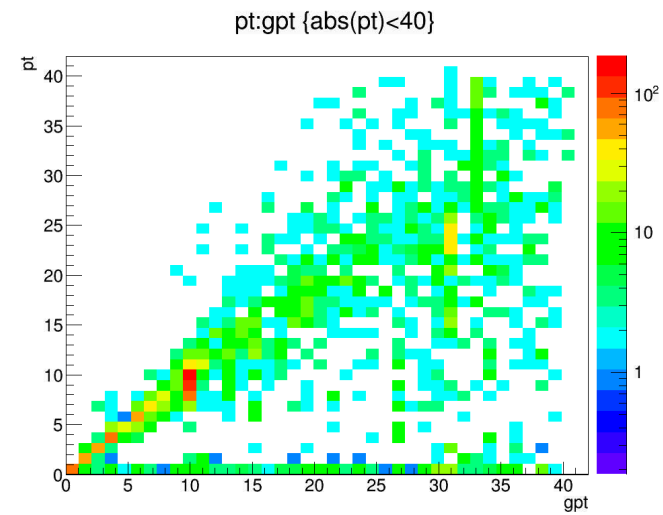
1pion/event



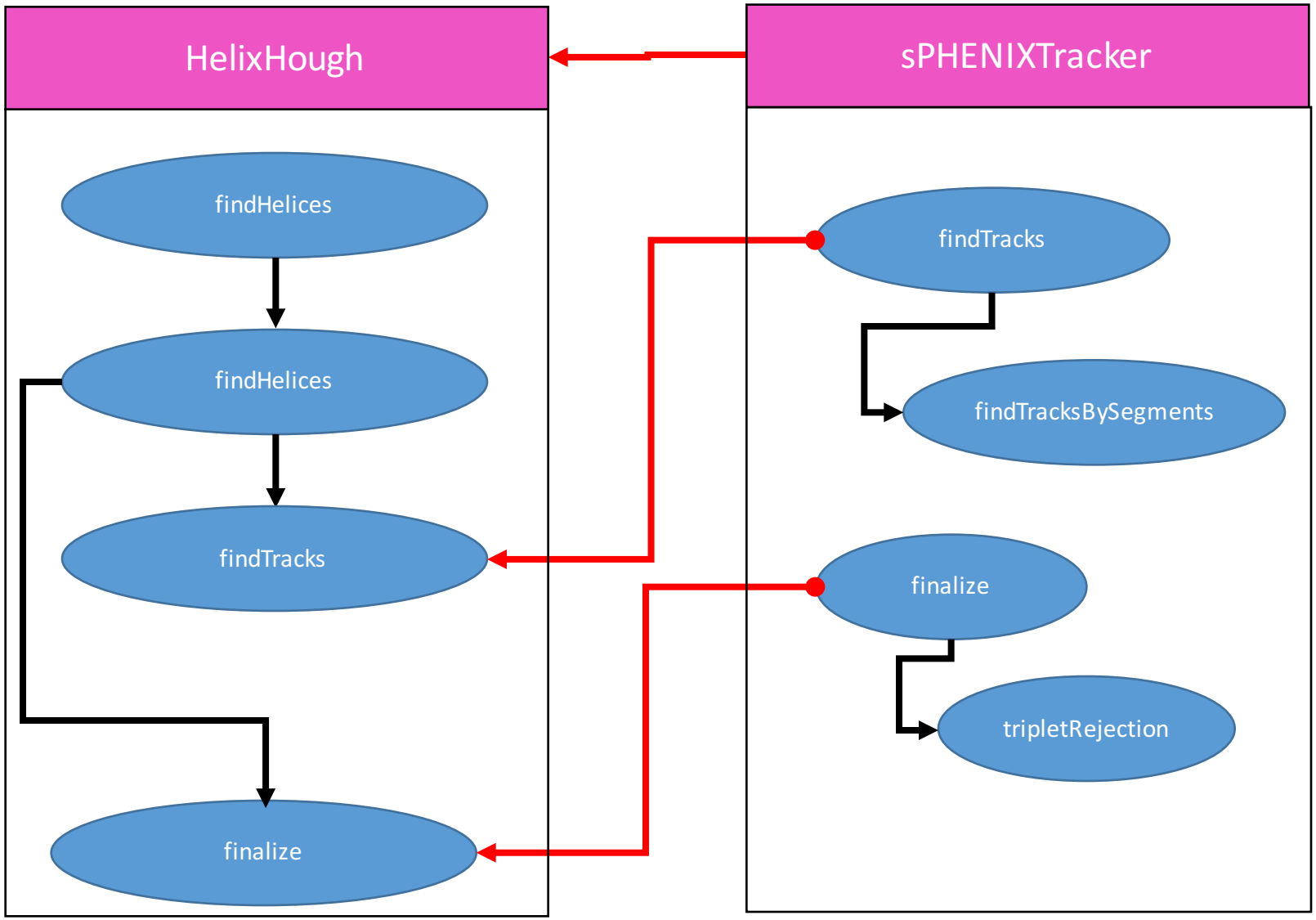


1000pion/event





Backups


HelixHough
findHelices
findHelices
findTracks
finalize
sPHENIXTracker
findTracks
findTracksBySegments
finalize
tripletRejection

KalmanFitStatus:FitStatus

```
double getForwardChi2() const {return fChi2_;}
double getBackwardChi2() const {return FitStatus::getChi2();}
double getForwardNdf() const {return fNdf_;}
double getBackwardNdf() const {return FitStatus::getNdf();}
// virtual double getPVal() : not overridden, as it does the right thing.
double getForwardPVal() const {return std::max(0.,ROOT::Math::chisquared_cdf_c(fChi2_, fNdf_));}
double getBackwardPVal() const {return FitStatus::getPVal(); }

void setNumIterations(unsigned int numIterations) {numIterations_ = numIterations;}
void setIsFittedWithDaf(bool fittedWithDaf = true) {fittedWithDaf_ = fittedWithDaf;}
void setIsFittedWithReferenceTrack(bool fittedWithReferenceTrack = true) {fittedWithReferenceTrack_ = fittedWithReferenceTrack;}
void setTrackLen(double trackLen) {trackLen_ = trackLen;}
void setForwardChi2(double fChi2) {fChi2_ = fChi2;}
void setBackwardChi2(double bChi2) {FitStatus::setChi2(bChi2);}
```

genfit:::Track

vector::<TrackPoint>
= > map< AbsTrackRep*, AbsFitterInfo* >

Where measurements and KalmanFitterInfo resides
So there is no "track state/cov" only "track state/cov at a TrackPoint"

vector::<AbsTrackRep>

maps::<AbsTrackRep*, FitStatus>

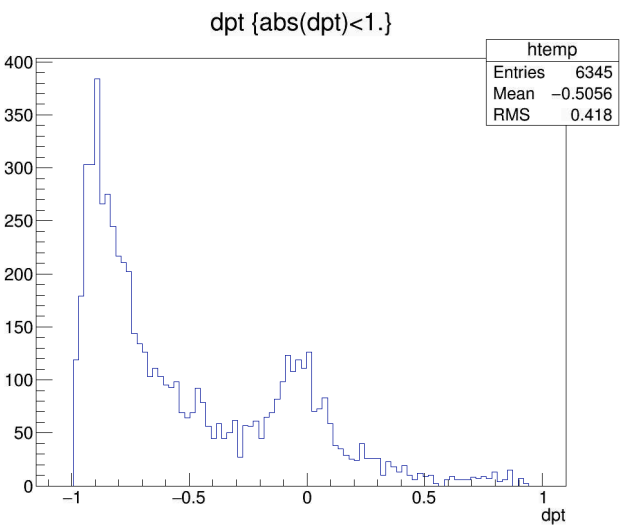
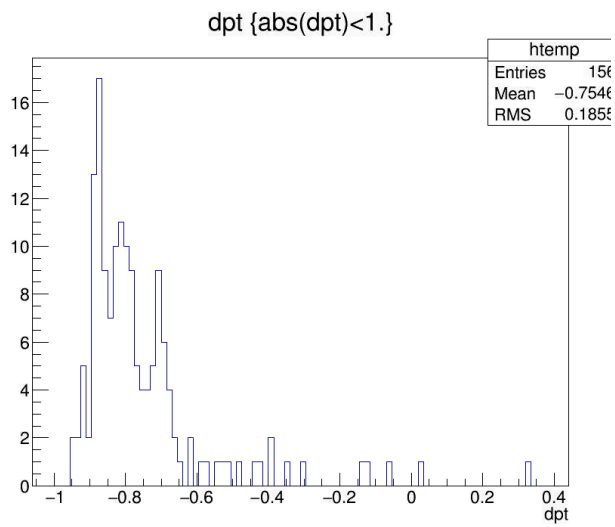
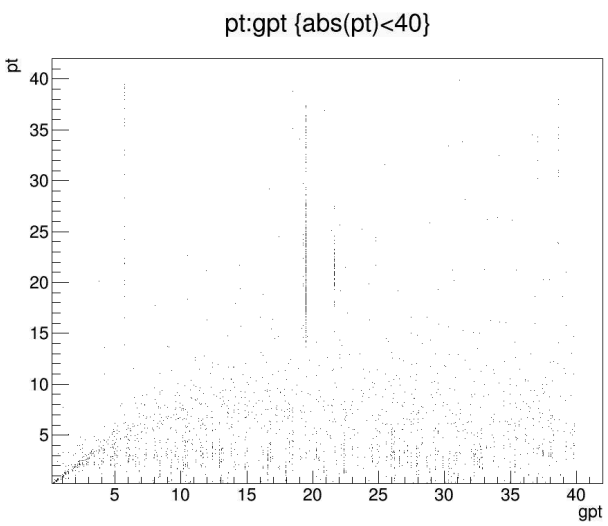
Fitting Status: converged? chi2?

TVectorD stateSeed_; // 6D: position, omentum
TMatrixDSym covSeed_; // 6D

Only affect the starting TrackPoint => trackPointID == 0

genfit::KalmanFitter

KalmanFitter::processTrackPoint(
Point* tp,
AbsTrackRep* rep,
le& chi2,
le& ndf,
int direction)

Cylinder MAPS+INTT seeding: (7, 7)

	Time	Finalized Seeds	
1 pion			
100 pion			
200 pion			
500 pion	93s	1014	
1000 pion	158s	6000