THREE POINT HELIX CHECK ERRORS

MIRIAM DIAMOND

AUGUST 9 2017

github issue 126

See last week's software meeting for technical details

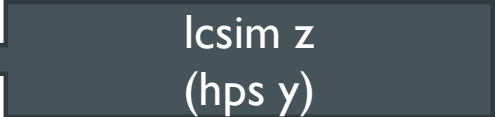# DOWN THE RABBIT-HOLE

- org.lcsim.recon.tracking.seedtracker.FastCheck ▶ThreePointHelixCheck
triplet-finding for track seeds

  - For each of the 3 hits, calculates contribution to *z error* ← lcsim z (hps y)

    ```
    dztot += _nsig * Math.sqrt(hit.getCovMatrix()[5]);
    ```

  - Then
    ```
    //  Add multiple scattering error here - for now, just set it to 1 mm
    dztot += 1.;   dztot += _nsig * MSerror;
    ```

  - Compares *total z error* to *(predicted – actual) z position* of middle hit

    ```
    if (Math.abs(zpred - z[1]) > dztot) return false;
    ```

- Implementing a proper MSerror makes ~no difference to tracking output. Why?
- Because *even without any MSerror*, dztot is far bigger than zpred-z[1], meaning no seeds get thrown out here anyway

2

DOWN THE RABBIT-HOLE

- Why is this potentially a problem?
  - We do want to avoid throwing out decent candidates at seeding stage, but if we're not throwing out any seeds, we might as well not bother with this check at all
  - Intuitively, dztot *should* be dominated by MSerror. But it *is* dominated by hit errors.
- Why are the hit errors so big?
  1. Big _nsig
  2. Big hit.getCovMatrix()[5]
  3. Contributions summed linearly, not in quadrature

OPTIONS

A. "Make seeding cuts great again" to throw out some seeds
  - Look at distributions of (phat.u) to get proper uncertainty for it in covariance matrix
  - Revisit strip.du() values (issue 135)
  - Perform dedicated studies to decide value of _nsig

B. Decide it's OK to keep all seeds
  - Simply eliminate dztot cut in ThreePointHelixCheck since it's not accomplishing anything

- Proto-study: performed reco and CPU Time profiles with
  - Aggressive (A): dztot summed in quadrature, _nsig=1
 - Conservative (A): dztot summed in quadrature, _nsig=2.5
  - (B)

OPTIONS

- ThreePointHelixCheck code that performs dztot cut takes up very little time (~0.5% of total TrackerReconDriver)

    - Other code in ThreePointHelixCheck, which performs min p_T cut properly (still needed!), takes longer

- Seeding cuts do eliminate a few tracks, but not many
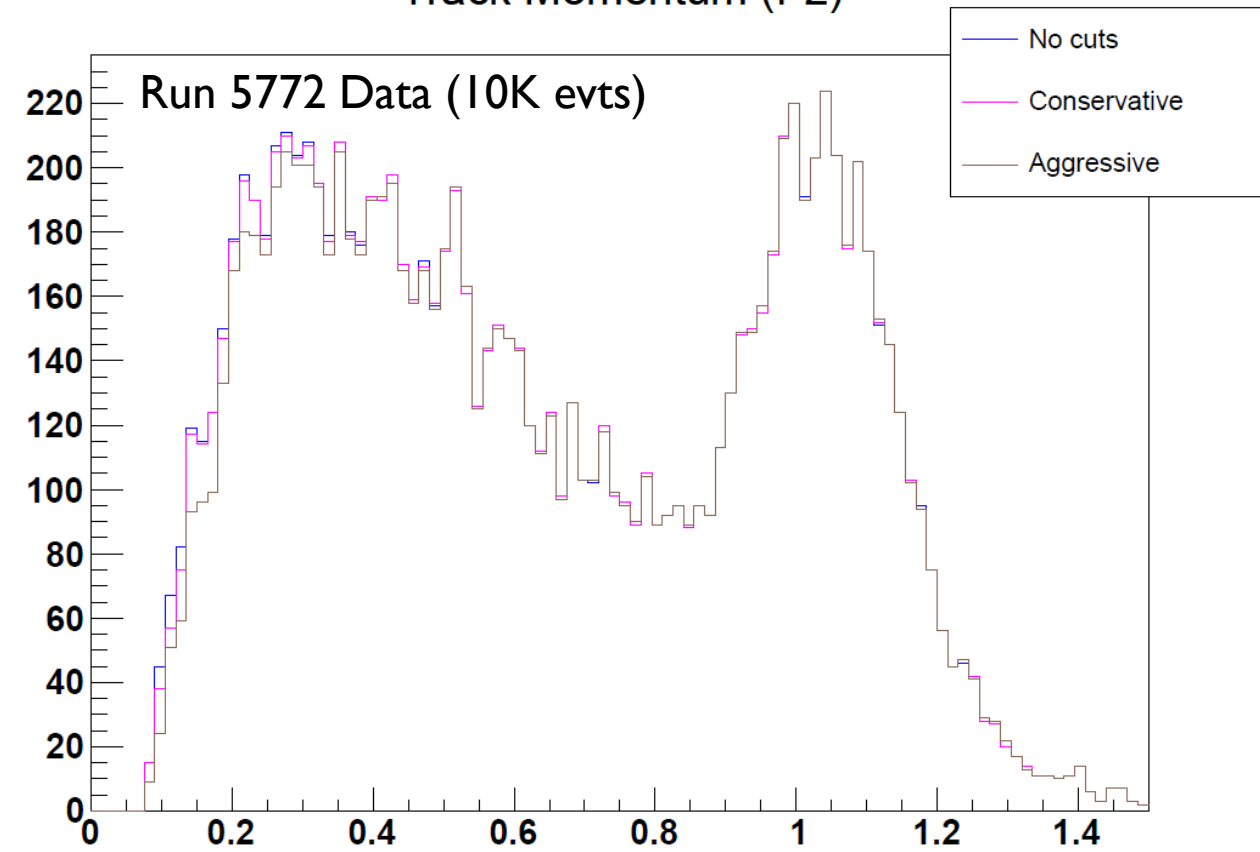
    - In data, more effect at low than at high p_T

	#Tracks (1700 MC Prompt A' Events)	#Tracks (10K Data Run 5772 Events)
(B)	2217	11329
Conservative (A)	2216	11290
Aggressive (A)	2211	11084

OPTIONS

Track Momentum (Pz)

Prompt A' MC (1700 evts)

Track Momentum (Pz)

Run 5772 Data (10K evts)
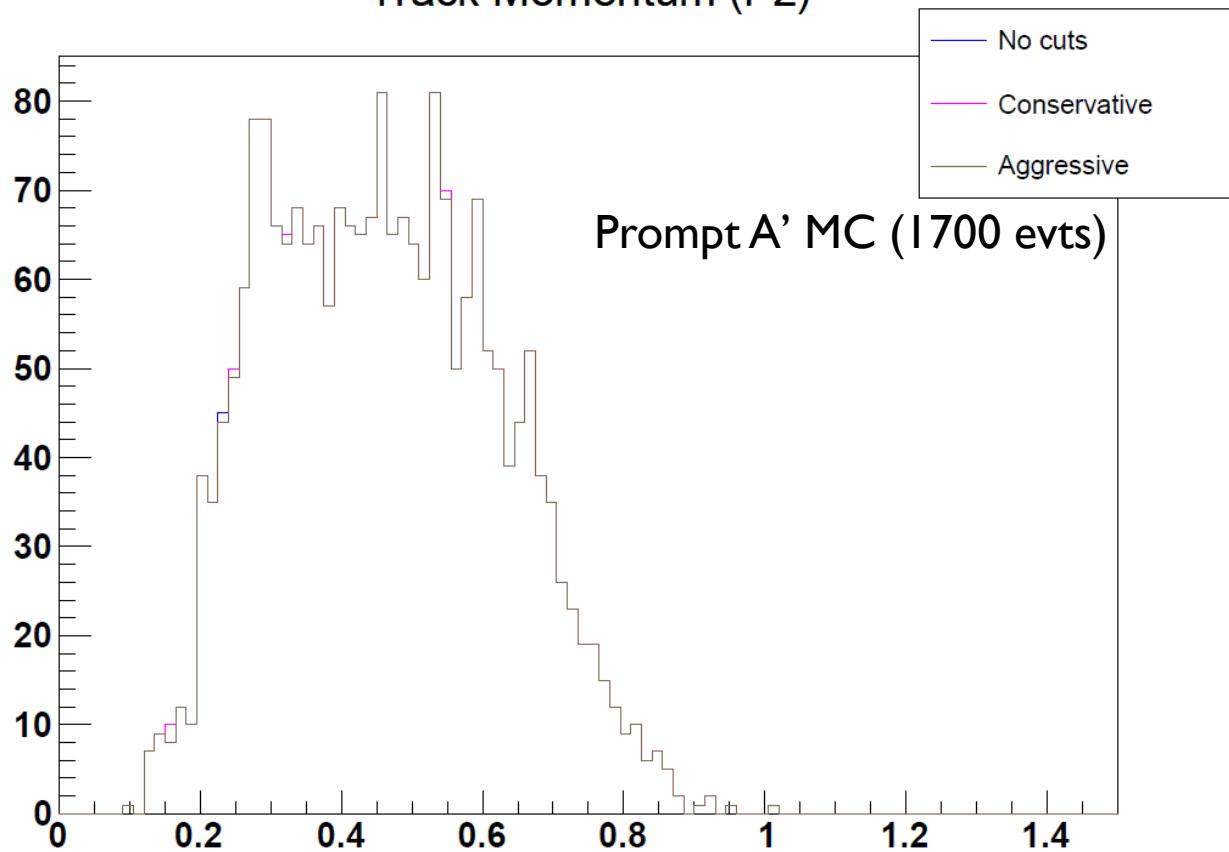
PROFILER: AGGRESSIVE (A)
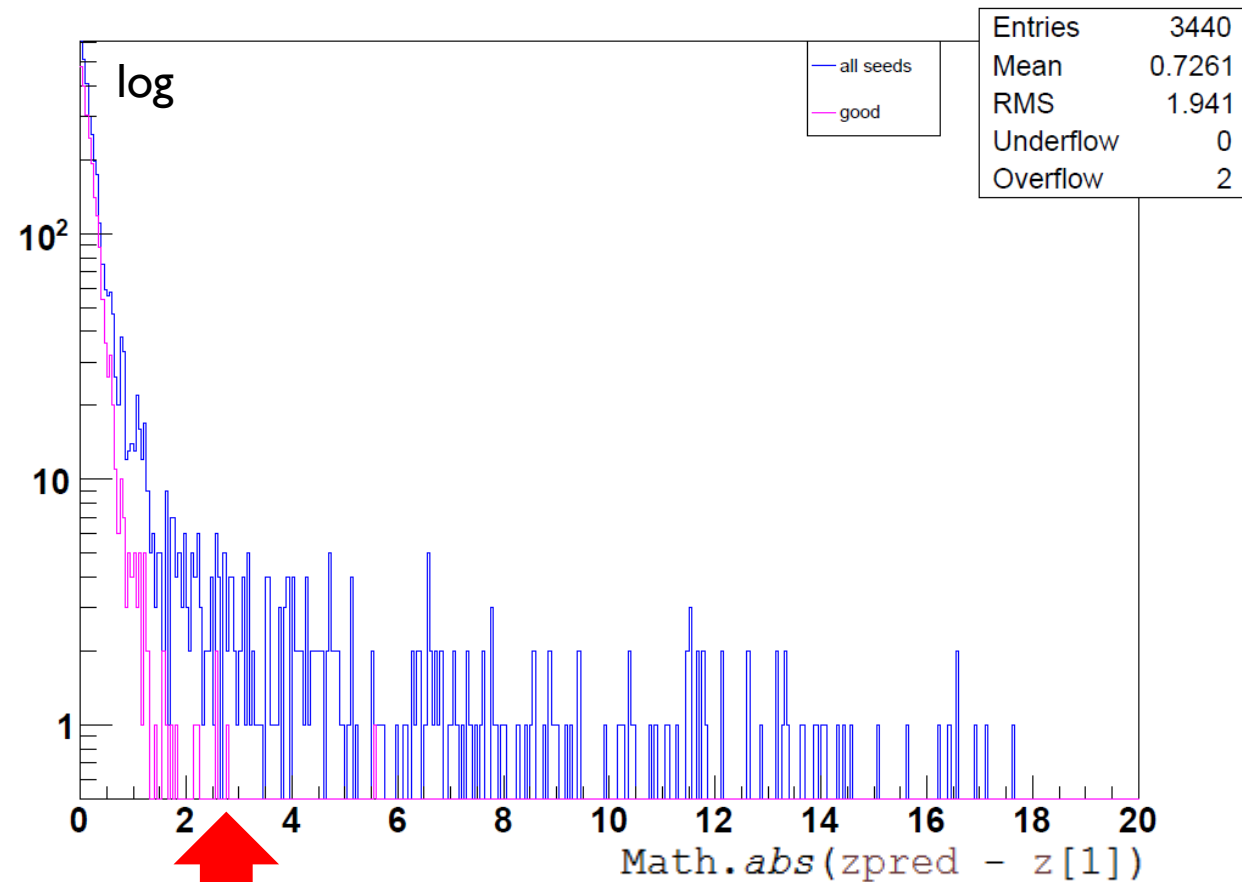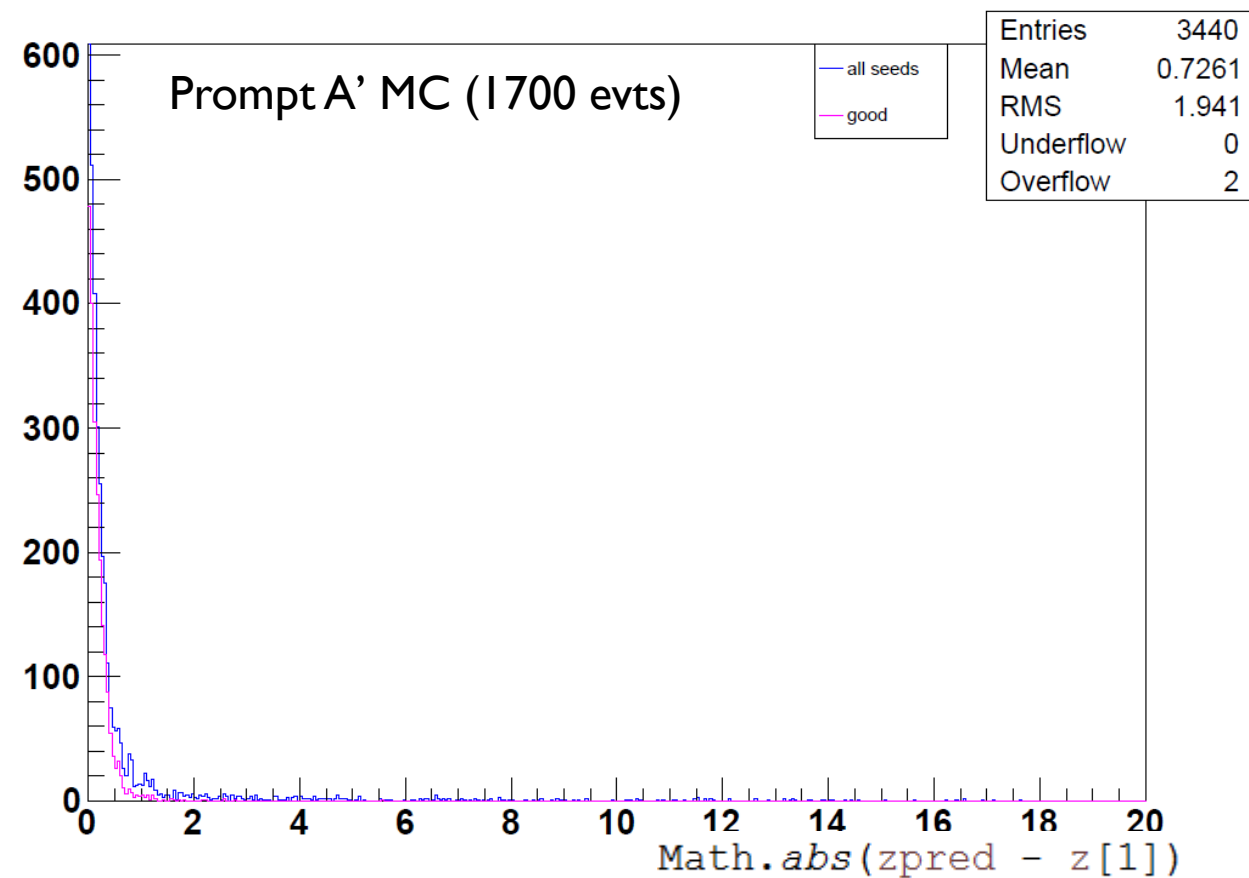
⊟ ⇘ org.hps.recon.tracking.TrackerReconDriver.**process** (org.lcsim.event.EventHeader)	10,670 ms	(3.5%)	500
⊟ ⇘ org.lcsim.util.Driver.**process** (org.lcsim.event.EventHeader)	10,665 ms	(3.5%)	500
⊟ ⇘ org.lcsim.util.Driver.**processChildren** (org.lcsim.event.EventHeader)	10,664 ms	(3.5%)	500
⊟ ⇘ org.lcsim.util.Driver.**doProcess** (org.lcsim.event.EventHeader)	10,664 ms	(3.5%)	500
⊟ ⇘ org.hps.recon.tracking.SeedTracker.**process** (org.lcsim.event.EventHeader)	10,663 ms	(3.5%)	500
⊟ ⇘ org.lcsim.recon.tracking.seedtracker.SeedTrackFinder.**FindTracks** (org.lcsim.recon.tracking.seedtracker.SeedStrategy, double	9,729 ms	(3.2%)	500
⊞ ⇘ org.lcsim.recon.tracking.seedtracker.ConfirmerExtender.**Extend** (org.lcsim.recon.tracking.seedtracker.SeedCandidate, org.l	6,046 ms	(2%)	3955
⊞ ⇘ org.lcsim.recon.tracking.seedtracker.ConfirmerExtender.**Confirm** (org.lcsim.recon.tracking.seedtracker.SeedCandidate, org.	1,852 ms	(0.6%)	1933
⊞ ⇘ org.lcsim.recon.tracking.seedtracker.HelixFitter.**FitCandidate** (org.lcsim.recon.tracking.seedtracker.SeedCandidate, org.lcs	1,504 ms	(0.5%)	2055
⊞ ⇘ org.hps.recon.tracking.FastCheck.**ThreePointHelixCheck** (org.lcsim.fit.helicaltrack.HelicalTrackHit, org.lcsim.fit.helicaltra	227 ms	(0.1%)	4730
⊞ ⇘ org.hps.recon.tracking.FastCheck.**TwoPointCircleCheck** (org.lcsim.fit.helicaltrack.HelicalTrackHit, org.lcsim.fit.helicaltrack	41.9 ms	(0%)	1663
◷ Self time	25.2 ms	(0%)	500
⊞ ⇘ org.hps.recon.tracking.HitTimeTrackCheck.**checkSeed** (org.lcsim.recon.tracking.seedtracker.SeedCandidate)	11.6 ms	(0%)	13060
⊞ ⇘ org.lcsim.recon.tracking.seedtracker.SeedSectoring.**<init>** (org.lcsim.recon.tracking.seedtracker.HitManager, org.lcsim.rec	10.4 ms	(0%)	500
⊞ ⇘ org.lcsim.recon.tracking.seedtracker.SeedCandidate.**addHit** (org.lcsim.fit.helicaltrack.HelicalTrackHit)	8.46 ms	(0%)	36436
◷ org.lcsim.recon.tracking.seedtracker.SeedCandidate.**<init>** (org.lcsim.recon.tracking.seedtracker.SeedStrategy, double)	1.44 ms	(0%)	15853
◷ org.lcsim.recon.tracking.seedtracker.FastCheck.**setDoSectorBinCheck** (org.lcsim.recon.tracking.seedtracker.SectorManag	0.054 ms	(0%)	500
⊞ ⇘ org.lcsim.recon.tracking.seedtracker.HelixFitter.**FitCandidate** (org.lcsim.recon.tracking.seedtracker.SeedCandidate, org.lcsim.	872 ms	(0.3%)	1692
⊞ ⇘ org.lcsim.recon.tracking.seedtracker.MakeTracks.**Process** (org.lcsim.event.EventHeader, java.util.List, double)	25.2 ms	(0%)	500
⊞ ⇘ org.lcsim.recon.tracking.seedtracker.HitManager.**OrganizeHits** (java.util.List)	20.6 ms	(0%)	500
◷ Self time	9.44 ms	(0%)	500
⊞ ⇘ org.hps.recon.tracking.FastCheck.**<init>** (org.lcsim.recon.tracking.seedtracker.SeedStrategy, double, org.lcsim.recon.tracking	4.87 ms	(0%)	500
◷ org.lcsim.recon.tracking.seedtracker.SeedTrackFinder.**clearTrackSeedList** ()	0.687 ms	(0%)	1000
⊞ ⇘ org.lcsim.event.base.BaseLCSimEvent.**get** (Class, String)	0.555 ms	(0%)	500
◷ org.lcsim.recon.tracking.seedtracker.FastCheck.**<clinit>**	0.008 ms	(0%)	1

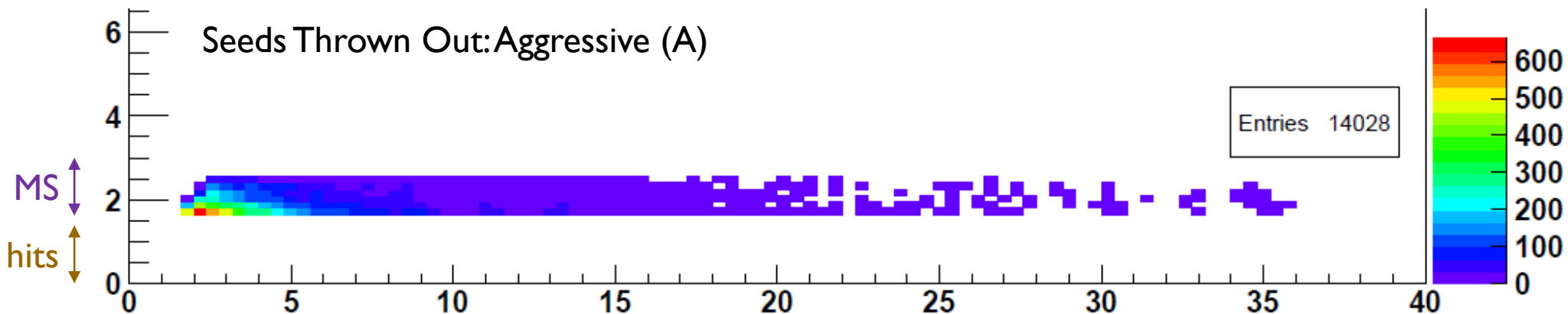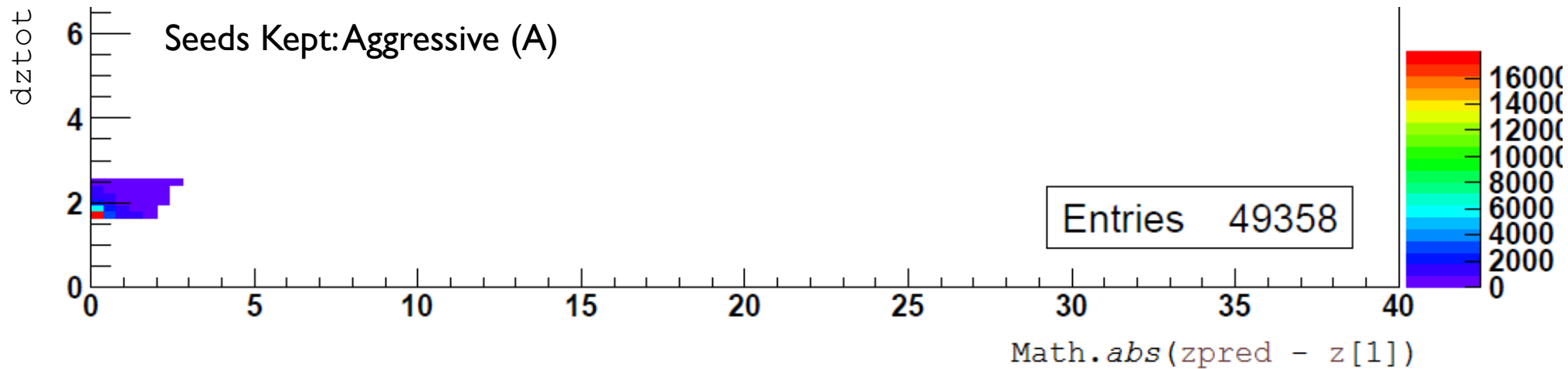# PROFILER: (B)

	Time	%	Count
⊟ org.hps.recon.tracking.TrackerReconDriver.**process** (org.lcsim.event.EventHeader)	12,980 ms	(4.3%)	500
⊟ org.lcsim.util.Driver.**process** (org.lcsim.event.EventHeader)	12,975 ms	(4.3%)	500
⊟ org.lcsim.util.Driver.**processChildren** (org.lcsim.event.EventHeader)	12,975 ms	(4.3%)	500
⊟ org.lcsim.util.Driver.**doProcess** (org.lcsim.event.EventHeader)	12,974 ms	(4.3%)	500
⊟ org.hps.recon.tracking.SeedTracker.**process** (org.lcsim.event.EventHeader)	12,974 ms	(4.3%)	500
⊟ org.lcsim.recon.tracking.seedtracker.SeedTrackFinder.**FindTracks** (org.lcsim.recon.tracking.seedtracker.SeedStrategy, double	12,046 ms	(4%)	500
⊞ org.lcsim.recon.tracking.seedtracker.ConfirmerExtender.**Extend** (org.lcsim.recon.tracking.seedtracker.SeedCandidate, org.l	7,370 ms	(2.4%)	5163
⊞ org.lcsim.recon.tracking.seedtracker.ConfirmerExtender.**Confirm** (org.lcsim.recon.tracking.seedtracker.SeedCandidate, org.	2,401 ms	(0.8%)	2462
⊞ org.lcsim.recon.tracking.seedtracker.HelixFitter.**FitCandidate** (org.lcsim.recon.tracking.seedtracker.SeedCandidate, org.lcs	1,950 ms	(0.6%)	2758
⊞ org.hps.recon.tracking.FastCheck.**ThreePointHelixCheck** (org.lcsim.fit.helicaltrack.HelicalTrackHit, org.lcsim.fit.helicaltra	213 ms	(0.1%)	4730
⊞ org.hps.recon.tracking.FastCheck.**TwoPointCircleCheck** (org.lcsim.fit.helicaltrack.HelicalTrackHit, org.lcsim.fit.helicaltrack	41.9 ms	(0%)	1663
⊕ Self time	34.1 ms	(0%)	500
⊞ org.hps.recon.tracking.HitTimeTrackCheck.**checkSeed** (org.lcsim.recon.tracking.seedtracker.SeedCandidate)	14.6 ms	(0%)	13598
⊞ org.lcsim.recon.tracking.seedtracker.SeedSectoring.**<init>** (org.lcsim.recon.tracking.seedtracker.HitManager, org.lcsim.reco	11.5 ms	(0%)	500
⊞ org.lcsim.recon.tracking.seedtracker.SeedCandidate.**addHit** (org.lcsim.fit.helicaltrack.HelicalTrackHit)	7.68 ms	(0%)	36436
⊕ org.lcsim.recon.tracking.seedtracker.SeedCandidate.**<init>** (org.lcsim.recon.tracking.seedtracker.SeedStrategy, double)	1.46 ms	(0%)	15853
⊕ org.lcsim.recon.tracking.seedtracker.FastCheck.**setDoSectorBinCheck** (org.lcsim.recon.tracking.seedtracker.SectorManage	0.046 ms	(0%)	500
⊞ org.lcsim.recon.tracking.seedtracker.HelixFitter.**FitCandidate** (org.lcsim.recon.tracking.seedtracker.SeedCandidate, org.lcsim.	867 ms	(0.3%)	1740
⊞ org.lcsim.recon.tracking.seedtracker.HitManager.**OrganizeHits** (java.util.List)	22.6 ms	(0%)	500
⊞ org.lcsim.recon.tracking.seedtracker.MakeTracks.**Process** (org.lcsim.event.EventHeader, java.util.List, double)	21.3 ms	(0%)	500
⊕ Self time	9.54 ms	(0%)	500
⊞ org.hps.recon.tracking.FastCheck.**<init>** (org.lcsim.recon.tracking.seedtracker.SeedStrategy, double, org.lcsim.recon.tracking	4.66 ms	(0%)	500
⊕ org.lcsim.recon.tracking.seedtracker.SeedTrackFinder.**clearTrackSeedList** ()	0.780 ms	(0%)	1000
⊞ org.lcsim.event.base.BaseLCSimEvent.**get** (Class, String)	0.547 ms	(0%)	500
⊕ org.lcsim.recon.tracking.seedtracker.FastCheck.**<clinit>**	0.008 ms	(0%)	1

WHERE "SHOULD" DZTOT CUT BE?

Prompt A' MC (1700 evts)

Entries	3440
Mean	0.7261
RMS	1.941
Underflow	0
Overflow	2

all seeds
good

$Math.abs(zpred - z[1])$

log

Entries	3440
Mean	0.7261
RMS	1.941
Underflow	0
Overflow	2

all seeds
good

$Math.abs(zpred - z[1])$

~ reasonable cut

WHERE "SHOULD" DZTOT CUT BE?

Seeds Kept: Aggressive (A)

Entries   49358

Math.abs(zpred − z[1])

Seeds Thrown Out: Aggressive (A)

Entries   14028

MS

hits

MORE ON COVARIANCES

hit.getCovMatrix()[5] fetches z entry from matrix uncorrected for track dir'n

- Typical uncorrected covariance matrix:

$$\begin{pmatrix} 0.3 & -11.5 & -0.6 \\ -11.5 & 400.5 & 20.0 \\ -0.6 & 20.0 & \boxed{2.0} \end{pmatrix}$$

Typical corrected covariance matrix:

$$\begin{pmatrix} 6e\text{-}5 & -2e\text{-}3 & -1e\text{-}4 \\ -2e\text{-}3 & 0.08 & 4e\text{-}3 \\ -1e\text{-}4 & 4e\text{-}3 & \boxed{2e\text{-}4} \end{pmatrix}$$

In uncorrected cov matrix calculation, uncertainty factor due to unknown track dir'n ~5x too big

lcsim-tracking ▸ 📁 src/main/java ▸ 📦 org.lcsim.fit.helicaltrack ▸ Ⓖ HitUtils ▸ CovarianceFromOrigin(HelicalTrackStrip, HelicalTrackStrip) : SymmetricMatrix

```java
// Calculate the uncertainty in the unmeasured coordinate due to not knowing the track direction
// by assuming phat . u has an uncertainty of 2/sqrt(12) so dv = 2 / sqrt(12) * seperation / sin(alpha)
double dv = Math.abs(2. * separation / (Math.sqrt(12) * salpha));
```

MORE ON COVARIANCES

- But … reducing this factor causes side effect! More frequent error message:

  org.hps.recon.tracking.TrackerReconDriver process :: Discarding track with bad HelicalTrackHit (correction distance 0.000000, chisq penalty 0.000000)

- Why? org.lcsim.fit.helicaltrack.HelicalTrackCross.setTrackDirection (which corrects HelicalTrackHits) has a `errok` check

  - if check fails, calls resetTrackDirection, which puts correction and chi-squared to 0 instead of setting them properly

  - check relies partly on uncorrected covariance matrix cov!

  - could try changing _eps arbitrarily, but would like to know rationale behind this check

```
//  Check to make sure we have sane errors in r-phi, r, and z - problems can occur
//  if the track direction is nearly parallel to the sensor plane
boolean errok = (drphicalc(poscor, covcor) < drphicalc(pos, cov)   + _eps) &&
        (drcalc(poscor, covcor)    < drcalc(pos, cov)       + _eps) &&
        (Math.sqrt(covcor.e(2,2))  < Math.sqrt(cov.e(2,2)) + _eps);
```