Track–bump matching disaster
What happened?

1) A significant part of the new track-cluster matching was committed by Gagan and released by me. The tag included also cleanups.

2) David Lang discovered changes in Emc distributions which were not understood.

3) David backed out the EmcReco tag

4) More minor changes/cleanups to EmcReco

5) I confuse tags and think the important changes are in, only cleanups including some bug have been backed out. This is wrong.

6) Disaster.
Current situation

• All R16 and R18 releases contain this bug.

  • When **reading or skimming** data which has been processed/converted in/to R14 EmcTMInfo objects will contain wrong consistency values. The consistency value is not used in official code, as far as I know. If somebody uses this number in his privat code, he gets wrong consistencies mainly for low energy tracks. Actual trk-bmp matches are not changed.
    → This is probably not a big deal.

  • When **processing** data, the track matches will be wrong.
    → This will cause all kinds of serious problems.
What to do?

- Can we solve this without reprocessing?
  
  - Track matching is redone at read time (cache mode) but the information
  - is only used to create EmcTMInfo objects.
  - In a special read mode persisted BtaCandidates are basically thrown away and recreated on basis of the recomputed track-matching.
  - During Skimming the new created BtaCands will be persisted.

  → We may be able to fix this during skimming, BUT

- what is with Cluster splitting? Does it depend on track-bump matching?
  
  → If yes, we probably can't fix it during skimming
New tag with Gagan's original code

This tag has EmcGeomTrkMatchMethod with Gagan's original code.

**EmcReco rsd-01Jun05-1**

It works in 18.1.0 - you probably need to recompile:

```plaintext
BetaMini
BetaMiniSequences
BetaMiniUser
EmcDataK
EmcSequence
```

To make the changes visible:

```plaintext
talkto LoadMiniBtaCandidates {
  ListType set Reco
  UseEmcIdentify set false
}
```

This code has known problems!
It may only be used to investigate this issue!
What needs to be done next?

Asume Cluster Splitting does not depend on track match bug:

• The 'reco' mode read code relies on David Browns replacement for EncIdentify. This does not produce the same results as EmcIdentify. Needs to be understood and fixed.

• The initial problem which caused the EmcReco tag to be backed out needs to be understood.

• Put together a release which can be used for skimming and check if it is really doing the right thing.

• We obviously don't have reliable QA to detect this problem

Asume Cluster Splitting does depend on track match bug:

• do as above
• Try to find a work around or reprocess
The initial problem