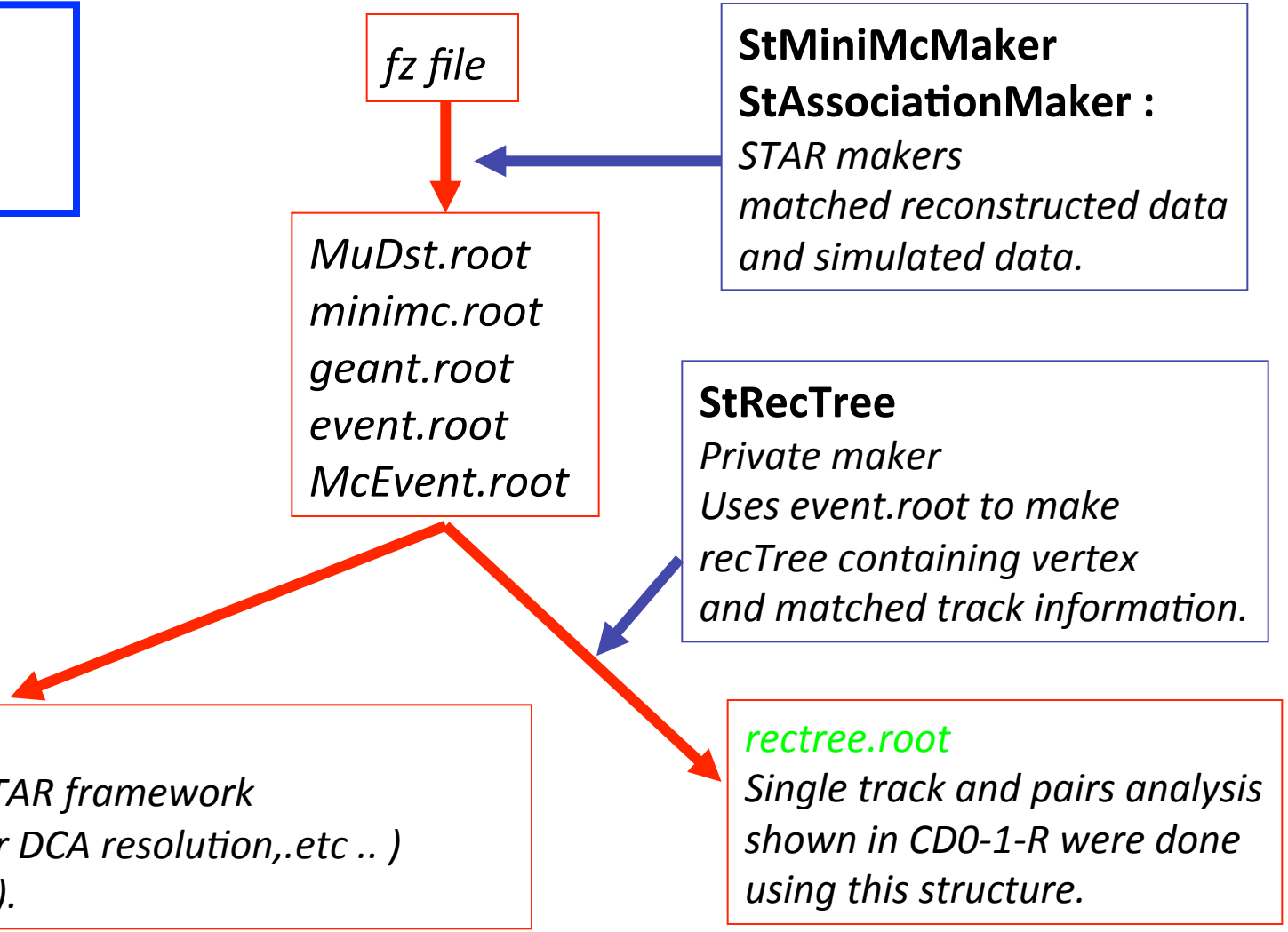
Evaluation Structures

J. Bouchet/S. Margetis, KSU

- Evaluation/Analysis structures
  - Current (STAR) way
  - An example (to avoid)

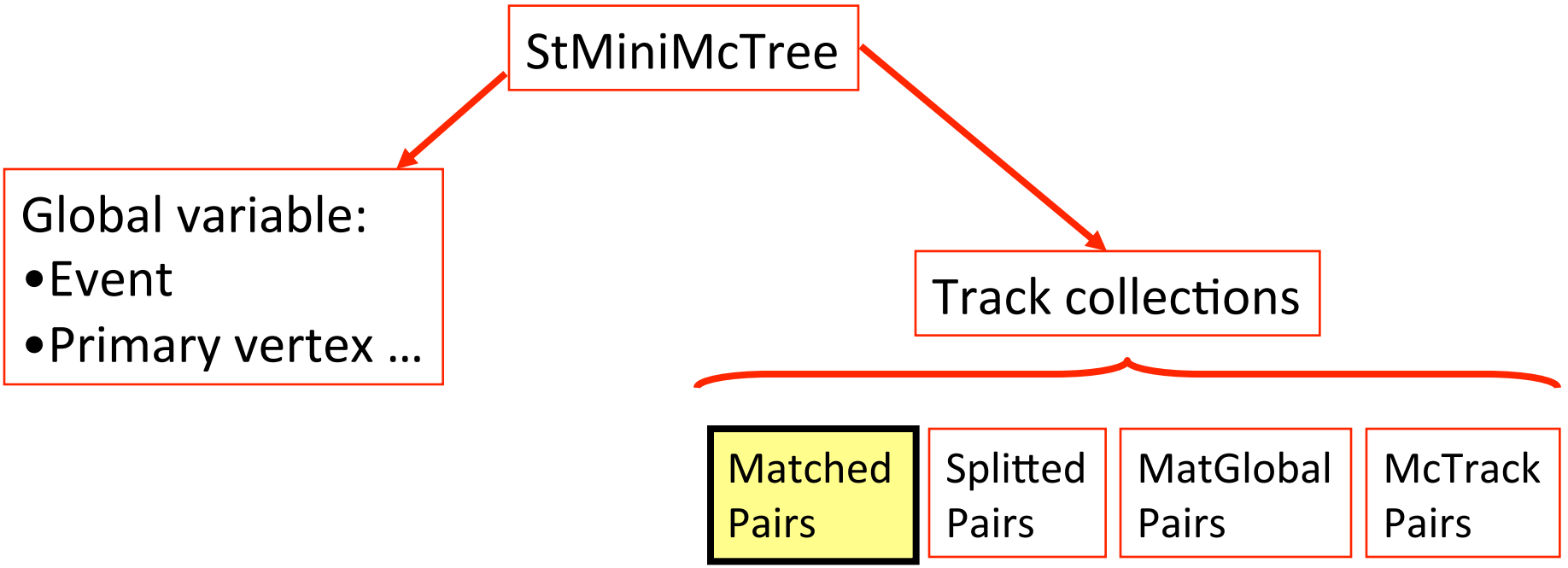
- The things we can do/propose

Chain analysis



→Does the minimc.root contains all the information needed for our analysis ?
→ If not, proposal to modify (without breaking its scheme) the current minimc tree with our needs.

StMiniMcTree structure



1. All track collections have the same tree structure.
2. For analysis we are using mostly the matchedPairs collection.

MiniMcTree structure

- **Event level :**
  - ☑Reconstructed primary vertex position and geant primary vertex position.
  - ☑Multiplicity, number of tracks, event id.
- **Track level (for both the MonteCarlo track and the reconstructed track) :**
  - ☑Kinematic : Pt,Pz,η,ϕ
  - ☑Geant information : particle geant id, particle parent geant id, start and stop points.
  - Number of hits in detector used during tracking :
    - ☑TPC, SSD
    - ☒PIXEL, IST
    - ☒TOF information : NTOF hits, TOF Tray Id, TOF module, TOF local hits, TOF pathLength.

☑(☒)means (not) in common in RecTree

MiniMcTree structure (cont')

The DCA is also needed for analysis

1. **in minimc**:
   1. dcaXYPr, dcaZPr,dcaXYPrMcV,dcaZPrMcV are saved : DCA of primary tracks wrt the reconstructed and MC primary vertex
   2. The same for global track

   ➔We should make sure that it corresponds to DCA filled in MuDst tree.
2. **in rectree** :
   1. DCAXYPr (DCA in transverse plan wrt the reconstructed primary vertex) and DCAZPr (DCA along Z) are reconstructed (the same is saved also wrt the MC primary vertex)

   ➔We should also make sure that studies done minimc reproduce the results done with RecTree.

Answer to question

4) Then propose what info might be saved in minimc

➔ currently :

- Minimc has only information about reconstructed track (daughters tracks) :
  - Its start and stop point.
  - For evaluation of secondary vertex, we need the stop point of the parent.
  - This information is saved in geant.root

➔To avoid multiple loops over different files to recover this information, we would like this information in the minimc tree.

➔We need the PIXEL and IST hits in the minimc tree

- done in private code but should check the results.
- issue : code crashes in DEV (bug #2137) because of StIstHitLoader

➔We need the TOF information

- Not only related to HFT.

Answer to question

5) Then propose some kind of structure for HFT (initial proposal)

➔Kalman secondary vertex for pairs analysis.

➔Should be directly integrated in a general output tree containing event,vertex, track information that all users can use. (according to their analysis).

➔That means to have a code in /offline/users/HFT (or any official place) where MuDst (real data) are processed to add the secondary vertex information.

A case study – SVT/SSD analysis of D^0s

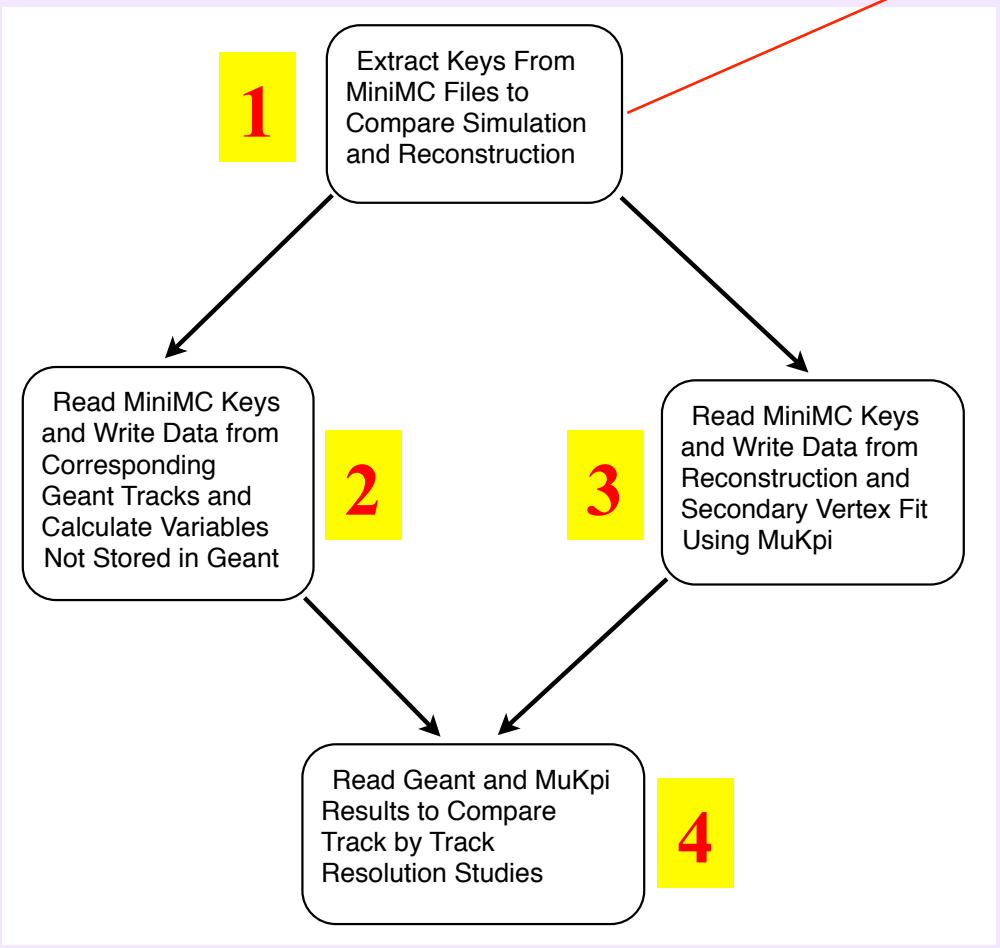
The following quantities are filled for each event
RunId : id of run
EventId : id of event
Vz : primary vertex Z position
NTrk : number of all tracks
gRefMult : gRefMultiplicity
QXE: component X of the Q vector using tracks with eta >0
QYE : component Y of the Q vector using tracks with eta >0
QXW : component X of the Q vector using tracks with eta <0
QYW : component Y of the Q vector using tracks with eta <0
EPE: event plane using using tracks with eta>0, before recentering
EPW : event plane using using tracks with eta>0, before recentering
Candidates : number of pairs K-pi
EventPlane : not filled

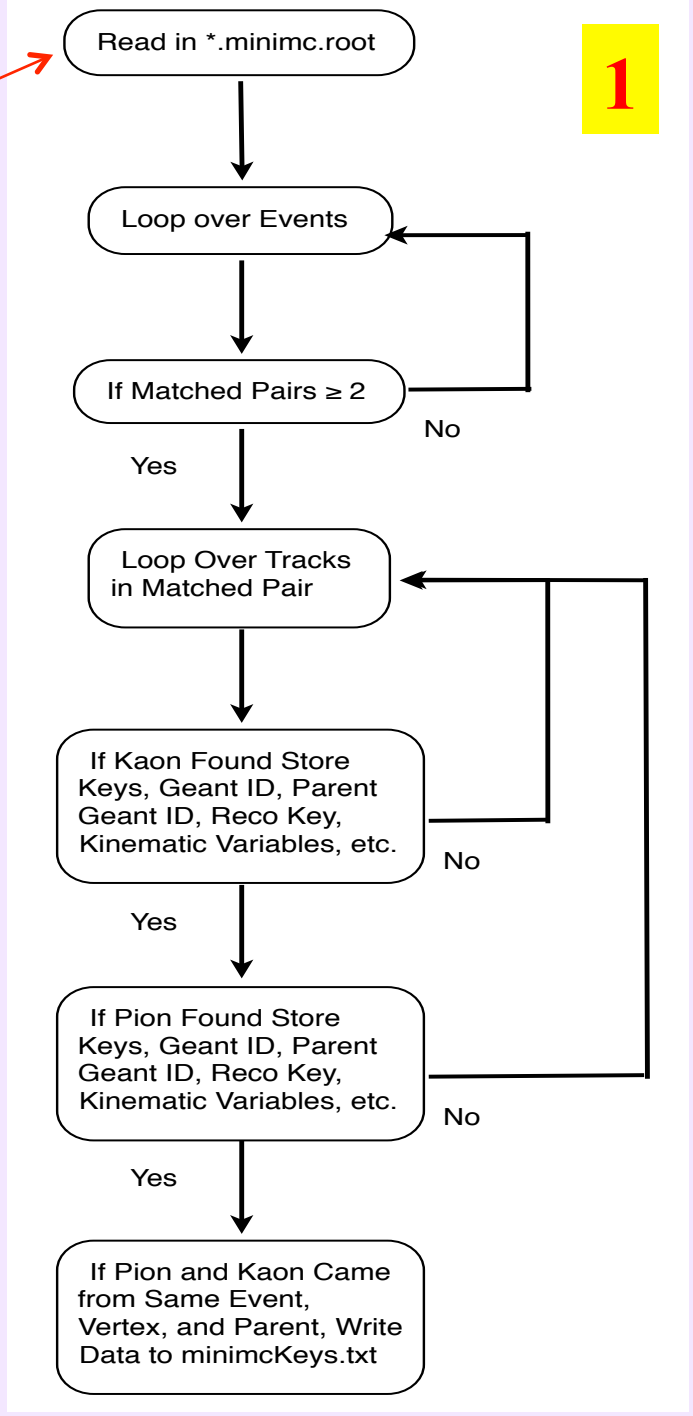
The following are filled for each candidate
PtD0 : transverse momentum of D0 candidate
PD0 : momentum of D0 candidate
MassD0 : mass of D0 candidate
EtaD0 : pseudo-rapidity of D0 candidate
RapD0 : rapidity of D0 candidate
AziD0 : azimuthal angle of d0 candidate
PtKaon : transverse momentum of kaon daughter
PtPion : transverse momentum of pion daughter
PKaon : momentum of kaon daughter
PPion : momentum of pion daughter
ChargeKaon : charge of kaon daughter
ChargePion : charge of pion daughter
SiKaon : number of silicon hits for kaon daughter
SiPion : number of silicon hits for pion daughter
dEdxKaon : dEdx of kaon daughter
dEdxPion : dEdx of pion daughter

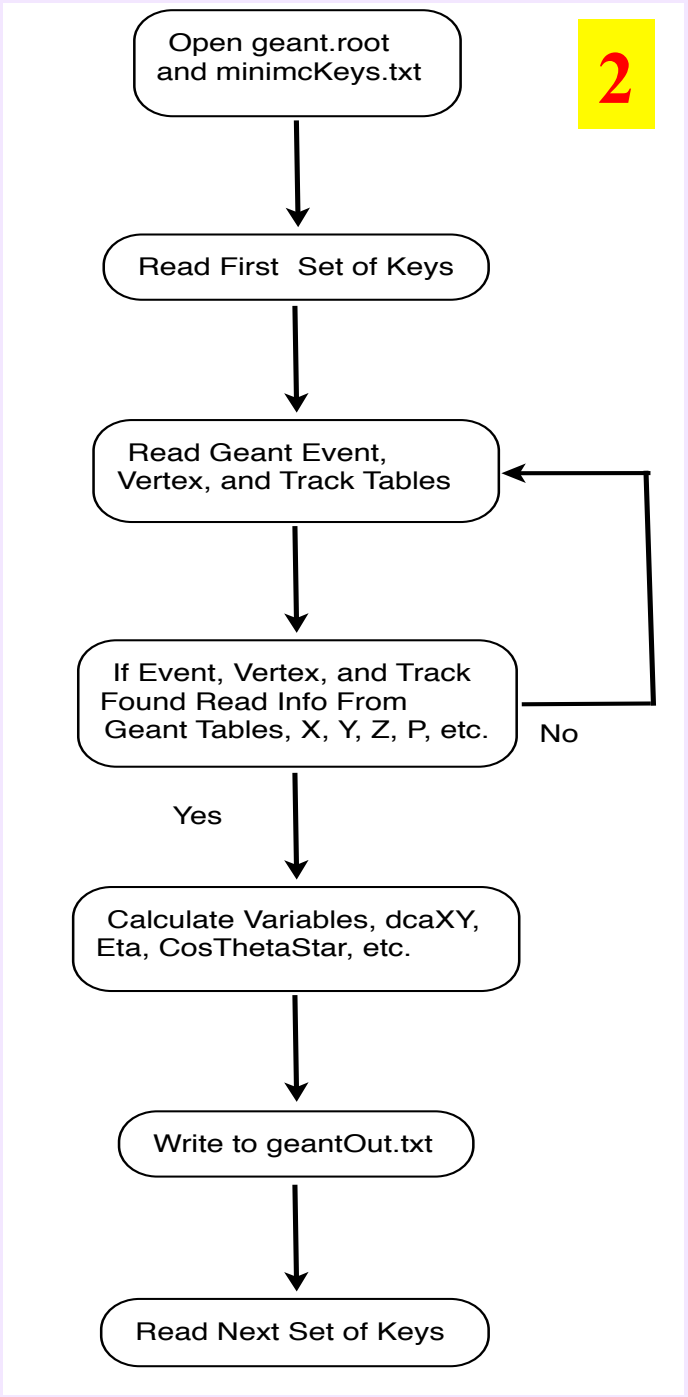
ndEdxKaon : ndEdx of kaon daughter
ndEdxPion : ndEdx of pion daughter
dcaXYKaon : distance of closest approach to primary
dcaXYPion : distance of closest approach to primary v
dcaZKaon : distance of closest approach to primary ve
dcaZPion : distance of closest approach to primary ve
PhiKaon : azimuthal angle of kaon daughter
PhiPion : azimuthal angle of pion daughter
SigmaDcaXYKaon : error of DCA in transverse direc
SigmaDcaXYPion :error of DCA in transverse directi
DcaTrackTXY : distance between daughter tracks at t
DcaTrackTZ : distance between daughter tracks at the
slength : signed decay length
dslength : error of signed decay length
probability : probability of fit
CosPointing : angle beteew the line joining the primary vertex and se
thetaGJ : angle of kaon daughter with the D0 momentum in D0 frame
kLen0 : signed length of kaon daughter to secondary vertex (from TC
kLen1 : signed length of pion daughter to secondary vertex (from TC
dkLen0 : error of signed length of kaon daughter to secondary vertex
dkLen1 : error of signed length of pion daughter to secondary vertex

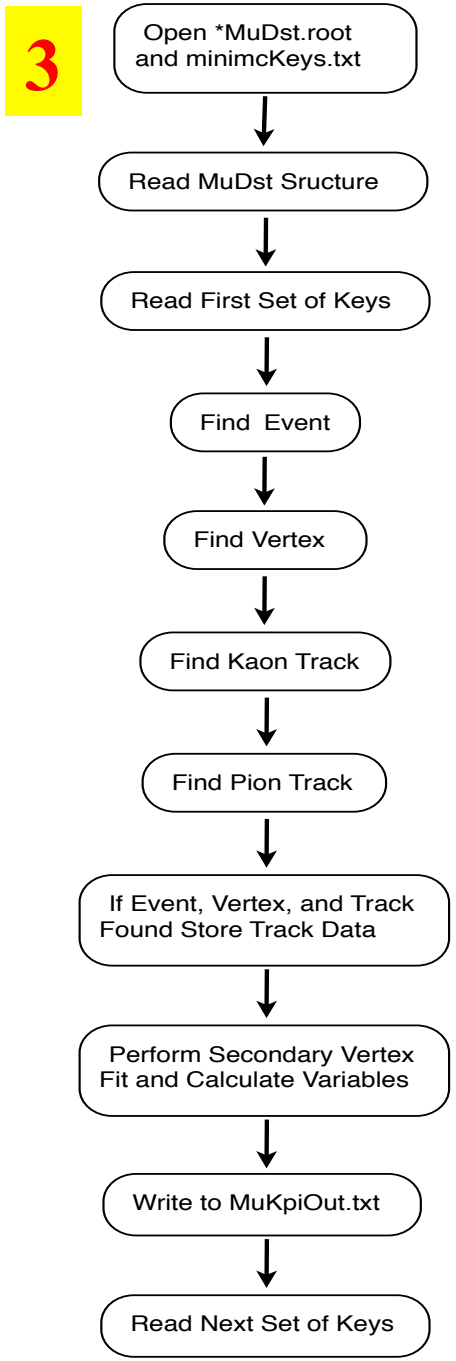
Output of user/fit code

Evaluation chain






2
Open geant.root
and minimcKeys.txt
Read First Set of Keys
Read Geant Event,
Vertex, and Track Tables
If Event, Vertex, and Track
Found Read Info From
Geant Tables, X, Y, Z, P, etc.
No
Yes
Calculate Variables, dcaXY,
Eta, CosThetaStar, etc.
Write to geantOut.txt
Read Next Set of Keys


3
Open *MuDst.root
and minimcKeys.txt
Read MuDst Sructure
Read First Set of Keys
Find Event
Find Vertex
Find Kaon Track
Find Pion Track
If Event, Vertex, and Track
Found Store Track Data
Perform Secondary Vertex
Fit and Calculate Variables
Write to MuKpiOut.txt
Read Next Set of Keys

4

Open GeantOut.txt and MuKpiOut.txt

Read Geant Data into Array

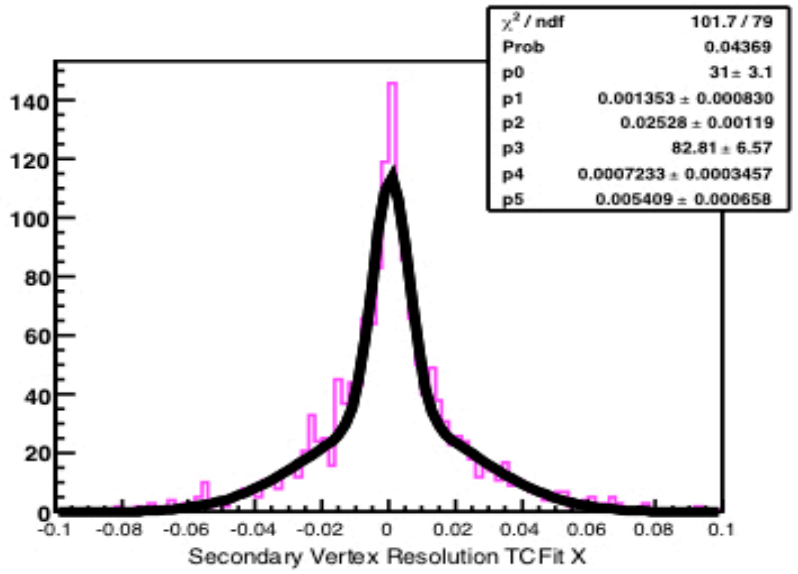
Read MuKpi Data into Array

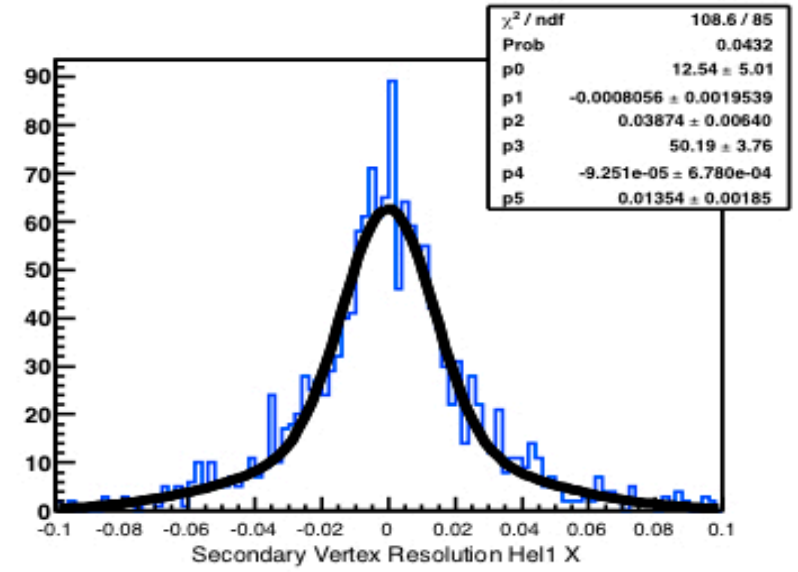
Loop Over Geant Data

Loop Over MuKpi Data

If Same Event, Vertex, and Tracks Found, Write Both Sets of Variables to Tree for

No





Xrec-Xmc

Summary

- Streamlining the Evaluation/Analysis structure is a big advantage
- It can be simple augmentation of current structures
- It can be more (replace V0-finder with Kalman in BFC)
- Need to prepare a comprehensive proposal of modifications/streamlining Evaluation/Analysis:
  - In consultation with Software-Infra group
  - Circulate in group for feedback
  - Yifei, Jan, Jonathan and I can start on this