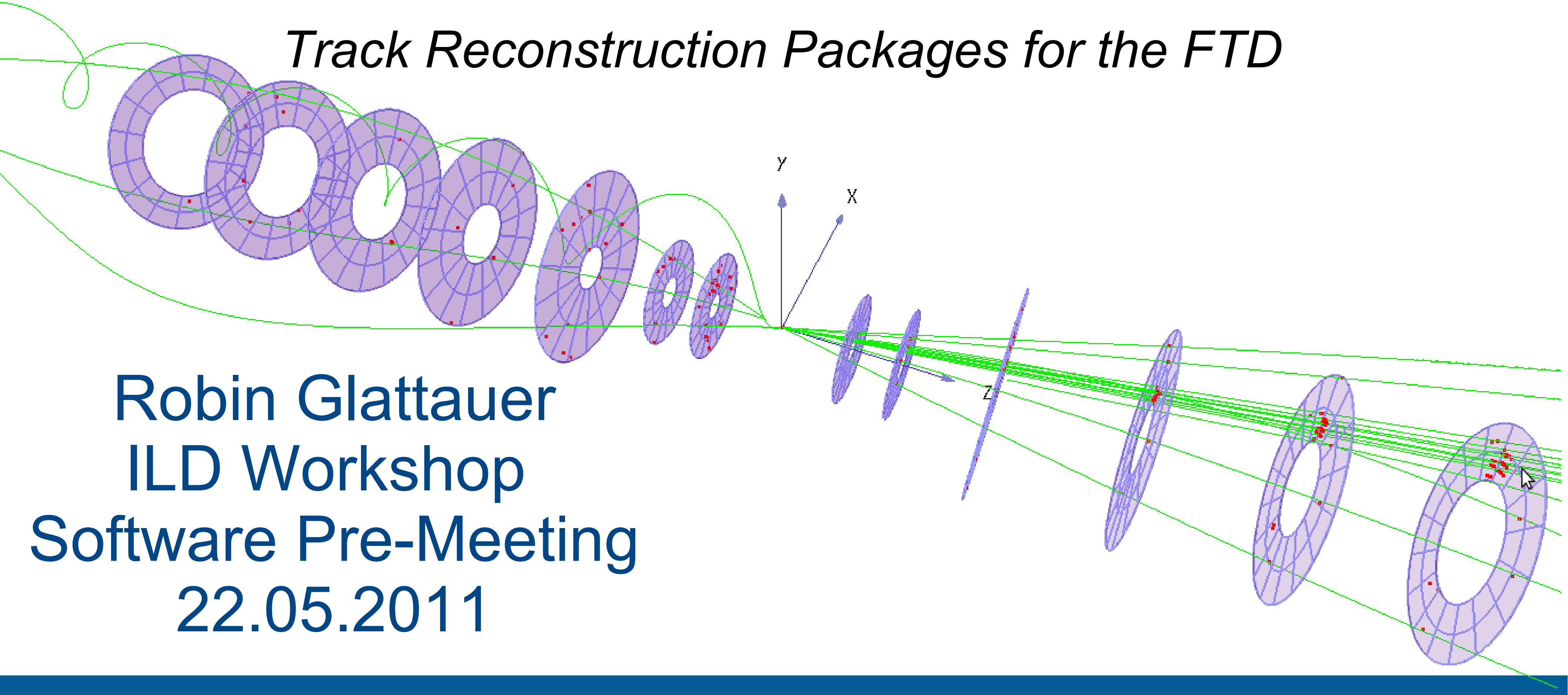
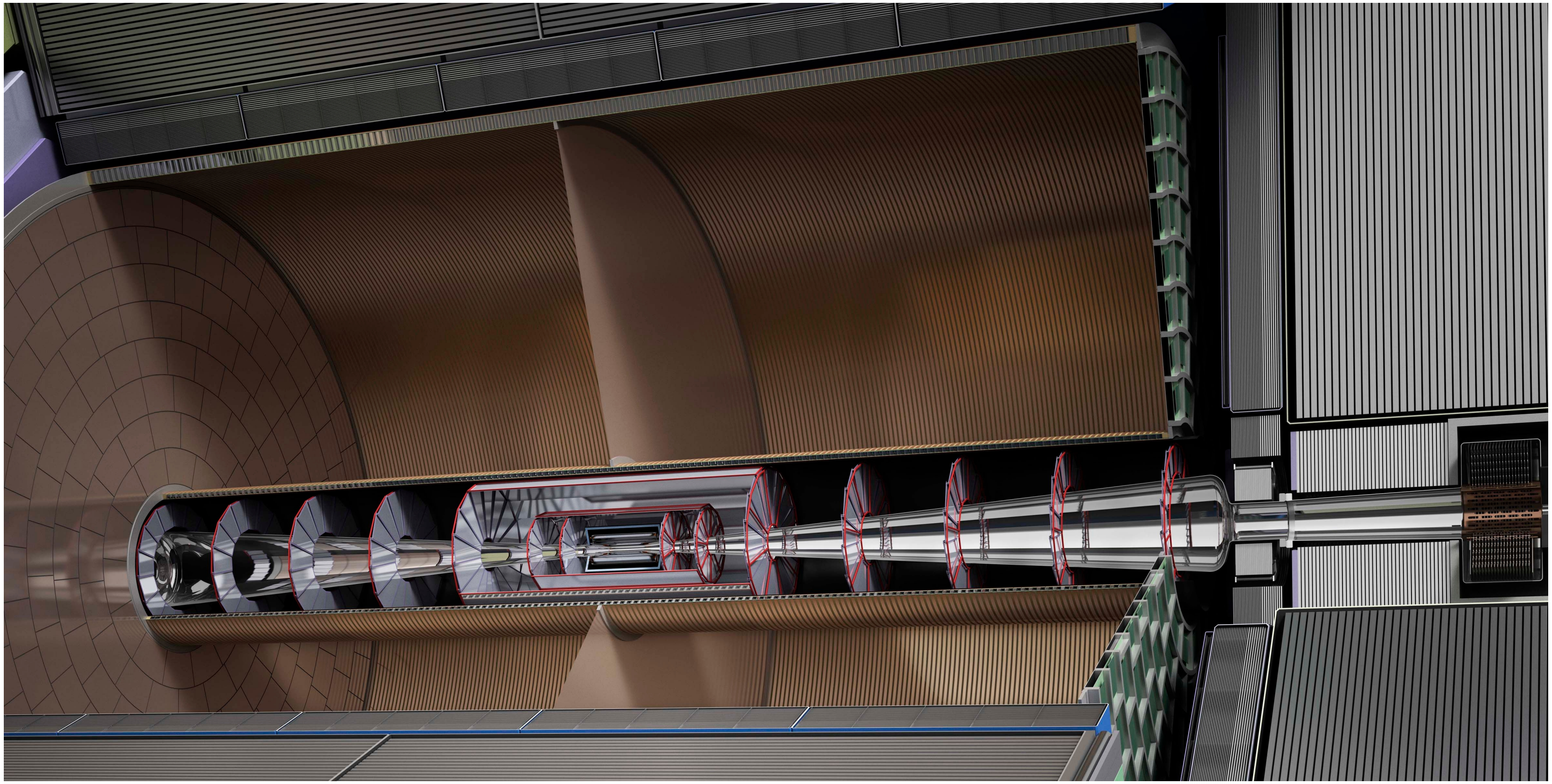
KiTrack and ForwardTracking

Track Reconstruction Packages for the FTD



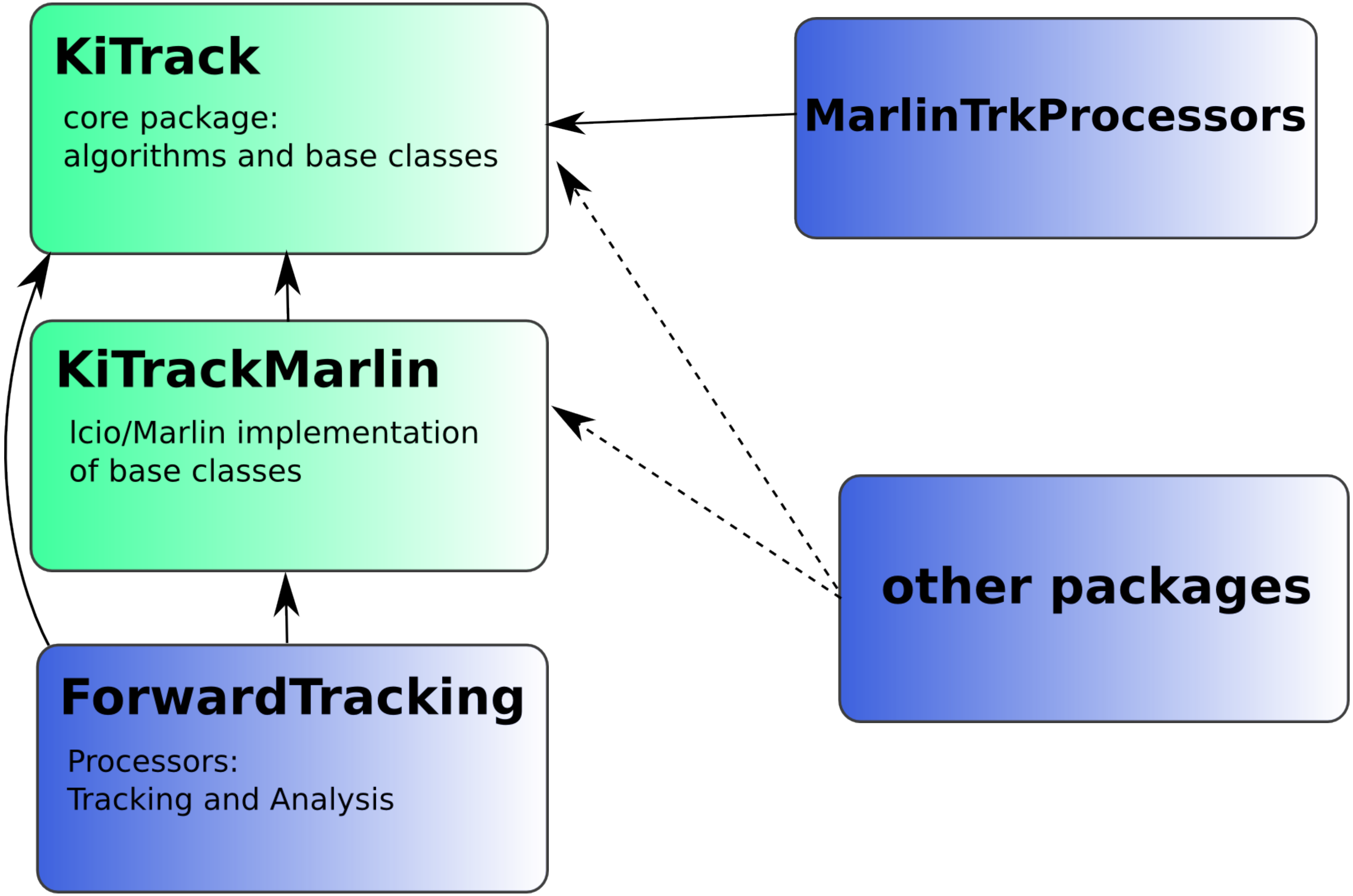
Robin Glattauer
ILD Workshop
Software Pre-Meeting
22.05.2011

The Forward Region



FTD 0,1 : pixel detector
FTD 2-6 : dual layer strip detector

The Packages



KiTrack

core package:
algorithms and base classes

MarlinTrkProcessors

KiTrackMarlin

lcio/Marlin implementation
of base classes

other packages

ForwardTracking

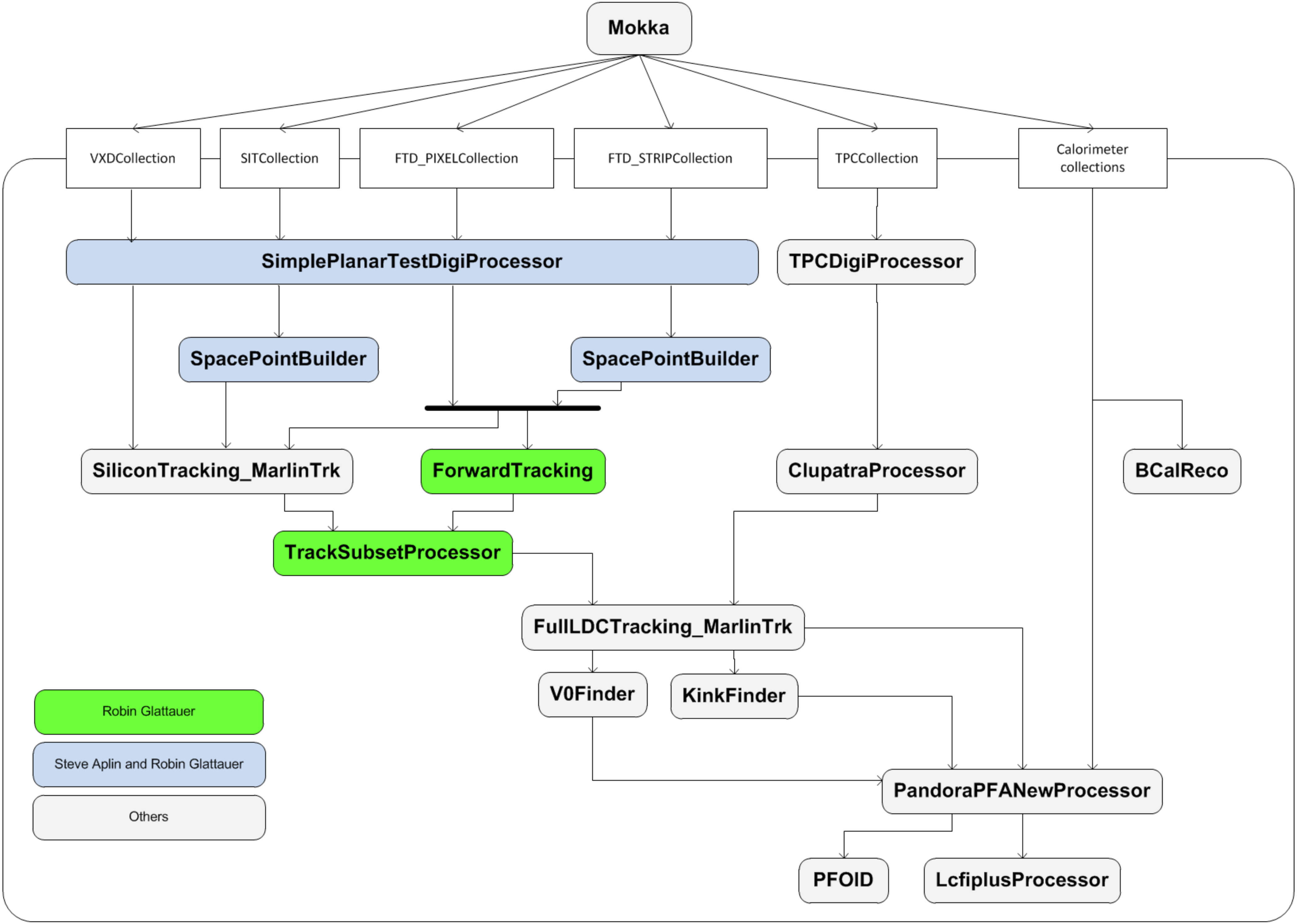
Processors:
Tracking and Analysis

ForwardTracking

- Depends on KiTrack and KiTrackMarlin
- Reconstructs tracks through the FTD
- For all tracks with
  - p_T > 100 MeV
  - 4 hits or more in FTD (needed for fitting)
- Parallel to SiliconTracking
- Tracks combined by TrackSubsetProcessor

TrackSubsetProcessor

- Situation: tracks reconstructed by multiple packages with different algorithms
- Which ones to take? Not all are compatible
- TrackSubsetProcessor creates one track collection with completely compatible tracks
- Aim: Maximize quality of tracks
- Uses the Hopfield Neural Network from KiTrack



Mokka

VXDCollection

SITCollection

FTD_PIXELCollection

FTD_STRIPCollection

TPCCollection

Calorimeter collections

SimplePlanarTestDigiProcessor

TPCDigiProcessor

SpacePointBuilder

SpacePointBuilder

SiliconTracking_MarlinTrk

ForwardTracking

ClupatraProcessor

BCalReco

TrackSubsetProcessor

FullLDCTracking_MarlinTrk

V0Finder

KinkFinder

PandoraPFANewProcessor

PFOID

LcfiplusProcessor

🟩 Robin Glattauer

🟦 Steve Aplin and Robin Glattauer

⬜ Others

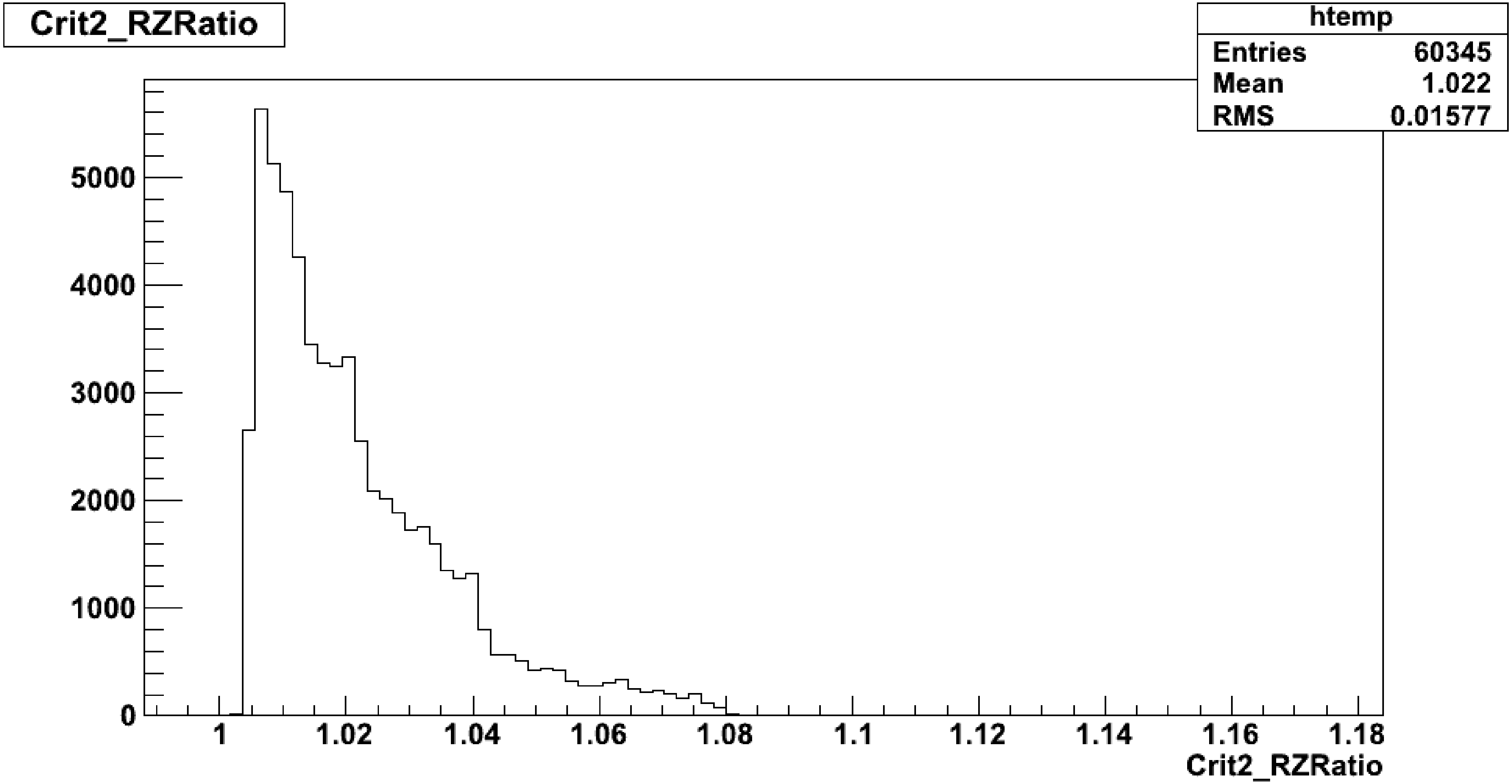
KiTrack + ForwardTracking: The Main Methods

- Cellular Automaton: find tracks
- Kalman Filter: fit tracks, gain quality indicator and sort out
- Hopfield Neural Network: find a compatible subset
- Alternative methods possible for every step

Cellular Automaton

- Use as much information as possible
- As early as possible:
  - When could 2,3,4 hits belong to a true track?
  - Start with 2 hits and sort out in every step
  - Do the segments form a possible track, i.e. are they connected? (e.g. to the IP)
  - Only when this is finished use time consuming methods like the Kalman Filter

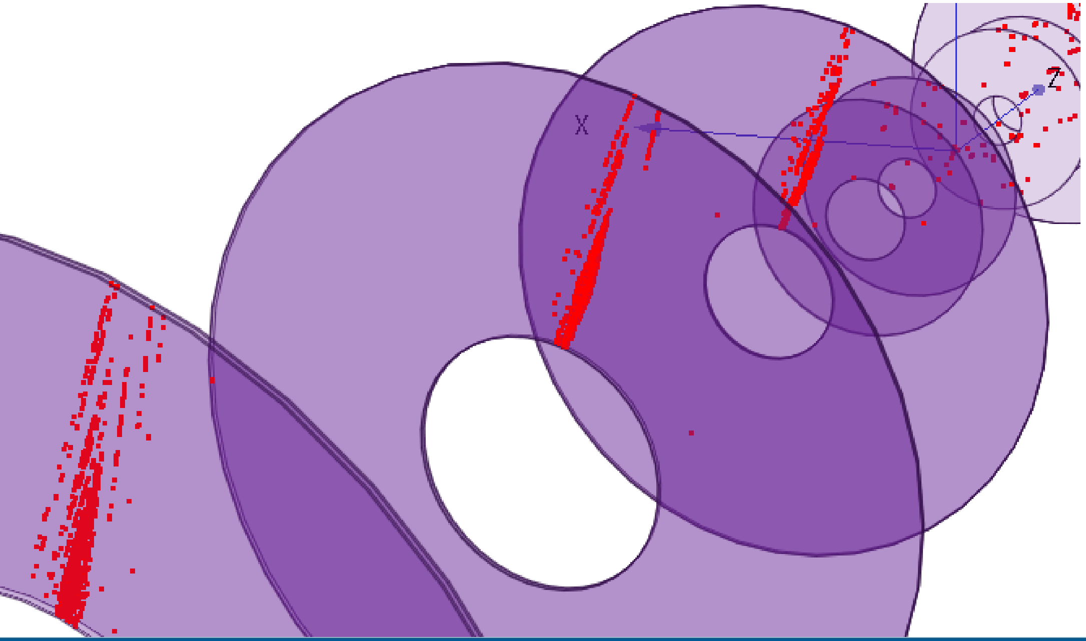
Cellular Automaton: example for a criterion to apply on two hits



Cellular Automaton: Recent Developments

- Deal with “overdose” of hits: rerun Automaton with different parameters → tighten the cuts

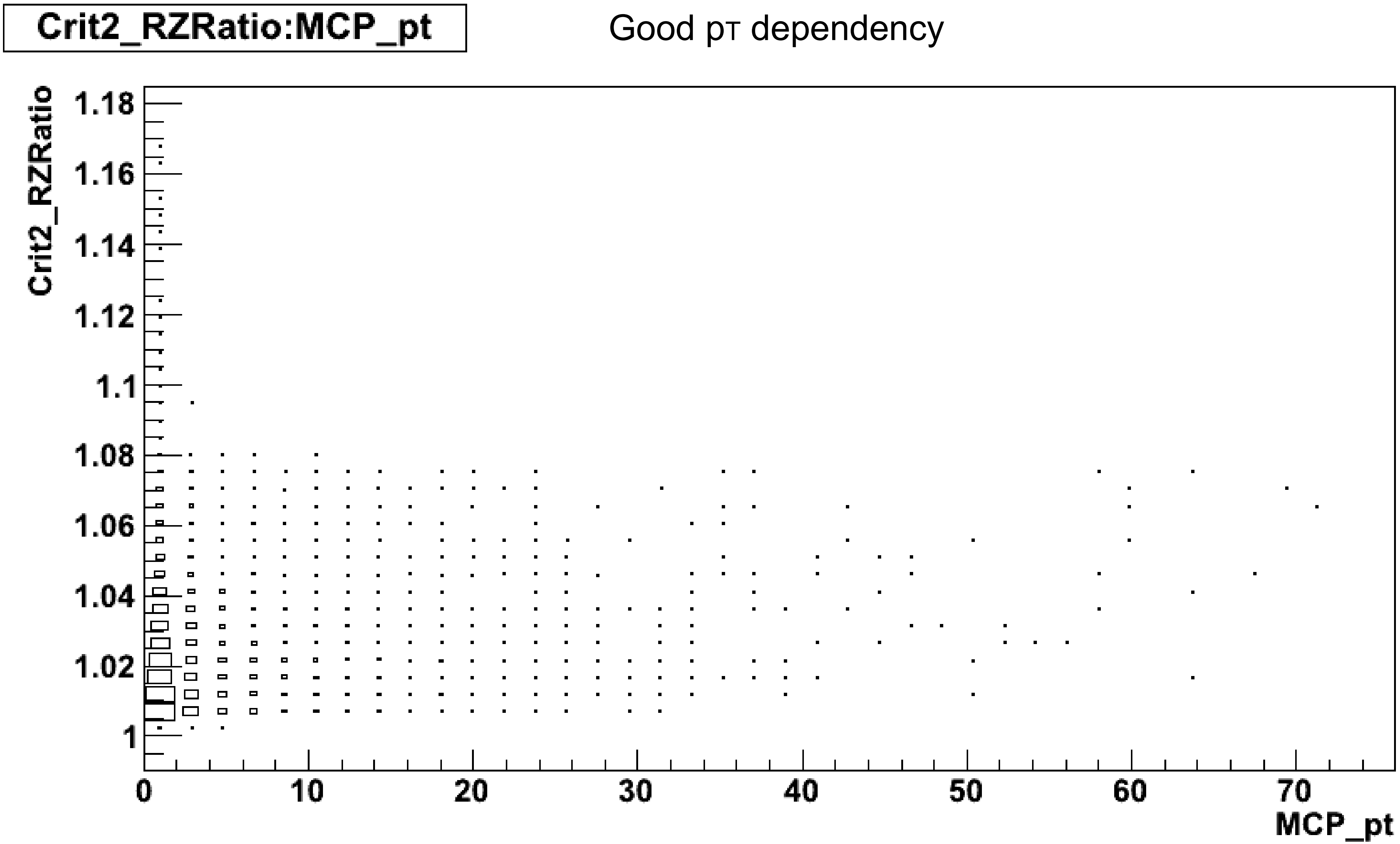
(not yet committed)

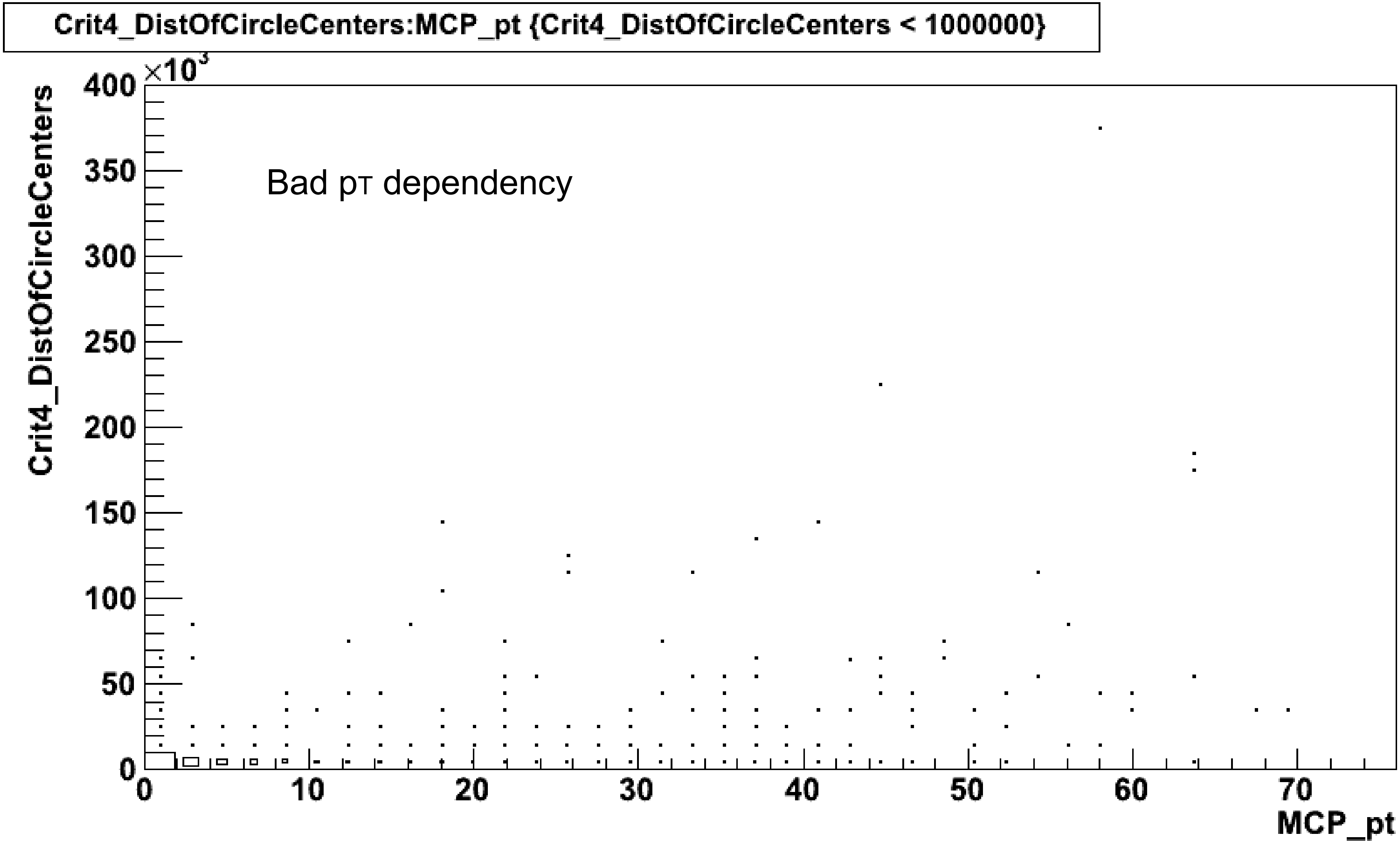


Cellular Automaton: Recent Developments

- Taking care of p_T dependencies

Good pT dependency





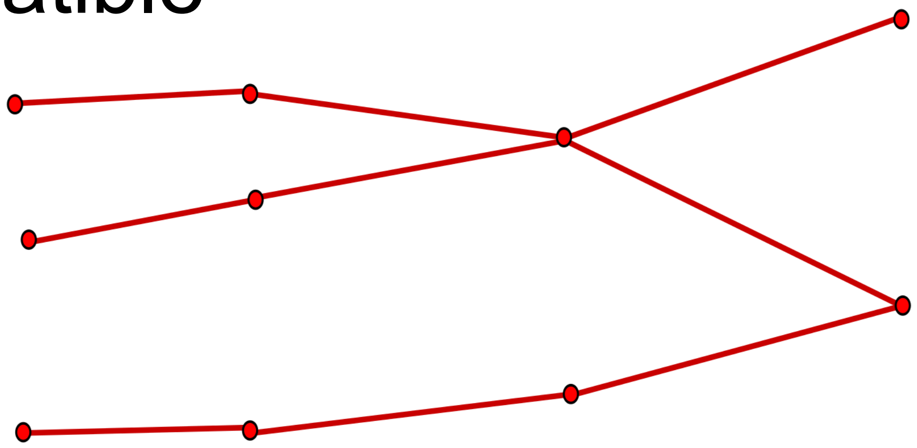
Bad pT dependency

Kalman Filter

- KalTest + KalDet + MarlinTrk
- Use χ² probability as quality indicator
- Make a cut at 0.005
- Faster to use additional prefilter like helix fit,
- Needs more investigation (effect on efficiency)

Hopfield Neural Network

- Goal: find compatible subset
- Hit sharing tracks mainly from combinatorial background → incompatible



- Now a template class

Results

- See talk on Thursday
- Promising, but needs more fine tuning
- Efficiency: good, but:
  - Drop in efficiency for high p_T needs resolving
  - Intermediate region

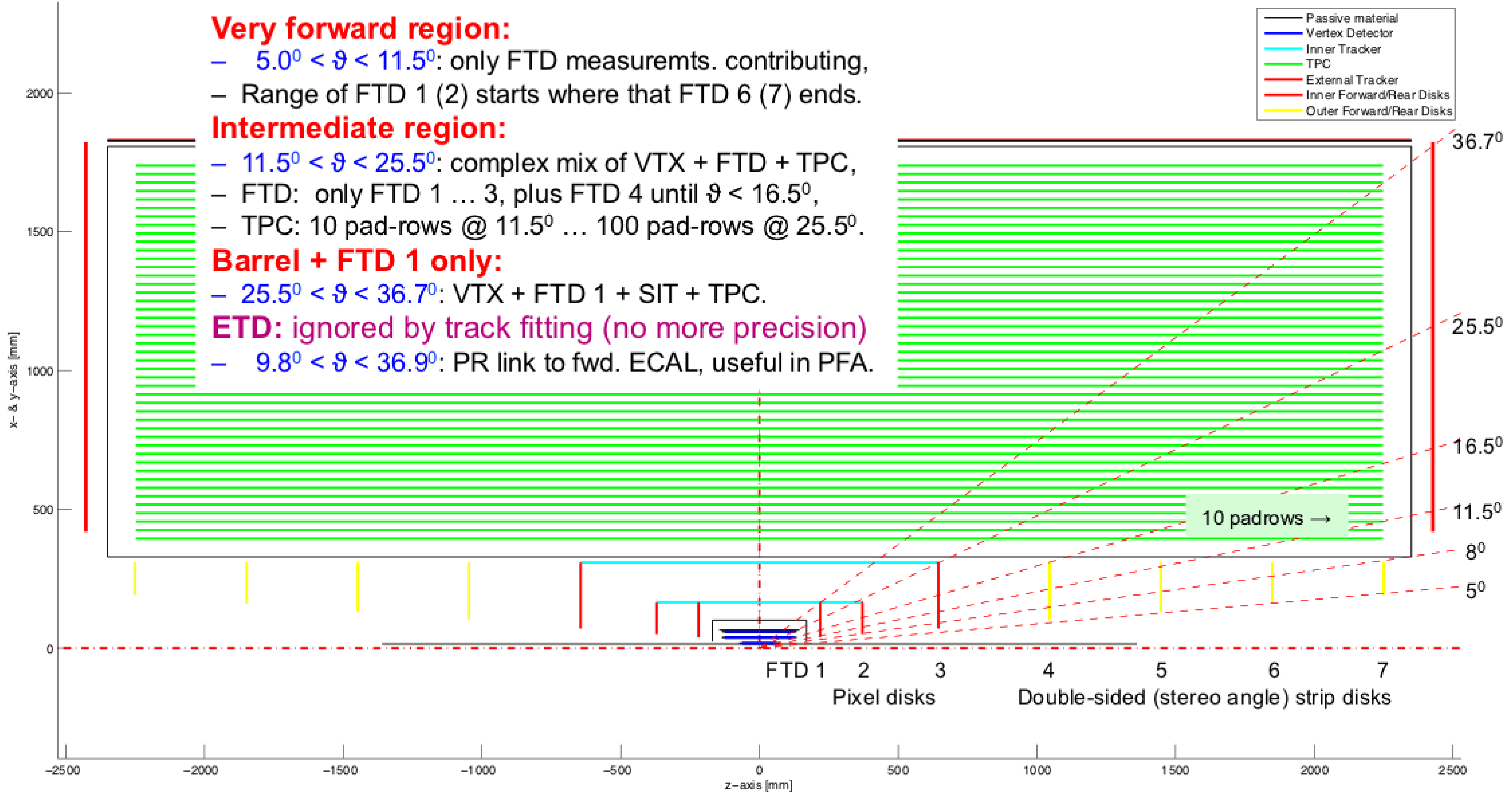
Possible Further Improvements

- More sophisticated steering
- Alternative and additional algorithms at each stage
- Flexible acting

Thank you!

Back Up

Regions



Very forward region:

- $5.0^0 < \vartheta < 11.5^0$: only FTD measuremts. contributing,
- Range of FTD 1 (2) starts where that FTD 6 (7) ends.

Intermediate region:

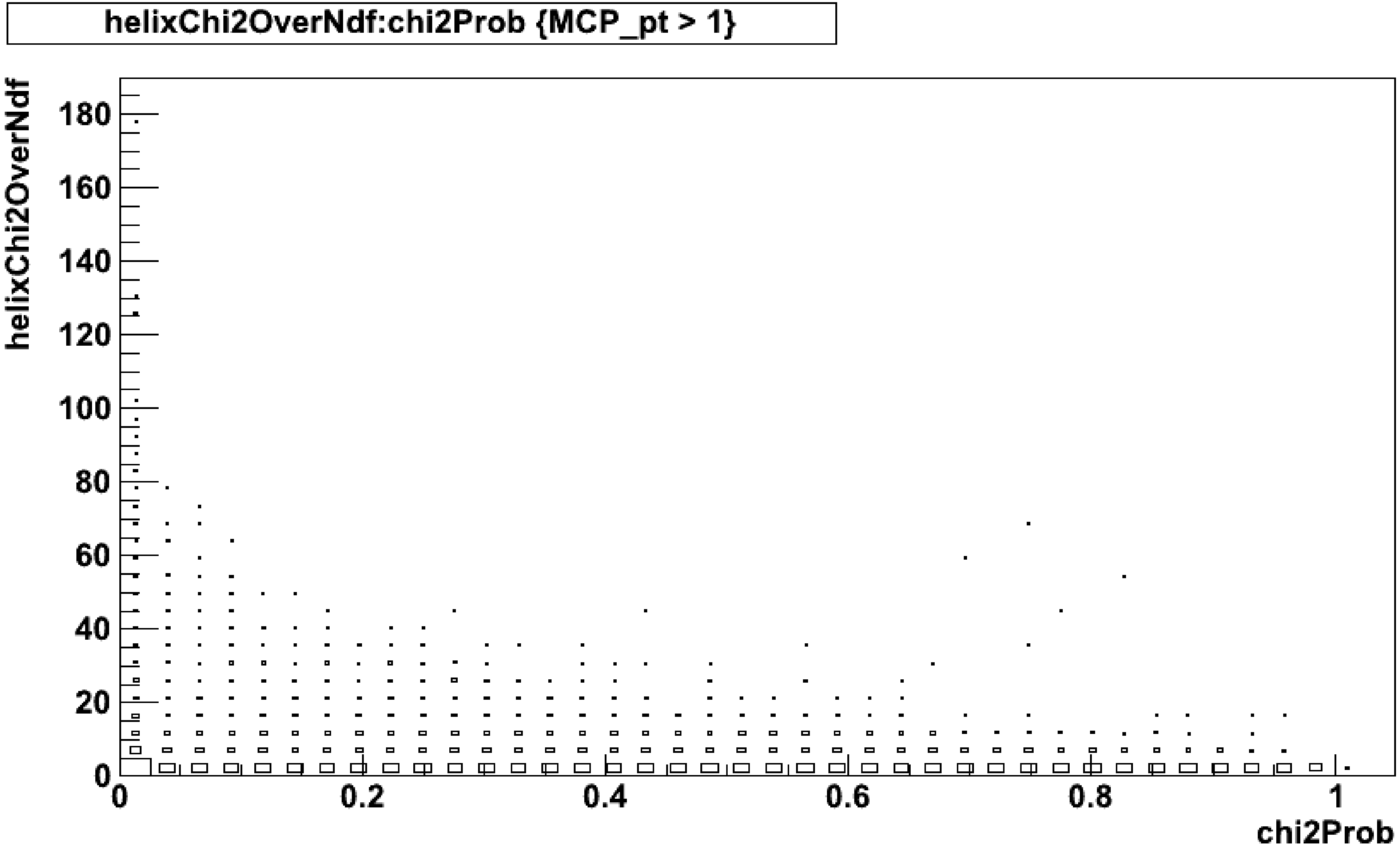
- $11.5^0 < \vartheta < 25.5^0$: complex mix of VTX + FTD + TPC,
- FTD: only FTD 1 … 3, plus FTD 4 until $\vartheta < 16.5^0$,
- TPC: 10 pad-rows @ 11.5^0 … 100 pad-rows @ 25.5^0.

Barrel + FTD 1 only:

- $25.5^0 < \vartheta < 36.7^0$: VTX + FTD 1 + SIT + TPC.

ETD: ignored by track fitting (no more precision)

- $9.8^0 < \vartheta < 36.9^0$: PR link to fwd. ECAL, useful in PFA.


helixChi2OverNdf:chi2Prob {MCP_pt > 1}
helixChi2OverNdf
0
20
40
60
80
100
120
140
160
180
0
0.2
0.4
0.6
0.8
1
chi2Prob