TestBeam Event Reconstruction Software

M. Bomben[1], N. Neri[2], J. Walsh[2]

[1]Universita' & INFN Trieste

[2]Universita' & INFN Pisa

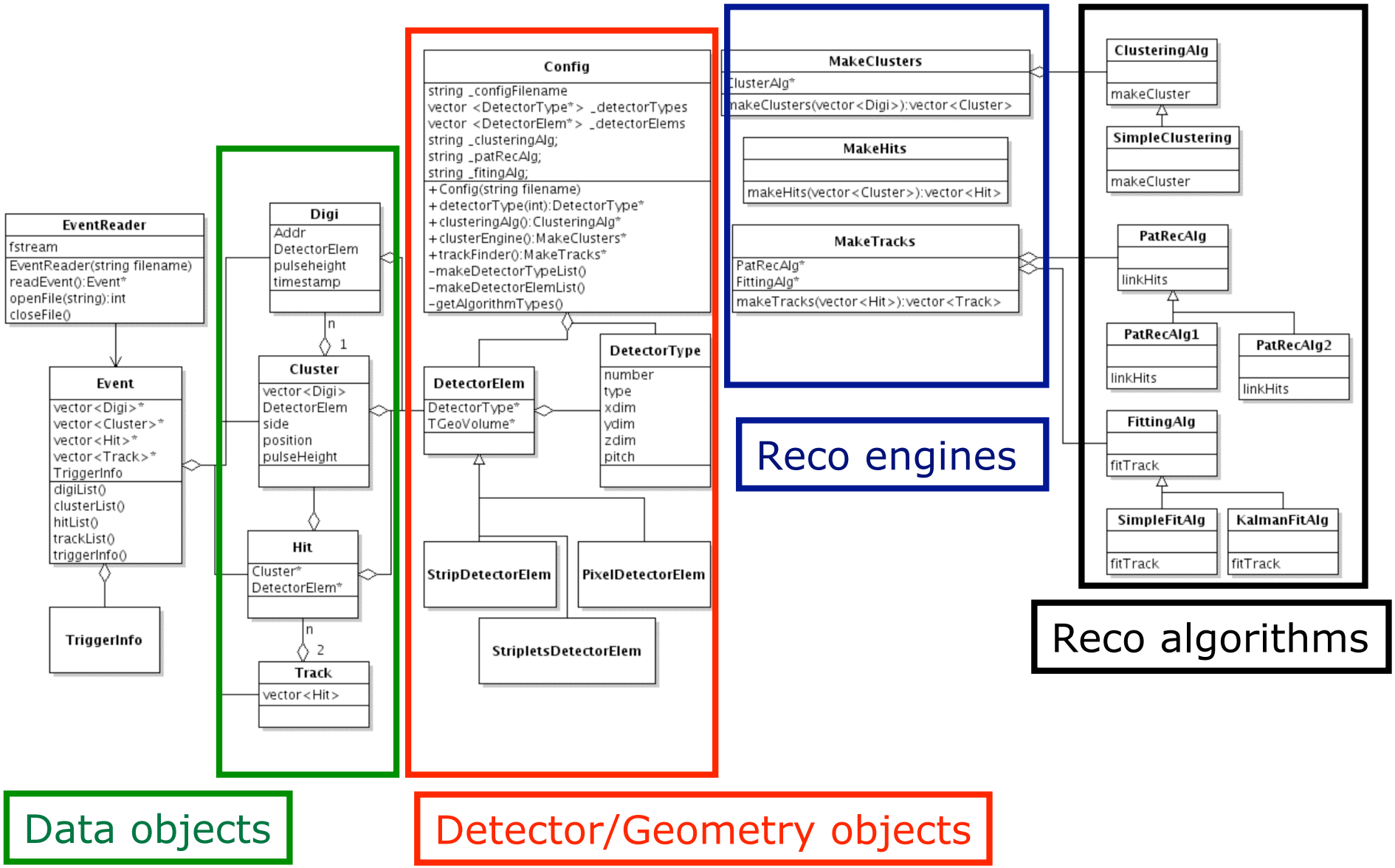
SuperB Meeting

Isola d'Elba 31 May 2008

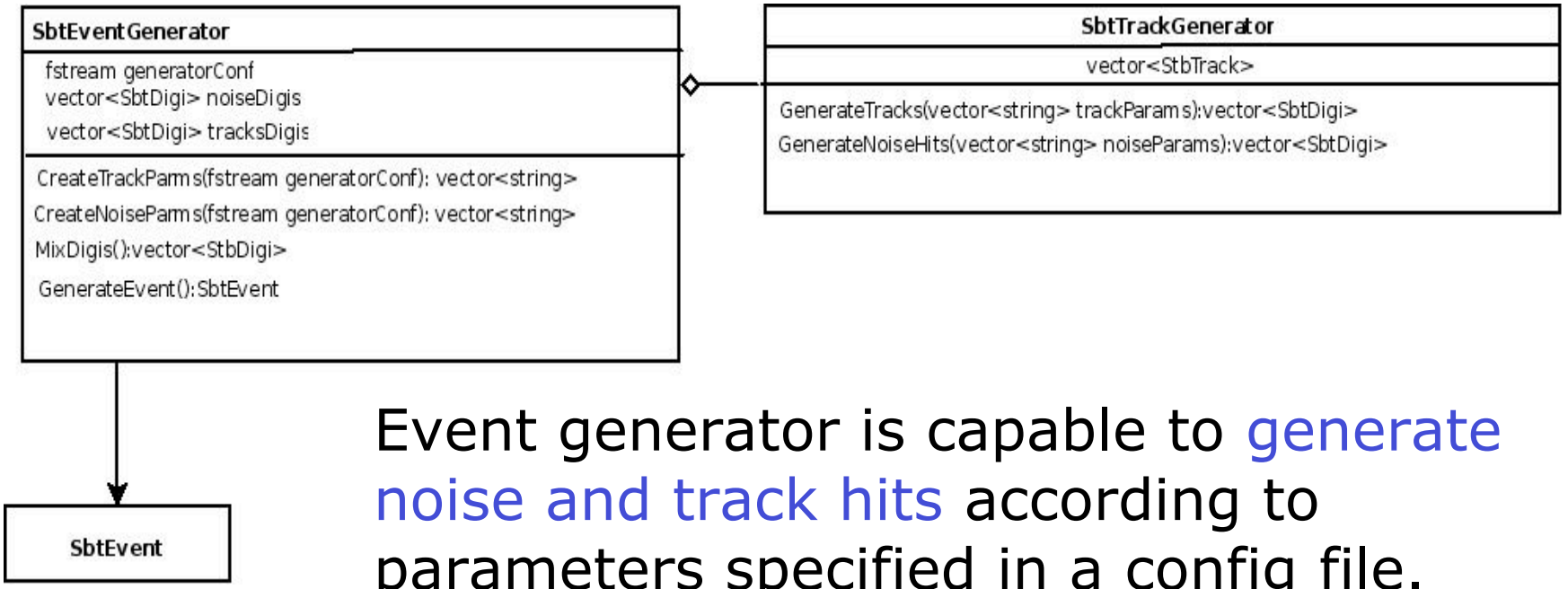
Software Design

- Try to balance good design with simplicity
- This is a small project, so it doesn't need a complex design:
- Use java program Violet for creating UML class diagrams: http://www.horstmann.com/violet/
- However, let's try to design code that is:
  - flexible
    - don't fix number of layers or different types of detectors, e.g.
  - re-usable
    - might be necessary for Test Beam '09, '10, etc.
  - easy to read and understand

Reco class diagrams



Event Generator design



Event generator is capable to generate noise and track hits according to parameters specified in a config file.

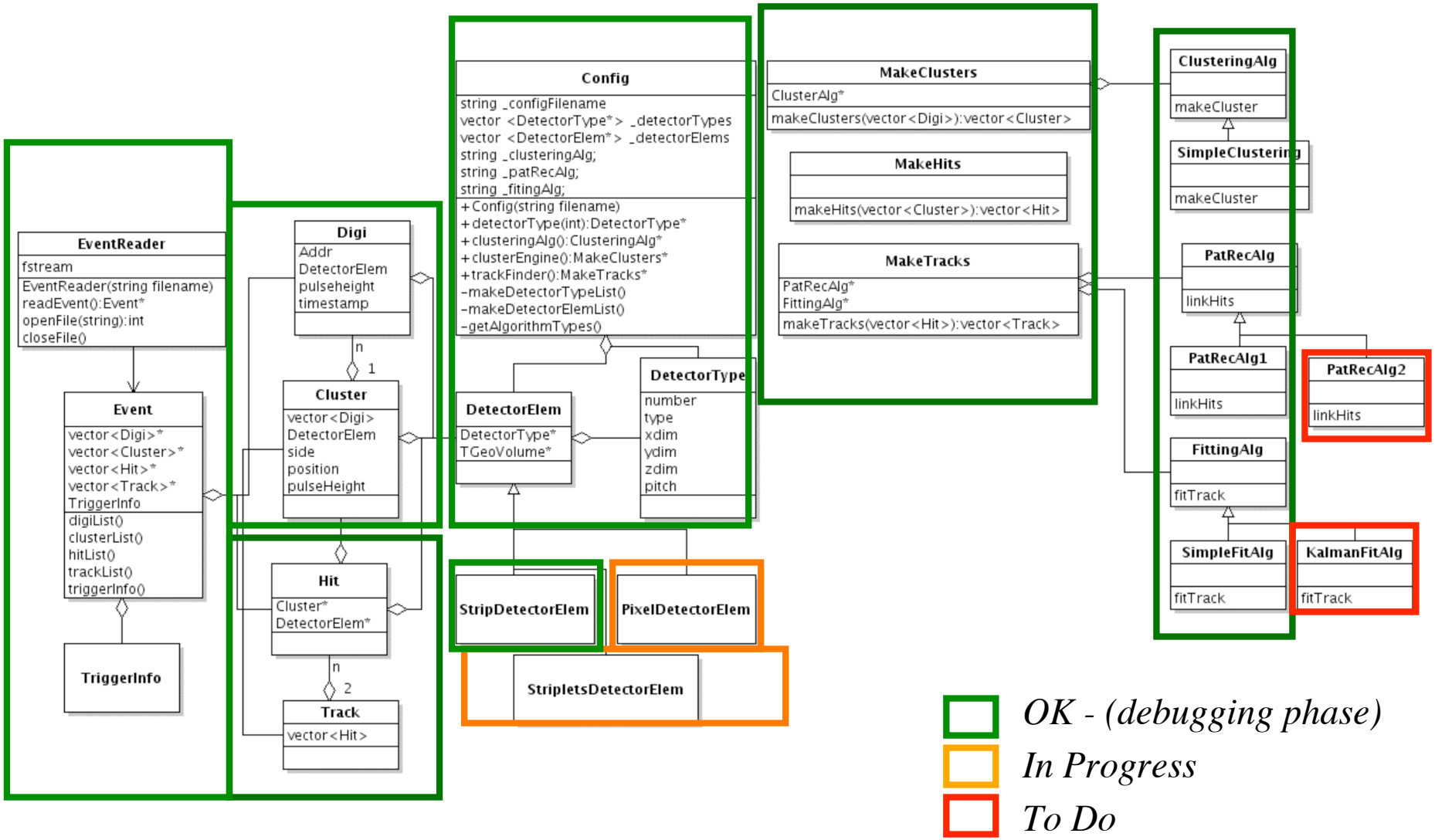
Output has identical format to DAQ files for event reconstruction.

Necessary tool for Associative Memories trigger database training.

Code status

- Code is available on CVS repository
- Use ROOT libraries to build detector geometry
- Software design is finished
- Implementation nearing completion
- Debugging of individual classes has started
- Debugging of complete reconstruction chain has started

Implementation status



Pending issues

- Associative memories trigger:
  - Generate patterns for trigger database training. Straight forward.
- Alignment:
  - we have ideas how to include in present design, to be done
- Support for MAPS:
  - FEE Calibrations:
    - provide ASCII files vs thr. - from DAQ files - in order to allow the calibration debugging
  - Radioactive sources test:
    - Same as for FEE calibrations

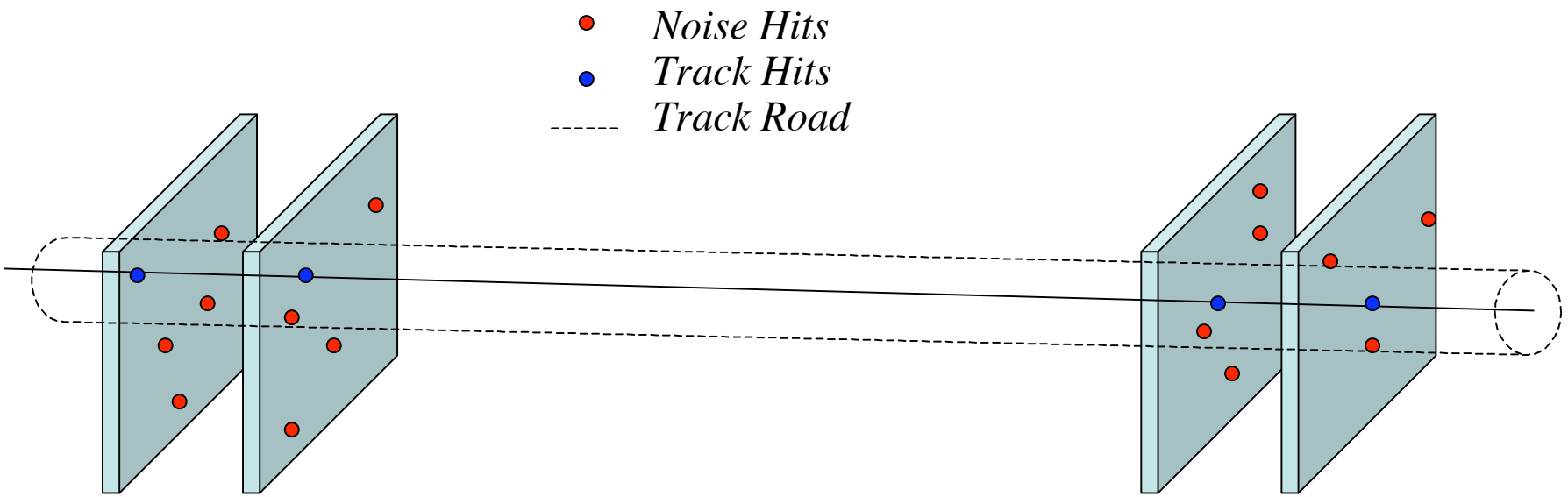
Preliminary studies for PatRec and Fit Algorithm

- Caveats:
  - –ToyMC study
  - –we haven't considered eliminating noise hits using pulse height information
  - –we've assumed no timing info within a single event is available

These could potentially improve performance

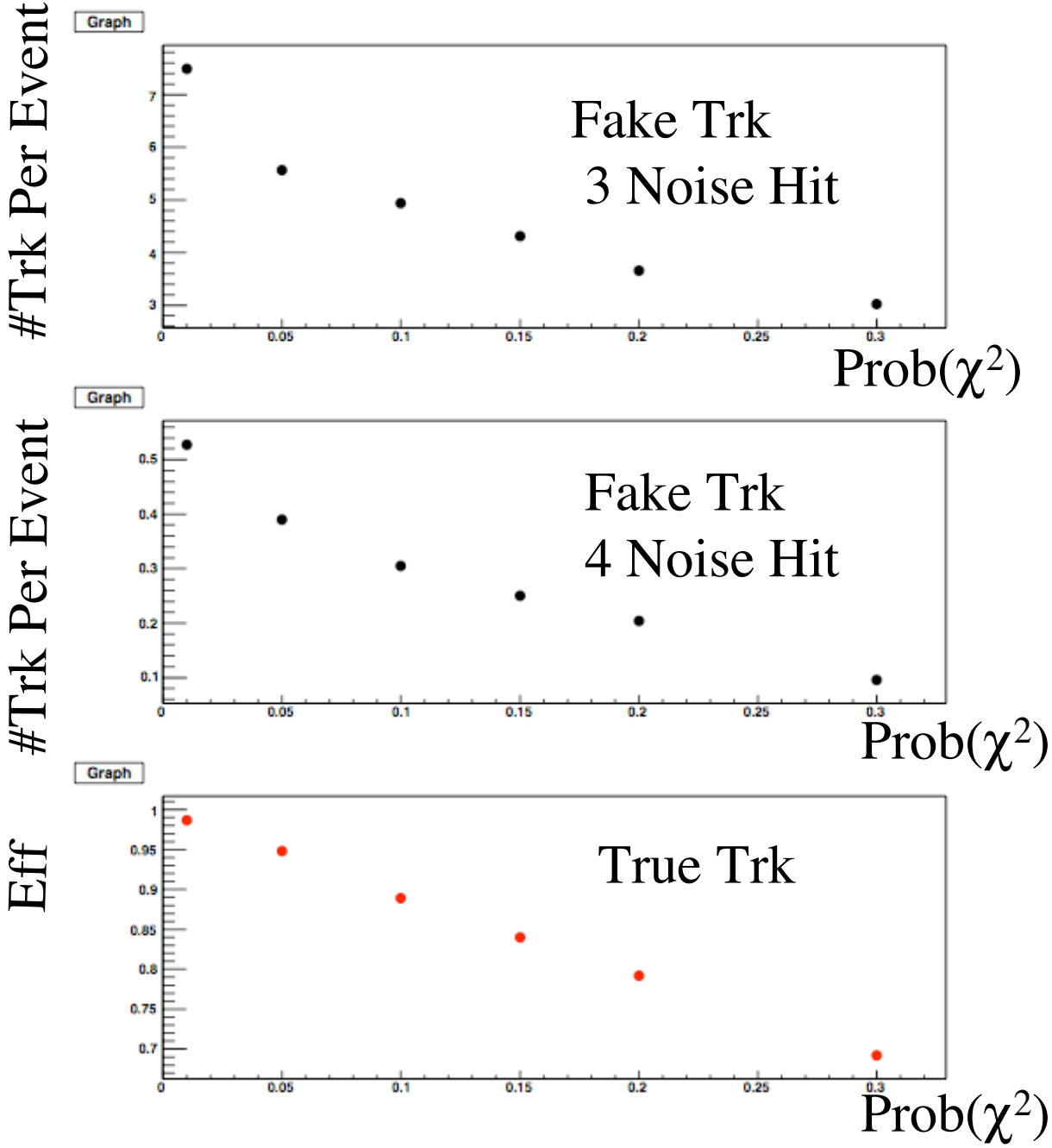
The following estimates should be considered as worst case scenario.

Simulation Program



Generated Track Hits (Multiple Scattering taking into account) and Noise Hits (Uniform distribution in the detector) to evaluate the performance of Reco Software as a function of the detector occupancy.

Detector Occupancy 3%



Fake tracks per event after requiring Prob(χ^2) cuts

Prob(χ^2)>0.01

Occ (%)	3Hit Fake Trk	4Hit Fake Trk
1	0.15	0.002
3	7.5	0.6
5	38	4

Prob(χ^2)>0.10

Occ (%)	3Hit Fake Trk	4Hit Fake Trk
1	0.10	0.001
3	5	0.3
5	25	2.5

Prob(χ^2)>0.30

Occ (%)	3Hit Fake Trk	4Hit Fake Trk
1	0.06	0.0002
3	3	0.1
5	15	1

Efficiency for true tracks correspond to 1-Prob(χ^2)

(Tentative) Conclusions

- Assume that number of fake tracks should be <<1 per event
- Can work with up to 3% occupancy, cutting hard on χ^2 (which is ok)
- Can possibly work with just 3 layers (if a layer dies, for example) if occupancy $\leq$ 1%
- Caveat reminder:
  - we haven't considered eliminating noise hits using pulse height information
  - we've assumed no timing info within a single event is available

Summary

- Reconstruction code has been implemented
- Debugging process underway
- Alignment algorithm is next big issue
- Software byproducts:
  - Associative memories trigger database training
  - ROOT/ASCII files for MAPS FEE calibrations