e/g and pflow merged electrons

David Sabes
LLR-Ecole polytechnique

Introduction

- **GSFelectron collection merging e/g and pflow electrons**
  - **Motivations**
  - **Algorithms**
  - **Datasets**

- **Results & investigation**
  - **Effect of pflow merging on efficiency**
  - **Efficiency found by each method**
  - **Fake rate**
  - **Characteristic of Trackerdriven only electrons**

- **Conclusion**

Motivations

- **E/g e⁻ reconstruction developed with the isolated case in mind**
  - **Large supercluster energy collection area**
  - **although not assuming isolated electrons**

It ensures coherent approach with HLT and calibration from data (Zee)

It has been optimized for efficient reconstruction down to pT=5GeV/c

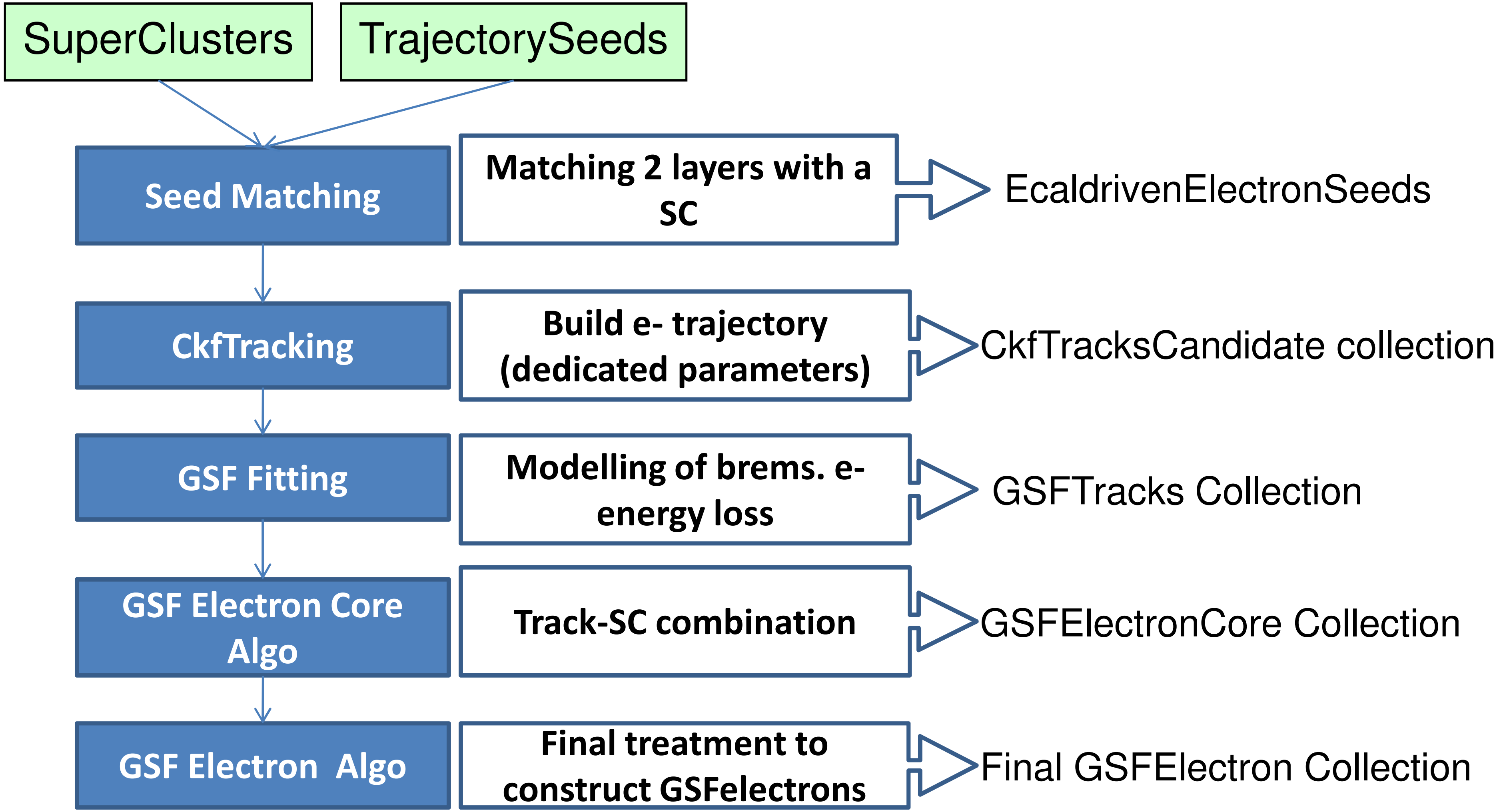
  - **Most demanding channel H->ZZ->4e used a benchmark for efficiency**

- **Pflow reconstruction has been developed with non isolated case in mind**
  - **electron in b-jets, low pT**
  - **therefore a specific clustering algorithm**
  - **and a tracker driven approach, starting from efficient tracking**

It is also efficient on isolated electrons

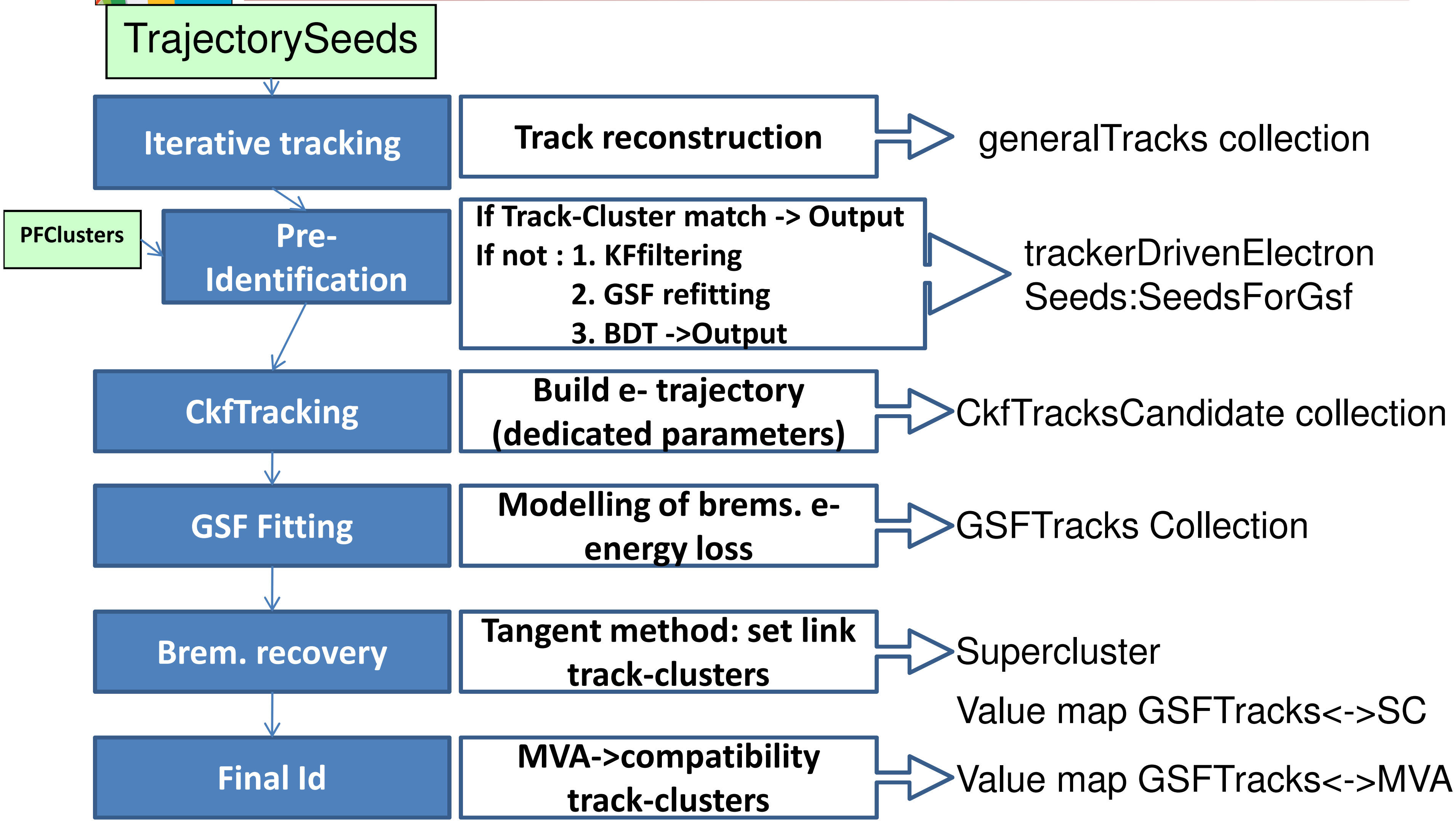
- **Aim at providing all reconstructed electrons in a coherent way**
  - **benefit from the two approached (combined efficiency)**
  - **single electron collection**

Ecaldriven method



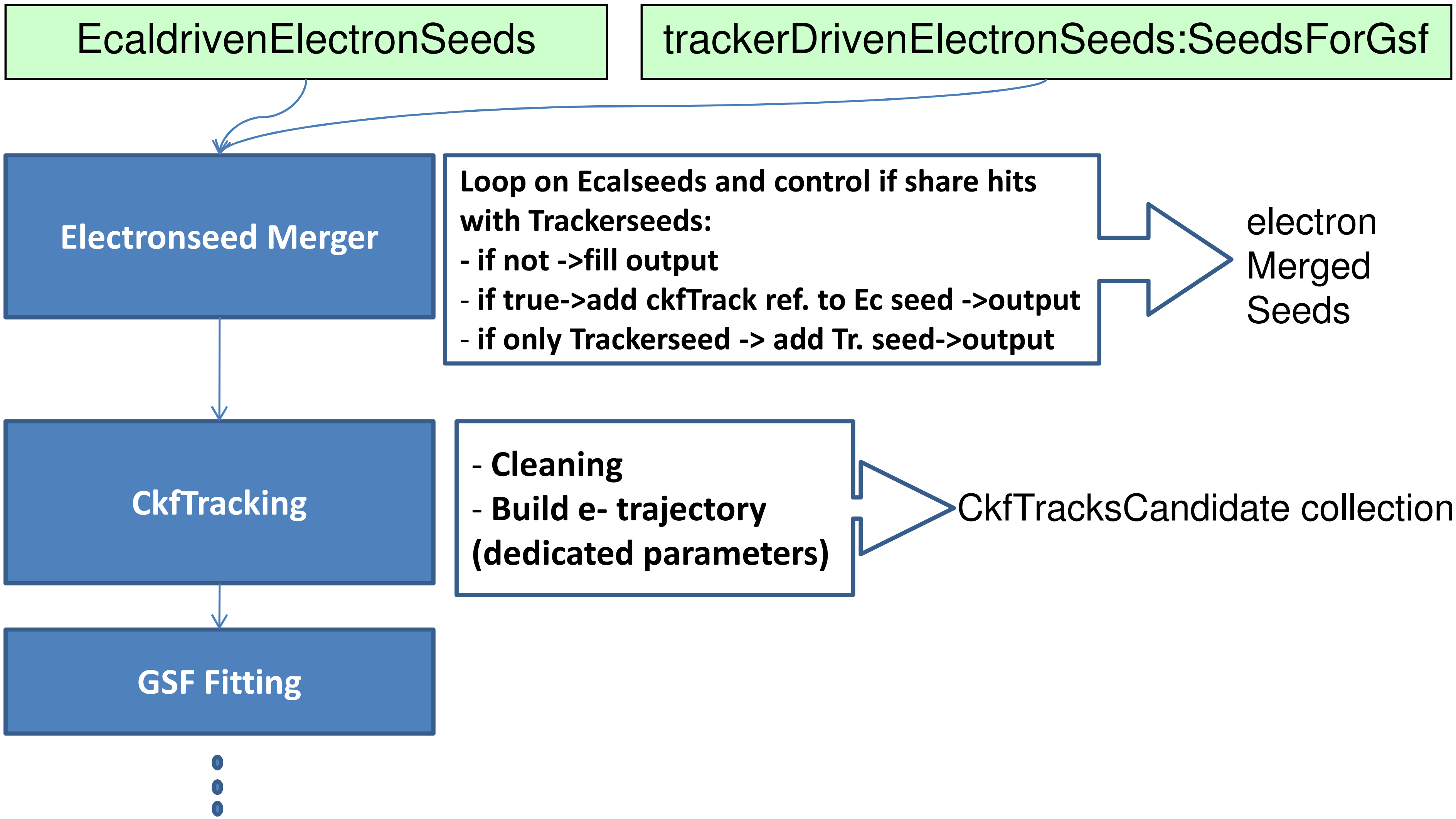
- This method is initiated by the SC reconstruction

Trackerdriven method



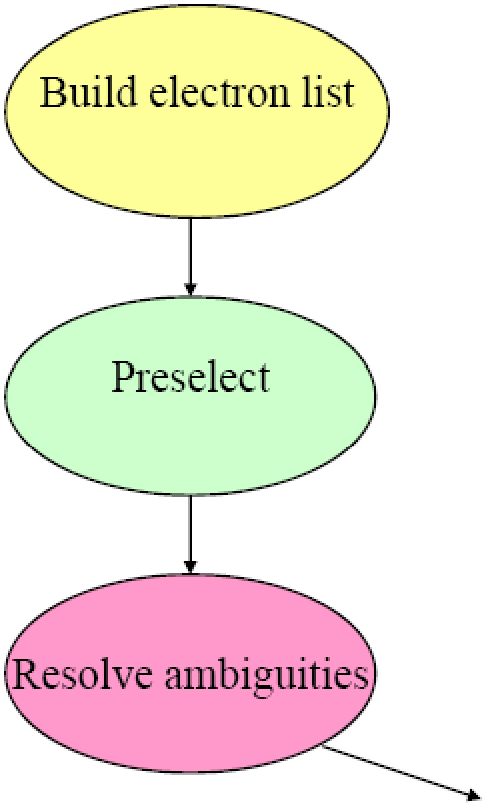
- This method is initiated by the reconstruction of general tracks

Merging



EcaldrivenElectronSeeds
trackerDrivenElectronSeeds:SeedsForGsf
Electronseed Merger
Loop on Ecalseeds and control if share hits with Trackerseeds:
- if not ->fill output
- if true->add ckfTrack ref. to Ec seed ->output
- if only Trackerseed -> add Tr. seed->output
electron Merged Seeds
CkfTracking
- Cleaning
- Build e- trajectory
(dedicated parameters)
CkfTracksCandidate collection
GSF Fitting

Integration: GsfElectronAlgo



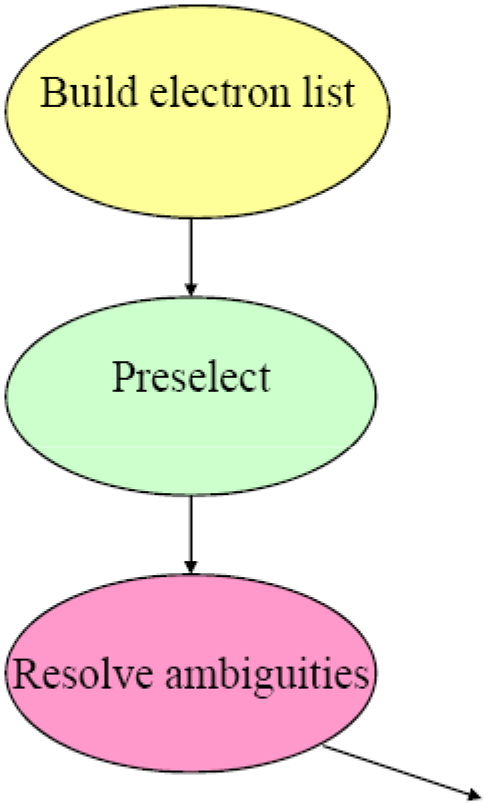
- loop on ElectronCores
- if ecalDriven, compute all electron quantities from e/g SC
    - add ref. to pflow SC and MVA when available
- if !ecalDriven, compute electron quantities from pflow SC

- if ecalDriven, apply e/g preselection
- if !ecalDriven, currently no preselection

- curently using pflow SC in case electron is trackerDriven only

Final electron list

Integration: GsfElectronAlgo



Final electron list

- loop on ElectronCores
- if ecalDriven, compute all electron quantities from e/g SC
  - add ref. to pflow SC and MVA when available
- if !ecalDriven, compute electron quantities from pflow SC

- if ecalDriven, apply e/g preselection
- if !ecalDriven, currently ~~no~~ preselection

In CMSSW_3_1_pre6

- curently using pflow SC in case electron is trackerDriven only

Software

- SW version : **CMSSW_3_1_0_pre4**

- Tags list :
  - ✓ V06-02-04 CondFormats/DataRecord
  - ✓ V01-00-12 CondFormats/EcalObjects
  - ✓ V01-13-17-01 Configuration/StandardSequences
  - ✓ V02-04-00 DataFormats/CaloRecHit
  - ✓ V03-02-03 DataFormats/EgammaCandidates
  - ✓ V00-10-06 DataFormats/EgammaReco
  - ✓ V11-00-00 DataFormats/ParticleFlowCandidate
  - ✓ V12-00-00 DataFormats/ParticleFlowReco
  - ✓ V00-01-15 RecoEcal/Configuration
  - ✓ V00-06-04 RecoEcal/EgammaClusterAlgos
  - ✓ V00-06-33 RecoEcal/EgammaClusterProducers
  - ✓ V00-05-28 RecoEcal/EgammaCoreTools
  - ✓ V00-04-00 RecoEgamma/Configuration
  - ✓ V01-03-00 RecoEgamma/EgammaElectronAlgos
  - ✓ V01-02-01 RecoEgamma/EgammaElectronProducers

- Tags list :
  - ✓ V06-02-04 CondFormats/DataRecord
  - ✓ V00-03-01 RecoEgamma/EgammaIsolationAlgos
  - ✓ V00-03-01 RecoEgamma/EgammaTools
  - ✓ V00-02-03 RecoEgamma/ElectronIdentification
  - ✓ V01-04-01 RecoEgamma/Examples
  - ✓ V10-04-02 RecoParticleFlow/Configuration
  - ✓ V10-04-00 RecoParticleFlow/PFAlgo
  - ✓ V10-02-02 RecoParticleFlow/PFBlockAlgo
  - ✓ V10-03-02 RecoParticleFlow/PFBlockProducer
  - ✓ V11-00-00 RecoParticleFlow/PFClusterAlgo
  - ✓ V11-00-00 RecoParticleFlow/PFClusterProducer
  - ✓ V07-02-00 RecoParticleFlow/PFClusterShapeAlgo
  - ✓ V06-01-03 RecoParticleFlow/PFClusterShapeProducer
  - ✓ V10-02-03 RecoParticleFlow/PFClusterTools
  - ✓ V10-05-03 RecoParticleFlow/PFProducer
  - ✓ V10-03-06 RecoParticleFlow/PFTracking
  - ✓ V02-02-02 TrackingTools/GsfTools
  - ✓ V02-02-00 TrackingTools/GsfTracking

- From slide 17 → SW version : **CMSSW_3_1_0_pre6**

Datasets

Single Electron pt10 & 35 GeV/c :

- relval/CMSSW_3_1_0_pre4/RelValSingleElectronPt10/GEN-SIM-DIGI-RAW-HLTDEBUG/IDEAL_30X_v1
- relval/CMSSW_3_1_0_pre4/RelValSingleElectronPt35/GEN-SIM-DIGI-RAW-HLTDEBUG/IDEAL_30X_v1

Flat Pt 5 to 100 GeV/c :

- relval/CMSSW_3_1_0_pre4/RelValSingleElectronFlatPt5To100/GEN-SIM-DIGI-RAW-HLTDEBUG/IDEAL_30X_v1

QCD Pt 80 to 120 GeV/c :

- relval/CMSSW_3_1_0_pre4/RelValQCD_Pt_80-120/GEN-SIM-DIGI-RAW-HLTDEBUG/IDEAL_30X_v1

Flat Pt 5 to 150 GeV/c : (**CMSSW_3_1_pre6**)

- /SingleElectronFlatPt5to150/sabes-SingleElectronFlatPt5to150-ba4f98800c9984cf40d6ddf4137b010/USER

Provenance: the 2 methods

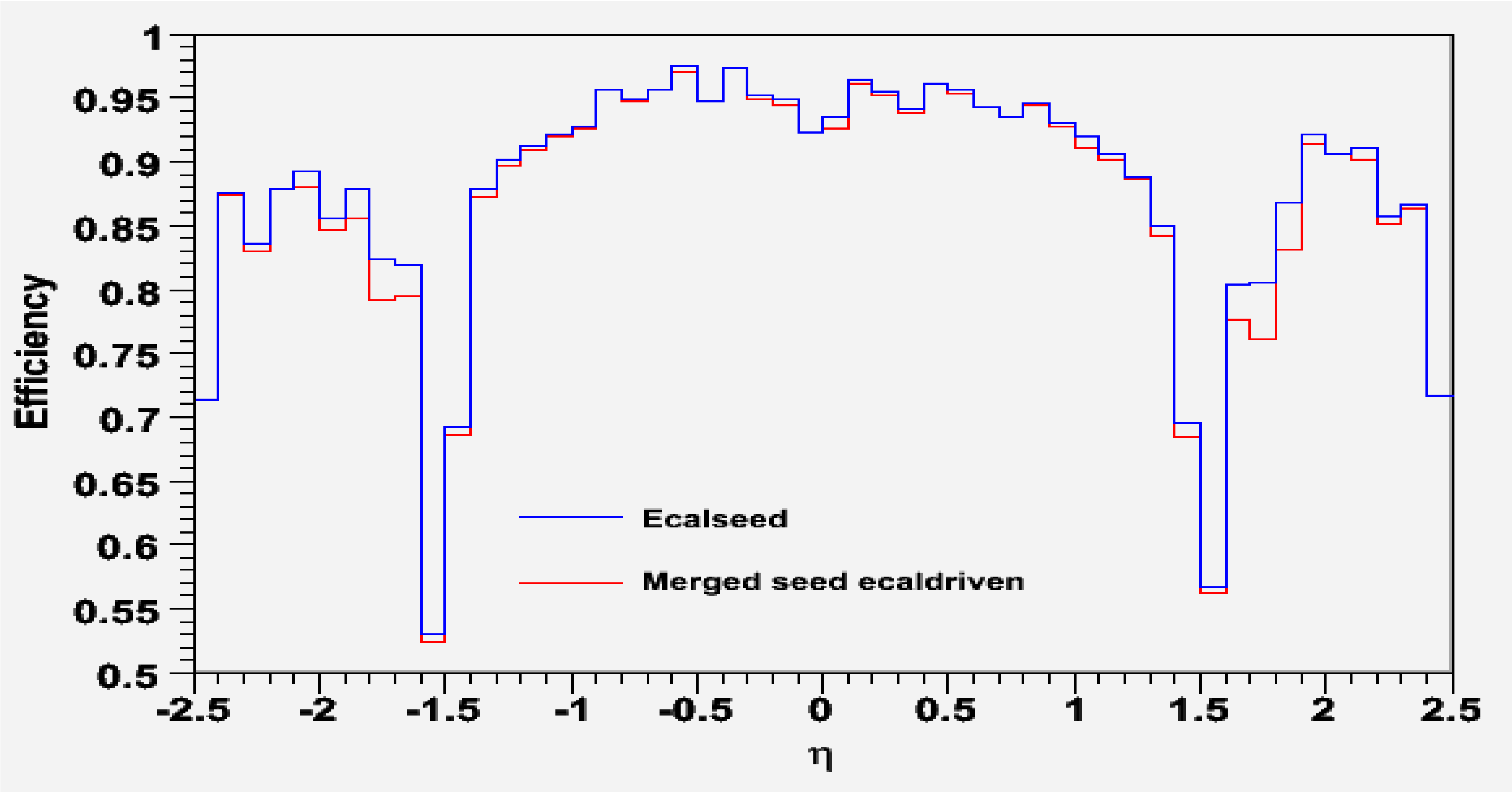
Two ways to determine provenance

- **First method** : considers ElectronSeed references which are
  - Either a Calocluster reference >> ecaldriven (called ***Ecaldriven configuration*** on graphs)
  - Or a CkfTrack reference >> trackerdriven
  - Or both (called ***Standard configuration*** on graphs)

- **Second method**: do not apply merging. Act on CkftrackcandidateMaker input seeds collection
  - ecalDrivenElectronSeeds (called ***Ecaldriven seed*** on graphs)

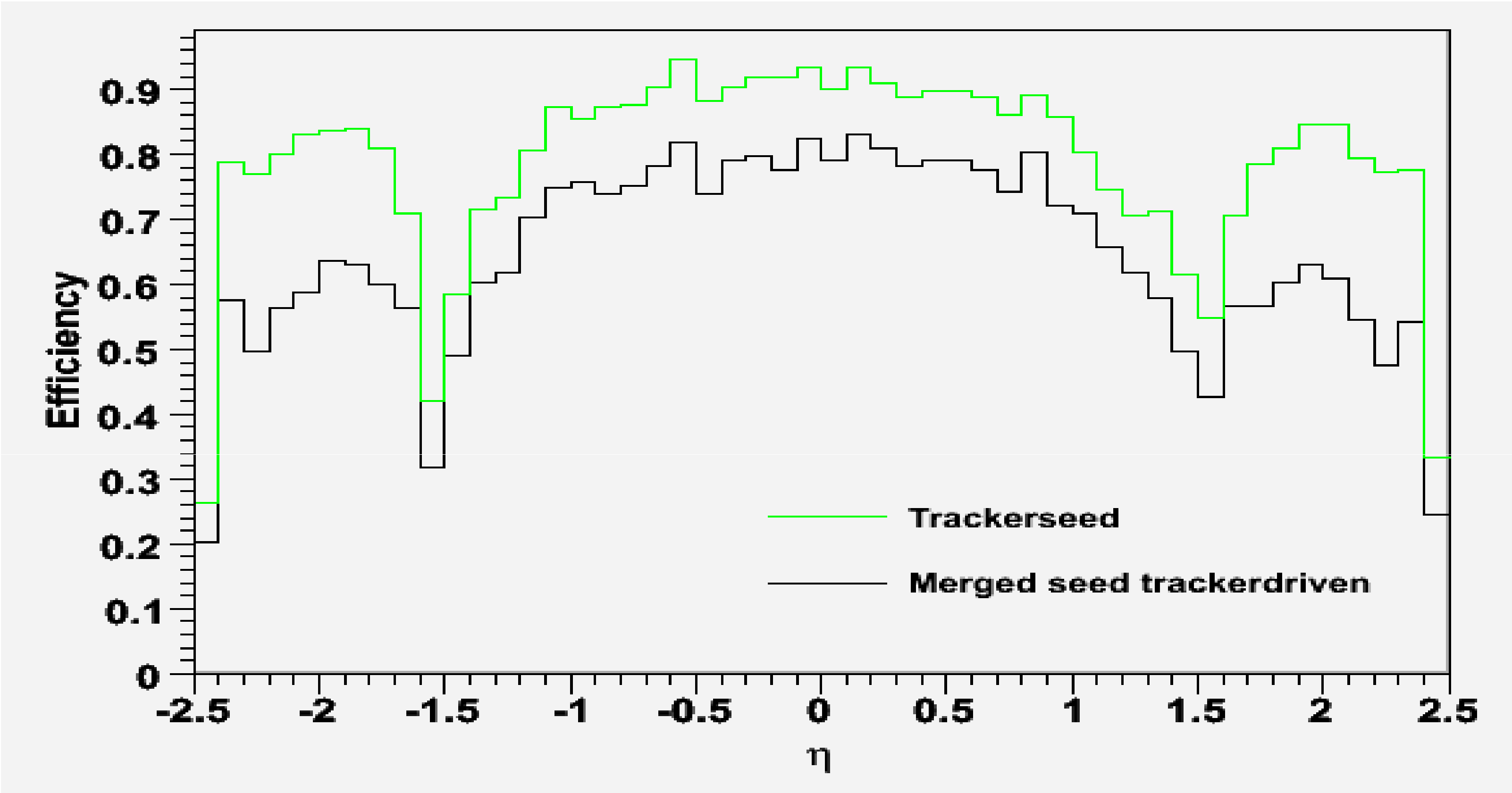
  OR
  - trackerDrivenElectronSeeds:SeedsForGsf (called ***Trackerdriven seed*** on graphs)

Checking provenance (Pt10GeV)



- These two graphs should be identical

Checking provenance(Pt10GeV)

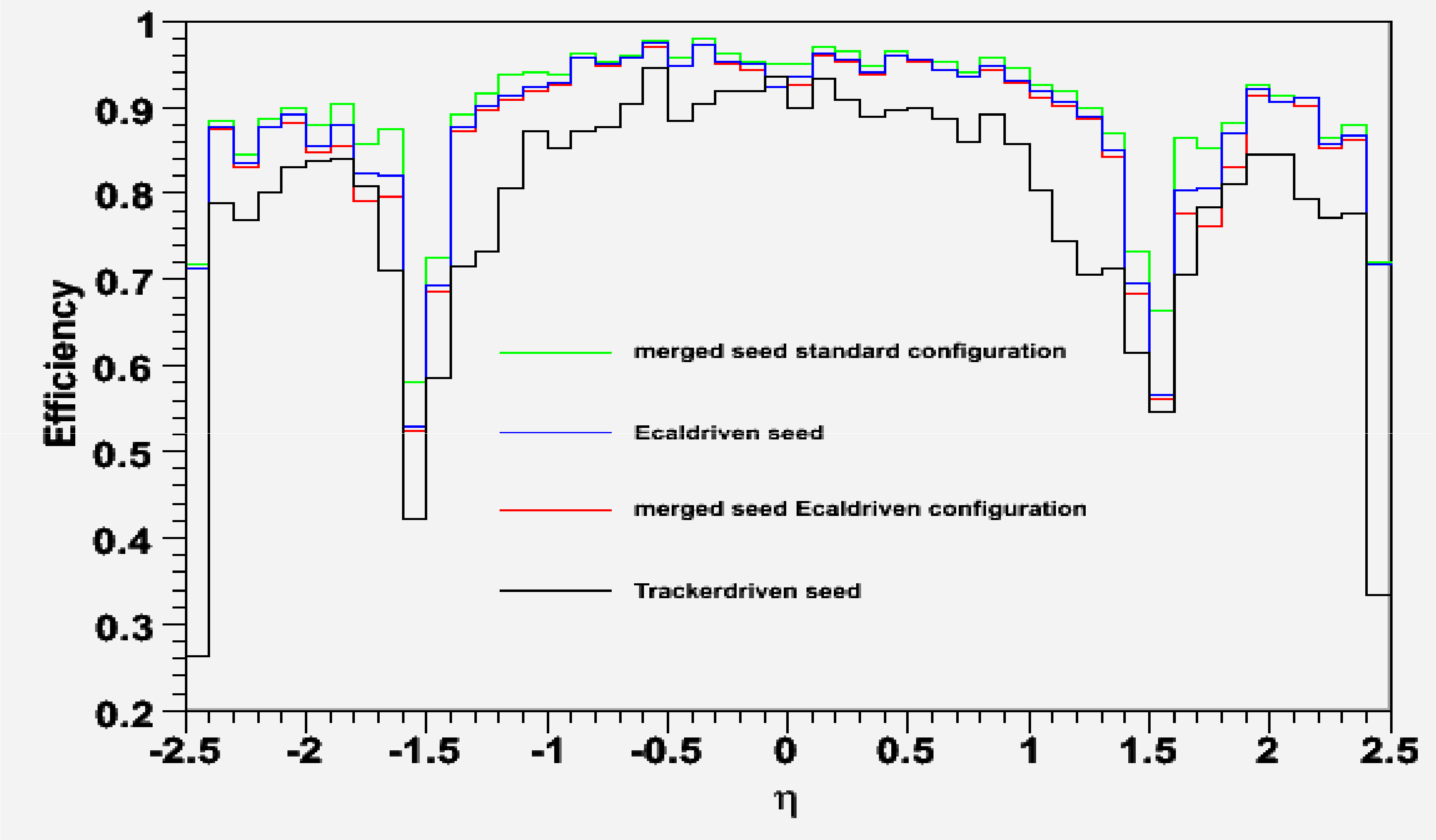


These two graphs should be identical

Due to:

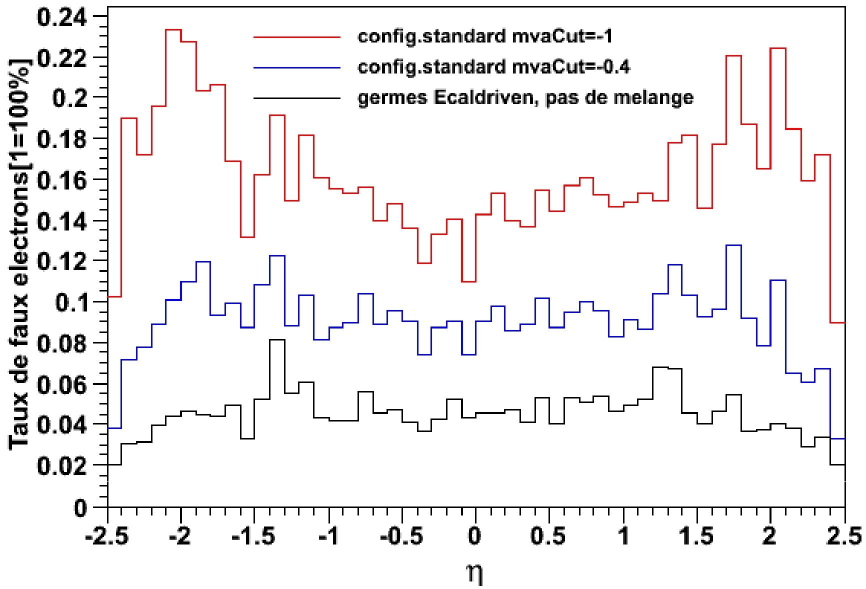
- Merging of seeds → **solved now**
- Trajectory cleaning → **under way...**

Efficiency result (Pt10GeV)

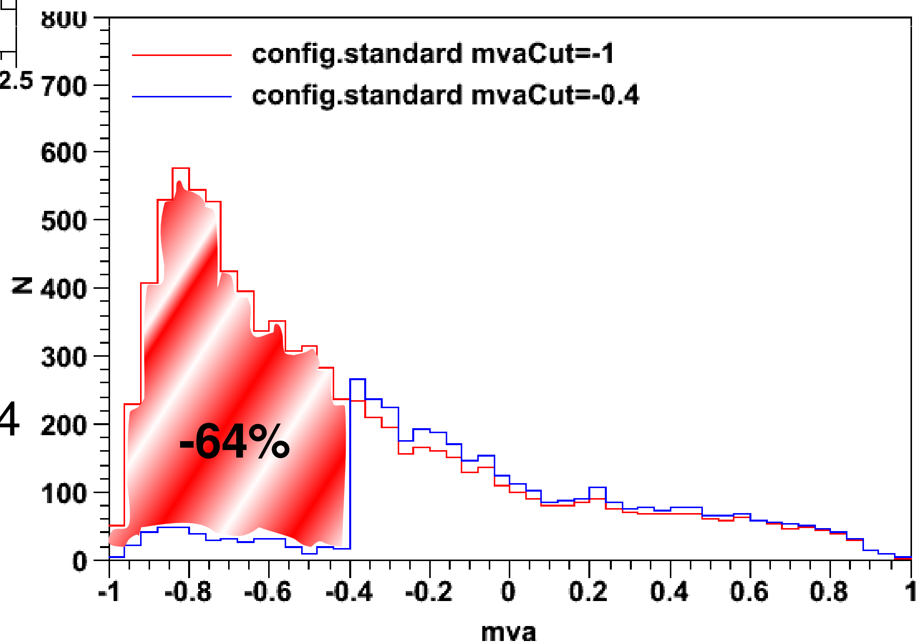


- Trackerdriven significantly less efficient (on isolated electron)
- Adding these electrons gives +2.3% increase in efficiency

Fake rate

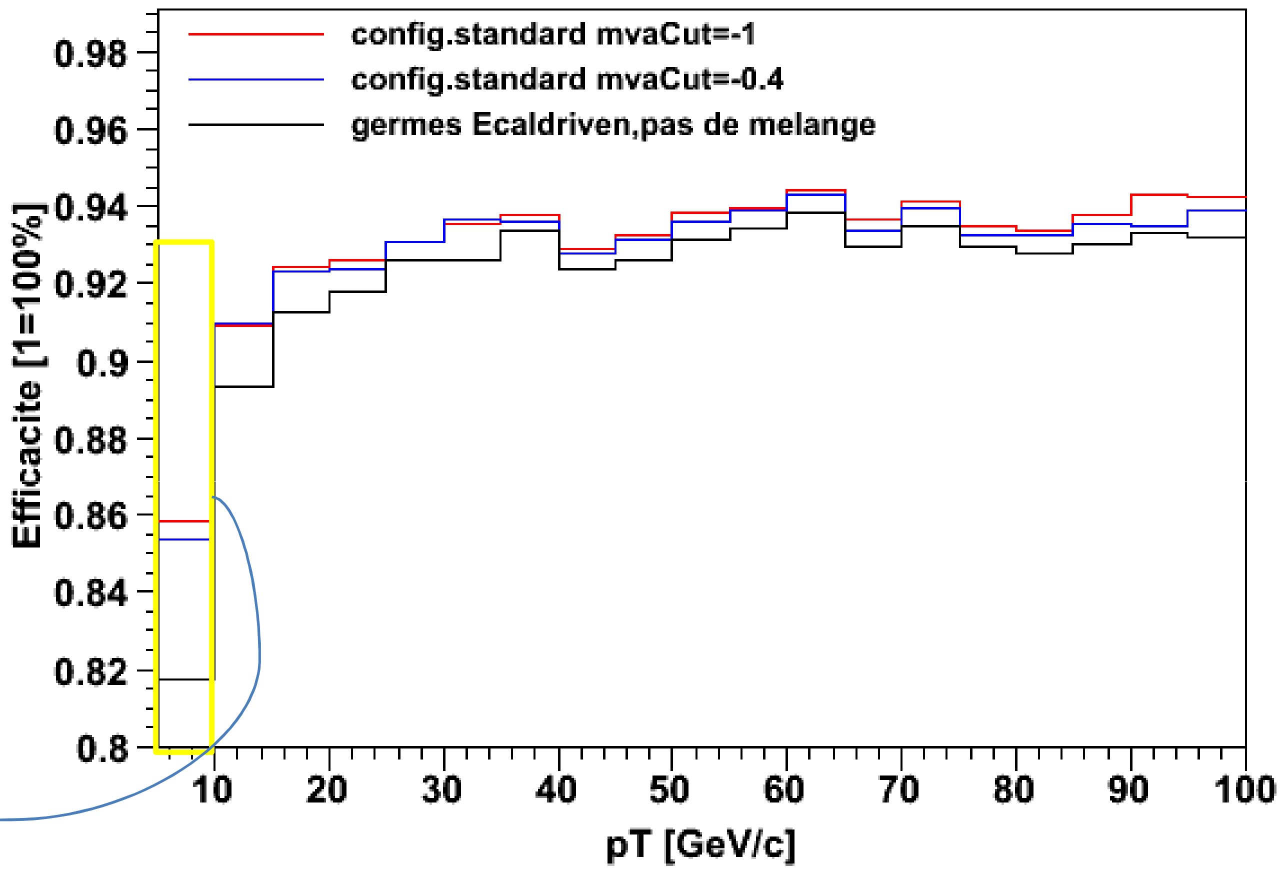


Fake rate en η	
Standard config., no cut	16.2%
Standard config., mvaCut=-0.4	9%
Ecaldriven seeds, no merging.	4.5%



- mva cut on candidates Trackerdriven only
- -64% of fake electron with mvaCut=-0.4
- with cut: Fakerate of 9% instead of 16,2%

Efficiency:Pt=5-100GeV



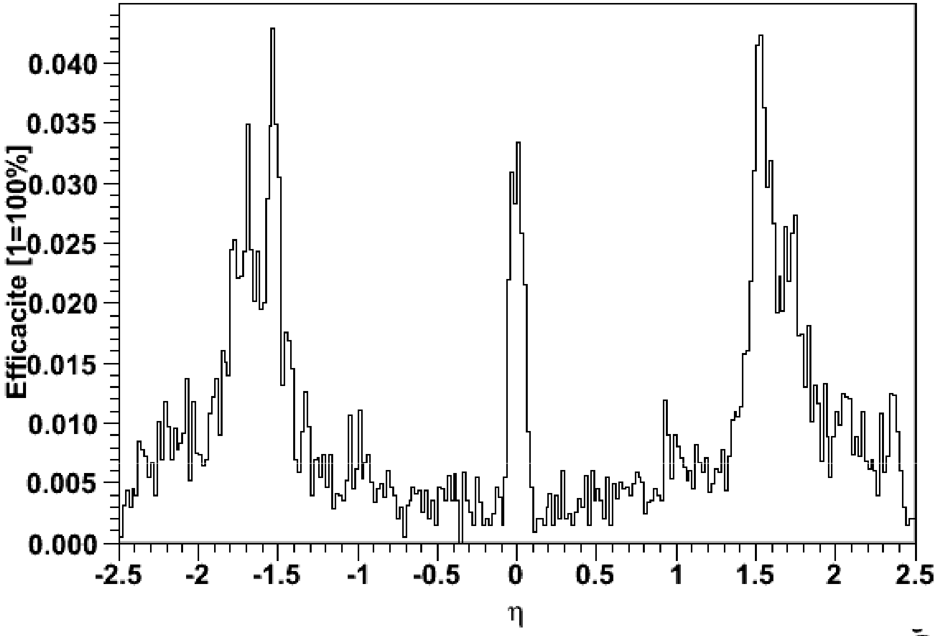
+4% (5<pT<10) without cut
+3.6% (5<pT<10) with cut

Global increase : +0.9% without cut
+0.7% with

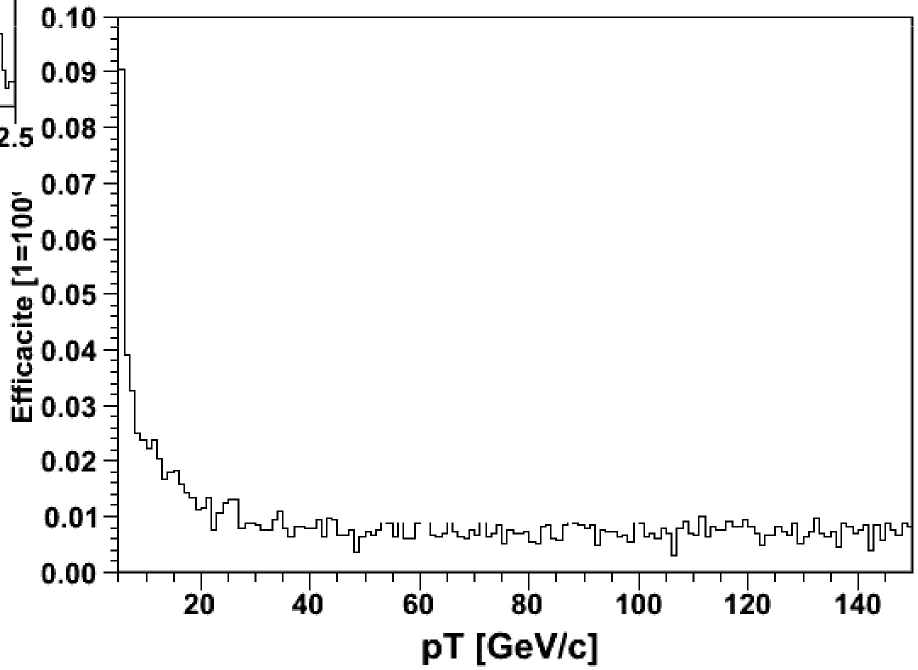
- **More gain at low pT**

Efficiency Trackerdriven only

Where are the electrons not found by ecaldriven algorithm?

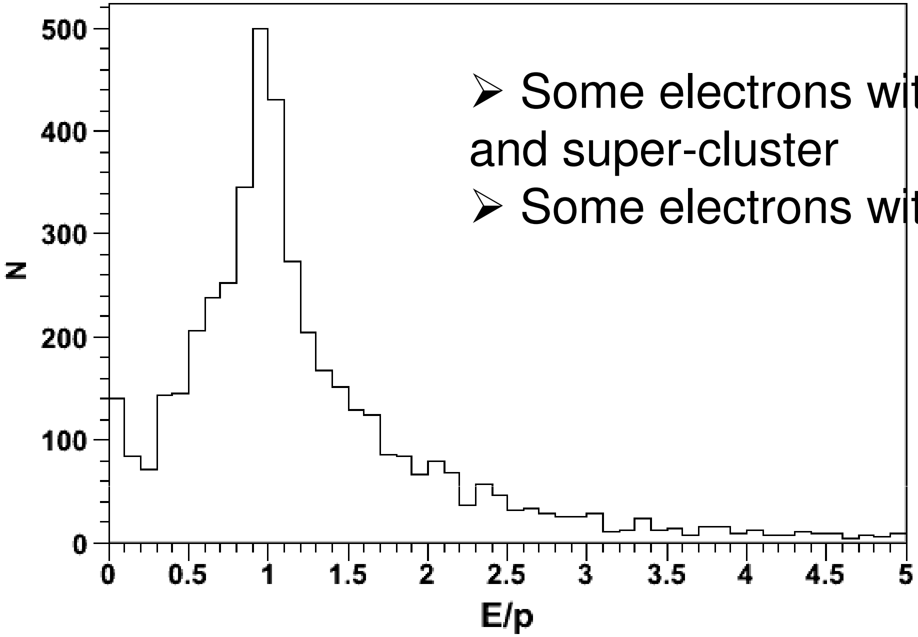


- ➢Main increase in the crack regions (eta=0 and eta=1.5)
- ➢Increase in the region where the materiel budget is high



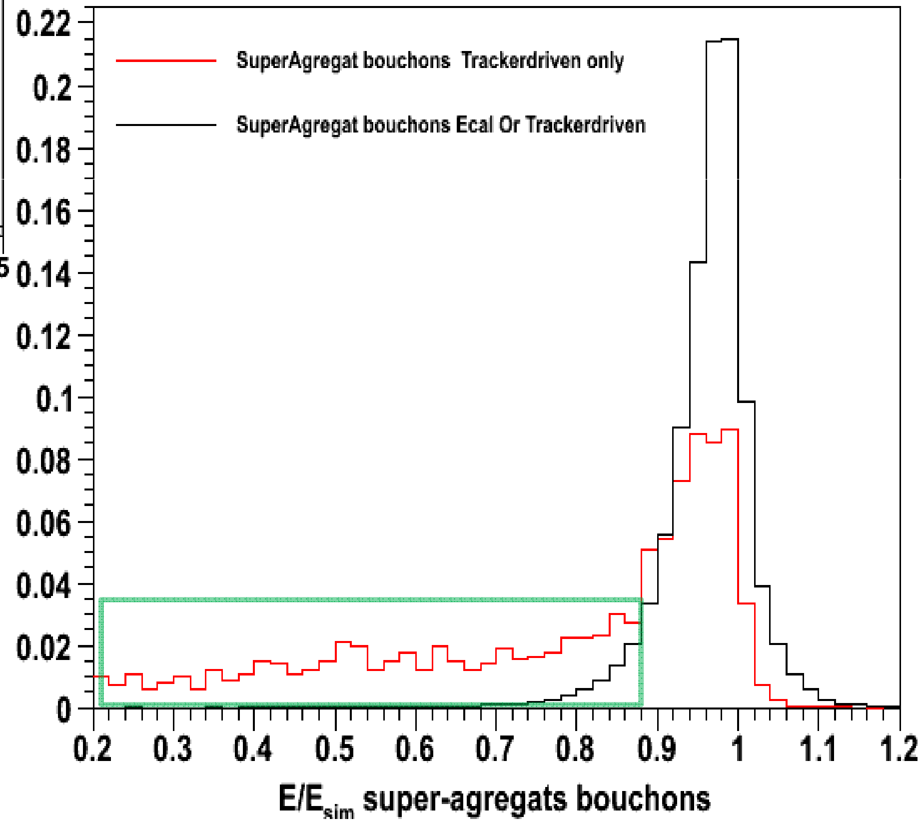
- ➢ Main increase at low pT
- ➢+9% at pT= 5 GeV/c
- ➢ Ecaldriven approach is at its limit at 5 GeV/c
- ➢Trackerdriven more suited for lower pT

Energy Trackerdriven only



- Some electrons with a good matching between track and super-cluster
- Some electrons with a low E/p

- Some electrons with bad valuation of E_{SC} perhaps due to conversions



Conclusion

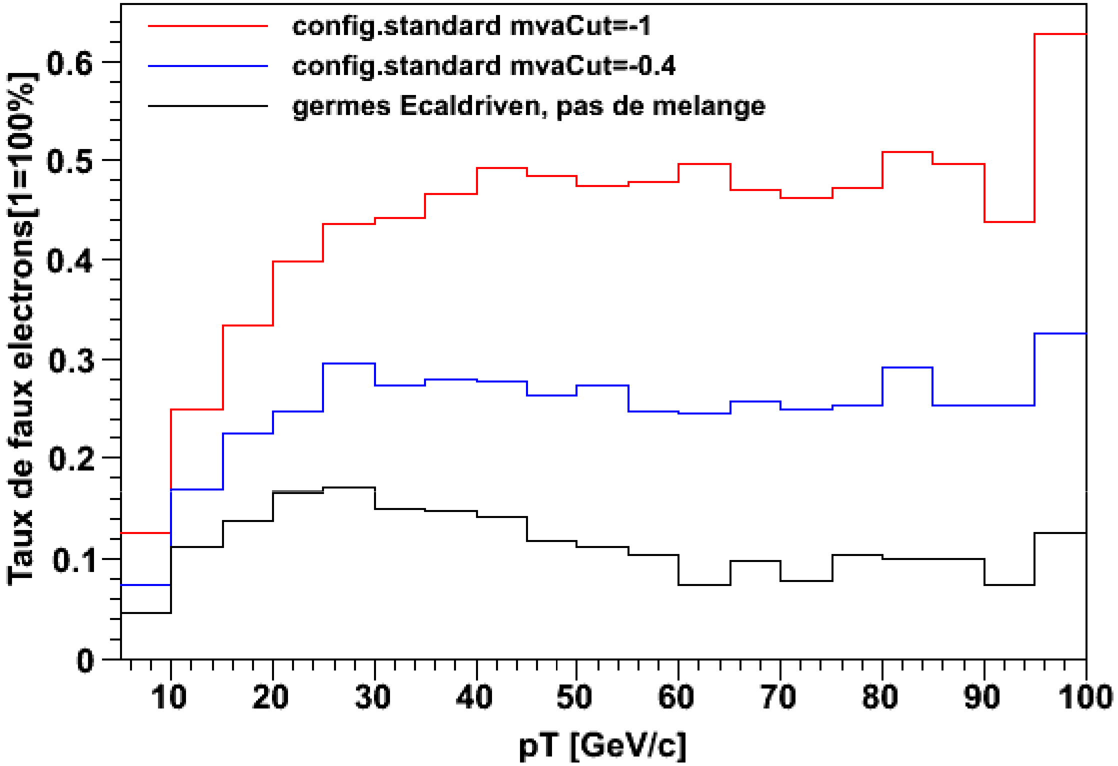
- We have now in CMSSW 31X a merged collection of GSFElectrons
    - Benefit from the two approaches
    - Efficiency : + 0.7%(FlatPt5to100GeV)

- The Ecaldriven approach becomes increasingly more difficult going down to very low pT
    - 5 GeV nearly the limit of ecaldriven approach
    - +9% by merging with trackerdriven at this pT

- Apart from low pT, the merging with trackerdriven electrons provide increase in efficiency in the crack region

- The fake rate is under control
    - although larger by nearly a factor 2
    - includes real electrons (e.g. from conversion)

Backup



Fake rate en pT	
Config.standard, sans coupure	44%
Config.standard, mvaCut=-0.4	25%
Germes Ecaldriven, sans mél.	11.4%

Fake rate pour 5<pT<10 GeV/c	
Config.standard, sans coupure	12.7%
Config.standard, mvaCut=-0.4	7.5%
Germes Ecaldriven, sans mél.	4.6%