Pion-Proton separation from H2 TestBeam

Antoine Pingault

Department of Physics and Astronomy
Ghent University

02 July, 2018

GHENT UNIVERSITY

FACULTY OF SCIENCES

Cerenkov counters analysis

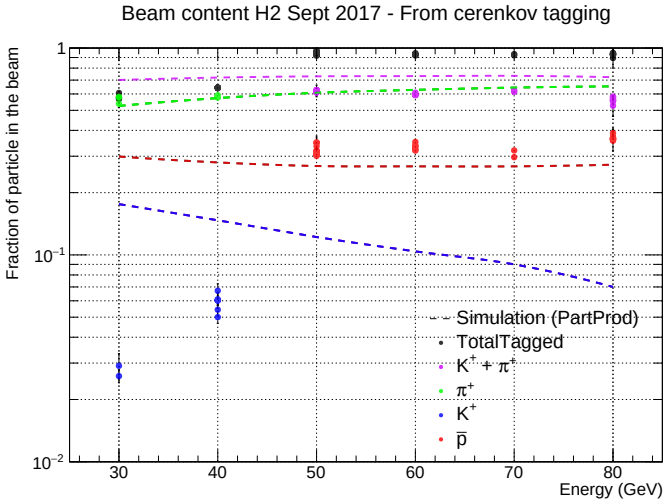
Set of python packages to:

1. Simulate counters of various characteristics (gas, length, intrinsic efficiciency)
2. Extract counters performance from dedicated run in test beam. (tag efficiency vs pressure/momentum for each particle)
3. Parse elog to extract Cerenkovs info from the comment section.
   - elog with all relevant info then stored in a json file. (easier parsing...)
4. Compute and compare real efficiency from expected.
5. Estimate particle content in the beam.

Particle content in H2 1/2

- Fractions defined as amount of tag from the cerenkov counters compared to total number of reconstructed event.
- Simulation (PARTPROD from CERN) values computed at the interaction point, i.e. no correction applied for loss/decay along the beam to SDHCAL.
- At 30-40GeV, Total tag only at ~60-65%: Because Protons/Kaons not tagged
- After 40GeV, Total Tag ~ 95%.
  - Found out both cerenkovs are going blind simultaneously during some sub-spill.
  - Efficiency could be higher if we can fix this (more on that later).

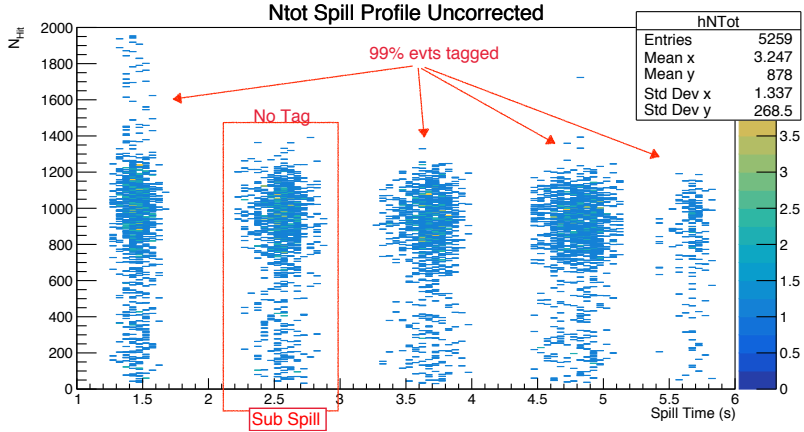
Particle content in H2 2/2



Tag inefficiency 1/2

- Most event not tagged are within the same subspill
- Seems random throughout all runs.
- If I remove these subspills: $>99\%$ hadrons are tagged.
- Will discuss with CERN experts before next testbeam

Tag inefficiency 1/2



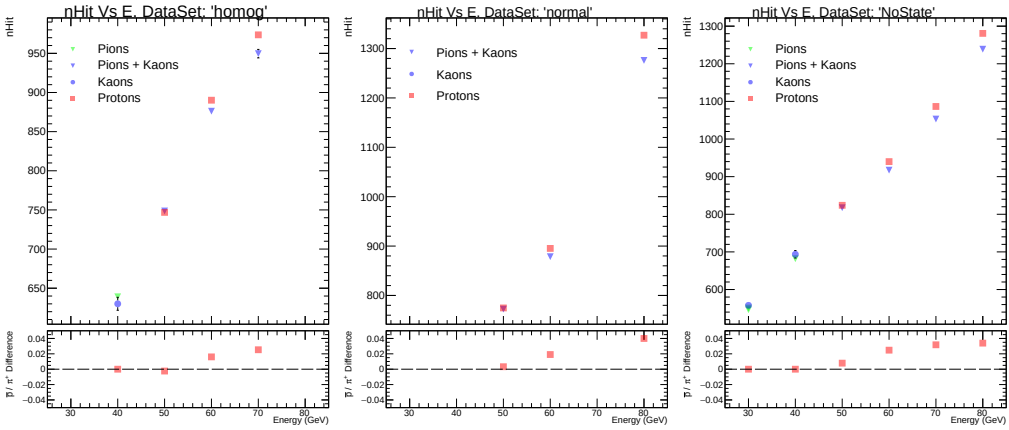
Cerenkov in the data

- Cerenkov parsing done in Trivent, info is stored in the output lcio file (`TDHCAL_${runNumber}.slcio`):
  - A bool in the evt header (`CerenkovTag`)
  - Separate `CalorimeterHit` collection (`CERENKOV_HIT`) with cerenkovID coded in the `energy`.
- Also included in sdhcal analysis from arnaud/guillaume (using CaloSoftWare).
  - Info stored in root tree with all other sdhcal variables. (nHit, etc.).

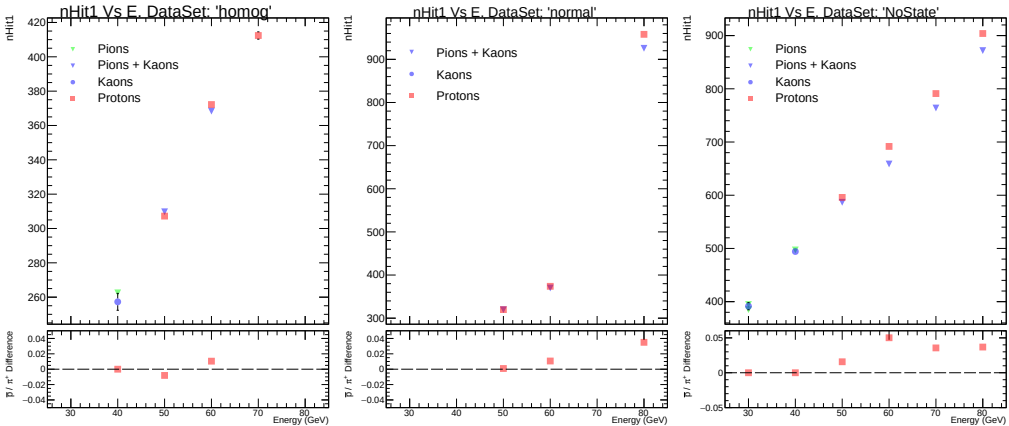
Pions vs Protons

- Plots of a few sdhcal variables for the 3 tested State in last test beam.
- Stats is rather low (2-8k hadrons per point), with ~40% protons & ~60% pi/k for E>50GeV.
- Realistically it's Proton vs Kaon + Pion, not Proton vs Pion.
- Still need to compare with simulation for proper result.

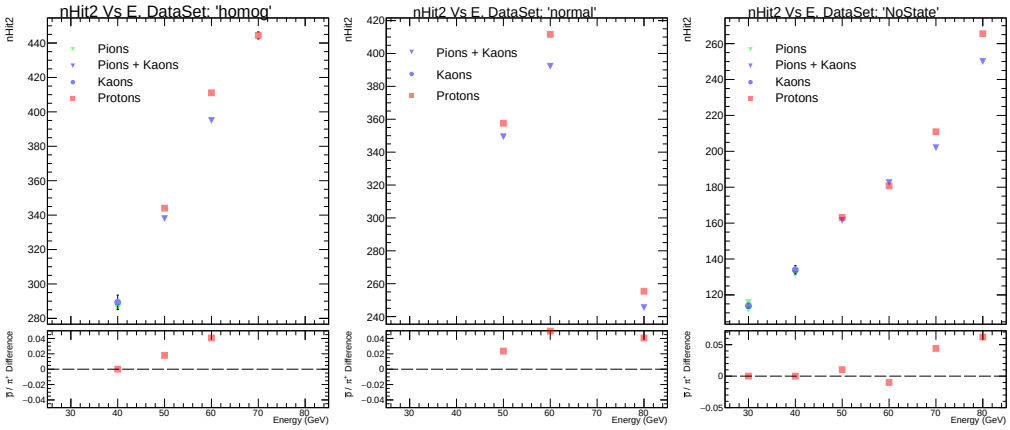
Nhit


nHit Vs E, DataSet: 'homog'
nHit Vs E, DataSet: 'normal'
nHit Vs E, DataSet: 'NoState'
nHit
Pions
Pions + Kaons
Kaons
Protons
p / π Difference
Energy (GeV)

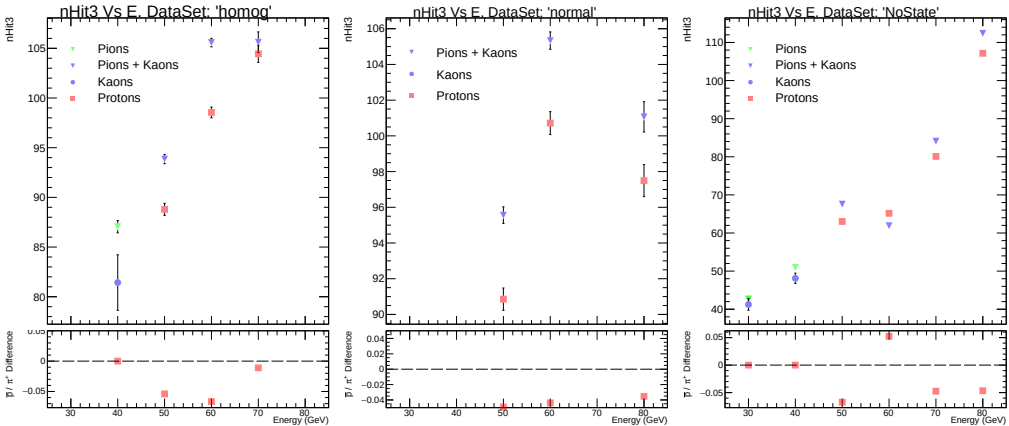
Nhit1


nHit1 Vs E, DataSet: 'homog'
nHit1 Vs E, DataSet: 'normal'
nHit1 Vs E, DataSet: 'NoState'
nHit1
Pions
Pions + Kaons
Kaons
Protons
π / π Difference
Energy (GeV)

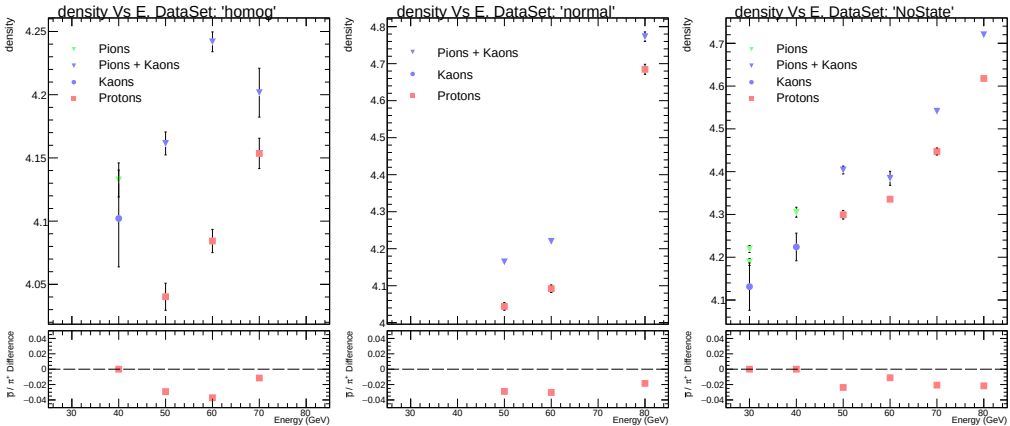
Nhit2


nHit2 Vs E, DataSet: 'homog'
nHit2 Vs E, DataSet: 'normal'
nHit2 Vs E, DataSet: 'NoState'
nHit2
Pions
Pions + Kaons
Kaons
Protons
p / π Difference
Energy (GeV)

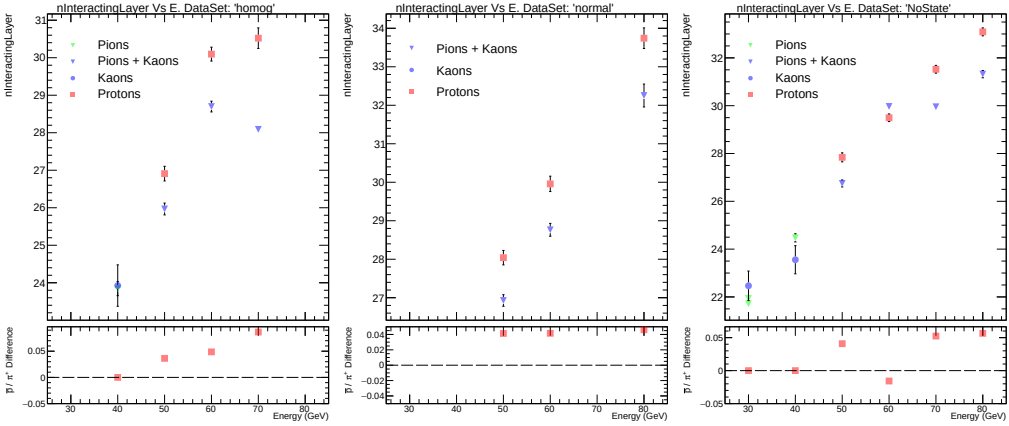
Nhit3


nHit3 Vs E, DataSet: 'homog'
nHit3 Vs E, DataSet: 'normal'
nHit3 Vs E, DataSet: 'NoState'
nHit3
Pions
Pions + Kaons
Kaons
Protons
p / π Difference
Energy (GeV)

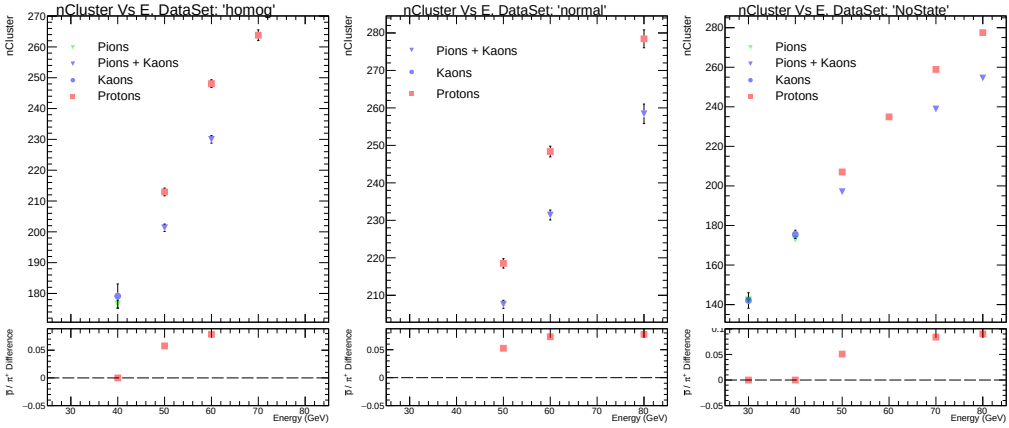
Density


density Vs E, DataSet: 'homog'
density Vs E, DataSet: 'normal'
density Vs E, DataSet: 'NoState'
density
Pions
Pions + Kaons
Kaons
Protons
π / π⁺ Difference
Energy (GeV)

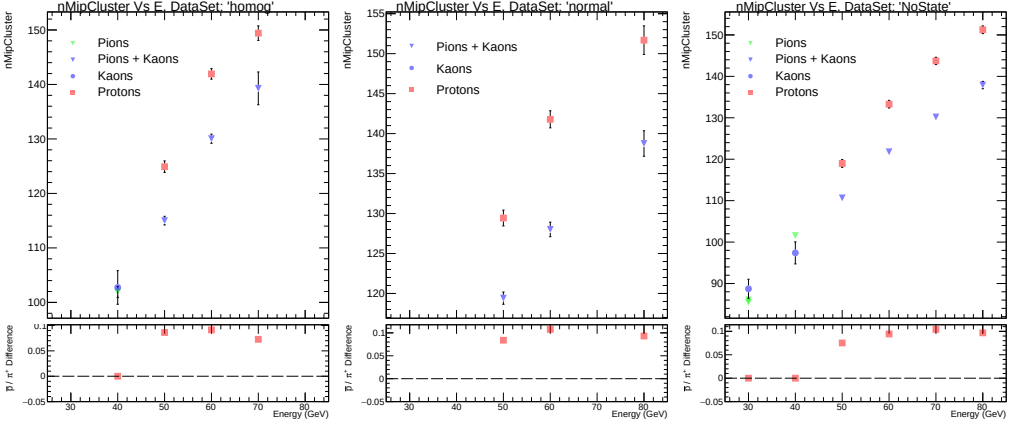
nInteractingLayer



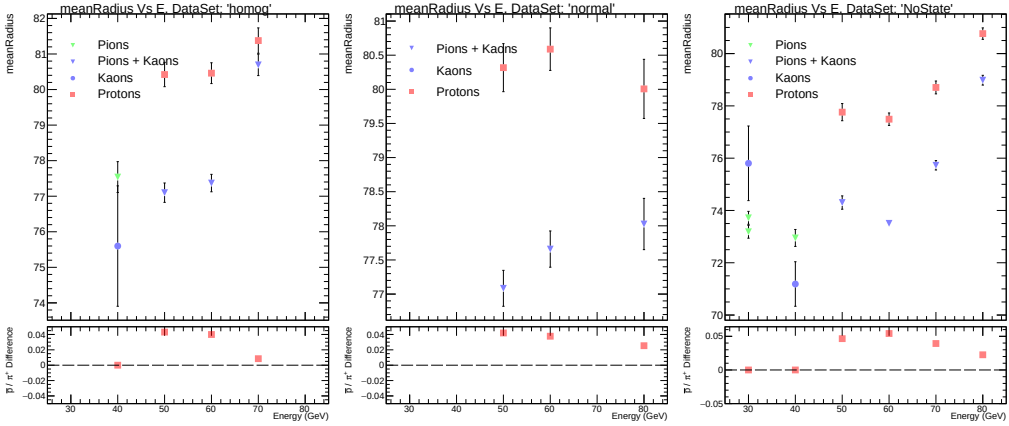
nCluster


nCluster Vs E, DataSet: 'homog'
nCluster Vs E, DataSet: 'normal'
nCluster Vs E, DataSet: 'NoState'
nCluster
Pions
Pions + Kaons
Kaons
Protons
p / π Difference
Energy (GeV)

nMipCluster



meanRadius


meanRadius Vs E, DataSet: 'homog'
meanRadius
Pions
Pions + Kaons
Kaons
Protons
p / π Difference
Energy (GeV)
meanRadius Vs E, DataSet: 'normal'
meanRadius
Pions + Kaons
Kaons
Protons
p / π Difference
Energy (GeV)
meanRadius Vs E, DataSet: 'NoState'
meanRadius
Pions
Pions + Kaons
Kaons
Protons
p / π Difference
Energy (GeV)