Seeding

FLORENCIA CASTILLO

RÉUNION ATRAPP AVEC IJCLAB

25/04/2023

Introduction

- For this example we are using ttbar mu 0
  - mc21_14TeV.601229.PhPy8EG_A14_ttbar_hdamp258p75_SingleLep.recon.RDO.e8481_s4038_r14362
- How to extract seeding information from Athena in two different ways:
  - Summary tables for the track finder printed in the reco log file and the creation of SeedCollections
  - Ntuple creation (InDetPhysValMonitoring)

Tables from log.RAWtoALL

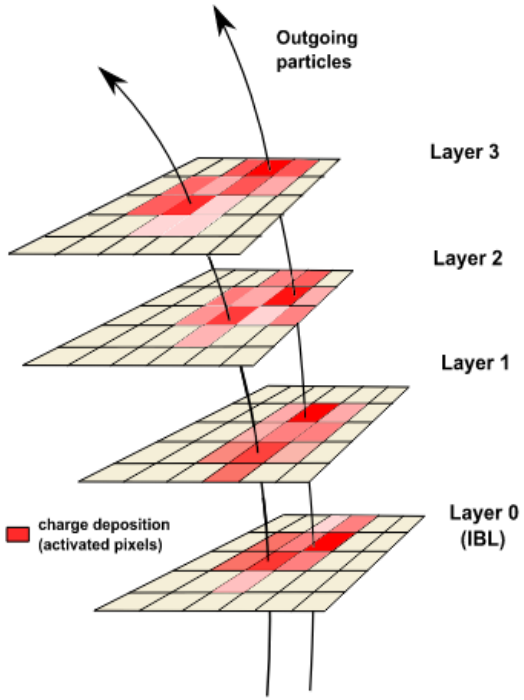
for example a simulated pion may have left 9 hits, but our reco only added 8 of them to the corresponding track --> **1 "lost"**

the "**wrong**" clusters are something like the opposite - hits added to a track that *do not come from the same particle*

SiSPSeededTracks = Seeding + Kalman extension

Statistic for SiSPSeededTracks

Probability to lose	0	1	2	3	4	>=5 clusters
For all particles	0.9062	0.0355	0.0163	0.0084	0.0066	0.0149
For + particles	0.9048	0.0356	0.0162	0.0085	0.0062	0.0160
For - particles	0.9077	0.0354	0.0163	0.0082	0.0071	0.0137
Barrel region						
0 wrong clusters	0.9141	0.0226	0.0124	0.0059	0.0054	0.0081
1 wrong clusters	0.0106	0.0020	0.0010	0.0005	0.0003	0.0007
2 wrong clusters	0.0038	0.0004	0.0002	0.0000	0.0000	0.0003
3 wrong clusters	0.0003	0.0001	0.0000	0.0000	0.0001	0.0000
>=4 wrong clusters	0.0005	0.0001	0.0001	0.0000	0.0000	0.0002
Transition region						
0 wrong clusters	0.8922	0.0294	0.0141	0.0086	0.0056	0.0130
1 wrong clusters	0.0131	0.0029	0.0009	0.0005	0.0003	0.0007
2 wrong clusters	0.0034	0.0007	0.0003	0.0002	0.0001	0.0006
3 wrong clusters	0.0004	0.0002	0.0000	0.0000	0.0000	0.0001
>=4 wrong clusters	0.0013	0.0001	0.0001	0.0000	0.0000	0.0003



the ideal tracking would give zero lost and zero wrong hits in every case, with 100% efficiency and 0% fake rate

Tables from log.RAWtoALL

SiSPSeededTracks = Seeding + Kalman extension

Endcap region						
0 wrong clusters	0.8652	0.0395	0.0158	0.0076	0.0069	0.0179
1 wrong clusters	0.0187	0.0030	0.0018	0.0005	0.0005	0.0015
2 wrong clusters	0.0044	0.0009	0.0004	0.0001	0.0001	0.0003
3 wrong clusters	0.0008	0.0001	0.0000	0.0000	0.0000	0.0001
>=4 wrong clusters	0.0015	0.0002	0.0001	0.0000	0.0000	0.0001
Forward region						
0 wrong clusters	0.8570	0.0433	0.0190	0.0116	0.0080	0.0149
1 wrong clusters	0.0167	0.0024	0.0017	0.0002	0.0003	0.0009
2 wrong clusters	0.0018	0.0006	0.0000	0.0002	0.0002	0.0006
3 wrong clusters	0.0002	0.0000	0.0000	0.0000	0.0000	0.0002
>=4 wrong clusters	0.0012	0.0000	0.0000	0.0000	0.0000	0.0003

```
Efficiency reconstruction (number lose+wrong < 3) =   0.95391 (  0.86039 )
                           For barrel     region =   0.96536 (  0.87123 )
                           For transition region =   0.95507 (  0.85205 )
                           For endcap     region =   0.94656 (  0.85329 )
                           For forward    region =   0.94025 (  0.86991 )
```

high precision if you gained / lost good candidates

Seed redundancy

Tracks/inputs= 0.095

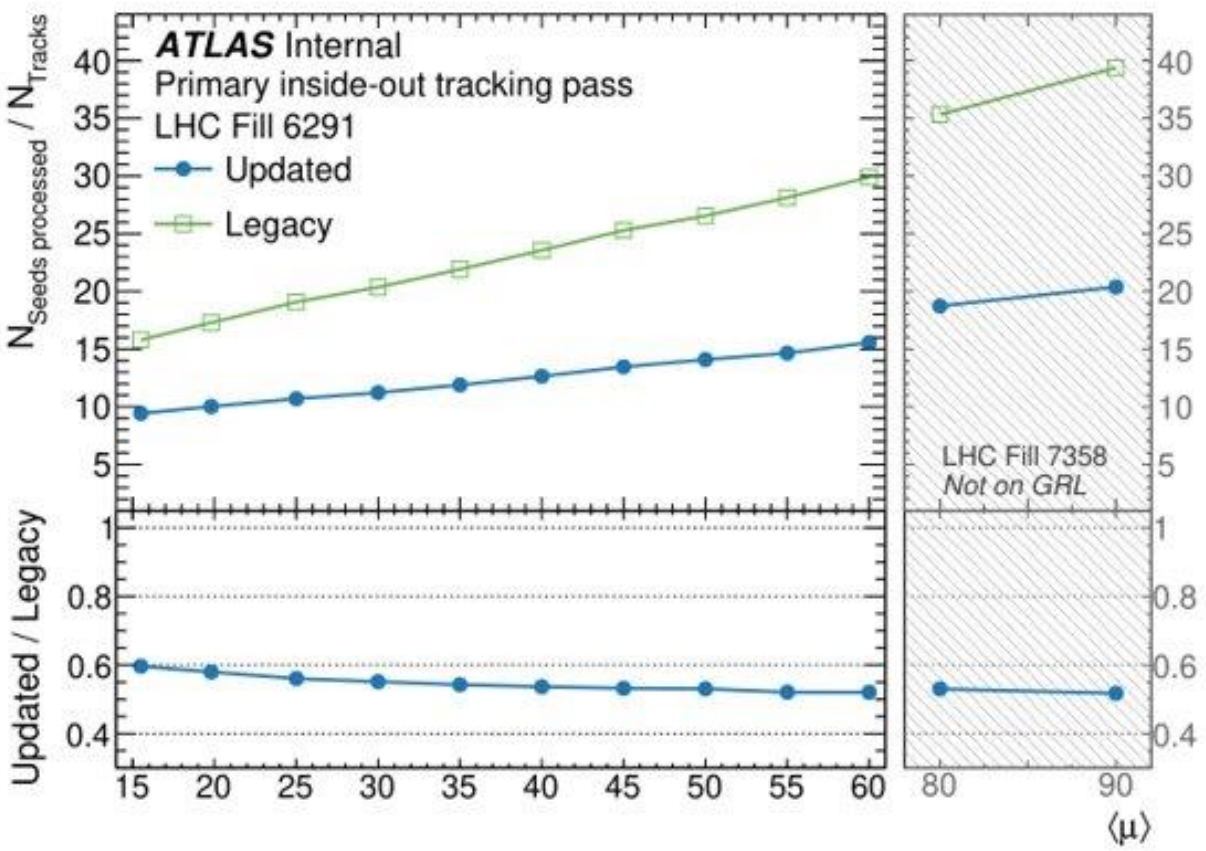
Kind of seed	PPP	PPS	PSS	SSS	ALL
Input seeds	413998	0	0	211827	625825
No track parameters	1274	0	0	10624	11898
Used seeds	57947	0	0	67600	125547
Used seeds brem	0	0	0	0	0
Det elements in road	80.67	0	0	64.789	72.12
Two clusters on DE	0	0	0	0	0
Wrong DE road	0	0	0	23	23
Wrong initialization	0	0	0	0	0
Can not find track	12167	0	0	10466	22633
It is not new track	5459	0	0	5194	10653
Attempts brem model	0	0	0	0	0
Output tracks	40572	0	0	53065	93637
Output extra tracks	1015	0	0	2257	3272
Output tracks brem	0	0	0	0	0
Seeds with track	39557	0	0	50808	90365

Tracks/inputs= 0.24

Seeding redundancy = 625825/90365 ~ 7

- We first build SCT seeds ("SSS")
- Then we run track finding with those
- And only then we build Pixel seeds ("PPP") and make tracks from those, removing any we found previously.
- As a result, it will always look in the table like the PPP seeds are less efficient at being turned into tracks - but this is not a property of the Pixel per-se, but just the result of our choice to first run the SCT seeds

Seed redundancy



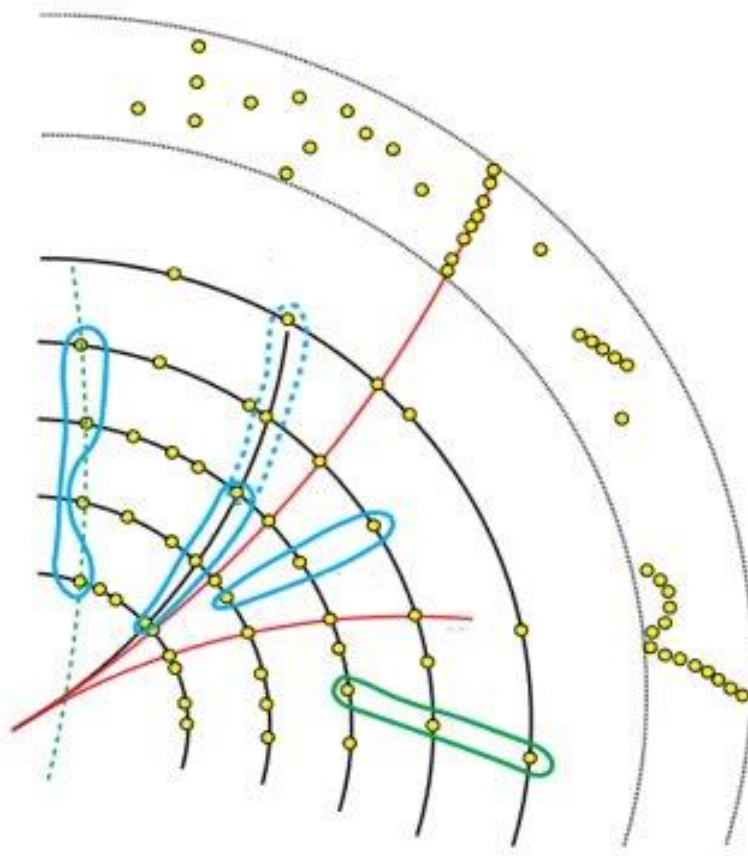
Documentation:

- https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PUBNOTES/ATL-PHYS-PUB-2021-012/
- https://cds.cern.ch/record/2815589

Seeding redundancy ~ 7 (mu=0. Itk)

Efficiency and fake rate plots from dummy tracks

- These seeds are **not expected to map 1:1** to tracks - there will always be many "good" seeds corresponding to one track, and an efficient seeder will not only find at least one seed per track, but also **avoid processing too many seeds per track** (otherwise, we would be wasting CPU cycles finding the same thing over and over)
- If we want to study the **fraction of seeds where not all hits come from the same true particle,** we can look to the "fake" plots IDPVM. **Caveat**
  - The seed track collection basically uses the three space-points to estimate a trajectory and then attempts to add the detector measurements that are used to form the 3D space points forming the seeds to the trajectory to build a track candidate. But it **may drop** some of them, **if they are not compatible with the trajectory model**, and the trajectory estimate itself is also not really a part of the seed.
    - So seed eta/pt/phi are not the pt/eta/phi the seed reports in the real reconstruction



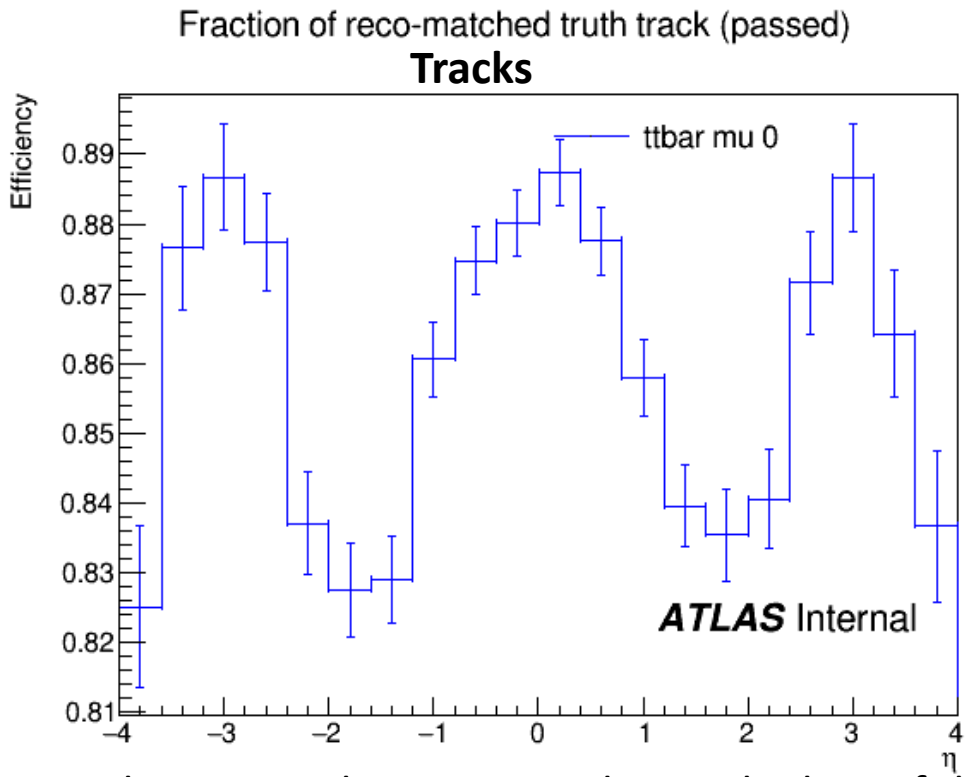
Efficiency and fake plots

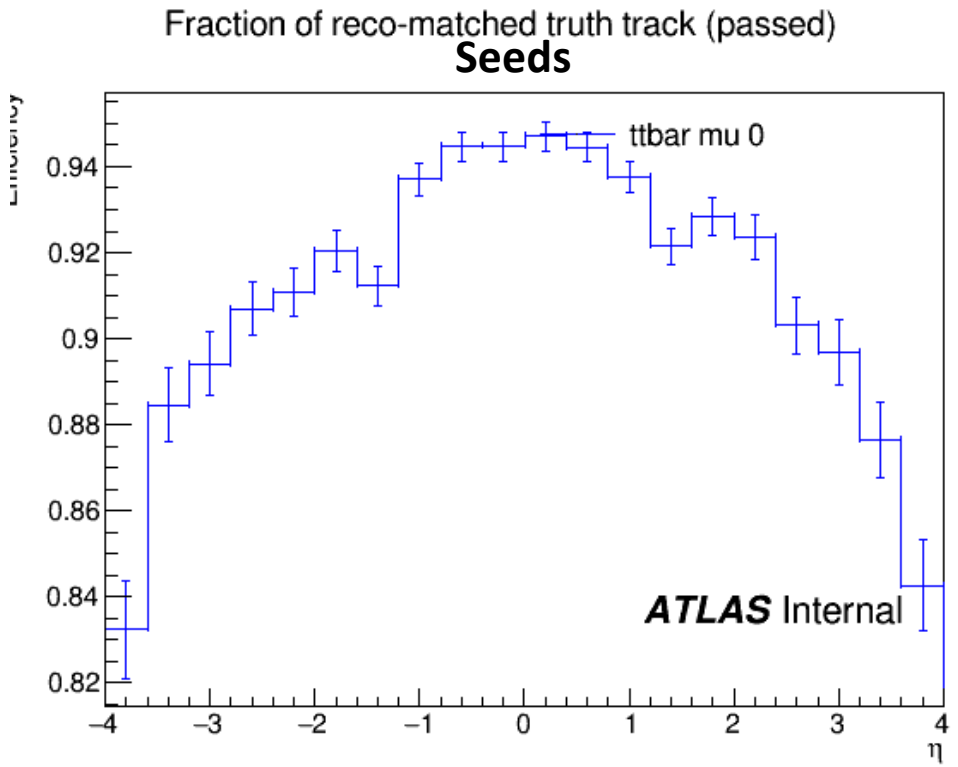
But another way of looking at it could be to monitor

- Efficiencies
- Seed redundancy (number of seeds processed : number of tracks reconstructed)
  - It is something like a measure of how many times - on average, including both "good" duplicates and random combinatorics - we process a seed per track we find
- CPU speed

Together, these will encompass the effect of both genuine and incorrect seeds, in a metric which may be more meaningful than a "seed fake rate" on its own

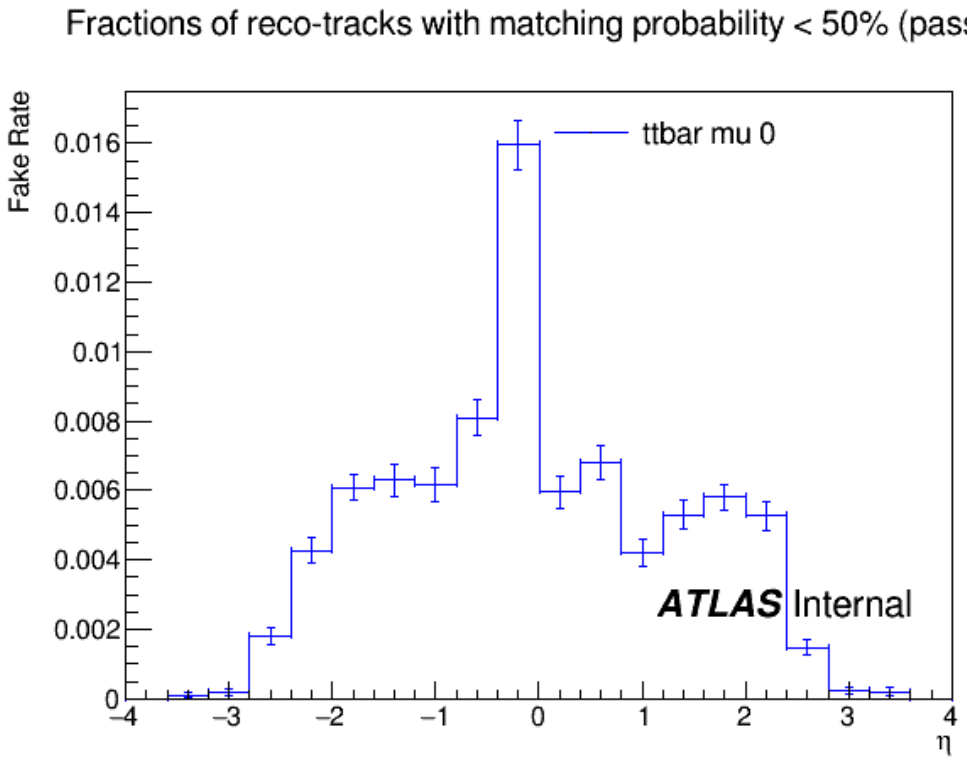
Efficiencies

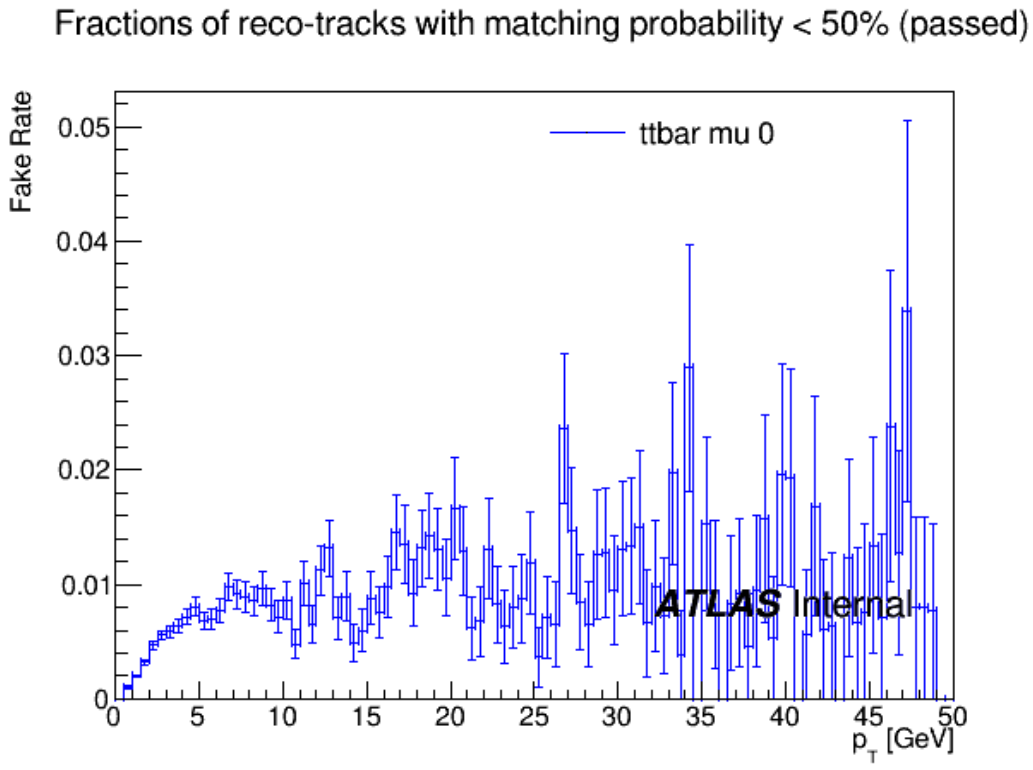




Efficiency drop around 1 < eta < 2 due to the loss of the seeds.
It can be recover but at the coast of CPU in seeding and track finding:
https://indico.cern.ch/event/1268668/contributions/5346716/attachments/2625407/4540151/UpgradeTracking_230405.pdf

Fake rates in seeds





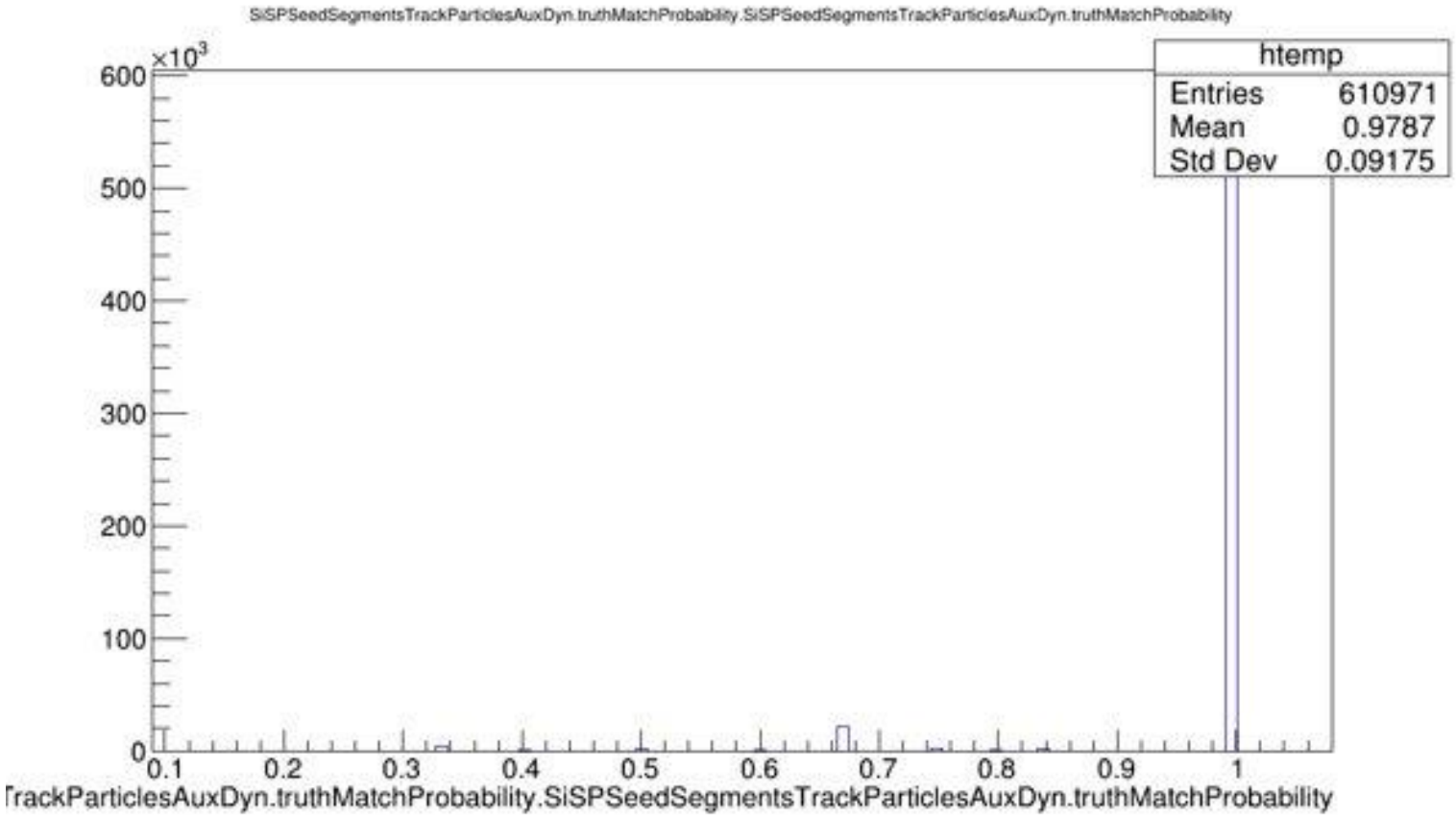
CPU Time

- Virtual memory (vMem) is memory that is allocated by virtualization systems.
- The malloc() function stands for memory allocation, that allocate a block of memory dynamically

```
INFO ================================================================================
INFO ================================================================================
INFO                          Component Level Monitoring
INFO ================================================================================
INFO Step           Count     CPU Time [ms]  Vmem [kB]      Malloc [kB]         Component
INFO --------------------------------------------------------------------------------
INFO Execute          999       4325512.92     0              871270              SiSPSeedSegmentsTrackParticleCnvAlg
INFO Execute          999       98679.64       0              1545859             ITkAmbiguitySolver
INFO Execute          999       45177.39       0              3449461             ITkSiSpTrackFinder
INFO Execute          999       27997.41       0              3643086             ITkPRD_MultiTruthMakerSi
INFO Execute          999       25750.39       0              24520               OutputStreamAOD
INFO Execute          999       23201.12       0              5048018             ITkPixelClusterization
INFO Execute          999       20911.10       0              3159431             GEN_AOD2xAOD
INFO Execute          999       9014.07        0              51477               TrackTimeExtensionAlg
INFO Execute          999       8884.31        0              3500401             ITkStripClusterization
INFO Execute          999       7030.72        0              146965              ITkTrackParticleCnvAlg
INFO Execute          999       6477.99        0              -62                 ITkTrackClusterAssValidation
INFO Execute          999       6006.54        0              -202                IncidentProcAlg1
INFO Execute          999       4981.60        0              2213969             ITkSiTrackerSpacePointFinder
INFO Execute          999       4595.46        0              122875              SiSPSeedSegmentsDetailedTruthMaker
INFO Execute          999       2910.00        0              95684               InDetPriVxFinder
INFO Execute          999       2459.16        0              17200               SiSPSeededTracksDetailedTruthMaker
INFO Execute          999       2291.13        0              10881               ResolvedTracksDetailedTruthMaker
```

Purity

- Number of HITS that are generated by true particles. "TruthMatchProbability" decoration on the seed track collection. Caveat
  - These are defined as the fraction of the measurements on the seed that come from the same truth particle
  - So 1.0 = perfectly pure, all measurements from the same truth particle



Outlook

- First time looking to the seeding in ITK
  - We already know what kind of information we can extract from Athena
  - The idea is to discuss with ITK experts the interesting studies/comparisons to follow up and see if the current seeding performance is good enough for HL-LHC phase
  - Most likely to be part of the Software Performance of the ATLAS Track Reconstruction for HL-LCH paper/note