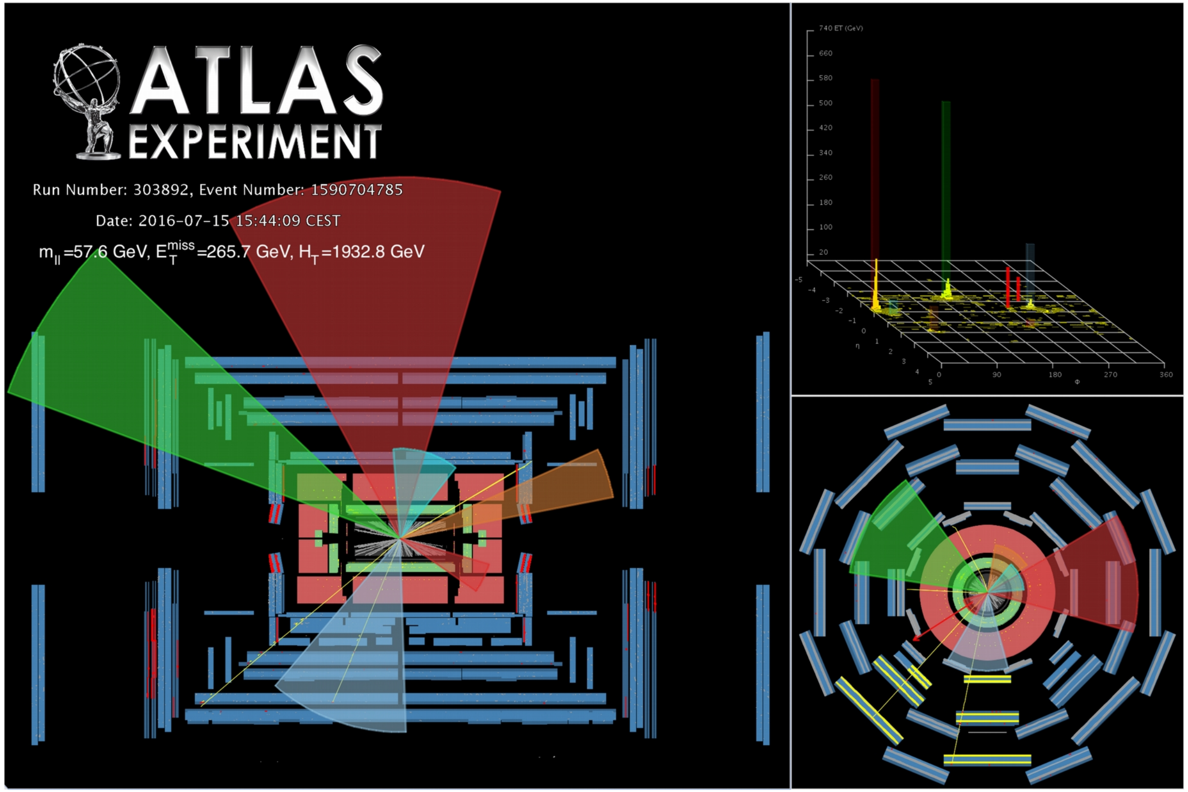
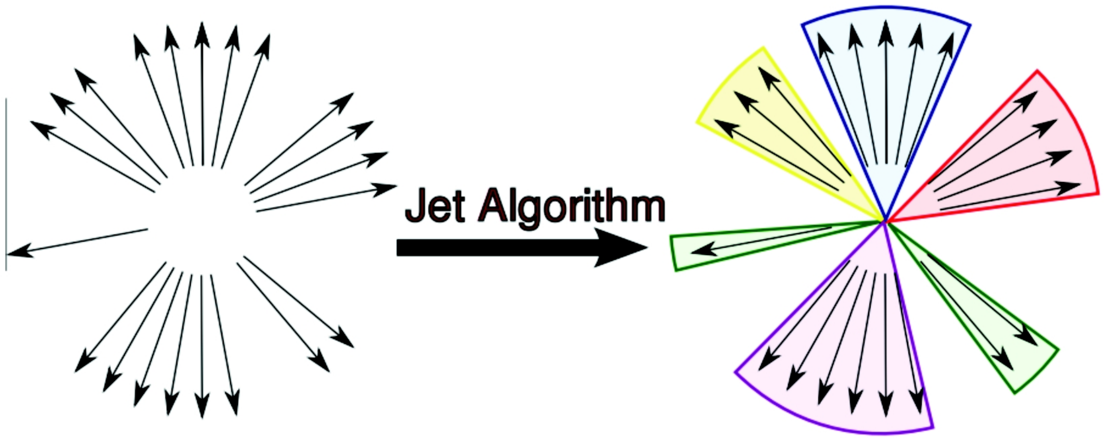
Jet Reconstruction and Inputs

Chris Young



What are Jets?

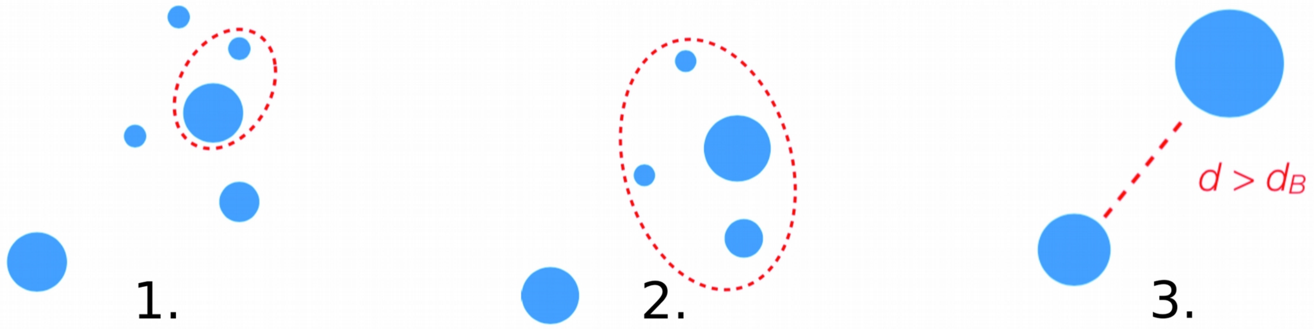
- Jet reconstruction is an organizational principle where we group sets of particles to form jets.
- Several jet algorithms could be run on the same event to get different “views” - none is “correct”.
- The jet algorithm defines how the particles are clustered into these jets – we can apply it to calorimeter objects, tracks, combinations of the two, and truth particles.
- We tend to use the same algorithm on stable truth particles as the reference for performance or measurements – note a jet is not the same as a parton!

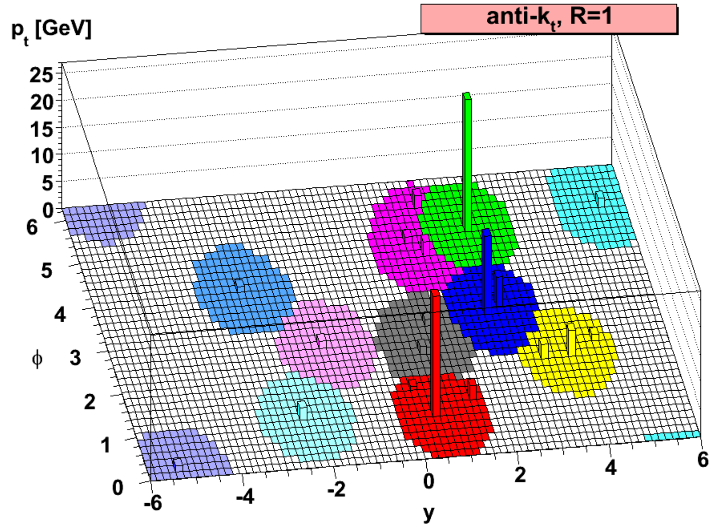


The Anti-k_t Algorithm

- Most jets in ATLAS use the Anti-k_t algorithm.
- This is a sequential combination algorithm defined by a “radius parameter”, R.
- We define a distance measure d_{ij}, and then combine the two particles with smallest d.
- Then repeat combining pairs with the lowest d_{ij} until the smallest d_{ij} is > d_B
- This results in reasonably round jets in the η,φ space.

$$d = \min(p_{T,i}^{-1}, p_{T,j}^{-1})\frac{\Delta_{i,j}^2}{R^2} \qquad d_B = p_{T,i}^{-2}$$



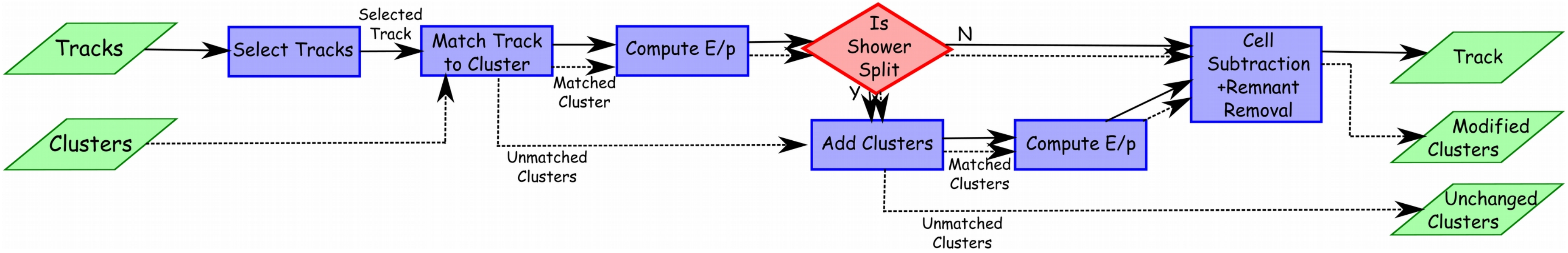


Topoclusters as Inputs

- The simplest jets use calorimeter topoclusters as inputs.
- These can either be at the electro-magnetic scale (EMTopo) or local calibration scale (LCTopo).
- Topoclusters were covered in the Calo talk of the tutorial – reminder;
  1. Take cells with $|E|>4\ \sigma(E)$ (note σ includes pile-up noise)
  2. Add adjacent cells with $|E|>2\ \sigma(E)$
  3. Add a final layer without any $|E|/\sigma(E)$ requirement.
  4. Split clusters based on multiple local maxima.
- The LC calibration accounts of differences in the response of hadrons and EM-particles, energy of the shower falling outside of clusters, and energy deposited in dead material.
- Historically small-R jets in Run II have been primarily EMTopo jets (but this is no longer what is recommended). Large-R jets often use LCTopo inputs – these provide some cluster level calibration for forming sub-structure variables from the jet inputs.

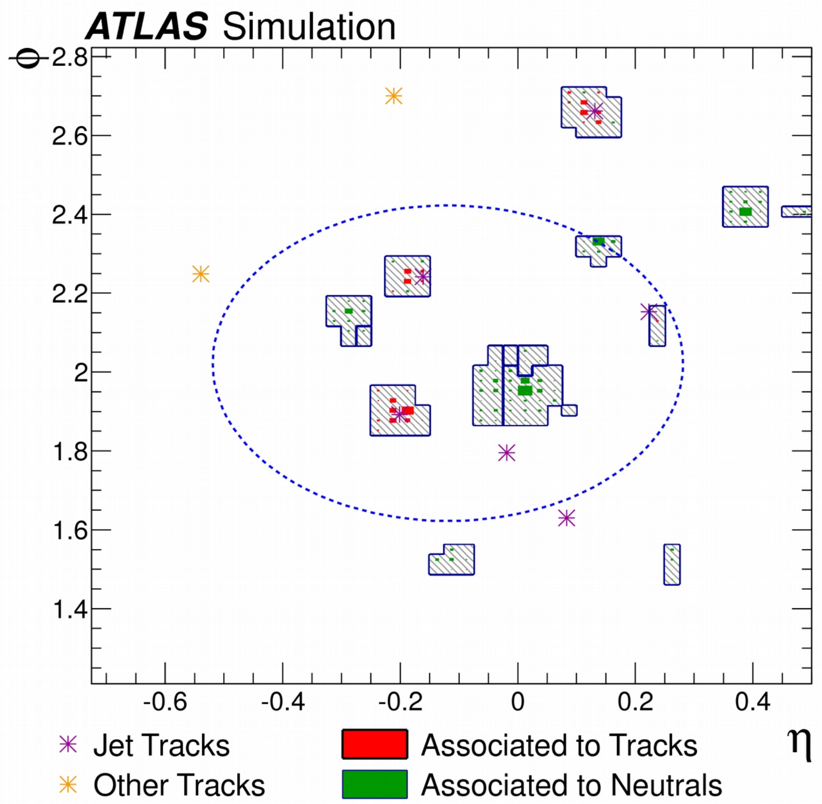
Particle Flow in ATLAS

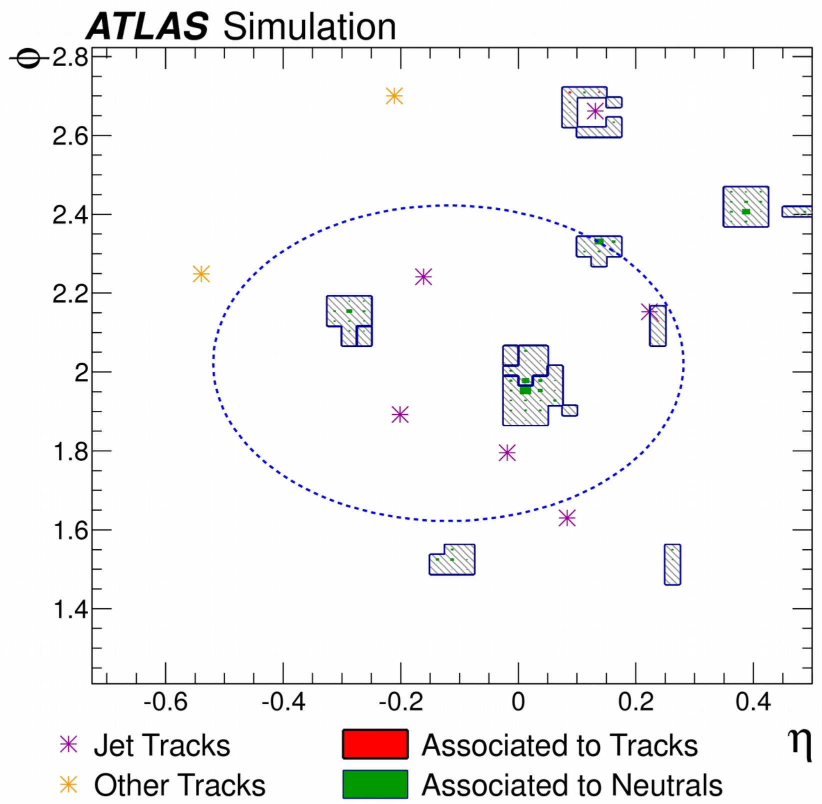
- Particle Flow aims to combine tracker and calorimeter measurements to improve performance.
- Pflow in ATLAS is primarily for jets – we do not try to do the event globally like CMS.
- The tracker advantages are; better p_T resolution at low p_T, better angular resolution, can distinguish pile-up.
- The calorimeter advantages are; better p_T resolution at high p_T, can reconstruct neutral particles.
- It is optimized to improve the resolution of low p_T jets, and maintain the resolution of high p_T jets by gradually turning off.
- For a given track we decide if the tracker will give a significantly better measurement, if so, then we subtract the cells from the calorimeter that are associated with this particle's shower (note this is sometimes imperfect).



Particle Flow Visualization

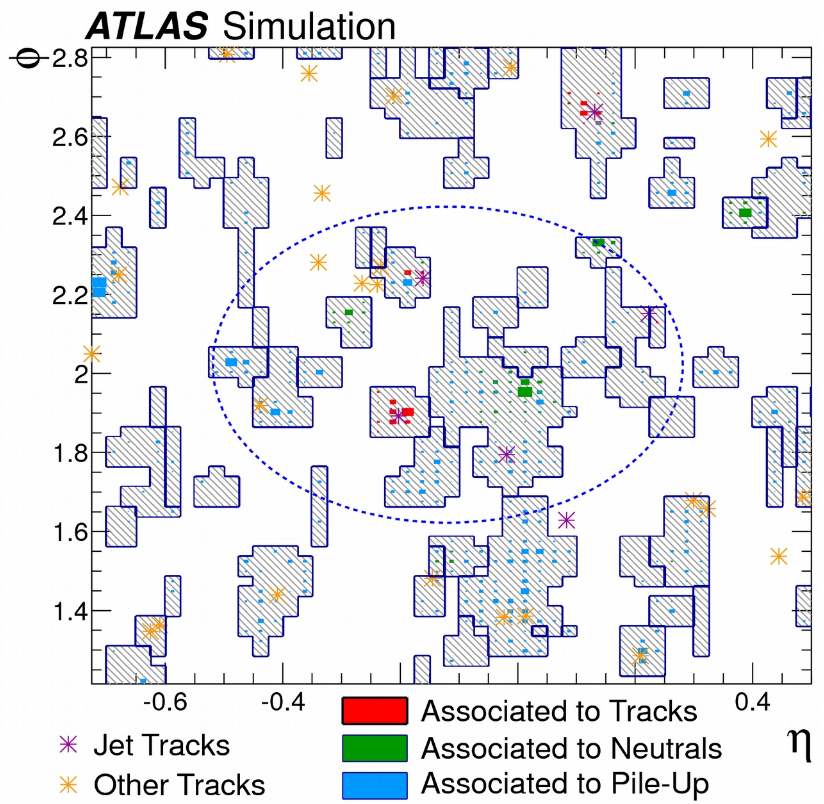
- Below you see the effect (left to right) of the particle flow subtraction in an event in the 2nd layer of the EM calorimeter with no pile-up and a 30 GeV jet.

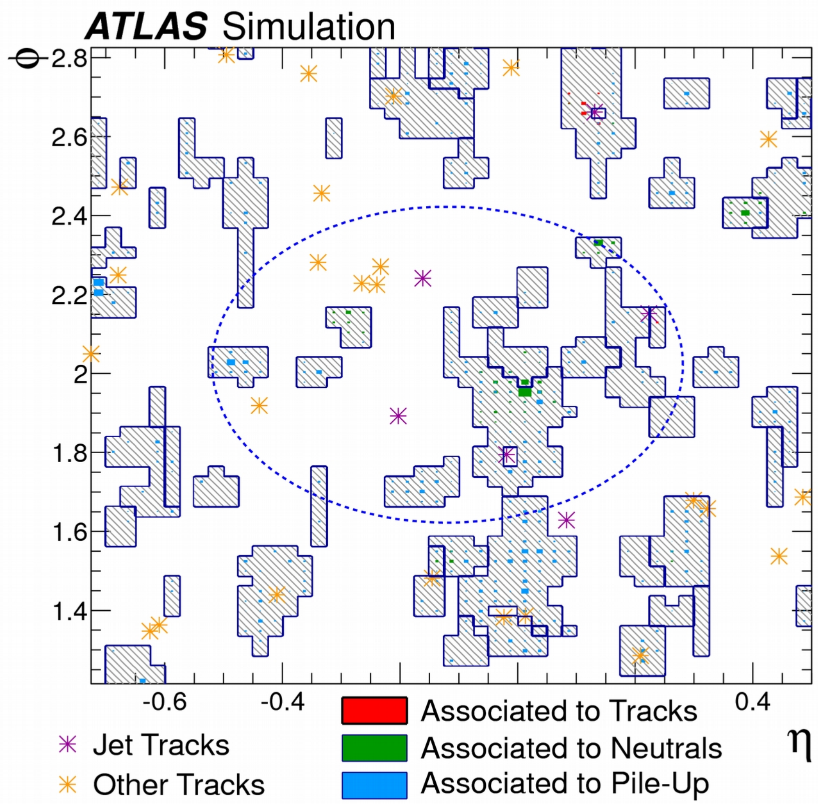




Particle Flow Visualization

- Below you see the effect (left to right) of the particle flow subtraction in an event in the 2nd layer of the EM calorimeter with 40 interactions per crossing pile-up and a 30 GeV jet.





Track-Calo-Clusters (TCCs)

- At high momentum boosted hadronic objects like $W\rightarrow q\bar{q}$, $t\rightarrow bq\bar{q}$ become very close together.
- This makes resolving their sub-structure difficult using the calorimeter due to the finite cell granularity.
- The track-calo-cluster algorithm uses the better angular resolution of the tracker to split clusters, and in the case of a single match take the track spacial coordinates.

$TCC1 = (p_T^{c1}, \eta^{t1}, \varphi^{t1}, 0)$

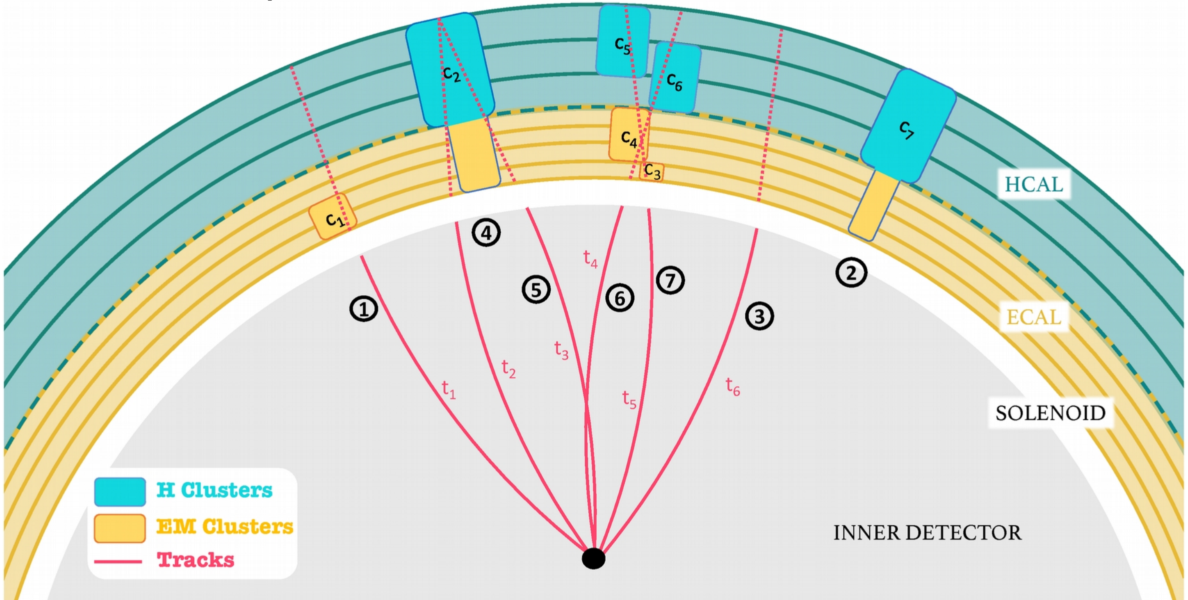
$TCC2 = (p_T^{c7}, \eta^{c7}, \varphi^{c7}, 0)$

$TCC3 = (p_T^{t6}, \eta^{t6}, \varphi^{t6}, 0)$

$TCC4 = (p_T^{c2} x (p_T^{t2}/p_T^{(t2+t3)}), \eta^{t2}, \varphi^{t2}, 0)$

$TCC5 = (p_T^{c2} x (p_T^{t3}/p_T^{(t2+t3)}), \eta^{t3}, \varphi^{t3}, 0)$

Etc...



When to use what and the EDM

- Pflow was optimized for small-R jets with particular attention to low p_T – it is the default collection for AntiKt4 jets.
- If your signal has no tracks, is out-of-time, etc then you may still have to use calorimeter (EMTopo) jets.
- For large-R jets most analyses currently use LCTopo jets – good performance across all p_T.
- For very boosted searches eg. VV resonance significant gains come from TCC as they have best tagging performance at highest p_T.
- We are working on releasing a combination of the Pflow and TCC algorithms (called Unified Flow Object (UFO)) which would provide the best large-R jet performance across all p_T.

- In the xAOD we store CaloCluster::CaloCalTopoClusters (EM+LC clusters), TrackParticle::InDetTrackParticles, PFO::JetETMissChargedParticleFlowObjects, PFO::JetETMissNeutralParticleFlowObjects.
- TCCs have their own EDM and are built in the derivation step.
- Most people don’t need to use these objects as jets can easily be built from these objects in the derivation framework.

Conclusions

- Jets are built either at Tier-0 or in the derivation framework using the Anti-k_t algorithm.
- They can be built from several different input containers; CaloCluster, PFO, TCC.
- Depending on your use case you might prefer to use a different input collection to get the best performance – Pflow for small-R, LCTopo large-R, TCC large-R at high pT, (and in the future UFO!).
- Throughout this presentation you have seen how people can improve our object reconstruction making the most of out detector, and how this can influence many analyses at once – consider joining the JetEtmiss group to further optimize jet reconstruction!
- Have fun building jets – helper functions to build jets in the derivation framework can be found in JetCommon.py and ExtendedJetCommon.py