Truth Algorithms - DBD status and Updates
`TrueJet` and `RecoMCTruthLinker`

Mikael Berggren[1]

[1]DESY, Hamburg

High Level Reconstruction Week, DESY-HH, July 6-10, 2015

Physics ⇒ Whizard ⇒ Parton shower ⇒ hadronisation ⇒ decays ⇒ Geant ⇒ MarlinReco ⇒ Pandora ⇒ Jet clustering ⇒YOU

The `TrueJet` processor tries to connect YOU with the Physics using the true information about the event.

- The connection from Geant to You is done by the `RecoMCTruthLinker` processor, linking PFOs (and jets) to MCParticles.
- `TrueJet` takes care of the rest: How does the MCParticles connect to the hard event.

Physics ⇒ .... ⇒ Geant ⇒ MarlinReco ⇒ Pandora ⇒ Jet clustering ⇒YOU

From ReconstructedParticle to MCParticle and back :

`RecoMCTruthLinker`

- There is a gap between MCParticles an ReconstructedParticles:
  - From Geant, the MCParticle creating every SimHit is known.
  - In digitisation and further reconstruction, SimHits are input, but the connection to the true particle creating the hit is not carried on.
  - `RecoMCTruthLinker` takes care of re-establishing this link.
- Input is the relations SimTrackerHit/SimCaloHit ↔ MCParticle, TrackerHit/CaloHit ↔ Track/Cluster and Track/Cluster ↔ ReconstructedParticle.
- Output is navigators between ReconstructedParticle/Track/Cluster and MCParticle.

Physics $\Rightarrow$ .... $\Rightarrow$ Geant $\Rightarrow$ MarlinReco $\Rightarrow$ Pandora $\Rightarrow$ Jet clustering $\Rightarrow$YOU

From ReconstructedParticle to MCParticle and back : `RecoMCTruthLinker`

- There is a gap between MCParticles an ReconstructedParticles:
  - From Geant, the MCParticle creating every SimHit is known.
  - In digitisation and further reconstruction, SimHits are input, but the connection to the true particle creating the hit is not carried on.
  - `RecoMCTruthLinker` takes care of re-establishing this link.
- Input is the relations SimTrackerHit/SimCaloHit $\leftrightarrow$ MCParticle, TrackerHit/CaloHit $\leftrightarrow$ Track/Cluster and Track/Cluster $\leftrightarrow$ ReconstructedParticle.
- Output is navigators between ReconstructedParticle/Track/Cluster and MCParticle.

Physics ⇒ Whizard ⇒ Parton shower ⇒ hadronisation ⇒ decays ⇒ Geant ⇒ ... ⇒YOU

From MCParticles to Physics and back: `TrueJet`

- To link further back, `TrueJet` joins hadrons from the final colour singlets to di-jets.
- The di-jet is split into two jets, connected to the final quarks.
- It follows the decay-chain of the primary hadrons, and assigns each of them to the jet of it's parent.
- The process continues from generated to simulated particles.
- Then the final quark is followed back through the parton-shower.
- Ultimately, the initial colour singlet is found.

The initial colour singlet is the closest one gets to the initial physics (W,Z,h,...).

RecoMCTruthLinker: DBD

`RecoMCTruthLinker` has been around in iLCSoft for a long time. In the DBD version:

- RecoMCTruthLink is uni-directional in the definition of the weight.
- It has no info about what true particles contribute to the cluster of a charged PFO.
- For neutrals, it only contains the link to the true particle that contributes the most to a cluster.
- ClusterMCTruthLink is also uni-directional, but does link all contributing true particles, and does so independently of the charge of the PFO that the cluster is attached to. However, it was not created in the mass-production for the DBD (?)
- TrackMCTruthLink and MCTruthTrackLink is bi-directional in the weight definition.

In addition, `RecoMCTruthLinker` decides what should go to the skimmed MCParticle collection.

RecoMCTruthLinker: DBD

`RecoMCTruthLinker` has been around in iLCSoft for a long time. In the DBD version:

- RecoMCTruthLink is uni-directional in the definition of the weight.
- It has no info about what true particles contribute to the cluster of a charged PFO.
- For neutrals, it only contains the link to the true particle that contributes the most to a cluster.
- ClusterMCTruthLink is also uni-directional, but does link all contributing true particles, and does so independently of the charge of the PFO that the cluster is attached to. However, it was not created in the mass-production for the DBD (?)
- TrackMCTruthLink and MCTruthTrackLink is bi-directional in the weight definition.

In addition, `RecoMCTruthLinker` decides what should go to the skimmed MCParticle collection.

RecoMCTruthLinker: DBD

`RecoMCTruthLinker` has been around in iLCSoft for a long time. In the DBD version:

- RecoMCTruthLink is uni-directional in the definition of the weight.
- It has no info about what true particles contribute to the cluster of a charged PFO.
- For neutrals, it only contains the link to the true particle that contributes the most to a cluster.
- ClusterMCTruthLink is also uni-directional, but does link all contributing true particles, and does so independently of the charge of the PFO that the cluster is attached to. However, it was not created in the mass-production for the DBD (?)
- TrackMCTruthLink and MCTruthTrackLink is bi-directional in the weight definition.

In addition, `RecoMCTruthLinker` decides what should go to the skimmed MCParticle collection.

RecoMCTruthLinker: DBD

`RecoMCTruthLinker` has been around in iLCSoft for a long time. In the DBD version:

- RecoMCTruthLink is uni-directional in the definition of the weight.
- It has no info about what true particles contribute to the cluster of a charged PFO.
- For neutrals, it only contains the link to the true particle that contributes the most to a cluster.
- ClusterMCTruthLink is also uni-directional, but does link all contributing true particles, and does so independently of the charge of the PFO that the cluster is attached to. However, it was not created in the mass-production for the DBD (?)
- TrackMCTruthLink and MCTruthTrackLink is bi-directional in the weight definition.

In addition, `RecoMCTruthLinker` decides what should go to the skimmed MCParticle collection.

RecoMCTruthLinker: Updates

The "new" `RecoMCTruthLinker` extends the capabilities:

- RecoMCTruthLink is supplemented with MCTruthRecoLink to make it bi-directional in weight.
- Optionally, the weight can be redefined to contain weights to and from both clusters and tracks. (This feature is in the DBD version, but not used).
- For neutrals, it links all true particles that contributes.
- ClusterMCTruthLink is supplemented with MCTruthClusterLink to make it bi-directional in weight.
- TrackMCTruthLink and MCTruthTrackLink is bi-directional in the weight definition, as before, but with a different definition of true-to-seen weight (to be discussed). Possible issue with TrackerHitRelations before and after reconstruction ?!

RecoMCTruthLinker: Updates

The "new" `RecoMCTruthLinker` extends the capabilities:

- RecoMCTruthLink is supplemented with MCTruthRecoLink to make it bi-directional in weight.
- Optionally, the weight can be redefined to contain weights to and from both clusters and tracks. (This feature is in the DBD version, but not used).
- For neutrals, it links all true particles that contributes.
- ClusterMCTruthLink is supplemented with MCTruthClusterLink to make it bi-directional in weight.
- TrackMCTruthLink and MCTruthTrackLink is bi-directional in the weight definition, as before, but with a different definition of true-to-seen weight (to be discussed). Possible issue with TrackerHitRelations before and after reconstruction ?!

RecoMCTruthLinker: Updates

The "new" `RecoMCTruthLinker` extends the capabilities:

- RecoMCTruthLink is supplemented with MCTruthRecoLink to make it bi-directional in weight.
- Optionally, the weight can be redefined to contain weights to and from both clusters and tracks. (This feature is in the DBD version, but not used).
- For neutrals, it links all true particles that contributes.
- ClusterMCTruthLink is supplemented with MCTruthClusterLink to make it bi-directional in weight.
- TrackMCTruthLink and MCTruthTrackLink is bi-directional in the weight definition, as before, but with a different definition of true-to-seen weight (to be discussed). Possible issue with TrackerHitRelations before and after reconstruction ?!

RecoMCTruthLinker: Updates

The "new" `RecoMCTruthLinker` extends the capabilities:

- RecoMCTruthLink is supplemented with MCTruthRecoLink to make it bi-directional in weight.
- Optionally, the weight can be redefined to contain weights to and from both clusters and tracks. (This feature is in the DBD version, but not used).
- For neutrals, it links all true particles that contributes.
- ClusterMCTruthLink is supplemented with MCTruthClusterLink to make it bi-directional in weight.
- TrackMCTruthLink and MCTruthTrackLink is bi-directional in the weight definition, as before, but with a different definition of true-to-seen weight (to be discussed). Possible issue with TrackerHitRelations before and after reconstruction ?!

RecoMCTruthLinker: Updates

The "new" `RecoMCTruthLinker` extends the capabilities:

- RecoMCTruthLink is supplemented with MCTruthRecoLink to make it bi-directional in weight.
- Optionally, the weight can be redefined to contain weights to and from both clusters and tracks. (This feature is in the DBD version, but not u
- For neutr
- ClusterM usterLink to make it bi-directional in weight.
- TrackMCTruthLink and MCTruthTrackLink is bi-directional in the weight definition, as before, but with a different definition of true-to-seen weight (to be discussed). Possible issue with TrackerHitRelations before and after reconstruction ?!

The aim is that the linking should be complete in both directions, ie. not just right in 98 % of the cases, but always !

RecoMCTruthLinker: Linking clusters

- The idea: Cluster $\leftrightarrow$ All particles hitting the calorimeters, and that contribute with at least one calo-hit to the cluster.
- The weights are
  - In one direction: "The fraction of all calorimetric energy this true particle created that it gave to the considered cluster"
  - In the other direction: "The fraction of the energy of this cluster that came from the considered true particle"
- The part "all particles hitting the calorimeter" is tricky:
  - Sometimes, a calo-hit comes from an MCParticle created inside the calorimeter: Not what was indented with the calohit $\leftrightarrow$ MCParticle navigator !
  - Back-scatters: Do they end up in the same cluster they came from ?
- This is (hopefully) solved in `RecoMCTruthLinker`.
- Output mapping calohit $\rightarrow$ MCParticle $\Rightarrow$ Create a TrueShower collection ? Stand-alone code $\exists$ ....

RecoMCTruthLinker: Linking clusters

- The idea: Cluster $\leftrightarrow$ All particles hitting the calorimeters, and that contribute with at least one calo-hit to the cluster.
- The weights are
  - In one direction: "The fraction of all calorimetric energy this true particle created that it gave to the considered cluster"
  - In the other direction: "The fraction of the energy of this cluster that came from the considered true particle"
- The part "all particles hitting the calorimeter" is tricky:
  - Sometimes, a calo-hit comes from an MCParticle created inside the calorimeter: Not what was indented with the calohit $\leftrightarrow$ MCParticle navigator !
  - Back-scatters: Do they end up in the same cluster they came from ?
- This is (hopefully) solved in `RecoMCTruthLinker`.
- Output mapping calohit $\rightarrow$ MCParticle $\Rightarrow$ Create a TrueShower collection ? Stand-alone code $\exists$ ....

RecoMCTruthLinker: Linking clusters

- The idea: Cluster $\leftrightarrow$ All particles hitting the calorimeters, and that contribute with at least one calo-hit to the cluster.
- The weights are
  - In one direction: "The fraction of all calorimetric energy this true particle created that it gave to the considered cluster"
  - In the other direction: "The fraction of the energy of this cluster that came from the considered true particle"
- The part "all particles hitting the calorimeter" is tricky:
  - Sometimes, a calo-hit comes from an MCParticle created inside the calorimeter: Not what was indented with the calohit $\leftrightarrow$ MCParticle navigator !
  - Back-scatters: Do they end up in the same cluster they came from ?
- This is (hopefully) solved in `RecoMCTruthLinker`.
- Output mapping calohit $\rightarrow$ MCParticle $\Rightarrow$ Create a TrueShower collection ? Stand-alone code $\exists$ ....

TrueJet: Decoding MCParticles

Idea: Since the history is created by Pythia: Re-create the Pythia arrays `p` and `k` from the MCParticle collection.

- Fix parent-child relations:
  1. If the true particle is decayed in the generator, check if any of the children is created in simulation. If so, E and p will be inconsistent.
     $\Rightarrow$
     - Promote parent to stable
     - Mark **all** children as created in simulation.
  2. A CMShower should have two parents - sometimes not the case. Fix that.
     - A partial fixup of this issue is already in the stdhep-reader. However, sometimes (mostly in 6-lepton events) it is wrong.
  3. Determine pairing initial particles
     - Easy for quarks, tricky for leptons.
  4. $t\bar{t}$ is a mess and need special treatment.

TrueJet: Decoding MCParticles

Idea: Since the history is created by Pythia: Re-create the Pythia arrays `p` and `k` from the MCParticle collection.

- Fix parent-child relations:
  1. If the true particle is decayed in the generator, check if any of the children is created in simulation. If so, E and p will be inconsistent.
     $\Rightarrow$
     - Promote parent to stable
     - Mark **all** children as created in simulation.
  2. A CMShower should have two parents - sometimes not the case. Fix that.
     - A partial fixup of this issue is already in the stdhep-reader. However, sometimes (mostly in 6-lepton events) it is wrong.
  3. Determine pairing initial particles
     - Easy for quarks, tricky for leptons.
  4. $t\bar{t}$ is a mess and need special treatment.

TrueJet: Decoding MCParticles

Idea: Since the history is created by Pythia: Re-create the Pythia arrays `p` and `k` from the MCParticle collection.

- Fix parent-child relations:
  1. If the true particle is decayed in the generator, check if any of the children is created in simulation. If so, E and p will be inconsistent.
     $\Rightarrow$
     - Promote parent to stable
     - Mark **all** children as created in simulation.
  2. A CMShower should have two parents - sometimes not the case. Fix that.
     - A partial fixup of this issue is already in the stdhep-reader. However, sometimes (mostly in 6-lepton events) it is wrong.
  3. Determine pairing initial particles
     - Easy for quarks, tricky for leptons.
  4. $t\bar{t}$ is a mess and need special treatment.

TrueJet: Decoding MCParticles

Idea: Since the history is created by Pythia: Re-create the Pythia arrays `p` and `k` from the MCParticle collection.

- Fix parent-child relations:
  1. If the true particle is decayed in the generator, check if any of the children is created in simulation. If so, E and p will be inconsistent.
     $\Rightarrow$
     - Promote parent to stable
     - Mark **all** children as created in simulation.
  2. A CMShower should have two parents - sometimes not the case. Fix that.
     - A partial fixup of this issue is already in the stdhep-reader. However, sometimes (mostly in 6-lepton events) it is wrong.
  3. Determine pairing initial particles
     - Easy for quarks, tricky for leptons.
  4. $t\bar{t}$ is a mess and need special treatment.

TrueJet: Decoding MCParticles

Idea: Since the history is created by Pythia: Re-create the Pythia arrays `p` and `k` from the MCParticle collection.

- Fix parent-child relations:
  1. If the true particle is decayed in the generator, check if any of the children is created in simulation. If so, E and p will be inconsistent.
     $\Rightarrow$
     - Promote parent to stable
     - Mark **all** children as created in simulation.
  2. A CMShower should have two parents - sometimes not the case. Fix that.
     - A partial fixup of this issue is already in the stdhep-reader. However, sometimes (mostly in 6-lepton events) it is wrong.
  3. Determine pairing initial particles
     - Easy for quarks, tricky for leptons.
  4. $t\bar{t}$ is a mess and need special treatment.

TrueJet: Assigning jets

- Find hard leptons, if any and assign each one, and their decay-products and any FSR, to a jet.
- Assign the ISR photons to one jet each.
- Find "clusters" - two quarks joined together into a bound state during the PS. Assign jets to the the decay products.
  - Normally: cluster $\rightarrow$ one hadron. But they are created by two quarks $\Rightarrow$ two jets assigned - one will often be empty !
- Find strings - easy. Their descendants are hadrons, their first and last parents are final quarks.
- For clusters and strings: back-track to the initial hard system.
  - Following the quarks - ignore the gluons.
  - If a final quark comes from a gluon-splitting $\Rightarrow$ backtrack the gluon, but stop assigning the parents to jets. Note jet which jet radiated the gluon.
- During the back-tracking, note if inner beamstrahlung occurred.
  - Add this photon to the jet that its parent quark gives rise to.

TrueJet: Assigning jets

- Find hard leptons, if any and assign each one, and their decay-products and any FSR, to a jet.
- Assign the ISR photons to one jet each.
- Find "clusters" - two quarks joined together into a bound state during the PS. Assign jets to the the decay products.
  - Normally: cluster $\rightarrow$ one hadron. But they are created by two quarks $\Rightarrow$ two jets assigned - one will often be empty !
- Find strings - easy. Their descendants are hadrons, their first and last parents are final quarks.
- For clusters and strings: back-track to the initial hard system.
  - Following the quarks - ignore the gluons.
  - If a final quark comes from a gluon-splitting $\Rightarrow$ backtrack the gluon, but stop assigning the parents to jets. Note jet which jet radiated the gluon.
- During the back-tracking, note if inner beamstrahlung occurred.
  - Add this photon to the jet that its parent quark gives rise to.

TrueJet: Assigning jets

- Find hard leptons, if any and assign each one, and their decay-products and any FSR, to a jet.
- Assign the ISR photons to one jet each.
- Find "clusters" - two quarks joined together into a bound state during the PS. Assign jets to the the decay products.
  - Normally: cluster $\rightarrow$ one hadron. But they are created by two quarks $\Rightarrow$ two jets assigned - one will often be empty !
- Find strings - easy. Their descendants are hadrons, their first and last parents are final quarks.
- For clusters and strings: back-track to the initial hard system.
  - Following the quarks - ignore the gluons.
  - If a final quark comes from a gluon-splitting $\Rightarrow$ backtrack the gluon, but stop assigning the parents to jets. Note jet which jet radiated the gluon.
- During the back-tracking, note if inner beamstrahlung occurred.
  - Add this photon to the jet that its parent quark gives rise to.

TrueJet: Assigning jets

- Find hard leptons, if any and assign each one, and their decay-products and any FSR, to a jet.
- Assign the ISR photons to one jet each.
- Find "clusters" - two quarks joined together into a bound state during the PS. Assign jets to the the decay products.
  - Normally: cluster $\rightarrow$ one hadron. But they are created by two quarks $\Rightarrow$ two jets assigned - one will often be empty !
- Find strings - easy. Their descendants are hadrons, their first and last parents are final quarks.
- For clusters and strings: back-track to the initial hard system.
  - Following the quarks - ignore the gluons.
  - If a final quark comes from a gluon-splitting $\Rightarrow$ backtrack the gluon, but stop assigning the parents to jets. Note jet which jet radiated the gluon.
- During the back-tracking, note if inner beamstrahlung occurred.
  - Add this photon to the jet that its parent quark gives rise to.

TrueJet: Assigning jets

- Find hard leptons, if any and assign each one, and their decay-products and any FSR, to a jet.
- Assign the ISR photons to one jet each.
- Find "clusters" - two quarks joined together into a bound state during the PS. Assign jets to the the decay products.
  - Normally: cluster $\rightarrow$ one hadron. But they are created by two quarks $\Rightarrow$ two jets assigned - one will often be empty !
- Find strings - easy. Their descendants are hadrons, their first and last parents are final quarks.
- For clusters and strings: back-track to the initial hard system.
  - Following the quarks - ignore the gluons.
  - If a final quark comes from a gluon-splitting $\Rightarrow$ backtrack the gluon, but stop assigning the parents to jets. Note jet which jet radiated the gluon.
- During the back-tracking, note if inner beamstrahlung occurred.
  - Add this photon to the jet that its parent quark gives rise to.

TrueJet: Assigning jets

- Find hard leptons, if any and assign each one, and their decay-products and any FSR, to a jet.
- Assign the ISR photons to one jet each.
- Find "clusters" - two quarks joined together into a bound state during the PS. Assign jets to the the decay products.
  - Normally: cluster $\rightarrow$ one hadron. But they are created by two quarks $\Rightarrow$ two jets assigned - one will often be empty !
- Find strings - easy. Their descendants are hadrons, their first and last parents are final quarks.
- For clusters and strings: back-track to the initial hard system.
  - Following the quarks - ignore the gluons.
  - If a final quark comes from a gluon-splitting $\Rightarrow$ backtrack the gluon, but stop assigning the parents to jets. Note jet which jet radiated the gluon.
- During the back-tracking, note if inner beamstrahlung occurred.
  - Add this photon to the jet that its parent quark gives rise to.

TrueJet: Assigning jets

- For clusters and strings, assign the first generation hadrons to a jet induced by the final quark to which it is closest to in angle.
  - There is always two, and only two, quarks as immediate parents.
- Follow the decay-chain of each hadron, assigning any product to the same jet.
  - NB: Done to the end of the MCParticle parent-child chain. $\Rightarrow$ Both generator and simulator particles assigned to jets.
- All particles (post-PS) that are leftover are from overlaid events, and are grouped together in a single jet.

TrueJet: Assigning jets

- For clusters and strings, assign the first generation hadrons to a jet induced by the final quark to which it is closest to in angle.
  - There is always two, and only two, quarks as immediate parents.
- Follow the decay-chain of each hadron, assigning any product to the same jet.
  - NB: Done to the end of the MCParticle parent-child chain. $\Rightarrow$ Both generator and simulator particles assigned to jets.
- All particles (post-PS) that are leftover are from overlaid events, and are grouped together in a single jet.

TrueJet: Assigning jets

- For clusters and strings, assign the first generation hadrons to a jet induced by the final quark to which it is closest to in angle.
  - There is always two, and only two, quarks as immediate parents.
- Follow the decay-chain of each hadron, assigning any product to the same jet.
  - NB: Done to the end of the MCParticle parent-child chain. $\Rightarrow$ Both generator and simulator particles assigned to jets.
- All particles (post-PS) that are leftover are from overlaid events, and are grouped together in a single jet.

TrueJet: Output Collections

`TrueJet` is a normal Marlin processor. The only parameters are the in/output collection names.

- Jets and ancestors
  - TrueJets : (RecoParticles). `getParticles` gives all PFOs in the jet, `getParticleIDs` returns the type as
    1. string
    2. lepton
    3. cluster
    4. ISR
    5. overlay
  - FinalColorNeutrals : (RecoParticles). `getEnergy` etc. gives true values for the dijet from the final quarks. `getParticles` gives the `TrueJets` this colour-neutral object gives rise to (always two).
    - For the beam jet it is the sum of the weight 1 MCParticles) .
  - InitialColorNeutrals: (RecoParticles)
    - If there are no gluon-induced jets: same a above.
    - If there are, it is the true values of all jets (gluon and quark) coming from the same initial quark pair.

TrueJet: Output Collections

- Relations:
  - TrueJetPFOLink : link from PFO:s to true jets
  - TrueJetMCParticleLink : link from jets to MCParticles. Meaning of the weight:
    - 0 - in the parton-shower.
    - 1 - stable to be used for eg. total E.
    - 2 - un-stable.

    This code is not the same as `getGeneratorStatus`, see above. Summing MCPs with weight == 1 should always be correct - no double-counting or lost energy. Anything else is a bug !
  - FinalElementonLink : points from FinalColorNeutrals a to the elementon (an MCParticle) at the end of the parton-shower that gives rise to the jet
  - InitialElementonLink : points from an InitialColorNeutrals to the parton (an MCParticle) at the beginning of the parton-shower that gives rise to the jet.
  - FinalColorNeutralLink: link from TrueJet to the final colour-neutral object it comes from.

TrueJet: Usage

- To create, just do as any Marlin processor - compile, add to MARLIN_DLL, add the processor description and call in the xml.
- To use the information in your processor, there is a helper class - `TrueJet_Parser`.
  - Let your processor inherit `TrueJet_Parser`. In the header:

```
.
.
.
#include "TrueJet_Parser.h"
.
.
.
class My_processor : public Processor , public TrueJet_Parser {
 public:
    virtual Processor*  newProcessor() { return new My_processor ; }
.
.
.
    std::string get_recoMCTruthLink(){ return _recoMCTruthLink  ; } ;
.
.
```

TrueJet: Usage

- Then ...
  - In the ctor of My_processor, cut'n'paste calls to `registerInputCollection` for all the output collections from `TrueJet` - see README.
  - Then in My_processor::processEvent,

```
TrueJet_Parser* tj= this ;tj->getall(evt);
```

  - Once done, add

```
if ( tj ) delall();}
```

    at the end of My_processor::processEvent, to avoid leaks.
- There is an example processor - `Use_TrueJet` - that contains calls to all methods of `TrueJet_Parser`.

Summary

- The new `RecoMCTruthLinker` will be useful for studying particle-flow in detail.
- It is a pre-requisite for `TrueJet` to be maximally useful.
- `TrueJet` will be useful for disentangling effects of jet clustering from particle flow, from combinatorics, for detector effects.
- It is also useful for testing and developing overlay-removal, jet-clustering and secondary vertex methods.

Summary

- The new `RecoMCTruthLinker` will be useful for studying particle-flow in detail.
- It is a pre-requisite for `TrueJet` to be maximally useful.
- `TrueJet` will be useful for disentangling effects of jet clustering from particle flow, from combinatorics, for detector effects.
- It is also useful for testing and developing overlay-removal, jet-clustering and secondary vertex methods.

Outlook for the week

- For `TrueJet`:
  - All Whizard generated background event-types have been tested and works - except $\gamma\gamma$.
  - Not tested on $\gamma\gamma \rightarrow$ *hadrons* from Pythia.
  - Right now, it does not work for 8-fermion samples from Physim, nor Whizard-samples with higgs in the f.s.
- Hopefully, all these issues can be addressed during the week.
- For `RecoMCTruthLinker`:
  - Further checks of the new version needed.
  - The issue with the TrackHitRelations on RECO files need to be clarified.
  - Would a TrueShower collection be generally useful ?
- Make sure that `TrueJet` and `RecoMCTruthLinker` does work together.

Outlook for the week

- For `TrueJet`:
  - All Whizard generated background event-types have been tested and works - except $\gamma\gamma$.
  - Not tested on $\gamma\gamma \rightarrow$ *hadrons* from Pythia.
  - Right now, it does not work for 8-fermion samples from Physim, nor Whizard-samples with higgs in the f.s.
- Hopefully, all these issues can be addressed during the week.
- For `RecoMCTruthLinker`:
  - Further checks of the new version needed.
  - The issue with the TrackHitRelations on RECO files need to be clarified.
  - Would a TrueShower collection be generally useful ?
- Make sure that `TrueJet` and `RecoMCTruthLinker` does work together.

Outlook for the week

- For `TrueJet`:
  - All Whizard generated background event-types have been tested and works - except $\gamma\gamma$.
  - Not tested on $\gamma\gamma \rightarrow$ *hadrons* from Pythia.
  - Right now, it does not work for 8-fermion samples from Physim, nor Whizard-samples with higgs in the f.s.
- Hopefully, all these issues can be addressed during the week.
- For `RecoMCTruthLinker`:
  - Further checks of the new version needed.
  - The issue with the TrackHitRelations on RECO files need to be clarified.
  - Would a TrueShower collection be generally useful ?
- Make sure that `TrueJet` and `RecoMCTruthLinker` does work together.

Outlook for the week

- For `TrueJet`:
  - All Whizard generated background event-types have been tested and works - except $\gamma\gamma$.
  - Not tested on $\gamma\gamma \rightarrow$ *hadrons* from Pythia.
  - Right now, it does not work for 8-fermion samples from Physim, nor Whizard-samples with higgs in the f.s.
- Hopefully, all these issues can be addressed during the week.
- For `RecoMCTruthLinker`:
  - Further checks of the new version needed.
  - The issue with the TrackHitRelations on RECO files need to be clarified.
  - Would a TrueShower collection be generally useful ?
- Make sure that `TrueJet` and `RecoMCTruthLinker` does work together.

Outlook for the week

- For `TrueJet`:
  - All Whizard generated background event-types have been tested and works - except $\gamma\gamma$.
  - Not tested on $\gamma\gamma \rightarrow$ *hadrons* from Pythia.
  - Right now, it does not work for 8-fermion samples from Physim, nor Whiz
- Hopefully … e week.
- For `Recc`
  - Furth…
  - The issue with the TrackHitRelations on RECO files need to be clarified.
  - Would a TrueShower collection be generally useful ?
- Make sure that `TrueJet` and `RecoMCTruthLinker` does work together.

Most importantly

- Integrate these tools into your studies.
- Evaluate if they are useful for you !