# Truth information : `RecoMCTruthLinker` and `TrueJet`

Mikael Berggren[1]

[1]DESY, Hamburg

ILD Benchmarking Days, UTA, Oct. , 2018

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

The `TrueJet` and `RecoMcTruthLinker` processors 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.

# RecoMCTruthLinker

The "new" (wrt. DBD) `RecoMCTruthLinker`:

- 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.

# RecoMCTruthLinker

The "new" (wrt. DBD) `RecoMCTruthLinker`:

- 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.

# RecoMCTruthLinker

The "new" (wrt. DBD) `RecoMCTruthLinker`:

- 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.

# RecoMCTruthLinker

The "new" (wrt. DBD) `RecoMCTruthLinker`:

- 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.

# RecoMCTruthLinker

The "new" (wrt. DBD) `RecoMCTruthLinker`:

- RecoMCTruthLink is supplemented with MCTruthRecoLink to make it bi-directional in weight.
- Optionall … ghts to and from both … BD version, but not u …
- 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.

> 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: $E_{calo}$(from MCP *in this cluster*)/$E_{calo}$(from MCP)
  - In the other: $E_{calo}$(In cluster *from this MCP*)/$E_{calo}$(In cluster)
- The part "all particles hitting the calorimeter" is tricky:
  - Back-scatters: Do they end up in the same cluster they came from ?
  - The dogma is that one can figure out that a particle started in the tracker by knowing that it's mother ended there.
  - Not! Non-destructive interactions: In Geant (but never in the generator) a particle might create new particles without disappearing. In this case, one must
    1. Detect that the thing happened (easy work-around in Geant4)
    2. Use a set of hits to figure out what happened in the event

# 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: $E_{calo}$(from MCP *in this cluster*)/$E_{calo}$(from MCP)
  - In the other: $E_{calo}$(In cluster *from this MCP*)/$E_{calo}$(In cluster)
- The part "all particles hitting the calorimeter" is tricky:
  - Back-scatters: Do they end up in the same cluster they came from ?
  - The dogma is that one can figure out that a particle started in the tracker by knowing that it's mother ended there.
  - Not! Non-destructive interactions: In Geant (but never in the generator) a particle might create new particles without disappearing. In this case, one must
    1. Detect that this thing happened (and work around a bug in MOKKA)
    2. Use a set of tricks to figure out/guess if it happened in the tracker.

# 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: $E_{calo}$(from MCP *in this cluster*)/$E_{calo}$(from MCP)
  - In the other: $E_{calo}$(In cluster *from this MCP*)/$E_{calo}$(In cluster)
- The part "all particles hitting the calorimeter" is tricky:
  - Back-scatters: Do they end up in the same cluster they came from ?
  - The dogma is that one can figure out that a particle started in the tracker by knowing that it's mother ended there.
  - Not! Non-destructive interactions: In Geant (but never in the generator) a particle might create new particles without disappearing. In this case, one must
    1. Detect that this thing happened (and work around a bug in MOKKA)
    2. Use a set of tricks to figure out/guess if it happened in the tracker.

# RecoMCTruthLinker: Linking clusters

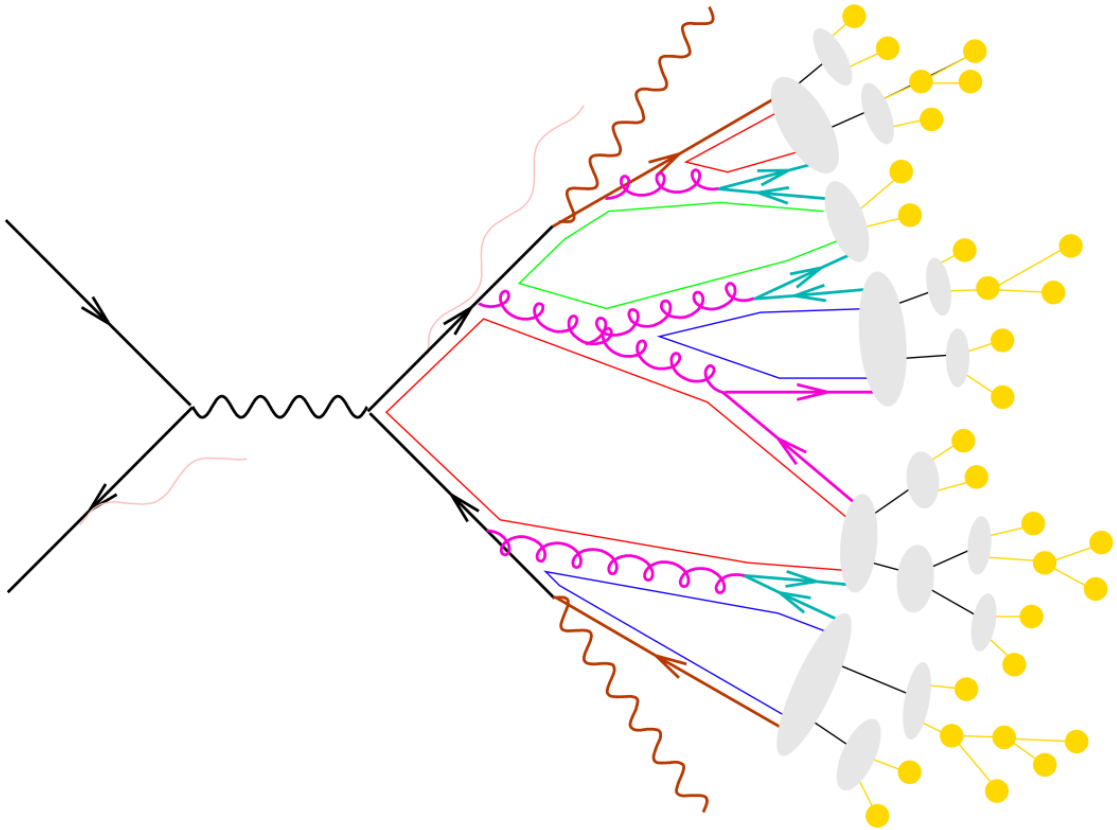
- The idea: Cluster $\leftrightarrow$ All particles hitting the calorimeters, and that contribute with at least one calo-hit to the cluster.
- The weights are
  - In one direction: $E_{calo}$(from MCP *in this cluster*)/$E_{calo}$(from MCP)
  - In the other: $E_{calo}$(In cluster *from this MCP*)/$E_{calo}$(In cluster)
- The part "all particles hitting the calorimeter" is tricky:
  - Back-scatters: Do they end up in the same cluster they came from ?
  - The dogma is that one can figure out that a particle started in the tracker by knowing that it's mother ended there.
  - Not! Non-destructive interactions: In Geant (but never in the generator) a particle might create new particles without disappearing. In this case, one must
    1. Detect that this thing happened (and work around a bug in MOKKA)
    2. Use a set of tricks to figure out/guess if it happened in the tracker.

# 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: $E_{calo}$(from MCP *in this cluster*)/$E_{calo}$(from MCP)
  - In the other: $E_{calo}$(In cluster *from this MCP*)/$E_{calo}$(In cluster)
- The part "all particles hitting the calorimeter" is tricky:
  - Back-scatters: Do they end up in the same cluster they came from ?
  - The dogma is that one can figure out that a particle started in the tracker by knowing that it's mother ended there.
  - Not! Non-destructive interactions: In Geant (but never in the generator) a particle might create new particles without disappearing. In this case, one must
    1. Detect that this thing happened (and work around a bug in MOKKA)
    2. Use a set of tricks to figure out/guess if it happened in the tracker.

# Physics ⇒ Whizard ⇒ Parton show er ⇒ hadronisation ⇒ decays ⇒ Geant *rrow* ... ⇒YOU

Pictorially:



- hard scattering
- (QED) initial/final state radiation
- partonic decays, e.g. $t \longrightarrow bW$
- parton shower evolution
- nonperturbative gluon splitting
- colour singlets
- colourless clusters
- cluster fission
- cluster $\longrightarrow$ hadrons
- hadronic decays

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

From MCParticles to Physics: `TrueJet`

- To link further back, `TrueJet` joins hadrons from the final colour neutrals 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 neutral is found.

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

# 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 bremsstrahlung 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 bremsstrahlung 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 bremsstrahlung 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 bremsstrahlung 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 bremsstrahlung 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 bremsstrahlung 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` then uses `RecoMCTruthLink` to add PFOs to the jet, on a *first seen, first assigned* basis. Because of the *many-to-many* relation between MCPs and PFOs, this should be seen as a preliminary assignment!
- Energy and momentum of the jet is the sum of the assigned PFOs, ie. the seen quantaties.

# 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` then uses `RecoMCTruthLink` to add PFOs to the jet, on a *first seen, first assigned* basis. Because of the *many-to-many* relation between MCPs and PFOs, this should be seen as a preliminary assignment!
- Energy and momentum of the jet is the sum of the assigned PFOs, ie. the seen quantaties.

# 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` then uses `RecoMCTruthLink` to add PFOs to the jet, on a *first seen, first assigned* basis. Because of the *many-to-many* relation between MCPs and PFOs, this should be seen as a preliminary assignment!
- Energy and momentum of the jet is the sum of the assigned PFOs, ie. the seen quantaties.

# 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` then uses `RecoMCTruthLink` to add PFOs to the jet, on a *first seen, first assigned* basis. Because of the *many-to-many* relation between MCPs and PFOs, this should be seen as a preliminary assignment!
- Energy and momentum of the jet is the sum of the assigned PFOs, ie. the seen quantaties.

# TrueJet: Output Collections

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

- Jets
  - TrueJets : (RecoParticles). `getParticles` gives all PFOs in the jet, and `getEnrgy` and `getMomentum` gives the sum of energy and momentum of these.
    `getParticleIDs` returns the type as
    1. string
    2. lepton
    3. cluster
    4. ISR
    5. overlay

    These are *negated* if there is no seen trace of the jet ($\nu$s, ISR down the beam-pipe, whatever)

# TrueJet: Output Collections

- Jet ancestors
  - FinalColorNeutrals : (RecoParticles). `getEnergy` etc. gives true values for the dijet from the final quarks. `getParticles` gives the `TrueJets` this colour-singlet 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. For overlay particles, 30 is added to the weight. Summing MCPs with weight == 1 should always be correct - no double-counting or lost energy. Anything else is a bug !

# TrueJet: Output Collections

- Relations, continued:
  - FinalElementonLink : points from FinalColorNeutrals to the parton (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 it comes from.
  - InitialColorNeutralLink: link from TrueJet to the initial colour-neutral it comes from. Note that there is no such thing for the “overlay”-jet: check for the jet-type ($|type| \neq 5$) before asking for this relation !

# TrueJet: Usage

- `TrueJet` is in `MarlinReco.so`, just put it in the *execute*-stanza in the xml - default parameters should be OK.
- To use the information in your processor, there is a helper class - `TrueJet_Parser` (also in `MarlinReco`)
  - 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`.

# TrueJet: What's happening?!

- `TrueJet` contains extensive reporting of what is being done, by setting `verbosity` to `DEBUG0` (highest detail) `DEBUG9` (least detail).
- However, an iLCSoft instalation done with the production CMake settings ignores debugging output $\Rightarrow$ To get detailed reporting, grab the source and re-make after modifying the CMake. Unluckily, not just `TrueJet`, but all of MarlinReco needs to be rebuilt, to avoid clashes....

# Conclusions and Outlook

- `RecoMCTruthLinker` and `TrueJet` are in `MarlinReco/Analysis`
- `.RecoMCTruthLinker` is run in standard production, while `TrueJet`, which only needs DST-input, is not.
- `TrueJet` is 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 and jet-clustering methods.
- Status:
  - All Whizard generated event-types have been tested and works - except $\gamma\gamma$ (which has, however been successfully tested at the generator output level)
  - Right now, it does not work for 8-fermion samples from Physim - all Whizard generated event-types have been tested and works.
- Another Truth-info processor in the works: `TrueClusters`, which connects un-confused calorimeter info to it's creating true particle

# Conclusions and Outlook

- `RecoMCTruthLinker` and `TrueJet` are in `MarlinReco/Analysis`
- `.RecoMCTruthLinker` is run in standard production, while `TrueJet`, which only needs DST-input, is not.
- `TrueJet` is 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 and jet-clustering methods.
- Status:
  - All Whizard generated event-types have been tested and works - except $\gamma\gamma$ (which has, however been successfully tested at the generator output level)
  - Right now, it does not work for 8-fermion samples from Physim - all Whizard generated event-types have been tested and works.
- Another Truth-info processor in the works: `TrueClusters`, which connects un-confused calorimeter info to it's creating true particle

# Conclusions and Outlook

- `RecoMCTruthLinker` and `TrueJet` are in `MarlinReco/Analysis`
- `.RecoMCTruthLinker` is run in standard production, while `TrueJet`, which only needs DST-input, is not.
- `TrueJet` is 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 and jet-clustering methods.
- Status:
  - All Whizard generated event-types have been tested and works - except $\gamma\gamma$ (which has, however been successfully tested at the generator output level)
  - Right now, it does not work for 8-fermion samples from Physim - all Whizard generated event-types have been tested and works.
- Another Truth-info processor in the works: `TrueClusters`, which connects un-confused calorimeter info to it's creating true particle

# Conclusions and Outlook

- `RecoMCTruthLinker` and `TrueJet` are in `MarlinReco/Analysis`
- . `RecoMCTruthLinker` is run in standard production, while `TrueJet`, which only needs DST-input, is not.
- `TrueJet` is 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 and jet-clustering methods.
- Status:
  - All Whizard generated event-types have been tested and works - except $\gamma\gamma$ (which has, however been successfully tested at the generator output level)
  - Right now, it does not work for 8-fermion samples from Physim - all Whizard generated event-types have been tested and works.
- Another Truth-info processor in the works: `TrueClusters`, which connects un-confused calorimeter info to it's creating true particle.