Status and use of truth tools : `MCParticle`, `RecoMCTruthLinker` and `TrueJet`

Mikael Berggren[1]

[1]DESY, Hamburg

ILD SW & Ana meeting, Feb. 3, 2021

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. Ie. `McParticles` with `PandoraPFO`.

- 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.

Note: The LHC experiments have no tools like this !

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. Ie. `McParticles` with `PandoraPFO`.

- 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.

Note: The LHC experiments have no tools like this !

RecoMCTruthLinker

`RecoMCTruthLinker` has been around in iLCSoft for a long time. Since DBD times, some ameliorations are implemented:

- RecoMCTruthLink is supplemented with MCTruthRecoLink to make it bi-directional in weight.
- The weight is defined to contain weights to and from both clusters and tracks. (This feature is in the DBD version, but not used) $\Rightarrow$ If you used these weights in your DBD analysis, it needs to be revised !
- 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

`RecoMCTruthLinker` has been around in iLCSoft for a long time. Since DBD times, some ameliorations are implemented:

- RecoMCTruthLink is supplemented with MCTruthRecoLink to make it bi-directional in weight.
- The weight is defined to contain weights to and from both clusters and tracks. (This feature is in the DBD version, but not used) $\Rightarrow$ If you used these weights in your DBD analysis, it needs to be revised !
- 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

`RecoMCTruthLinker` has been around in iLCSoft for a long time. Since DBD times, some ameliorations are implemented:

- RecoMCTruthLink is supplemented with MCTruthRecoLink to make it bi-directional in weight.
- The weight is defined to contain weights to and from both clusters and tracks. (This feature is in the DBD version, but not used) $\Rightarrow$ If you used these weights in your DBD analysis, it needs to be revised !
- 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

`RecoMCTruthLinker` has been around in iLCSoft for a long time. Since DBD times, some ameliorations are implemented:

- RecoMCTruthLink is supplemented with MCTruthRecoLink to make it bi-directional in weight.
- The weight is defined to contain weights to and from both clusters and tracks. (This feature is in the DBD version, but not used) $\Rightarrow$ If you used these weights in your DBD analysis, it needs to be revised !
- 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

`RecoMCTruthLinker` has been around in iLCSoft for a long time. Since DBD times, some ameliorations are implemented:

- RecoMCTruthLink is supplemented with MCTruthRecoLink to make it bi-directional in weight.
- The weight is defined to contain weights to and from both clusters and track[illegible] t used) ⇒ If you usec[illegible] s to be revised !

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

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

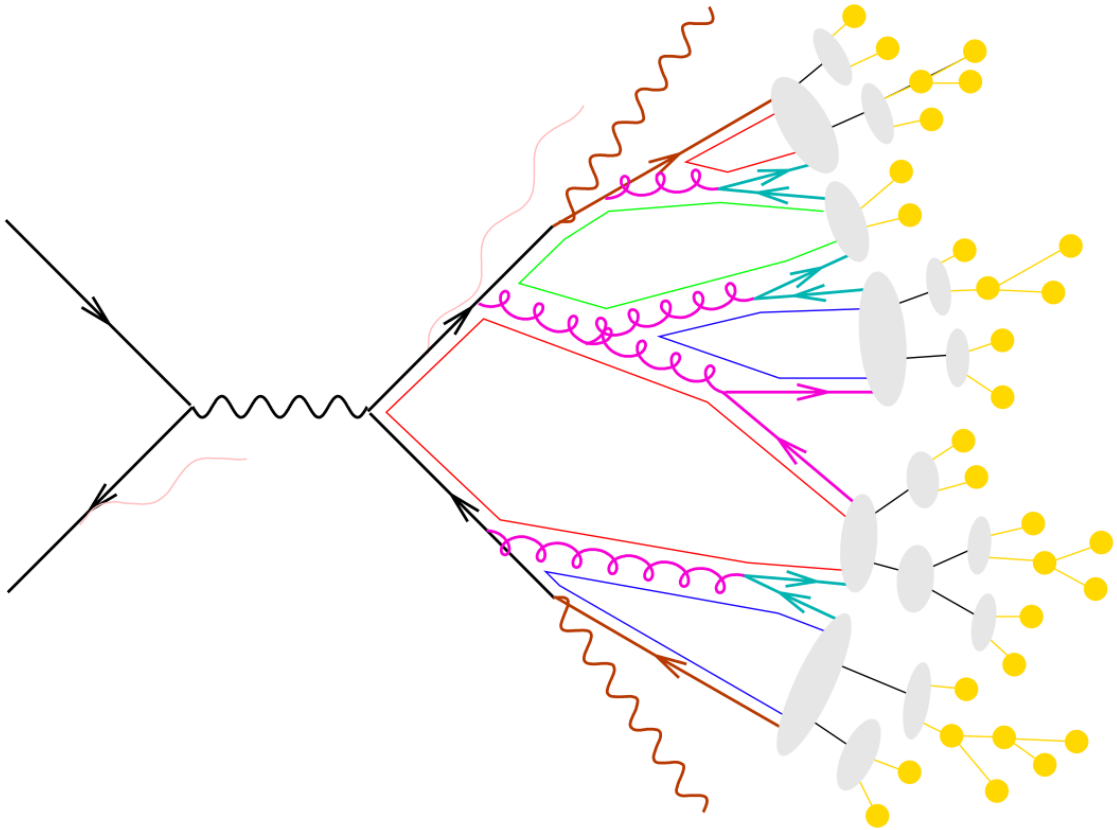
From MCParticles to Physics: `TrueJet`

- 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,...).

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

TrueJet: Assigning jets

- Find hard leptons, if any and assign each one, and their decay-products and any FSR, to a jet.
- NEW: Find any "hard" photon and assign to one jet each. "Hard" γ:s: present in the M.E. evaluation, e.g. in $H \to \gamma\gamma$.
- Assign the ISR photons to one jet each.
- Find strings (and/or "clusters") - 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.
- NEW: Find any "hard" photon and assign to one jet each. "Hard" γ:s: present in the M.E. evaluation, e.g. in $H \rightarrow \gamma\gamma$.
- Assign the ISR photons to one jet each.
- Find strings (and/or "clusters") - 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.
- NEW: Find any "hard" photon and assign to one jet each. "Hard" γ:s: present in the M.E. evaluation, e.g. in $H \rightarrow \gamma\gamma$.
- Assign the ISR photons to one jet each.
- Find strings (and/or "clusters") - 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.
- NEW: Find any “hard” photon and assign to one jet each. “Hard” γ:s: present in the M.E. evaluation, e.g. in $H \to \gamma\gamma$.
- Assign the ISR photons to one jet each.
- Find strings (and/or “clusters”) - 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.
- NEW: Find any "hard" photon and assign to one jet each. "Hard" γ:s: present in the M.E. evaluation, e.g. in $H \to \gamma\gamma$.
- Assign the ISR photons to one jet each.
- Find strings (and/or "clusters") - 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.
- NEW: Find any "hard" photon and assign to one jet each. "Hard" γ:s: present in the M.E. evaluation, e.g. in $H \rightarrow \gamma\gamma$.
- Assign the ISR photons to one jet each.
- Find strings (and/or "clusters") - 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, ie. not very sophisticated.
  - 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.
  - `RecoMCTruthLink` is finally used to assign the seen PFOs to jets (with a 'majority vote' in case a PFO has contributions from several 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, ie. not very sophisticated.
  - 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.
  - `RecoMCTruthLink` is finally used to assign the seen PFOs to jets (with a 'majority vote' in case a PFO has contributions from several 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, ie. not very sophisticated.
  - 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.
  - `RecoMCTruthLink` is finally used to assign the seen PFOs to jets (with a 'majority vote' in case a PFO has contributions from several jets)
- All particles (post-PS) that are leftover are from overlaid events, and are grouped together in a single jet.

TrueJet: Grouping jets

- A final step is to group jets with a common (boson) ancestor.
- There are two types of groupings:
  - Final colour-neutrals: These are the two jets at the end of the parton-shower.
  - Initial colour-neutrals: These are the jets stemming from the quarks the boson produced, i.e. at the beginning of the parton-shower.
    - The initial group might have more than two entries due to radiation. This will (maybe) be summed to reconstruct the initial boson.
- The initial group is formed either ...
  - ... because `Whizard` had colour-connected the two,
  - ... or because `Whizard` had made a 'resonance insertion',
  - ... or because `Pythia` had grouped them into a 'CMShower' object,
  - ... or, by default, as a consistent particle – antiparticle pair.
- The result of the grouping is documented in the output of `TrueJet`, including an indication of the initial boson.

TrueJet: Grouping jets

- A final step is to group jets with a common (boson) ancestor.
- There are two types of groupings:
  - Final colour-neutrals: These are the two jets at the end of the parton-shower.
  - Initial colour-neutrals: These are the jets stemming from the quarks the boson produced, i.e. at the beginning of the parton-shower.
    - The initial group might have more than two jets (gluon radiation!). The full group should be summed to reconstruct the initial boson
- The initial group is formed either ...
  - ... because `Whizard` had colour-connected the two,
  - ... or because `Whizard` had made a 'resonance insertion',
  - ... or because `Pythia` had grouped them into a 'CMShower' object,
  - ... or, by default, as a consistent particle – antiparticle pair.
- The result of the grouping is documented in the output of `TrueJet`, including an indication of the initial boson.

TrueJet: Grouping jets

- A final step is to group jets with a common (boson) ancestor.
- There are two types of groupings:
  - Final colour-neutrals: These are the two jets at the end of the parton-shower.
  - Initial colour-neutrals: These are the jets stemming from the quarks the boson produced, i.e. at the beginning of the parton-shower.
    - The initial group might have more than two jets (gluon radiation!). The full group should be summed to reconstruct the initial boson
- The initial group is formed either ...
  - ... because `Whizard` had colour-connected the two,
  - ... or because `Whizard` had made a 'resonance insertion',
  - ... or because `Pythia` had grouped them into a 'CMShower' object,
  - ... or, by default, as a consistent particle – antiparticle pair.
- The result of the grouping is documented in the output of `TrueJet`, including an indication of the initial boson.

TrueJet: Grouping jets

- A final step is to group jets with a common (boson) ancestor.
- There are two types of groupings:
  - Final colour-neutrals: These are the two jets at the end of the parton-shower.
  - Initial colour-neutrals: These are the jets stemming from the quarks the boson produced, i.e. at the beginning of the parton-shower.
    - The initial group might have more than two jets (gluon radiation!). The full group should be summed to reconstruct the initial boson
- The initial group is formed either ...
  - ... because `Whizard` had colour-connected the two,
  - ... or because `Whizard` had made a 'resonance insertion',
  - ... or because `Pythia` had grouped them into a 'CMShower' object,
  - ... or, by default, as a consistent particle – antiparticle pair.
- The result of the grouping is documented in the output of `TrueJet`, including an indication of the initial boson.

TrueJet: Output Info

- True Jets and the colour-neutrals.
- Navigators from/to reconstructed objects and true particles to true jets, true jets to colour-neutrals, quarks/leptons to colour-neutrals.
- Helper-class to answer specific questions exists. Use it!
- NB: The actual true jets not so important: It's the di-jets we are mostly interested in!!!

TrueJet: Output Info

- True Jets and the colour-neutrals.
- Navigators from/to reconstructed objects and true particles to true jets, true jets to colour-neutrals, quarks/leptons to colour-neutrals.
- Helper-class to answer specific questions exists. Use it!
- NB: The actual true jets not so important: It's the di-jets we are mostly interested in!!!

All the gory details:

Look in the README.

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`. Compile that too, and add to `MARLIN_DLL`. Then:
  - 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`.
- Look at the header file `TrueJet_parser.h`. All methods are explained there.

TrueJet: Breaking news

In the up-coming patch-release, there is an updated version of `TrueJet` & co.

- First of all: It works !
  - The previous version does not work on mc2020 DSTs, due to changes in the structure of the `MCParticle`.
  - The new version works both for IDR and mc2020 DSTs, except for the very mixed-up $t\bar{t}$ case (W.I.P.).
- A new jet type: hard photon (type = 6) (see above).
- Better grouping of beam-remnants: If only one - will be in a single jet-group.
- Does not require that there are any reconstructed particles. Means that it also works directly on generator output LCIO files.
- True-of-Seen (i.e. the jet properties made from the true values of all particles that were detected) has been ameliorated (avoid double-counting from e.g. decays in flight).

TrueJet: Breaking news

In the up-coming patch-release, there is an updated version of `TrueJet` & co.

- First of all: It works !
  - The previous version does not work on mc2020 DSTs, due to changes in the structure of the `MCParticle`.
  - The new version works both for IDR and mc2020 DSTs, except for the very mixed-up $t\bar{t}$ case (W.I.P.).
- A new jet type: hard photon (type = 6) (see above).
- Better grouping of beam-remnants: If only one - will be in a single jet-group.
- Does not require that there are any reconstructed particles. Means that it also works directly on generator output LCIO files.
- True-of-Seen (i.e. the jet properties made from the true values of all particles that were detected) has been ameliorated (avoid double-counting from e.g. decays in flight).

TrueJet: Breaking news

In the up-coming patch-release, there is an updated version of `TrueJet` & co.

- First of all: It works !
  - The previous version does not work on mc2020 DSTs, due to changes in the structure of the `MCParticle`.
  - The new version works both for IDR and mc2020 DSTs, except for the very mixed-up $t\bar{t}$ case (W.I.P.).
- A new jet type: hard photon (type = 6) (see above).
- Better grouping of beam-remnants: If only one - will be in a single jet-group.
- Does not require that there are any reconstructed particles. Means that it also works directly on generator output LCIO files.
- True-of-Seen (i.e. the jet properties made from the true values of all particles that were detected) has been ameliorated (avoid double-counting from e.g. decays in flight).

TrueJet: Breaking news

In the up-coming patch-release, there is an updated version of `TrueJet` & co.

- First of all: It works !
  - The previous version does not work on mc2020 DSTs, due to changes in the structure of the `MCParticle`.
  - The new version works both for IDR and mc2020 DSTs, except for the very mixed-up $t\bar{t}$ case (W.I.P.).
- A new jet type: hard photon (type = 6) (see above).
- Better grouping of beam-remnants: If only one - will be in a single jet-group.
- Does not require that there are any reconstructed particles. Means that it also works directly on generator output LCIO files.
- True-of-Seen (i.e. the jet properties made from the true values of all particles that were detected) has been ameliorated (avoid double-counting from e.g. decays in flight).

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, from detector effects.
- It is also useful for testing and developing overlay-removal and jet-clustering methods.
- Status:
  - All Whizard2.85 (mc2020) generated event-types available as DSTs have been tested and works. This means 2-, 3-, 4- and 5-fermions, dedicated higgs-samples, and $\gamma\gamma \rightarrow$ 2- or 4-fermions.

    For the latter: real γ:s fully tested, virtual ones on generator level.
  - All Whizard1.95 (DBD/IDR) generated event-types have been tested and works, except $t\bar{t}$.
  - An exception: $H \rightarrow gg$ does not work in either sample.

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, from detector effects.
- It is also useful for testing and developing overlay-removal and jet-clustering methods.
- Status:
  - All Whizard2.85 (mc2020) generated event-types available as DSTs have been tested and works. This means 2-, 3-, 4- and 5-fermions, dedicated higgs-samples, and $\gamma\gamma \rightarrow$ 2- or 4-fermions.
    For the latter: real γ:s fully tested, virtual ones on generator level.
  - All Whizard1.95 (DBD/IDR) generated event-types have been tested and works, except $t\bar{t}$.
  - An exception: $H \rightarrow gg$ does not work in either sample.

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, from detector effects.
- It is also useful for testing and developing overlay-removal and jet-clustering methods.
- Status:
  - All Whizard2.85 (mc2020) generated event-types available as DSTs have been tested and works. This means 2-, 3-, 4- and 5-fermions, dedicated higgs-samples, and $\gamma\gamma \rightarrow$ 2- or 4-fermions.

    For the latter: real γ:s fully tested, virtual ones on generator level.
  - All Whizard1.95 (DBD/IDR) generated event-types have been tested and works, except $t\bar{t}$.
  - An exception: $H \rightarrow gg$ does not work in either sample.

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, from detector effects.
- It is also useful for testing and developing overlay-removal and jet-clustering methods.
- Status:
  - All Whizard2.85 (mc2020) generated event-types available as DSTs have been tested and works. This means 2-, 3-, 4- and 5-fermions, dedicated higgs-samples, and $\gamma\gamma \rightarrow$ 2- or 4-fermions.

    For the latter: real γ:s fully tested, virtual ones on generator level.
  - All Whizard1.95 (DBD/IDR) generated event-types have been tested and works, except $t\bar{t}$.
  - An exception: $H \rightarrow gg$ does not work in either sample.