FCC-hh
analysis tutorial

Michele Selvaggi
CERN

Outline

- Analysis tutorial
- Sample production

Goal of the tutorial

- No prior knowledge required
- On a simple physics example **H → 4l**, learn how to:
  - **generate** signal and background samples with ***Pythia8, FCCSW***
  - run ***Delphes*** detector simulation
  - apply event selection with ***Heppy***
  - produce **cut-flow** tables
  - produce **flat n-tuples** with observables of interest
  - produce plots with ***ROOT***

fccsw.web.cern.ch/fccsw/tutorials/fcc-tutorials/FccFullAnalysis.html

Webpage (I)

FCCSW | Home | **Tutorials** | Latest Releases: | FCCSW v0.8pre | FCCSW v0.7 | Snapshot

Getting started with the FCC common software

Getting Started is an introduction to the FCC software. It is the recommended way to get started. It also explains how to set up the software.

Everything below documents the latest release (version 0.8pre). Before continuing please make sure that this version is installed, for the corresponding versions of fcc-edm, fcc-physics and heppy, please see the release page.

Important: for all tutorials, we assume that you are using the bash shell. If not the case, we advise you to change shell.

For your CERN AFS account, you can do that on this page:
`https://resources.web.cern.ch/resources/Manage/Linux/Settings.aspx?login=YOUR_USER_NAME` . <- insert your actual AFS user name here.

List of available tutorials

- Getting Started
- Github workflow and contribution guide
- Getting started with the production and analysis of fast-simulated events
- The FCC Software Framework
  - Full analysis example using Pythia, Delphes and Heppy
  - CMake in FCCSW
- Working locally on your laptop
  - Using Virtual Machine
  - Installing the standalone packages
- Writing Documentation
- Questions & Answers
- How to do software releases

Tutorial - generate samples

Part I: Generate and simulate Events with FCCSW

First, log into lxplus, and **install the FCC software**, using git:

```
git clone https://github.com/HEP-FCC/FCCSW.git
cd FCCSW
source ./init.sh
make -j 12
```

For this tutorial we will consider the following **physics processes**:

- p p -> H -> 4 l
- p p -> Z/gamma Z/gamma -> 4 l

Pythia can be configured to hadronize previously generated hard scattering in the form of Les Houches event files (.lhe), or generate the hard process itself and then run the parton shower and hadronization. **In either case, the FCCSW takes as input a Pythia8 configuration file** (.cmd), and does not need to know which approach was used.

For this tutorial, we are going to run Pythia8 on previously produced LHE files (with MG5_aMCatNLO). Additional Pythia8 configurations are present in `Generation/data`.

The following commands will run Pythia8 and Delphes and produce the relevant signal and background samples:

```
./run fccrun.py Sim/SimDelphesInterface/options/PythiaDelphes_config.py --inputfile=Generation/data/Pythia_pp_h_4
l.cmd --outputfile=pp_h_4l.root --nevents=1000
./run fccrun.py Sim/SimDelphesInterface/options/PythiaDelphes_config.py --inputfile=Generation/data/Pythia_pp_zgzg
_4l.cmd --outputfile=pp_zgzg_4l.root --nevents=1000
```

Tutorial - analyse samples

Part II: Analyze the output with Heppy

Heppy is a python framework suitable for analyzing the FCCSW output.

First install HEPPY:

```
git clone https://github.com/HEP-FCC/heppy.git
cd heppy
source init.sh
cd ..
```

Understand the configuration file for this **H->4l analysis**: `heppy/test/analysis_pp_hTo4l_simple_cfg.py` This is where **filters** on input collections and **event selection** are defined. The sequence is divided in two parts, a gen level analysis, and a reco level.

- The **gen level analysis** simply filters interesting leptons (`gen_leptons`) and stores pT, eta in in flat tree (`gen_tree`).

Have a look at the corresponding code in `heppy/analyzers/examples/hzz4l/HTo4lGenTreeProducer.py`.

- The **reco level analysis** first **selects isolated leptons** (`selected_muons`, `selected_electrons`), merges them into a single collection (`selected_leptons`), **builds Z candidates** (`zeds`) and finally **builds higgs candidates** (`higgses`). After that an **event selection** is applied (`selection`).

Open `heppy/analyzers/examples/hzz4l/selection.py` and understand the event selection.

Finally another flat tree is produced `HTo4lTreeProducer`. This tree contains contains all relevant information for the two reconstructed Z bosons, the Higgs, and the four associated leptons. For comparison, also the MC level counterparts of the reconstructed quantities are stored.

To summarize, when designing a new analysis, you will have to define:

- a configuration file containing the analysis sequence
- an event selection
- one or several tree producer(s) where the variables to be stored in the output tree(s) are specified
- optionally, new modules that are specific to your analysis (e.g. `LeptonicZedBuilder` here)

Tutorial - produce plots (gen)

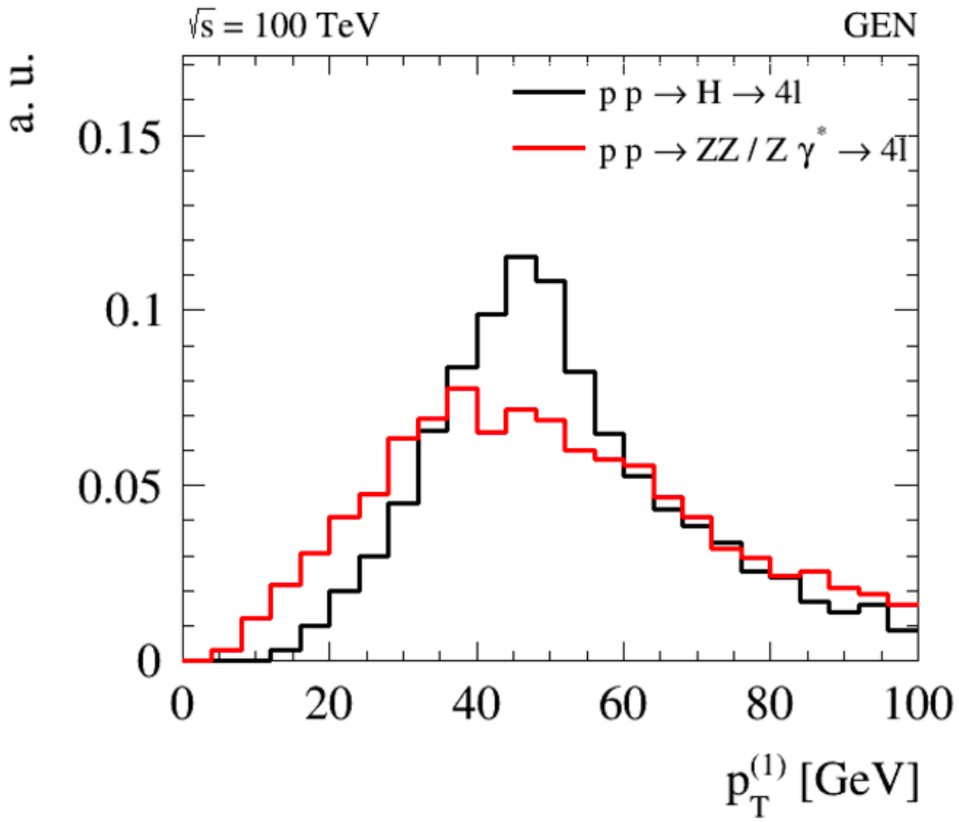
Part III: Produce plots

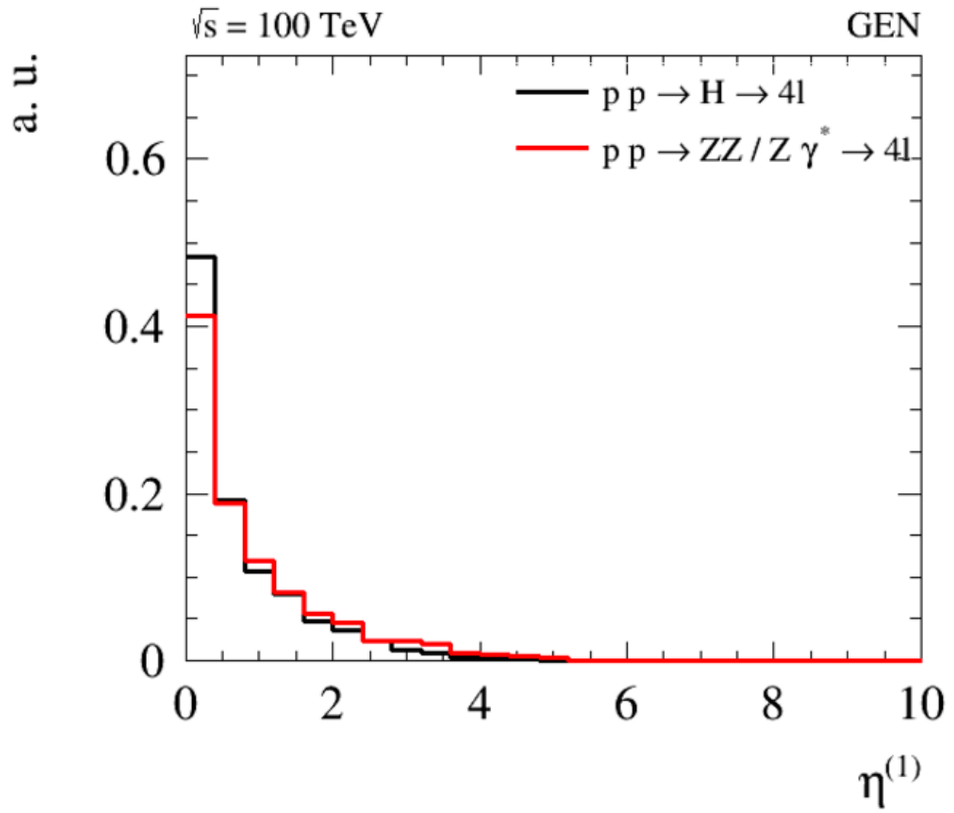
Download the python code:

```
git clone https://github.com/selvaggi/tutorials.git
```

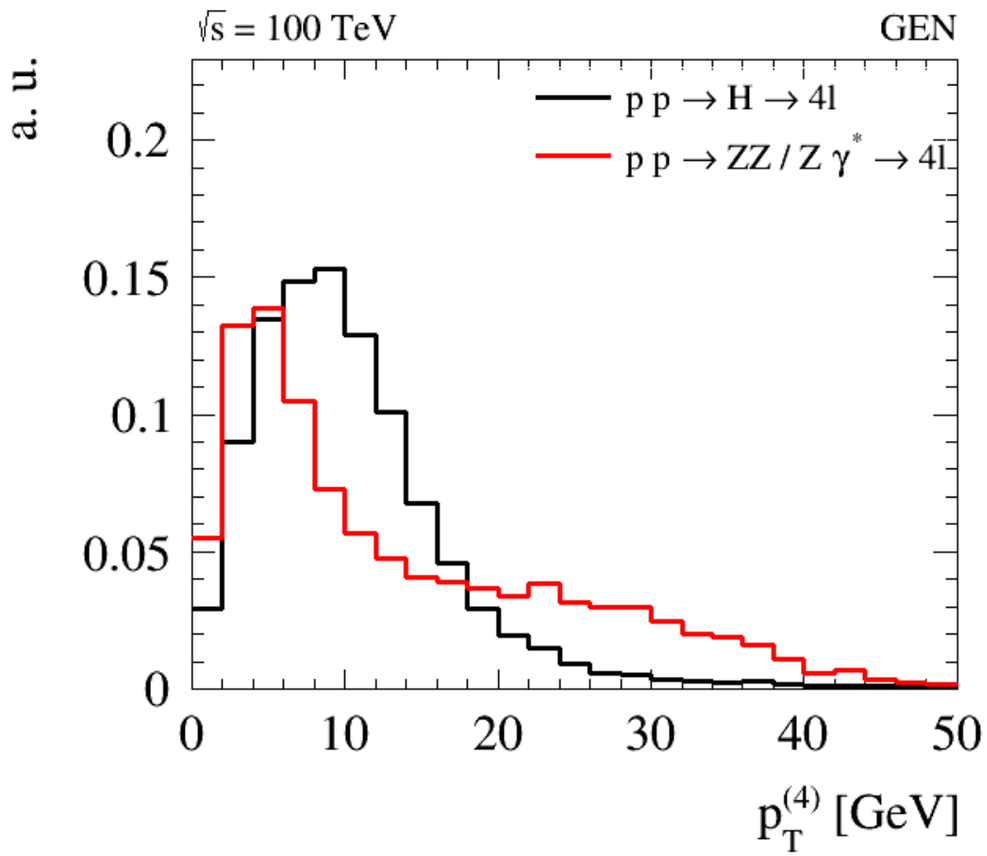
Produce Gen-level plots:

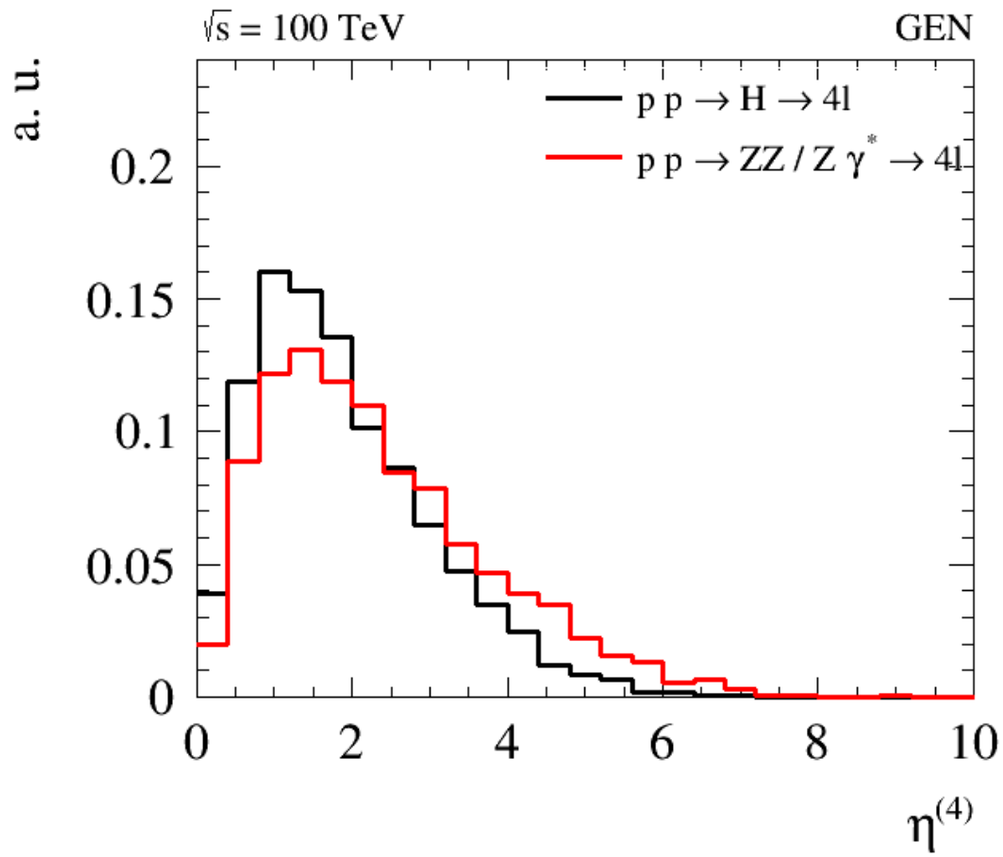
```
python tutorials/fcc/createGenHistos.py
eog plots/lep*.png
```





Tutorial - produce plots (gen)





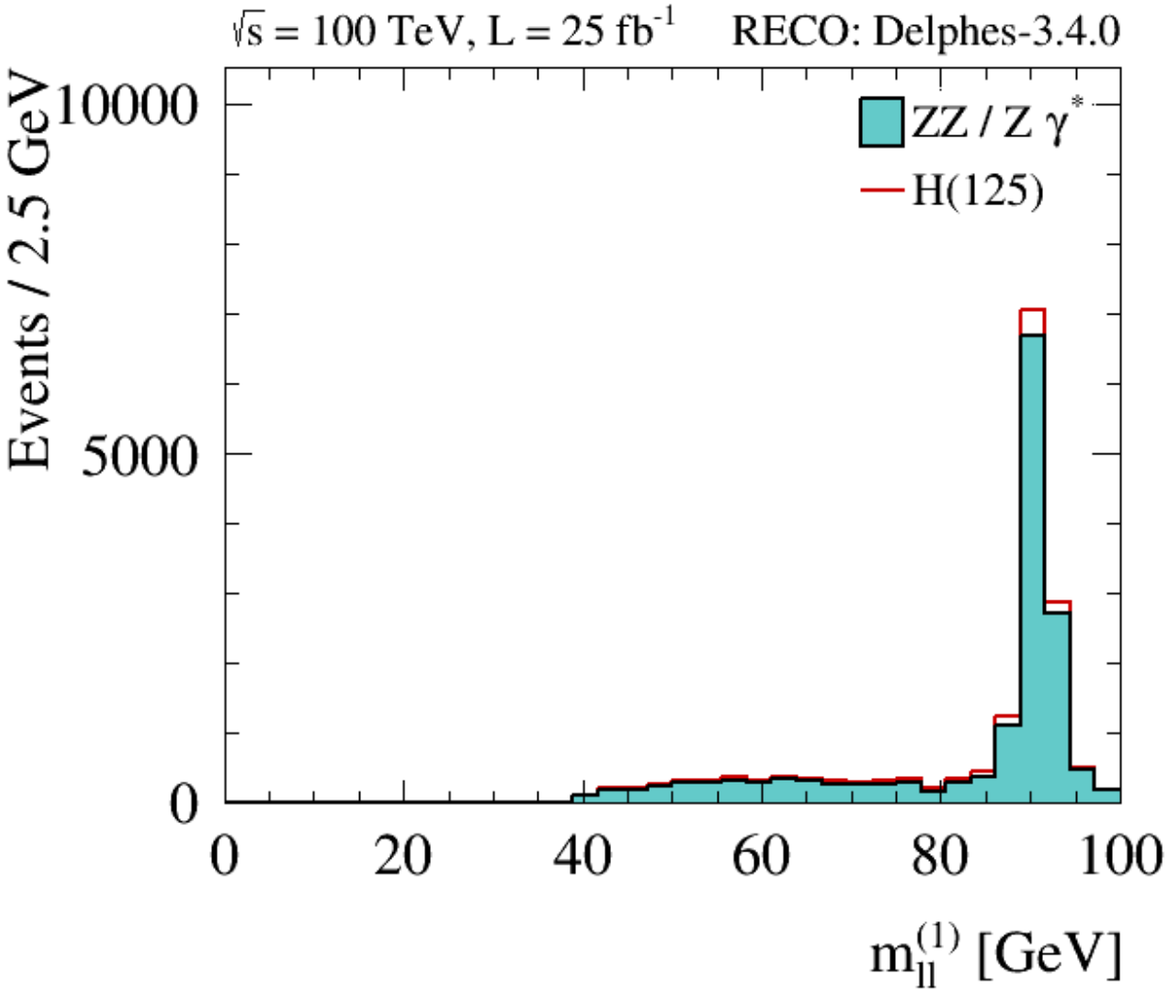
Tutorial - cutflow

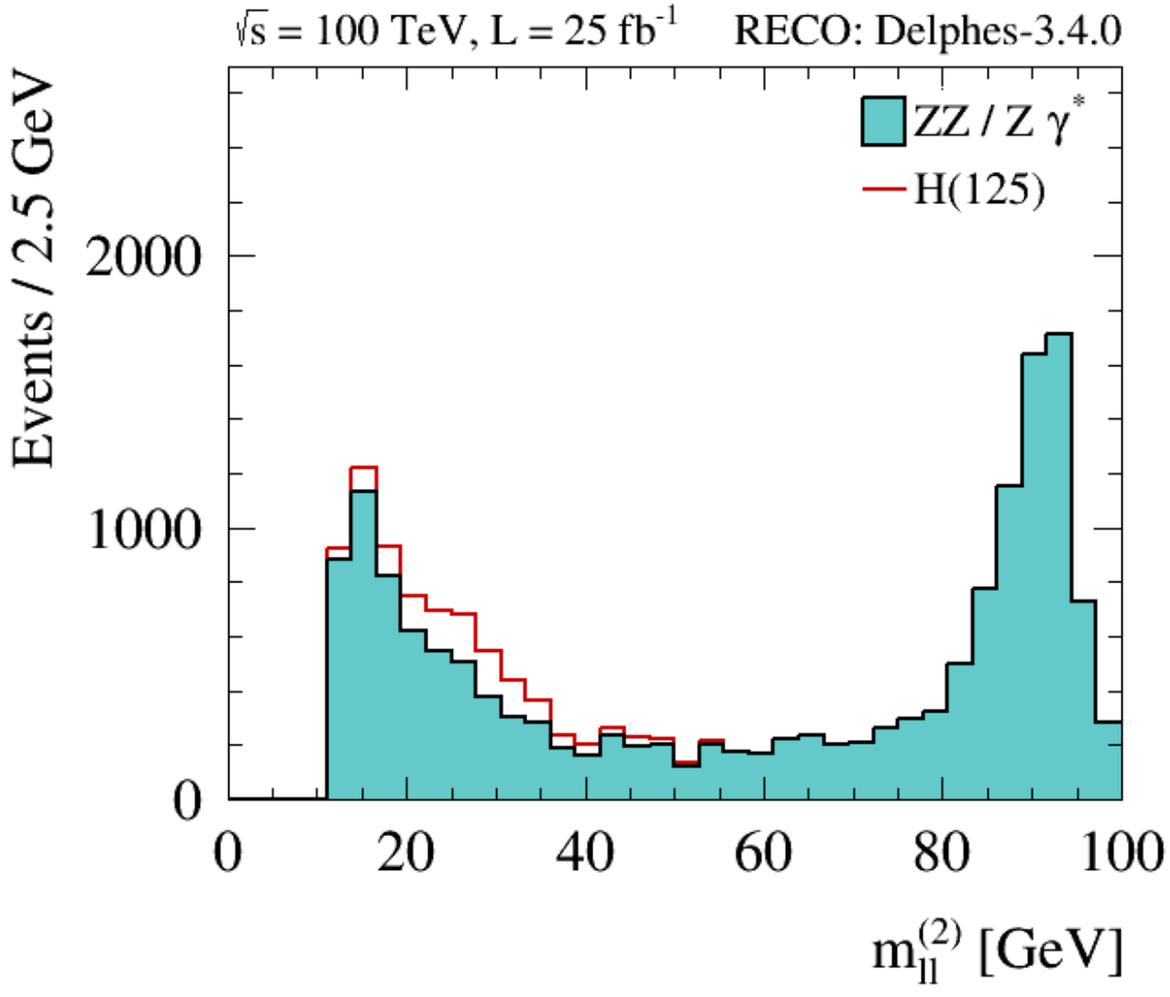
```
Counter cut_flow :
         All events                                  10000       1.00    1.0000
         At least one Z -> l+ l- candidates           9961       1.00    0.9961
         40 < mZ1 < 120                               8643       0.87    0.8643
         At least a second Z -> l+ l- candidates      5931       0.69    0.5931
         12 < mZ2 < 120                               5681       0.96    0.5681
         leading lepton pT > 20                       5634       0.99    0.5634
         sub-leading lepton pT > 10                   5629       1.00    0.5629
```

- Heppy creates a process_dir. “pp_h_4l, pp_zgzg_4l”
- Selection cut table can be found in:

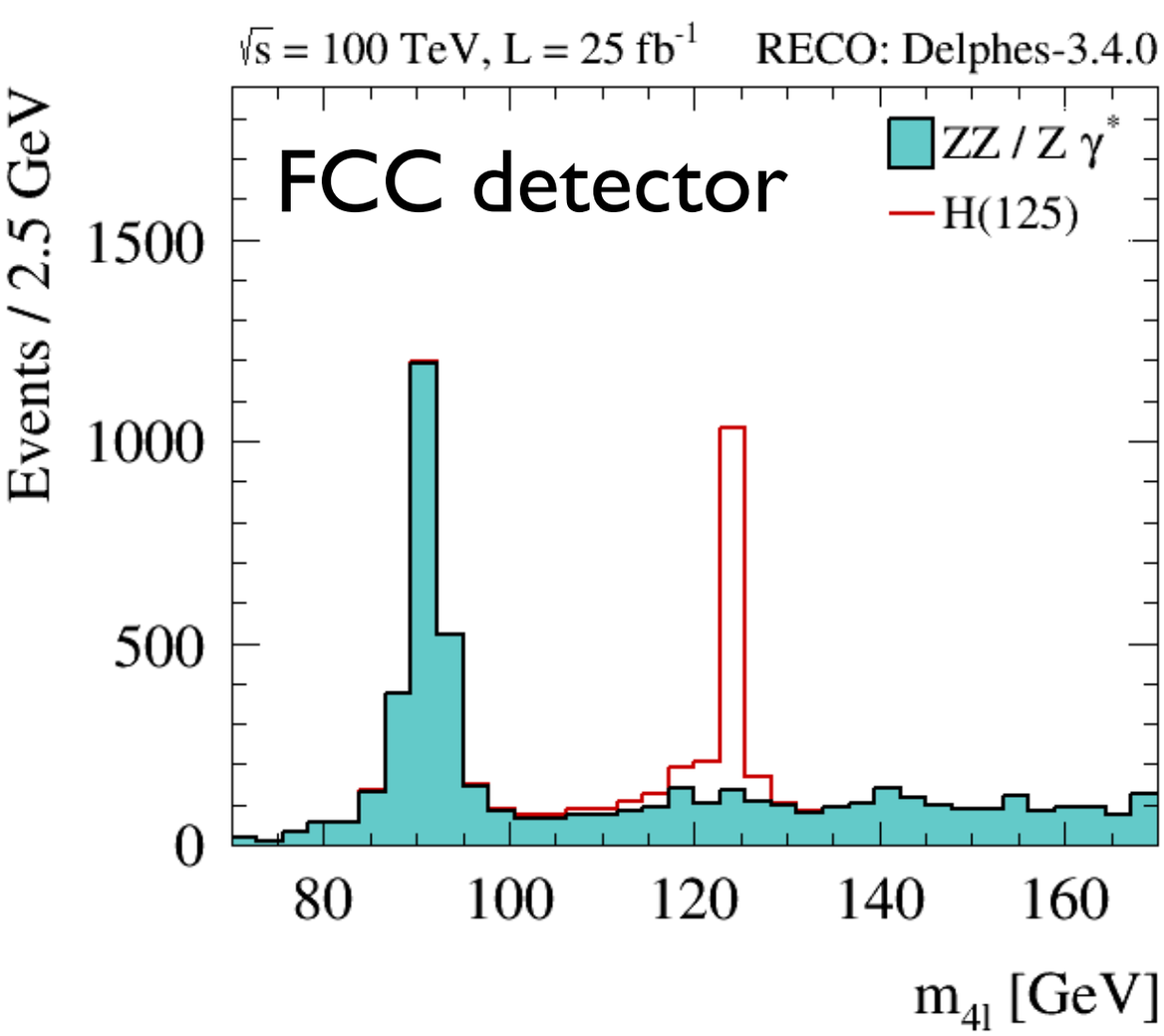
process_dir/example/XXX.Selection_cuts/cut_flow.txt

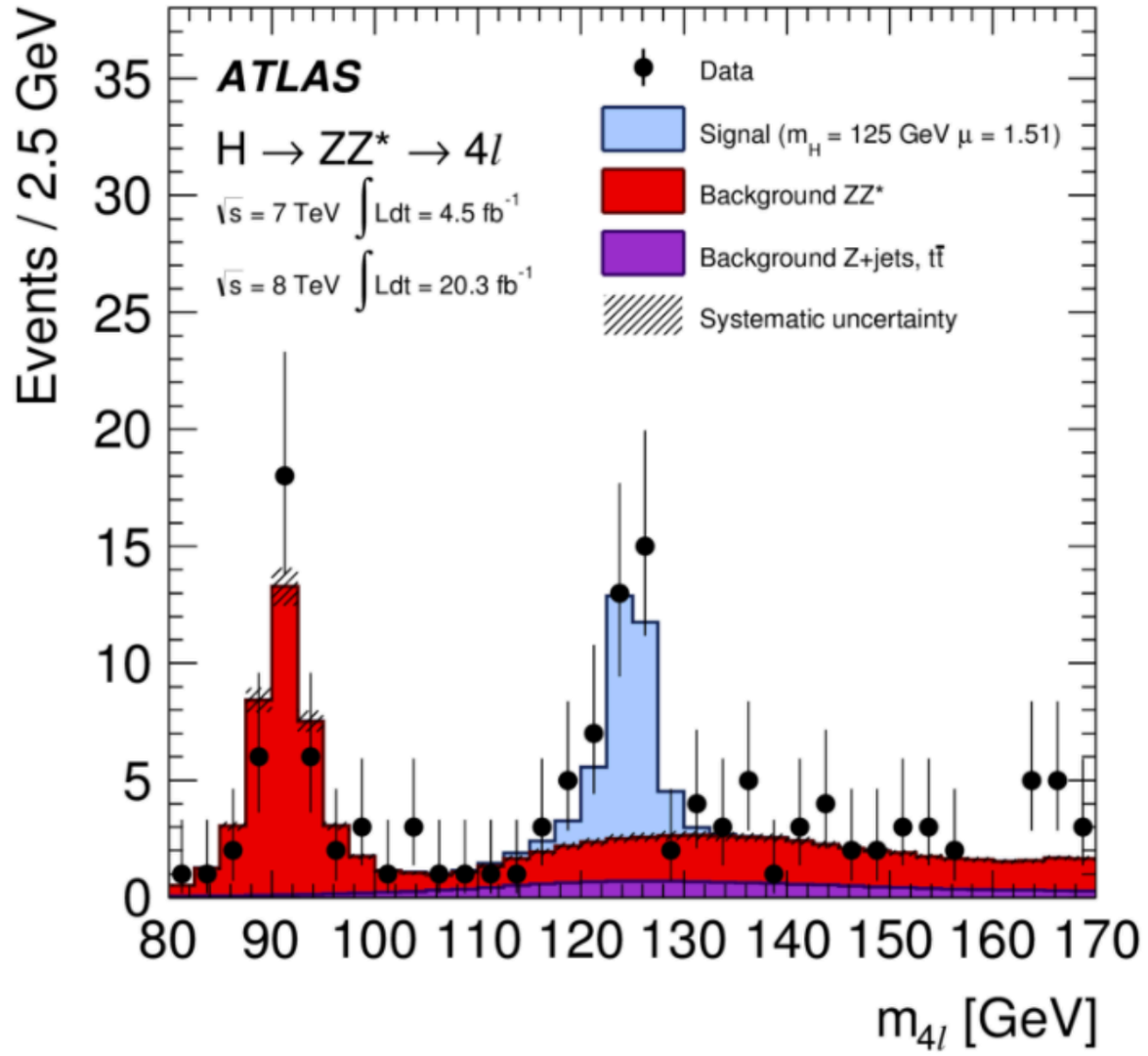
Tutorial - produce plots (reco)



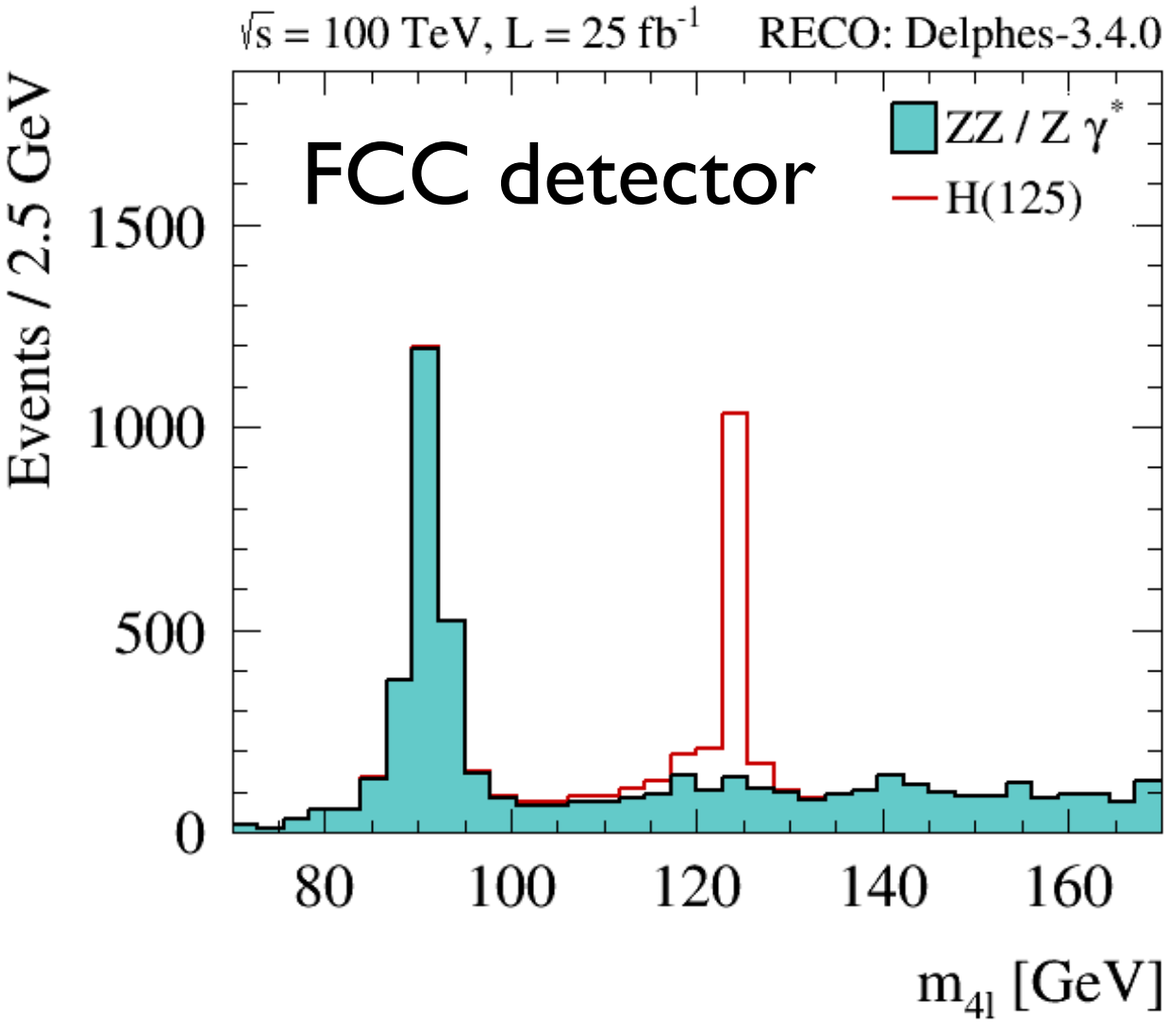


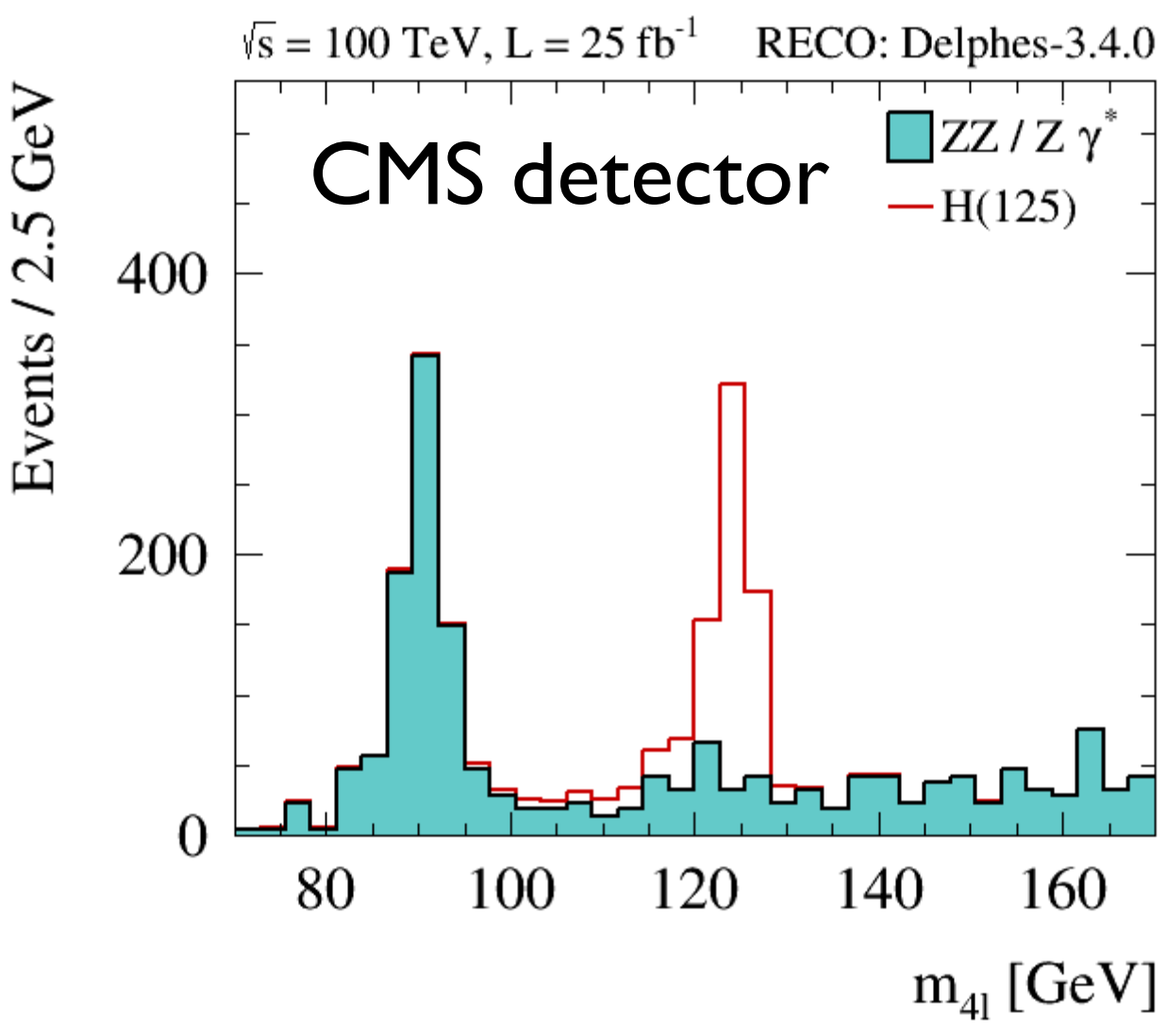
Tutorial - produce plots (reco)





Tutorial - produce plots (reco)





Comments

- very **simple example** implemented in the form of a **tutorial**

- try it out (including **homework section**)

- should be easy enough to **get started** with FCCSW and Pythia8/Delphes simulation

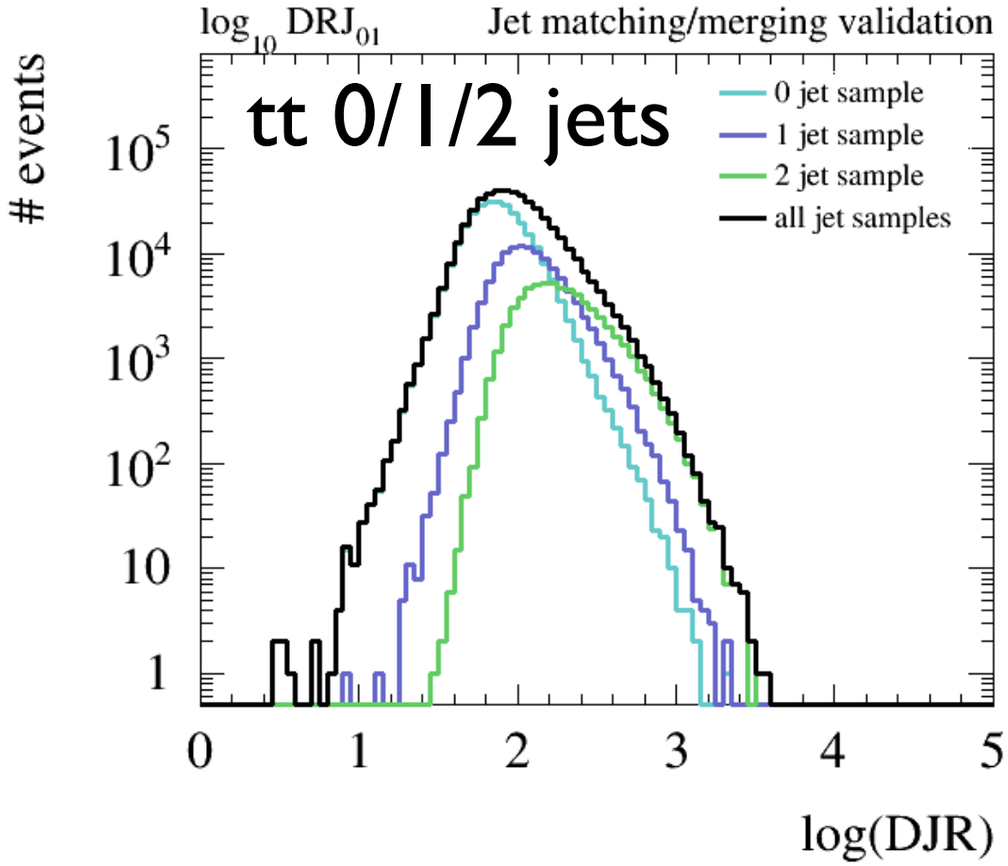
Sample production

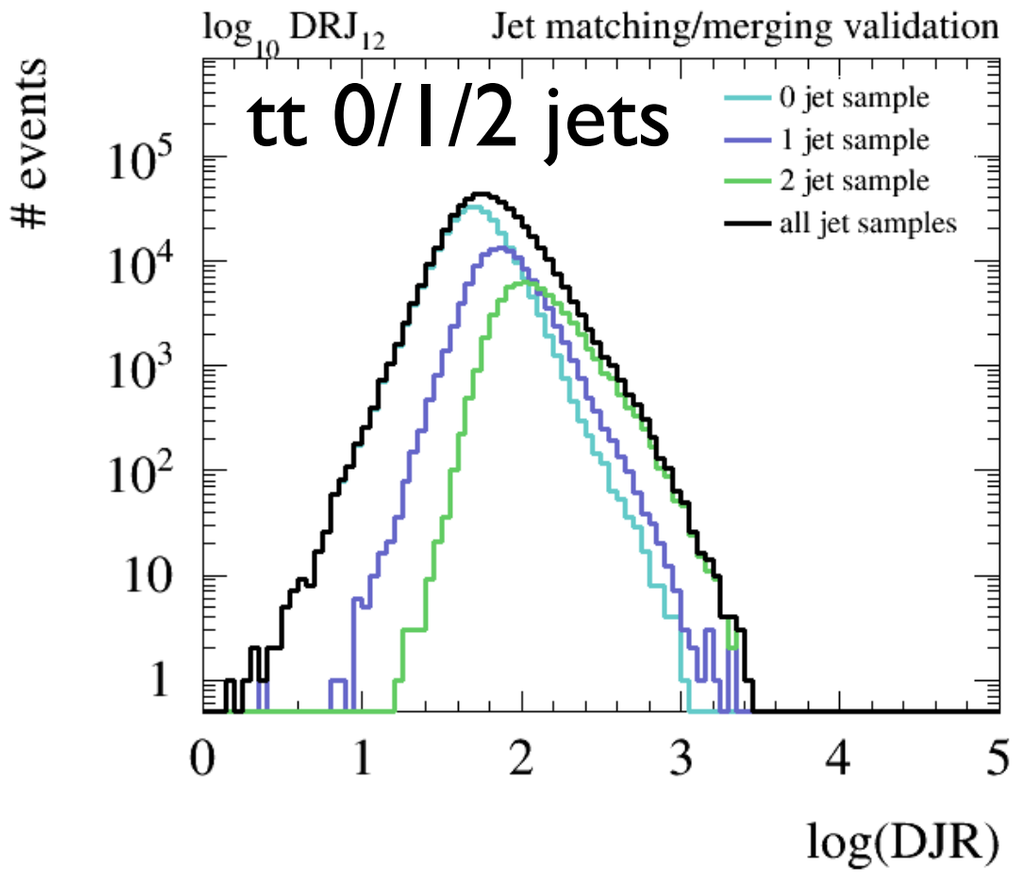
- We started **producing** some general samples **centrally** (and store them in eos)
  - W+jets
  - Z+jets
  - t t + jets
  - QCD
- Clement Helsens working on **automatising** the following workflow :
  - produce grid-packs with MG5_aMC@NLO
  - generate LHE files on batch (eventually condor)
  - shower/hadronize/merge with Pythia8 on queue

Matching/merging

- **Matching/merging** is necessary to ensure good description of emissions in **soft/col** and **hard** (PS vs. ME)
- Major (CKKW/MLM) **merging schemes** are already implemented in **Pythia8**
- Pythia8 interface in FCCSW **has been modified** to include matching
- **Automatic plots** are produced to validate the choice of the scales

Matching/merging validation





To be done:

- allow for **multiple** choice of **matching scales** qCut at once
- produce **matched cross section** numbers together with matchin plots

Matching/merging tutorial

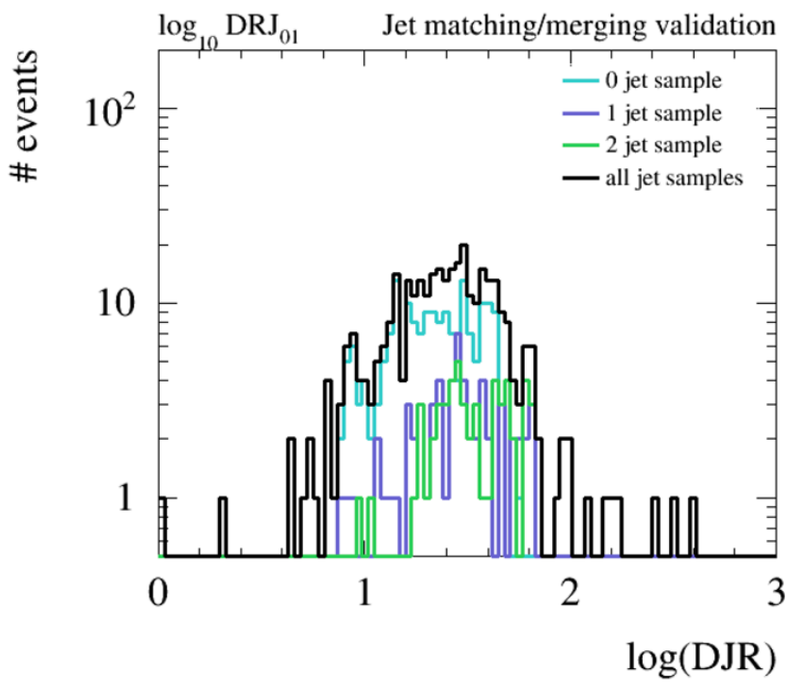
https://github.com/HEP-FCC/fcc-physics/blob/master/pythia8/validation/README.md

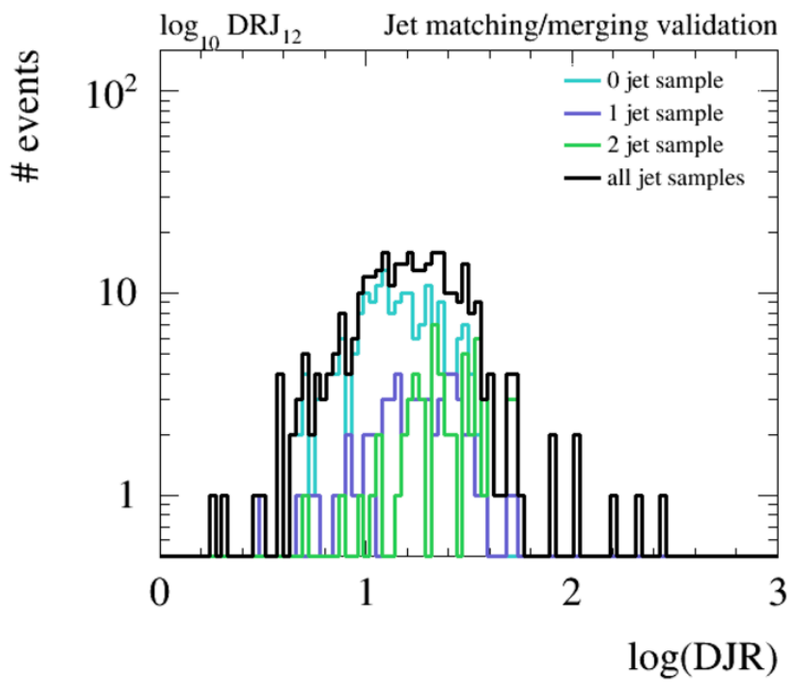
Within FCCSW, run this example:

```
./run fccrun.py Sim/SimDelphesInterface/options/PythiaDelphes_config.py --inputfile=Generation/data/Pythia_
```

To produce matching plots, fetch the creatingMatchingPlots.py script in this repository, and run it:

```
python createMatchingPlots.py fccOutput.root
```





Sample database

https://test-fcc.web.cern.ch/test-FCC/index.php

Home - Les houches events - Delphes events - Contact Us

Les Houches Events

	name	nevents	nfiles	outputdir	mainprocess	finalstates	merging
1	pp_hh_bbaa	600000	60	/eos/fcc/hh/generation/mg5_amcatnlo/lhe/pp_hh_bbaa/	gluon gluon fusion di-higgs	bbbar gammagamma	
2	pp_jj012j_5f	40000	4	/eos/fcc/hh/generation/mg5_amcatnlo/lhe/pp_jj012j_5f/	dijet + 0,1,2 jets 5 flavor scheme		
3	pp_jjaa01j_5f	0	0		dijet diphoton + 0,1,2 jets 5 flavor scheme		
4	pp_tt012j_5f	1000000	50	/eos/fcc/hh/generation/mg5_amcatnlo/lhe/pp_tt012j_5f/	ttbar + 0,1,2 jets 5 flavor scheme	inclusive decays	xqcut = 10, qCut = 15
5	pp_tt012j_5f_v2	1000000	50	/eos/fcc/hh/generation/mg5_amcatnlo/lhe/pp_tt012j_5f_v2/	ttbar + 0,1,2 jets 5 flavor scheme	inclusive decays	xqcut = 60, qCut = 90
6	pp_w012j_5f	1000000	50	/eos/fcc/hh/generation/mg5_amcatnlo/lhe/pp_w012j_5f/	w + 0,1,2 jets 5 flavor scheme	inclusive decays	xqcut = 10, qCut = 15
7	pp_w012j_5f_v2	1000000	50	/eos/fcc/hh/generation/mg5_amcatnlo/lhe/pp_w012j_5f_v2/	w + 0,1,2 jets 5 flavor scheme	inclusive decays	xqcut = 30, qCut = 45
8	pp_z012j_5f	1000000	50	/eos/fcc/hh/generation/mg5_amcatnlo/lhe/pp_z012j_5f/	z + 0,1,2 jets 5 flavor scheme	inclusive decays	xqcut = 10, qCut = 15
9	pp_z012j_5f_v2	980000	49	/eos/fcc/hh/generation/mg5_amcatnlo/lhe/pp_z012j_5f_v2/	z + 0,1,2 jets 5 flavor scheme	inclusive decays	xqcut = 30, qCut = 45

Comments/conclusion

- tutorial for **producing** and **validating** matched/samples can be found here:

https://github.com/HEP-FCC/fcc-physics/blob/master/pythia8/validation/README.md

- we have already **produced centrally** > 500k events of main backgrounds **tt, w, z + jets**

  /eos/fcc/hh/generation/DelphesEvents/v0_0/pp_tt012j_5f_v2
  /eos/fcc/hh/generation/DelphesEvents/v0_0/pp_w012j_5f_v2
  /eos/fcc/hh/generation/DelphesEvents/v0_0/pp_z012j_5f_v2

- would be good to have some discussion to decide **which (signal/ background) samples to be centrally produced** and if **LOPS merged vs. NLO** (maybe during FCC physics week)