Using Rivet in CMSSW

A. Knutsson (DESY)

CMSSW Tutorial

DESY MC School - 18th March 2011

Running Rivet in CMSSW
Using your own Rivet analysis in CMSSW

Reminder - Rivet

Rivet is a toolkit for analysis of MC events.

"The Rivet project (Robust Independent Validation of Experiment and Theory) is a toolkit for validation of Monte Carlo event generators. /.../ ***Rivet provides a set of experimental analyses*** *useful for generator sanity checks, as well as a* ***convenient infrastructure for adding your own analyses****."*

- Rivet does not generate event or call generators. It is reading files with events stored in **the HepMC format**.

  Input: HepMC format (ASCII)
  Output: Histograms in the Aida-format. (aida2root- and aida2flat-scripts provided in Rivet)

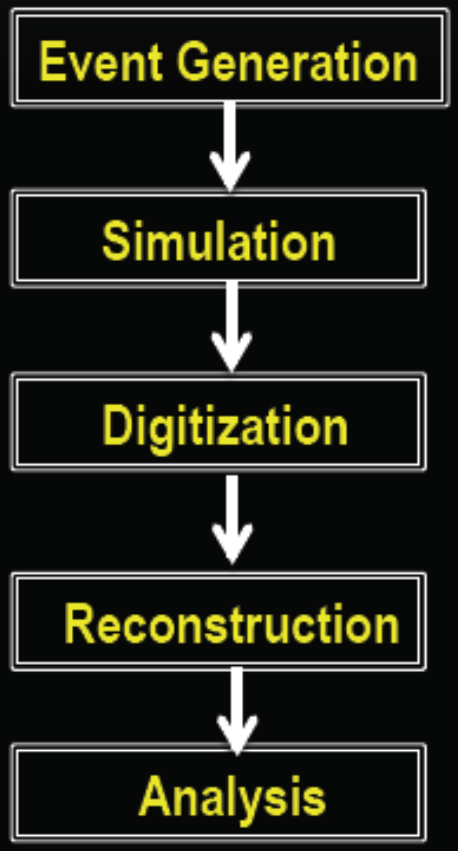
- Studies are made on the output from the generators, i.e. pure MC studies, and **no detector simulations** involved.

Rivet manual: [Bucklet et al, arXiv:1003.0694v4]

What we did yesterday...

From Altan Cakirs presentation:

ANALYSIS CHAIN



First step is to generate the events of your required physics problem, using event generators like Pythia, MadGraph, SOFTSUSY etc.

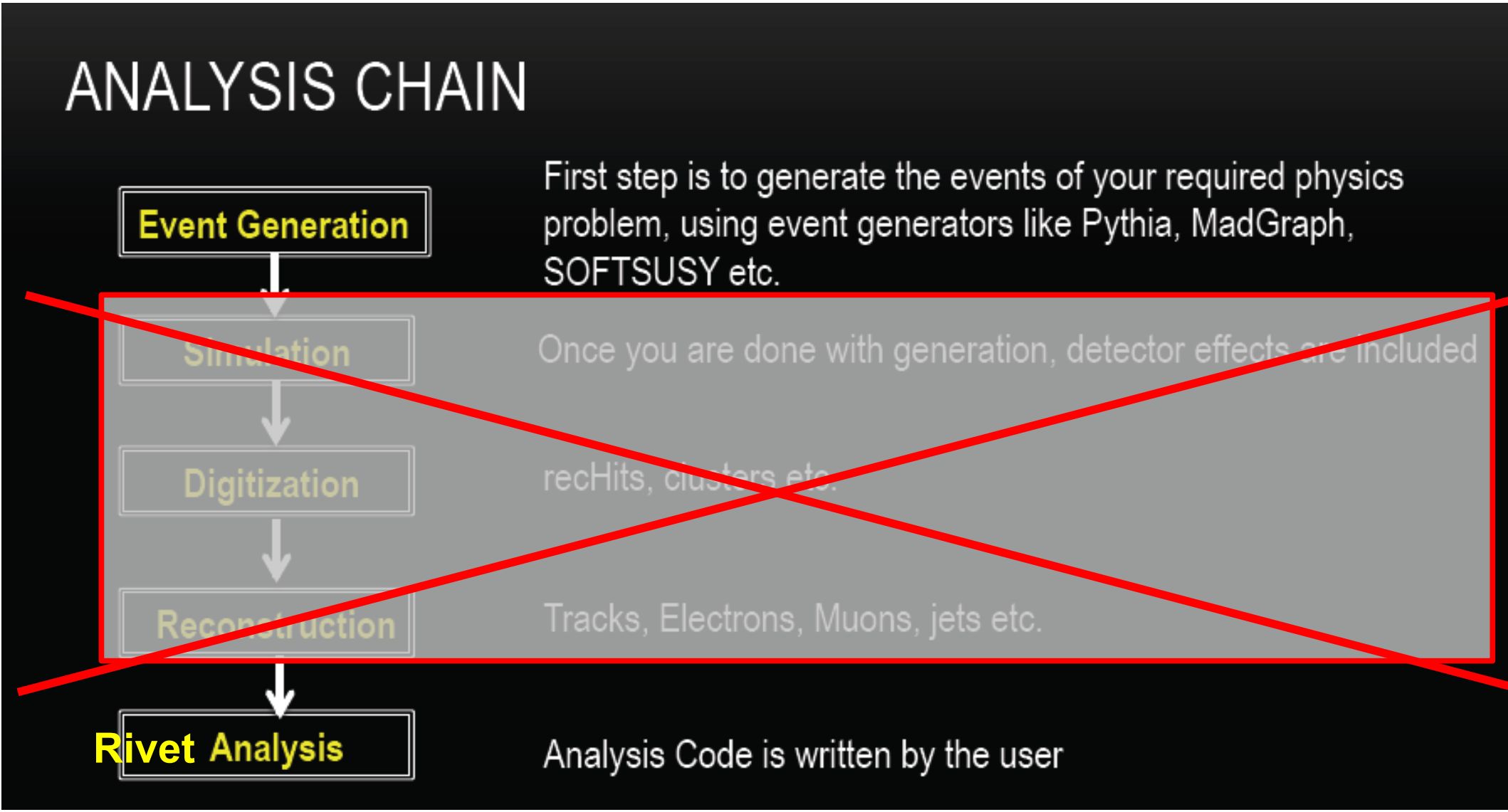
Once you are done with generation, detector effects are included

recHits, clusters etc.

Tracks, Electrons, Muons, jets etc.

Analysis Code is written by the user

Sim and Digi - Not an option for Rivet



Reminder - Rivet

Rivet is a toolkit for analyses of MC events.

"The Rivet project (Robust Independent Validation of Experiment and Theory) is a toolkit for validation of Monte Carlo event generators. /.../ ***Rivet provides a set of experimental analyses*** *useful for generator sanity checks, as well as a* ***convenient infrastructure for adding your own analyses****."*

- Rivet does not generate event or call generators. It is reading files with events stored in **the HepMC format**.

  Input: HepMC format (ASCII)
  Output: Histograms in the Aida-format. (aida2root- and aida2flat-scripts provided in Rivet)

- Studies are made on the output from the generators, i.e. pure MC studies, and **no detector simulations** involved.

Reminder - Rivet

- The ***Offical Rivet*** *1.4.0* comes with ~10 LEP, ~40 Tevatron and a couple of SPS, HERA and RHIC analyses. ~5 ATLAS analyses are included.
  No official CMS analyses at the moment. (We need your help for this.)

  If you run these analyses you can make your own MC comparisons to published data.

  Or you can **make MC studies by your own** by writing your own analyzer. You can then study whatever physics you like, for example some pre-physics studies for a data analysis your plan to perform.

Rivet in CMSSW

- Running Rivet in CMSSW is easy, and only a little different to running Rivet standalone, e.g. you **compile with scram** (as usual in CMSSW) and you **run the generator and Rivet with cmsRun**.

- Rivet is only recently implemented in CMSSW, and you will only find it in later CMSSW releases.

- A couple of CMS analyses are included in the CMSSW Rivet version.

- Since you have been working with Rivet the last days we will not go into depth of the functionality. We focus on the CMSSW part. (However I am willing to help with coding and answer questions after the tutorial...)

Setting up the CMSSW version

- Almost everything you need you will find on the CMS Rivet Twiki:
  https://twiki.cern.ch/twiki/bin/view/CMS/Rivet

  On the next slides we will walk through the instructions.

Start of tutorial.

In this tutorial we will use **CMSSW_4_2_0_pre5**.

- Set up this environment on an lxplus machine (see Altans tutorial how to login etc):

```
setenv SCRAM_ARCH slc5_amd64_gcc434
cmsrel CMSSW_4_2_0_pre5
cd CMSSW_4_2_0_pre5/src
cmsenv
```

- Test if you have things properly set up at this stage, e.g. list all available analyses:

```
rivet --list-analyses
```

Rivet in CMSSW

- Get the generator config file you want to use.

  Let's use Pythia with Tune Z2 (which is the most recent CMS tune (Rick Field)):

  cvs co Configuration/GenProduction/python/MinBias_TuneZ2_7TeV_pythia6_cff.py

  or Herwig++ (QCD 2to2):

  cvs co Configuration/GenProduction/python/
  QCD_Pt_15to3000_Tune23_Flat_7TeV_herwigpp_cff.py

- Get the CMSSW config file for Rivet:
  cvs co Configuration/GenProduction/python/rivet_customize.py

- Get the RivetInterface:
  cvs co GeneratorInterface/RivetInterface

- Compile (still in CMSSW_4_2_0_pre5/src):
  scram b

Rivet in CMSSW

• Create the config-file from the generator- and the rivet-steering - for cmsDriver:

cmsDriver.py Configuration/GenProduction/python/MinBias_TuneZ2_7TeV_pythia6_cff.py -s GEN --datatier=GEN-SIM-RAW --conditions auto:mc --eventcontent RAWSIM --no_exec -n 5000 --python_filename=rivet_cfg.py --customise=Configuration/GenProduction/rivet_customize.py

(one line)

This creates the file **rivet_cfg.py**, which is a config file for cmsRun. It contains parameters for both Rivet and the generator.

cmsDriver you learnt to use yesterday. The options for Rivet differ slightly from what Altan showed yesterday. For Rivet we don't need/want the SIM, DIGI, etc...

Rivet in CMSSW

Open **rivet_cfg.py** using your favorite text editor.

• You will find “standard” CMSSW variables, that you recognize from yesterday, e.g. how many events you want to generate. (You asked for 5000 when you created the config-file.)

• Make sure that your center the mass energy is set to 7 TeV

• You can choose what Rivet analysis to run. Let's run only *the charged particle spectra analysis*. It has the ID *CMS_2010_S8656010*. (8656010 refers to publication ID on Spires.)

Locate and edit the following line in rivet_cfg.py:

process.rivetAnalyzer.AnalysisNames = cms.vstring('CMS_2010_S8656010').

(If you will create many config files for different generators it can be a good idea to instead modify Configuration/GenProduction/python/rivet_customize.py before you repeatedly execute cmsDriver.)

A look at the Rivet analyser

Open the Rivet analyzer.
The unofficial CMS analyses are stored in RivetInterface/src:

nedit GeneratorInterface/RivetInterface/src/CMS_2010_S8656010.cc &

Identify the following things in the code:
(Depending on the knowledge you already gained the last days you can skip this step.)

- The **Init()** function. It is *called once before the event generation*.
  Here projections are declared and histogram booked.
- The **Analyze()** function. This function is *called for each event*.
  Here the event analyses is performed and histograms are filled.
- The **Finalize()** function. It is called *once the event generation is finished*.
  Here histograms are normalized. A good place to write out some statistics of the run.
- Identify the so called "**projection**" for charged particles and what initial selection that is applied to the charges particles. (For info about projections I refer to the Rivet manual)

Running Rivet in CMSSW

- Run the generator and analyse the generated events with Rivet:

  cmsRun rivet_cfg.py

  cmsRun will run the CMSSW installed generator and Rivet in parallel.
  The MC generator produce events (pp collisions) which are analyzed by Rivet on the fly.

  Hint: Just hit ctrl+c if you get tired of waiting for all the event.

- An AIDA-file is produced by Rivet. This contains the filled histograms.

  The AIDA file is an ASCII file. If you want, take a look at the AIDA-file: **less out.aida**
  (You will find bin centers, errors and contents, and some other stuff.)

  You most likely want to work with ROOT. Convert the .aida to .root:

  aida2root out.aida

Running Rivet in CMSSW

- The experimental data for the measurement is stored under GeneratorInterface/RivetInterface/data/
  Convert also this .aida to .root.

  aida2root GeneratorInterface/RivetInterface/data/CMS_2010_S8656010.aida

- Look at the histograms, and compare the MC to the data.

  root -l out.root CMS_2010_S8656010.root
  TBrowser l;
  etc...

  Note: The “histograms” are stored in the .root file as TGraphErrors, which means that you need to **switch to “alp”** in the TBrowser before you click on the TGraphErrors.

  Put the data and the MC in the same plot for one of the distributions.

  The MC and the data does not agree...
  Can you say one reason why? (Tricky.)

End of part 1.

Part 2 – Compile and run your own Rivet analysis in CMSSW

Working with your own Rivet analysis in CMSSW

1) For simplicity I created a new example analysis for you.
   Copy it to your GeneratorInterface/RivetInterface/src.

cp ~knutsson/public/rivet/RIVET_TEMPLATE_CMS.cc GeneratorInterface/RivetInterface/src/.

Working with your own Rivet analysis in CMSSW

1) For simplicity I created a new example analysis for you.
Copy it to your GeneratorInterface/RivetInterface/src.

cp ~knutsson/public/rivet/RIVET_TEMPLATE_CMS.cc GeneratorInterface/RivetInterface/src/.

2) Open **RIVET_TEMPLATE_CMS.cc** in order to get a feeling what it does.
What projections are used? How many histograms are filled? What are they filled with?

Working with your own Rivet analysis in CMSSW

1) For simplicity I created a new example analysis for you.
Copy it to your GeneratorInterface/RivetInterface/src.

cp ~knutsson/public/rivet/RIVET_TEMPLATE_CMS.cc GeneratorInterface/RivetInterface/src/.

2) Open **RIVET_TEMPLATE_CMS.cc** in order to get a feeling what it does.
What projections are used? How many histograms are filled? What are they filled with?

3) You realize that the example routine use jets from the FastJet package.
Add the FastJet package to the BuildFile.
(GeneratorInterface/RivetInterface/BuildFile.xml should contain **<use name="fastjet"/>**)

Working with your own Rivet analysis in CMSSW

1) For simplicity I created a new example analysis for you.
Copy it to your GeneratorInterface/RivetInterface/src.

cp ~knutsson/public/rivet/RIVET_TEMPLATE_CMS.cc GeneratorInterface/RivetInterface/src/.

2) Open **RIVET_TEMPLATE_CMS.cc** in order to get a feeling what it does.
What projections are used? How many histograms are filled? What are they filled with?

3) You realize that the example routine use jets from the FastJet package.
Add the FastJet package to the BuildFile.
(GeneratorInterface/RivetInterface/BuildFile.xml should contain **<use name="fastjet"/>**)

4) Compile (still in CMSSW_4_2_0_pre5/src).
scram b

Working with your own Rivet analysis in CMSSW

1) For simplicity I created a new example analysis for you.
Copy it to your GeneratorInterface/RivetInterface/src.

cp ~knutsson/public/rivet/RIVET_TEMPLATE_CMS.cc GeneratorInterface/RivetInterface/src/.

2) Open **RIVET_TEMPLATE_CMS.cc** in order to get a feeling what it does.
What projections are used? How many histograms are filled? What are they filled with?

3) You realize that the example routine use jets from the FastJet package.
Add the FastJet package to the BuildFile.
(GeneratorInterface/RivetInterface/BuildFile.xml should contain **<use name="fastjet"/>**)

4) Compile (still in CMSSW_4_2_0_pre5/src).
scram b

5) Check if the new analysis is available.
rivet --list-analyses

Working with your own Rivet analysis in CMSSW

1) For simplicity I created a new example analysis for you.
Copy it to your GeneratorInterface/RivetInterface/src.

cp ~knutsson/public/rivet/RIVET_TEMPLATE_CMS.cc GeneratorInterface/RivetInterface/src/.

2) Open **RIVET_TEMPLATE_CMS.cc** in order to get a feeling what it does.
What projections are used? How many histograms are filled? What are they filled with?

3) You realize that the example routine use jets from the FastJet package.
Add the FastJet package to the BuildFile.
(GeneratorInterface/RivetInterface/BuildFile.xml should contain **<use name="fastjet"/>**)

4) Compile (still in CMSSW_4_2_0_pre5/src).
scram b

5) Check if the new analysis is available.
rivet --list-analyses

6) Analyse a couple of events and see if the histograms are filled. Go back a few slides if you already forgot how. (Hint: you need to edit the rivet_cfg.py.)

Working with your own Rivet analysis in CMSSW

You are now ready to write your own Monte Carlo analysis!
Feel free to modify the template.

Suggestion: Do this for your own CMS analysis.

If you have any **questions or problem** with running Rivet in CMSSW after following the CMS twiki,
please **contact me** or use the **generator tools hypernews**.

Good luck and have a lot of fun with your MC studies!