

Editorial Manager(tm) for Journal of Physics: Conference Series
Manuscript Draft

Manuscript Number: CHEP226R1

Title: HepMCAnalyser - a tool for Monte Carlo generator validation

Article Type: Poster

Corresponding Author: Dr. Cano Ay,

Corresponding Author's Institution: University of Goettingen

First Author: Cano Ay

Order of Authors: Cano Ay; Judith Katzy, Dr; Sebastian Johnert, phd; Zhonghua Qin, Dr

HepMCAnalyser: A tool for Monte Carlo Generator Validation

C. Ay¹, S. Johnert², J. Katzy², Zhonghua Qin²

¹ University Goettingen, II. institutes of Physics, Germany

² DESY, Hamburg, Germany

Abstract.

HepMCAnalyser is a tool for Monte Carlo (MC) generator validation and comparisons. It is a stable, easy-to-use and extendable framework allowing for easy access/integration to generator level analysis. It comprises a class library with benchmark physics processes to analyse MC generator HepMC output and to fill root histograms. A web-interface is provided to display all or selected histograms, compare to references and validate the results based on Kolmogorov Tests. Steerable example programs can be used for event generation. The default steering is tuned to optimally align the distributions of the different MC generators. The tool will be used for MC generator validation by the Generator Services (GENSER) LCG project, e.g. for version upgrades. It is supported on the same platforms as the GENSER libraries and is already in use at ATLAS.

1. Introduction

As the GENSER project (Generator Services project[1]) provides only generator libraries it is a natural extension to provide the environmental set-up and the executables to allow users to run the generators and to histogram the results. Hence the idea was to develop a framework with a minimal overhead that allows an easy access to generator studies such as regression tests and debugging to discriminate between bugs in the experimental software and genuine generator bugs. In addition, it should serve as a starting point for students.

For this purpose a package has been developed that provides a minimal set-up script and a Makefile to form executables to generate events in HepMC format and analyse them with a class library. The class library provides access to the FastJet JetFinder[2] and produces output in ROOT format. Therefore with the access to a the GENSER installation, the only additional package is ROOT[3]. The general setup is described in Sect. 3. In addition steering is provided to ease the study with various generator steering and to give examples on event generation for a few benchmark processes. This is described in Sect. 5. The class library for generator analysis in HepMC[4] format covers the various aspects of event generation such as hard process, parton shower and underlying event. As the class library uses the HepMC format to interface to the generator output it is cleanly separated from the generating code. This gives the additional advantage that the same analysis can be run in different contexts to check on possible deviations due to different computing environments. This is particularly important to discriminate between genuine generator bugs and bugs introduced by the experimental framework. The class library is briefly described in Sect. 4. In addition the scripts are provided to create a web-interface to

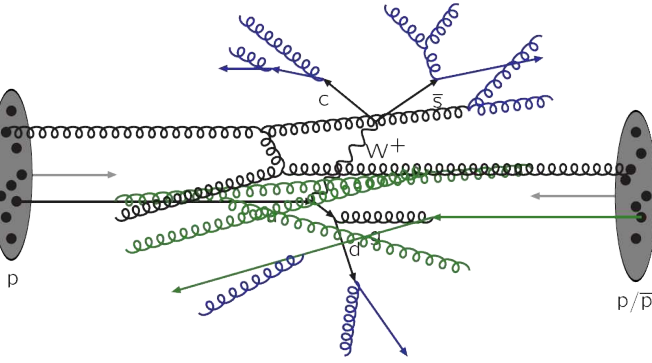


Figure 1. Example of an event for proton-proton collision[5]. The sketch shows some of the aspects of event generation, like the hard process, parton shower and underlying event.

display the analysis results in an automated way. This is described in Sect. 6. The installation is described in Sect. 7.

2. The HepMCAnalyser: design goals

The aim is to provide an easy to use minimal framework for regression tests and as a starting point for user analysis. It should allow to study the various aspects of event generation such as

- hard process,
- parton shower,
- multiple interaction and underlying event,
- hadronisation.

An example of an event generation for proton-proton collision is illustrated in Fig. 1, which shows the hard process, parton shower and multiple interaction.

The main focus for this framework is

- simplicity: framework overhead as low as possible,
- robustness: minimal dependency on other packages and the programming language should be standard C++,
- scalability: extendable for other user analysis or applications.

3. Technical implementation

The framework provides all parts needed for an analysis:

- 1 a software environment to run and analyse MC generators, i.e. set-up script and Makefile to, link the MC generator software with the parton-density function (pdf) library, HepMC, the HepMC Analysis library and ROOT into an executable main program,
- 2 HepMC Analysis class library with physics analysis for regression test and example of user analysis,
- 3 steering files for meaningful generator tunings and standard process generation,
- 4 scripts for web-display with comparison to reference histograms.

The technical workflow is illustrated in Fig. 2. For each MC generator an example program is provided. Following the design goals described in Sect. 2 the framework does not include any generator libraries. The libraries are delivered by the GENSER project, which provide LCG compliant code for both the theoretical and experimental communities at the LHC[6, 7]. GENSER also provides the libraries for HepMC, Fastjet and the HepMC Analysis library. Only

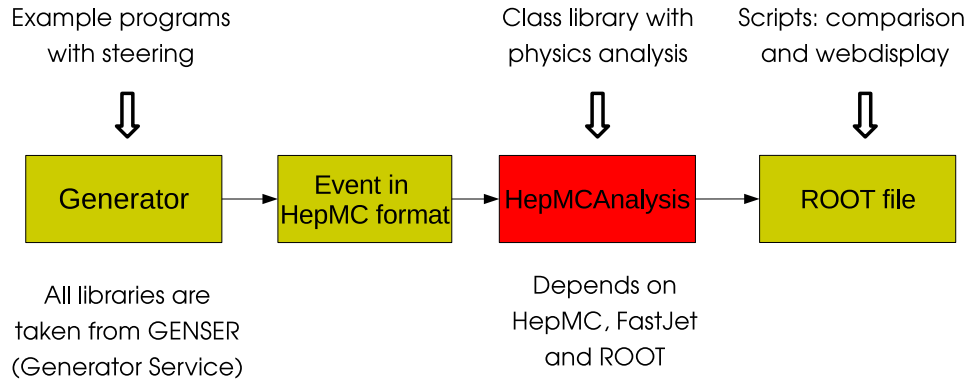


Figure 2. The HepMCAnalyser workflow shows the steps from event generation to to the comparison and display.

ROOT does not come from GENSER.

Those example programs include the MC generator specific functions for configuration, initialisation and event generation. The external generator returns the events in HepMC format, which are then processed by HepMCAnalyser analysis classes. Those physics analysis classes only depend on Fastjet to find the jets in the event and on ROOT to create and fill the histograms. Example programs exist for the generators Pythia6[8], Pythia8[9], Herwig[10], Herwig++[11] and Cascade[12]. An example program for Sherpa[13] will be implemented soon.

All the histograms created by the selected analysis are stored in one ROOT file. Finally the histograms of different generators or different versions of one generator can be compared and displayed.

4. Class library

One analysis class is dedicated to each physics analysis topic. Currently the following classes do exist:

- TopAnalysis: top quark related analysis,
- JetAnalysis: jets related analysis,
- TauAnalysis: τ -lepton related analysis,
- WAnalysis: W-Boson related analysis,
- ZAnalysis: Z-Boson related analysis,
- ZplusJetAnalysis: Z-Boson and additional Jet analysis,
- UEAnalysis: analysis for underlying events,
- User Analysis: class skeleton to allow for easy implementation of user analysis.

It is planned to ass further classes.

All those classes inherit from the class BaseAnalysis, which provide the base interface functions

- Init(),
- Process(HepMC::GenEvent* hepmcevt),
- Finalize().

and common algorithms such as jetfinder and missing transverse energy \cancel{E}_T calculations. In addition it has special functions, e.g. for particle identification.

5. Steering mechanism

This package provides a homogenous configuration mechanism for all supported generators. For this purpose a class¹ reads a global configuration file (for example `Process.config`), which hold the following information:

- configuration file for generator parameter, e.g. physics process, pdf sets, center of mass energy,
- switch on/off analysis class and analysis class parameter settings,
- global settings, for example fiducial area, number of events to be generated, ROOT-Output filename.

The generator parameter are written in separate configuration files for each generator. Many generator configuration files and/or tunes² do exist for the supported generators. It is possible to specify a list of generator configuration files, which are then read in sequentially. The global as well as the generator configuration file is interpreted line by line.

Example of a global configuration file (`Process.config`):

```
# Nr of Events to be generated
nevents                100
# randomseed ...
# if no randomseed is specified time in seconds will be setntepre
rseed                  1234567

# The histograms will be saved into File
OutputFileName        Pythia6_diJet.root

# use the following Configuration files for the generator
# several may be specified
ConfigFileNames       configfiles/Pythia6_Common.config
ConfigFileNames       configfiles/Pythia6_PDF.config
ConfigFileNames       configfiles/Pythia6_top.config

# enable Analysis to be performed
dijet_analysis        false
z_analysis            false
wplusjet_analysis     false
tau_analysis          false
top_analysis          true
ue_analysis           true

# master switches for FSR, ISR and MI
FSR                   true
ISR                   true
MI                    true

# set Jet parameter, for diJet and WplusJet analysis
jet_coneRadius        1.0
jet_overlapThreshold  0.1
jet_ptmin             5.0
```

¹ the class is called `Configuration`

² a tune is a special set of parameter settings

```
# set other parameter
# fiducial eta range
max_eta 100
```

Example of a generator configuration file (Pythia6_top.config):

```
# The top settings

# Selecting ttbar events (number 6):
pysubs msel 6

# q qbar -> c cbar / b bbar
pysubs msub 81 1

# g g -> c cbar / b bbar
pysubs msub 82 1

# for decay of top
pydat3 mdme 46 1 1
```

6. Histogram comparison and webdisplay

After the histograms have been saved in ROOT files, they can be compared and displayed in an automated way. For this purpose the package provides a ROOT-macro³ to

- open ROOT files and extract histograms,
- extract and display the same histograms from input ROOT files in one Canvas,
- perform Kolmogorov- and χ^2 -test,
- create image files with statistics and test results,
- create html-page with results.

7. Installation

A tarfile of the package can be downloaded from the web portal[14]. In addition to the analysis classes and the example programs a setup script (setup.sh) and a Makefile is provided. The setup.sh script allows an automatic setup of the environment variables for all platforms supported by GENSER for computers in the desy.de domain and in the lxplus cluster[15]. To follow the design goals from Sect. 2 also the Makefile to build the class libraries and the example executables are kept simple (40 lines).

In addition the webportal provides the Doxygen documentation[16] for the framework and also links to already created validation/comparison distributions for the MC generators Pythia6, Pythia8, Herwig, Herwig++ and Cascade.

8. Conclusion

The HepMCAnalyser framework provides a lightweight and robust tool for MC generator analysis. It is simple and easy to install. It is designed and successfully used for regression tests and as a starting point for generator studies in private analysis, e.g. by students and during the MC schools. In addition the framework is used by GENSER for histogram based validation of generators as well as in ATLAS[17] for generator validation.

³ macro is named plot.C

References

- [1] <http://lcgapp.cern.ch/project/simu/generator/>.
- [2] M. Cacciari. Fastjet: a code for fast k_t clustering, and more. Technical Report hep-ph/0607071. LPTHE-P-2006-04, Paris 11. Lab. Phys. Thor. Hautes Energ., Orsay, Jul 2006.
- [3] R. Brun, F. Rademakers, and S. Panacek. Root, an object oriented data analysis framework. 2000.
- [4] <http://lcgapp.cern.ch/project/simu/HepMC/>.
- [5] <http://home.thep.lu.se/~torbjorn/talks/texel08a.pdf>.
- [6] J. L. Pinfold. The challenge of lhc experiments. 2004.
- [7] <http://public.web.cern.ch/PUBLIC/en/LHC/LHCExperiments-en.html>.
- [8] T. Sjostrand, S. Mrenna, and P. Z. Skands. Pythia 6.4 physics and manual. *J. High Energy Phys.*, 05(hep-ph/0603175). FERMILAB-Pub-2006-052-CD-T. LU-TP-2006-13):026. 570 p, Mar 2006.
- [9] T. Sjostrand, S. Mrenna, and P. Skands. A brief introduction to pythia 8.1. oai:cds.cern.ch:1064095. *Comput. Phys. Commun.*, 178(arXiv:0710.3820). CERN-LCGAPP-2007-04. LU TP 07-28. FERMILAB-PUB-07-512-CD-T):852867. 27 p, Oct 2007.
- [10] G. Corcella, I. G. Knowles, G. Marchesini, S. Moretti, K. Odagiri, P. Richardson, M. H. Seymour, and B. R. Webber. Herwig 6.3 release note. herwig 6.3. Technical Report hep-ph/0107071. CAVENDISH-HEP-2001-08. CERN-TH-2001-173. DAMTP-2001-61, CERN, Geneva, Jul 2001.
- [11] S. Gieseke, D. Grellscheid, K. Hamilton, A. Ribon, P. Richardson, M. H. Seymour, P. Stephens, and B. R. Webber. Herwig++ 2.0 release note. oai:cds.cern.ch:986755. herwig++ 2.0. Technical Report hep-ph/0609306. CERN-PH-TH-2006-182. CAVENDISH-HEP-2006-18. IFJPAN-IV-2006-6, CERN, Geneva, Sep 2006.
- [12] H. Jung. Ccfm prediction on forward jets and f_2 : parton level predictions and a new hadron level monte carlo generator cascade. (hep-ph/9908497), 1999.
- [13] T. Gleisberg and F. Krauss. Automating dipole subtraction for qcd nlo calculations. *Eur. Phys. J. C*, 53(arXiv:0709.2881):501–523. 42 p, Sep 2007. Comments: 42 pages, 8 figures.
- [14] <http://hepmcanalysistool.desy.de/>.
- [15] <http://plus.web.cern.ch/plus/>.
- [16] <http://www.doxygen.org/>.
- [17] *ATLAS detector and physics performance: Technical Design Report, 2*. Technical Design Report ATLAS. CERN, Geneva, 1999. Electronic version not available.