

# Rivet in a nutshell

**Hendrik Hoeth**

Lund University

+ Rivet groups

(Berlin, CERN, Durham, Lund, UCL)



# Overview

- The Idea Behind it
- Design Overview
- Brief Usage Guide

# Physics Validation and Global Comparisons

Useful validation needs to use a wide range of experimental data.

Systematic global validation is essential for generator development and tuning.

*Rivet* is a validation/comparison framework for generators, based on the HepMC event record. It is used for example to

- ensure that generators describe a wide range of data
- regression-test generators between releases
- provide input for tunings

Used by MCnet generator authors (and being introduced in ATLAS, ...).

# Rivet analyses

Analyses from  $\sim 40$  publications already implemented (i.e. hundreds of distributions), and more are being added all the time:

*LEP*: event shapes, frag. functions, flavour spectra, multiplicities

*Tevatron*:  $W/Z$  + jets, jet shapes, jet cross-sections, jet angular correlation,  $W/Z$   $p_T$ , underlying event analyses in min bias, Drell-Yan, jets, etc.

*HERA*: jet energy flow, resolved photoproduction

We are working on more Tevatron analyses, but also on  $b$ -factory hadron spectra, RHIC  $pp$  data, ...

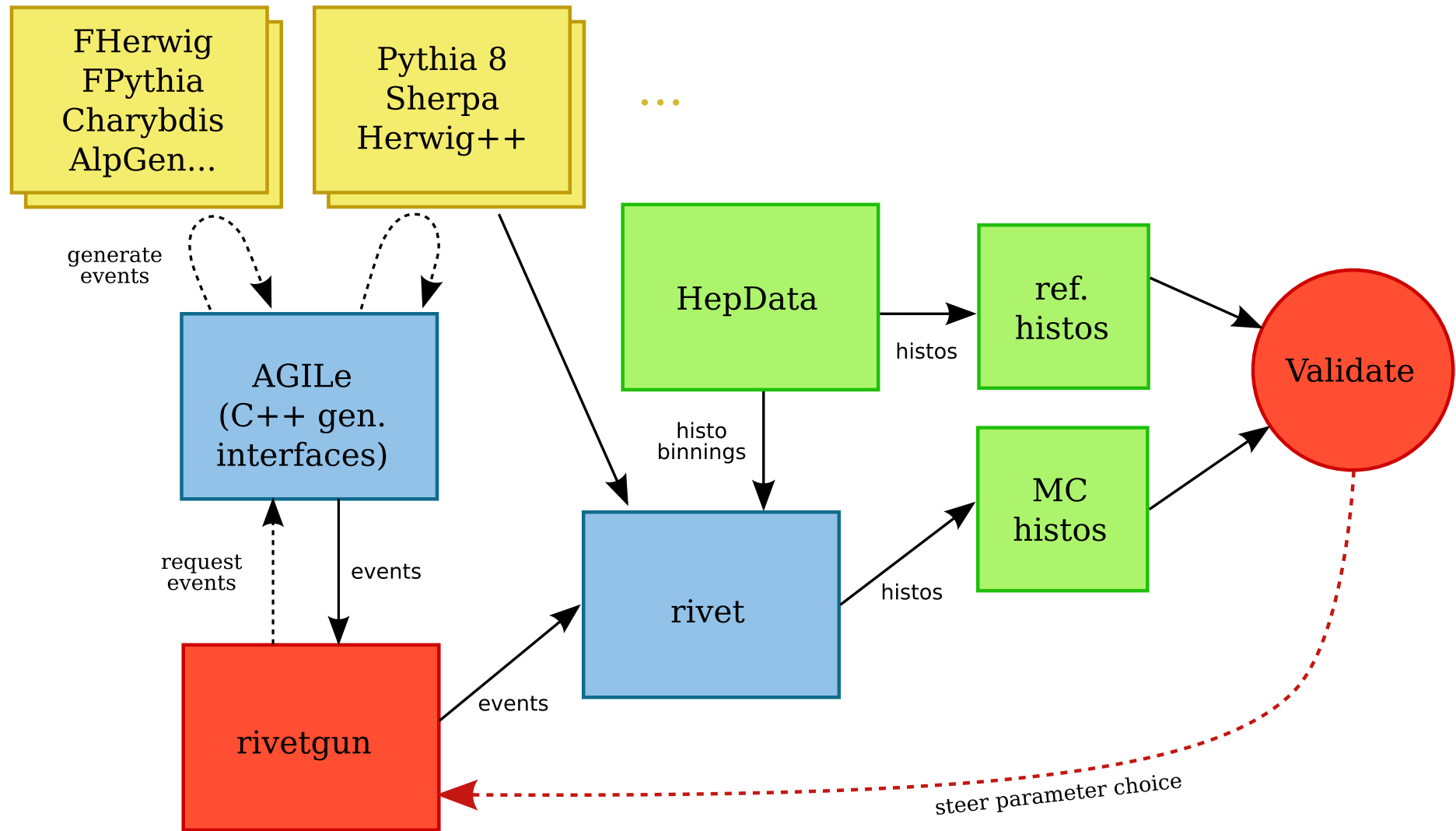
# Rivet Design

Rivet is a library of tools (event shape calculators, jet algorithms, final state definitions, ...) and so far about 40 analysis routines which use them.

- Command line tools for running generators, analysing events, making comparison plots
- Histograms can be auto-booked from reference data files
- Observables are automatically cached
- User analyses loaded as plugins
- Reference data is included in the Rivet release
- Output in AIDA format – can be converted to ROOT or plain text files



# Design Overview



# Modes of Running Rivet

Basically there are three different modes of running Rivet:

1. Run over events stored in a HepMC file
2. Read HepMC events from a pipe while running the generator on-the-fly – recommended for the C++ generators and what we will be doing in the tutorials
3. Produce events on-the-fly and let rivetgun steer the generator via AGILE – needed mainly for the Fortran generators

# A Few Command Line Options

You get all command line options by running `rivetgun -h`.  
Today and tomorrow we need:

- a select an analysis (can be used more than once)
- i specify the name of the HepMC input file
- H specify the name of the AIDA output file

Some important options for running Fortran generators through Rivet:

- g choose a generator
- P specify a text file containing generator parameters
- p change a parameter on the command line
- n number of events to produce



# Where to Find Things

The analysis code is located in `src/Analyses/` and `include/Rivet/Analyses/` in the Rivet source tree. Most analysis header files contain comments with some meta information on the analysis, including a *very* brief summary.

Users can develop their own analysis plugins in the `plugindemo/` directory.

Analysis names have a common format:

`<experiment>_<year>_S<spires-ID>`

For example `CDF_2001_S4751469` is a CDF analysis published in 2001 and you can find the paper and the HepData entry by entering `find key 4751469` in Spires.

## More Information

Documentation, contact information, download etc:  
<http://projects.hepforge.org/rivet/>