Rivet: past and future

Chris Pollard
University of Glasgow

for the Rivet team

MCnet Meeting
2017 04 06

What is Rivet?

Robust **I**ndependent **V**alidation
of **E**xperiment and **T**heory

In other words: “toolkit for robust comparison of physics models to experimental data”

http://rivet.hepforge.org

What it's made of

1. C++/python framework for **developing and documenting particle-level physics analyses**

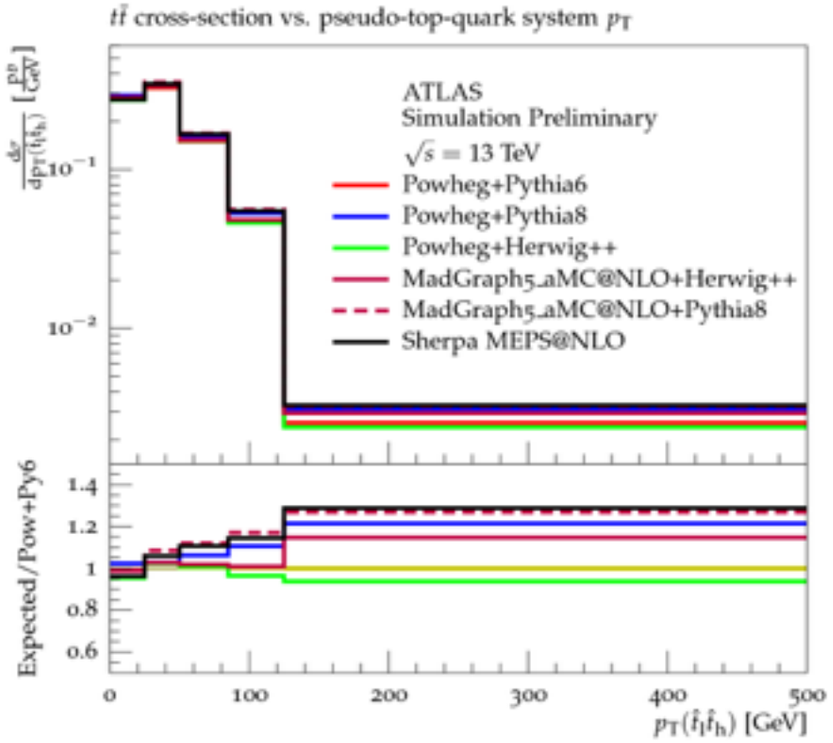
```
// initialize physic object definitions
// and histograms
void init() {
    // use all final, visible particles wi
    FinalState calofs(Cuts::abseta < 4.8);

    // build anti-kt 0.4 jets out of these
    FastJets fj(calofs, FastJets::ANTIKT,

    // declare these jets for use in selec
    declare(fj, "TruthJets");
```

2. Collection of **analyses** (code) and **corresponding data**

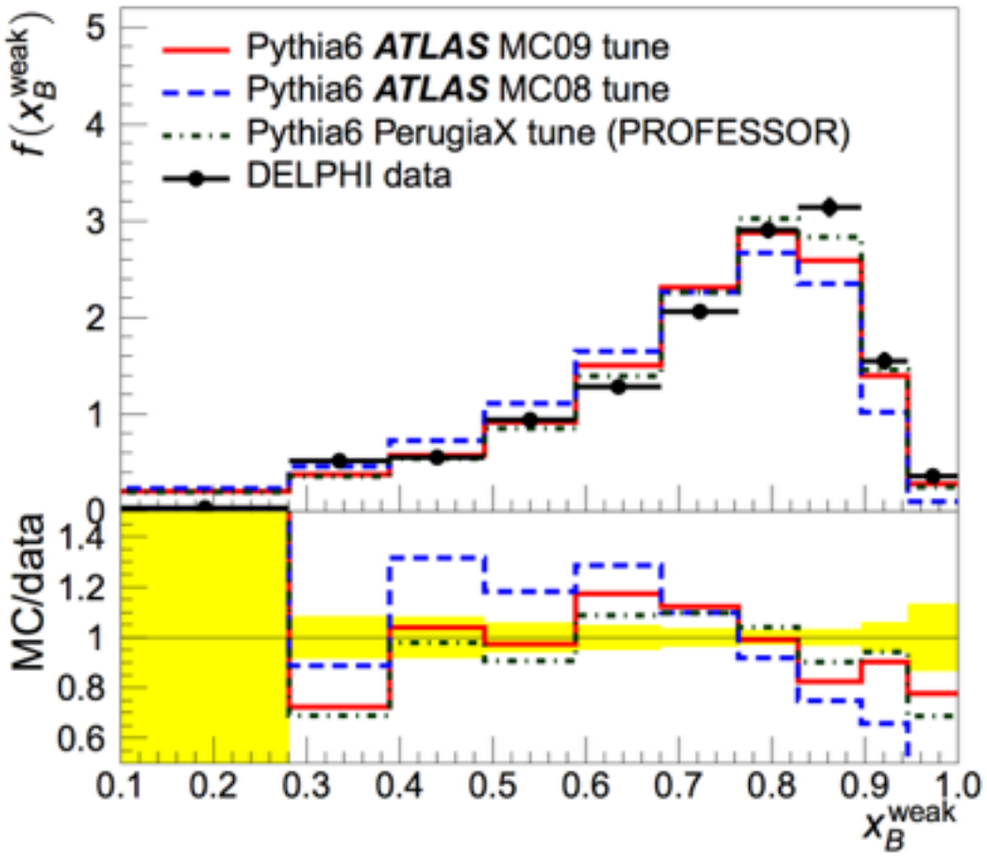
3. Tools for analysis object (e.g. histogram) manipulation, storage, visualization, etc. (**YODA**: http://yoda.hepforge.org)



Analysis preservation

- Rivet releases include both **the recorded data** and **the corresponding fiducial definition** (code).

- We work with both theory and experimental communities in order to **simplify**



1. preservation for experimentalists: short, understandable code, easy-to-use API

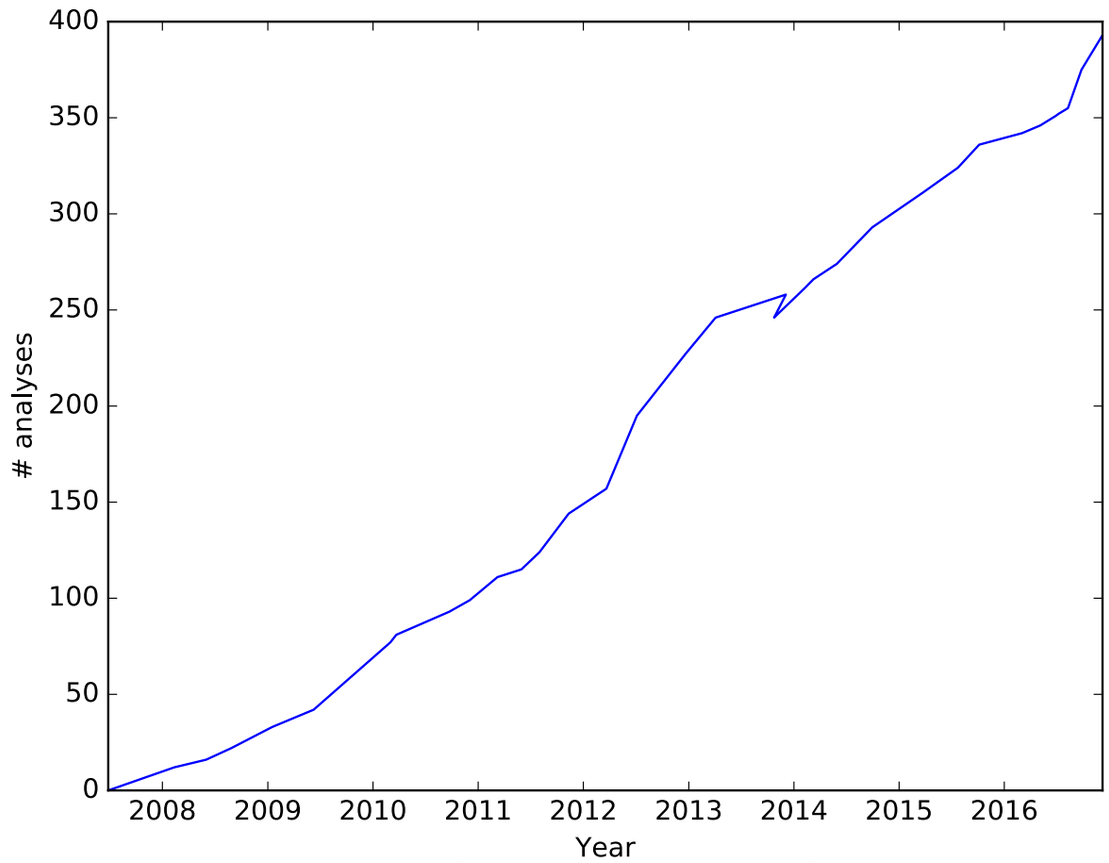
2. interpretation: easy-to-run analyses, useful output, automatic comparisons to data

How we do it

- Nowadays there is a dedicated **Rivet contact person** in each of the four major LHC experiments.

- This contact is responsible for **code style**, **validation** (analysis and data), **errata**, and **analysis submission**.

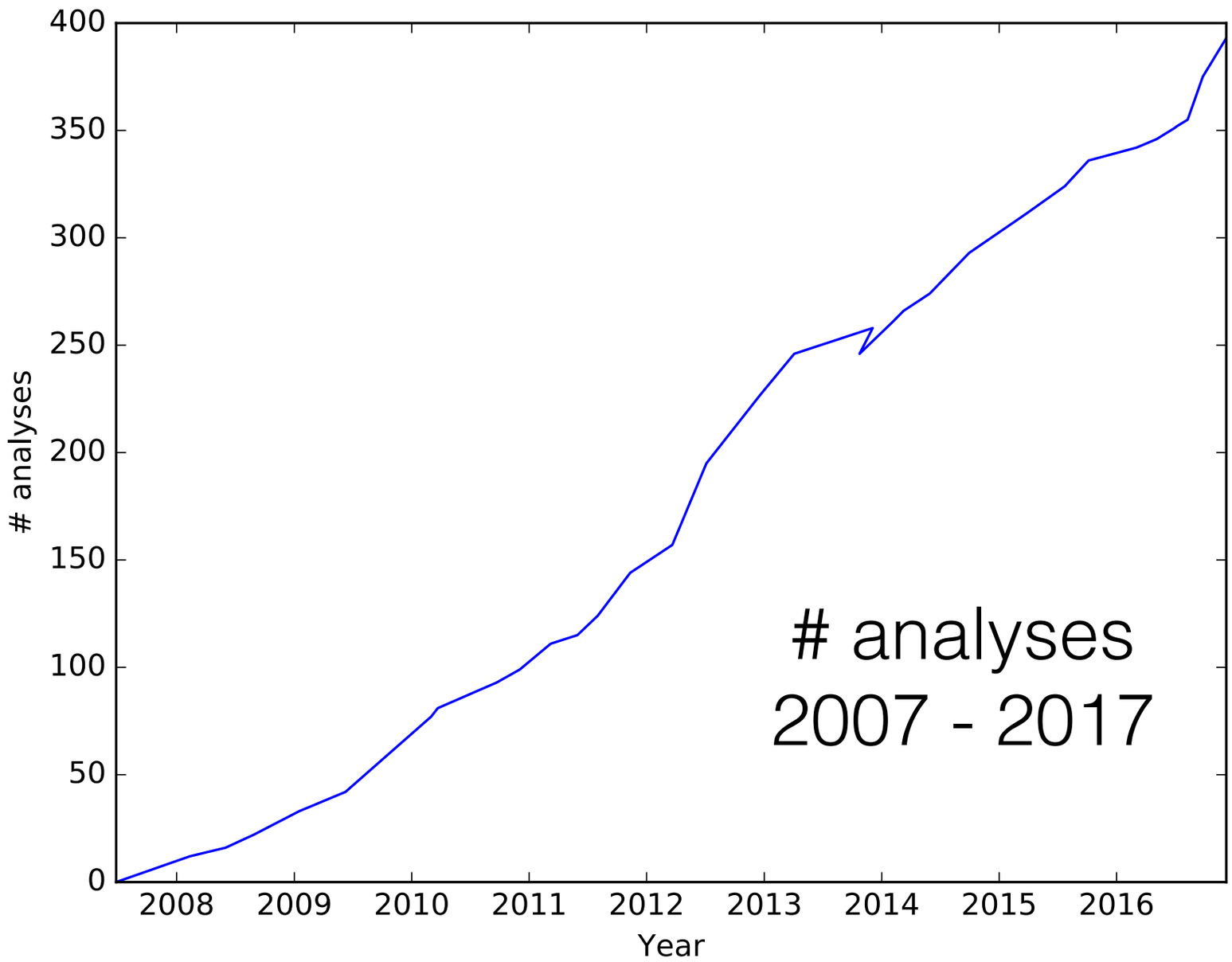
- After submission, **analyses are maintained by the Rivet team.**



analyses included in rivet releases 2007 - 2017

Analyses

now almost 400 analyses are included in latest release. (!!)



several are added with each release *by the experiment contacts*

now a requirement for publication in many experimental physics groups

PS: that's a lot of maintenance…

Why we need it

Rivet is the *de facto* standard for preserving HEP collider measurements (LHC+Tevatron+LEP+…).

At this point, **Rivet is critical for the preservation of HEP data for current and future interpretation.**

Rivet in experiments

Rivet is now also used for a variety of tasks **within the experiments**

- Validation of MC generation
- MC generator tuning
- Fiducial cross section and uncertainty determination
- Internal analysis preservation

Rivet in experiments

Rivet is now also used for a variety of tasks within the experiments

there is a strong push in the ATLAS SM and top groups (and from the physics modeling group) for analyses to write their fiducial definition (Rivet analysis) *from the beginning*.

this leads to

- **better-thought-out fiducial definitions**
- **observables which are known to be sensitive to discrepancies between calculations**
- **faster publication of the fiducial definitions**
- **better validation of MC samples**

I can't speak for the other experiments, but perhaps an exchange of ideas/infrastructure in this direction could be good.

rethink analysis distribution model?

- the current analysis distribution model involves:
  - someone contributes an analysis (preferably from the experiment)
  - the analysis is included in all subsequent Rivet releases
  - the analysis is maintained by the Rivet developers (e.g. API changes are propagated)
- so all analyses are (monolithically) included in a given release
- benefits:
  - users automatically get a huge number of analyses to run their models through
  - analyses should “just work” because they are shipped *with the library*

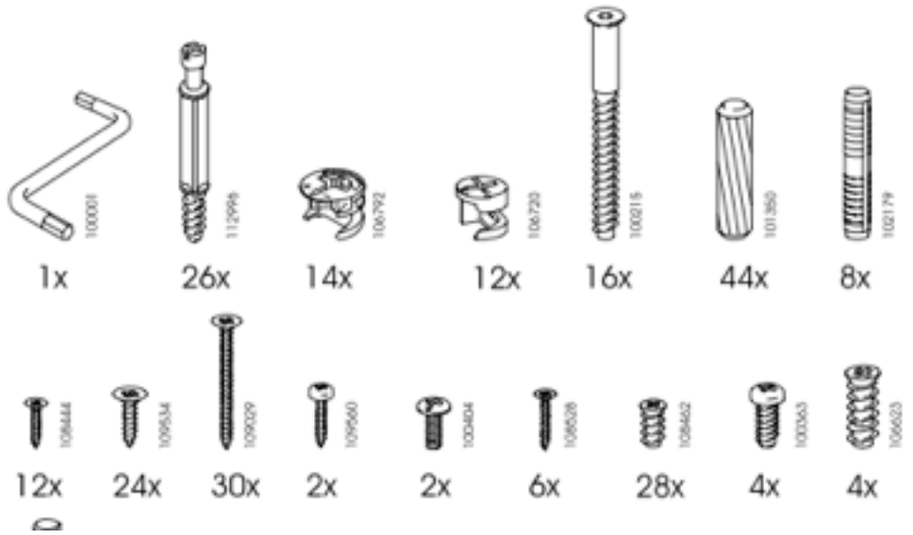
rethink analysis distribution model?

- drawbacks:
  - an analysis can only depend on particular libraries (those Rivet depends on)
    - some libraries move rather quickly (e.g. FJContrib)
    - some analyses depend on a *particular version* of an external library
- we've discussed the idea of having a non-centralized repository of analyses that users can "check out" and run, but there are clear logistical hurdles to be overcome…

Rivet: systematic weights

- Next major release (Rivet 3.0) will provide automatic handling of events with
  - **multiple weights** (eg. scale uncertainties, PDF variations)
  - **NLO sub events** (to be treated statistically as fully correlated)
- Each histogram, profile, etc. will have several versions stored—one corresponding to each event weight variation.

- **Rivet 3.0alpha1** already released (a very long time ago…) with an implementation for 1D histograms.
- Analysis code will *change very little*: won't explicitly handle the event weight.
- As always, **analyses already in Rivet will be updated by the Rivet team. (this has been time consuming…)**



Rivet: other plans I

There are several other new features/enhancements we are working on:

- "Re-entrant histogramming"
  - Can merge Rivet runs (even those with eg. histogram divisions)
  - Can plot histograms during the Rivet run to make sure they look ok (already partially implemented)
- Better and faster plotting tools based on common python libraries
- Much faster compile times (already partially there)

Rivet: other plans II

There are several other new features/ enhancements we are working on:

- HepMC 3 support
- Scatter plots with more than one associated uncertainty (eg. separate syst, stat)
- Thread safety for large parallel runs (already there to some degree)
- Luminosity info in analyses
- Support for running groups of analyses by *tag* (eg. all ttbar analyses, all dijet analyses)

Rivet: summary

- Hopefully that was a useful summary of recent and future developments in Rivet.

- As always, we are happy for feedback “both sides of the aisle”.

- Rivet and YODA websites with documentation: http://rivet.hepforge.org, http://yoda.hepforge.org

- Rivet developer mailing list: rivet@projects.hepforge.org