

Rivet monthly dev meeting

2 June 2021

Last-month's activity

- Not much? [Review MRs](#)
 - See to-do's from last time:
 - Louie: YODA API, only non-const annotation/error access method needed
 - ~~■ Chris G: checked and manually merged div-by-zero protection to master~~
 - ~~■ Frank S: ping Gurpreet re. CMS min-bias missing datasets~~
 - Analysis coverage script-migration finished, checked, and merged (AB, CG)
 - More Spires/Inspire aliasing and clean-up. "Only" D0 and CDF remaining. (AB)
 - Vector aliases, and removing eta finiteness hack (AB, LL, PR)
 - rivet-merge scale-factor syntax (CG)
 - LHCb analysis fix (Felix Kling, CG) -> **ongoing second-bin-width issues**

Next-release tasks

- **Issues for 3.1.5 patch release**

- Issue with YODA 2D histo filling?! (report by Joao Silva) -> AB to follow up
- Issue with DressedLeptons? Constructor? (flagged by Sophie Morard)
-> request full code
AB added missing DL origin vertex to the standard dR-clustering method

~~○ Issue with YODA Python3? Reported, no sign in CI, no reply~~

- Complete Spires/Inspire aliasing
- Merge new CMS and HERA analyses (Markus S, Hannes J) -> AB/CG, JB
- CMS minbias ratios?
- Encourage LHCb to submit their lots of nice HF jet analyses!!

<https://indico.cern.ch/event/1009701/timetable/>

Releases and work-plan

- **In parallel: focus on v3.2.0:**
 - time-saving no-copy of the HepMC event, etc.
 - jet clustering of any ParticleBase
 - deprecation clean-out and enum rationalisation (started)
 - add early versions of automatic “object flattening”, and no-width scaling?
We promised a discussion meeting on this...
- **In parallel:** plotting development (GSoC) and YODA 2 finishing
- ~~● Cross section default currently $\{xsec, xsecerr\} = \{0, 0\}$.
Wouldn't be $\{1, 0\}$ better for post-hoc scaling? Prefer to “overwrite” in rivet-merge?~~

Big picture tasks (near duplicate from April & May)

- **Stats objects are our major technical bottleneck**
 - Plotting API — **starts next week via 2 x GSoC students, Nick & Simon (AB & CB)**
 - YODA type extension (build on Nick R GSoC 2020 work)
 - HDF analysis data and new YODA format
 - `post-finalize()` always “flatten” stats objects to “binned measurement” type
 - [finish multiweight-fill optimisation (Aditya GSoC 2020)]
- **Scaling**
 - Analysis distribution system... again
 - Ref-data and analysis data particularly problematic: decouple data from code??
 - HD consistency
- **Standardising:**
 - MCnet weight-name/structure [proposal](#): **productive meeting on May 21, lots of agreement, AB to update and recirculate proposal**
 - [Event-record content: excessive size and physicality...]



Events

- **Organisation, meetings, etc.**

- ~~HF-QGP-HI workshop April 8~~ <https://indico.cern.ch/event/1022351/>
Report from Christian B; HepMC issues, non-MC model testing?,
expts to release no-data Rivet codes? **maybe** → **ISMD panel discussion...**
- [HepMC in Heavy Ion Collisions workshop, June 7th](#) — **OSCAR format converter?**
- Christian B Rivet tutorial at [RHIC/AGS User Meeting, June 8th](#)
- MCnet tutorials: China (June) and Dresden (Sept) — **AB RAL slides -> CG**
- End-June GDRQCD talks on overview+HI+ep — **volunteers? Louie ~agreed**
- MCnet “regional retreats” for CEDAR @ Les Houches?
Les Houches fully (?) cancelled, but [Scotland retreat](#), hopefully... TBA
- MCnet extended by 3 months — effectively to end-of-year 2021
more C19-safe workshop proposals welcome! also online tutorials?

BACKUP

Major (stats) work plans

- **“Flattening”**: convert finalize output to inert objects (scatters/binneameas)
 - Final objects really will mean “what was plotted/listed in the paper”
 - Allow eager conversion to solve “no-bin-width issue”
 - Best that we wait for binned measurement YODA2 types: no more scatters!
- **HDF5 analysis data machinery (Holger) *Status?***
 - Also interested in HepMC and YODA HDF5 formats
 - **Holger to ping CMS, prototype interface**
- **Plotting (Christian B et al)**
 - Plan: generate Python MPL scripts *without* TeX, .plot styles → YAML
 - Rivet labels tested: MathText fails due to missing std symbols. Can we extend?
 - Stalled for a while... restarting? Possible student help from David Grellscheid
 - **Christian to prototype the Python-script generation**
 - **Chris to extract weight-handling logic from rivet-cmphysos**

Performance in Rivet and YODA (Aditya Kumar, AB)

- **Profiling revealed bottlenecks: thanks Aditya!**

- HepMC ASCII I/O (obviously) — taken out of tests by event-reuse
- GenEvent copying — for sanitising, but hardly used: removed from Rivet.
Could/should generators write smaller “essential” events by default?
Awkwardness: we still normalise GenEvent units... so not quite analysing a const GenEvent.
But can't justify an expensive copy for *unit conversion*...
- PID functions — sped up charge lookups by special-cases. Marginal gain
- Multiweight calls to histo fill() *very* expensive: ~40-50% CPU!
100+ consecutive fills with same x: tried caching in YODA but no benefit:
cache-check costs the same as linear bin lookup! *Maybe cache in Rivet?*

- **Thread-safety.** *“Just store a ProjectionHandler in AnalysisHandler: easy!”...?*

- But then who do Projection constructors (recursively) register their contained projections with, before they themselves have been bound to a PH?
- “Declare queue” implemented: not yet working (thx, unique_ptr), but should do
What should the Projection ownership be?!

YODA generalised datatypes (Nick Rozinsky, LC, AB)

- **Long-understood limitations of YODA types and design**
 - Overreach in attempted non-factorisable binnings: composed 1D axes are fine
 - Complexity/mess in 2D overflows: need “infinity binning”
 - Need for binned “dead” data objects... or any type, actually
 - Want programmatic access to axis number and global/local bin indexing
 - Want labelled/discrete binnings as well as continuous
 - Code duplication, particularly in Cython interface building
- **Major YODA redesign using modern C++ magic. Thanks Nick!**
 - E.g. Histo1D → wrapper of a BinnedStorage<CAxis, Dbn<1>> + sugar
 - + arbitrary mixtures, e.g. 3D binnings of doubles, discretely labelled counters, ...
 - Adaptors used to map fill/set behaviours. Can do the same for I/O read/write?
- **Path to a YODA2 release:**
 - Needs I/O adaptors and user-facing refinements. Tie in with HDF5 format?