

Rivet monthly dev meeting

7 April 2021

Releases and work-plan

- **Rivet 3.1.4 + YODA 1.9.0 released last/this week** (huge thanks to Chris G)
 - A couple of weeks later than planned, but more content!
 - Fixes analysis builds issue for Genser/ATLAS, obviously. Many small tweaks
 - Also improvements in multiweight handling, and fast pairwise merging
 - CMS min bias analysis with custom data currently part-disabled
 - Pairwise yodamerge → split off stacking in new yodastack command (hence 1.9.0)
 - Also introducing Binned, Fillable, Scatter base classes, rmPoints method, etc.
- **Next: 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?
 - **In parallel:** plotting development (GSoC) and YODA 2 finishing

Big picture tasks

- **Stats objects are our major technical bottleneck**
 - Plotting API
 - 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:**
 - **Weight-name/structure [proposal](#)** drafted by Chris G and AB; needs final meeting
 - [Event-record content: excessive size and physicality...]

Events

- **Organisation, meetings, etc.**
 - HF-QGP HI workshop April 8 (tomorrow) <https://indico.cern.ch/event/1022351/>
 - HI workshop with RHIC as target audience June 8th (?)
 - STRONG-2020 (FAIR, HF combinations(?)) this spring, dates TBC.
 - GSoC 2021 in recruitment/testing phase:
 - Rivet plotting, ~~YODA~~ multiweight, and LHAPDF projects
 - MCnet “regional retreats” for CEDAR @ Les Houches?
 - Lots of last-minute MCnet CEDAR shortie applications, mostly south-Asian:
 - suggestion to run a workshop there for BSM/HI Rivet & Contur...
 - pandemic and MCnet funding extension permitting

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?