

Rivet developer & BSM workshop: intro and talking points

Andy Buckley
University of Glasgow

RIVET DEV / BSM WORKSHOP, LUNGA HOUSE, 27 SEPT 2016



University
of Glasgow



Welcome!

- ▶ **Welcome to Lunga House & Croabh Haven!**
- ▶ Everything is provided – hope you enjoy your time here and we get some nice work done/started
- ▶ Lunch at 1pm, dinner at 7.30pm
- ▶ Keep a tally of how much beer you drink and settle up before leaving! Hopefully there will be whisky...
- ▶ **10 spaces available on boat excursion to Corryvreckan whirlpool, Wed morning – takers?**

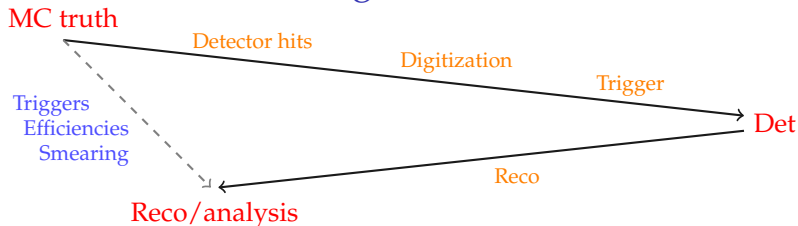
Discussion topics / groups

- ▶ BSM issues, e.g. detector smearing, experiment appeal/lowering barriers, ref data access, stat interpretation
- ▶ SM analysis methods, e.g. core projections, analysis details
- ▶ Statistical machinery, e.g. multi-weights, re-entry
- ▶ YODA stuff: multiple scatter errors, more?
- ▶ Others, e.g. HepMC3, zipped I/O, docs...

Demo / status update

- ▶ Smearing machinery
 - `SmearedParticles`, `SmearedJets`, `SmearedMET`
- ▶ Lambda filters and Cuts
- ▶ `Cuts::abspid`, `Cuts::charge3`, etc.
 - Neater than `IdentifiedFinalState`, `ChargedFinalState`, etc.!

BSM / detector smearing



Explicit fast-sim takes the “long way round”. Efficiencies dominate: no det-sim needed.

- ▶ Review and complete ATLAS/CMS detector eff/smearing functions – tracking effs & res?
- ▶ Make clustered leptons smearable – connect to `ParticleFinder` restructure
- ▶ `SmearedJets` tag efficiency – w.r.t. what baseline defn of “*b*-jet”?
- ▶ Wrap-around limits machinery, LHADA, ref data and correlations?

SM/core physics machinery improvements

ParticleFinder and FinalStates

- ▶ Rationalise `Particle` and `Finder/FinalState` interfaces
 - Uniform access to `particles()` (semantic, maybe composite) and `rawParticles()` (with `GenParticle`)
- ▶ `FastJets` etc. to use `ParticleFinder`, NOT `FinalState`
- ▶ `ParticleFinder::clear()` and `PF::particles(SEL, SORT)`,
`JetAlg::jets(SEL, SORT)`
- ▶ Deprecate `ChargedFS`, `IdentifiedFS`, `VetoedFS`, etc.?

SM/core physics machinery improvements

Boson, jet and MET finder refinements

- ▶ Promptness and ***Finder** (David Yallup)
- ▶ **WFinder**: `rm constituentNeutrinos()`? `rm WFinder???`
- ▶ **ZFinder** → **Dileptons** with backward alias?
- ▶ **WFinder** and **ZFinder** multi-lepton modes (via `cuts` or new enum classes?)
- ▶ **FastJets** built-in jet p_T subtraction option ⇒ how reliable?
- ▶ $\sum p_T$ (vec *and* scalar?) on **MissingMomentum**?

SM/core physics machinery improvements

Misc – too much detail!!

- ▶ More dual use of `cut` and `FN` in Projection constructors, storage and `cmp` as `std::function`
- ▶ `CutFn` with casts to `std::function`?
- ▶ `reversed()`, `sorted()`, and `i`-variants?
- ▶ `isnan`, `min`, `max`, etc. – why the trouble with requiring explicit `std::?`

Multi-weights and all that

- ▶ Multi-weight computation and delay
 - Using sleight of hand with histogram ptrs, system handles weights
- ▶ “Re-entrant” histogramming: init histo state from (merged) YODA file
 - Allow merging of distinct runs before finalize, e.g. HI
- ▶ Re-running finalize
- ▶ NLO counter-event groups and resolution issues

Distribution

- ▶ Big analysis collection is unwieldy
- ▶ And makes development awkward, due to length of full rebuilds
- ▶ Want to decouple analyses from core – but still retain control for maintenance etc.
- ▶ Could use an hg repo with tagging to indicate analysis/Rivet version compatibility?
- ▶ Integration with HepData/Invenio etc. repositories... can that work?
- ▶ Add “Luminosity” and “Tags/Keywords” info keys – *ability to run analysis groups by tag*

Other stuff

- ▶ Plotting: replace make-plots with something...
 - programmable, fast, high-quality, flexible, etc., etc. Matplotlib?!
- ▶ YODA multiple scatter errors – without making simple case awkward
- ▶ YODA restructure – expose binning machinery (for arbitrary types), use inheritance more
- ▶ Thread-safety for embedding
- ▶ Review vectors, matrices, etc. – can we drop GSL dependency?
- ▶ Python 3 and Cython compatibility