Rivet tutorial introduction

Christian Bierlich, Raghav Kunnawalkam Elayavalli, Antonio Silva
bierlich@thep.lu.se
Lund University

April 8th, 2021, RHIC & AGS Annual Users' meeting 2021

The plan

- Introduction to Rivet and streamed tutorials.
- The session is recorded, so you can go back.
  1. Christian: *Introduction to Rivet*.
  2. Raghav: *Your first analysis*.
  3. Antonio: *Using centrality in Rivet*.
  4. Raghav: *More advanced concepts.*
  5. Christian: *End and final questions.*
- No direct support with installation today. But if you want to "write along":
- Use Raghav's installation on RCF.

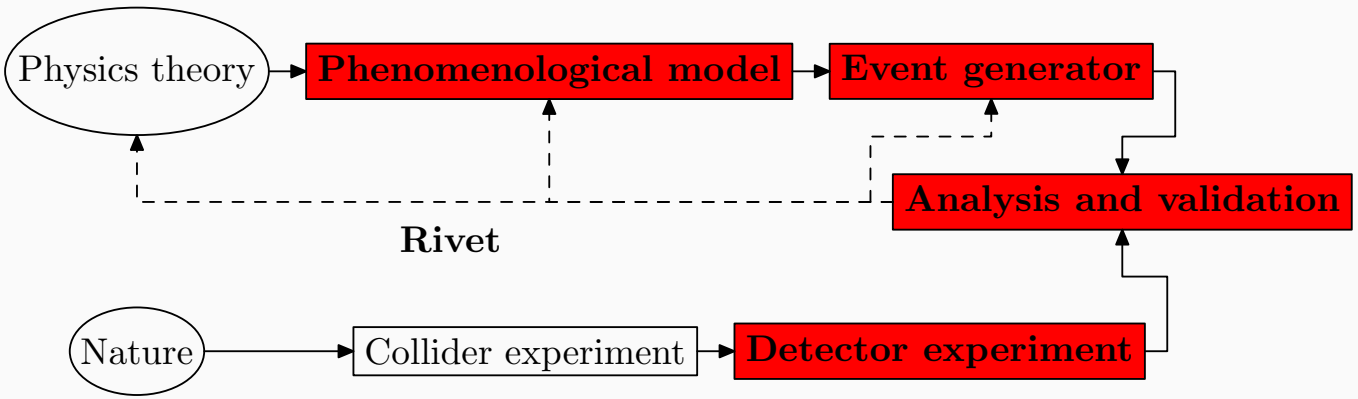
```
bash
./gpfs01/star/pwg/elayavalli/setuprivet313.sh
```

- (on Linux machine): Download and run bootstrap installer:

```
https://gitlab.com/hepcedar/rivetbootstrap/-/blob/3.1.4/rivet-bootstrap
chmod +x rivet-bootstrap
./rivet-bootstrap
```

The big picture

- Rivet is a *language* facilitating communication between:
  1. experiment & pheno.
  2. pheno & pheno.
  3. experiment & experiment.
  4. experiment & future experiment.
- Point is to ensure common definitions (as in any language).
- Consistently $\approx 50\%$ of MCnet (montecarlonet.org) papers refer to Rivet.



A bit of history

- Born out of HZTOOL: HERA (H1 and ZEUS) probing low-x physics.
  - Many models only implemented as MCEG.
  - Complicated kinematics, are there any apples for comparison?
  - Designed for concept-driven cross-experiment, cross-generator comparison, with difference in details (particle level cuts).
- Lessons:
  - Driver for progress: Best way to end a discussion is to reproduce a key plot!
  - Model independence: Model dependent observables are bad for MCEG. Might also be unphysical.
  - Easy predictions: Ensure that an observable is actually *observable*.
  - Standardisation: Common, evolvable interfaces are key.
  - Modularisation: Keep analyses separate, allows interface to grow. Must be scalable.

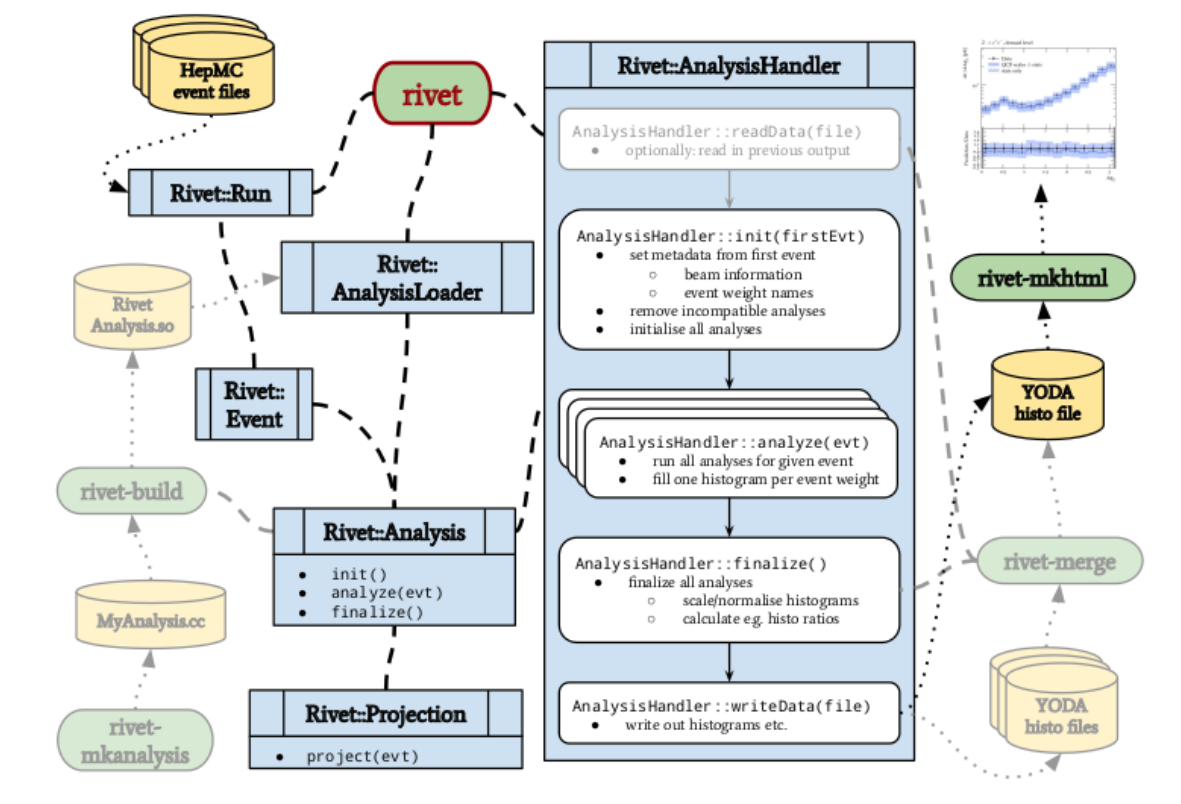
Rivet design

- Language: C++ with Python interface; Dependencies: yoda (histograms), HepMC (event format), FastJet (jets and event shapes). No generator dependencies.
- Core vs. analyses: Common functionality supplied by Rivet, analyses as pluggable modules by users.
- Division of tasks: Experiments validate analysis correctness, Rivet dev team keeps the code running with updates.
- Projections $\mathcal{O}(kN) \rightarrow \mathcal{O}(N)$:
  - Event properties calculated once, should not be calculated again.
  - "Final states" re-usable across many analyses.
  - Very scalable!
- Data synchronization:
  - Data points synced with/taken from HepData.
  - Ensure consistency, allows errata.
  - Auto-booking based on HepData records:

```
book(hist, "hepdata-id");
```

Robust Independent Validation of Experiment and Theory

- Standard validation package for MCnet generators.
- Version 1 in 2010 for LHC Run 1.
- Present: Rivet v. 3 (June 2019) (arXiv:1912.05451) (now: 3.1.4).
- Improved: documentation, tutorial, validation, docker images, gitlab hosting and many new physics features.



Utility for experimentalists

- Preservation: Store your analysis *once*, and others will maintain it.
- Reproducibility: What happens when your student graduates?
- Ensure that your results are used.
- Don't leave it to theorists to re-implement your analysis!
- "Do upon others...": Generate MC tunes using other people's work!

Utility for experimentalists

- Preservation: Store your analysis *once*, and others will maintain it.
- Reproducibility: What happens when your student graduates?
- Ensure that your results are used.
- Don't leave it to theorists to re-implement your analysis!
- "Do upon others...": Generate MC tunes using other people's work!

Can I be sure that the framework will live on?

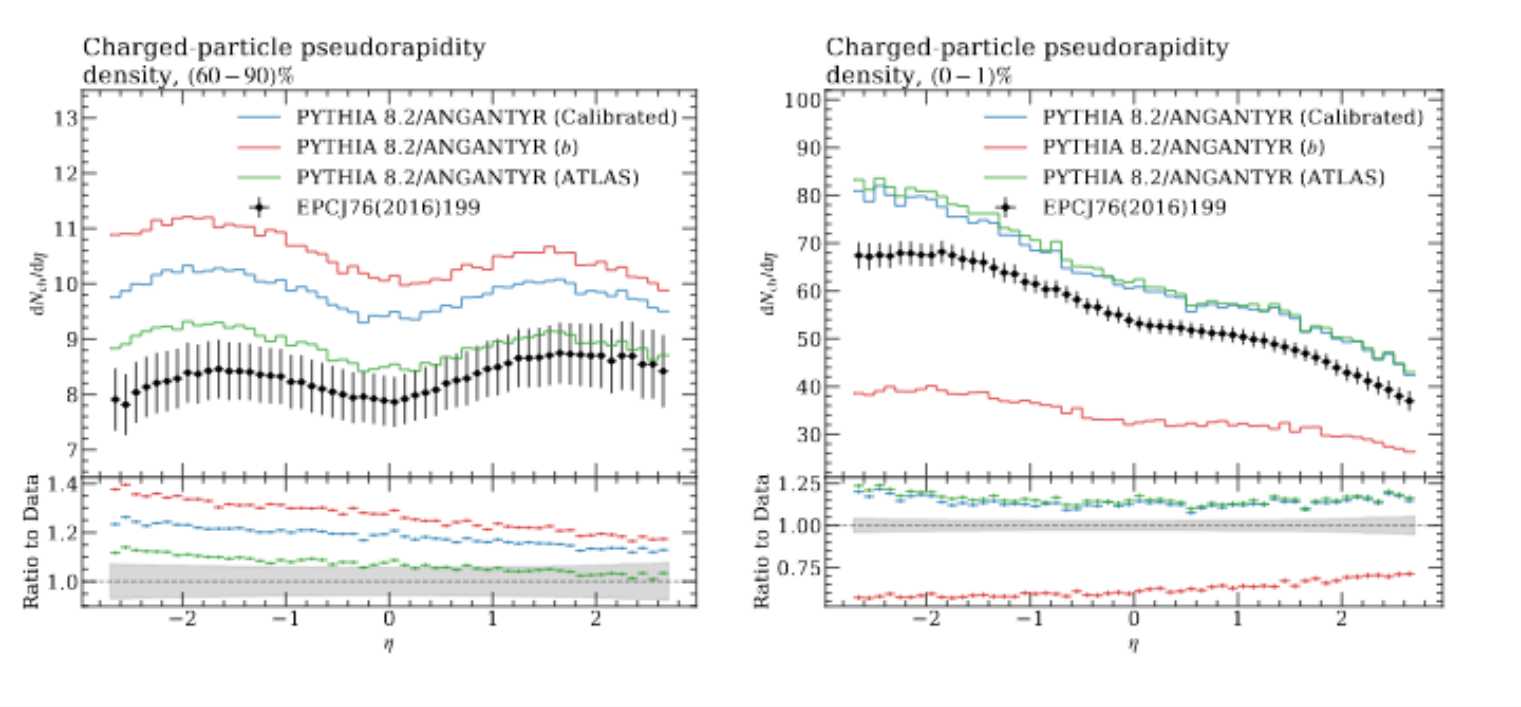
Yes! Large investment by HEP community and MCEG authors. $\mathcal{O}(1000)$ analyses already implemented. Dev team open for new directions: If a feature is needed, we might find a way.

New features example: Rivet for Heavy ions

- Good example: Recent venture into heavy ion physics: Rivet for Heavy Ions (2001.10737 [hep-ph])
- Rivet for heavy ions is/was:
  - ◇ A dedicated crunch towards including HI functionality.
  - ◇ Included several people from both sides.
  - ◇ Documented in the paper above, and included in Rivet proper.
  - ◇ Not a done deal. Many potential improvements possible.
- Rivet for heavy ions is *not*:
  - ◇ Something separate from Rivet proper.
- Result: Features to allow comparison between heavy ion data and MC.

Utility for theorists, honest data validation: Centrality

- Can't do HI without centrality.
- Theory level definition not the same as experimental.
- Subtle biases quantified: especially in *p*A.



Rivet for HI

◇ Includes centrality calibration.

◇ Introduce analysis options to select calibration.

Honest data validation II: Flow

- Key heavy ion observables:

$$E\frac{\mathrm{d}^3N}{\mathrm{d}^3p} = \frac{1}{2\pi}\frac{\mathrm{d}^2N}{p_\perp \mathrm{d}p_\perp \mathrm{d}y}\left(1 + 2\sum_{n=1}^{\infty} v_n \cos([n(\phi - \Psi_n)])\right).$$

- Naively, but cumbersome:

$$\langle v_n^2 \rangle \approx \langle \cos(n(\phi_1 - \phi_2)))\rangle = \langle \exp(in(\phi_1 - \phi_2))\rangle$$

- Rewrite with $Q_n = \sum_{k=1}^{M} w_k \exp(in\phi_k)$, it turns out that all harmonics to all orders can be rewritten like this, eg.

$$\langle 2 \rangle_n = \frac{|Q_n|^2 - M}{M(M-1)}.$$

- Non-flow reduced by increasing orders, or requiring event gaps.
- Framework gives massive speedup over naive methods.

Flow observables – generic framework implementation

Rivet for HI

- ◇ Generic framework and add-ons (1010.0233, 1312.4572).
- ◇ Calculate any $\langle\langle M \rangle\rangle_{m,n}$.
- ◇ Automatic subtraction of lower orders and error calculation.

Flow observables – generic framework implementation

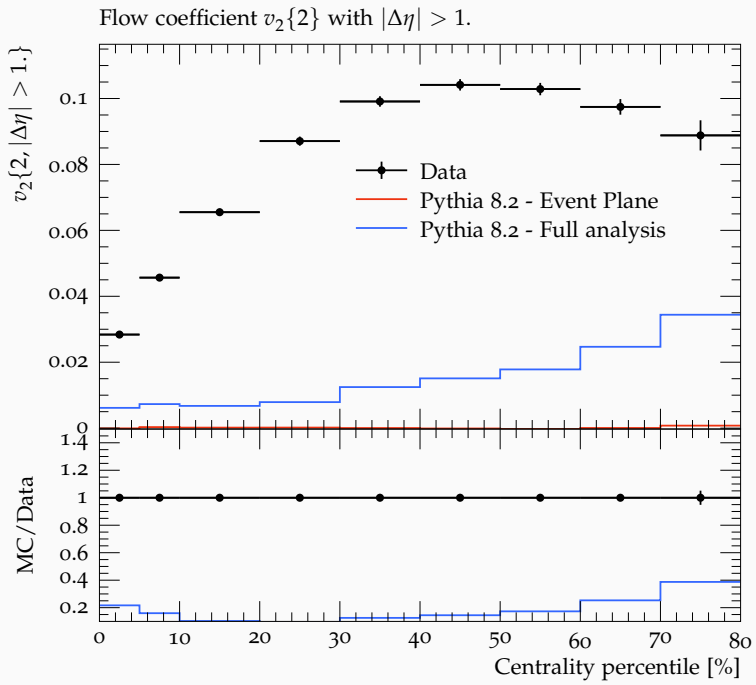
Rivet for HI

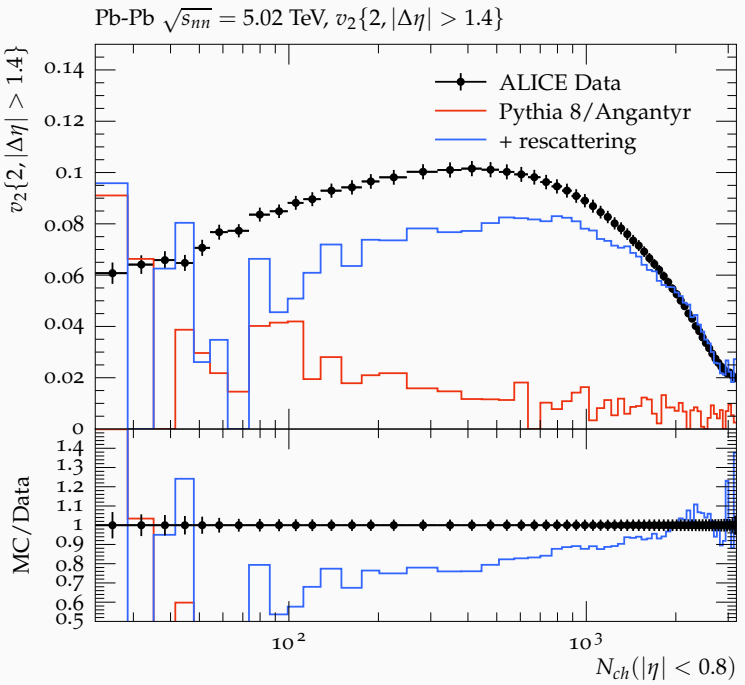
- ◇ Generic framework and add-ons (1010.0233, 1312.4572).
- ◇ Calculate any $\langle\langle M \rangle\rangle_{m,n}$.
- ◇ Automatic subtraction of lower orders and error calculation.

```
hc24 = bookScatter2D("c24",120,0,120);
ec22 = bookECorrelator<2,2>("ec22",hc22);
ec24 = bookECorrelator<2,4>("ec24",hc24);
...
ec22->fill(...);
ec24->fill(...);
...
// c_n{4} = <<4>>_{n,-n} - 2 * <<2>>_{n,-n}
cnFourInt(hc24, ec22, ec24);
```

Flow observables – Sample results

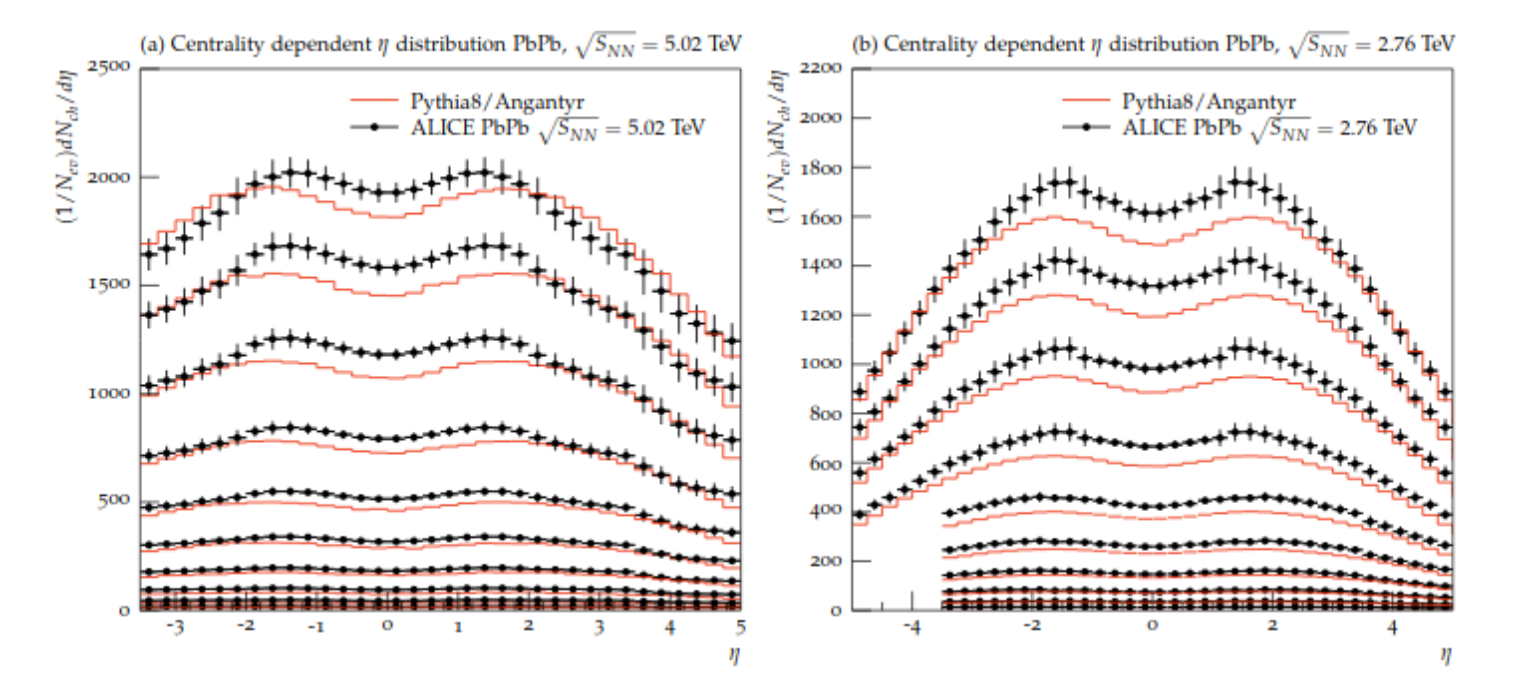
- Some HI analyses implemented, here: `ALICE_2016_I1419244` and `ALICE_2019_I1723697` (from 2103.09665).
- Correlators and cumulants can also be plotted without data.





Honest data validation III: triggers and particle definitions

- Correctness is important. Another example (Angantyr: 1806.10820 [hep-ph])
- Both are 10% effects, same as MC accuracy.



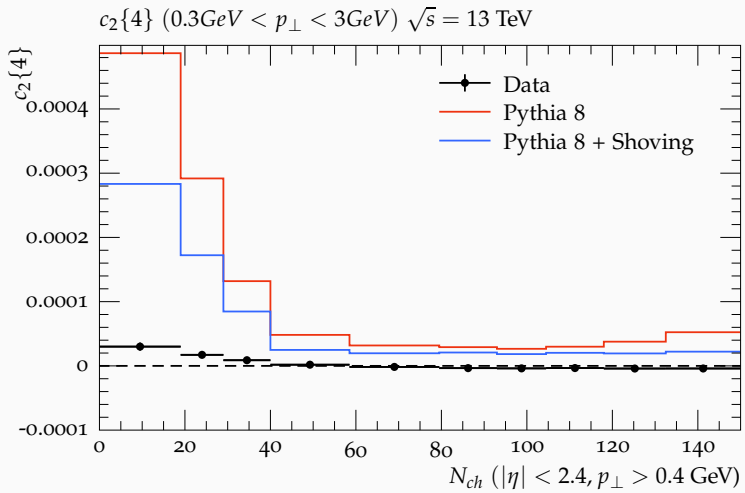
Rivet for HI

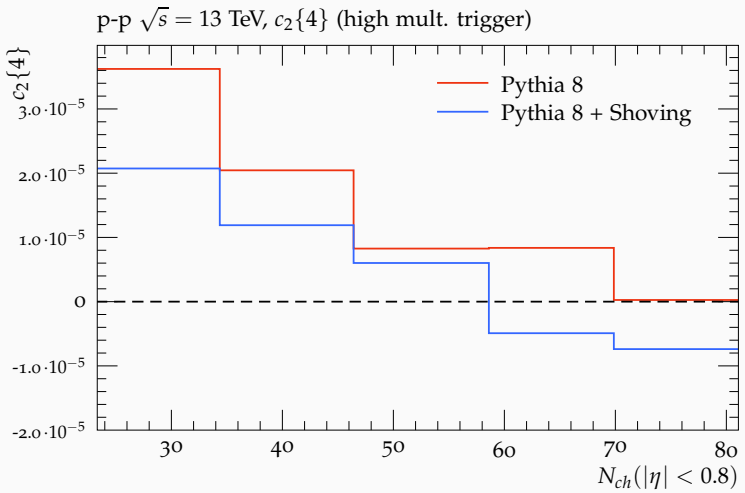
◇ Includes `ALICE::` trigger projections.

◇ Includes `ALICE::` primary particle projections.

Triggers and particles cont'd

- Complexity of observables aside...
- Small systems results particularly sensitive to trigger choice!
- Hot topic example (from 2010.07595 [hep-ph], data CMS)





- Physics interpretation depends on low-level experimental choice!

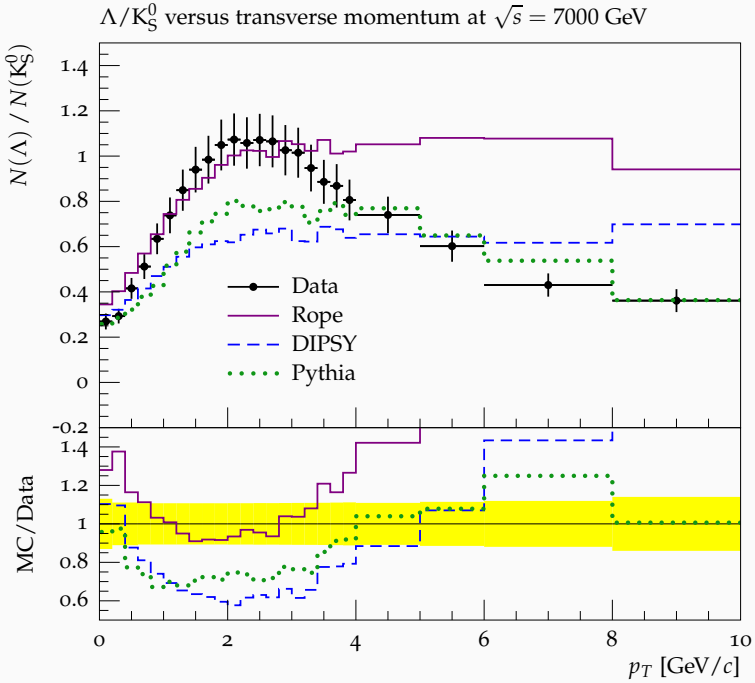
DIS and towards EIC

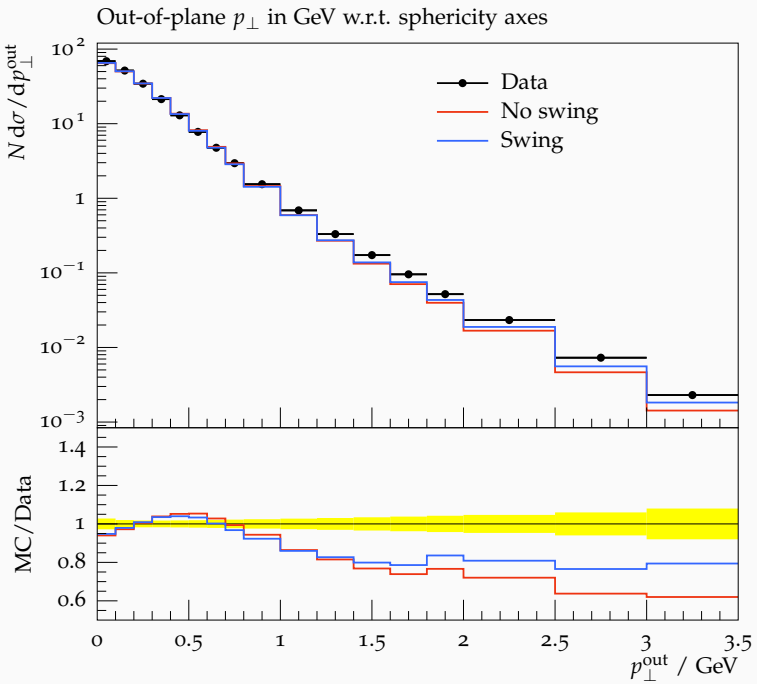
- Correct definitions of DIS kinematics (heritage from HZTool).
- Dedicated `DISKinematics` projection.
  - Remove DIS lepton.
  - Calculate event kinematics.
  - Boost to right frame.
- Ongoing efforts by dedicated group (JLab) to port existing analyses.
- Possibility to get in on the ground floor!
- Basic EIC tutorial available (self study):

https://gitlab.com/hepcedar/rivet/-/blob/release-3-1-x/doc/tutorials/

Use as model development tool

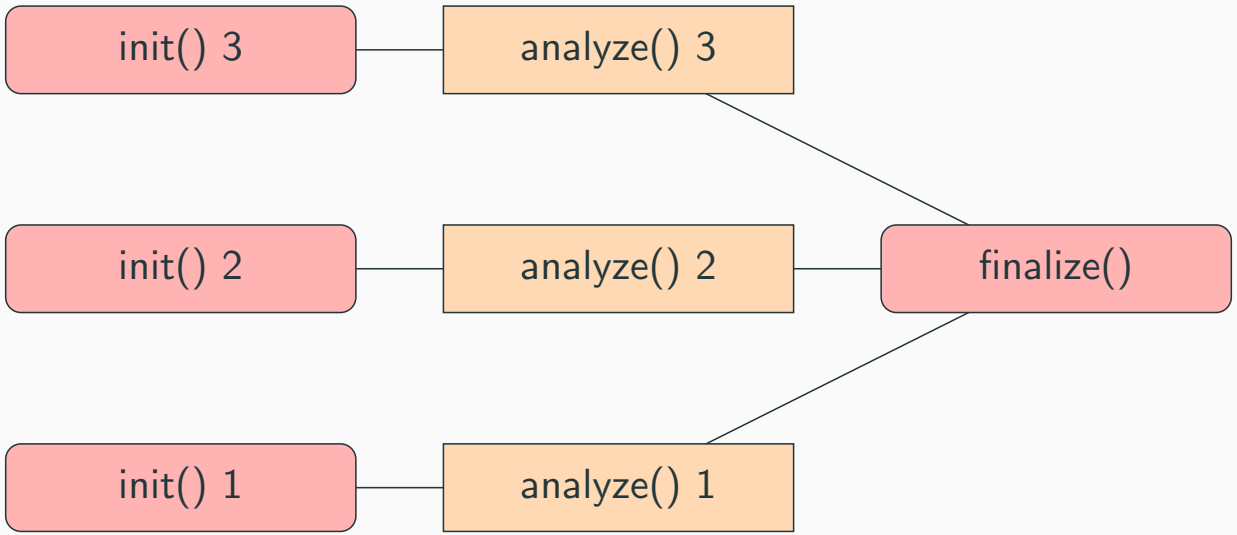
- Seeds test driven development: Sometimes your idea needs help.
- Provides a target, but also baseline which should not be destroyed.
- Prevents "single-observable" models and over fitting.
- Data from CMS and DELPHI (example from 1412.6259 [hep-ph]).





“Big data” I: perfect run combination

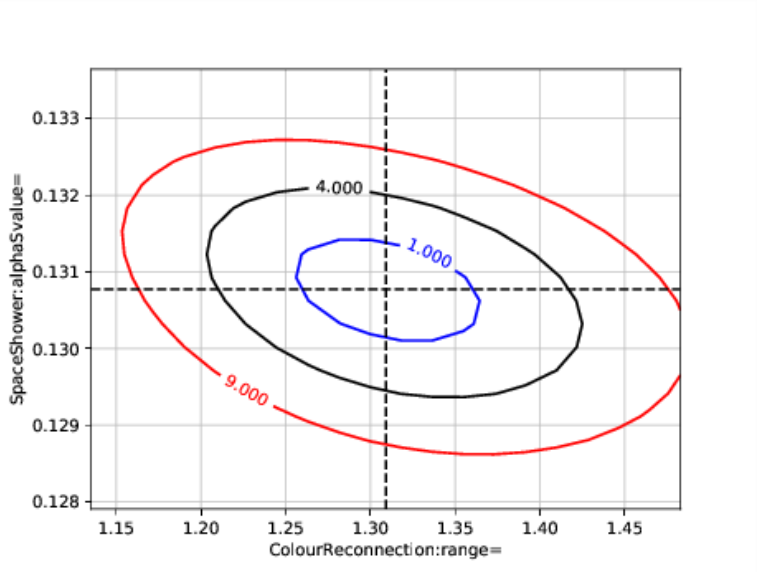
- Parallelization is necessary but potentially difficult.
- Old solution `yoda-merge` only for special cases.
- Consider: flavour ratios, R_{AA}, flow...
- Solution: `rivet-merge` before finalization.



- Let analyser implement merging $\rightarrow$ *perfect run combination*.

Big data II: Generator tuning

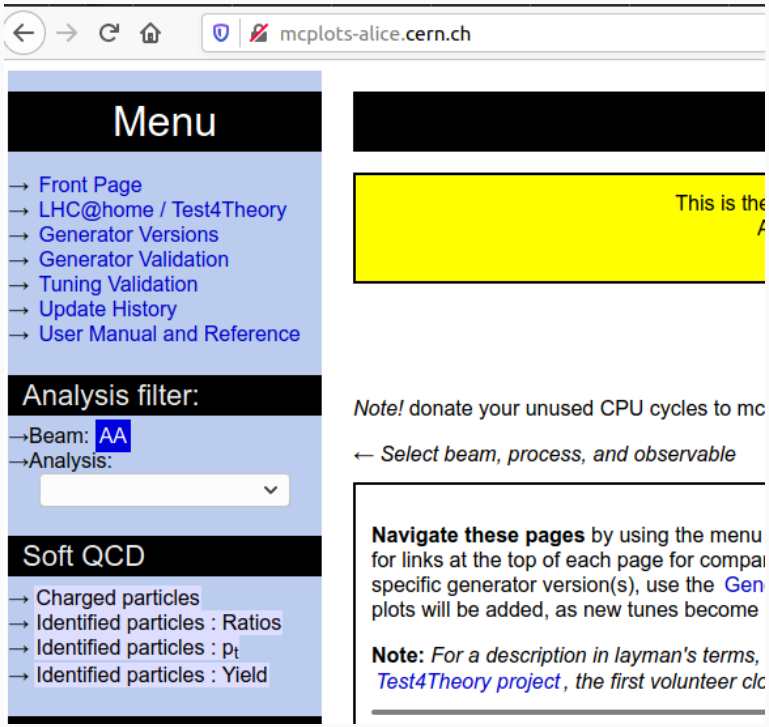
- With many available analyses comes possibilities.
- Systematized generator tuning is one! (https://professor.hepforge.org/)
- This is not a tuning talk, but...



- Future ALICE efforts possibly include compatibility of freezeout models.
- Full statistical framework for free! Large scale tests of QGP models? (like Contur for BSM)

Big data III: MCplots

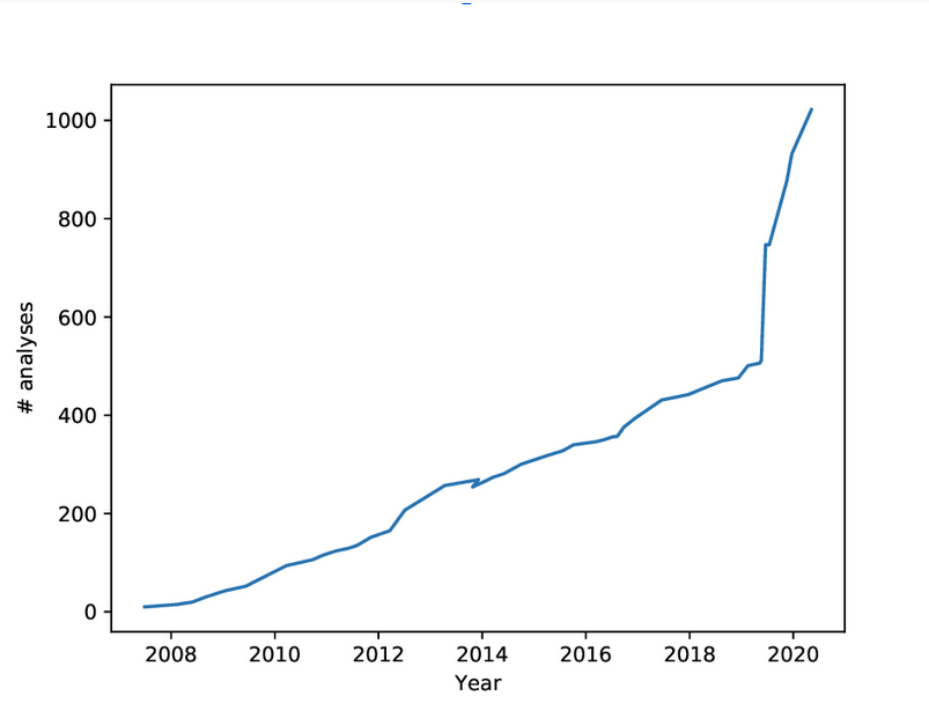
- Collection of data together with many MC tunes and types `mcplots.cern.ch`.
- Pilot heavy ion project by ALICE `mcplots-alice.cern.ch/`.



- Systematic inspiration for models and measurements.
- Huge opportunity for guiding further research for exp and pheno.

Large user base, many analyses and applications

- Vision: standard toolkit for "truth level" observables across collision systems and colliders.
- Contributions from many communities, active user base.



- Activities:
- Part of publication procedure for ATLAS and CMS.
- Many new initiatives and buy-in from ALICE.
- Ongoing efforts for RHIC experiments and EIC.
- Analysis contributions from NuSea, LHCb, LHCf, TeVatron, UA5, NAXX ... (and many, many more)
- Standard for MCnet event generators, more adding support.
- "Big data": Professor (tuning), MCplots, PDFs, TopMass fitting, Contur (BSM), ...

Conclusions

- Rivet has come a long way since HZTOOL, but we are not done!
- Standards and reproducibility at the core: drives good physics.
- Lots of new features apart from heavy ions:
  - Multiweights: complex handling of event weight variations for MC systematics.
  - Transfer function based detector simulation.
- Easy for students and outreach (Google summer of code).
- Increased need for support, more hands welcome.
- Room for new directions and ideas.

Thank you for your attention! Let's get hacking!