RIVET analysis for heavy-ion physics

JETSCAPE Workshop
Aug 05, 2022

Antonio Silva
antonio.silva@cern.ch
University of Tennessee - Knoxville

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Outline

- **Rivet and analysis preservation**
  - **What is Rivet?**
  - **Analysis preservation**

- **Recent developments in Rivet for heavy ions**
  - **What Rivet can already do**

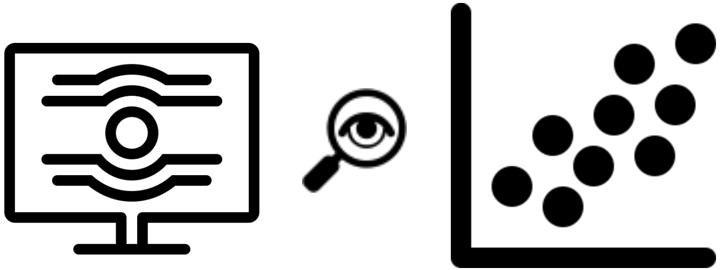
- **A brief demonstration**

Rivet and analysis preservation

What's Rivet?

Robust Independent Validation of Experiment and Theory (Rivet)

Analysis Code Repository

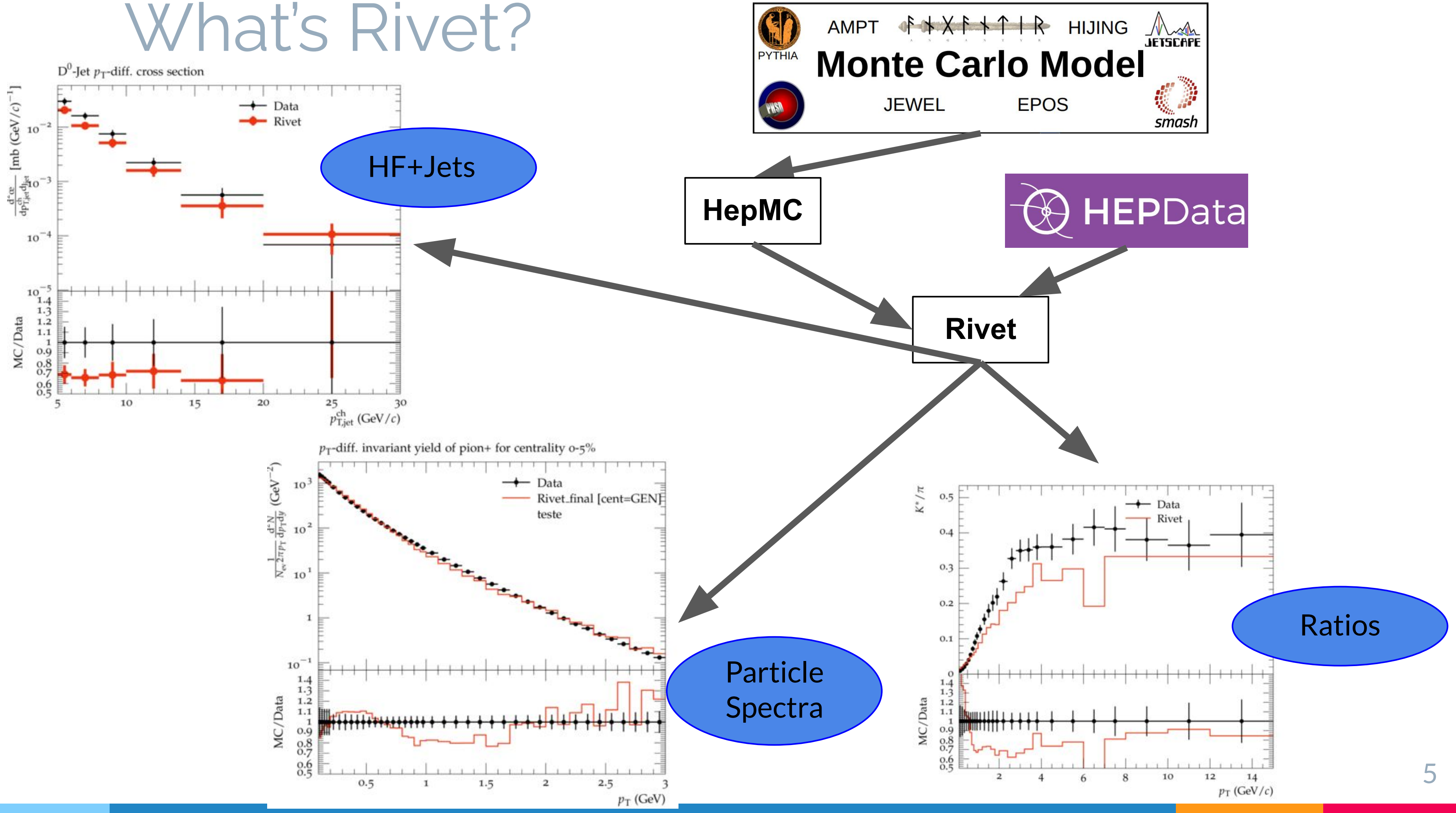


Comparison between theory and data

Search for data
EXPERIMENT_YEAR_I<InspireNumber>

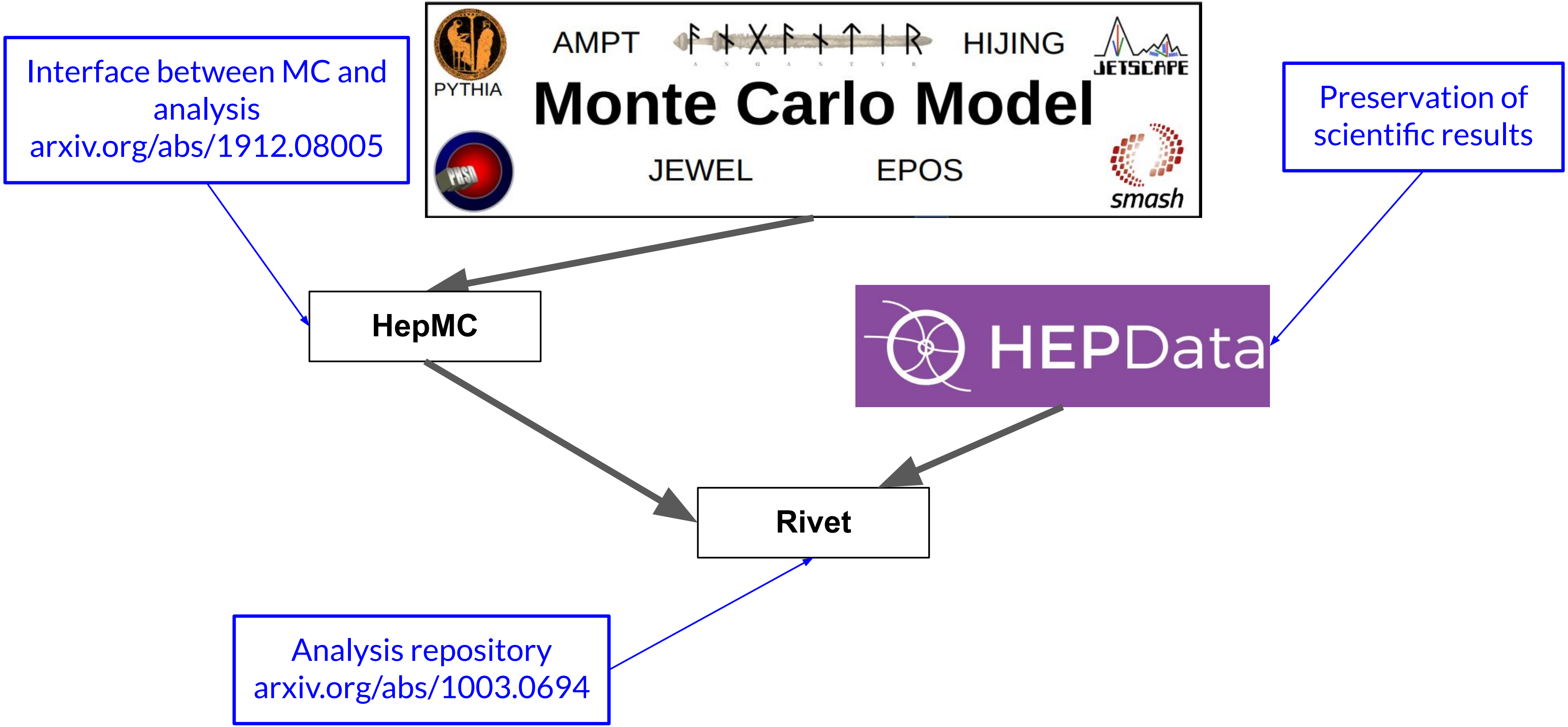
Relatively easy to use

What's Rivet?



AMPT
HIJING
PYTHIA
JETSCAPE
Monte Carlo Model
JEWEL
EPOS
smash
HepMC
HEPData
Rivet
HF+Jets
Particle Spectra
Ratios
D^0-Jet p_T-diff. cross section
Data
Rivet
MC/Data
$p_{T,jet}^{ch}$ (GeV/c)
p_T-diff. invariant yield of pion+ for centrality 0-5%
Rivet_final [cent=GEN]
teste
p_T (GeV)
K^*/π
p_T (GeV/c)

What's Rivet?


Interface between MC and analysis
arxiv.org/abs/1912.08005
AMPT
HIJING
JETSCAPE
PYTHIA
Monte Carlo Model
JEWEL
EPOS
smash
Preservation of scientific results
HepMC
HEPData
Rivet
Analysis repository
arxiv.org/abs/1003.0694

Analysis preservation

How Rivet can contribute to analysis preservation

- The details related to the methods an analysis uses are not always well described in the article
  - Even internal analysis notes could be incomplete
  - Recover the tiny details of an analysis after many years can be very time consuming!

- For people outside experiments, it is not clear how observables and estimators are defined
  - Ex. multiplicity, centrality, primary particles, etc

- Convenient for theoreticians interested in testing models
  - Knowledge of the large number of experimental methods is not required

Recent developments in Rivet for heavy ions

What Rivet can already do

- ALICE primary particles definition (from https://cds.cern.ch/record/2270008/files/cds.pdf)

> A primary particle is a particle with a mean proper lifetime τ larger than $1\,\mathrm{cm}/c$, which is either a) produced directly in the interaction, or b) from decays of particles with τ smaller than $1\,\mathrm{cm}/c$, restricted to decay chains leading to the interaction.

  - The definition of primary particles is experiment-dependent

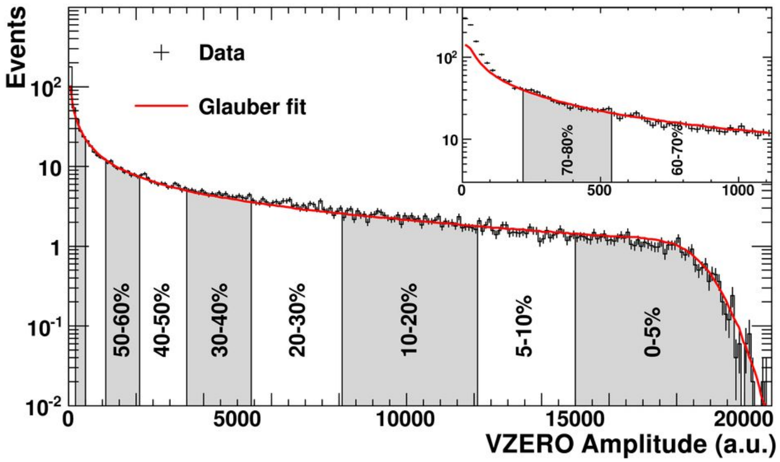
- Currently, some of the ALICE estimators (forward pseudorapidity) for multiplicity/centrality are already available
  - pp: charged-particle multiplicity in the acceptance of the V0
  - p-Pb: charged-particle multiplicity in the acceptance of the V0A
  - Pb-Pb: charged-particle multiplicity in the acceptance of the V0

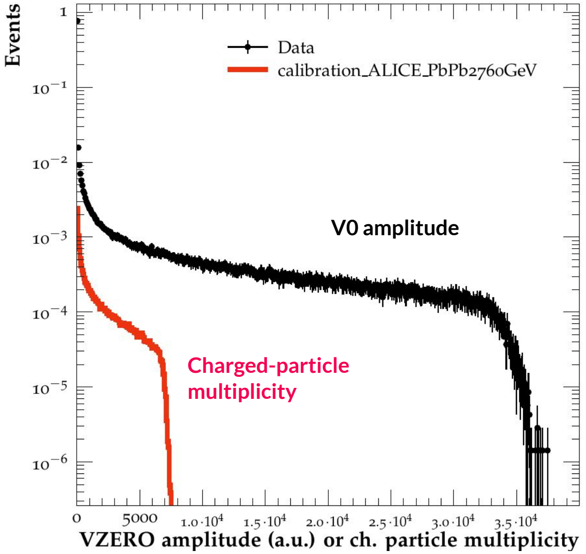
What Rivet can already do

Centrality determination in Rivet

- A calibration file has to be produced before running the analysis
  - Each event generator needs a different calibration
  - A dedicated plugin is used to create the calibration files

- The calibration creates a probability density of number of charged particles per event in the acceptance of the V0 detector
  - $2.8 < \eta_{V0A} < 5.1$ and $-3.7 < \eta_{V0C} < -1.7$

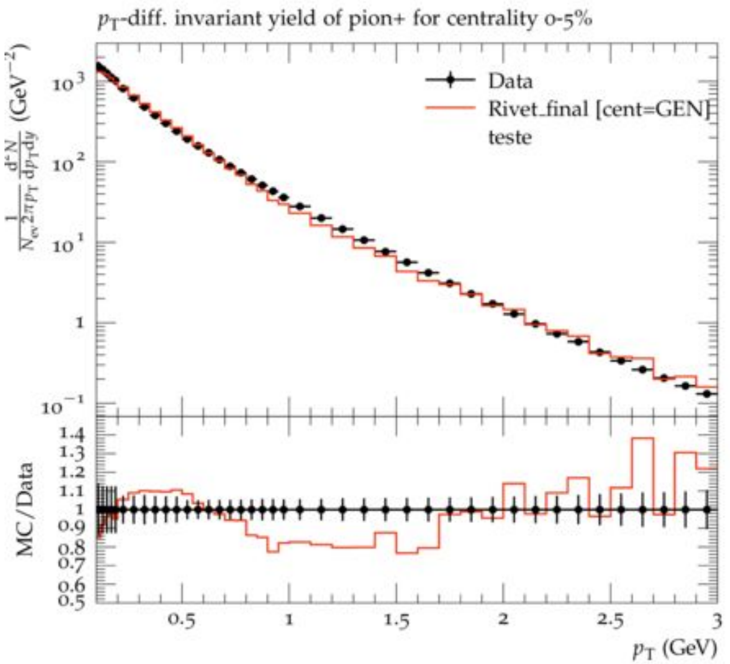




What Rivet can already do

Centrality determination in Rivet

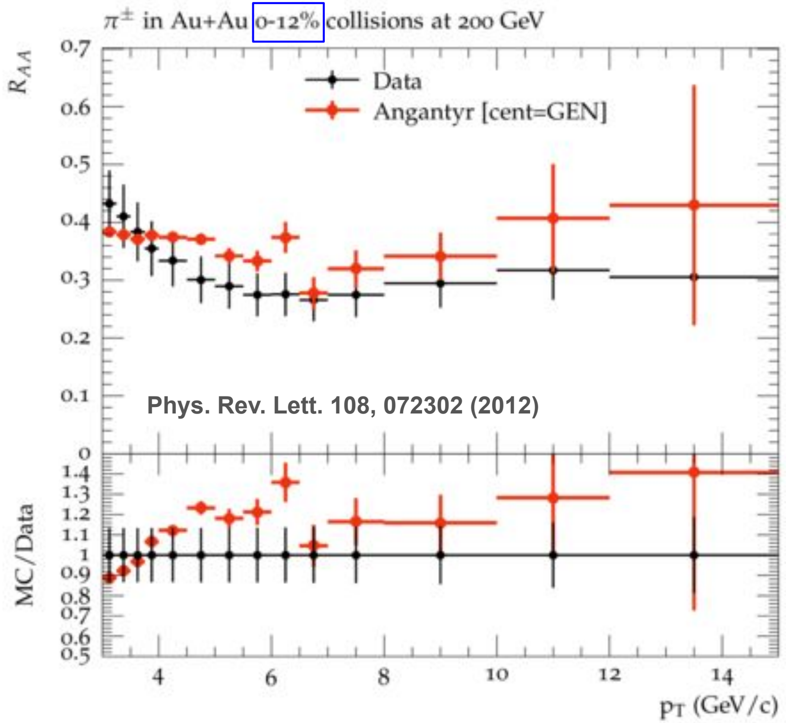
- The calibration file is given to Rivet as a pre-load
- During the analysis run, the centrality is calculated in each event

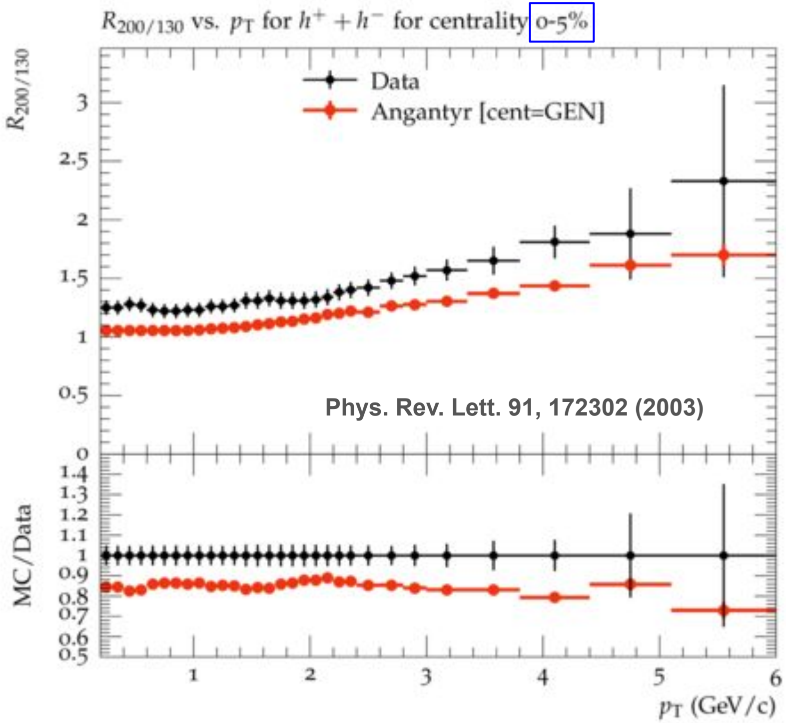


- Centrality is calculated in a way analogous to what is done in the experiment
- Simple implementation
- Previous knowledge of experimental methods is not necessary
- Not a black box! Code is open and methods can be understood

Centrality for STAR and PHENIX

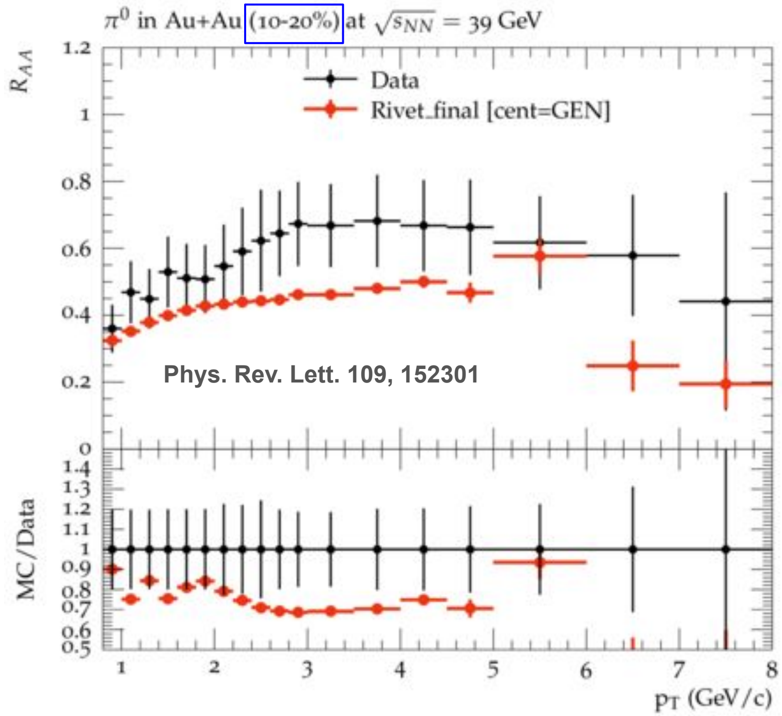
- **The centrality determination in Rivet is based on the same methods used by the respective experiments**
- **Critical feature for the implementation of heavy-ion analyses in Rivet**

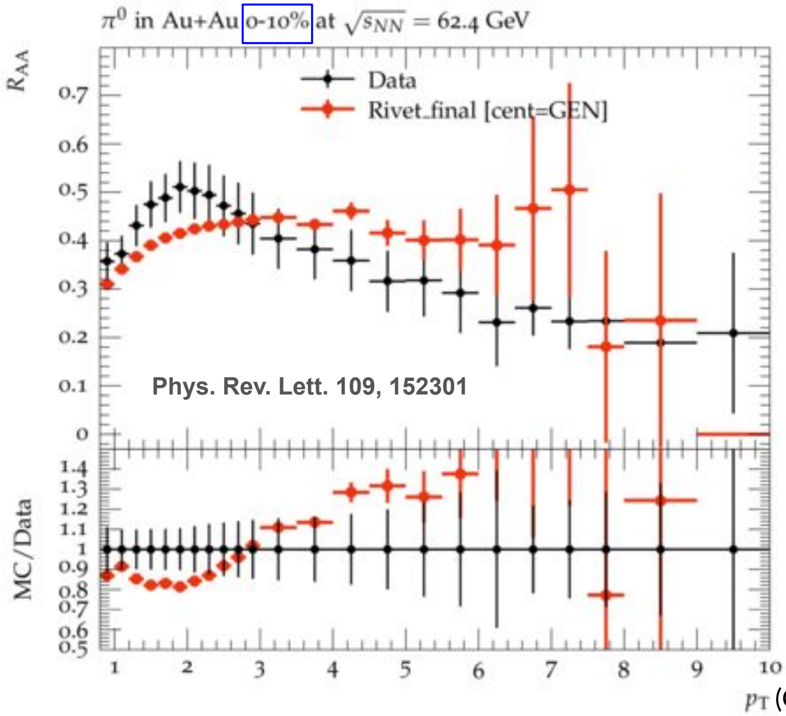




Centrality for STAR and PHENIX

- The centrality determination in Rivet is based on the same methods used by the respective experiments
- Critical feature for the implementation of heavy-ion analyses in Rivet





A brief demonstration

Creating an analysis

- **A initial template analysis can be created with**

```
rivet-mkanalysis <EXPERIMENT>_<YEAR>_I<INSPIRE_NUMBER>
```

- **In this demonstration we are going to use**

```
rivet-mkanalysis ALICE_2021_I1797443
```

- https://arxiv.org/abs/2005.11120
- https://www.hepdata.net/record/ins1797443

- **Rivet will created a template analysis and download the data from HepData**

- **We then delete everything inside analyze() and finalize(). In init() we keep only**

```
const FinalState fs(Cuts::abseta < 4.9);
book(_h["AAAA"], 1, 1, 1);
```

Creating an analysis

```
/// @brief Add a short analysis description here
class ALICE_2021_I1797443 : public Analysis {
public:

  /// Constructor
  RIVET_DEFAULT_ANALYSIS_CTOR(ALICE_2021_I1797443);
  /// @name Analysis methods
  /// @{

  /// Book histograms and initialise projections before the run
  void init() {
    const FinalState fs(Cuts::abseta < 4.9);
    book(_h["AAAA"], 1, 1, 1);
  }

  /// Perform the per-event analysis
  void analyze(const Event& event) {

  }

  /// Normalise histograms etc., after the run
  void finalize() {

  }

  /// @}

  /// @name Histograms
  /// @{
  map<string, Histo1DPtr> _h;
  map<string, Profile1DPtr> _p;
  map<string, CounterPtr> _c;
  /// @}
};
```

- **The class ALICE_2021_I1797443 inside ALICE_2021_I1797443.cc will look like this**

Creating an analysis

Inside init():

- **Add the particle selection from the paper**
  - **We also add a selection to have only pions**
- **declare it and associate a string to the object**
- **Change string of the histogram**
- **Book a counter for the number of events**

```
/// Book histograms and initialise projections before the run
void init() {
  const FinalState fs(Cuts::absrap < 0.5 && Cuts::abscharge > 0 && Cuts::abspid == 211);
  declare(fs, "fs");
  book(_h["ChPionPt"], 1, 1, 1);
  book(_c["sow"], "sow");

}
```

Creating an analysis

Inside analyze():

- **Add FinalState using the string that is associated to the FinalState projection**
- **Add entry to counter (no argument)**
- **Loop over particles (pions)**
- **Fill histogram with particle p_T in GeV/c**

```
/// Perform the per-event analysis
void analyze(const Event& event) {

  const FinalState fs = applyProjection<FinalState>(event, "fs");
  const Particles particles = fs.particles();

  _c["sow"]->fill();

  for(auto p : particles)
  {
    _h["ChPionPt"]->fill(p.pT()/GeV);
  }
}
```

Creating an analysis

Inside finalize():

- **Scale the histogram by the number of events**

```
/// Normalise histograms etc., after the run
void finalize() {

  _h["ChPionPt"]->scaleW(1./_c["sow"]->sumW());

}
```

Rivet analysis

- **Compile your code with**

```
rivet-build RivetALICE_2021_I1797443.so ALICE_2021_I1797443.cc
```

- **Run Rivet**

```
Rivet --pwd -a ALICE_2021_I1797443 -o Rivet.yoda HepMC_File.hepmc
```

- **Make plots with**

```
rivet-mkhtml --pwd Rivet.yoda
```

Centrality

```
/// Book histograms and initialise projections before the run
void init() {

  declareCentrality(ALICE::V0MMultiplicity(), "ALICE_2015_PBPBCentrality", "V0M","V0M");


}
```

init()
- Centrality declaration
- Depends on experiment

```
/// Perform the per-event analysis
void analyze(const Event& event) {

  // The centrality projection.
  const CentralityProjection& centProj = apply<CentralityProjection>(event,"V0M");

  // The centrality.
  const double cent = centProj();

}
```

analyze()
- Get the centrality projection
- Get the centrality of the event

Centrality

In heavy-ions, in order to get the centrality of the events, some additional parameters are needed

```
rivet --pwd -p calibration.yoda -a ANALYSIS_NAME:cent=GEN -o Rivet.yoda HepMC_File.hepmc
```

File containing particle multiplicity or impact parameter distributions
→ Calibration file

- Depends on model, beam, energy

Centrality strategy

Summary

- **Rivet** → **Experimental analysis for MC repository**
- **HepData** → **Repository of data**
- **HepMC** → **Interface between MC and analyses**

- Data and analysis preservation
- Easy comparison of data and theory

- **Recipe given by experiments**
  - **Maximum fidelity to methods used in the measurement**

- **Other models will be compared using the same code**

Where to find Rivet: https://gitlab.com/hepcedar/rivet

Need help? rivet-support@cern.ch