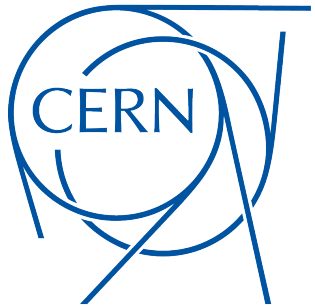
Rivet development for the heavy-ion community

Przemek Karczmarczyk[1,2], Jan Fiete Grosse-Oetringhaus[2], Andreas Morsch[2], Jochen Klein[3]

[1]Warsaw University of Technology, Faculty of Physics
[2]CERN, the European Organization for Nuclear Research
[3]INFN Turin, Italy



17th MCnet Meeting, 10.04.2018

Overview

Introduction

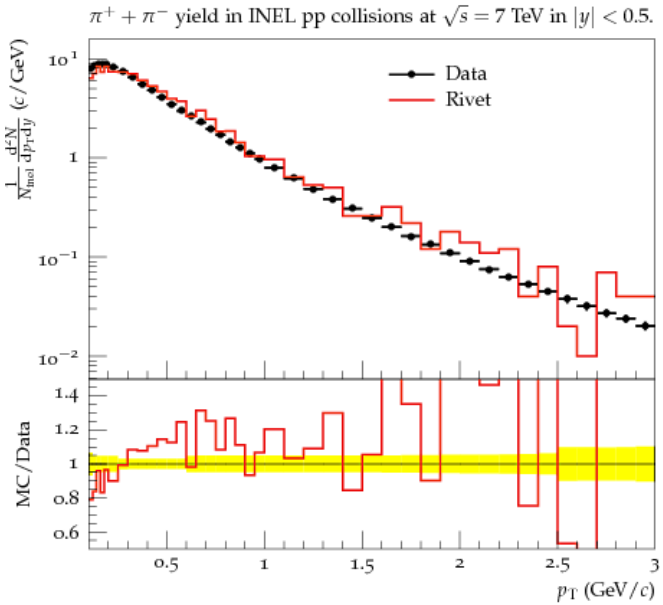
Main idea

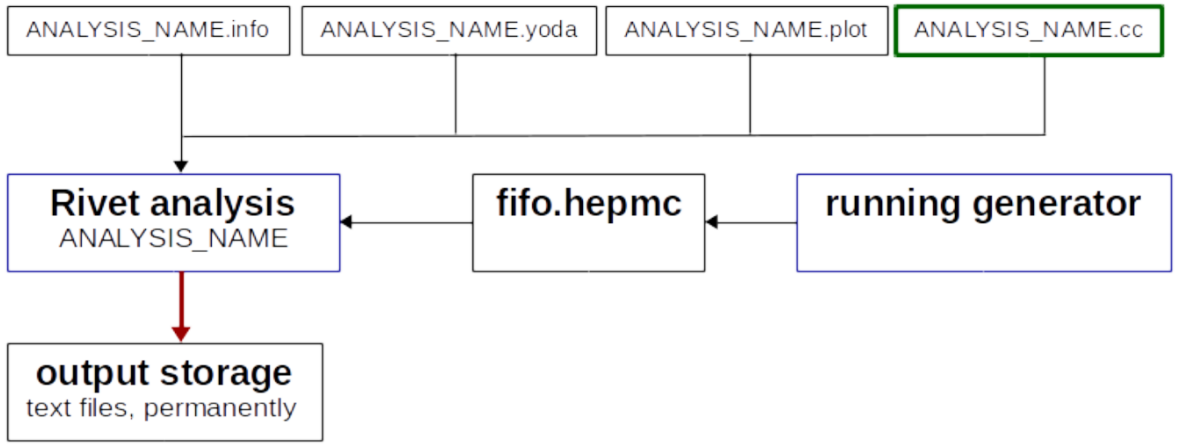
Create an interface between experimental data and theoretical predictions for the heavy-ion community

- What do we have?
  - Analyses (usually with finite acceptance, non-trivial cuts, non-trivial algorithms of extracting physics objects, etc.)
  - Data from different experiments published in papers (values usually published in HepData)
  - Large number of different Monte Carlo (MC) event generators
- What do we want?
  - Test different MC models by comparing their results to the experimentals ones

Rivet

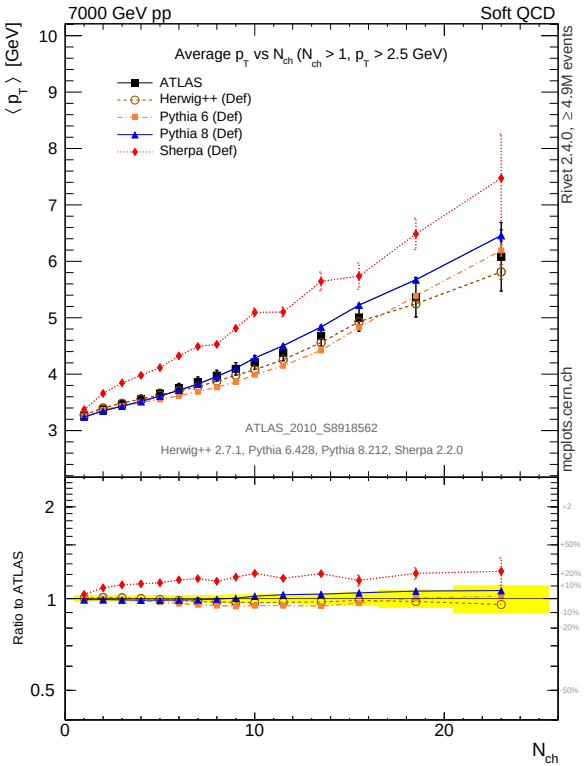
- Rivet: Robust Independent Validation of Experiment and Theory
  - Generator-independent analysis framework
  - System for validation and tuning of event generators
  - Fast and direct comparison between experimental data and generators
- Rivet became the LHC standard for analysis preservation
  - In ATLAS & CMS: standard for analysis teams to write a Rivet routine
  - Interest in other communities, e.g. **heavy-ion community**

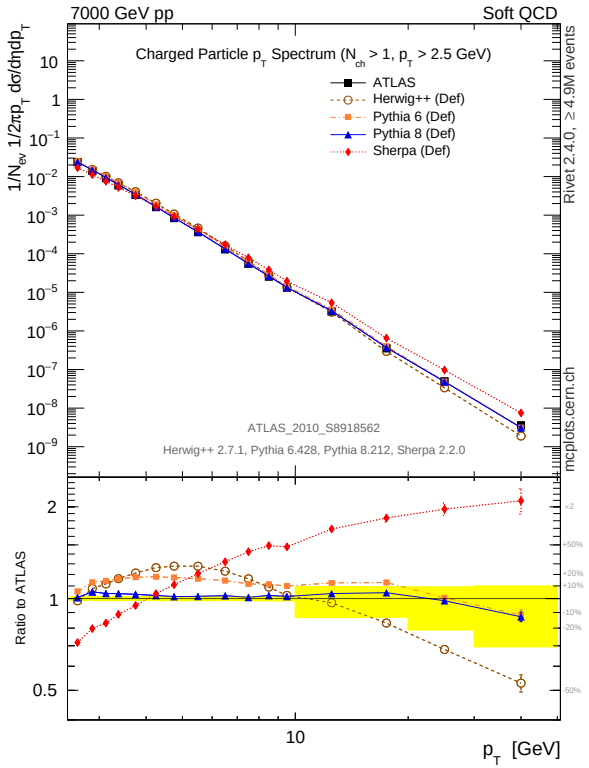




MCPLOTS project

- MCPLOTS: a particle physics resource based on volunteer computing
  - The idea: enable anyone to quickly get an idea of how well a particular model describes various data sets
  - Automatize Rivet creation of comparison plots
  - Provides a simple online repository of plots





Analysis filter:

→ Beam: pp/ppbar ee
→ Analysis:

Z (Drell-Yan)

→ $1/\sigma d\sigma(Z)/d\varphi^*_\eta$
→ $d\sigma(Z)/dpTZ$
→ $1/\sigma d\sigma(Z)/dpTZ$

W

→ Charge asymmetry vs η
→ dσ(jet)/dpT
→ Jet multiplicity

Top (MC only)

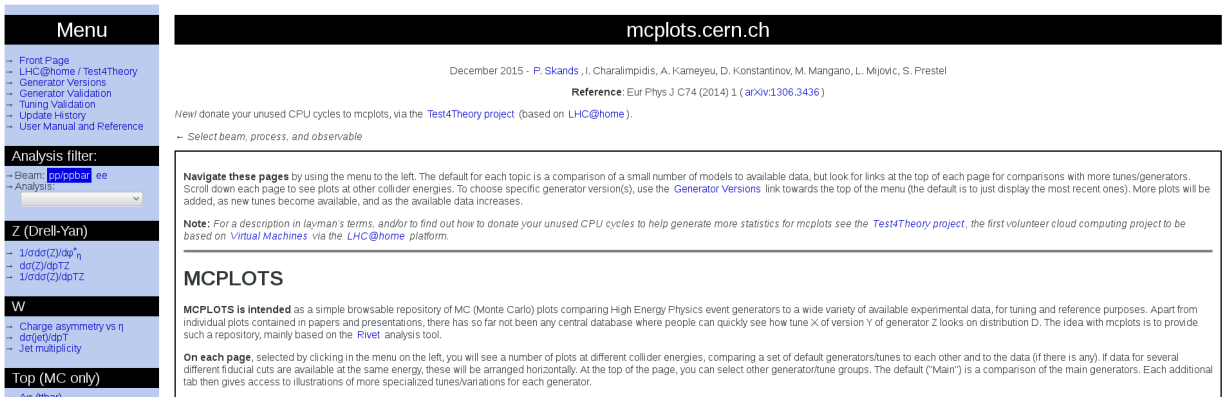
→ Δφ (ttbar)
→ Δy (ttbar)
→ |Δy| (ttbar)
→ M (ttbar)
→ pT (ttbar)
→ Cross sections
→ y (ttbar)
→ Asymmetry
→ Individual tops

Jets

→ Transverse Minor
→ Transverse Thrust
→ Di-jet χ
→ Di-jet Δφ
→ Di-jet mass
→ HT
→ Jet Fragmentation
→ Differential Jet Shape

Rivet in ALICE

- Every PWG (Physics Working Group) in ALICE is working on rivetizing their analyses
  - This will become a common practice
  - No support for heavy-ion analyses $\rightarrow$ the work is limited to pp studies
- What do we want to do?
  - Use existing Rivet and MCPLOTS framework and extend it to match the heavy-ion requirements
  - Create an experiment-theory interface for the ALICE experiment, which can also be used by the whole heavy-ion community



(a) MCPLOTS

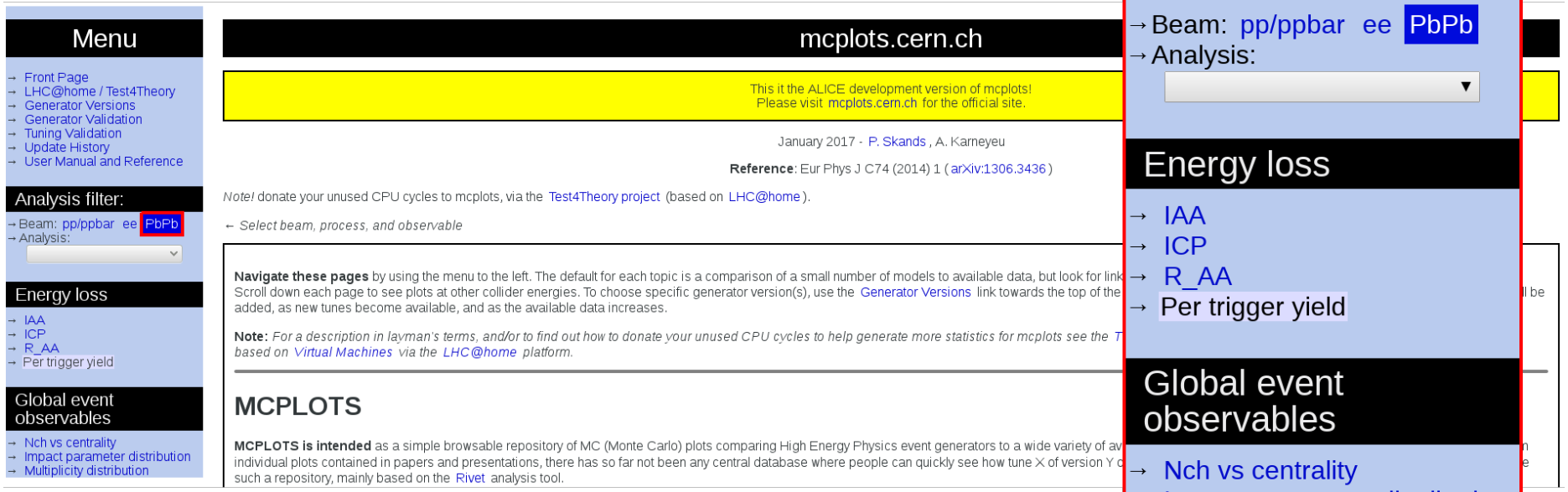
(b) Rivet

Rivet/MCPLOTS limitations

- Typical scenarios:
  - Analysis divided according to **global event variables**, eg. centrality
  - Possibly **2 different beams** needed to perform analysis, eg. pp + AA
  - In some analyses it is required to further **process results** in order to obtain final values, eg. perform fitting
  - Usually require **a lot of computation time** to get reasonable results
  - ... and much more!
- The MCPLOTS and/or Rivet framework does not account for certain needs of heavy-ion analyses:
  - No accounting for global event observables (like centrality, thrust, etc.)
  - No possibility of providing calibration files for the analysis
  - No possibility of running an analysis with provided input files, e.g. create PbPb/pp ratio
  - Inability to split one analysis into several jobs when running on batch system

Tools

- Development machine with:
  - Rivet (2.6.0 version as a baseline)
  - MCPLOTS (https://svnweb.cern.ch/trac/mcplots/browser/trunk/)
  - YODA ('release-1-6' branch from https://yoda.hepforge.org/hg/yoda)
  - HepMC 2.06.09 (with some modifications) + other dependencies
  - Few MC generators (currently using local version of EPOS LHC and JEWEL for testing)
- Webpage: http://mcplots-alice-dev2.cern.ch/

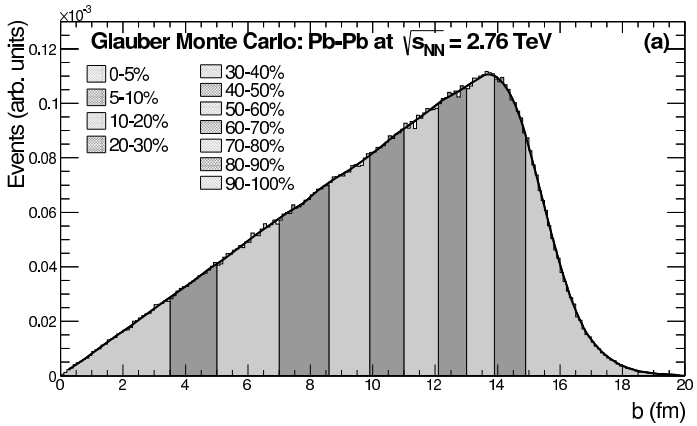


Development

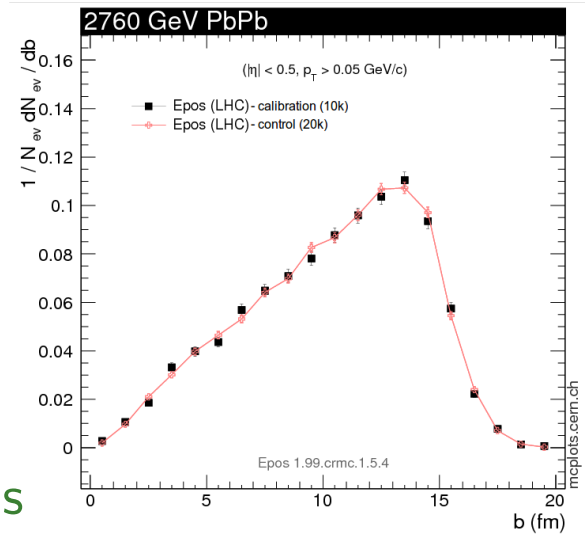
- Three locally developed analyses to cover most use cases:
  - ALICE_2010_I880049: Centrality dependence of the charged-particle multiplicity density at mid-rapidity in Pb–Pb collisions at $\sqrt{s_{\mathrm{NN}}} = 2.76$ TeV
    - Requires: centrality calibration, job splitting
  - ALICE_2012_I1127497: Centrality Dependence of Charged Particle Production at Large Transverse Momentum in Pb–Pb Collisions at $\sqrt{s_{\mathrm{NN}}} = 2.76$ TeV
    - Requires: post-processing, job splitting
  - ALICE_2012_I930312: Particle-yield modification in jet-like azimuthal di-hadron correlations in Pb-Pb collisions at $\sqrt{s_{\mathrm{NN}}} = 2.76$ TeV
    - Requires: centrality calibration, post-processing, job splitting
- Rivet & MCPLOTS developed towards our goals: full support of heavy-ion analyses

Input files

- Calibration step
  - Providing an input YODA file(s) containing a distribution of some parameter (impact parameter, multiplicity, etc.)
  - Using this distribution to calculate global event observable (centrality, thrust, etc.)
  - One file can be re-used in several analyses
- Calibration & post-processing: required method for reading of input YODA files
- Discussed with Rivet main developers and agreed on the solution
- Already implemented, available in the latest Rivet release (2.6.0)



Heavy-ion base class

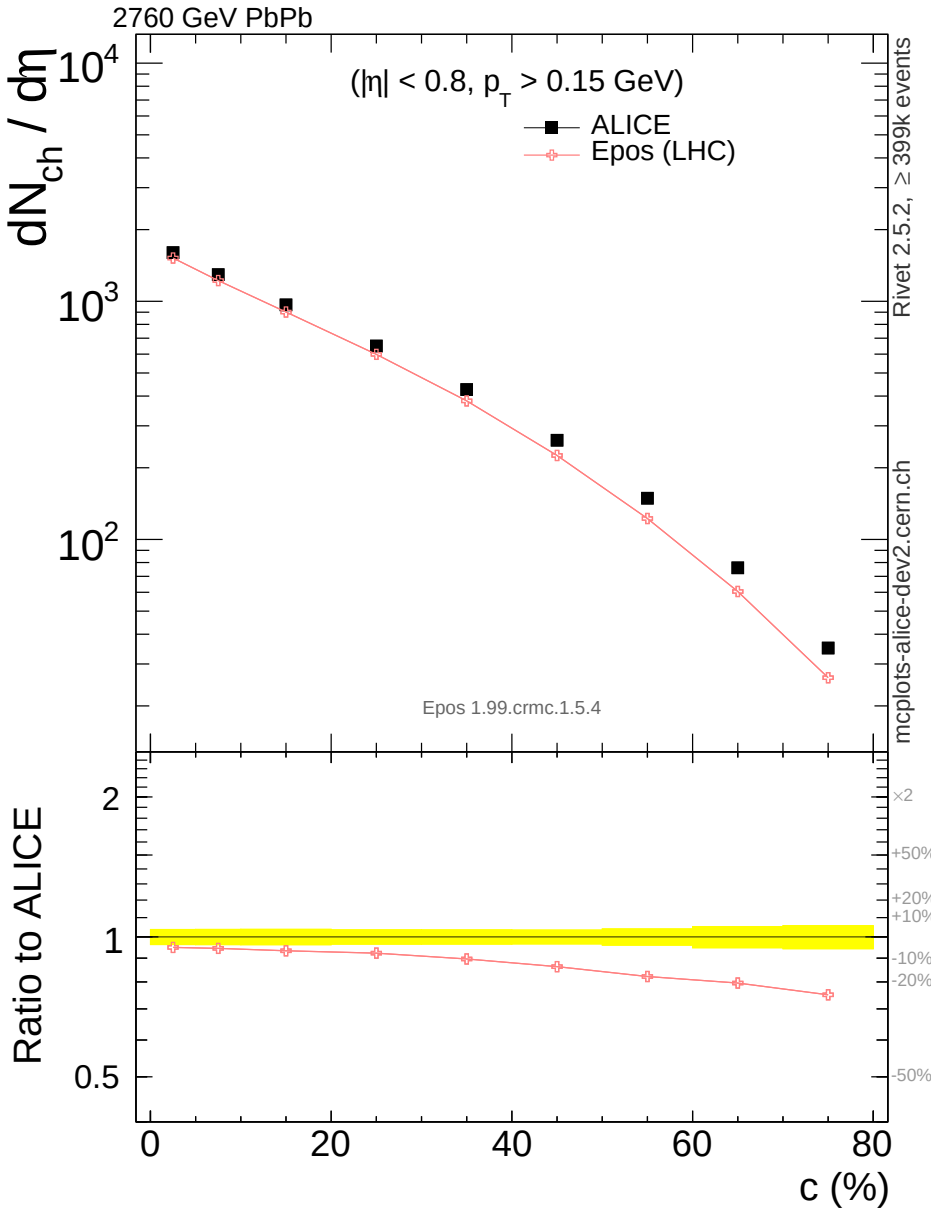


- Heavy-ion analyses have some specific properties not covered by Rivet
- They have a lot in common → heavy-ion base class
- Functionalities of base class:
  - Creating/filling/reading calibration histograms
  - Selecting methods of global event observables calculation
    - e.g. using impact parameter, multiplicity, etc.
    - there may be > 1 method in one analysis
  - Calculating global event observables (centrality, thrust, etc.) using those histograms
  - Creating on-the-fly 'control' histograms for global event observables
    - may be used to increase statistics
    - may be used to control the correctness of calibration histograms

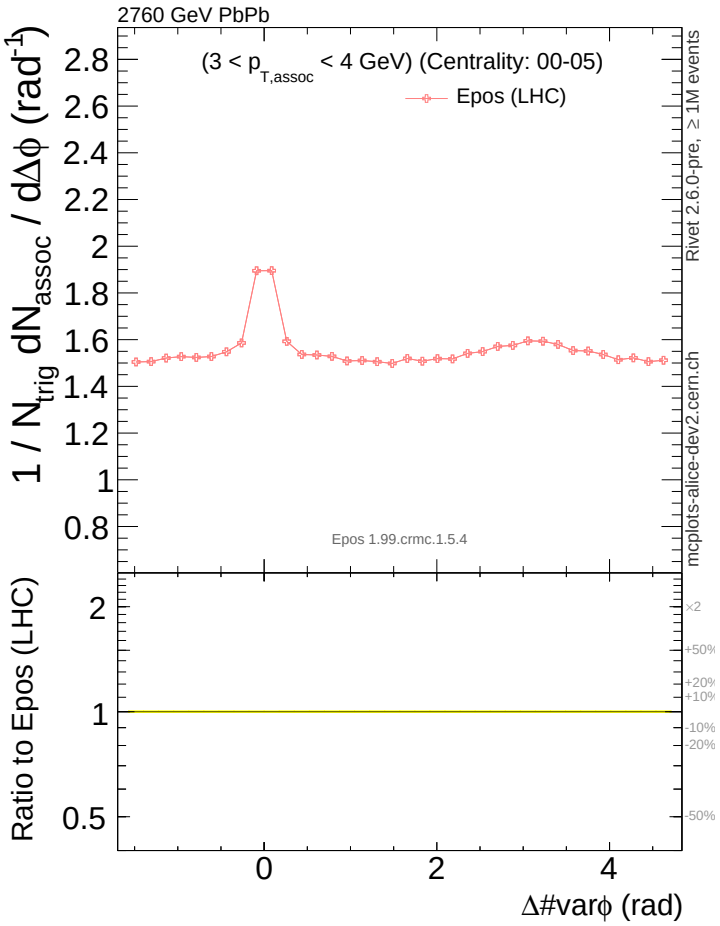
Post-processing

- Analyses which use > 1 beam require 'post-processing' to create ratios, comparisons etc.
- Idea to add Analysis::post() replaced with 'reentrant finalize'
- Reentrant finalize solution
  - Uses analysis' finalize() method several times in a different way
  - Idea: running the same analysis for different beams and preloading results from the previous runs triggers the 'post-processing' step
- Already agreed, but not implemented yet

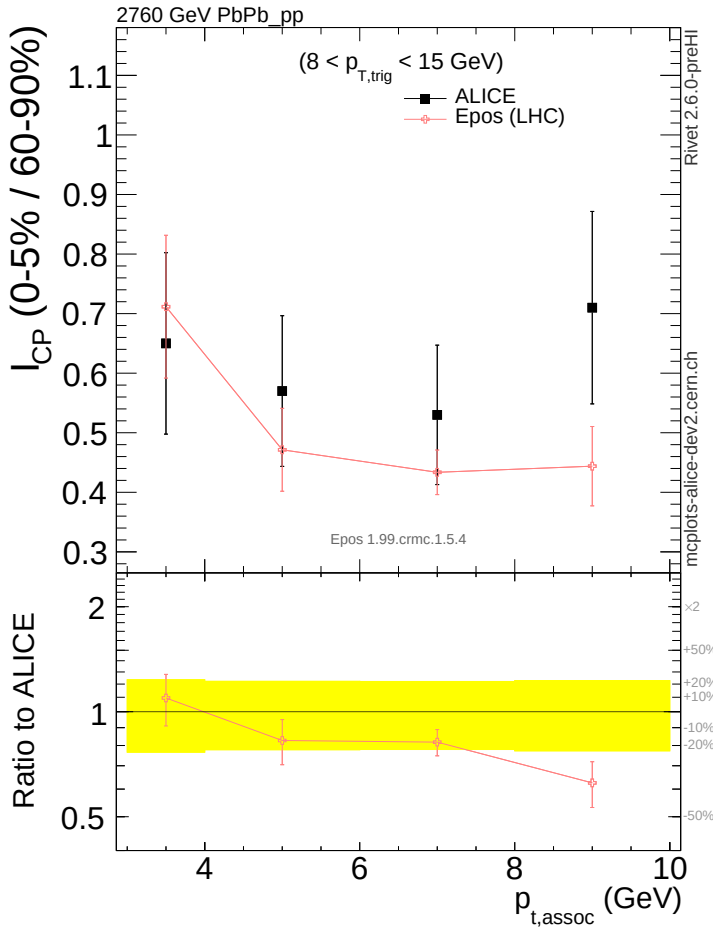
Example: $\mathrm{d}N_{\mathrm{ch}}/\mathrm{d}\eta$ analysis using EPOS



Example: I_{AA}/I_{CP} analysis using EPOS



(a) Pb-Pb, $Y_{0-5\%}$

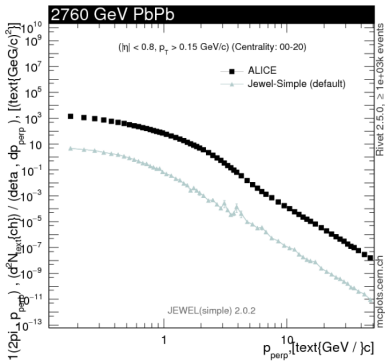


(b) I_{CP}, away side peak

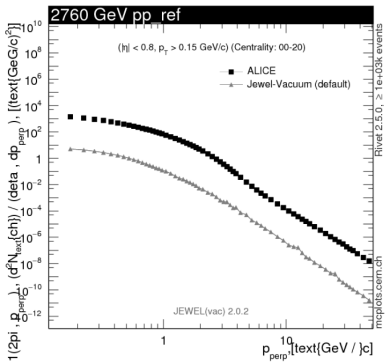
$I_{CP} = Y_{0-5\%}/Y_{60-90\%}$, where $Y_{0-5\%}(Y_{60-90\%})$ is per trigger particle yield in central (peripheral) Pb-Pb collisions

Example: R_{AA} analysis using JEWEL

- $\mathrm{d}N_{AA}/\mathrm{d}p_T$: JEWEL-Simple (0-20%)



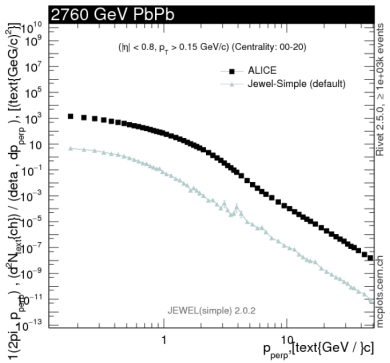
- $\mathrm{d}N_{pp}/\mathrm{d}p_T$: JEWEL-Vac



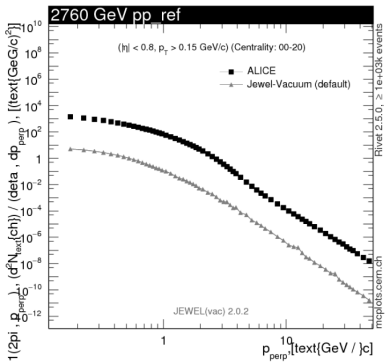
Remark: too much CPU time is needed to generate sufficient events (for JEWEL: ∼20 s per event), rebinning for the MC histograms is needed

Example: R_{AA} analysis using JEWEL

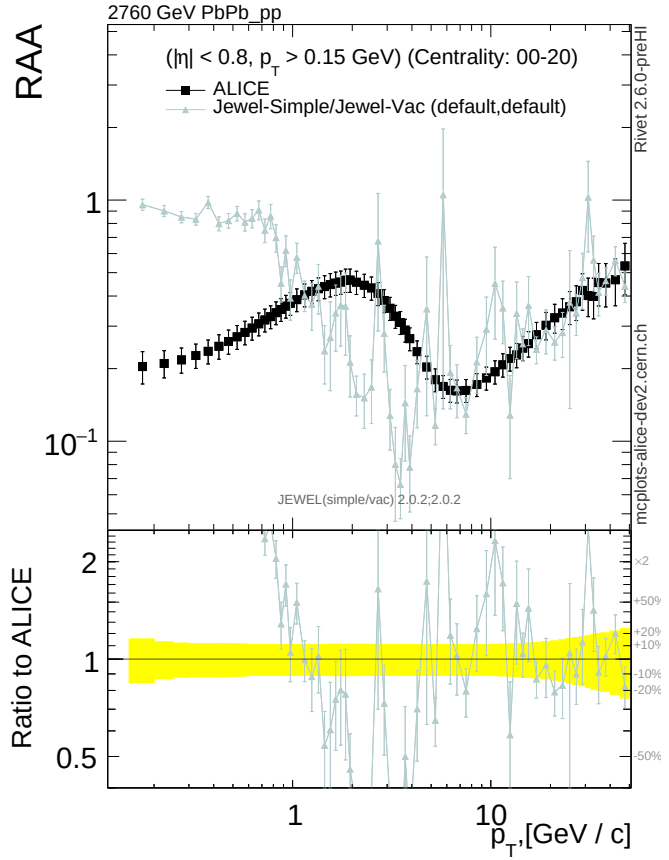
- $\mathrm{d}N_{AA}/\mathrm{d}p_T$: JEWEL-Simple (0-20%)



- $\mathrm{d}N_{pp}/\mathrm{d}p_T$: JEWEL-Vac



- Ratio: R_{AA} (0-20%)



Remark: too much CPU time is needed to generate sufficient events (for JEWEL: ∼20 s per event), rebinning for the MC histograms is needed

Summary

- Rivetization of pp analyses in ALICE - ongoing

Summary

- Rivetization of pp analyses in ALICE - ongoing
- First prototype of Rivet/MCPLOTS for heavy-ion analyses prepared

Summary

- Rivetization of pp analyses in ALICE - ongoing
- First prototype of Rivet/MCPLOTS for heavy-ion analyses prepared
- Already possible to run heavy-ion analyses which require:
  - Calibration histograms as an input to calculate global event variables
  - Two different beam types to calculate final results
  - Presenting plots with different bins in global event observables
  - Splitting jobs into subjobs (because of too large computation time)

Summary

- Rivetization of pp analyses in ALICE - ongoing
- First prototype of Rivet/MCPLOTS for heavy-ion analyses prepared
- Already possible to run heavy-ion analyses which require:
  - Calibration histograms as an input to calculate global event variables
  - Two different beam types to calculate final results
  - Presenting plots with different bins in global event observables
  - Splitting jobs into subjobs (because of too large computation time)
- Much more to do:
  - Include our changes in the official release of Rivet - ongoing, but we need faster integration
  - More analyses to develop to show different heavy-ion aspects
  - Implementing more heavy-ion requirements to Rivet: event mixing, fitting, etc.
  - Implementing more HepMC plugins for heavy-ion generators

References

- Development version of mcplots: http://mcplots-alice-dev2.cern.ch/ (available inside CERN network)
- Source code:
  - Rivet source code: https://rivet.hepforge.org/hg/rivet
  - Rivet source code with heavy-ion extensions: https://github.com/alisw/rivet-hi
  - YODA source code: https://yoda.hepforge.org/hg/yoda, branch: 'release-1-6'

Thank you!

Backup

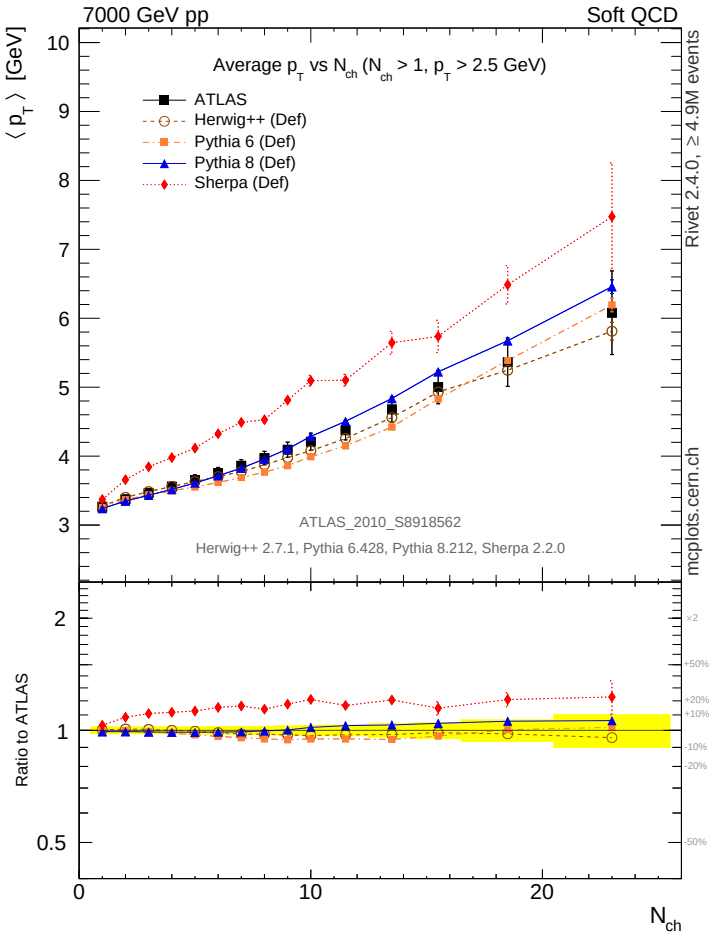
More about Rivet

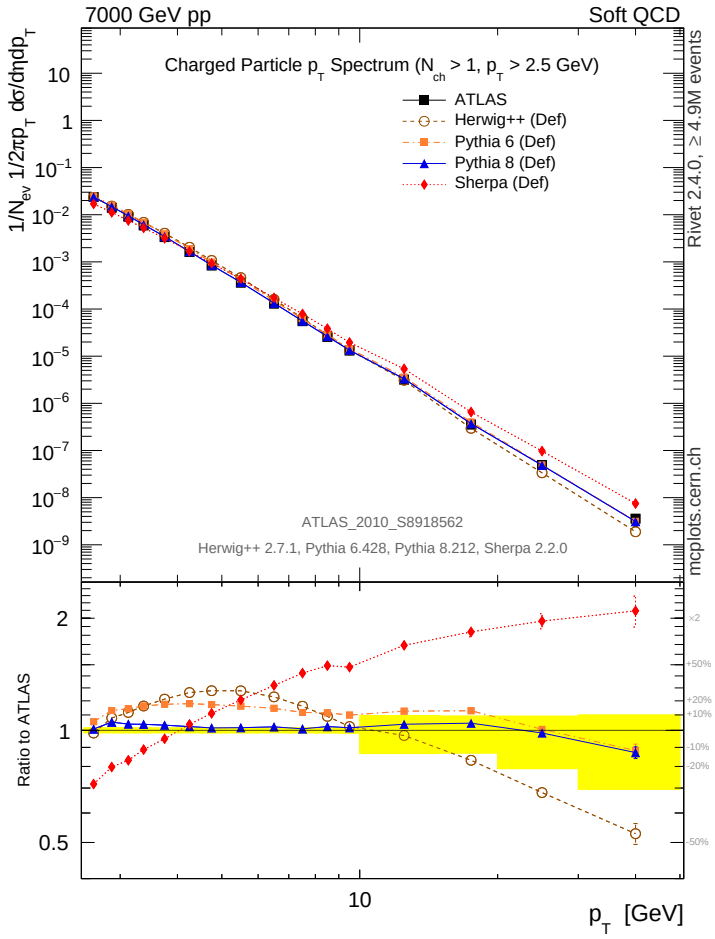
- Rivet can:
  - Read input from Monte Carlo generator (from file or FIFO)
  - Run one or more (validated) analyses on the input data from generator
  - Produce plots corresponding to available measurements
  - Produce comparisons with the corresponding experimental data
- Rivet features
  - Provides a large set of experimental analyses (currently over 350...) useful for MC generator development, validation, and tuning
  - Analyses correspond to the actual paper results
  - Analysis algorithms in object-oriented C++
  - Analysis code separated $\rightarrow$ easy way to add a new one
  - Using the HepMC event format $\rightarrow$ independed of MC generators
  - Plotting based on YODA framework

MCPLOTS project

- MCPLOTS: a particle physics resource based on volunteer computing
  - The idea: enable anyone to quickly get an idea of how well a particular model describes various data sets
  - Automatize Rivet creation of comparison plots
- The mcplots.cern.ch web site
  - Provides a simple online repository of plots made with high-energy-physics generators and comparisons with a wide variety of experimental data
  - Relies on the following basic prerequisites:
    - The HEPDATA database of experimental results
    - The Rivet Monte Carlo analysis tool
    - Monte Carlo event generators (currently implemented: ALPGEN, EPOS, HERWIG++, PHOJET, PYTHIA 6, PYTHIA 8, SHERPA, VINCIA)
    - The LHC@HOME 2.0 framework for volunteer cloud computing

MCPLOTS examples





Analysis filter:

→ Beam: pp/ppbar ee
→ Analysis:

Z (Drell-Yan)

→ $1/\sigma d\sigma(Z)/d\varphi^*_\eta$
→ $d\sigma(Z)/dpTZ$
→ $1/\sigma d\sigma(Z)/dpTZ$

W

→ Charge asymmetry vs η
→ dσ(jet)/dpT
→ Jet multiplicity

Top (MC only)

→ Δφ (ttbar)
→ Δy (ttbar)
→ |Δy| (ttbar)
→ M (ttbar)
→ pT (ttbar)
→ Cross sections
→ y (ttbar)
→ Asymmetry
→ Individual tops

Jets

→ Transverse Minor
→ Transverse Thrust
→ Di-jet χ
→ Di-jet Δφ
→ Di-jet mass
→ HT
→ Jet Fragmentation
→ Differential Jet Shape

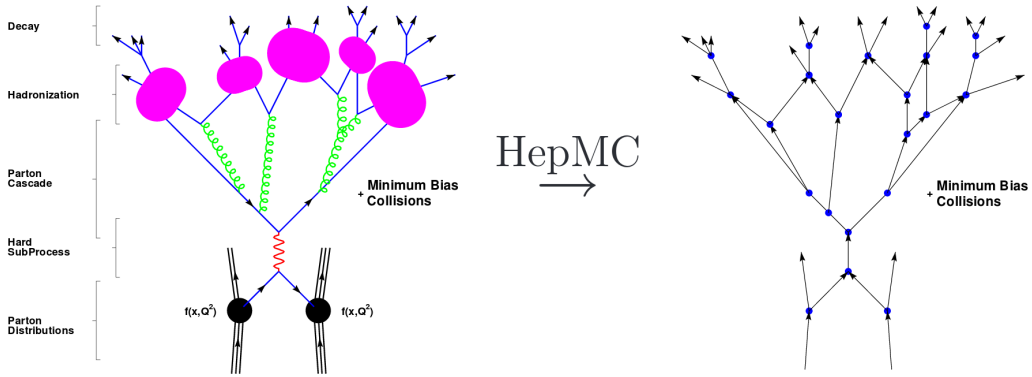
Rivet analysis

```
// -*- C++ -*-
#include "Rivet/Analysis.hh"
#include "Rivet/Projections/ChargedFinalState.hh"

namespace Rivet {

  class ALICE_2010_I123456 : public Analysis {
  public:
    ALICE_2010_I123456() :
      Analysis("ALICE_2010_I123456")
    { }

    void init() {
      _histNevVsImpactPar = bookHisto1D(1, 1, 1);
    }

    void analyze(const Event& event) {
      const float impactPar = event.genEvent()->heavy_ion() ? event
            .genEvent()->heavy_ion()->impact_parameter() : -1;
      _histNevVsImpactPar->fill(impactPar, event.weight());
    }

    void finalize() {
      scale(_histNevVsImpactPar, 1./sumOfWeights());
    }

  private:
    Histo1DPtr _histNevVsImpactPar;
    m_centrRegions;
  };
```

- init(): serves to book histograms, initialize variables etc.
- analyze(...): analysis handler passes events to this function, events are being analysed and histograms are being filled here
- finalize(): histograms are modified (scaled, divided, ...) for the final data comparison

HepMC & YODA

- HepMC:
  - Standarized event record library for High Energy Physics Monte Carlo generators and simulation
  - A direct output of most of modern generators
  - File format contains information about final state particles
  - Why? We need an interface between generator and analysis



- YODA (Yet more Objects for Data Analysis):
  - A histogramming toolkit developed as a lightweight common system for MC event generator validation analyses
  - The core histogramming system in Rivet

YODA logo