EPOS (meets experiments...)

Klaus Werner

Role of theorists

□ **Have a code using HepMC format (OK for EPOS)**

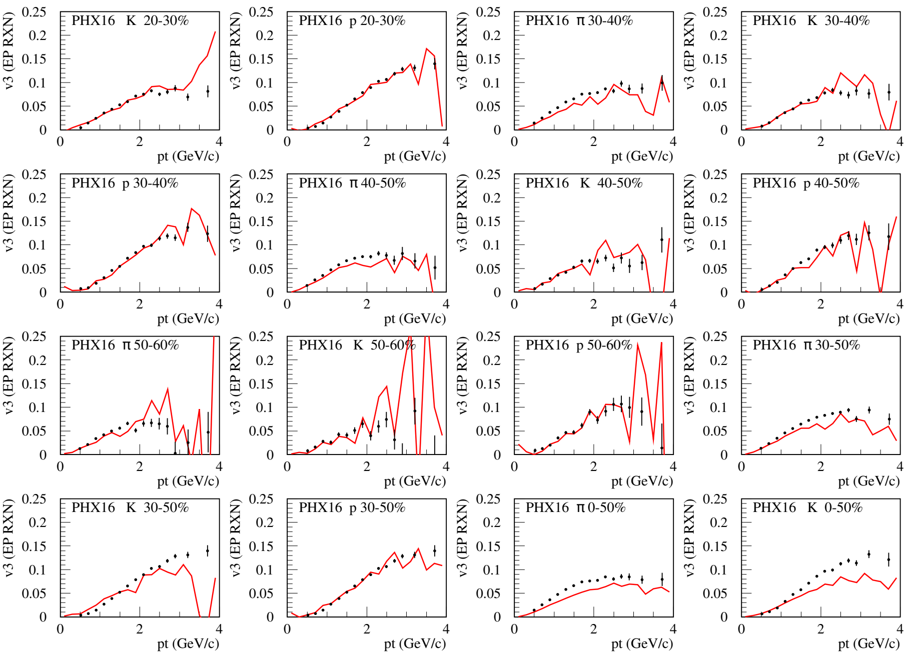
□ **Make the code public (OK for EPOS-LHC,
very soon for EPOS4)**

□ **Wait for Rivet analyses**
(checking HepData, Alice papers since 2018, I found 3)

Our local analysis

So far we are using extensively our “local” analysis tool, based on

- □ **generic analysis code, like**
  - **analyze X as a function of Y**
  - **with conditions Z1, Z2, ...**
- □ **for an ensemble of (100s of) analyses : some script**
  - **which provides the parameters (cuts, particle types etc)**
  - **the instructions for plotting**
  - **one-line instructions for adding data points**
- □ **use macros with place holders for pieces of the script**


PHX16 K 20-30%
PHX16 p 20-30%
PHX16 π 30-40%
PHX16 K 30-40%
PHX16 p 30-40%
PHX16 π 40-50%
PHX16 K 40-50%
PHX16 p 40-50%
PHX16 π 50-60%
PHX16 K 50-60%
PHX16 p 50-60%
PHX16 π 30-50%
PHX16 K 30-50%
PHX16 p 30-50%
PHX16 π 0-50%
PHX16 K 0-50%
v3 (EP RXN)
pt (GeV/c)
0.25
0.2
0.15
0.1
0.05
0
0
2
4

To get these plots: 5-6 line script for 1st plot (EP method to get v2)

- □ **1 line to define event plane**
- □ **1 line to compute v2´= <cos2(phi-phiEP>**
- □ **1 line to compute v2 = v2′ / EP-resolution**
- □ **2-3 lines to define plot (text etc) + data**

Time needed :

- □ **5 min for 1st plot**
- □ **+ one minute the the others, when using macros / place holders**

Philosophy: Separate coding and providing parameters

Rivet project 2019

with Maria Stefaniak, Gabriela Pokropska, Johannes Jahan

Can we use Rivet to replace our local stuff?

Advantages: we get (for free)

- □ **the analysis (made by people who know the experiment)**
- □ **the data**

We concentrate on code development

To start with (and understand it) we developed an analysis: STAR_2017_PRC96_044904 (an exception, not work for theorists)

STAR_2017_PRC96_044904.cc	700 line C++ code
STAR_2017_PRC96_044904.plot	plot instructions
STAR_2017_PRC96_044904.info	some infos
STAR_2017_PRC96_044904.yoda	data
STAR_BES_CEN.cc	centrality definition
AuAu39_10evts.hepmc	MC data from EPOS4

Then one executes (to get the final yoda file)

```
rive          #define environment
rivet-buildplugin RivetSTAR_2017_PRC96_044904.so STAR_2017_PRC96_044904.cc
rivet-buildplugin RivetSTAR_BES_CEN.so STAR_BES_CEN.cc
rivet EPOS_3259_AuAu39.hepmc --pwd -a STAR_BES_CEN \
          -o RivetEPOS_STAR_CEN.yoda
rivet EPOS_3259_AuAu39.hepmc --pwd --ignore-beams \
         -a STAR_2017_PRC96_044904:cent=GEN -p STAR_BES_CEN.yoda -o X.yoda
```

My conclusion (concerning Rivet development for non-experts)

□ **quite heavy for a simple analysis (concerning code writing)**

□ **we encountered quite often technical problems (environment-wise)**

Concerns with respect to Rivet

□ **None ...**
as long as other people provide the analysis code

Concerns with respect to data/model comparisons

- □ **Data should be data (and not partially theory), N_part etc are not measurable**
- □ **Glauber MC should not be mixed with data**
  (Glauber can be used for total cross sections, but it makes absolutely no sense to use it for particle production)
- □ **Provide proper centrality definitions**
  - **Experimentally one uses often some forward detector (VZA for Alice), but I never get a properly normalized distribution when I ask for**
  - **We need dN_ev / dX, whith X being the centrality defining variable**

□ **When comparing data to models, one should identify (and not show together)**

- **publicly available general purpose event generators**
- **and "private" toy models, specialized for few observables**

□ **Experimental collaborations should have a pool of models, and use them, nothing else.**