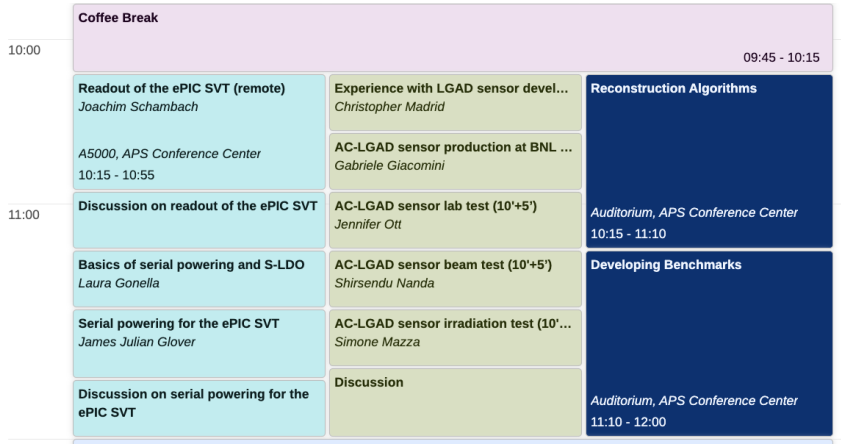
Benchmarking for ePIC physics analysis

A short walk-thru and a few tips to get started

Kong Tu, BNL
(zhoudunming@bnl.gov)
Jan 11, 2024

Benchmarks in ePIC

- "*Developing benchmarks*" was part of the first day software tutorials – the official introduction.
- The simple idea is to *freeze* the analysis so that it can be run over and over against different developments in geometry, reconstruction, software, etc.



Here I'll walk you through an example of physics benchmark, how to run it, how to access the simulation output files, what the macro looks like, etc.

An example – eA diffractive phi.

https://github.com/KongTu/EICreconOutputReader/tree/benchmark-july-2023

One can follow my steps on your laptop now, if one wants.

- download your eic-shell if you don't have one.

- run "**./eic_shell**".

- run "**git clone https://github.com/KongTu/EICreconOutputReader.git --branch benchmark-july-2023**"

- This github repo is to analyze the diffractive coherent phi production in eAu 18x110 Sartre sample.

Accessing files via XROOTD

Anywhere in the eic_shell environment, now one can check the official simulated output samples:

- **xrdfs root://dtn-eic.jlab.org**
- **ls /work/eic2/EPIC/RECO/23.12.0/epic_craterlake/EXCLUSIVE/DIFFRACTIVE_PHI_ABCONV/Sartre/Coherent/**

To access those files, one can use this line to directly open the root file:

- **auto f = TFile::Open("root://dtn-eic.jlab.org//work/eic2/EPIC/RECO/23.12.0/epic_craterlake/EXCLUSIVE/DIFFRACTIVE_PHI_ABCONV/Sartre/Coherent/sartre_bnonsat_Au_phi_ab_eAu_1.0000.eicrecon.tree.edm4eic.root"**

Put them together

Setting up the environment

Fresh start with an eic_shell, if one hasn't done it yet:

```
wget --output-document install.sh http://get.epic-eic.org --no-check-certificate
bash install.sh
```

Run it:

`./eic-shell`

EICrecon reader

Install EICreconOutputReader

`git clone https://github.com/KongTu/EICreconOutputReader.git --branch benchmark-july-2023`

Running diffractive VM (e.g., phi) samples:

Previously, we access the files via S3. Look into getInputFromS3-exclusive.sh to modify accordingly what to grab from S3:

`./getInputFromS3-exclusive.sh` (this is grabing a few files on s3 for an example)

Note that for official compaign simulation, the software simulation or validation team knows better where the generators are. Simply replace `input/rec-batch_5_official_\*.eicrecon.tree.edm4eic.root` to whatever the directory+name will be.

Now, it is much easier to use xrootd. So replace the 1st argument with `root://dtn-eic.jlab.org//work/eic2/EPIC/RECO/23.12.0/epic_craterlake/EXCLUSIVE/DIFFRACTIVE_PHI_ABCONV/Sartre/Coherent/sartre_bnonsat_Au_phi_ab_eAu_1.0000.eicrecon.tree.edm4eic.root`, where one can change the file name or directory. Wildcard * is supported.

Run with TTreeReader:

```
./runDiffractiveVMReader.sh root://dtn-eic.jlab.org//work/eic2/EPIC/RECO/23.12.0/epic_craterlake/EXCLUSIVE/DIFFRACTIVE_PHI_ABCONV/Sartre/Coherent/sartre_bnonsat_Au_phi_ab_eAu_1.0000.eicrecon.tree.edm4eic.root output/eicrecon-sartre_coherent_phi
```

with input files in `input`, output results in `output`, and benchmark figures in `figures`.

Step 1:
Getting the eic_shell

Step 2:
Download the benchmark script (one that you would like to develop for your own analysis)

Step 3:
Run it and check the output.

Let’s test them

See my terminal screen.

Results with 2000 files

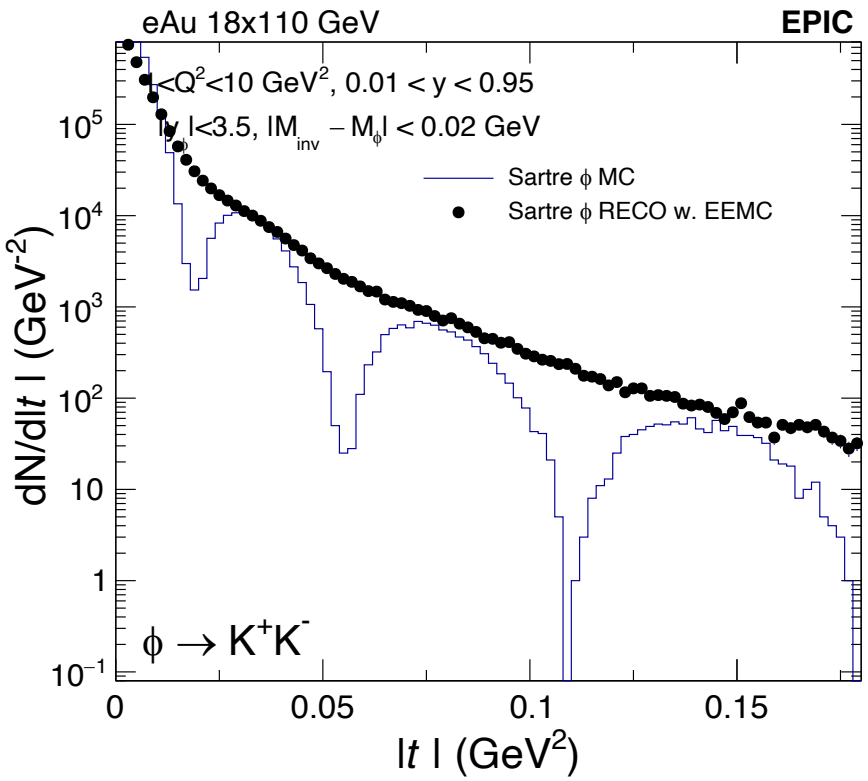


Figure 1

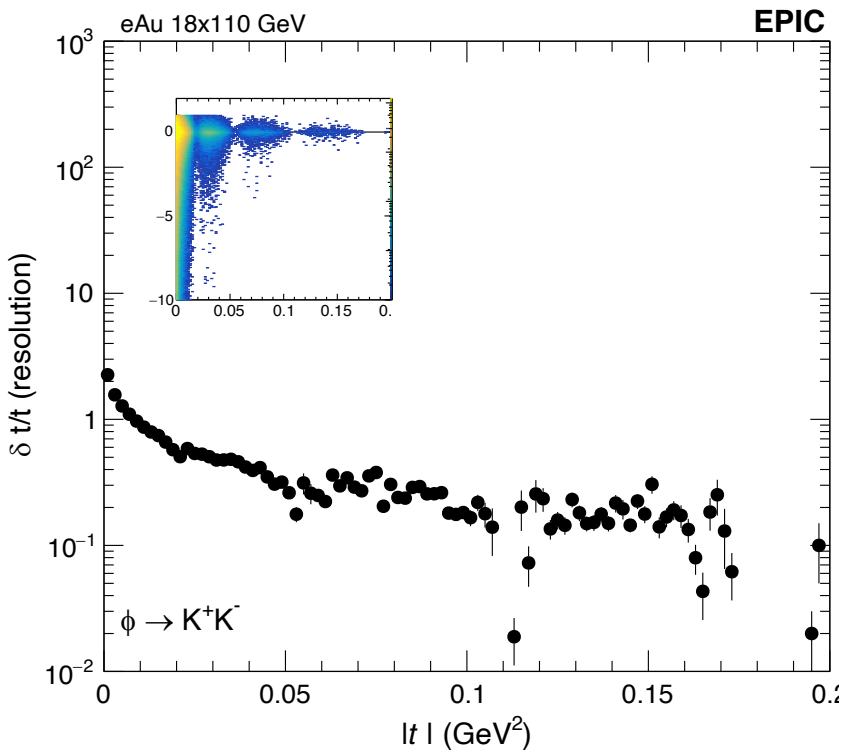


Figure 2

Results with 2000 files

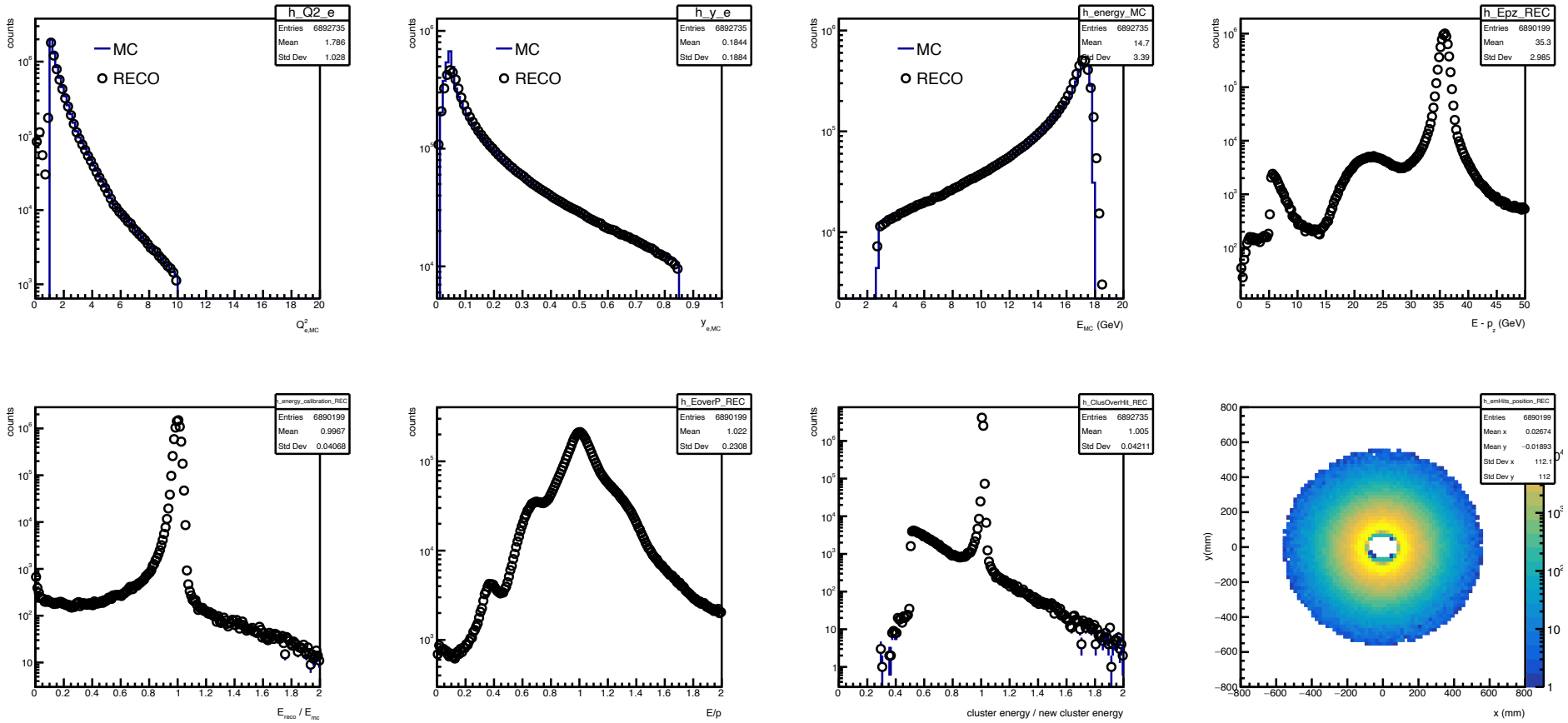


Figure 3

Tips to develop a benchmark analysis

- Start simple, e.g., x and Q^2 analysis in DIS, the output is just two histograms. Limit the complexity of the macros.
- Start with samples that are already produced (if your physics samples are not available yet)
- Build a `workflow` around it, with easy-to-use script to run, access files, macro to make plots, etc.
- Use existing benchmark as an example.
- Refine the benchmark scripts

Advanced level

- https://eicweb.phy.anl.gov/EIC/benchmarks/physics_benchmarks/-/tree/master?ref_type=heads

- After the analysis has been written and tested, one can try to build a version that is similar to this repo.
- CI and visualizations are available for the benchmarks (test version was shown in Dec 2023 in the software meeting, https://indico.bnl.gov/event/21488/contributions/84326/attachments/51461/88015/epic_validation_1262023.pdf)

  https://eic.jlab.org/epic/epic_validation.html

- This part is to work closely with the software team.

Summary

- Benchmark is easy to make - first version shouldn’t take more than one day (~ a couple of hours)
- In the Exclusive/Diffractive/Tagging + FF + FB working groups, we need at least on the order of ~ 10 benchmarks, if not more, to target different aspect of the detector performance.
- We can use the workfest hours to work on ”getting started” together.