

# Professor MC tuning tutorial

Andy Buckley  
University of Glasgow

HSF MC Tuning Workshop  
28 June 2023

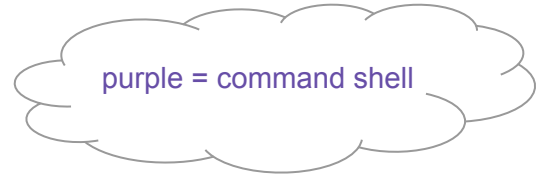


University  
of Glasgow

# Sampling and generating

## ❖ Working environment: Docker

- `$ docker pull hepstore/prof2-tutorial`
- Run the container interactively, with cleanup and a mapped-in dir:  
`$ docker run -it --rm -v $PWD:/host hepstore/prof2-tutorial`
- Optional! ⇒ `# apt-get update && apt-get install vim` (or `emacs-nox`)
- Note: for now these Dockers are amd64 architecture: slow on M1 Macs, will be improved asap



## ❖ Producing the inputs factorises from the tuning

- Image built on Rivet+Pythia 8.3. Docs: <https://pythia.org/latest-manual/Welcome.html>
- Can't assume a particular generator, batch-farm interface, etc. ⇒ sample with script+cfg templating
- Look in the `tmpl/` directory: template files for MB Py8+Rivet jobs
- `# nano tmpl/mbrun.sh` ⇒ reduce number of events if you want a quick local run
- `# prof2-sample -t tmpl/mbrun.sh -t tmpl/py8mb.cmd -n 20 tmpl/paramranges.dat`
- `# ls scan/*`
- Check the contents, values, etc. — is it clear what's going on?
- And run, e.g.: `# for i in 000*/mbrun.sh; do nice -5 bash $i & done`
- It's quite plausible to generate small samples like this on a laptop! But multiple processes, multiple energies, different cuts, and in particular *far* more expensive matrix elements -> cluster/Grid/HPC

# Inspecting, interpolating, tuning

## ❖ Once the run is finished, you can inspect the outputs written into each run dir

- A good idea to clean out unnecessary data:  
# for i in 001\*; do (cd \$i; yoda2yoda -M "/RAW.\*" mb.yoda tmp.yoda; mv tmp.yoda mb.yoda); done
- # cd /work && prof2-envelopes -d /usr/local/share/Rivet/ scan/ or use the pre-prepared mc/ dir now
- Copy output back to the host to view: # cp -r envelopes /host/
- Similar with Rivet plotting: # rivet-mkhtml-mpl scan/001\*/mb.yoda -o /host/rivet-plots

## ❖ Might as well immediately build a surrogate interpolation (“ipol”)

- # prof2-ipol -h
- # prof2-ipol mc/ Unfortunately, prof2-residuals is currently broken...
- Use the ipol-listing tool to generate a starter weights file:
- # prof2-ls -w ipol.dat > weights0.dat
- # cp weights{0,1}.dat && nano weights1.dat
- Edit to cut out bad bins, tweak the fit toward things you care about... this is the creative bit!

## ❖ And... tune!

- # prof2-tune -d \$(rivet-config --datadir) -w weights1.dat
- Plot the output: # rivet-mkhtml-mpl tunes/ipolhistos.yoda -o /host/rivet-plots-tune1
- And iterate! Unfortunately the eigentunes script needs a fix, so no demo: dev help is welcome!!

Thanks for coming!

