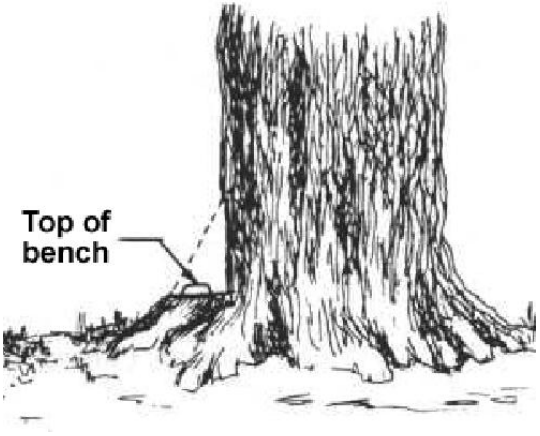
Introduction to DataPipe benchmarking

Tomas Bylund

DataPipe F2F Meeting and Hackathon

October 25, 2023

What is benchmarking?



- Originally a surveying term: a bench mark locates the reference point with a known height to use
- In computing, a benchmark is the result of running a software designed to measure performance (usually of hardware)

The required benchmarking in CTA

B-DPPS-1220 Data Release Consistency Check

The DPPS shall produce an IRF science performance metric report for a given data release that can be compared with previous versions [...]

B-DPPS-3120 Algorithm and Configuration Improvements

The DPPS shall produce, in an automatic and controlled way, a report on the science performance parameters for each new analysis configuration or pipeline-software upgrade [...]

The required benchmarking in CTA

B-DPPS-1220 Data Release Consistency Check

The DPPS shall produce an IRF science performance metric report for a given data release that can be compared with previous versions [...]

B-DPPS-3120 Algorithm and Configuration Improvements

The DPPS shall produce, in an automatic and controlled way, a report on the science performance parameters for each new analysis configuration or pipeline-software upgrade [...]

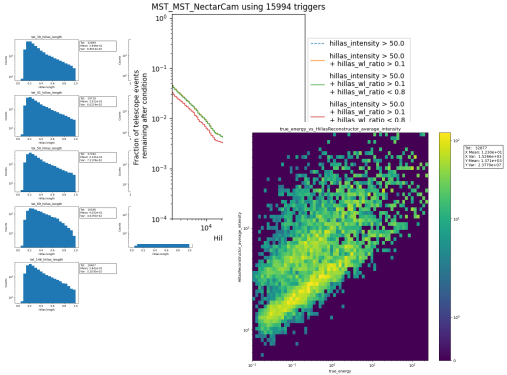
It would seem benchmarking only involves estimating science performance

The (actually) expected benchmarking in CTA

In practice the benchmarking system is expected to produce a large number of intermediate level performance plots, for example

- pixel level: charge resolution
- image level: fraction of surviving triggers
- image level: Hillas parameters
- stereo level: impact parameter
- training level: loss and validation curves

Many individual performance plots expected for at each data level



Practical level requirements for benchmarking

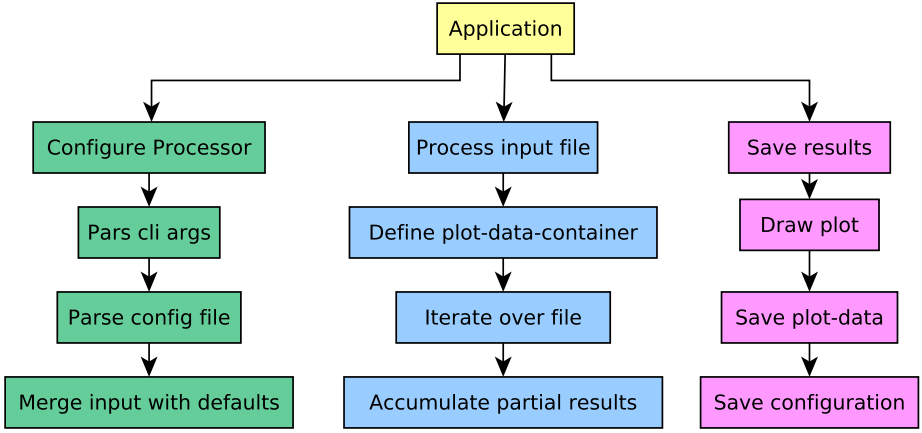
The system needs to easily

- produce a lot of plots
- handle very large files
- produce reproducible results
- store results durably
- compare several benchmark results

Which implies that the system is

- highly automated, once the configuration has been given no further user input is required
- able to traverse files in chunks and accumulate partial results
- able to save data for reproducing plots separately
- capable of individually configure each plot
- able to generate reports collecting partial results

Conceptual design



Current status

Prototype is set of tools currently able to make large number of 1-D histograms for DL1 and DL2 parameters, and collect results into report as well as notebook for comparison. Also a tool that can make 2-D histograms.

- neobench-image-param: DL1 image parameters
- neobench-ml-param: DL2 parameters used by the machine learning
- neobench-hist2d: DL2 parameters (limited reporting)
- neobench-fraction-surviving: fraction of events surviving a given quality cut (no reporting)

Prototype repository found here:
https://gitlab.cta-observatory.org/tomas.bylund/neobench

Demo

Time to show some examples

Open issues to discuss

- Who should the cli target? Currently using new library for the cli partially to make first use more friendly
- Should the tools target developers of new algorithms / code optimisation, or pipeline production managers?
- Is it important that “external” users can easily add new plots?
- Should measurements ever be 1-D, or is it always preferable to have the option of knowing energy dependence

End

Thank you for your attention