End2End cryoEM benchmarks
Case 1: Final Report

Martyn Winn and Tom Burnley, 2nd May 2021

Selected benchmarks

In this study, we considered the following three datasets:

Dataset	EMPIAR ID	Size (GB)	Description
RELION tutorial	10204	8	Small dataset from the Relion tutorial for rapid testing of setup
Rabbit muscle aldolase	10338	1360	Described in https://doi.org/10.1016/j.yjsbx.2020.100020 (re-processed by Martyn)
Coronavirus-HKU1 haemagglutinin esterase	10390	TBC	Tom's example.

The first two datasets were fully processed to give intermediate files, and example benchmark results. The dataset sizes given above include these intermediate files, after gentle cleaning of the project directory by Relion. Further details at https://github.com/stfc-sciml/e2e/blob/master/case1/docs/DATASETS.md We note that copying the large aldolase dataset to different locations was non-trivial, and is best avoided.

For reference, the Relion site has other benchmark results at https://www3.mrc-lmb.cam.ac.uk/relion/index.php/Benchmarks_%26_computer_hardware which use a Plasmodium ribosome dataset available from EMPIAR entry 10028. The benchmarks use just the processed shiny particles, of which there are 105k. Our aldolase benchmark is larger, using 219k particles in the final reconstruction.

Hardware platforms

The main study utilised CPU and GPU nodes of the STFC SCARF cluster, see https://www.scarf.rl.ac.uk/scarf_hardware.html The benchmark jobs were also run on the EPCC nextgenio cluster, containing Cascade Lake nodes. Further details are in https://github.com/stfc-sciml/e2e/blob/master/case1/docs/aldolase_performance_benchmarks.pdf

Software used

We used the Open Source software package Relion version 3.1. Relion provides wrappers to several 3rd party programmes, but the only one we used was Ctffind4. We also used a CCP-EM python package relion-pipeline, which allows scheduling and running job sequences from a Python script. Further details at https://github.com/stfc-sciml/e2e/blob/master/case1/docs/CODE.md

Key results and conclusions

The complete end-to-end workflow for the aldolase dataset is illustrated in https://github.com/stfc-sciml/e2e/blob/master/case1/docs/aldolase_flowchart.pdf with all jobs listed in https://github.com/stfc-sciml/e2e/blob/master/case1/docs/Aldolase%20results.docx

The workflow is complex, as is typical for real datasets. The timings given in the .docx file show that the compute requirements are dominated by Class2D, Class3D and Refine3D jobs, although many additional jobs are required to analyse results and prepare input files.

At a high level, the workflow is trying to do two things: a) select out a consistent set of 2D particles to make the final reconstruction, and b) set some parameters to do with microscope optics and particle motion, which you can only model well when you have a partial solution. At a practical level, the amount of iterative improvement is often limited by compute resource, and improvements in the computational efficiency could allow better workflows and ultimately better scientific results.

With some jobs taking days to run, and with competition for access to shared compute resources, the full aldolase workflow took several weeks to complete. The result was a complete Relion project with all intermediate files. This allowed us to re-run individual steps, and benchmark different hardware, compiler options or runtime options. Example results are given in https://github.com/stfc-sciml/e2e/blob/master/case1/docs/aldolase_performance_benchmarks.pdf

Taking into account wallclock limits and competition for resources, it is virtually impossible to run the major jobs without some form of acceleration. Typically people will use GPUs, and even the relatively old GPU cards on SCARF (K80s) give the best performance. CPU acceleration gives a 4-fold improvement, but still does not compete with GPU platforms.

Note that Relion supports both MPI and multi-threading of each MPI task. Finding the right balance of tasks and threads is largely trial and error. The Relion tutorial, and various forums and blogs, make suggestions, but experience shows that it is hard to generalise to different datasets and hardware platforms. The Class2D, Class3D and Refine3D jobs typically run through 25 iterations, but the time taken is highly non-linear, so trial timings taken for a small number of iterations may not be indicative of the full run. There are also other runtime parameters which may have an effect on the runtime – these are highlighted in the JobFiles used by the benchmarks.

In conclusion, running a real dataset with hundreds of thousands of particles through Relion remains computationally challenging. Acceleration of the code is useful, as would be guidance in the choice of runtime parameters. We found that the scientific results can be affected by the choice of executable or runtime parameters. Though that probably reflects the stochastic nature of the algorithm, it would be worth further investigation. The benchmark files provided can act as a reference for future investigations.