The purpose of these scripts is to process the HELICS Benchmark results and produce different forms of analysis, such as graphs and spreadsheets. These scripts also provide the capability of expanding the analysis of the HELICS Benchmark results.

Below, more descriptions of the scripts are provided.

Data Proccessing:

* benchmark\_post\_processing.py takes the single machine benchmark results, reads each of the separate files, grabs the information, and turns the information into a JSON file. This allows for easy reading and writing of the benchmark results.
* multimachine\_postprocessing.py takes the multi-machine benchmark results, reads each of the separate files, grabs the information, and turns the information into a JSON file. This allows for easy reading and writing of the benchmark results.
* multimachine\_preprocessing.py is used for adding missing metadata to a subset of the multi-machine benchmark results. A word of caution: this should only be used if the user deems it necessary to add the metadata to the files that do not already contain that information. The make\_dataframe.py fills in the missing data after creating a pandas DataFrame object. Thus, when using the DataFrame object elsewhere, it contains all of the information. Hence, manually adding the missing information to the subset of results files is unnecessary.
* make\_dataframe.py takes either the single or multi-machine JSON file and turns it into a pandas DataFrame. pandas is a powerful tool for data processing and analysis. This provides the ability to perform statistical analysis, make graphs, read in the data, and write out the data in several formats, such as .xlsx and .csv.
* v1\_v2\_bm\_name\_converter.py is used to convert the existing benchmark test names contained in the existing results to conform to the (v2) standard. v2 naming goes into effect on benchmark results starting 2019-12-05.

Data Analysis:

* benchmark\_intra\_run.py compares single machine results from two benchmarks for a single run-id. This script performs that analysis and saves the analysis as a PDF, which includes information about the run e.g. the version of HELICS used, the platform, etc. Graphs of the results are created, saved as PNG files, and included in the report.
* benchmark\_intra\_run\_pdf.py creates the report for the benchmark\_intran\_run.py file.
* bmk\_plotting.py provides the capability of creating any plot for any analysis desired. bmk\_plotting.py uses HoloViews’ hvplot because it works well with pandas DataFrame objects. HoloViews contains Python plotting mechanisms such as matplotlib and bokeh. This script has room for expansion on other potential desired graphs for analysis.
* bmk\_tracking\_pdf.py provides a basis for analyzing HELICS performance over time. It takes the single machine key benchmark results, creates graphs for each of those benchmarks, and adds them to an analysis report. For simplicity, the functionality for creating a report and performing analysis are contained in the same script.
* cross\_run\_id.py compares single machine benchmark results for multiple run-ids. This script performs that analysis and saves the analysis as a PDF, which includes information about the runs e.g. the version of HELICS used, the platform, etc. Graphs of the results are created, saved as PNG files, and included in the report.
* cross\_run\_id\_pdf.py creates the report for the cross\_run\_id.py file.
* multinode\_analysis\_pdf.py compares multi-machine benchmark results. This script performs that analysis and saves the analysis as a PDF, which includes information about the run e.g. the version of HELICS used, the platform, etc. Graphs of the results are created, saved as PNG files, and included in the report. For simplicity, the functionality for creating a report and performing analysis are contained in the same script.
* standard\_analysis.py compares single machine results for a single run-id. This script performs that analysis and saves the analysis as a PDF, which includes information about the run e.g. the version of HELICS used, the platform, etc. Graphs of the results are created, saved as PNG files, and included in the report.
* standard\_analysis\_pdf.py creates the report for the standard\_analysis.py file.
* summary\_spreadsheet.py provides a summary for either the single or multi-machine results in tabular form. This script also creates metrics to evaluate HELICS’ performance. This script saves this analysis as .csv and .xlsx files. This script also creates a separate file that contains the metadata for the runs.