**Registration Benchmark results\_visualize.py Manual**

1. Program Overview

**results\_visualize.py** is a Python that reads data from a JSON metrics file, processes it, and generates tabular results and visualization graphs including bar charts and heatmaps.

2. Function Descriptions

**read\_metrics\_file(file\_path: str) -> list or None**

This function takes a string file\_path as an argument and reads a metrics file line by line. Each line of the file is parsed as a JSON object. It returns a list of these JSON objects if the file exists; otherwise, it returns None.

**display\_and\_save\_results(algorithms: list, dataset\_names: list, rte\_thresholds: list, rre\_thresholds: list, results\_filename: str, metrics\_file\_path: str) -> None**

This function takes a list of algorithms, a list of dataset\_names, two lists of thresholds (for RTE and RRE), a results\_filename and a metrics\_file\_path as arguments. It calculates the mean of each metric, constructs a table, prints the table to the console, and saves it as a CSV file with the name provided by results\_filename.

**plot\_bar\_chart(dataset\_name: str, table\_data: list, metrics: list, algorithms: list, header: list) -> None**

This function takes a dataset\_name, table\_data (which is a list of lists, each representing a row in the table), a list of metrics, a list of algorithms, and a list of headers. For each metric, it generates a bar chart comparing the average values across different algorithms for the given dataset. The bar charts are saved as .png files in a directory named 'graphresults'.

**plot\_heatmaps(table\_data: list, header: list, units: dict) -> None**

This function takes table\_data, header (which is a list of strings, column names for the table data) and units (a dictionary with metrics as keys and their units as values). It generates heatmaps for each metric using Seaborn's heatmap function. The heatmaps are saved as .png files in a directory named 'graphresults'.

**plot\_result\_heatmaps() -> None**

This function generates heatmaps for each metric in the results summary, which is read from a CSV file named 'results\_summary.csv'. The heatmaps are saved as .png files in a directory named 'graphresults'. The units of metrics are defined within this function.

3. Imports and Dependencies

* **numpy**: Used for numerical operations such as calculating mean.
* **pathlib**: Used for handling file paths.
* **tabulate**: Used for creating visually appealing tables in console output.
* **os**: Used for operating system dependent functionalities such as checking if a file or directory exists.
* **json**: Used for parsing JSON objects.
* **matplotlib.pyplot** and **seaborn**: Used for generating plots.
* **pandas**: Used for data manipulation and analysis, especially for working with tabular data.

This script requires Python 3.6 or later. You will also need the following Python packages, which can be installed using pip:

* numpy
* pathlib
* tabulate
* os
* json
* matplotlib
* seaborn
* pandas

4. Operation/Usage

To run the `main.py` script, use the following command-line syntax:

To use the **main.py** script, run the following command in your terminal, replacing the placeholders with appropriate values:

python main.py --datasets [your\_datasets] --algorithms [your\_algorithms] --voxelsize [your\_voxelsize] --overlap [your\_overlap] --range\_t [your\_range\_t] --range\_r [your\_range\_r] --rte-thresholds [your\_rte\_thresholds] --rre-thresholds [your\_rre\_thresholds] -o [your\_results\_filename]

A typical usage example (using some defaults) would be :

python main.py --datasets 'eth' --algorithms 'ICP' -o 'my\_results.csv'

5. Output and Interpretation

The program displays the message: "Analyzing experiment results..." upon completion of experiments. The results are displayed and saved to the specified file. Heatmaps of the results are also generated and displayed.