Reproducibility Report for ACM SIGMOD 2023 Paper: "Discovering Similarity Inclusion Dependencies"

ALEXANDER KRAUSE, Technische Universität Dresden, Germany
PRAJNA UPADHYAY, BITS Pilani Hyderabad, India

The core thesis of the paper was easily reproducible. The authors provided a set of scripts that automatically execute the experiments, collect the data and plot the charts. The resulting figures were near identical to the submitted paper, on a relative basis. Absoloute performance numbers varied due to different hardware.

1 INTRODUCTION

The paper [1] relaxes the traditional definition of inclusion dependencies towards *similarity* inclusion dependencies. This allows their system, Sawfish, to find inclusion dependencies even if the surveyed data is dirty and thus traditional constraints may be violated.

2 SUBMISSION

The authors provide a selfcontained github repository. Inside the repository, all necessary information can be found, e.g. how to setup the environment, requried tools, etc. The execution environment is completely encapsulated within a Docker image.

A list with a summary of the submission contents is also useful. For example:

- GitHub repository with code and scripts at: https://github.com/HPI-Information-Systems/Sawfish
- Detailed README file at https://github.com/HPI-Information-Systems/Sawfish/blob/main/REPRODUCIBILITY_INSTRUCTIONS.md
- Experimental data is handled via git LFS

3 HARDWARE AND SOFTWARE ENVIRONMENT

Table 1 compares one of our machines to one of the employed servers from the authors.

Table 1. Hardware & Software environment

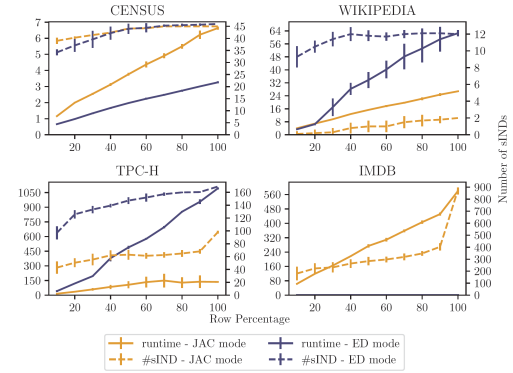
	Paper	Repro Review #1	Repro Reviewer #3
CPU	Intel® Xeon® CPU E5-2650 @ 2.00GHz	Intel® Xeon® Gold 6240R CPU @ 2.40GHz	Intel® Xeon® E-2136 CPU @ 3.30GHz
Cores	2x 8 (16 Threads)	2x 24 (48 Threads)	2x 12 (24 Threads)
RAM	256GB	768GB	64GB

4 REPRODUCIBILITY EVALUATION

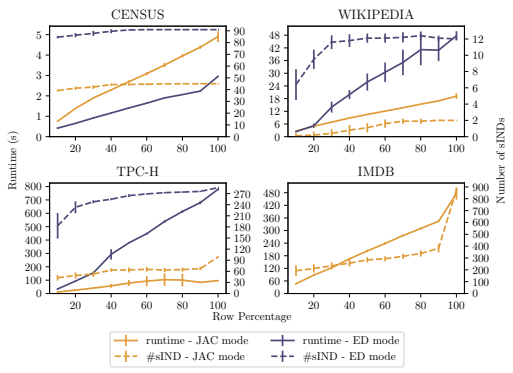
The experiments, data collection and result plotting could be executed fully automated ran for about a week. Make sure to have the correct docker and docker-compose versions on your system.

4.1 Process

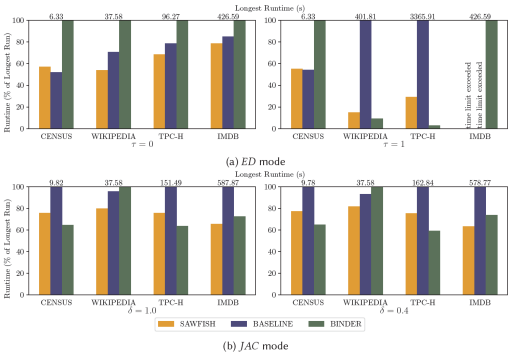
Setting up the repository is straightforward and thoroughly explained in the Github repository. Everything is in place after cloning the repository and initializing with git lfs. Result artifacts, plots and the final paper can be either created via a master script or each step can be executed in isolation with an explicit command, which is also provided.



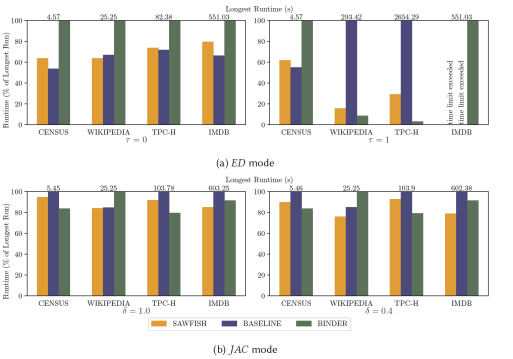
(a) Original Figure 2



(b) Reproduced Figure 2



(c) Original Figure 7



(d) Reproduced Figure 7

4.2 Results

All figures could be reproduced and look mostly similar to the original paper. However, some deviations occurred, most notable in Figure 2 for the CENSUS dataset. Even though the difference in found #sIND is considerable, the runtime is almost equal. This effect occurred, since we did **not** use the *ignoreShortStrings* flag, as the authors, because it was not automatically set. The CENSUS dataset owns a column with 1-sized strings, which would thus be ignored and are easy to find, i.e. the runtime overhead to identify them is negligible. For Figure 7, we reproduced lower "longer runtimes" for all but the IMDB dataset, for which we observed higher "longer runtimes".

5 SUMMARY

The authors put considerable effort into making the reproducibility of their paper as easy and as smooth as possible. Overall, the process to recreate the paper was easy to execute.

REFERENCES

[1] Youri Kaminsky, Eduardo HM Pena, and Felix Naumann. 2023. Discovering Similarity Inclusion Dependencies. *Proceedings of the ACM on Management of Data* 1, 1 (2023), 1–24.