



Reproducibility Report for ACM SIGMOD 2024 Paper: “Fast Maximal Quasi-clique Enumeration: A Pruning and Branching Co-Design Approach”

Hein Meling
hein.meling@uis.no
University of Stavanger
Stavanger, Norway

Kaiqiang Yu
kaiqiang002@e.ntu.edu.sg
Nanyang Technological University
Singapore, Singapore

Yue Gong
yuegong@uchicago.edu
The University of Chicago
Chicago, IL, USA

Cheng Long
c.long@ntu.edu.sg
Nanyang Technological University
Singapore, Singapore

Abstract

The core thesis of the paper has been successfully reproduced. The authors provided a set of scripts that can execute the experiments and generate the individual plots. The resulting figures were near identical to the submitted paper.

CCS Concepts

• **Mathematics of computing** → **Graph algorithms.**

Keywords

cohesive subgraph enumeration; quasi-clique; branch-and-bound

ACM Reference Format:

Hein Meling, Yue Gong, Kaiqiang Yu, and Cheng Long. 2024. Reproducibility Report for ACM SIGMOD 2024 Paper: “Fast Maximal Quasi-clique Enumeration: A Pruning and Branching Co-Design Approach”. In *Reproducibility Reports of the 2024 International Conference on Management of Data (SIGMOD ARI Reports '24)*, June 9–15, 2024, Santiago, AA, Chile. ACM, New York, NY, USA, 3 pages. <https://doi.org/10.1145/3687998.3717040>

1 Introduction

The original paper [1] has been published at ACM SIGMOD 2024:

Kaiqiang Yu and Cheng Long. 2023. Fast Maximal Quasi-clique Enumeration: A Pruning and Branching Co-Design Approach. *Proc. ACM Manag. Data* 1, 3, Article 211 (September 2024), 25 pages.

Summary of the Original Paper. The paper addresses the problem of extracting maximal γ quasi-cliques (MQC) from graphs, a fundamental task in graph analysis with applications such as community detection and biomolecule structure discovery. The paper introduces *FastQC*, a novel branch-and-bound algorithm that improves on previous methods by combining new pruning techniques and branching strategies, achieving better worst-case time complexity ($O^*(\alpha_k^n)$ with $\alpha_k < 2$). Furthermore, a divide-and-conquer strategy

enhances *FastQC*'s performance. Experimental results demonstrate that *FastQC* outperforms state-of-the-art methods by up to two orders of magnitude in real data sets.

Summary of the Reproducibility Result. The reproducibility package recreates the main experiments in the paper and the reproduced results are trend-wise similar to the results reported in the original publication [1]. The main claim that *FastQC* outperforms the state-of-the-art seems to hold.

2 Submission

The reproducibility package provides a set of scripts, two scripts for each of the figures and subfigures (Figure 7, Figure 8(a)-(d), and Figure 9(a)-(d)) in the original paper. The first script for each figure, named `Figure*-Exp.sh` are used to run the corresponding experiment. The second script for each figure, named `Plot_Fig*.sh`, are used to generate the plots. The figures assess the running time of *FastQC* and its competitor *Quick+* over a variety of datasets, and two different parameters, θ and γ , showing a significant advantage to *FastQC* over *Quick+*.

The reproducibility package on GitHub provides a README file that gives an overview of the two algorithms, the hardware and software requirements (packages and libraries needed), and a link to the KONECT website where the data sets used in the experiments can be found. The GitHub repository includes four of the smaller datasets in a zip file.

Furthermore, the readme provides detailed instructions for building the *FastQC* and *Quick+* algorithms, and instructions for running each experiment and generating the plots.

- GitHub repository with code, scripts and detailed instructions at: <https://github.com/KaiqiangYu/SIGMOD24-ARI>
- Data sets are available at: <https://github.com/KaiqiangYu/SIGMOD24-MQCE>

3 Hardware and Software Environment

The paper used a Linux machine with a 2.10 GHz Intel CPU and 128 GB memory. For the reproduction, a virtual machine running Ubuntu 22.04.5 LTS (GNU/Linux 5.15.0-130-generic x86_64) was used.



This work is licensed under a Creative Commons Attribution 4.0 International License. *SIGMOD ARI Reports '24*, Santiago, AA, Chile
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1179-4/2024/06
<https://doi.org/10.1145/3687998.3717040>

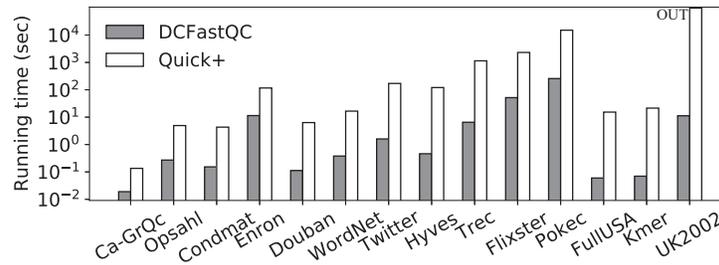


Figure 1: Original: Comparison on all real datasets (Figure 7)

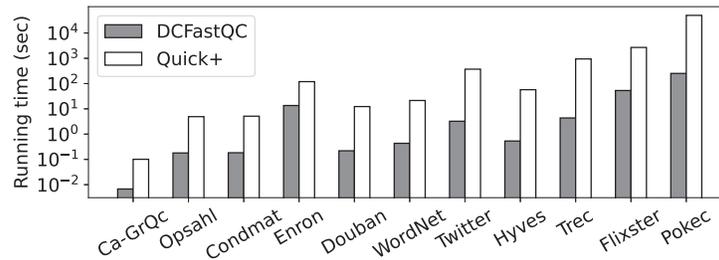


Figure 2: Reproduction: Comparison on all real datasets (Figure 7)

Table 1: Hardware & Software environment

	Paper	Repro
CPU	Intel Xeon Gold 6230R	AMD EPYC 7763
Cores	26	64
Clock	2.10 GHz	2.45 GHz
RAM	128 GB	32 GB
Storage	2 TB HDD	3 TB HDD

4 Reproducibility Evaluation

4.1 Process

The reproducibility process involved several stages, including setup, execution, and debugging to address specific challenges. While the overall artifact was well-documented and functional, certain aspects required additional attention.

The input paths in the provided scripts (e.g., `Figure7-Exp.sh`) were initially incorrect, which required manual correction. Additionally, running scripts for figures such as `Plot_Fig7.py` and `Plot_Fig8c.py` encountered errors due to missing baseline outputs, misaligned data shapes, and issues converting strings to floats. These problems were raised with the authors, who responded promptly by providing a revised artifact hosted on GitHub.

The updated artifact addressed most issues. Changes included fixing typos, adding log-printing commands to improve execution transparency, and revising the output format for better compatibility across platforms. Despite these fixes, reproducing certain figures (e.g., Figure 8(d) and Figure 9(d)) required further iterations due to platform-specific inconsistencies, especially on macOS with an M2 processor compared to Linux on the `x86_64` architecture.

These differences highlighted an intriguing aspect of platform dependence, which the authors acknowledged as worth investigating in future work.

Ultimately, the experiments were successfully reproduced on Linux, confirming trends consistent with the paper’s claims. The remaining platform-specific variations warrant further exploration to understand their underlying causes.

4.2 Results

The reproducibility evaluation successfully replicated key findings from the original paper, focusing on Figures 7, 8(a)-(d), and 9(a)-(d). Below are the main outcomes:

- **Figures 7, 8(a)-(d), 9(a)-(d):** The reproduced figures maintained the same trends as the original, confirming the validity of the key results. Variations in numerical values, attributed to differences in hardware configurations, were observed but did not affect the overall conclusions.
- **Figure 7:** The three right-most bars (datasets FullUSA, Kmer, UK2002) were not reproduced due to the large size of the datasets.
- **Figures 10-12:** These were not included in the scope of reproduction.
- **Platform-Specific Trends:** On macOS with an M2 processor, slight variations in trends were noted compared to Linux (`x86_64`), suggesting platform-specific performance characteristics. This warrants further investigation.

5 Summary

The reproduction confirmed the main findings of the paper with minor discrepancies attributed to platform differences and dataset

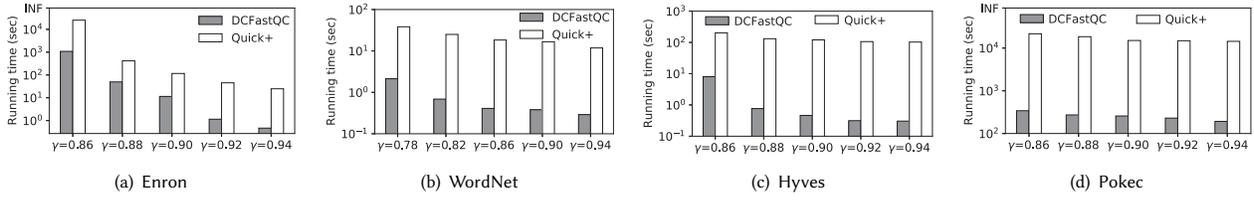


Figure 3: Original: Comparison by varying γ (Figure 8)

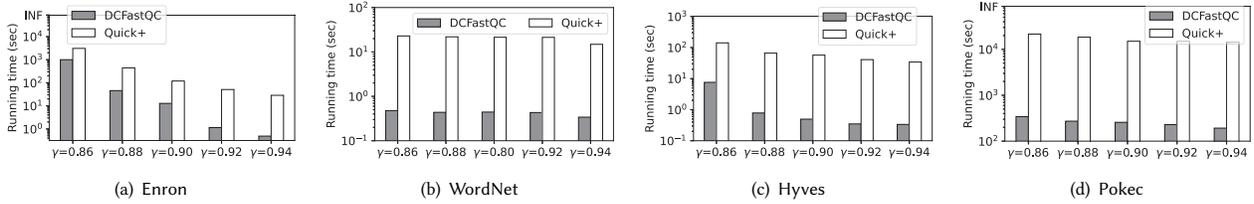


Figure 4: Reproduction: Comparison by varying γ (Figure 8)

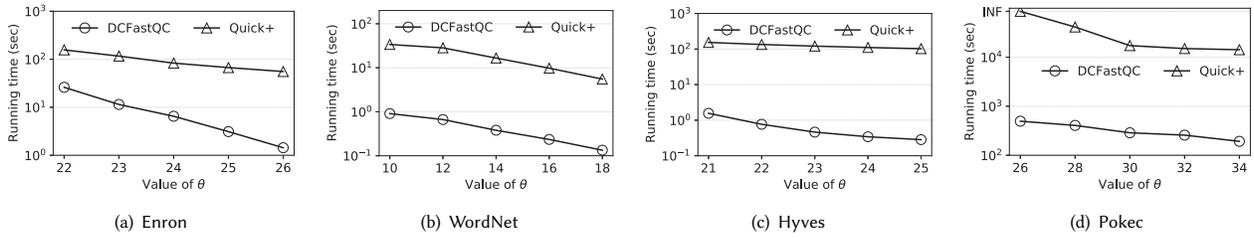


Figure 5: Original: Comparison by varying θ (Figure 9)

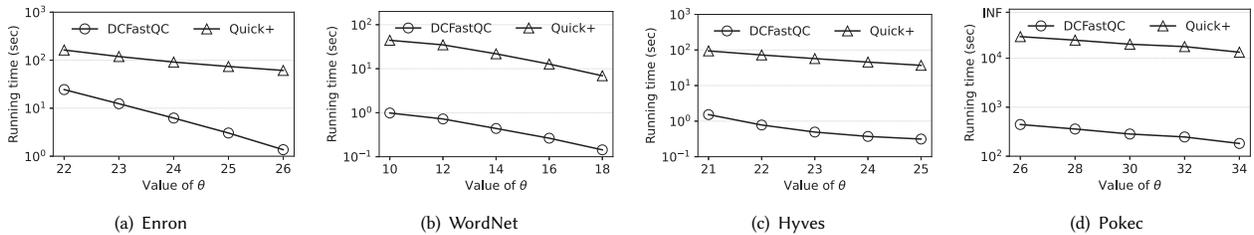


Figure 6: Reproduction: Comparison by varying θ (Figure 9)

constraints. The collaborative effort with the authors was instrumental in achieving these results.

References

[1] Kaiqiang Yu and Cheng Long. 2023. Fast Maximal Quasi-clique Enumeration: A Pruning and Branching Co-Design Approach. *Proceedings of the ACM on Management of Data* 1, 3 (2023), 1–26.