# Minor Project Proposal: MSc Bioinformatics (III) 2025-26

| S.No. | Item | Details | |
|---|---|---|---|
| **1.** | **Project Title** | **TracEon:** An Exploratory C++ Implementation of an In-Memory Cache for Accelerating Genomic Data Access | |
| **2.** | **Proposer(s)** | **Trainee:** Adnan Raza | **Mentor:** Prof. K. Mustafa |
| **3.** | **Learning Outcomes** | ● Demonstrate mastery of core C++ principles (memory management, data structures, file I/O) through the practical design of a high-performance system.<br>● Gain expertise in performance analysis and empirical benchmarking to quantify the trade-offs between disk-based and in-memory data access strategies.<br>● Develop a deep, intuitive understanding of the computational challenges inherent in modern, data-intensive bioinformatics.<br>● Bridge the critical gap between biological domain knowledge and fundamental computer science, creating a strong foundation for future research. | |
| **4.** | **Description** | This project addresses the evolved performance bottleneck in modern bioinformatics, where the challenge is not just disk I/O latency but also the repetitive, CPU-intensive cost of parsing massive genomic files. It proposes the development of "TracEon," an experimental in-memory key-value store built from scratch in C++. The project's primary goal is educational and exploratory, to serve as a 'learning vehicle' for mastering the low-level mechanics of high-performance data handling. By implementing a "parse once, access many" model, TracEon will provide a tangible platform for analyzing and understanding the performance gains of in-memory computing, thereby building a critical skillset for advanced computational research. | |

| | | |
|---|---|---|
| **5.** | **Goals** | ● To build a robust, portfolio-quality C++ application that showcases the ability to design and implement a solution to a relevant bioinformatics problem.<br>● To conduct a rigorous performance analysis that empirically validates the advantages of an in-memory caching model for genomic data.<br>● To establish a strong foundational codebase and a clear intellectual framework that can be directly extended into a Major Project or Master's thesis. |
| **6.** | **Objectives** | ● To design and implement a stable, single-threaded, in-memory key-value store in C++ for string-based biological data.<br>● To develop a simple command-line interface for data manipulation and a robust persistence mechanism for saving/loading the cache state.<br>● To write a suite of benchmarking scripts to systematically compare the data retrieval times from TracEon versus traditional file parsing.<br>● To produce a comprehensive final report and a well-documented public GitHub repository suitable for review by academic supervisors. |
| **7.** | **Issues & Challenges** | ● Efficient memory management to handle large biological datasets without excessive consumption.<br>● Ensuring data integrity and atomicity during save/load operations (persistence).<br>● Designing a simple yet powerful command-line interface that is intuitive for bioinformaticians.<br>● Achieving significant, measurable performance gains over traditional file-parsing methods. |
| **8.** | **Platform, Tools & Methods** | ● **Platform:** Linux / macOS / Windows<br>● **Language:** C++ (C++17 standard or higher)<br>● **Tools:** CLion IDE, Git, GitHub, CMake<br>● **Methods:** Object-Oriented Design, Performance Benchmarking, Data-Driven Analysis. |

| | | |
|---|---|---|
| **9.** | **Deliverables** | ● A functional command-line prototype of the "TracEon" system.<br>● A comprehensive Final Project Report detailing the architecture, implementation, and a rigorous performance analysis.<br>● A professional, public GitHub repository serving as a permanent portfolio piece. |
| **10.** | **Action Plan** | ● **Phase-I:** Research, System Architecture, and Core Data Structure Design (1 month)<br>● **Phase-II:** Implementation of Core Engine, CLI, and Persistence Layer; Unit Testing (2 months)<br>● **Phase-III:** Development of Benchmarking Suite, Performance Analysis, and Final Report Writing (1 month) |
| **11.** | **Future Scope** | ● This minor project is explicitly designed as Phase 1 of a potential Master's thesis. The foundational codebase will enable advanced research explorations in the following areas:<br>○ Networking & Concurrency: Evolving TracEon into a multi-threaded, client-server application to serve data to distributed analysis pipelines, exploring challenges in concurrent data access.<br>○ Domain-Specific Querying: Implementing a specialized query language to support biological questions directly (e.g., GET_GC_CONTENT, FIND_MOTIF, FETCH_OVERLAPPING_FEATURES).<br>○ Advanced Data Structures: Moving beyond simple key-value pairs to support complex data types like genomic interval trees or graph structures for interaction networks.<br>○ Advanced Compression: Implementing and benchmarking domain-specific compression algorithms (e.g., 2-bit encoding vs. RLE) against general-purpose libraries like zlib. |

| | | | |
|---|---|---|---|
| **11.** | **Useful References** | 1. Garrido-Martin, D., et al. (2023). **A software engineering perspective on the challenges and opportunities of long-read sequencing analysis.** *Genome Biology*, 24(1), 198. https://doi.org/10.1186/s13059-023-03043-4<br>2. Yin, Y., et al. (2023). **Navigating the data deluge: a review of data-intensive approaches in microbial multi-omics research.** *Briefings in Bioinformatics*, 24(5), bbad287. https://doi.org/10.1093/bib/bbad287<br>3. Breton, B., et al. (2024). **Data-intensive methods for accelerating discovery in cancer genomics.** *Trends in Cancer*, 10(1), 58-71. https://doi.org/10.1016/j.trecan.2023.10.003 | |
| **12.** | **Signature** | **Trainee:** | **Mentor:** |