
Research Interests

My research is on applied algorithms and data structures for managing massive volumes of data in resource constrained environments, particularly in the context of problems with applications to areas of scientific computing such as computational biology. My primary goals are both to design new algorithms that can be implemented efficiently in practice and to develop theoretical models that capture the aspects of data intensive problems important to system designers. Currently, my focus is on edge computing for mobile DNA analytics.

Education

- Aug. 2021 – **Doctor of Philosophy, Computer Science and Engineering**,
May 2026 *University at Buffalo, the State University of New York*, GPA 4.0
Jan. 2020 – **Bachelor of Science, Computer Science**,
May 2021 *University at Buffalo, the State University of New York*, GPA 3.96

Professional Experience

- May 2021 – **Research Assistant in Scalable Computing Research Group (SCoRe)**,
Present *University at Buffalo, Amherst, NY*
Sep. 2020 – **Research Aide in Scalable Computing Research Group (SCoRe)**,
May 2021 *University at Buffalo, Amherst, NY*
Aug. 2020 – **Undergraduate Teaching Assistant for Data Structures (CSE 250)**,
Dec. 2020 *University at Buffalo, Amherst, NY*
May 2020 – **Research Intern (NSF REU) in Scalable Computing Research Group (SCoRe)**,
Aug. 2020 *University at Buffalo, Amherst, NY*

Honors and Awards

- Dec. 2022 **Russell L. Agrusa CSE Student Innovation Competition (Second Place)**,
University at Buffalo
Dec. 2022 **CSE PhD Poster Competition (First Place)**,
University at Buffalo
Aug. 2021 **Presidential Fellowship**,
University at Buffalo

Publications

A.J. Mikalsen and J. Zola. “Coriolis: Enabling Metagenomic Classification on Lightweight Mobile Devices”. In: *Intelligent Systems for Molecular Biology (ISMB)*. 2023, pp. i66–i75

Software

DNAsbt: *Implementation of the compact string B-tree data structure: a practical full-text index for performing exact pattern matching in external storage.*

<https://gitlab.com/SCoRe-Group/dnasbt>

- Designed the compact string B-tree and formally proved its theoretical properties, such as its I/O optimality, generality to multiple sequences, and alphabet independence.
- Implemented the compact string B-tree as a C++ library, featuring a parallel construction algorithm, on-the-fly compression and decompression, and a concurrent LFU cache.
- Introduced abstractions enabling DNAsbt to work with numerous interchangeable data managers, including different caching policies, memory mapping, and direct I/O.
- Demonstrated empirically that compact string B-trees are extremely effective in practice, easily achieving 30× speedup over the competing methods (e.g., FM-Index).

SMARTEn: *Programming model and C++ framework for mobile DNA analytics*

<https://cse.buffalo.edu/~jzola/smarten/>

- Co-designed and co-implemented SMARTEn's parallel runtime system for automatically parallelizing and optimizing algorithms written in our framework.
- Utilized compact string B-trees to design and implement Coriolis, a metagenomic classifier capable of performing DNA analytics on mobile devices.
- Demonstrated that Coriolis can run in-step with DNA sequencing and basecalling with only MBs of memory, enabling real-time and in the field DNA analytics for the first time.

OneDataShare, *Distributed SaaS for optimized data transfers over WAN,*

<https://onedatashare.org/>

- Redesigned the service's REST API in collaboration with other project members to decouple the REST API from the front-end and enable effective programmatic use.
- Developed an abstract, high-level design for the service's SDK to be used for both C++ and Python implementations, as well as by the Python CLI.
- Implemented the service's C++ SDK.

Service

Reviewer

- International Conference on Parallel Computing (ICPP), 2023.
- International Parallel and Distributed Processing Symposium (IPDPS), 2024.

Skills

Languages: C++, C, Rust, Python, Bash, Go, Java, Scala, TLA⁺, ARM Assembly

Development Tools: Git, CMake, Vim, Neovim, VS Code, IntelliJ, Eclipse, Jupyter

Operating Systems: Linux, Unix, macOS, Windows

Technical Knowledge: Parallel Computing, Edge Computing, Internet of Things, Big Data, Distributed Systems, Operating Systems, Algorithms, Data Structures, Databases, Bioinformatics, Machine Learning