BLISS: Basic Life Sciences Subroutines for Next Generation Sequence Analysis

Tony Pan, Patrick Flick, and Srinivas Aluru
School of Computational Science and Engineering, Georgia Institute of Technology, Atlanta, GA 30332

Introduction

Next Generation Sequencing (NGS) has enabled rapid and low cost DNA and RNA sequencing for whole genome and transcripts. Sequencing cost is shrinking to just $1000 for reading 600 billion bases in a single experiment, enough to sample a human genome 200 times over.

The high throughput nature of NGS technology leads to Big Data challenges in storage, management, and analysis of petabytes of data as biomedical research expands and clinical practices adopt the use of genomic data. However, majority of existing bioinformatics software is serial in nature.

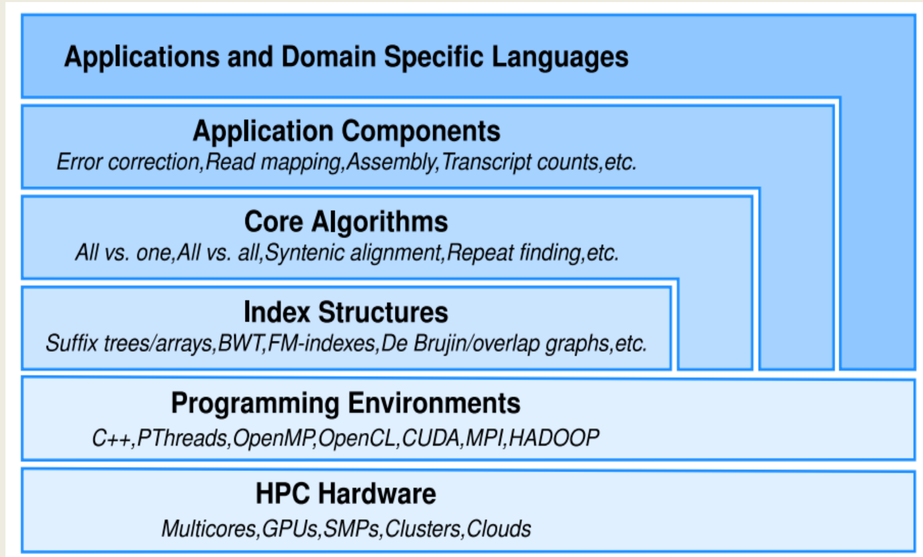
Basic Life Sciences Subroutines, or BLISS, aims to provide the bioinformatics community with a set of fast and efficient distributed data structures and parallel algorithms. These building blocks will enable the community to easily create efficient parallel applications for de novo genome assembly, resequencing, transcriptome sequencing, comparative genomics, metagenomics, in silico gene expression analysis, and others.

Acknowledgement

This work is supported by a mid-scale Big Data award from the National Science Foundation (IIS-1416259).

Architecture

The BLISS library consists of specification (API) and implementation. The goal of this separation is to enable a community of bioinformatics researchers and computer scientists to provide implementations for different algorithms and hardware architectures, yet remain interoperable at the specification level. Parallel algorithms are implemented in BLISS to efficiently leverage distributed and parallel architectures.



Implementation

BLISS, while implementing efficient parallel data structures and algorithms, also focuses on usability and performance:

Consistency and Correctness

- C++ compile-time type checking and template instantiation
- Regression testing

Performance

- Parallelized implementation using MPI, OpenMP, and std threads.
- Optimized implementation for data types and architectures
- C++ Templates minimize use of inheritance and virtual functions

Ease of Use

- Use Iterator and Interval abstractions for streaming data and parallel computation
- Minimal external library dependencies

Distributed k-mer Index

K-mers are DNA or RNA sequences with length k. K-mers are useful as indices into a longer sequence (e.g. whole genome) or set of sequences (e.g. NGS reads). K-mer indexing is a fundamental building block for applications ranging from error correction to sequence assembly.

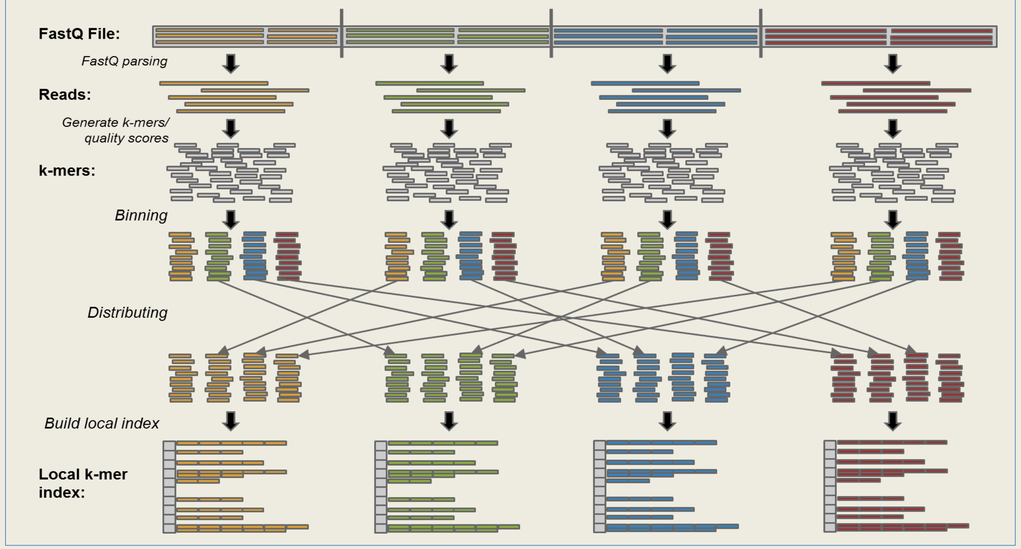
API

```
create(sequences)
update(sequences)
query_frequency(k-mer)
query_occurances(k-mer)
frequency_distribution()
...
```

Implementation

Distributed k-mer index is created through

- parallel file I/O
- multi-threaded, streaming k-mer indexing of input sequences
- incremental, non-blocking MPI-based index re-distribution
- local k-mer hash table insertion.



Summary

BLISS is an evolving open source library that currently provides parallel K-mer indexing, and aims to support a variety of basic parallel data structures and algorithms for bioinformatics. The library will enable bioinformaticians to easily create efficient parallel applications and workflows to facilitate big data analysis for scientific discovery.