SneakySnake: A New Fast and Highly Accurate Pre-Alignment Filter on CPU and FPGA for Accelerating Sequence Alignment

Mohammed Alser¹,² Can Alkan² Onur Mutlu¹,²,³

1 ETH zurich 2 Bilkent University 3 Carnegie Mellon

1: Read Mapping

Fact: It remains challenging to sequence the entire DNA molecule as a whole.

As a workaround: high throughput DNA sequencing (HTS) technologies can sequence only segments of the original molecule. This is relatively quick and cost-effective but it results in an excessive number of genomic reads.

Hence we need read mapping to link the reads together and construct back the donor’s complete genome by 1) determining the location of each read within reference genome and 2) calculating its optimal sequence alignment.

2: Problem

Calculating sequence alignment is a major performance bottleneck:

- Uses computationally expensive dynamic programming algorithms.
- Bottlenecked by memory bandwidth, e.g., Illumina NovaSeq 6000 generates 6 Terabases in < 24 hours
- They are unavoidable as they provide accurate information about the quality of the alignment.
- Majority of candidate locations in the reference genome do not align with a given read due to high dissimilarity.

3: Our Goal

Significantly reduce the time spent on calculating the sequence alignment of short sequences using pre-alignment filtering.

To this end: We introduce new, fast, and very accurate pre-alignment filters, SneakySnake (for CPU) and Snake-on-Chip (for FPGA).

4: Key Ideas

Quickly and accurately filters out highly dissimilar sequence pairs before applying sequence alignment algorithms.


Judiciously leverages the parallelism-friendly architecture of modern FPGAs to greatly speed up the SneakySnake algorithm.

5: Single Net Routing (SNR) Problem

SNR Problem: Finding the optimal routing path that:

- Includes the least number of horizontal escape segments,
- Passes through the minimum number of obstacles,
- Connects two IO terminals on a special grid layout.

The number of obstacles in the solution to the SNR problem is a lower bound on the actual number of edits between two genomic sequences.

Solving the SNR problem is much faster than solving the sequence alignment problem, as calculating the routing path after facing an obstacle is independent of the calculated path before this obstacle.

6: SneakySnake Walkthrough

Dataset Description: Set_1 & Set_2: each has 30 million pairs from mapping ERR240727_1 to the human genome using mrFAST’s e= 2, 40, respectively.

Set_3 & Set_4: each has 30 million pairs from mapping SRR82647_1 using mrFAST’s e= 8, 100, respectively.

Filtering Accuracy vs. Existing Filters

Filtering Accuracy vs. Existing Filters

7: Evaluation & Key Takeaways

Speedup vs. Existing Filters

Key Results

- < 31412×, 20603×, and 64.1× fewer falsely-accepted sequences compared to Gatekeeper / SHD (using Set_4, E= 10%) Shouji (using Set_4, E= 10%), and MAGNET (using Set_1, E= 1%), respectively.

- < 37.6× and 43.9× speedup with the addition of SneakySnake to Edlib (Šolíč, Bioinformatics 2017) [using Set_4, E= 0%] and Parasail [Daily+, Bioinformatics BMC 2016] (using Set_4, E= 2%), respectively.

- < 154.7× and 150.2× speedup with the addition of Snake-on-Chip to Edlib (Šolíč, Bioinformatics 2017) [E= 0%] and Parasail [Daily+, Bioinformatics BMC 2016] (E= 0%), respectively. < 1.4×, 3.4×, and 1.8× more speedup compared to that provided by adding Shouji, MAGNET, and Gatekeeper as a pre-alignment filter, respectively.

SneakySnake & Snake-on-Chip

- Open-source: [https://github.com/CMU-SAFARI](https://github.com/CMU-SAFARI)
- do not replace sequence alignment step.
- do not sacrifice any of the sequence aligner capabilities (scoring and backtracking), as they do not modify the aligner.

Contact Us: [https://people.inf.ethz.ch/alserm/](https://people.inf.ethz.ch/alserm/)