

SeGraM: A Universal Hardware Accelerator for Genomic Sequence-to-Graph and Sequence-to-Sequence Mapping

Damla Senol Cali (damlasenolcali@gmail.com), Konstantinos Kanellopoulos, Joel Lindegger, Zulal Bingol, Gurpreet S. Kalsi, Ziyi Zuo, Can Firtina, Meryem Banu Cavlak, Jeremie Kim, Nika Mansouri Ghiasi, Gagandeep Singh, Juan Gomez-Luna, Nour Almadhoun Alserr, Mohammed Alser, Sreenivas Subramoney, Can Alkan, Saugata Ghose, and Onur Mutlu

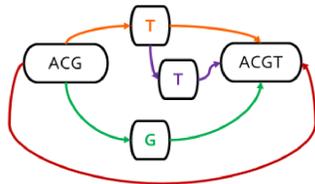


Genome Graphs

Genome graphs:

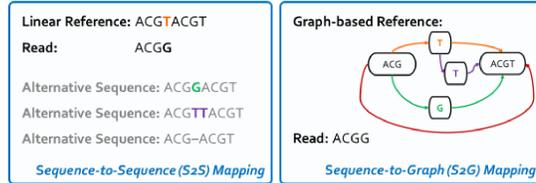
- Combine the **linear reference genome** with the **known genetic variations in the entire population** as a graph-based data structure
- Enable us to move away from aligning with a single linear reference genome (**reference bias**) and **more accurately express the genetic diversity in a population**

Sequence #1: ACGTACGT
Sequence #2: ACGGACGT
Sequence #3: ACGTTACGT
Sequence #4: ACGACGT



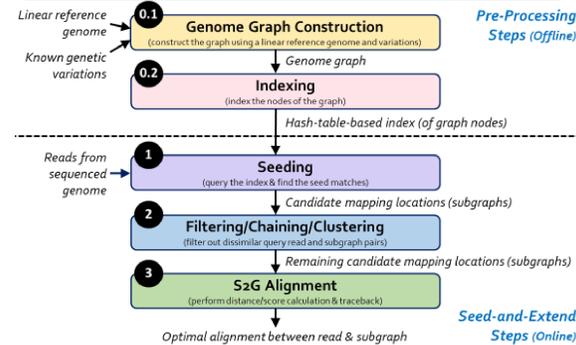
Genome Sequence Analysis

Mapping the reads to a reference genome (i.e., **read mapping**) is a **critical step** in genome sequence analysis

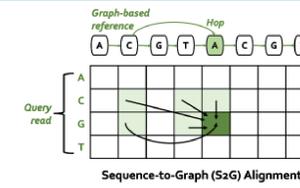


Sequence-to-graph mapping results in **notable quality improvements**. However, it is a **more difficult** computational problem, with **no prior hardware design**.

Sequence-to-Graph Mapping Pipeline



S2S vs. S2G Alignment



In contrast to **S2S alignment**, **S2G alignment** must incorporate **non-neighboring characters** as well whenever there is an edge (i.e., **hop**) from the non-neighboring character to the current character

Analysis of State-of-the-Art Tools

- Based on our analysis with **GraphAligner** and **vg**:
 - Observation 1:** Alignment step is the bottleneck
 - Observation 2:** Alignment suffers from high cache miss rates
 - Observation 3:** Seeding suffers from the DRAM latency bottleneck
 - Observation 4:** Baseline tools scale sublinearly
- Observation 5:** Existing S2S mapping accelerators are unsuitable for the S2G mapping problem
- Observation 6:** Existing graph accelerators are unable to handle S2G alignment

SeGraM: First Graph Mapping Accelerator

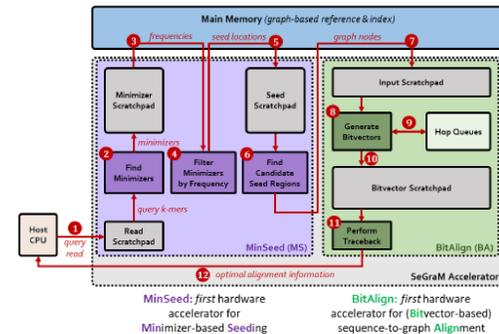
Our Goal: Specialized, high-performance, scalable, and low-cost algorithm/hardware co-design that alleviates bottlenecks in multiple steps of sequence-to-graph mapping

SeGraM: First universal algorithm/hardware co-designed genomic mapping accelerator that can support both sequence-to-graph mapping and sequence-to-sequence mapping, for both short and long reads

- We base SeGraM upon a **minimizer-based seeding algorithm**
- We propose a **novel bitvector-based alignment algorithm** to perform approximate string matching between a read and a graph-based reference genome

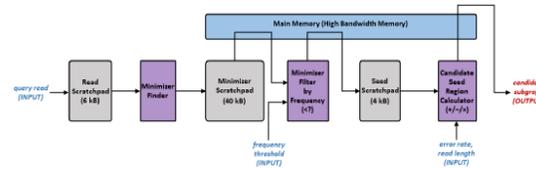
We co-design both algorithms with **high-performance, scalable, and efficient hardware accelerators**

SeGraM Hardware Design



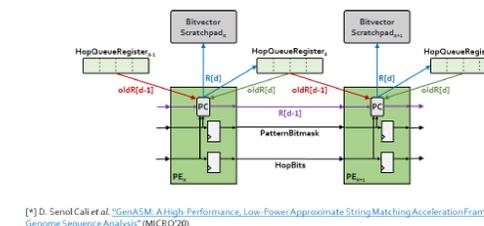
MinSeed HW

- MinSeed = 3 computation modules + 3 scratchpads + memory interface**
 - Computation modules:** Implemented with simple logic
 - Scratchpads:** 50kB in total; employ double buffering technique to hide the latency of MinSeed
 - High-Bandwidth Memory (HBM):** Enables low-latency and highly-parallel memory access

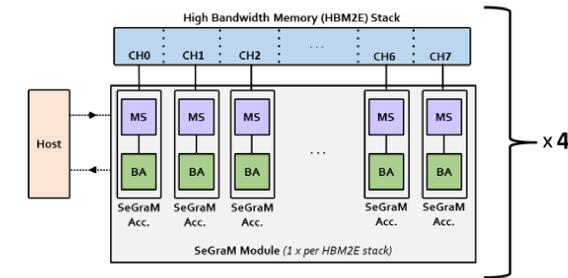


BitAlign HW

- Linear cyclic systolic array-based accelerator**
- Based on the GenASM hardware design*
- Incorporates **hop queue registers** to feed the bitvectors of non-neighboring characters/nodes (i.e., **hops**)



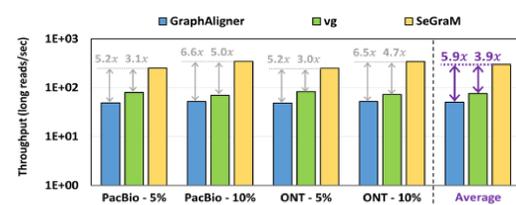
Overall System Design of SeGraM



Use Cases of SeGraM

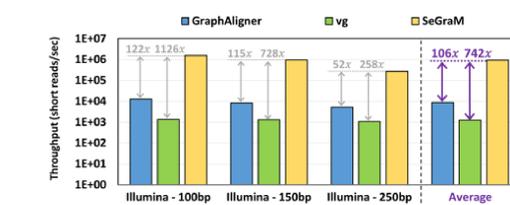
- Sequence-to-Graph Mapping:** MS (Minimizer) and BA (BitAlign)
- Sequence-to-Graph Alignment:** MS or Other and BA
- Sequence-to-Sequence Alignment:** MS or Other and BA
- Seeding:** MS and BA or Other

Key Results – SeGraM with Long Reads



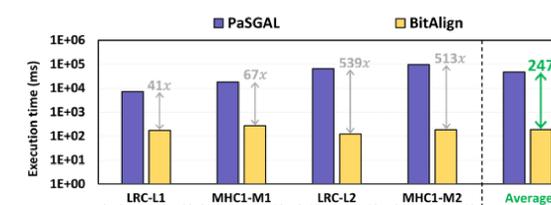
SeGraM provides **5.9x and 3.9x** throughput improvement over GraphAligner and vg, while reducing the power consumption by **4.1x and 4.4x**

Key Results – SeGraM with Short Reads



SeGraM provides **106x and 742x** throughput improvement over GraphAligner and vg, while reducing the power consumption by **3.0x and 3.2x**

Key Results – BitAlign (S2G Alignment)



BitAlign provides **41x-539x** speedup over PaSGAL

Key Results – BitAlign (S2S Alignment)

- BitAlign can also be used for sequence-to-sequence alignment
 - The cost of more functionality: **extra hop queue registers**
 - We do **not sacrifice any performance**
- For long reads (over GACT of Darwin and GenASM):
 - 4.8x and 1.2x** throughput improvement,
 - 2.7x and 7.5x** higher power consumption, and
 - 1.5x and 2.6x** higher area overhead
- For short reads (over SillaX of GenAx and GenASM):
 - 2.4x and 1.3x** throughput improvement

Evaluation Methodology

- Performance, Area and Power Analysis:**
 - Synthesized SystemVerilog models of the MinSeed and BitAlign accelerator datapaths
 - Simulation- and spreadsheet-based performance modeling
- Baseline Comparison Points:**
 - GraphAligner, vg, and HGA for sequence-to-graph mapping
 - PaSGAL for sequence-to-graph alignment
 - Darwin, GenAx, and GenASM for sequence-to-sequence alignment
- Datasets:**
 - Graph-based reference: GRCh38 + 7 VCF files for HG001-007
 - Simulated datasets for both short and long reads

Key Results – Area & Power

Based on our synthesis of MinSeed and BitAlign accelerator datapaths using the Synopsys Design Compiler with a 28nm process (@ 1GHz):

Component	Area (mm ²)	Power (mW)
MinSeed - Logic	0.017	10.8
Read Scratchpad (6 kB)	0.012	7.9
Minimizer Scratchpad (40 kB)	0.055	22.7
Seed Scratchpad (4 kB)	0.008	6.4
BitAlign - Edit Distance Calculation Logic with Hop Queue Registers (64 PEs)	0.393	378.0
BitAlign - Traceback Logic	0.020	2.7
Input Scratchpad (24 kB)	0.033	13.3
Bitvector Scratchpads (128 kB)	0.329	316.2
Total - 1 SeGraM Accelerator	0.867	758.0 (0.8 W)
Total - 4 SeGraM Modules (32 SeGraM Accelerators)	27.744	24.3 W
HBM2E (4 stacks)	--	3.8 W

Additional Details in the Paper

- Details of the **pre-processing steps of SeGraM**
- Details of the **MinSeed and BitAlign algorithms**
- Details of the **MinSeed and BitAlign hardware designs**
- Bottleneck analysis** of the existing tools
- Evaluation methodology details** (datasets, baselines, performance model)
- Additional results** for the three evaluated use cases
- Sources of improvements in SeGraM**
- Comparison of GenASM and SeGraM**

SeGraM [ISCA 2022] – Paper & Talk

SeGraM: A Universal Hardware Accelerator for Genomic Sequence-to-Graph and Sequence-to-Sequence Mapping

Damla Senol Cali¹, Konstantinos Kanellopoulos², Joel Lindegger², Zulal Bingol³, Gurpreet S. Kalsi⁴, Ziyi Zuo⁵, Can Firtina², Meryem Banu Cavlak², Jeremie Kim², Nika Mansouri Ghiasi², Gagandeep Singh², Juan Gomez-Luna², Nour Almadhoun Alserr², Mohammed Alser², Sreenivas Subramoney¹, Can Alkan¹, Saugata Ghose⁶, Onur Mutlu²

¹Bionano Genomics, ²ETH Zürich, ³Bilkent University, ⁴Intel Labs, ⁵Carnegie Mellon University, ⁶University of Illinois Urbana-Champaign

SeGraM – Source Code & Datasets

Source code for the software implementation of SeGraM is available on GitHub: <https://github.com/CMU-SAFARI/SeGraM>

Source code for the datasets is available on GitHub: <https://github.com/CMU-SAFARI/SeGraM-Datasets>