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

Damla Senol Cali, Ph.D.

damlasenolcali@gmail.com
https://damlasenolcali.github.io/

Konstantinos Kanellopoulos, Joel Lindegger, Zulal Bingol, Gurpreet S. Kalsi, Ziyi Zuo, Can Firtina, Meryem Banu Cavlak, Jeremie S. Kim, Nika Mansourí Ghiasi, Gagandeep Singh, Juan Gomez-Luna, Nour Almadhoun Alserr, Mohammed Alser, Sreenivas Subramoney, Can Alkan, Saugata Ghose, Onur Mutlu

Carnegie Mellon  ETH zürich  Bilkent University

intel  University of Illinois Urbana-Champaign  SAFARI
Genome Sequencing

- **Genome sequencing**: Enables us to determine the order of the DNA sequence in an organism’s genome
  - Plays a pivotal role in:
    - Personalized medicine
    - Outbreak tracing
    - Understanding of evolution

- Modern genome sequencing machines extract smaller randomized fragments of the original DNA sequence, known as **reads**
  - **Short reads**: a few hundred base pairs, error rate of ~0.1%
  - **Long reads**: thousands to millions of base pairs, error rate of 10–15%
Mapping the reads to a reference genome (i.e., read mapping) is a critical step in genome sequence analysis.

**Linear Reference:** ACGTACGT

**Read:** ACGG

**Alternative Sequence:** ACGGACGT

**Alternative Sequence:** ACGTTACGT

**Alternative Sequence:** ACG-ACGT

**Sequence-to-Sequence (S2S) Mapping**

**Graph-based Reference:**

**Read:** ACGG

**Sequence-to-Graph (S2G) Mapping**

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

- Sequence-to-graph mapping
- Sequence-to-sequence mapping
- Both short and long reads
Use Cases & Key Results

(1) Sequence-to-Graph (S2G) Mapping
   - 5.9×/106× speedup, 4.1×/3.0× less power than GraphAligner for long and short reads, respectively (state-of-the-art SW)
   - 3.9×/742× speedup, 4.4×/3.2× less power than vg for long and short reads, respectively (state-of-the-art SW)

(2) Sequence-to-Graph (S2G) Alignment
   - 41×–539× speedup over PaSGAL with AVX-512 support (state-of-the-art SW)

(3) Sequence-to-Sequence (S2S) Alignment
   - 1.2×/4.8× higher throughput than GenASM and GACT of Darwin for long reads (state-of-the-art HW)
   - 1.3×/2.4× higher throughput than GenASM and SillaX of GenAX for short reads (state-of-the-art HW)
Outline

- Introduction
- Background
  - Genome Graphs
  - Sequence-to-Graph Mapping
- SeGraM: Universal Genomic Mapping Accelerator
  - High-Level Overview
  - MinSeed
  - BitAlign
  - Use Cases
- Evaluation
- Conclusion
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
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
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
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
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
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 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
Sequence-to-Graph Mapping Pipeline

**Pre-Processing Steps (Offline)**

1. **Genome Graph Construction**
   - (construct the graph using a linear reference genome and variations)

2. **Indexing**
   - (index the nodes of the graph)

**Seed-and-Extend Steps (Online)**

1. **Seeding**
   - (query the index & find the seed matches)

2. **Filtering/Chaining/Clustering**
   - (filter out dissimilar query read and subgraph pairs)

3. **S2G Alignment**
   - (perform distance/score calculation & traceback)

**Optimal alignment between read & subgraph**
S2S vs. S2G Alignment

\[ \text{Single linear reference} \]

\[ \text{Sequence-to-Sequence (S2S) Alignment} \]

\[ \text{Query read} \]
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.

**S2S vs. S2G Alignment**

**Sequence-to-Graph (S2G) Alignment**

**Query read**

**Graph-based reference**

**Hop**

A C G T A C G T
## Analysis of State-of-the-Art Tools

Based on our analysis with **GraphAligner** and **vg**:  

<table>
<thead>
<tr>
<th>Observation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Alignment step is the bottleneck</td>
</tr>
<tr>
<td>2</td>
<td>Alignment suffers from high cache miss rates</td>
</tr>
<tr>
<td>3</td>
<td>Seeding suffers from the DRAM latency bottleneck</td>
</tr>
<tr>
<td>4</td>
<td>Baseline tools scale sublinearly</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Observation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Existing S2S mapping accelerators are unsuitable for the S2G mapping problem</td>
</tr>
<tr>
<td>6</td>
<td>Existing graph accelerators are unable to handle S2G alignment</td>
</tr>
</tbody>
</table>
Outline

- Introduction
- Background
  - Genome Graphs
  - Sequence-to-Graph Mapping
- SeGraM: Universal Genomic Mapping Accelerator
  - High-Level Overview
  - MinSeed
  - BitAlign
  - Use Cases
- Evaluation
- Conclusion
SeGraM: Universal Genomic Mapping Accelerator

- **First universal genomic mapping accelerator** that can support both sequence-to-graph mapping and sequence-to-sequence mapping, for both short and long reads

- **First algorithm/hardware co-design** for accelerating sequence-to-graph mapping

- 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

Main Memory (graph-based reference & index)

- Minimizer Scratchpad
- Seed Scratchpad
- Find Minimizers
- Filter Minimizers by Frequency
- Find Candidate Seed Regions
- Read Scratchpad
- MinSeed (MS)

- Input Scratchpad
- Generate Bitvectors
- Hop Queues
- Bitvector Scratchpad
- Perform Traceback
- BitAlign (BA)

**MinSeed**: first hardware accelerator for Minimizer-based Seeding

**BitAlign**: first hardware accelerator for (Bitvector-based) sequence-to-graph Alignment

Damla Senol Cali
SeGraM Hardware Design

MinSeed: first hardware accelerator for Minimizer-based Seeding

BitAlign: first hardware accelerator for (Bitvector-based) sequence-to-graph Alignment
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*)

[*] D. Senol Cali et al. "GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework for Genome Sequence Analysis" (MICRO’20)
Overall System Design of SeGraM

High Bandwidth Memory (HBM2E) Stack

CH0  CH1  CH2  ...  CH6  CH7

SeGraM Module (1 x per HBM2E stack)

Host

MS  BA  SeGraM Acc.

MS  BA  SeGraM Acc.

MS  BA  SeGraM Acc.

MS  BA  SeGraM Acc.

MS  BA  SeGraM Acc.

MS  BA  SeGraM Acc.
Use Cases of SeGraM

(1) Sequence-to-Graph Mapping

(2) Sequence-to-Graph Alignment

(3) Sequence-to-Sequence Alignment

(4) Seeding
Outline

- Introduction
- Background
  - Genome Graphs
  - Sequence-to-Graph Mapping
- SeGraM: Universal Genomic Mapping Accelerator
  - High-Level Overview
  - MinSeed
  - BitAlign
  - Use Cases
- Evaluation
- Conclusion
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
Based on our synthesis of MinSeed and BitAlign accelerator datapaths using the Synopsys Design Compiler with a **28nm** process (@ **1GHz**):

<table>
<thead>
<tr>
<th>Component</th>
<th>Area (mm²)</th>
<th>Power (mW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MinSeed – Logic</td>
<td>0.017</td>
<td>10.8</td>
</tr>
<tr>
<td>Read Scratchpad (6 kB)</td>
<td>0.012</td>
<td>7.9</td>
</tr>
<tr>
<td>Minimizer Scratchpad (40 kB)</td>
<td>0.055</td>
<td>22.7</td>
</tr>
<tr>
<td>Seed Scratchpad (4 kB)</td>
<td>0.008</td>
<td>6.4</td>
</tr>
<tr>
<td>BitAlign – Edit Distance Calculation Logic with Hop Queue Registers (64 PEs)</td>
<td>0.393</td>
<td>378.0</td>
</tr>
<tr>
<td>BitAlign – Traceback Logic</td>
<td>0.020</td>
<td>2.7</td>
</tr>
<tr>
<td>Input Scratchpad (24 kB)</td>
<td>0.033</td>
<td>13.3</td>
</tr>
<tr>
<td>Bitvector Scratchpads (128 kB)</td>
<td>0.329</td>
<td>316.2</td>
</tr>
<tr>
<td>Total – 1 SeGrA.M Accelerator</td>
<td>0.867</td>
<td>758.0 (0.8 W)</td>
</tr>
<tr>
<td>Total – 4 SeGrA.M Modules (32 SeGrA.M Accelerators)</td>
<td><strong>27.744</strong></td>
<td>24.3 W</td>
</tr>
<tr>
<td>HBM2E (4 stacks)</td>
<td>--</td>
<td><strong>3.8 W</strong></td>
</tr>
</tbody>
</table>
Key Results – SeGraM with Long Reads

SeGraM provides 5.9× and 3.9× throughput improvement over GraphAligner and vg, while reducing the power consumption by 4.1× and 4.4×
Key Results – SeGraM with Short Reads

SeGraM provides 106× and 742× throughput improvement over GraphAligner and vg, while reducing the power consumption by 3.0× and 3.2×
Key Results – BitAlign (S2G Alignment)

BitAlign provides **41×-539× 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):
  - 4.8× and 1.2× throughput improvement,
  - 2.7× and 7.5× higher power consumption, and
  - 1.5× and 2.6× higher area overhead

- For short reads (over SillaX of GenAx and GenASM):
  - 2.4× and 1.3× throughput improvement
Outline

- Introduction
- Background
  - Genome Graphs
  - Sequence-to-Graph Mapping
- SeGraM: Universal Genomic Mapping Accelerator
  - High-Level Overview
  - MinSeed
  - BitAlign
  - Use Cases
- Evaluation
- Conclusion
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
Conclusion

- **SeGraM**: First universal algorithm/hardware co-designed genomic mapping accelerator that supports:
  - Sequence-to-graph (S2G) & sequence-to-sequence (S2S) mapping
  - Short & long reads
  - **MinSeed**: First minimizer-based seeding accelerator
  - **BitAlign**: First (bitvector-based) S2G alignment accelerator

- SeGraM supports multiple use cases:
  - End-to-end S2G mapping
  - S2G alignment
  - S2S alignment
  - Seeding

- SeGraM outperforms state-of-the-art software & hardware solutions
SeGraM [ISCA 2022]

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

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


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

Damla Senol Cali¹ Konstantinos Kanellopoulos² Joël Lindegger² Zülal Bingöl³ Gurpreet S. Kalsi⁴ Ziyi Zuo⁵ Can Firtina² Meryem Banu Cavlak² Jeremie Kim² Nika Mansouri Ghiasi² Gagandeep Singh² Juan Gómez-Luna² Nour Almadhoun Alserr² Mohammed Alser² Sreenivas Subramoney⁴ Can Alkan³ Saugata Ghose⁶ Onur Mutlu²

¹Bionano Genomics ²ETH Zürich ³Bilkent University ⁴Intel Labs ⁵Carnegie Mellon University ⁶University of Illinois Urbana-Champaign
SeGraM (Software implementations and datasets will be available soon!)

SeGraM is a universal genomic mapping accelerator that supports both sequence-to-graph mapping and sequence-to-sequence mapping, for both short and long reads. SeGraM consists of two main components: (1) MinSeed, the first minimizer-based seeding accelerator, which finds the candidate mapping locations (i.e., subgraphs) in a given genome graph; and (2) BitAlign, the first bitvector-based sequence-to-graph alignment accelerator, which performs alignment between a given read and the subgraph identified by MinSeed. MinSeed is built upon a memory-efficient minimizer-based seeding algorithm, and BitAlign is built upon our novel bitvector-based, highly-parallel sequence-to-graph alignment algorithm.
SeGraM: A Universal Hardware Accelerator for Genomic Sequence-to-Graph and Sequence-to-Sequence Mapping

Damla Senol Cali, Ph.D.

damlasenolcali@gmail.com
https://damlasenolcali.github.io/

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

Carnegie Mellon
ETH Zürich
Bilkent University
Intel
University of Illinois
SAFARI