Biology Needs Computer Architects
Keeping up with genomic-scale data

Sneha Goenka
AHA affiliates meeting
June 14, 2023
Significantly reducing cost of sequencing a human genome
Genomics growth rate >> CPU performance

Hennessy and Patterson, “A New Golden Age of Computer Architecture”, CACM 2019
GenBank
One millionth of the human genome!
Whom are we closest to?
Comparative genomics enables genome interpretation

A Neanderthal OAS1 isoform protects individuals of European ancestry against COVID-19 susceptibility and severity

Chimp Genome Helps Scientists Learn More About Human DNA

Prediction of functional elements

‘Conserved’ region

Mutations
Thousand-genome era is already here

> 3 years for comparing genomes; unable to keep up with the assembly rate

81 days on an 800 CPU core cluster for 600-way alignment using Cactus

NCBI database
Armstrong et. al. (Nature 2020)
4 million babies born in the US annually

12% babies admitted to the NICU

1/3 NICU hospitalizations have a genetic cause

40% longer hospitalizations when disorders are genetic

ICU hospitalizations cost $15-20k/day

PMID: 31564432; PMID: 29449963; PMID: 28973083; PMID: 32411386
Faster genetic diagnosis has significant impact
Enabling biologists to leverage sequencing

Comparative Genomics

SegAlign: A Scalable GPU-based whole genome aligner
[Supercomputing Conference]

Darwin-WGA: A fast and highly sensitive co-processor for whole genome alignments
[HPCA]

Clinical Genomics

An ultra-rapid workflow for clinical whole genome sequencing
[New England Journal of Medicine Nature Biotechnology]
Enabling biologists to leverage sequencing

**Comparative Genomics**

SegAlign: A Scalable GPU-based whole genome aligner  
*[Supercomputing Conference]*

Darwin-WGA: A fast and highly sensitive co-processor for whole genome alignments  
*[HPCA]*

**Clinical Genomics**

An ultra-rapid workflow for clinical whole genome sequencing  
*[New England Journal of Medicine Nature Biotechnology]*
Whole Genome Alignment (WGA) – first step in comparative genomics

‘Conserved’ region

Mutations

Prediction of functional elements

Pairwise Whole Genome Alignment

3 billion bases
Understanding whole genome alignment

<table>
<thead>
<tr>
<th>Match</th>
<th>Deletion</th>
<th>Insertion</th>
</tr>
</thead>
<tbody>
<tr>
<td>human</td>
<td>ACCTATTGTTTTTTTTGTAAAAATATA</td>
<td></td>
</tr>
<tr>
<td>chimp</td>
<td>ACCTATTGTTTTTTTTGTAAAAATATA</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>human</td>
<td>TGTTGAAAAAGGAAGTGACTATATATAT</td>
<td></td>
</tr>
<tr>
<td>chimp</td>
<td>TGTTGAAAAAGGAAGTGACTATATATAT</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>GGTTGATACCTTTTTTTTGTTT</td>
</tr>
</tbody>
</table>

LASTZ is the state-of-the-art whole genome aligner, based on the **seed-filter-extend** algorithm

Seeding finds small, local matching base-pairs

Seed hit
R ...CTTGGGTATTCCGTA...
Q ...CTTGGGTATTCCTTA...
Seed

Seed hit

Query (Q)
Target (R)
Seeding finds small, local matching base-pairs

1B seeds
Seed
10B seed hits
Filter

Diagram:
- Seed
- Seed hits
- Query (Q)
- Target (R)

Seed hit
Filtering aligns ~100bp around seed hits
Filtering aligns ~100bp around seed hits
High-scoring Segment Pair reduced to Anchor

1B seeds

Seed

10B seed hits

Filter

1M anchors

Extend

Query (Q)

Target (R)

Anchor
Extension results in the final alignments

Dynamic Programming Equations

\[
I(i, j) = \max \{H(i, j - 1) - o, I(i, j - 1) - e\}
\]

\[
D(i, j) = \max \{H(i - 1, j) - o, D(i - 1, j) - e\}
\]

\[
H(i, j) = \max \begin{cases} 
0 \\
I(i, j) \\
D(i, j) \\
H(i - 1, j - 1) + W(r_i, q_j) 
\end{cases}
\]
Extension results in the final alignments

<table>
<thead>
<tr>
<th></th>
<th>human</th>
<th>mouse</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>AGGTAGCAAGGGGAACAGGAG</td>
<td>GGGGCC</td>
</tr>
<tr>
<td>1</td>
<td>AGGCAAGGAAGGGGACAGGAACACAGGCTGCAAGGCTGGA</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>AGGAAGGAGACAGGAG</td>
<td>- TGGCCAGAAGTGGCCAGGA</td>
</tr>
<tr>
<td>36</td>
<td>AGGAAGGGGCAAGGAACACAGGCTGCAAGGGTT</td>
<td>- AGGA</td>
</tr>
<tr>
<td>60</td>
<td>GGGGGCGAGG</td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>GGGGGGAGGG</td>
<td></td>
</tr>
</tbody>
</table>

Alignment

Query (Q)

Target (R)

Alignment

Anchor
Filtering stage dominates the runtime

Runtime distribution per stage (LASTZ)

- Seed: 1B seeds, 10B seed hits, 1M anchors
- Filter: 10k alignments
- Extend: 0.05% (1B seeds), 97.95% (10B seed hits), 2% (1M anchors)
Enabling biologists to leverage sequencing

**Comparative Genomics**

*SegAlign*: A Scalable GPU-based whole genome aligner

*Darwin-WGA*: A fast and highly sensitive co-processor for whole genome alignments

---

**Clinical Genomics**

An ultra-rapid workflow for clinical whole genome sequencing

*New England Journal of Medicine*

*Nature Biotechnology*
SegAlign system for single chromosome pair

1. Read & construct seed tables
2. Interval work queue
3. Generate Seeds
4. Seed chunks work queue

- Divide query into intervals and adds to the queue
- Each available thread takes the next interval
- Seeds chunks added to the queue
- Each available GPU takes the next chunk

CPU - GPU
Naïve approach allocates 1 seed hit per thread

• Considerably varying seed hit positions -> inefficient uncoalesced memory accesses within a warp
  • Warp - basic unit for scheduling execution and memory accesses

2. Divergent branches within a warp due to the dynamic X-drop condition for each thread
SegAlign allocates 1 seed hit per thread warp
1 seed hit/warp results in high GPU DRAM bandwidth efficiency

• Efficient bandwidth gains with coalesced memory accesses

• Exploiting data locality within each partition
Double buffering improves load balancing

Chromosome Pairs one at a time on single GPU

Over multiple GPUs

Optimized over multiple GPUs

Target Chromosome
Query Chromosome
GPU Seed & Filter

Runtime reduction

1 1 2 3 2

1 1 2 3 2

1 2 3 2
13.5x-14x speedup across different species pairs at ~2x cost improvement

<table>
<thead>
<tr>
<th></th>
<th>HW config</th>
<th>Cost per hour</th>
</tr>
</thead>
<tbody>
<tr>
<td>LASTZ</td>
<td>96 vCPUs</td>
<td>$4.08</td>
</tr>
<tr>
<td>SegAlign</td>
<td>8 V100 GPU 64 vCPUs</td>
<td>$24.48</td>
</tr>
</tbody>
</table>
Multi-node version: Seed-and-Filter phase

Target Chromosome
Query Chromosome

All chromosome pairs

Node 1
Node 2
Node 3

Segment File

Main Node

CPU
GPU
CPU
CPU
GPU
CPU
CPU
CPU

All chromosome pairs

Query Chromosome

Target Chromosome

Main Node

CPU
GPU
CPU
Multi-node version: Extension phase
SegAlign’s Ungapped extension kernel now in NVIDIA GenomeWorks library

https://github.com/clara-parabricks/GenomeWorks

GenomeWorks

Overview

GenomeWorks is a GPU-accelerated library for biological sequence analysis. This section provides a brief overview of the different components of GenomeWorks. For more detailed API documentation please refer to the documentation.
SegAlign for 1000+ way vertebrate alignment

SegAlign-integrated Cactus multiple genome aligner will be used to generate the pairwise alignments for the 1000+ vertebrate multiple alignment at UCSC, and reduce the compute time from months to days.

Cactus

Cactus is a reference-free whole-genome multiple alignment program. The principal algorithms are described here: https://doi.org/10.1101/gr.123356.111

⚠️ v1.1.0

Notable Changes:

- `cactus-prepare` improvements including:
  - WDL / Terra support
  - GPU lastz support

Armstrong et. al. (Nature 2020)
Improved search heuristics find 20 000 new alignments between human and mouse genomes

Martin C. Frith and Laurent Noé

_Nucleic Acids Research_, Volume 42, Issue 7, 1 April 2014, Page e59,
https://doi.org/10.1093/nar/gku104

Published: 31 January 2014  Article history

Increased alignment sensitivity improves the usage of genome alignments for comparative gene annotation

Virag Sharma, Michael Hiller

_Nucleic Acids Research_, Volume 45, Issue 14, 21 August 2017, Pages 8369–8377,
https://doi.org/10.1093/nar/gkx554

Published: 21 June 2017  Article history
Increasing indel frequency => increasing need for gapped filtering

-88 Mil yrs
-6.4 Mil yrs

1 gap (indel) per 625 base pairs
1 gap (indel) per 30 base pairs

LASTZ’s filter threshold
Replacing ungapped filtering by gapped filtering slows down the software by 200x!
Enabling biologists to leverage sequencing

**Comparative Genomics**

SegAlign: A Scalable GPU-based whole genome aligner
[Supercomputing Conference]

Darwin-WGA: A fast and highly sensitive co-processor for whole genome alignments
[HPCA]

**Clinical Genomics**

An ultra-rapid workflow for clinical whole genome sequencing
[New England Journal of Medicine Nature Biotechnology]
Specialized Operations

On 14nm CPU
35 ALU ops, 15 load/store
37 cycles
81nJ

On 40nm Special Unit
1 cycle (37x speedup)
3.1pJ (26,000x efficiency)
300fJ for logic (remainder is memory)

\[
\begin{align*}
I(i, j) &= \max \{H(i, j - 1) - o, I(i, j - 1) - e\} \\
D(i, j) &= \max \{H(i - 1, j) - o, D(i - 1, j) - e\} \\
H(i, j) &= \max \begin{cases} 
0 & \\
I(i, j) & \\
D(i, j) & \\
H(i - 1, j - 1) + W(r_i, q_j) & 
\end{cases}
\end{align*}
\]
Exploiting inner-loop parallelism with systolic arrays

- Initial values
- Control logic for termination
- Dual Port BRAM
- Start / Stop

Query (q) vs Target (r)

- Stripe 1
- Stripe 2
- Stripe 3
- Stripe 4
Can we adapt this architecture for extension?

Diagram showing:
- Dual Port BRAM
- Control logic for termination
- PE 0, PE 1, PE 2, PE 3
- Initial values
- Start / Stop
- Query (Q)
- Target (R)
- Alignment
- Anchor
Utilizing local memory – size is prohibitive for larger compute 😞

![Diagram showing memory hierarchy with Local SRAM (KB), On-chip SRAM (MB), and DRAM (GB) with respective energy per word costs: 5pJ/word, 50pJ/word, and 640pJ/word.]
Overlapped extension uses constant on-chip memory

- Tiled (tile size T, overlap O) implementation inspired by GACT in Darwin

- Origin of the next tile lies at the intersection of the current traceback path with the overlap
Overlapped extension uses constant on-chip memory

- Extension along a direction continues until a tile is encountered with a non-positive maximum score.
The divergence condition within each tile
=> small memory requirement

• Y-drop implementation within each tile
• Adaptive band with traceback
• Reduces on-chip memory requirement compared to computing whole tile
• Reduces compute time
Workloads in LASTZ v/s Darwin-WGA

LASTZ

Seeding

13B seed hits

Ungapped filtering

300k anchors

Extend

150k alignments

Darwin-WGA

Seeding (D-SOFT)

13B seed hits

Gapped filtering

1.2M anchors

Extend

700k alignments
Darwin-WGA finds genes that LASTZ does not

Indels (shown by arrows) around each seed hit – dropped by ungapped filtering (LASTZ) but retained by gapped filtering (Darwin-WGA)
**Darwin-WGA is more sensitive than LASTZ**

<table>
<thead>
<tr>
<th>Top-10 Alignment Chain Scores</th>
<th>Matching Base-pairs within Alignments</th>
<th>Number of Aligning Exons (protein-coding genes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upto +5.73%</td>
<td>Upto 3.12x</td>
<td>Upto 2.70%</td>
</tr>
</tbody>
</table>

- Represents **orthologs** sequences (indicates speciation)
- Represents **paralogs** sequences (indicative of duplication)
- Represents **functionally relevant** sequences, under some selective pressure at least in the target species

False positive rate (2-mer shuffled genome): 0.0007%
Darwin-WGA (FPGA) is 20x faster than iso-sensitive software

<table>
<thead>
<tr>
<th>Darwin-WGA (FPGA)</th>
<th>LASTZ</th>
<th>Iso-sensitivity Software</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.1x (slowdown)</td>
<td>20x (speed and cost)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HW config</th>
<th>Cost per hour</th>
</tr>
</thead>
<tbody>
<tr>
<td>LASTZ 36 vCPUs</td>
<td>$1.59</td>
</tr>
<tr>
<td>Darwin-WGA (FPGA) 1 Xilinx Virtex Ultrascale+ FPGA 8vCPUs</td>
<td>$1.65</td>
</tr>
</tbody>
</table>

Diagram showing the process:

1. Disk
2. CPU
   - Seeding (D-SOFT)
   - Seed hits
3. FPGA / ASIC
   - Filtering (Banded SW)
   - Anchor positions
4. Extension (GACT-X)
   - Alignments
5. Disk
### ASIC

TSMC 40nm DC synthesis (not a chip prototype)

<table>
<thead>
<tr>
<th></th>
<th>Configuration</th>
<th>Area (mm$^2$)</th>
<th>Power (W)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Filtering</strong></td>
<td>Logic</td>
<td>16.6</td>
<td>25.6</td>
</tr>
<tr>
<td><strong>Extension</strong></td>
<td>Logic</td>
<td>4.2</td>
<td>6.72</td>
</tr>
<tr>
<td><strong>Traceback</strong></td>
<td>12 x (64PE x 16KB/PE)</td>
<td>15.1</td>
<td>7.92</td>
</tr>
<tr>
<td><strong>DRAM</strong></td>
<td>DDR4-2400R</td>
<td>3.10</td>
<td></td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td></td>
<td>35.9</td>
<td>43.34</td>
</tr>
</tbody>
</table>

~80% logic area, ~60% chip power
## Darwin-WGA is 2 orders of magnitude faster than iso-sensitive software

<table>
<thead>
<tr>
<th></th>
<th>LASTZ</th>
<th>Iso-sensitive software</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Darwin-WGA (FPGA)</em></td>
<td>0.1x (slowdown)</td>
<td>20x (speed and cost)</td>
</tr>
<tr>
<td><em>Darwin-WGA (ASIC)</em></td>
<td>1.5x</td>
<td>300x (1500x perf/Watt)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>HW config</th>
<th>Cost per hour</th>
</tr>
</thead>
<tbody>
<tr>
<td>LASTZ</td>
<td>36 vCPUs</td>
<td>$1.59</td>
</tr>
<tr>
<td>Darwin-WGA (FPGA)</td>
<td>1 Xilinx Virtex Ultrascale+ FPGA 8vCPUs</td>
<td>$1.65</td>
</tr>
<tr>
<td>Darwin-WGA (ASIC)</td>
<td>36 mm², 43 Watt, 40nm TSMC</td>
<td></td>
</tr>
</tbody>
</table>
Enabling biologists to leverage sequencing

**Comparative Genomics**

**SegAlign**: A Scalable GPU-based whole genome aligner
[Supercomputing Conference]

**Darwin-WGA**: A fast and highly sensitive co-processor for whole genome alignments
[HPCA]

**Clinical Genomics**

An ultra-rapid workflow for clinical whole genome sequencing
[New England Journal of Medicine
Nature Biotechnology]
Enabling biologists to leverage sequencing

**Comparative Genomics**

SegAlign: A Scalable GPU-based whole genome aligner
[Supercomputing Conference]

Darwin-WGA: A fast and highly sensitive co-processor for whole genome alignments
[HPCA]

**Clinical Genomics**

An ultra-rapid workflow for clinical whole genome sequencing
[New England Journal of Medicine Nature Biotechnology]
• 13 years
• Two week history of dry cough, decreased appetite, chest pain and severe fatigue
• On admission to hospital in Oregon: weak heart
• Rapid deterioration
• Surgical implantation of Heartmate III (portable heart lung machine)
• Possible causes included potentially reversible (myocarditis) and irreversible (genetic heart disease)
• How to make transplant decision?
  • Biopsy?
  • Genetic testing?
State-of-the-art turn-around times

- Standard of Care
- Baylor Medicine (Rapid)
- Rady Children’s (Rapid)
- Stanford

Genetic Diagnosis Turn-around Time

- World Record - 14.5 hours
DNA Sequencing

Base calling

Alignment

Variant Discovery

Chr10 18,563 T -> A
Existing fastest turn-around in 14.5 hours

Sample Processing: 1.5 hr

Sequencing: 11 hr

Compute: 1 hr

Curation: 1 hr

Illumina HiSeqX

DRAGEN (FPGA)

Owen et al (2020)
Technology, new and old

Short reads

Long reads

https://vertebrategenomesproject.org/technology
Compute dominates the new pipeline with 30-hour turnaround time

- Longer reads => more complicated computation
- Higher error rate => double the amount of sequencing
- No custom ASIC/FPGA

Sample Processing 4 hr

Sample Diagnosis

Sequencing 1.5 hr

Compute 21 hr

Curation 2.5 hr

Nanopore Promethlon
Traditional computational pipeline

Sequencing Unit (PromethION 48) → Compute Tower (4x Tesla V100)

HG002
218 Gb
2.3 TB fast5

Sequencing: 1.5 hr
Base calling: 17.5 hr
Alignment: 2.5 hr
Nanopore’s “real-time” advantage

• Signal files are generated as soon as the strand passes through the nanopore
• Ideally, we can start base calling right away
Modified pipeline – overlap base calling and sequencing

Sequencing Unit (PromethION 48) – overlap base calling and sequencing

Sequencing

Compute Tower (4x Tesla V100)

HG002
218 Gb
2.3 TB fast5

1.5 hr
Sequencing

16 hr
Base calling

2.5 hr
Alignment

Sequencing Unit (PromethION 48)

Base calling (Guppy)

Alignment (minimap2)

Sequencing Unit (PromethION 48)

Compute Tower (4x Tesla V100)
Challenge 1: transfer TB data to cloud

- 2.3 TB of data in 1.5 hour = 3.4 Gbps
- Utilizing available bandwidth
Moving to cloud

**Challenge 1: transfer TB data to cloud**

1. VBZ compression for raw signal file – **30% less file size**
2. Optimize file size for
   (a) number of parallel uploads
   (b) latency overhead for each new file
Moving to cloud

Near real time I/O

Challenge 2: Optimized distributed system

- Support streaming dataflow
- Minimize orchestration/inter-node communication
- Make sure all resources are fully utilized based on rate of data generation
Moving to cloud

Near real time I/O

Challenge 2: Optimized distributed system

1. Analysis for specific set of flow cells assigned to each instance
2. Stateless pull architecture
3. Pipelining different compute stages

PromethiON sequencing (48 flow-cells)

Base calling and alignment

GPU

Basecalling (Guppy)

CPU

Alignment (minimap2)

GPUs/instance: 4x Tesla V100
CPUs/instance: 48
Total instances: 16
Near real time pipeline

HG002
218 Gb
2.3 TB fast5

Sequencing
1.5 hr

Base calling and Alignment
30 ± 10 min

GPUs/instance: 4x Tesla V100
CPUs/instance: 48
Total instances: 16

PromethION sequencing
(48 flow-cells)

Base calling and alignment

Signal files

Cloud storage

Signal files

Alignment files

Basecalling (Guppy)

Alignment (minimap2)
Variant Calling

HG00
218 Gb
2.3 TB fast5

Sequencing
Base calling and Alignment
Variant Calling

1.5 hr

30 ± 10 min
45 min
Compute-optimized Workflow

Sample Processing: 4 hr
Sequencing: 2 hr
Compute: 1.5 hr
Curation: 2.5 hr
Diagnosis
Co-design across stages

Sample Processing

Sequencing

Compute

Curation

Sample

Diagnosis

Negligible impact of sample identification step

Goenka et al. (Nature Biotechnology 2021)
Diagnosis made in 11.3 Hrs

Genetic Dilated Cardiomyopathy

The patient was urgently listed for transplant and received a new heart 21 days later.
Ultra-rapid pipeline with 8.5 hour turnaround time

Sample Processing: 3 hr
Sequencing: 2 hr
Compute: 1.5 hr
Curation: 1-2 hrs

Sample
DNA Extraction
DNA Fragmentation
Library Preparation
Sequence
Base Calling, Alignment
Variant Calling
Curation
Idle time
ID Positive finding

Patient ID
01
02
03
04
05
06
07
08
09
10
11
12

Runtime (hr)
0
4
8
12
16
20
24

Diagram showing the process flow and time breakdown of the pipeline for different patient IDs.
Diagnosis in 8.5 hours.. Or less

- Standard of Care
- Baylor Medicine (Rapid)
- Rady Children’s (Rapid)
- Stanford (Ultra-rapid)

- Old World Record - 14.5 hours
- World Record - 7.3 hours
This is just the beginning...

**Comparative Genomics**

**SegAlign**: A Scalable GPU-based whole genome aligner  
*Supercomputing Conference*

**Darwin-WGA**: A fast and highly sensitive co-processor for whole genome alignments  
*HPCA*

---

**Clinical Genomics**

An ultra-rapid workflow for clinical whole genome sequencing  
*New England Journal of Medicine*  
*Nature Biotechnology*