Brief Self Introduction

- **Can Firtina**
  - Ph.D. Student in [SAFARI Research Group](https://safari.ethz.ch/) led by [Prof. Onur Mutlu](https://safari.ethz.ch/)

- **Research interests:** Bioinformatics & Computer Architecture
  - Real-time genome analysis
  - Similarity search in a large space of genomic data
  - Hardware-Algorithm co-design to accelerate genome analysis
  - Genome editing
  - Error correction

- Get to know **our group and our research**
  - **Group website:** [https://safari.ethz.ch/](https://safari.ethz.ch/)
  - **Contact me:** canfirtina@gmail.com
  - **Website:** [https://cfirtina.com](https://cfirtina.com)
  - **Twitter (aka X):** [https://twitter.com/FirtinaC](https://twitter.com/FirtinaC)
Agenda for Today

- Introduction to Genomics
  - Today: The ways we analyze genomes
  - Intelligent genome analysis

- Step-by-step Genome Analysis
  - Sequencing
  - Read Mapping
  - Variant Calling

- Algorithmic & Hardware Acceleration

- Future Opportunities: New Technologies & Applications
The Goal of Computing: Beyond Numbers

“The purpose of computing is [to gain] insight, not numbers”

Richard Hamming

We need to gain insights and observations much more efficiently than ever before.
Big Data is Everywhere

Astronomy
25 zetta-bytes/year

Twitter (now X)
0.5-15 billion tweets/year

YouTube
500-900 million hours/year

Genomics
1 zetta-bases/year

Angstrom (10^{-10}m) Era of Semiconductors

Intel Process Technology Innovations

*Graphic is for illustrative purposes only and is not to scale

https://siliconangle.com/2021/07/26/...
Problems with Data Analysis Today

**Special-Purpose Machine** for Data Generation

**General-Purpose Machine** for Data Analysis

**FAST**

**SLOW**

Slow and inefficient processing capability
Large amounts of data movement
Intelligent Data Analysis

- The **science and art** of revealing previously unknown and potentially **valuable information** or knowledge from **data**

- While **meeting certain goals**:
  - Performance & latency
  - Energy consumption
  - Cost...

- Tailored for many **important applications**
  - AI/ML, Genomics, Medicine, Health...
Recall: Four Key Current Directions

- Fundamentally Secure/Reliable/Safe Architectures

- Fundamentally Energy-Efficient Architectures
  - Memory-centric (Data-centric) Architectures

- Fundamentally Low-Latency and Predictable Architectures

- Architectures for AI/ML, Genomics, Medicine, Health
Faster, Scalable & Accurate Genome Analysis

Understanding **genetic variations**, species, evolution, ...

Predicting the **presence** and **relative abundance** of **microbes** in a sample

Rapid surveillance of **disease outbreaks**

Developing **personalized medicine**

And, many, many other applications ...
What is a Genome?

https://onlinelearning.hms.harvard.edu/hmx/courses/genetic-testing/
https://www.genome.gov/genetics-glossary/
What is a Genome?

The entire set of DNA sequences in a cell
How Large is a (Human) Genome?

Andreasturm, Zurich

~3.2 billion genomic bases*

*~100 meters = 80 characters per line, 40 lines per A4 page, each page is 0.1mm thick

https://www.gigon-guyer.ch/de/project/andreasturm/
Cracking the 1st Human Genome Sequence

- **1990-2003:** The Human Genome Project (HGP) provides a complete and accurate sequence of all **DNA base pairs** that make up the human genome and finds 20,000 to 25,000 human genes.

3.2 x10⁹ bases
13 years
>3 x10⁹ $
Now: Complete human reference genome

Abstract
In 2001, Celera Genomics and the International Human Genome Sequencing Consortium published their initial drafts of the human genome, which revolutionized the field of genomics. While these drafts and the updates that followed effectively covered the euchromatic fraction of the genome, the heterochromatin and many other complex regions were left unfinished or erroneous. Addressing this remaining 8% of the genome, the Telomere to Telomere (T2T) Consortium has finished the first truly complete 3,055 billion base pair (bp) sequence of a human genome, representing the largest improvement to the human reference genome since its initial release. The new T2T-CHM13 reference includes gapless assemblies for all 22 autosomes plus Chromosome X, corrects numerous errors, and introduces nearly 200 million bp of novel sequence containing 2,226 paralogous gene copies, 115 of which are predicted to be protein coding. The newly completed regions include all centromeric satellite arrays and the short arms of all five acrocentric chromosomes, unlocking these complex regions of the genome to variational and functional studies for the first time.

https://genomics.ucsc.edu/2021/06/09/the-complete-sequence-of-a-human-genome/
https://time.com/collection/100-most-influential-people-2022/
How About Other Species?

Phi X174 virus  
5.386 Kilo bp

E. coli O157:H7  
5.44 Million bp

Homo Sapiens  
3.2 Billion bp

Onion, Allium Cepa  
16 Billion bp

Paris Japonica  
149 Billion bp
Human Chromosomes (23 Pairs)

Autosomes

Parent #1

Parent #2

Sex chromosomes

XX or XY
Human Chromosomes (23 Pairs)

Autosomes

Parent #1

Parent #2

= Adenine

= Thymine

= Cytosine

= Guanine

= Phosphate backbone

Sex chromosomes

Female: XX

Male: XY
DNA Under Electron Microscope

human chromosome #12 from HeLa’s cell
The Central Dogma of Molecular Biology

- **DNA Genotypes**
- **RNA**
- **Protein Phenotypes**

**Transcription**

**Replication**

**Translation**

- ATGC
- ACC
- TACGT

- UACG
- UUGG

- SAFARI
Cells of Different Organs and Tissues

- Most **cells** in a person's body have almost the **same DNA** and the **same genes**
  - Expression of the genes **differs** between cells
  - But **not all genes** are used or expressed by those cells
  - Mutations may occur over time

![Diagram showing expression of genes in different organs](image)

NIH 2009 National DNA Day

20,000-25,000 human genes
# Finding SNPs Associated with Complex Trait

| Individual #1 | ...ACATGCGACATTTCATAAGGCC... | 180 |
| Individual #2 | ...ACATGCGACATTTCATAAGGCC... | 175 |
| Individual #3 | ...ACATGCGACATTTCATAAGGCC... | 170 |
| Individual #4 | ...ACATGCGACATTTCATAAGGCC... | 165 |
| Individual #5 | ...ACATGCGACATTTCATAAGGCC... | 160 |
| Individual #6 | ...ACATGCGACATTTCATAAGGCC... | 145 |
| Individual #7 | ...ACATGCGACATTTCATAAGGCC... | 140 |
| Individual #8 | ...ACATGCGACATTTCATAAGGCC... | 130 |
| Individual #9 | ...ACATGTTCGACATTTCATAAGGCC... | 120 |
| Individual #10| ...ACATGTTCGACATTTCATAAGGCC... | 120 |
| Individual #11| ...ACATGTTCGACATTTCATAAGGCC... | 115 |
| Individual #12| ...ACATGTTCGACATTTCATAAGGCC... | 110 |
| Individual #13| ...ACATGTTCGACATTTCATAAGGCC... | 110 |
| Individual #14| ...ACATGTTCGACATTTCATAAGGCC... | 110 |
| Individual #15| ...ACATGTTCGACATTTCATAAGGCC... | 105 |
| Individual #16| ...ACATGTTCGACATTTCATAAGGCC... | 100 |

**SNP:** single nucleotide polymorphism
Genome-Wide Association Study (GWAS)

- Detecting genetic variants associated with phenotypes using two groups of people.

Manhattan plot

variant with higher frequency in cases than controls
Opportunities and challenges for transcriptome-wide association studies

Michael Wainberg¹, Nasa Sinnott-Armstrong ², Nicholas Mancuso ³, Alvaro N. Barbeira ⁴, David A. Knowles ⁵,⁶, David Golan², Raili Ermel⁷, Arno Ruusalepp⁷,⁸, Thomas Quertermous ⁹, Ke Hao ¹⁰, Johan L. M. Björkegren ⁸,¹⁰,¹¹,¹², Hae Kyung Im ¹⁴*, Bogdan Pasaniuc ³,¹³,¹⁴*, Manuel A. Rivas ¹⁵* and Anshul Kundaje ¹,²*

Transcriptome-wide association studies (TWAS) integrate genome-wide association studies (GWAS) and gene expression datasets to identify gene-trait associations. In this Perspective, we explore properties of TWAS as a potential approach to prioritize causal genes at GWAS loci, by using simulations and case studies of literature-curated candidate causal genes for schizophrenia, low-density-lipoprotein cholesterol and Crohn's disease. We explore risk loci where TWAS accurately prioritizes the likely causal gene as well as loci where TWAS prioritizes multiple genes, some likely to be non-causal, owing to sharing of expression quantitative trait loci (eQTL). TWAS is especially prone to spurious prioritization with expression data from non-trait-related tissues or cell types, owing to substantial cross-cell-type variation in expression levels and eQTL strengths. Nonetheless, TWAS prioritizes candidate causal genes more accurately than simple baselines. We suggest best practices for causal-gene prioritization with TWAS and discuss future opportunities for improvement. Our results showcase the strengths and limitations of using eQTL datasets to determine causal genes at GWAS loci.

SNPs and Personalized Medicine

SNP rs12979860

<table>
<thead>
<tr>
<th>Name</th>
<th>rs12979860</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chromosome</td>
<td>19</td>
</tr>
<tr>
<td>Position</td>
<td>39248147</td>
</tr>
<tr>
<td>Weight of evidence</td>
<td>926</td>
</tr>
</tbody>
</table>

Allele Frequency

- A: 49%
- T: 27%
- G: 23%
- C:
- -:
- 0:

Links to SNPpedia

<table>
<thead>
<tr>
<th>Title</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>rs12979860 T/T</td>
<td>~20-25% of such hepatitis c patients respond to treatment</td>
</tr>
<tr>
<td>rs12979860 C/C</td>
<td>~80% of such hepatitis c patients respond to treatment</td>
</tr>
<tr>
<td>rs12979860 C/T</td>
<td>~20-40% of such hepatitis c patients respond to treatment</td>
</tr>
</tbody>
</table>

https://opensnp.org/snps/rs12979860
Much Larger Structural Variations!

**AUTISM**
Deletion of 593 kb

**OBESITY**
Walters, *Nature* 2010
Deletion of 593 kb

**SCHIZOPHRENIA**
McCarthy, *Nat Genet* 2009
Duplication of 593 kb

**UNDERWEIGHT**
Duplication of 593 kb

Deletion in the short arm of chromosome 16 (16p11.2)

Duplication in the short arm of chromosome 16 (16p11.2)

CNV: copy number variation
Recommended Reading

nature reviews genetics

Explore our content  Journal information

nature › nature reviews genetics › review articles › article

Review Article  Published: 15 November 2019

Structural variation in the sequencing era

Steve S. Ho, Alexander E. Urban & Ryan E. Mills

Nature Reviews Genetics 21, 171–189 (2020)  Cite this article

15k Accesses  16 Citations  309 Altmetric  Metrics

Ho+, "Structural variation in the sequencing era", Nature Reviews Genetics, 2020
Does intelligent genome analysis really matter?
Intelligent Genome Analysis

- Fast genome analysis
  - Real-time analysis

- Large scale
  - Analyze the entire population

- Accurate analysis
  - Incorrect diagnosis of disease

- Using intelligent architectures
  - Specialized hardware with less data movement

- DNA is a valuable asset
  - Controlled-access analysis

Latency & Throughput
Scalability
Precision & Accuracy
Energy-efficiency & Bandwidth
Privacy
Intelligent Genome Analysis

Mohammed Alser, Joel Lindegger, Can Firtina, Nour Almadhoun, Haiyu Mao, Gagandeep Singh, Juan Gomez-Luna, Onur Mutlu

“From Molecules to Genomic Variations: Intelligent Algorithms and Architectures for Intelligent Genome Analysis”

Computational and Structural Biotechnology Journal, 2022

[source code]

Review

From molecules to genomic variations: Accelerating genome analysis via intelligent algorithms and architectures

Mohammed Alser*, Joel Lindegger, Can Firtina, Nour Almadhoun, Haiyu Mao, Gagandeep Singh, Juan Gomez-Luna, Onur Mutlu*

ETH Zurich, Gloriastrasse 35, 8092 Zürich, Switzerland
Fast genome analysis in mere seconds using limited computational resources (e.g., personal computer or a mobile device).
Farnaes+, “Rapid whole-genome sequencing decreases infant morbidity and cost of hospitalization”, NPJ Genomic Medicine, 2018
“From 2019, all seriously ill children in UK will be offered whole genome sequencing as part of their care”
Rapid Surveillance of Disease Outbreaks

Real-time, portable genome sequencing for Ebola surveillance

Figure 1: Deployment of the portable genome surveillance system in Guinea.
COVID-19 Outbreak and PCR Testing

- Outbreaks
  - Urgent need for testing

- PCR Testing
  - Reliable results for **known regions to target**
  - High latency to get the right answer
  - Hard to customize to any region (e.g., COVID-19 variants)
Massively scaled-up testing for SARS-CoV-2 RNA via next-generation sequencing of pooled and barcoded nasal and saliva samples

Joshua S. Bloom, Laila Sathe, […] Valerie A. Arboleda

Nature Biomedical Engineering 5, 657–665 (2021) | Cite this article

4675 Accesses | 110 Altmetric | Metrics

Bloom+, "Swab-Seq: A high-throughput platform for massively scaled up SARS-CoV-2 testing", Nature Biomedical Engineering, 2021
Large Scale Analysis

https://blog.wego.com/7-crowded-places-and-events-that-you-will-love/
Population-Scale Microbiome Profiling

Goal: What organisms are present in a given environment and how abundant are they?

[Image] https://blog.wego.com/7-crowded-places-and-events-that-you-will-love/
Characterizing genomic variations of 49,962 Icelanders took **4.15 million CPU hours** or 83 CPU hours per sample on average.

“GraphTyper2 enables population-scale genotyping of structural variation using pangenome graphs”, Nature Communications, 2019
City-Scale Microbiome Profiling

Afshinnekoo+, "Geospatial Resolution of Human and Bacterial Diversity with City-Scale Metagenomics", Cell Systems, 2015
Accurate genome analysis to make life-critical decisions and improving the quality of life
Plague (Yersinia Pestis)

What Is It?

Published: December, 2018

Plague is caused by Yersinia pestis bacteria. It can be a life-threatening infection if not treated promptly. Plague has caused several major epidemics in Europe and Asia over the last 2,000 years. Plague has most famously been called "the Black Death" because it can cause skin sores that form black scabs. A plague epidemic in the 14th century killed more than one-third of the population of Europe within a few years. In some cities, up to 75% of the population died within days, with fever and swollen skin sores.
Plague in New York Subway System?

Plague (Yersinia)

What Is It?

Published: December, 2018

Plague is caused by Yersinia treated promptly. Plague has lasted 2,000 years. Plague has cause skin sores that form b than one-third of the popul the population died within

The findings of Yersinia Pestis in the subway received wide coverage in the lay press, causing some alarm among New York residents

Failure of Bioinformatics

Rob Knight, a professor in the department of pediatrics at the University of California, San Diego, calls this type of error “a failure of bioinformatics,” in that Mason had assumed the gene fragments were unique to the pathogens, when in fact they can also be detected in other...

Living in a microbial world

Charles Schmidt


[https://www.nature.com/articles/nbt.3868](https://www.nature.com/articles/nbt.3868)
CAMI Consortium

F. Meyer, A. Fritz, Z.L. Deng, D. Koslicki, A. Gurevich, G. Robertson, Mohammed Alser, and others

“Critical Assessment of Metagenome Interpretation - the second round of challenges”, Nature Methods, 2022
[Source Code]
Metalign

Nathan LaPierre, Mohammed Alser, Eleazar Eskin, David Koslicki, Serghei Mangul

“Metalign: efficient alignment-based metagenomic profiling via containment min hash”

Genome Biology, September 2020.

[Talk Video (7 minutes) at ISMB 2020]

[Source code]
Using Intelligent Architectures & Reliability

Challenging Environment in Outer Space

Challenging Environment in Outer Space

DNA sequencing at the picogram level to investigate life on Mars and Earth

Jyothi Basapathi Raghavendra, Maria-Paz Zorzano, Deepak Kumaresan & Javier Martin-Torres

Scientific Reports 13, Article number: 15277 (2023)

Abstract

DNA is an incontrovertible biosignature whose sequencing aids in species identification, genome functionality, and evolutionary relationships. To study life within the rocks of Earth and Mars, we demonstrate, in an ISO5 clean room, a procedure based on nanopore technology that correctly identifies organisms at picogram levels of DNA without amplification. Our study with E. coli and S. cerevisiae DNA samples showed that MinION sequencer (Oxford Nanopore Technologies) can unequivocally detect and characterise microbes with as little as 2 pg of input with just 50 active nanopores. This result is an excellent advancement in sensitivity, immediately applicable to investigating low biomass samples. This value is also at the level of possible background contamination associated with the reagents and the environment. Cultivation of natural and heat-treated Martian analogue (MMS-2) regolith samples, exposed to atmospheric water vapour or in increasing water concentrations, led to the extraction of 600–1000 pg of DNA from 500 mg of soil. Applying the low detectability technology enabled through MinION sequencer for a natural low biomass setting, we characterised the dry MMS-2 and found few soil-related organisms and airborne contaminants. The picogram detection level and the procedure presented here, may be of interest for the future Mars sample Return program, and the life research and planetary protection studies that will be implemented through the sample safety assessment.
Intelligent Architecture?

Modern systems

- FPGAs
- Heterogeneous Processors and Accelerators
- Hybrid Main Memory
- (General Purpose) GPUs
- Sequencing Machine
- Persistent Memory/Storage
Intelligent Architecture?

Modern systems

FPGAs

(General Purpose) GPUs

Heterogeneous Processors and Accelerators

Hybrid Main Memory

Persistent Memory/Storage

Sequencing Machine

https://nanoporetech.com/products/smidgion
DNA is a Valuable Asset to Protect

Worried about the 23andMe hack? Here’s what you can do.

A bad actor offered to sell information on 23andMe’s users, calling out Jewish people specifically

By Tahum Hurtzer

Updated October 13, 2023 at 3:18 p.m. EDT | Published October 12, 2023 at 7:00 a.m. EDT

Fourteen million people have shared their genetic information with 23andMe in hopes of learning more about their heritage. After a hack that appeared to target people with Jewish ancestry, some might be wondering how to cut ties with the company.
Privacy-Preserving Genome Analysis

Fig. 5. A completion attack.

Alser+, "Can you really anonymize the donors of genomic data in today’s digital world?" 10th International Workshop on Data Privacy Management (DPM), 2015.
Privacy-Preserving DNA Test

Our DNA Test, Reports, and Technology

- **Whole Genome Sequencing.** Decode 100% of your DNA with Whole Genome Sequencing and fully unlock your genetic blueprints.
- **Privacy First DNA Testing.** Begin your journey of discovery without risking the privacy of your most personal information.
- **Nebula Research Library.** Receive new reports every week that are based on the latest scientific discoveries.
- **Genome Exploration Tools.** Use powerful, browser-based genome exploration tools to answer any questions about your DNA.
- **Deep Genetic Ancestry.** Discover more about your ancestry with full Y chromosome and mitochondrial DNA sequencing and analysis.
- **Genomic Big Data Access.** Download your FASTQ, BAM, and VCF files and dive deeper into your Whole Genome Sequencing data.
- **Ready for Diagnostics.** Our Whole Genome Sequencing data is of the highest quality and can be used by physicians and genetic counselors.

SAFARI  https://nebula.org/whole-genome-sequencing/
Faster, Scalable & Accurate Genome Analysis

Understanding **genetic variations, species, evolution, ...**

Predicting the **presence and relative abundance of microbes** in a sample

Rapid surveillance of **disease outbreaks**

Developing **personalized medicine**

And, many, many other applications ...
Applications are only limited by our imagination.
DNA Computing

Massive parallelism to solve (hard) problems!
New Personalized Shopping Paradigm

https://www.dnanudge.com/
How and where to enable fast, accurate, cheap, privacy-preserving, and exabyte scale analysis of genomic data?
An embedded device that can perform comprehensive genome analysis in real time (within a minute)

- Which of these DNAs does this DNA segment match with?
- What is the likely genetic disposition of this patient to this drug?
- What disease/condition might this particular DNA/RNA piece associated with?
- . . .
Algorithm-Arch-Device Co-Design is Critical

Computer Architecture (expanded view)

- Problem
- Algorithm
- Program/Language
- System Software
- SW/HW Interface
- Micro-architecture
- Logic
- Devices
- Electrons
Accelerating Genome Analysis [DAC 2023]

Onur Mutlu and Can Firtina,
"Accelerating Genome Analysis via Algorithm-Architecture Co-Design"
[Slides (pptx) (pdf)]
[Talk Video (38 minutes, including Q&A)]
[Related Invited Paper]
[arXiv version]

Accelerating Genome Analysis via Algorithm-Architecture Co-Design

Onur Mutlu    Can Firtina
ETH Zürich

A Bright Future for Intelligent Genome Analysis

Mohammed Alser, Zülal Bingöl, Damla Senol Cali, Jeremie Kim, Saugata Ghose, Can Alkan, Onur Mutlu


Accelerating Genome Analysis: A Primer on an Ongoing Journey
DOI Bookmark: 10.1109/MM.2020.3013728

FPGA-Based Near-Memory Acceleration of Modern Data-Intensive Applications
DOI Bookmark: 10.1109/MM.2021.3088396
Agenda for Today

- Introduction to Genomics
  - Today: The ways we analyze genomes
  - Intelligent genome analysis

- Step-by-step Genome Analysis
  - Sequencing
  - Read Mapping
  - Variant Calling

- Algorithmic & Hardware Acceleration

- Future Opportunities: New Technologies & Applications
Genome Analysis in Real Life

Sample Collection

DNA Molecule

Chopped DNA Fragments

Library Preparation

Sequencing

Computational Steps

Raw Sequencing Data

Genomic Analyses
### Many Genome Analysis Tools

**Sequencing Technology:**
- Illumina
- ONT
- PacBio (HiFi)

**Species:**
- E. Coli
- Human
- Yeast
- Zebra Fish
- Mice
- Fruit Fly

**Coverage:**
- Low 2x - 30x
- Moderate 30x - 100x
- High >250x

**Read Length:**
- Short 100bp - 250bp
- Long 200bp – 2Mbp (>200bp)
- HiFi 10K-20Kbp

**Basecalling** → **Read Set** → **Read Correction** → **Reference Genomes** → **Sketching/ Indexing** → **Mapping** → **Assembly** → **Polishing** → **Variant Calling** → **Taxonomy Profiling**

**Read Mapper:**
- BWA-MEM2
- Minimap2
- NGM-LR
- Bowtie2

**Assembly:**
- Canu
- Miniasm (uses Minimap2)

**De novo Assembler (Long Reads):**
- Canu
- Miniasm (uses Minimap2)

**De novo Assembler (Short Reads):**
- ABYSS
- SPAdes (small genomes)

**Variant Caller:**
- LuMPY
- VariationHunter
- GATK
- TaRDiS
- Freebayes
- DELLY
- Platypus
- SAMtools
- Genome STRIP

**Assembly Polisher:**
- Apollo
- Racon
- Pilon
- Quiver (PB reads)
- Arrow (PB reads, Not published yet)
- NanoPolish (ONP reads)

**Variant Caller:**
- LuMPY
- VariationHunter
- GATK
- TaRDiS
- Freebayes
- DELLY
- Platypus
- SAMtools
- Genome STRIP

**Taxonomy Profiling:**
- Kraken2
- Metalign
- MiCoP
Alser+, “Going From Molecules to Genomic Variations to Scientific Discovery: Intelligent Algorithms and Architectures for Intelligent Genome Analysis”, CSBJ, 2022
Obtaining Sequencing Data

Alser+, “Going From Molecules to Genomic Variations to Scientific Discovery: Intelligent Algorithms and Architectures for Intelligent Genome Analysis”, CSBJ, 2022
Complete Genome in One Piece?

NO

machine gives the **complete** sequence of genome as output
DNA Sequencing

- **Goal:**
  - Find the complete sequence of A, C, G, T’s in an organism’s DNA

- **Challenge:**
  - There is no machine that takes long DNA as an input, and gives the complete sequence as output
  - All sequencing machines chop DNA into pieces and identify relatively small pieces (but not how they fit together)
DNA Sequencer is a Chopper

Large DNA molecule

Small DNA fragments

ACGTACCCCGT

GATACACTGTG

TTTTTTAATT

CTAGGGACCTT

ACGACGTAGCT

AAAAA

ACGAGCGGGT

GATACTG

AAAAAA

CTAGGACCTT

ACGACGTA

GCT

Reads
Genome Sequencer is a Chopper

Genome Analysis Pipeline

Genomic Sample ➔ Sequencing Machine ➔ Reads ➔ Genomic Variants

- **Read Mapping**
- **1x10^{12} bases**
- **44 hours**
- **<1000 $**

* NovaSeq 6000
Current sequencing machine provides small randomized fragments of the original DNA sequence.
Sequencing Technologies

- Oxford Nanopore Technologies (ONT)
- PacBio (HiFi, CLR)
- Illumina

... and more! All produce data with different properties.
## Illumina Sequencers

<table>
<thead>
<tr>
<th></th>
<th>iSeq 100</th>
<th>MiniSeq</th>
<th>MiSeq</th>
<th>NextSeq 550</th>
<th>NextSeq 2000</th>
<th>NovaSeq 6000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run time</td>
<td>9.5–19 hrs</td>
<td>4–24 hrs</td>
<td>4–55 hrs</td>
<td>12–30 hrs</td>
<td>24-48 hrs</td>
<td>13-44 hrs</td>
</tr>
<tr>
<td>Max. reads per run</td>
<td>4 million</td>
<td>25 million</td>
<td>25 million</td>
<td>400 million</td>
<td>1 billion</td>
<td>20 billion</td>
</tr>
<tr>
<td>Max. read length</td>
<td>2 × 150 bp</td>
<td>2 × 150 bp</td>
<td>2 × 300 bp</td>
<td>2 × 150 bp</td>
<td>2 × 150 bp</td>
<td>2 × 250</td>
</tr>
<tr>
<td>Max. output</td>
<td>1.2 Gb</td>
<td>7.5 Gb</td>
<td>15 Gb</td>
<td>120 Gb</td>
<td>300 Gb</td>
<td>6000 Gb</td>
</tr>
<tr>
<td>Estimated price</td>
<td>$19,900</td>
<td>$49,500</td>
<td>$128,000</td>
<td>$275,000</td>
<td>$335,000</td>
<td>$985,000</td>
</tr>
</tbody>
</table>

[SAFARI](https://www.illumina.com/systems/sequencing-platforms.html)
## Oxford Nanopore Sequencers

<table>
<thead>
<tr>
<th>Model</th>
<th>MinION Mk1B</th>
<th>MinION Mk1C</th>
<th>GridION Mk1</th>
<th>PromethION 24</th>
<th>PromethION 48</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Read length</strong></td>
<td>&gt; 2Mb</td>
<td>&gt; 2Mb</td>
<td>&gt; 2Mb</td>
<td>&gt; 2Mb</td>
<td>&gt; 2Mb</td>
</tr>
<tr>
<td><strong>Yield per flow cell</strong></td>
<td>50 Gb</td>
<td>50 Gb</td>
<td>50 Gb</td>
<td>220 Gb</td>
<td>220 Gb</td>
</tr>
<tr>
<td><strong>Number of flow cells per device</strong></td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>24</td>
<td>48</td>
</tr>
<tr>
<td><strong>Yield per device</strong></td>
<td>&lt;50 Gb</td>
<td>&lt;50 Gb</td>
<td>&lt;250 Gb</td>
<td>&lt;5.2 Tb</td>
<td>&lt;10.5 Tb</td>
</tr>
<tr>
<td><strong>Starting price</strong></td>
<td>$1,000</td>
<td>$4,990</td>
<td>$49,995</td>
<td>$195,455</td>
<td>$327,455</td>
</tr>
</tbody>
</table>

**SAFARI**  [https://nanoporetech.com/products/comparison](https://nanoporetech.com/products/comparison)
Nanopore sequencing technology and tools for genome assembly: computational analysis of the current state, bottlenecks and future directions


[Open arxiv.org version] [Slides (pptx) (pdf)] [Talk Video at AACBB 2019]
Different Raw Sequencing Data

**Illumina**
- Multiple images
- .BCL/.CBCL

**ONT**
- Squiggle
- .FAST5

**PacBio**
- 30-hour movie
- .BAM
Types of Genomic Reads

Wenger+, "Accurate circular consensus long-read sequencing improves variant detection and assembly of a human genome", Nature Biotechnology, 2019

https://labs.wsu.edu/genomicscore/illumina-sequencing/
https://pacbio.gs.washington.edu/
Nanopore Sequencing

Nanopore Sequencing: a widely used sequencing technology

- **Long** reads (up to >2 million genomic bases)
- **Portable** sequencing
- Cost-effective
- More unique features: **real-time analysis**
Nanopore Sequencing & Real-time Analysis

**Raw Signals:** Ionic current measurements generated as DNA passes through the nanopore at a certain speed

**(Real-Time) Analysis:** Translating to bases or directly analyzing raw signals

**Real-Time Decisions:** Stopping sequencing early based on real-time analysis
New Frontiers: Raw Signal Analysis

- Can Firtina, Nika Mansouri Ghiasi, Joel Lindegger, Gagandeep Singh, Meryem Banu Cavlak, Haiyu Mao, and Onur Mutlu,
  "RawHash: Enabling Fast and Accurate Real-Time Analysis of Raw Nanopore Signals for Large Genomes"

Proceedings of the 31st Annual Conference on Intelligent Systems for Molecular Biology (ISMB) and the 22nd European Conference on Computational Biology (ECCB), Jul 2023

[Bioinformatics Journal version]
[Slides (pptx) (pdf)]
[RawHash Source Code]

Bioinformatics, 2023, 39, i297–i307
https://doi.org/10.1093/bioinformatics/btad272
ISMB/ECCB 2023

RawHash: enabling fast and accurate real-time analysis of raw nanopore signals for large genomes

Can Firtina, Nika Mansouri Ghiasi, Joel Lindegger, Gagandeep Singh, Meryem Banu Cavlak, Haiyu Mao, Onur Mutlu

1Department of Information Technology and Electrical Engineering, ETH Zurich, 8092 Zurich, Switzerland
*Corresponding author. Department of Information Technology and Electrical Engineering, ETH Zurich, Gloriastrasse 35, 8092 Zurich, Switzerland.
E-mail: firtinac@ethz.ch (C.F.), omutlu@ethz.ch (O.M.)
Fast and Accurate Real-Time Genome Analysis

Can Firtina, Melina Soysal, Joel Lindegger, and Onur Mutlu,
"RawHash2: Accurate and Fast Mapping of Raw Nanopore Signals using a Hash-based Seeding Mechanism"
Preprint on arxiv, September 2023.
[arXiv version]
[RawHash2 Source Code]

RawHash2: Accurate and Fast Mapping of Raw Nanopore Signals using a Hash-based Seeding Mechanism

Can Firtina  Melina Soysal  Joel Lindegger  Onur Mutlu

ETH Zürich
Fast and Accurate Real-Time Genome Analysis

- Joel Lindegger, Can Firtina, Nika Mansouri Ghiasi, Mohammad Sadrosadati, Mohammed Alser, and Onur Mutlu,

"RawAlign: Accurate, Fast, and Scalable Raw Nanopore Signal Mapping via Combining Seeding and Alignment"

Preprint on arxiv, October 2023.
[arXiv version]
[RawAlign Source Code]

RawAlign: Accurate, Fast, and Scalable Raw Nanopore Signal Mapping via Combining Seeding and Alignment

Joël Lindegger§
Mohammad Sadrosadati§
Can Firtina§
Mohammed Alser§
Nika Mansouri Ghiasi§
Onur Mutlu§

§ETH Zürich
Accelerating Basecallers: Software Methods

- M. Banu Cavlak, Gagandeep Singh, Mohammed Alser, Can Firtina, Joel Lindegger, Mohammad Sadrosadati, Nika Mansouri Ghiasi, Can Alkan, and Onur Mutlu, "TargetCall: Eliminating the Wasted Computation in Basecalling via Pre-Basecalling Filtering". Proceedings of the 21st Asia Pacific Bioinformatics Conference (APBC), Changsha, China, April 2023.
  - [TargetCall Source Code]
  - [arxiv.org Version]
  - [Talk Video at BIO-Arch 2023 Workshop]

TargetCall: Eliminating the Wasted Computation in Basecalling via Pre-Basecalling Filtering

Meryem Banu Cavlak\textsuperscript{1} Gagandeep Singh\textsuperscript{1} Mohammed Alser\textsuperscript{1} Can Firtina\textsuperscript{1} Joël Lindegger\textsuperscript{1} Mohammad Sadrosadati\textsuperscript{1} Nika Mansouri Ghiasi\textsuperscript{1} Can Alkan\textsuperscript{2} Onur Mutlu\textsuperscript{1}

\textsuperscript{1}ETH Zürich \quad \textsuperscript{2}Bilkent University
Sequencing in Action

Chemistry type:
R10.4.1

Pack size:
Select ...

1 Flow cell  $900.00
$900.00 each

12 Flow cells $9,480.00
$790.00 each

MinION
Portable DNA/RNA sequencing for anyone

https://store.nanoporetech.com/flow-cell-r9-4-1.html
Machine Learning for Nanopore Machine

Wan+

“Beyond sequencing: machine learning algorithms extract biology hidden in Nanopore signal data”
*Trends in Genetics, October 25, 2021*

Review

Beyond sequencing: machine learning algorithms extract biology hidden in Nanopore signal data

Yuk Kei Wan,1,2 Christopher Hendra,3,1 Ploy N. Pratanwanich,1,4,5 and Jonathan Göke1,6,*
Sequencing using the Illumina Machines
Sequencing using the Illumina Machines

Glass flow cell surface

Optical Sensor

DNA fragment = Read

Billions of Short Reads

ATAATACGTAATGTAACGT
TTTAGTACGTACGT
ATAACGTATAGTACGT
ACGCCCCGTACGTA
ACGTAATAGTACGT
TTAGTACGTACGT
TTACGTACTAAGTACGT
TTACGTACTAAGTACGT
TTTTAATAACGTA
GGGAGTACGTACGT

AFARI
How Does Illumina Machine Work?

Check Illumina virtual tour:
https://emea.illumina.com/systems/sequencing-platforms/iseq/tour.html

DNA fragment = Read
Common Disadvantages!

Regardless the sequencing machine, reads still lack information about their order and location (which part of genome they are originated from)
One Problem

Need to construct the entire genome from many sequenced reads
Solving the Puzzle

Reference genome

Reads

Sequencing Output

Small pieces of a puzzle short reads (Illumina)

Large pieces of a puzzle long reads (ONT & PacBio)

Which sequencing technology is the best?

- 100-300 bp
- low error rate (~0.1%)

- 500-2000 bp
- high error rate (~5%)

HiFi Reads (PacBio)

Wenger+, "Accurate circular consensus long-read sequencing improves variant detection and assembly of a human genome", Nature Biotechnology, 2019

https://labs.wsu.edu/genomicscore/illumina-sequencing/
https://pacbio.gs.washington.edu/
Genome Analysis Pipeline

1. Obtaining Genomic Sequencing Data
   - DNA Extraction
   - DNA Fragmentation
   - Library Preparation

2. Basecalling
   - Illumina
   - ONT
   - PacBio

3. Quality Control
   - FastQ

4. Read Mapping
   - Indexing
   - Seeding
   - Pre-alignment Filtering
   - Sequence Alignment

5. Variant Calling
   - Processing Mapping Data
   - Variation Classification
   - Generating Variant Calls

---

Alser+, “Going From Molecules to Genomic Variations to Scientific Discovery: Intelligent Algorithms and Architectures for Intelligent Genome Analysis”, CSBJ, 2022
Read Mapping

A. Indexing
- Reference Genome
- Seeds

<table>
<thead>
<tr>
<th>Seed content</th>
<th>Seed locations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1, 4, 6</td>
<td>1, 4, 6</td>
</tr>
<tr>
<td>3, 5, 12</td>
<td>3, 5, 12</td>
</tr>
<tr>
<td>50, 52</td>
<td>2, 100</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

B. Seeding
- FASTQ
- Read Seeds
- Locating common seeds

C. Pre-alignment Filtering
- Filter 1
- Filter 2
- Filter N

D. Sequence Alignment
- Reference Subsequence
- Dynamic Programming Matrix

Alser+, “Going From Molecules to Genomic Variations to Scientific Discovery: Intelligent Algorithms and Architectures for Intelligent Genome Analysis”, CSBJ, 2022
Read Mapping

Solving life’s puzzle from sequencing output
Read Mapping with a Reference Genome

- Map many short DNA fragments (reads) to a known reference genome with some differences allowed

Mapping short reads to reference genome is challenging (billions of 50-300 base pair reads)
Read Mapping without a Reference Genome

Genome (Non-human-readable)

Sequencing

Reads (Human-readable)

Overlap

Find the ordering (i.e., Layout)

Consensus (i.e., assembly)
Read Mapping: A Naïve Approach

Very expensive!

$O(m^2kn)$

- $m$: read length
- $k$: no. of reads
- $n$: reference genome length
Faster Read Mapping in 3 Steps

.FASTA file:

>NG_008679.1:5001-38170 Homo sapiens paired box 6 (PAX6)
ACCCCTTCATTGACATTTTAACTCTGGGGCAGGGAACGCGGCTGTCAAGATCT
GCCACTTTCCCCTGCGAGCGGGCCTGGAAGAATGGTGGGAACCGGCCTGGCGTCACCTGCCCTCCGCG
CCTCCGCTCCCAGGTAACCCGCCGCCGCCTGGGTCGCCGCGGCGGGGCTTCTCGCTG
CCAGCGACTGCTGGTCTCCCAATTCAAAGCCCGCCCAAGTGGCAGCAGGCTTCTTTGCGT
GAGGAGTAGATGAAGGACGAGGTAATGAGGAGGAGATAGAAGGGGCTTGTGAAGAGGACTTGTGCTT
TCGGGATGTCTACTCTAAAGGTAGCAATGATCTCCTAATGTTCCAGTCTCCGCCCCT
GAGCTGGGAGTGAGGGGCAGGCTCTGCTGGTGGCTGGCTGGTGTCTAAAGGCTACCTCGGACCGCGGAAGTTG
GGAGGTGGGGACGCACTTTGCATCCAGACCTCCCTCTGCAATGGATGATGCCGAGTAGGACCTTACCGTT
TGGGAGGCGCCTCTCAGTAGAAGACACCCTGCGGCGGTGCCCGGGCGAGGTCAGCACAGAAGATTTC
GCCGGTGCAAGTGCGAGATGGCTGAGCGCCGAAACAGTGCTAGAGATGGGTTCTGTCTCAGAAAGACGC

.FASTQ file:

@HWI-EAS209_0006_FC706VJ:5:58:5894:21141#ATCACG/1
TAAATAATCTTTAGATNNNNNNNNNTAG
+
ecffccffccffccffccfffdffdfeed}`feed`\}^\_Ba^\__[YBBBBBBBBBBBBRRTT
Step 1: Indexing the (Reference) Genome

reference genome

?
Step 1: Indexing the Reference Genome

Which seeds (k-mers) to store?
Significantly affects accuracy and space requirements
Seeding Techniques – All K-mers

- Extracting all overlapping k-mers

Reference Genome

```
GCTATTACCTATG...
GCTATTA
CTATTAC
TATTACC
ATTACCT
TTACCTA
TACCTAT
ACCTATG
```

Overlapping k-mers (k=7)

- **Benefits:** High sensitivity as no information is lost
- **Downside:** Incurs large storage space requirements
Seeding Techniques – Minimizers

- Sampling the overlapping k-mers
  - Minimizers
  - Window length \((w)\): accuracy & performance trade-off

\[ \text{k-mers (}k = 7): \]

- **GCTATTA**
- **CTATTAC**
- **TATTACC**
- **ATTACCT**

**Window:** 4 k-mers

**Hash Table**

**Find Min**

**Minimizer k-mer**

**Benefits:** Reduced storage requirements due to (clever) sampling

**Downside:** Reduced sensitivity due to sampling
Seeding Techniques – Spaced Seeds

- Allowing mismatches at **certain positions**
  - **Spaced seeds**
  - **Choice of pattern** is critical for the effectiveness of spaced seeds

**Benefits:** Increase sensitivity due to allowing mismatches

**Downside:** Poor flexibility since the mismatch positions are fixed with pattern
Seeding Techniques – Strobemers

- Allowing insertions and deletions
  - **Linked k-mers** (e.g., strobemers)

**Benefits:** Increase sensitivity due to allowing mismatches and insertion/deletions (indels)

**Downside:** Reduced flexibility as selected k-mers must exactly match, potentially even in the same order.
Seeding Techniques – Fuzzy Seeds

• A mechanism for **finding fuzzy seed matches** can enable
  • Assigning the **same** hash values to **highly similar seeds**
  • **Different** hash values for **dissimilar seeds**
  • **High performance** (e.g., no distance or similarity calculation) and
  • **Space-efficient** (no multiple hash functions for a single sketch) seed matching

- Finding **useful and novel seed matches** that cannot be identified when finding only exact-matching seeds

- **Rethinking the seeding parameters** to achieve better trade-off between
  - Performance, memory, and accuracy
Can Firtina, Jisung Park, Mohammed Alser, Jeremie S. Kim, Damla Senol Cali, Taha Shahrroodi, Nika Mansouri Ghiasi, Gagandeep Singh, Konstantinos Kanellopoulos, Can Alkan, and Onur Mutlu,

"BLEND: A Fast, Memory-Efficient, and Accurate Mechanism to Find Fuzzy Seed Matches in Genome Analysis"


[Online link at NAR Genomics and Bioinformatics Journal](#)

[arXiv preprint](#)

[biorXiv preprint](#)

[BLEND Source Code](#)
Genome Index Properties

- The index is built **only once** for each reference.

- **Seeds** can be overlapping, non-overlapping, spaced, adjacent, Syncmers, Strobemers, BLEND, non-adjacent, minimizers, compressed, ...

<table>
<thead>
<tr>
<th>Tool</th>
<th>Version</th>
<th>Index Size*</th>
<th>Indexing Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>mrFAST</td>
<td>2.2.5</td>
<td>16.5 GB</td>
<td>20.00 min</td>
</tr>
<tr>
<td>minimap2</td>
<td>0.12.7</td>
<td>7.2 GB</td>
<td>3.33 min</td>
</tr>
<tr>
<td>BWA-MEM</td>
<td>0.7.17</td>
<td>4.7 GB</td>
<td>49.96 min</td>
</tr>
</tbody>
</table>

*Human genome = 3.2 GB
Performance of Human Genome Indexing

Alser+, "Technology dictates algorithms: Recent developments in read alignment", Genome Biology, 2021
Step 2: Query the Index Using Read Seeds

seeds

read 1: CCT TAG TAT TAT TAT A CT A G T A C G T T

read 2: TAT TCT TAC G T A C G T A C C G C C C

read 3: G C G T C T T A T A T A C G T A C T A T A C G T T

:::
Step 2: Query the Index Using Read Seeds

- **read 1:** CCTTAGTATAATAGCTACGGTT
- **read 2:** TATTCTTAGGATACGGTTCC
- **read 3:** GCCCTATATCCGTACAGTT

Seed locations:
- Seed from read 1: location list 1 9 16 30
- Seed from read 2: location list 2 7 60
- Seed from read 3: location list 3 5 12

Reference genome:

**SAFARI**
Step 2: Query the Index Using Read Seeds

Query the hash table with hash values of seeds **to quickly find**
list of **possible mapping locations**
Step 3: Sequence Alignment (Verification)

<table>
<thead>
<tr>
<th>C</th>
<th>G</th>
<th>T</th>
<th>T</th>
<th>A</th>
<th>G</th>
<th>T</th>
<th>C</th>
<th>T</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>7</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td>9</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>7</td>
<td>9</td>
<td>11</td>
<td>13</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>7</td>
<td>9</td>
<td>11</td>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>8</td>
<td>9</td>
<td>11</td>
<td>13</td>
<td>14</td>
<td>16</td>
</tr>
</tbody>
</table>

... .bam/.sam file contains necessary alignment information (e.g., type, location, and number of each edit)
**Step 3: Sequence Alignment (Verification)**

- **Edit distance** is defined as the minimum number of edits (i.e. insertions, deletions, or substitutions) needed to make the read exactly match the reference segment.

![Sequence Alignment Diagram]

- **organization x operation**
  - Ref Read: `o - - r g a n i z a t i o n` `op e r - - - - - - - - t i o n`
  - Ref Read: `o - - r g a n i z a t i o n` `op e r - a - - - - - - - - t i o n`
  - Edit distance = 7

- **organization x translation**
  - Ref Read: `o r g a n i z a t i o n`
  - Ref Read: `t r a n s l a t i o n`
  - Ref Read: `o r g a n i z a t i o n`
  - Ref Read: `t r a n s l a t i o n`
  - Edit distance = 4
Read Mapping Execution Time

>60% of the read mapper’s execution time is spent in sequence alignment.

ONT FASTQ size: 103MB (151 reads), Mean length: 356,403 bp, std: 173,168 bp, longest length: 817,917 bp
Challenges in Read Mapping

- Need to find many mappings of each read
  - A short read may map to many locations, especially with High-Throughput DNA Sequencing technologies
  - How can we find all mappings efficiently?

- Need to tolerate small variances/errors in each read
  - Each individual is different: Subject’s DNA may slightly differ from the reference (Mismatches, insertions, deletions) + Sequencer errors
  - How can we efficiently map each read with up to $e$ errors present?

- Need to map each read very fast (i.e., performance is important)
  - Human DNA is 3.2 billion base pairs long → Millions to billions of reads (State-of-the-art mappers take weeks to map a human’s DNA)
  - How can we design a much higher performance read mapper?
Why Is Read Alignment Slow?

- **Quadratic-time** dynamic-programming algorithm(s)

- **Data dependencies** limit the computation parallelism

- **Entire matrix** computed even though strings may be dissimilar
Computational Cost is Mathematically Proven

arXiv.org > cs > arXiv:1412.0348

Computer Science > Computational Complexity

[Submitted on 1 Dec 2014 (v1), last revised 15 Aug 2017 (this version, v4)]

Edit Distance Cannot Be Computed in Strongly Subquadratic Time (unless SETH is false)

Arturs Backurs, Piotr Indyk

The edit distance (a.k.a. the Levenshtein distance) between two strings is defined as the minimum number of insertions, deletions or substitutions of symbols needed to transform one string into another. The problem of computing the edit distance between two strings is a classical computational task, with a well-known algorithm based on dynamic programming. Unfortunately, all known algorithms for this problem run in nearly quadratic time.

In this paper we provide evidence that the near-quadratic running time bounds known for the problem of computing edit distance might be tight. Specifically, we show that, if the edit distance can be computed in time $O(n^{2-\delta})$ for some constant $\delta > 0$, then the satisfiability of conjunctive normal form formulas with $N$ variables and $M$ clauses can be solved in time $M^{O(1)2^{(1-\varepsilon)N}}$ for a constant $\varepsilon > 0$. The latter result would violate the Strong Exponential Time Hypothesis, which postulates that such algorithms do not exist.

https://arxiv.org/abs/1412.0348
Read Mapping in 111 pages!

In-depth analysis of 107 read mappers (1988-2020)

Mohammed Alser, Jeremy Rotman, Dhrithi Deshpande, Kodi Taraszka, Huwenbo Shi, Pelin Icer Baykal, Harry Taegyun Yang, Victor Xue, Sergey Knyazev, Benjamin D. Singer, Brunilda Balliu, David Koslicki, Pavel Skums, Alex Zelikovsky, Can Alkan, Onur Mutlu, Serghei Mangul

"Technology dictates algorithms: Recent developments in read alignment"

Genome Biology, 2021

[Source code]

Technology dictates algorithms: recent developments in read alignment

Mohammed Alser\textsuperscript{1,2,3\textdagger}, Jeremy Rotman\textsuperscript{4\textdagger}, Dhrithi Deshpande\textsuperscript{5}, Kodi Taraszka\textsuperscript{4}, Huwenbo Shi\textsuperscript{6,7}, Pelin Icer Baykal\textsuperscript{8}, Harry Taegyun Yang\textsuperscript{4,9}, Victor Xue\textsuperscript{4}, Sergey Knyazev\textsuperscript{8}, Benjamin D. Singer\textsuperscript{10,11,12}, Brunilda Balliu\textsuperscript{13},
David Koslicki\textsuperscript{14,15,16}, Pavel Skums\textsuperscript{8}, Alex Zelikovsky\textsuperscript{8,17}, Can Alkan\textsuperscript{2,18}, Onur Mutlu\textsuperscript{1,2,3\textdagger} and Serghei Mangul\textsuperscript{5\textdagger}\textsuperscript{\textdagger}
Review

From molecules to genomic variations: Accelerating genome analysis via intelligent algorithms and architectures

Mohammed Alser*, Joel Lindegger, Can Firtina, Nour Almadhoun, Haiyu Mao, Gagandeep Singh, Juan Gomez-Luna, Onur Mutlu*

*ETH Zurich, Gloriastrasse 35, 8092 Zürich, Switzerland
What happens if there is no available reference genome of an organism
De Novo Genome Assembly

Reference-free
Why do we rely on a single reference genome?
Recall: Solving the Puzzle

Reference Genome

For a Human:
3 Billion Characters (3GB)

Determines e.g., Eye Color, Shape of Face, Allergies, ...
Recall: Solving the Puzzle

Reference Genome

For a Human:
3 Billion Characters (3GB)

Determines e.g., Eye Color, Shape of Face, Allergies, ...

Reads

150 – 2,000,000 Characters Each

Origin Locations are Unknown
Reference Genome

For a Human:
3 Billion Characters (3GB)

Determines e.g., Eye Color, Shape of Face, Allergies, ...

Reads

150 – 2,000,000 Characters Each

Recovers the Origin Locations According to 1 Reference Genome
The Problem: Variants in the Genome

Reference Genome

Reads

Variants
The Problem: Variants in the Genome

Reference Genome

Some Reads Can Be Mapped due to Sufficient Context

Some Reads Fail to Be Mapped Because They are Too Different from the Single Reference

Reference Bias!
Solution 1: Attempt to map to all known reference genomes one-by-one

- For N times slowdown for N reference genomes
- There could be unknown reference genomes (e.g., hybrids)
Solution 2: Build a single graph-based reference that unifies all known genetic variations

- Avoids redundant computation and data
- Captures some unknown reference genomes
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
Changes in sequencing technologies can render some read mapping algorithms irrelevant.
Looking forward,
Will we be able to read
the entire genome sequence?
Genome Analysis Pipeline

Alser+, “Going From Molecules to Genomic Variations to Scientific Discovery: Intelligent Algorithms and Architectures for Intelligent Genome Analysis”, CSBJ, 2022
Variant Calling

A Processing Mapping Data

B Variation Classification

C Generating Variant Calls

Removing read duplicates and cleaning read mapping results

Reference

Aligned Reads

List of variants

SAFARI

Alser+, “Going From Molecules to Genomic Variations to Scientific Discovery: Intelligent Algorithms and Architectures for Intelligent Genome Analysis”, CSBJ, 2022
Goal of Variant Calling

1. **Get a consensus between the aligned reads**
   How does the sequenced genome *really* look at a given location?

2. **Compare the sequenced genome to the reference genome**
   Where do they differ? How?
   Is this variant a sequencing error or a real variant?
Types of Variants

- **SNP (Single Nucleotide Polymorphism)**
  - “A character substitution”
  - Conceptually simple to call
  - “Average” the CIGAR string
  - Complexity arises from the accuracy requirement

- **Indels (Insertions, Deletions)**
  - A block of inserted or deleted characters
  - Typically small, e.g., 1-50 characters
  - Conceptually difficult to call
  - Results from mapping require re-interpretation

- **SVs (Structural Variants)**
  - Rearrangements of substrings
  - Often span a large range

- **CNVs (Copy Number Variations)**
Approaches to Variant Calling*

1. “Naïve” Voting-based

2. Bayesian

3. Local Assembly + Bayesian

4. Deep Learning

* for SNPs and Small Indels
“Naïve” Voting-based Variant Calling

- **Goal:** Identify SNPs

- **Key Idea:** *Majority vote* each base among aligned reads

---

**Generating Variant Calls**

<table>
<thead>
<tr>
<th>Reference</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>G</td>
</tr>
<tr>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>A</td>
<td>G</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>G</td>
</tr>
<tr>
<td>C</td>
<td>A</td>
</tr>
<tr>
<td>T</td>
<td></td>
</tr>
</tbody>
</table>

**Aligned Reads**

<table>
<thead>
<tr>
<th>A</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>G</td>
</tr>
<tr>
<td>A</td>
<td>T</td>
</tr>
<tr>
<td>G</td>
<td>A</td>
</tr>
<tr>
<td>A</td>
<td>G</td>
</tr>
<tr>
<td>G</td>
<td>T</td>
</tr>
</tbody>
</table>

**List of variants**

<table>
<thead>
<tr>
<th>CHR</th>
<th>POS</th>
<th>ID</th>
<th>REF</th>
<th>ALT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chr 1</td>
<td>113834946</td>
<td>rs2476601</td>
<td>A</td>
<td>G</td>
</tr>
</tbody>
</table>

*VCF*
Goal: Identify SNPs

Key Idea: Majority vote each base among aligned reads
- To me this is what the previous figure implies

Problems
- Limited to SNPs
- Cannot report
  - Indels
  - Diploid variants (separate variants on the chromosomes from each parent)
  - Haplotype types of more than one base (e.g., a series of closely co-located SNPs)
- Ignores
  - Biased sequencing errors
  - Statistical distributions in the population
Bayesian Variant Calling

  - Includes probabilities for
    - SNPs already known from a database (or not)
    - Real SNPs vs. sequencing errors

From observed prior probability, the SNP rate is expected to be about 0.1% and the most common SNPs should already be present in dbSNP. Therefore, for positions without known polymorphisms, on one haploid, the reference bases will dominate the prior probability as 0.999; others will share the remaining 0.1% mutation rate. Because sequencing errors would look like heterozygous (HET) SNPs, a penalty factor of 0.001 is multiplied to the HET prior probability. At dbSNP sites, bases already observed dominate the prior probability equally and the HET penalty factor is 0.01. As a result, the prior probabilities were as follows: (1) 0.45 for a homozygote and 0.1 for a heterozygote at a SNP site that has been validated in dbSNP; (2) 0.495 for a homozygote and 0.01 for a heterozygote at a SNP site that has not been validated in dbSNP; and (3) $1 \times 10^{-6}$ for a homozygote and $2 \times 10^{-6}$ for a heterozygote at a potentially novel SNP site (one that is absent in dbSNP).
Bayesian Variant Calling

  - Includes probabilities for
    - SNPs already known from a database (or not)
    - Real SNPs vs. sequencing errors

- Problems
  - Limited to SNPs and short indels
  - Limited accuracy due to simplistic error model
Variant Calling: GATK HaplotypeCaller

- Poplin+, “Scaling accurate genetic variant discovery to tens of thousands of samples”, bioRxiv, 2017
  - GATK HaplotypeCaller
  - Use probabilities for
    - Sequencer error model
    - Base quality scores
    - Population distribution
    - Biological probability for homo-/heterozygosity

- Brief Overview over the GATK Haplotype Caller
Machine Learning-based Variant Calling

- Poplin+, “A universal SNP and small-indel variant caller using deep neural networks”, Nature Biotechnology, 2018
  - DeepVariant
  - Represent alignments as images
    - Create features accordingly

- Advantages
  - Higher accuracy than any previous method
  - Generalizes well to different sequencing technologies

https://google.github.io/deepvariant/posts/2020-02-20-looking-through-deepvariants-eyes/
GATK Haplotype Caller vs DeepVariant

Lin+, “Comparison of GATK and DeepVariant by trio sequencing”, Nature Scientific Reports”, 2022
Structural Variant Calling

- Large variations (>50 bases)
Structural Variant Calling

- Short reads are too small to cover entire variations
  - Generate large assemblies to span the variation

- Long reads are more advantageous
  - Can cover most, if not all, region of a variation

- Requires highly specialized tools that are not applicable to SNP/short indel calling
  - Mahmoud+, “Structural variant calling: the long and the short of it”, Genome Biology, 2019
Agenda for Today

- Introduction to Genomics
  - Today: The ways we analyze genomes
  - Intelligent genome analysis

- Step-by-step Genome Analysis
  - Sequencing
  - Read Mapping
  - Variant Calling

- Algorithmic & Hardware Acceleration

- Future Opportunities: New Technologies & Applications
Significant barriers to intelligent analyses
Significant Barriers to Intelligent Analyses

1. Performance gap between data generation and data processing
1 Sequencing

2 Read Mapping

Reference Genome

... CACTATACG

... CCTATAAATACG

3 Variant Calling

4 Scientific Discovery

Billions of Short Reads

TTTAGTACGTACGT

TTTAGTACGTACGT

TTTAGTACGTACGT

TTTAGTACGTACGT

TTTTAAAAACGTAA

GGGAGTACGTACGT

Reference: TTTATCGCTTTCCATGACGCAG

Read 1: ATCGC ATCC

Read 2: TATCGC ATC

Read 3: CATCCATGA

Read 4: CGCTTCCAT

Read 5: CCATGACGC

Read 6: TTCCATGAC
We Are Bottlenecked in Read Mapping

1. Sequencing
2. Read Mapping

Illumina HiSeq4000

300 M bases/min

on average

2 M bases/min

(0.6%)
The Read Mapping Bottleneck

300 Million bases/minute
Read Sequencing **

2 Million bases/minute
Read Mapping *

150x slower

* BWA-MEM
** HiSeqX10, MinION
## Technology Dictates Algorithm Complexity

### Short Reads (Illumina)

<table>
<thead>
<tr>
<th>Step</th>
<th>Task</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sequencing</td>
<td>Library preparation: 6.5 hours, Sequencing: 68.2 Gb/hour</td>
</tr>
<tr>
<td>2</td>
<td>Basecalling</td>
<td>104.4 Gb/hour</td>
</tr>
<tr>
<td>3</td>
<td>Quality Control</td>
<td>1339.2 Gb/hour</td>
</tr>
<tr>
<td>4</td>
<td>Read Mapping</td>
<td>0.2 Gb/hour</td>
</tr>
<tr>
<td>5</td>
<td>Variant Calling</td>
<td>1.2 Gb/hour</td>
</tr>
</tbody>
</table>

### Ultra-long Reads (ONT)

<table>
<thead>
<tr>
<th>Step</th>
<th>Task</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sequencing</td>
<td>Library preparation: 24 hours, Sequencing: 4.1 Gb/hour</td>
</tr>
<tr>
<td>2</td>
<td>Basecalling</td>
<td>0.833 Gb/hour</td>
</tr>
<tr>
<td>3</td>
<td>Quality Control</td>
<td>3420 Gb/hour</td>
</tr>
<tr>
<td>4</td>
<td>Read Mapping</td>
<td>1.7 Gb/hour</td>
</tr>
<tr>
<td>5</td>
<td>Variant Calling</td>
<td>0.044 Gb/hour</td>
</tr>
</tbody>
</table>

### Accurate Long Reads (PacBio)

<table>
<thead>
<tr>
<th>Step</th>
<th>Task</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sequencing</td>
<td>Library preparation: 24 hours, Sequencing: 5.3 Gb/hour</td>
</tr>
<tr>
<td>2</td>
<td>Basecalling</td>
<td>8.3 Gb/hour</td>
</tr>
<tr>
<td>3</td>
<td>Quality Control</td>
<td>1081 Gb/hour</td>
</tr>
<tr>
<td>4</td>
<td>Read Mapping</td>
<td>1.4 Gb/hour</td>
</tr>
<tr>
<td>5</td>
<td>Variant Calling</td>
<td>1.1 Gb/hour</td>
</tr>
</tbody>
</table>

---

Alser+, *Going From Molecules to Genomic Variations to Scientific Discovery: Intelligent Algorithms and Architectures for Intelligent Genome Analysis*, arXiv 2022
Problems with (Genome) Analysis Today

Special-Purpose Machine for Data Generation

General-Purpose Machine for Data Analysis

FAST

SLOW

Slow and inefficient processing capability
Large amounts of data movement

SAFARI This picture is similar for many “data generators & analyzers” today
Significant Barriers to Intelligent Analyses

1. Performance gap between data generation and data processing

2. Expensive data movements
Data Movement Dominates Performance

- **Data movement** dominates performance and is a **major system energy bottleneck** (accounting for 40%-62%)

---

* Boroumand et al., “Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks,” ASPLOS 2018
* Kestor et al., “Quantifying the Energy Cost of Data Movement in Scientific Applications,” IISWC 2013
* Pandiyan and Wu, “Quantifying the energy cost of data movement for emerging smart phone workloads on mobile platforms,” IISWC 2014

---

*Single memory request consumes >160x-800x more energy compared to performing an addition operation*. 

---

**SAFARI**
Data analysis is performed far away from the data.
Significant Barriers to Intelligent Analyses

1. Performance gap between data \textit{generation} and data \textit{processing}

2. Expensive data movements

3. Neglecting metadata
   1. Types of sequencing data
   2. Properties of intermediate data
   3. Quality of data
   4. Genome structure
Significant Barriers to Intelligent Analyses

1. Performance gap between data generation and data processing
2. Expensive data movements
3. Neglecting metadata
4. And many more barriers specific to each computational step ...
Challenges in Genome Analysis

- **Basecalling**: Each sequencing technology provides *different types* of raw sequencing data.

- **Error correction & quality control**: Sequencing error rates vary from 0.1%-15%.

- **Read mapping**: Regardless the sequencing machine, reads are still *small randomized fragments* of the original DNA sequence with unknown *order* and *location*.

- **Variant calling**: Small & complex *genomic differences* need to be maintained.

- **Metagenomic profiling**: The sample donor is *unknown*.
Richard Feynman, "There's Plenty of Room at the Bottom: An Invitation to Enter a New Field of Physics", a lecture given at Caltech, 1959.

Multiplying Two 4096-by-4096 Matrices

```
for i in xrange(4096):
    for j in xrange(4096):
        for k in xrange(4096):
            C[i][j] += A[i][k] * B[k][j]
```

<table>
<thead>
<tr>
<th>Implementation</th>
<th>Running time (s)</th>
<th>Absolute speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Python</td>
<td>25,552.48</td>
<td>1x</td>
</tr>
<tr>
<td>Java</td>
<td>2,372.68</td>
<td>11x</td>
</tr>
<tr>
<td>C</td>
<td>542.67</td>
<td>47x</td>
</tr>
<tr>
<td>Parallel loops</td>
<td>69.80</td>
<td>366x</td>
</tr>
<tr>
<td>Parallel divide and conquer</td>
<td>3.80</td>
<td>6,727x</td>
</tr>
<tr>
<td>plus vectorization</td>
<td>1.10</td>
<td>23,224x</td>
</tr>
<tr>
<td>plus AVX intrinsics</td>
<td>0.41</td>
<td>62,806x</td>
</tr>
</tbody>
</table>

Leiserson+, "There’s plenty of room at the Top: What will drive computer performance after Moore’s law?", Science, 2020
## FASTQ Parsing

<table>
<thead>
<tr>
<th>Program</th>
<th>Language</th>
<th>$t_{gzip}$ (s)</th>
<th>$t_{plain}$ (s)</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>fqcnt_rs2_needletail.rs</td>
<td>Rust</td>
<td>9.3</td>
<td>0.8</td>
<td>needletail; fasta/4-line fastq</td>
</tr>
<tr>
<td>fqcnt_c1_kseq.c</td>
<td>C</td>
<td>9.7</td>
<td>1.4</td>
<td>multi-line fasta/fastq</td>
</tr>
<tr>
<td>fqcnt_cr1_klib.cr</td>
<td>Crystal</td>
<td>9.7</td>
<td>1.5</td>
<td>kseq.h port</td>
</tr>
<tr>
<td>fqcnt_nim1_klib.nim</td>
<td>Nim</td>
<td>10.5</td>
<td>2.3</td>
<td>kseq.h port</td>
</tr>
<tr>
<td>fqcnt_jl1_klib.jl</td>
<td>Julia</td>
<td>11.2</td>
<td>2.9</td>
<td>kseq.h port</td>
</tr>
<tr>
<td>fqcnt_js1_k8.js</td>
<td>Javascript</td>
<td>17.5</td>
<td>9.4</td>
<td>kseq.h port</td>
</tr>
<tr>
<td>fqcnt_go1.go</td>
<td>Go</td>
<td>19.1</td>
<td>2.8</td>
<td>4-line only</td>
</tr>
<tr>
<td>fqcnt_lua1_klib.lua</td>
<td>LuaJIT</td>
<td>28.6</td>
<td>27.2</td>
<td>partial kseq.h port</td>
</tr>
<tr>
<td>fqcnt_py2_rfq.py</td>
<td>PyPy</td>
<td>28.9</td>
<td>14.6</td>
<td>partial kseq.h port</td>
</tr>
<tr>
<td>fqcnt_py2_rfq.py</td>
<td>Python</td>
<td>42.7</td>
<td>19.1</td>
<td>partial kseq.h port</td>
</tr>
</tbody>
</table>

[https://github.com/lh3/biofast](https://github.com/lh3/biofast)
We need intelligent algorithms and intelligent architectures that handle data well
Accelerating Read Mapping

Our Contributions

- GateKeeper [Bioinformatics'17]
- MAGNET [AACBB'18]
- Shouji [Bioinformatics'19]
- SneakySnake [Bioinformatics'20]
- GenASM [MICRO 2020]
- GateKeeper-GPU [arXiv'21]
- GRIM-Filter [BMC Genomics'18]
- Near-memory/In-memory Pre-alignment Filtering
- Specialized Pre-alignment Filtering Accelerators (GPU, FPGA)
- We will cover many of these works making genome analysis faster, more accurate & scalable

- GenStore [ASPLOS 2022]
- SneakySnake [Bioinformatics’20]

Sequencing Machine, Storage (SSD/HDD), Main Memory, Microprocessor
Pre-alignment Filtering Techniques

Sequence Alignment is expensive

Our goal is to reduce the need for dynamic programming algorithms
Key Idea

Genomic Strings

Dissimilar Strings
Ignore them if the number of differences exceeds a threshold.

Similar Strings
Find number and location of differences?

EXPENSIVE!
1. Filter out most of incorrect mappings.
2. Preserve all correct mappings.
3. Do it quickly.
GateKeeper: a new hardware architecture for accelerating pre-alignment in DNA short read mapping

Mohammed Alser, Hasan Hassan, Hongyi Xin, Oğuz Ergin, Onur Mutlu, Can Alkan

*Bioinformatics*, Volume 33, Issue 21, 01 November 2017, Pages 3355–3363,
https://doi.org/10.1093/bioinformatics/btx342

Published: 31 May 2017  Article history ▼

GateKeeper

- **Key observation:**
  - If two strings differ by \( E \) edits, then every bp match can be aligned in at most \( 2E \) shifts.

- **Key idea:**
  - Compute “Shifted Hamming Distance”: AND of \( 2E+1 \) Hamming vectors of two strings, to identify invalid mappings
    - Uses *bit-parallel operations* that nicely map to FPGA architectures

- **Key result:**
  - GateKeeper is 90x-130x faster than SHD (Xin et al., 2015) and the Adjacency Filter (Xin et al., 2013), with only a 7% false positive rate
  - The addition of GateKeeper to the mrFAST mapper (Alkan et al., 2009) results in 10x end-to-end speedup in read mapping
Hamming Distance ($\sum \oplus$)

3 matches  5 mismatches

*Edit = 1 Deletion*

To cancel the effect of a deletion, we need to shift in the *right* direction
Shifted Hamming Distance (Xin+ 2015)

**XOR**

ISTANBUL

**AND**

0 0 0 1 1 1 1 1

1 1 1 0 0 0 0 0

Count 1’s

0 0 0 0 1 0 0 0 0 0

7 matches 1 mismatches

Edit = 1 Deletion
GateKeeper Walkthrough

Generate 2E+1 masks

Amend random zeros:
101 → 111 & 1001 → 1111

AND all masks, ACCEPT iff number of ‘1’ ≤ Threshold

Query: GAGAGAGATTTTAGTGTTGCAGCACACTACAACACAAAAGAGGACCAACTTACGTTCTAAAAGGGAACATTGGTTGGCCGG
Reference: GAGAGAGATTTAGTGTTGCAGCACACTACAACACAAAAGAGGACCAACTTACGTTCTAAAAGGGAACATTGGTTGGCCGG

Hamming Mask:

1-Deletion Mask:

2-Deletion Mask:

3-Deletion Mask:

1-Insertion Mask:

2-Insertion Mask:

3-Insertion Mask:

AND Mask:

Our goal to track the diagonally consecutive matches in the neighborhood map.

Needleman-Wunsch Alignment:

S A F A R I
Alignment Matrix vs. Neighborhood Map

### Needleman-Wunsch

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>T</th>
<th>A</th>
<th>T</th>
<th>A</th>
<th>T</th>
<th>A</th>
<th>C</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-1</td>
<td>-2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-2</td>
<td>-3</td>
<td>-2</td>
<td>-1</td>
<td>-2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-3</td>
<td>-3</td>
<td>-2</td>
<td>-1</td>
<td>-2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-4</td>
<td>-3</td>
<td>-2</td>
<td>-1</td>
<td>-2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-4</td>
<td>-3</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Neighborhood Map

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>T</th>
<th>A</th>
<th>T</th>
<th>A</th>
<th>T</th>
<th>A</th>
<th>C</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Independent vectors can be processed in parallel using hardware technologies
Our Solution: GateKeeper

Alignment Filter + FPGA-based Alignment Filter =

\(10^{12}\) mappings

High throughput DNA sequencing (HTS) technologies

Billions of Short Reads

\(10^3\) mappings

Read Pre-Alignment Filtering
Fast & Low False Positive Rate

Read Alignment
Slow & Zero False Positives
GateKeeper: Speed & Accuracy Results

90x-130x faster filter
than SHD (Xin et al., 2015) and the Adjacency Filter (Xin et al., 2013)

4x lower false accept rate
than the Adjacency Filter (Xin et al., 2013)

10x speedup in read mapping
with the addition of GateKeeper to the mrFAST mapper (Alkan et al., 2009)

Freely available online
github.com/BilkentCompGen/GateKeeper
More on SHD (SIMD Implementation)

- Download and test for yourself
- https://github.com/CMU-SAFARI/Shifted-Hamming-Distance

Sequence analysis

Shifted Hamming distance: a fast and accurate SIMD-friendly filter to accelerate alignment verification in read mapping

Hongyi Xin¹,*, John Greth², John Emmons², Gennady Pekhimenko¹, Carl Kingsford³, Can Alkan⁴,* and Onur Mutlu²,*
More on GateKeeper

- Download and test for yourself
  [https://github.com/BilkentCompGen/GateKeeper](https://github.com/BilkentCompGen/GateKeeper)

Alser+, "GateKeeper: A New Hardware Architecture for Accelerating Pre-Alignment in DNA Short Read Mapping" in Bioinformatics, 2017.

Bioinformatics, Volume 33, Issue 21, 01 November 2017, Pages 3355–3363,
[https://doi.org/10.1093/bioinformatics/btx342](https://doi.org/10.1093/bioinformatics/btx342)

Published: 31 May 2017  Article history ▼
Can we do better? Scalability?
Mohammed Alser, Taha Shahroodi, Juan-Gomez Luna, Can Alkan, and Onur Mutlu, "SneakySnake: A Fast and Accurate Universal Genome Pre-Alignment Filter for CPUs, GPUs, and FPGAs" *Bioinformatics*, 2020.

[Source Code]
[Online link at Bioinformatics Journal]
Key observation:
- Correct alignment is a sequence of non-overlapping long matches.

Dot plot, dot matrix (Lipman and Pearson, 1985)
SneakySnake

- **Key observation:**
  - Correct alignment is a sequence of non-overlapping long matches

- **Key idea:**
  - Approximate edit distance calculation is similar to Single Net Routing problem in VLSI chip
Given two genomic sequences, a reference sequence $R[1 \ldots m]$ and a query sequence $Q[1 \ldots m]$, and an edit distance threshold $E$, we calculate the entry $Z[i, j]$ of the chip maze, where $1 \leq i \leq (2E + 1)$ and $1 \leq j \leq m$, as follows:

$$Z[i, j] = \begin{cases} 
0, & \text{if } i = E + 1, Q[j] = R[j], \\
0, & \text{if } 1 \leq i \leq E, Q[j - i] = R[j], \\
0, & \text{if } i > E + 1, Q[j + i - E - 1] = R[j], \\
1, & \text{otherwise}
\end{cases} \quad (1)$$

<table>
<thead>
<tr>
<th>column</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>3rd Upper Diagonal</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2nd Upper Diagonal</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1st Upper Diagonal</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Main Diagonal</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1st Lower Diagonal</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2nd Lower Diagonal</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3rd Lower Diagonal</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
SneakySnake Walkthrough

Building Neighborhood Map  Finding the Optimal Routing Path  Examining the Snake Survival

\[ E = 3 \]

[Diagram of neighborhood map with ENTRANCE and EXIT markers]
SneakySnake Walkthrough

- Building Neighborhood Map
- Finding the Optimal Routing Path
- Examining the Snake Survival

![Diagram of SneakySnake Walkthrough](image-url)
This is what you actually need to build and it can be done on-the-fly!
FPGA Resource Analysis

- FPGA resource usage for a single filtering unit of GateKeeper, Shouji, and Snake-on-Chip for a sequence length of 100 and under different edit distance thresholds (E).

<table>
<thead>
<tr>
<th></th>
<th>$E$ (bp)</th>
<th>Slice LUT</th>
<th>Slice Register</th>
<th>No. of Filtering Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>GateKeeper</td>
<td>2</td>
<td>0.39%</td>
<td>0.01%</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.71%</td>
<td>0.01%</td>
<td>16</td>
</tr>
<tr>
<td>Shouji</td>
<td>2</td>
<td>0.69%</td>
<td>0.08%</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>1.72%</td>
<td>0.16%</td>
<td>16</td>
</tr>
<tr>
<td>Snake-on-Chip</td>
<td>2</td>
<td>0.68%</td>
<td>0.16%</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>1.42%</td>
<td>0.34%</td>
<td>16</td>
</tr>
</tbody>
</table>
Key Results of SneakySnake

- SneakySnake is up to four orders of magnitude more accurate than Shouji (Bioinformatics‘19) and GateKeeper (Bioinformatics‘17)

- Using short reads, SneakySnake accelerates Edlib (Bioinformatics‘17) and Parasail (BMC Bioinformatics‘16) by
  - up to 37.7× and 43.9× (>12× on average), on CPUs
  - up to 413× and 689× (>400× on average) with FPGA/GPU acceleration

- Using long reads, SneakySnake accelerates Parasail and KSW2 by 140.1× and 17.1× on average, respectively, on CPUs
Data Movement Dominates Performance

- **Data movement** dominates performance and is a **major** system **energy bottleneck** (accounting for 40%-62%).

Single memory request **consumes >160x-800x more energy** compared to performing an **addition operation**.

* Boroumand et al., “Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks,” ASPLOS 2018
* Kestor et al., “Quantifying the Energy Cost of Data Movement in Scientific Applications,” IISWC 2013
* Pandiyan and Wu, “Quantifying the energy cost of data movement for emerging smart phone workloads on mobile platforms,” IISWC 2014
Read Mapping & Filtering in Memory

We need to design mapping & filtering algorithms that fit processing-in-memory
Processing Using Memory

Computer Architecture
Lecture 6a: RowHammer & Secure and Reliable Memory II

Prof. Onur Mutlu
ETH Zürich
Fall 2021
15 October 2021

https://www.youtube.com/watch?v=HNd4skQrt6I
Processing Using Memory II

Computer Architecture
Lecture 7: Processing using Memory II

Dr. Juan Gómez Luna
Prof. Onur Mutlu
ETH Zürich
Fall 2021
21 October 2021

Computer Architecture - Lecture 7: Processing using Memory II (Fall 2021)
558 views • Streamed live on Oct 21, 2021

Source: https://www.youtube.com/watch?v=k56x2qcaXWY
Computer Architecture
Lecture 8: Processing near Memory

Dr. Juan Gómez Luna
Prof. Onur Mutlu
ETH Zürich
Fall 2021
22 October 2021

Computer Architecture - Lecture 8: Processing near Memory (Fall 2021)
759 views • Streamed live on Oct 22, 2021

Onur Mutlu Lectures
19.6K subscribers

https://www.youtube.com/watch?v=kpgLmX9sdcI
Using Real PIM System

https://www.youtube.com/watch?v=TuVw_SKaTCo
Near-memory Pre-alignment Filtering

Gagandeep Singh, Mohammed Alser, Damla Senol Cali, Dionysios Diamantopoulos, Juan Gomez-Luna, Henk Corporaal, Onur Mutlu,

“FPGA-Based Near-Memory Acceleration of Modern Data-Intensive Applications”

[Source Code]
Near-memory SneakySnake

- Problem: Read Mapping is heavily bottlenecked by data movement from main memory

- Solution: Perform read mapping near where data resides (i.e., near-memory)

- We carefully redesigned the accelerator logic of SneakySnake to exploit near-memory computation capability on modern FPGA boards with high-bandwidth memory
Heterogeneous System: CPU+FPGA

We evaluate two POWER9+FPGA systems:

1. **HBM-based AD9H7 board**: Xilinx Virtex Ultrascale+™ XCVU37P-2
2. **DDR4-based AD9V3 board**: Xilinx Virtex Ultrascale+™ XCVU3P-2

FPGA + HBM on the same package substrate
Key Results of Near-memory SneakySnake

Near-memory pre-alignment filtering improves performance and energy efficiency by 27.4× and 133×, respectively, over a 16-core (64 hardware threads) IBM POWER9 CPU
Mohammed Alser, Taha Shahroodi, Juan-Gomez Luna, Can Alkan, and Onur Mutlu, "SneakySnake: A Fast and Accurate Universal Genome Pre-Alignment Filter for CPUs, GPUs, and FPGAs" *Bioinformatics*, 2020.
[Source Code]
[Online link at Bioinformatics Journal]
Jeremie S. Kim, Damla Senol Cali, Hongyi Xin, Donghyuk Lee, Saugata Ghose, Mohammed Alser, Hasan Hassan, Oguz Ergin, Can Alkan, and Onur Mutlu,
"GRIM-Filter: Fast Seed Location Filtering in DNA Read Mapping Using Processing-in-Memory Technologies"
*Proceedings of the 16th Asia Pacific Bioinformatics Conference (APBC),* Yokohama, Japan, January 2018.
GRIM-Filter

- **Key observation:** FPGA and GPU accelerators are Heavily bottlenecked by Data Movement.

- **Key idea:** exploiting the high memory bandwidth and the logic layer of 3D-stacked memory to perform highly-parallel filtering in the DRAM chip itself.

- **Key results:**
  - We propose an algorithm called **GRIM-Filter**
  - GRIM-Filter with processing-in-memory is 1.8x-3.7x (2.1x on average) faster than FastHASH filter (BMC Genomics’13) across real data sets.
  - GRIM-Filter has 5.6x-6.4x (6.0x on average) lower falsely accepted pairs than FastHASH filter (BMC Genomics’13) across real data sets.
GRIM-Filter: Bitvectors

- Represent each bin with a bitvector that holds the occurrence of all permutations of a small string (token) in the bin.

- To account for matches that straddle bins, we employ overlapping bins.
  - A read will now always completely fall within a single bin.
GRIM-Filter: Bitvectors

Storing all bitvectors requires $4^n \times t$ bits in memory, where $t =$ number of bins & $n =$ token length.

For bin size $\sim200$, and $n = 5$, memory footprint $\sim3.8$ GB
GRIM-Filter: Checking a Bin

How GRIM-Filter determines whether to **discard** potential match locations in a given bin **prior** to alignment

**INPUT:** Read Sequence \( r \)

\[ \text{GAACgTTGGAGTCTA} \cdots \text{CGAG} \]

**1. Get tokens**

**2. Read bitvector for bin_num(x)**

**3. Match tokens to bitvector**

**4. Sum**

\[ 1 \quad 0 \quad 1 \quad \cdots \quad 1 \quad 0 \quad 0 \]

**5. Compare**

\[ \geq \text{Threshold?} \]

- NO: Discard
- YES: Send to Read Mapper for Sequence Alignment
Key Properties of GRIM-Filter

1. **Simple Operations:**
   - To check a given bin, find the **sum** of all bits corresponding to each token in the read
   - **Compare** against threshold to determine whether to align

2. **Highly Parallel:** Each bin is operated on independently and there are many many bins

3. **Memory Bound:** Given the frequent accesses to the large bitvectors, we find that GRIM-Filter is memory bound

**These properties together make GRIM-Filter a good algorithm to be run in 3D-Stacked DRAM**
Each DRAM layer is organized as an array of banks

- A bank is an array of cells with a row buffer to transfer data

The layout of bitvectors in a bank enables filtering many bins in parallel
Customized logic for accumulation and comparison per genome segment
- Low area overhead, simple implementation
- For HBM2, we use 4096 incrementer LUTs, 7-bit counters, and comparators in logic layer

SAFARI  Details are in [Kim+, BMC Genomics 2018]
More on GRIM-Filter

Background: Read Mappers

We now have **sequenced reads** and want a **full genome** via Read Mapping.

We map **reads** to a known **reference genome** (>99.9% similarity across humans) with some minor errors allowed.

Because of high similarity, long sequences in **reads** perfectly match in the **reference genome**

```
GACTGTGTCAA
```

...GACTGTGTCGA ...

We can use a hash table to help quickly map the **reads**!

SAFARI

Livestream - P&S Accelerating Genome Analysis with FPGAs, GPUs, and New Execution Paradigms (Fall 2021)

**GRIM-Filter: Fast Seed Location Filtering in DNA Read Mapping w/ Processing-in-Memory**

- Jeremie Kim

[Link to YouTube video](https://www.youtube.com/watch?v=j5-I84iNVd8)
More on GRIM-Filter

Jeremie S. Kim, Damla Senol Cali, Hongyi Xin, Donghyuk Lee, Saugata Ghose, Mohammed Alser, Hasan Hassan, Oguz Ergin, Can Alkan, and Onur Mutlu,
"GRIM-Filter: Fast Seed Location Filtering in DNA Read Mapping Using Processing-in-Memory Technologies"

Proceedings of the 16th Asia Pacific Bioinformatics Conference (APBC), Yokohama, Japan, January 2018.
arxiv.org Version (pdf)
GenASM Framework [MICRO 2020]


[Lightning Talk Video (1.5 minutes)]
[Lightning Talk Slides (pptx) (pdf)]
[Talk Video (18 minutes)]
[Slides (pptx) (pdf)]
GenASM: ASM Framework for GSA

Our Goal:
Accelerate approximate string matching by designing a fast and flexible framework, which can accelerate *multiple steps of genome sequence analysis*

- **GenASM:** *First* ASM acceleration framework for GSA
  - Based on the *Bitap* algorithm
    - Uses *fast and simple bitwise operations* to perform ASM
  - Modified and extended ASM algorithm
    - *Highly-parallel Bitap* with long read support
    - *Bitvector-based* novel algorithm to perform *traceback*
  - Co-design of our modified scalable and memory-efficient algorithms with *low-power and area-efficient hardware accelerators*
GenASM: Hardware Design

**GenASM-DC:**
generates bitvectors and performs edit Distance Calculation

**GenASM-TB:**
performs TraceBack and assembles the optimal alignment
Linear cyclic systolic array based accelerator
- Designed to maximize parallelism and minimize memory bandwidth and memory footprint

Processing Block (PB)

Processing Core (PC)
GenASM-TB: Hardware Design

Very simple logic:

1. Reads the bitvectors from one of the TB-SRAMs using the computed address
2. Performs the required bitwise comparisons to find the traceback output for the current position
3. Computes the next TB-SRAM address to read the new set of bitvectors
Key Results

(1) Read Alignment

- 116× speedup, 37× less power than Minimap2 (state-of-the-art SW)
- 111× speedup, 33× less power than BWA-MEM (state-of-the-art SW)
- 3.9× better throughput, 2.7× less power than Darwin (state-of-the-art HW)
- 1.9× better throughput, 82% less logic power than GenAx (state-of-the-art HW)

(2) Pre-Alignment Filtering

- 3.7× speedup, 1.7× less power than Shouji (state-of-the-art HW)

(3) Edit Distance Calculation

- 22–12501× speedup, 548–582× less power than Edlib (state-of-the-art SW)
- 9.3–400× speedup, 67× less power than ASAP (state-of-the-art HW)
More on GenASM

Read Mapping

- Reference genome
  - Indexing
    - Hash-table based index
- Reads
  - Seeding
    - Potential mapping locations
- Reference segment
  - Pre-Alignment Filtering
    - Remaining potential mapping locations
- Query read
  - Read Alignment

Livestream - P&S Accelerating Genome Analysis with FPGAs, GPUs, and New Execution Paradigms (Fall 2021)
Comp. Architecture - Lecture 9a: GenASM: Approx. String Matching Accelerator (ETH Zürich, Fall 2020)

https://www.youtube.com/watch?v=XoLpzmN-Pas
Scrooge: Faster Approximate String Matching

Joël Lindegger, Damla Senol Cali, Mohammed Alser, Juan Gómez-Luna, Nika Mansouri Ghiasi, and Onur Mutlu,
"Scrooge: A Fast and Memory-Frugal Genomic Sequence Aligner for CPUs, GPUs, and ASICs"
[Online link at Bioinformatics Journal]
[arXiv preprint]
[Scrooge Source Code]

Scrooge: A Fast and Memory-Frugal Genomic Sequence Aligner for CPUs, GPUs, and ASICs

Joël Lindegger§
Juan Gómez-Luna§

Damla Senol Cali†
Nika Mansouri Ghiasi§

Mohammed Alser§
Onur Mutlu§

§ETH Zürich
†Bionano Genomics

https://github.com/cmu-safari/scrooge

Our Goals

Build a practical and efficient implementation of the GenASM algorithm for multiple computing platforms

Compete with state-of-the-art pairwise sequence aligners like Edlib, KSW2, and BiWFA
Three novel algorithmic improvements which address inefficiencies in the GenASM algorithm

Efficient open-source implementations for CPUs and GPUs

Key Results
Scrooge consistently outperforms GenASM
- 2.1x speedup over GenASM on CPU
- 5.9x speedup over GenASM on GPU
- 3.6x better area efficiency and 3.6x less power than GenASM as an ASIC

Scrooge consistently outperforms state-of-the-art CPU and GPU baselines, including KSW2, Edlib, and BiWFA
Scrooge: A fast and memory-frugal genomic sequence aligner for CPUs, GPUs and ASICs

Scrooge is a fast pairwise genomic sequence aligner. It efficiently aligns short and long genomic sequence pairs on multiple computing platforms. It is based on the GenASM algorithm (Senol Cali+, 2020), and adds multiple algorithmic improvements that significantly improve the throughput and resource efficiency for CPUs, GPUs and ASICs. For long reads, the CPU version of Scrooge achieves a 20.1x, 1.7x, and 2.1x speedup over KSW2, Edlib, and a CPU implementation of GenASM, respectively. The GPU version of Scrooge achieves a 4.0x 80.4x, 6.8x, 12.6x and 5.9x speedup over the CPU version of Scrooge, KSW2, Edlib, Darwin-GPU, and a GPU implementation of GenASM, respectively. We estimate an ASIC implementation of Scrooge to use 3.6x less chip area and 2.1x less power than a GenASM ASIC while maintaining the same throughput.

This repository contains Scrooge’s CPU and GPU implementations, and several evaluation scripts. We describe Scrooge in our paper on arXiv and in Bioinformatics.
Scrooge Talk Video

Long Read Throughput

For long reads, **Scrooge** outperforms GenASM by 2.1x on CPU and 5.9x on GPU.
Accelerating Sequence-to-Graph Mapping

- Damla Senol Cali, Konstantinos Kanellopoulos, Joel Lindegger, Zulal Bingol, Gurpreet S. Kalsi, Ziyi Zuo, Can Firtina, Meryem Banu Cavlak, Jeremie Kim, Nika MansouriGhiasi, 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"
[arXiv version]

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

Damla Senol Cali\textsuperscript{1} Konstantinos Kanellopoulos\textsuperscript{2} Joël Lindegger\textsuperscript{2} Zülal Bingöl\textsuperscript{3}
Gurpreet S. Kalsi\textsuperscript{4} Ziyi Zuo\textsuperscript{5} Can Firtina\textsuperscript{2} Meryem Banu Cavlak\textsuperscript{2} Jeremie Kim\textsuperscript{2}
Nika Mansouri Ghiasi\textsuperscript{2} Gagandeep Singh\textsuperscript{2} Juan Gómez-Luna\textsuperscript{2} Nour Almadhoun Alserr\textsuperscript{2}
Mohammed Alser\textsuperscript{2} Sreenivas Subramoney\textsuperscript{4} Can Alkan\textsuperscript{3} Saugata Ghose\textsuperscript{6} Onur Mutlu\textsuperscript{2}

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

Genome Sequence Analysis

- 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:

Graph-based Reference:

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
Sequence-to-Graph Mapping Pipeline

**Pre-Processing Steps (Offline)**

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

0.2 Indexing
- (index the nodes of the graph)
- Hash-table-based index (of graph nodes)

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

1 Seeding
- (query the index & find the seed matches)
- Candidate mapping locations (subgraphs)

2 Filtering/Chaining/Clustering
- (filter out dissimilar query read and subgraph pairs)
- Remaining candidate mapping locations (subgraphs)

3 S2G Alignment
- (perform distance/score calculation & traceback)
- Optimal alignment between read & subgraph

Linear reference genome
- Known genetic variations

Reads from sequenced genome

Genome Graph Construction

Indexing

Seeding

Filtering/Chaining/Clustering

S2G Alignment

SAFARI
SeGraM Hardware Design

MinSeed: first hardware accelerator for Minimizer-based Seeding

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

SAFARI
SeGraM Talk Video

SeGraM: A Universal HW Accelerator for Genomic Sequence-to-Graph Mapping - Damla Senol Cali (ISCA)

https://www.youtube.com/watch?v=qyjqYoyDP9s
SeGraM: A Universal Hardware Accelerator for Genomic Sequence-to-Graph and Sequence-to-Sequence Mapping

Damla Senol Cali\textsuperscript{1} Konstantinos Kanellopoulos\textsuperscript{2} Joël Lindegger\textsuperscript{2} Zülal Bingöl\textsuperscript{3}
Gurpreet S. Kalsi\textsuperscript{4} Ziyi Zuo\textsuperscript{5} Can Firtina\textsuperscript{2} Meryem Banu Cavlak\textsuperscript{2} Jeremie Kim\textsuperscript{2}
Nika Mansouri Ghiasi\textsuperscript{2} Gagandeep Singh\textsuperscript{2} Juan Gómez-Luna\textsuperscript{2} Nour Almadhoun Alserr\textsuperscript{2}
Mohammed Alser\textsuperscript{2} Sreenivas Subramoney\textsuperscript{4} Can Alkan\textsuperscript{3} Saugata Ghose\textsuperscript{6} Onur Mutlu\textsuperscript{2}

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

SAFARI  \url{https://arxiv.org/pdf/2205.05883.pdf}
In-Storage Genome Filtering [ASPLOS 2022]

- Nika Mansouri Ghiasi, Jisung Park, Harun Mustafa, Jeremie Kim, Ataberk Olgun, Arvid Gollwitzer, Damla Senol Cali, Can Firtina, Haiyu Mao, Nour Almadhoun Alserr, Rachata Ausavarungnirun, Nandita Vijaykumar, Mohammed Alser, and Onur Mutlu,

"GenStore: A High-Performance and Energy-Efficient In-Storage Computing System for Genome Sequence Analysis"


[Lightning Talk Slides (pptx) (pdf)]
[Lightning Talk Video (90 seconds)]

GenStore: A High-Performance In-Storage Processing System for Genome Sequence Analysis

Nika Mansouri Ghiasi¹ Jisung Park¹ Harun Mustafa¹ Jeremie Kim¹ Ataberk Olgun¹ Arvid Gollwitzer¹ Damla Senol Cali² Can Firtina¹ Haiyu Mao¹ Nour Almadhoun Alserr¹ Rachata Ausavarungnirun³ Nandita Vijaykumar⁴ Mohammed Alser¹ Onur Mutlu¹

¹ETH Zürich  ²Bionano Genomics  ³KMUTNB  ⁴University of Toronto
Genome Sequence Analysis

- Data Movement from Storage
- Storage System
- Main Memory
- Cache
- Computation Unit (CPU or Accelerator)

- Computation overhead
- Data movement overhead
Accelerating Genome Sequence Analysis

- Heuristics
- Accelerators
- Filters

- Storage System
- Main Memory
- Cache
- Computation Unit (CPU or Accelerator)

✓ Computation overhead

✗ Data movement overhead

SAFARI
Key Idea

Filter reads that do not require alignment inside the storage system

Filtered Reads

Exactly-matching reads
Do not need expensive approximate string matching during alignment

Non-matching reads
Do not have potential matching locations and can skip alignment
Filtering Opportunities

- Sequencing machines produce one of two kinds of reads
  - **Short reads**: highly accurate and short
  - **Long reads**: less accurate and long

**Reads that do not require the expensive alignment step:**

**Exactly-matching reads**
- Do not need expensive approximate string matching during alignment
  - Low sequencing error rates *(short reads)* combined with
  - Low genetic variation

**Non-matching reads**
- Do not have potential matching locations, so they skip alignment
  - High sequencing error rates *(long reads)* or
  - High genetic variation *(short or long reads)*
Challenges

Filter reads that do not require alignment inside the storage system

- Storage System
- Main Memory
- Cache
- Computation Unit (CPU or Accelerator)

Filtered Reads

Read mapping workloads can exhibit different behavior

There are limited hardware resources in the storage system
GenStore

Filter reads that do not require alignment inside the storage system

GenStore-Enabled Storage System

Main Memory

Cache

Computation Unit (CPU or Accelerator)

✓ Computation overhead

✓ Data movement overhead

GenStore provides significant speedup (1.4x - 33.6x) and energy reduction (3.9x – 29.2x) at low cost
In-Storage Genome Filtering [ASPLOS 2022]

- Nika Mansouri Ghiasi, Jisung Park, Harun Mustafa, Jeremie Kim, Ataberk Olgun, Arvid Gollwitzer, Damla Senol Cali, Can Firtina, Haiyu Mao, Nour Almadhoun Alserr, Rachata Ausavarungnirun, Nandita Vijaykumar, Mohammed Alser, and Onur Mutlu,

"GenStore: A High-Performance and Energy-Efficient In-Storage Computing System for Genome Sequence Analysis"


[Lightning Talk Slides (pptx) (pdf)]
[Lightning Talk Video (90 seconds)]

---

GenStore: A High-Performance In-Storage Processing System for Genome Sequence Analysis

Nika Mansouri Ghiasi¹  Jisung Park¹  Harun Mustafa¹  Jeremie Kim¹  Ataberk Olgun¹
Arvid Gollwitzer¹  Damla Senol Cali²  Can Firtina¹  Haiyu Mao¹  Nour Almadhoun Alserr¹
Rachata Ausavarungnirun³  Nandita Vijaykumar⁴  Mohammed Alser¹  Onur Mutlu¹

¹ETH Zürich  ²Bionano Genomics  ³KMUTNB  ⁴University of Toronto

SAFARI
Accelerating Basecalling + Read Mapping via PIM

- Haiyu Mao, Mohammed Alser, Mohammad Sadrosadati, Can Firtina, Akanksha Baranwal, Damla Senol Cali, Aditya Manglik, Nour Almadhoun Alserr, and Onur Mutlu, "GenPIP: In-Memory Acceleration of Genome Analysis via Tight Integration of Basecalling and Read Mapping". Proceedings of the 55th International Symposium on Microarchitecture (MICRO), Chicago, IL, USA, October 2022.
- [Slides (pptx) (pdf)]
- [Longer Lecture Slides (pptx) (pdf)]
- [Lecture Video (25 minutes)]
- [arXiv version]

GenPIP: In-Memory Acceleration of Genome Analysis via Tight Integration of Basecalling and Read Mapping

Haiyu Mao\textsuperscript{1} Mohammed Alser\textsuperscript{1} Mohammad Sadrosadati\textsuperscript{1} Can Firtina\textsuperscript{1} Akanksha Baranwal\textsuperscript{1} Damla Senol Cali\textsuperscript{2} Aditya Manglik\textsuperscript{1} Nour Almadhoun Alserr\textsuperscript{1} Onur Mutlu\textsuperscript{1}

\textsuperscript{1}ETH Zürich \hspace{2cm} \textsuperscript{2}Bionano Genomics

Genome Analysis Pipeline

1. Basecalling
   - Storage
   - Compute
   - Storage

2. Read Quality Control
   - Low-quality
   - Storage
   - Compute

3. Read Mapping
   - Reference genome
   - Compute
   - Storage
   - Compute
   - Store mapping results
   - Mapped
   - Unmapped
Limitation 1: Large Data Movement

- Using a human dataset in [NC’19] as an example:

Raw Signals → Basecalling → Reads → Read quality control → High-quality reads → Read mapping → Mapped reads

3913 GB → 546 GB → 437 GB → 382 GB

Large data movement between genome analysis steps

Limitation 2: Wasted Computation

Using a human dataset in [NC’19] as an example:

- **100%** Raw Signals → Basecalling → Reads
- **79.5%** Read quality control → High-quality reads → Read mapping → Mapped reads
- **20.5%** Low-quality reads
- **10%** Unmapped reads

A considerable amount of computation on **useless data** due to:
- Low-quality reads
- Unmapped reads

Overview: Two Limitations

Multiple steps in genome analysis

- Large data movement between multiple steps
- A lot of wasted computation done on data that is later discovered to be useless
Goal and Opportunities

**Goal:** Efficiently accelerate the entire genome analysis pipeline while minimizing data movement and useless computation.

- We perform a study to quantify potential performance benefits.
  - Results are normalized to the performance of GPU.

![Normalized Speedup Chart](chart.png)

- NVM-based PIM accelerators for separate basecalling and read mapping: 2.7x
- No data movement between the accelerators of analysis steps: 6.1x
- No data movement and no useless reads (ideal case): 9x
Overview: GenPIP

- **GenPIP:** A fast and energy-efficient *in-memory* acceleration system for the Genome analysis PIPeline via *tight integration of genome analysis steps*

- **GenPIP** has two key techniques
  - **Chunk-based pipeline (CP)**
    - Provides fine-grained collaboration of genome analysis steps
  - **Early rejection (ER)**
    - Timely stops the execution on useless data by predicting which reads will not be useful

- **GenPIP** outperforms state-of-the-art software & hardware solutions using **CPU**, **GPU**, and optimistic **PIM** by **41.6x**, **8.4x**, and **1.4x**, respectively.
Innovations Require Change

- CP processes reads **at the granularity of a chunk** instead of the complete read sequence, increasing parallelism and resource utilization by overlapping the execution of different steps.

GenPIP provides 41.6x and 8.4x speedup and 32.8x and 20.8x energy reduction compared to CPU and GPU state-of-the-art solutions.
GenPIP Implementation

Raw signals from the sequencing machine

In-memory Basecaller [Helix, PACT’20]

Base quality score

PIM-CQS
PIM chunk quality score calculation

Basecalling Module

Signal chunk

Basecalled chunk

eDRAM

Average Calculator

Quality score

ER Controller

GenPIP Controller

Chunk

Chunk mapping score

Read Mapping Controller

In-memory Read Mapping [PARC, ASPDAC’20]
+ Our design

Read mapping result

To storage

Key Results – Performance

GenPIP provides 41.6x, 8.4x, and 1.4x speedup over CPU, GPU, and optimistic PIM

Both CP and ER are critical to the speedup
Key Results – Energy Efficiency

GenPIP provides 32.8x, 20.8x, and 1.37x energy savings over CPU, GPU, and optimistic PIM.

ER is especially critical to the energy efficiency.
Accelerating Basecalling + Read Mapping via PIM

- Haiyu Mao, Mohammed Alser, Mohammad Sadrosadati, Can Firtina, Akanksha Baranwal, Damla Senol Cali, Aditya Manglik, Nour Almadhoun Alserr, and Onur Mutlu, "GenPIP: In-Memory Acceleration of Genome Analysis via Tight Integration of Basecalling and Read Mapping". Proceedings of the 55th International Symposium on Microarchitecture (MICRO), Chicago, IL, USA, October 2022.
  - [Slides (pptx) (pdf)]
  - [Longer Lecture Slides (pptx) (pdf)]
  - [Lecture Video (25 minutes)]
  - [arXiv version]

GenPIP: In-Memory Acceleration of Genome Analysis via Tight Integration of Basecalling and Read Mapping

Haiyu Mao¹, Mohammed Alser², Mohammad Sadrosadati³, Can Firtina¹, Akanksha Baranwal¹, Damla Senol Cali², Aditya Manglik¹, Nour Almadhoun Alserr¹, Onur Mutlu¹

¹ETH Zürich  ²Bionano Genomics
Near-memory Pre-alignment Filtering

Gagandeep Singh, Mohammed Alser, Damla Senol Cali, Dionysios Diamantopoulos, Juan Gomez-Luna, Henk Corporaal, Onur Mutlu,

"FPGA-Based Near-Memory Acceleration of Modern Data-Intensive Applications"

[Source Code]
Demeter: A Fast and Energy-Efficient Food Profiler Using Hyperdimensional Computing in Memory

Taha Shahroodi\textsuperscript{1}, Mahdi Zahedi\textsuperscript{1}, Can Firtina\textsuperscript{2}, Mohammed Alser\textsuperscript{1}, Stephan Wong\textsuperscript{1}, Onur Mutlu\textsuperscript{1}, Said Hamdioui\textsuperscript{1}, (Senior Member, IEEE), Onur Mutlu\textsuperscript{1}, (Senior Member, IEEE)

\textsuperscript{1}Q\&CE Department, EEMCS Faculty, Delft University of Technology (TU Delft), 2628 CD Delft, The Netherlands
\textsuperscript{2}SAFARI Research Group, D-ITET, ETH Zürich, 8092 Zürich, Switzerland
AIM (PIM Sequence Alignment Framework)

Safaa Diab, Amir Nassereldine, Mohammed Alser, Juan Gómez-Luna, Onur Mutlu, Izzat El Hajj
“A Framework for High-throughput Sequence Alignment using Real Processing-in-Memory Systems”
arXiv, 2022
[Source code]

A Framework for High-throughput Sequence Alignment using Real Processing-in-Memory Systems

Safaa Diab\textsuperscript{1}, Amir Nassereldine\textsuperscript{1}, Mohammed Alser\textsuperscript{2}, Juan Gómez Luna\textsuperscript{2}, Onur Mutlu\textsuperscript{2}, Izzat El Hajj\textsuperscript{1}

\textsuperscript{1}American University of Beirut, Lebanon \textsuperscript{2}ETH Zürich, Switzerland
What else can be done?
What if we got a new version of the reference genome?

Reference genome

Reads

Revisiting the Puzzle

Reference Genome Bias

“African pan-genome contains ~10% more DNA bases than the current human reference genome”
Time to Change the Reference Genome

"Switching to a consensus reference would offer important advantages over the continued use of the current reference with few disadvantages"
New Applications: Frequent Reference Updates

- Jeremie S. Kim, Can Firtina, M. Banu Cavlak, Damla Senol Cali, Nastaran Hajinazar, Mohammed Alser, Can Alkan, and Onur Mutlu, "AirLift: A Fast and Comprehensive Technique for Remapping Alignments between Reference Genomes"
  Proceedings of the 21st Asia Pacific Bioinformatics Conference (APBC), Changsha, China, April 2023.
  [AirLift Source Code]
  [arxiv.org Version (pdf)]
  [Talk Video at BIO-Arch 2023 Workshop]

METHOD

AirLift: A Fast and Comprehensive Technique for Remapping Alignments between Reference Genomes

Jeremie S. Kim¹⁺, Can Firtina¹⁺, Meryem Banu Cavlak², Damla Senol Cali³, Nastaran Hajinazar¹,⁴, Mohammed Alser¹, Can Alkan² and Onur Mutlu¹,²,³*
Key observation: Reference genomes are updated frequently. Repeating read mapping is a computationally expensive workload.

Key idea: Update the mapping results of only affected reads depending on how a region in the old reference relates to another region in the new reference.

Key results:
- reduces number of reads that needs to be re-mapped to new reference by up to 99%
- reduces overall runtime to re-map reads by 6.94x, 208x, and 16.4x for large (human), medium (C. elegans), and small (yeast) reference genomes
Clustering the Reference Genome Regions

Fig. 2. Reference Genome Regions.
New Applications: Frequent Reference Updates

  - [AirLift Source Code]
  - [arxiv.org Version (pdf)]
  - [Talk Video at BIO-Arch 2023 Workshop]

**METHOD**

AirLift: A Fast and Comprehensive Technique for Remapping Alignments between Reference Genomes

Jeremie S. Kim\(^1\), Can Firtina\(^1\), Meryem Banu Cavlak\(^2\), Damla Senol Cali\(^3\), Nastaran Hajinazar\(^1,4\), Mohammed Alser\(^1\), Can Alkan\(^2\) and Onur Mutlu\(^1,2,3\)*
Nanopore sequencing technology and tools for genome assembly: computational analysis of the current state, bottlenecks and future directions

Damla Senol Cali+, Jeremie S Kim, Saugata Ghose, Can Alkan, Onur Mutlu

*Briefings in Bioinformatics*, bby017, https://doi.org/10.1093/bib/bby017

**Published:** 02 April 2018  **Article history** ▼


[Open arxiv.org version]  [Slides (pptx) (pdf)]  [Talk Video at AACBB 2019]
RawHash: enabling fast and accurate real-time analysis of raw nanopore signals for large genomes

Can Firtina, Nika Mansouri Ghiasi, Joel Lindegger, Gagandeep Singh, Meryem Banu Cavlak, Haiyu Mao, and Onur Mutlu,

"RawHash: Enabling Fast and Accurate Real-Time Analysis of Raw Nanopore Signals for Large Genomes"
Proceedings of the 31st Annual Conference on Intelligent Systems for Molecular Biology (ISMB) and the 22nd European Conference on Computational Biology (ECCB), Jul 2023

Bioinformatics, 2023, 39, i297–i307
https://doi.org/10.1093/bioinformatics/btad272
ISMB/ECCB 2023

1Department of Information Technology and Electrical Engineering, ETH Zurich, 8092 Zurich, Switzerland
*Corresponding author. Department of Information Technology and Electrical Engineering, ETH Zurich, Gloriastrasse 35, 8092 Zurich, Switzerland.
E-mail: firtinac@ethz.ch (C.F.), omutlu@ethz.ch (O.M.)
Real-Time Analysis with Nanopore Sequencing

Raw Signals: Ionic current measurements generated at a certain throughput

Real-Time Analysis: Analyzing all raw signals by matching the throughput

Real-Time Decisions: Stopping sequencing early based on real-time analysis
Benefits of Real-Time Genome Analysis

- **Reducing latency** by overlapping the sequencing and analysis steps
- **Reducing sequencing time and cost** by stopping sequencing early

Sequencing is stopped early with a real-time decision.
Challenges in Real-Time Genome Analysis

- **Rapid analysis** to match the nanopore sequencer throughput
- **Timely decisions** to stop sequencing as early as possible
- **Accurate analysis** from noisy raw signal data
- **Power-efficient** computation for scalability and portability
RawHash – Key Idea

**Key Observation:** **Identical** nucleotides generate **similar** raw signals

**Challenge #1:** Generating the **same** hash value for **similar enough** signals

**Challenge #2:** **Accurately** finding similar regions **as few as possible**
Events (K-mers) in Raw Nanopore Signals

- **Event**: A segment of the raw signal
  - Corresponds to a particular k-mer

- **Event detection** finds these segments to identify k-mers
  - Start and end positions are marked by abrupt signal changes
  - Statistical methods identify these abrupt changes
  - **Event value**: average of signals **within an event**

Event Value (picoampere)

$k$ many nucleotides
Quantizing the Event Values

• **Observation:** Slight differences in raw signals from identical k-mers
  - **Challenge:** Direct event value matching is not feasible and accurate

• **Key Idea:** Quantize the event values
  - Enables assigning identical quantized values to similar event values

![Diagram showing quantization process](image)
Hashing for Fast Similarity Search

• Each event usually represents a very small k-mer (6 to 9 characters)
  - **Challenge:** Short k-mers are likely to appear in many locations

• **Key Idea:** Create longer k-mers from many consecutive events

• **Key Benefit:** Directly match hash values to quickly identify similarities

Consecutive k-mers

- CTATTA
- TATTAC
- ATTACC

Consecutive events

- CTATTA: -0.09 → Quantize → 11001
- TATTAC: 1.15 → Quantize → 00110
- ATTACC: 1.11 → Quantize → 00101

Hash value of consecutive events: 0x400D70A4

Hash

1100100110 → 01001

Pack
Real-Time Mapping using Hash-based Indexing

**Indexing (Offline)**

Reference Genome

...GCTATTACCTTAATGTG...

Reference-to-Event Conversion

Quantization

Hashing

Store 0x01

Hash Table

Matching Positions

**Mapping (Real-time)**

Raw Nanopore Signal

Signal-to-Event Conversion

Quantization

Hashing

Query 0x01

Chaining & Mapping

Yes: Process the next chunk

Continue Mapping?

No: Stop mapping

Read Until or Run Until
More on RawHash

- Can Firtina, Nika Mansouri Ghiasi, Joel Lindegger, Gagandeep Singh, Meryem Banu Cavlak, Haiyu Mao, and Onur Mutlu,

"RawHash: Enabling Fast and Accurate Real-Time Analysis of Raw Nanopore Signals for Large Genomes"

Proceedings of the 31st Annual Conference on Intelligent Systems for Molecular Biology (ISMB) and the 22nd European Conference on Computational Biology (ECCB), Jul 2023

[Bioinformatics Journal version]
[Slides (pptx) (pdf)]
[RawHash Source Code]

Bioinformatics, 2023, 39, i297–i307
https://doi.org/10.1093/bioinformatics/btad272
ISMB/ECCB 2023

RawHash: enabling fast and accurate real-time analysis of raw nanopore signals for large genomes

Can Firtina 1,* , Nika Mansouri Ghiasi 1 , Joel Lindegger 1 , Gagandeep Singh 1 , Meryem Banu Cavlak 1 , Haiyu Mao 1 , Onur Mutlu 1,*

1Department of Information Technology and Electrical Engineering, ETH Zurich, 8092 Zurich, Switzerland
*Corresponding author. Department of Information Technology and Electrical Engineering, ETH Zurich, Gloriastrasse 35, 8092 Zurich, Switzerland.
E-mail: firtinac@ethz.ch (C.F.), omutlu@ethz.ch (O.M.)
Fast and Accurate Real-Time Genome Analysis

- Can Firtina, Melina Soysal, Joel Lindegger, and Onur Mutlu,

"RawHash2: Accurate and Fast Mapping of Raw Nanopore Signals using a Hash-based Seeding Mechanism"

Preprint on arxiv, September 2023.
[arXiv version]
[RawHash2 Source Code]

RawHash2: Accurate and Fast Mapping of Raw Nanopore Signals using a Hash-based Seeding Mechanism

Can Firtina  Melina Soysal  Joel Lindegger  Onur Mutlu
ETH Zürich
What else can be done?
Fast and Accurate Real-Time Genome Analysis

- Joel Lindegger, Can Firtina, Nika Mansouri Ghiasi, Mohammad Sadrosadati, Mohammed Alser, and Onur Mutlu,

"RawAlign: Accurate, Fast, and Scalable Raw Nanopore Signal Mapping via Combining Seeding and Alignment"

Preprint on arxiv, October 2023.

[arXiv version]
[RawAlign Source Code]

RawAlign: Accurate, Fast, and Scalable Raw Nanopore Signal Mapping via Combining Seeding and Alignment

Joël Lindegger$§
Mohammad Sadrosadati$§
Can Firtina$§
Mohammed Alser$§
Nika Mansouri Ghiasi$§
Onur Mutlu$§

$ETH Zürich
Dynamic Time Warping (DTW)

- Aligning a pair of signals using DTW and Subsequence DTW (sDTW)

Standard DTW (Global Alignment)
Fast and Accurate Real-Time Genome Analysis

- Joel Lindegger, Can Firtina, Nika Mansouri Ghiasi, Mohammad Sadrosadati, Mohammed Alser, and Onur Mutlu,
  "RawAlign: Accurate, Fast, and Scalable Raw Nanopore Signal Mapping via Combining Seeding and Alignment"
  Preprint on arxiv, October 2023.
  [arXiv version]
  [RawAlign Source Code]

RawAlign: Accurate, Fast, and Scalable Raw Nanopore Signal Mapping via Combining Seeding and Alignment

Joël Lindegger$  Can Firtina$  Nika Mansouri Ghiasi$
Mohammad Sadrosadati$  Mohammed Alser$  Onur Mutlu$

$ETH Zürich
New Frontiers: Pre-Basecalling Filters

- M. Banu Cavlak, Gagandeep Singh, Mohammed Alser, Can Firtina, Joel Lindegger, Mohammad Sadrosadati, Nika Mansouri Ghiasi, Can Alkan, and Onur Mutlu, "TargetCall: Eliminating the Wasted Computation in Basecalling via Pre-Basecalling Filtering" Proceedings of the 21st Asia Pacific Bioinformatics Conference (APBC), Changsha, China, April 2023.
  [TargetCall Source Code]
  [arxiv.org Version]
  [Talk Video at BIO-Arch 2023 Workshop]

TargetCall: Eliminating the Wasted Computation in Basecalling via Pre-Basecalling Filtering

Meryem Banu Cavlak¹  Gagandeep Singh¹  Mohammed Alser¹  Can Firtina¹  Joël Lindegger¹
Mohammad Sadrosadati¹  Nika Mansouri Ghiasi¹  Can Alkan²  Onur Mutlu¹

¹ETH Zürich  ²Bilkent University

Agenda for Today

- Introduction to Genomics
  - Today: The ways we analyze genomes
  - Intelligent genome analysis

- Step-by-step Genome Analysis
  - Sequencing
  - Read Mapping
  - Variant Calling

- Algorithmic & Hardware Acceleration

- Future Opportunities: New Technologies & Applications
Adoption of hardware accelerators in genome analysis
Dream
and, they will come

Computing landscape is very different from 10-20 years ago
Recall Our Dream (from 2007)

- An embedded device that can perform comprehensive genome analysis in real time (within a minute)

- Still a long ways to go
  - Energy efficiency
  - Performance (latency)
  - Security & privacy
  - **Huge memory bottleneck**
Things Are Happening In Industry
Illumina DRAGEN Bio-IT Platform (2018)

- Processes whole genome at 30x coverage in ~25 minutes with hardware support for data compression

[Link to product page](emea.illumina.com/products/by-type/informatics-products/dragen-bio-it-platform.html)
NVIDIA Clara Parabricks (2020)

A University of Michigan startup in 2018 joined NVIDIA in 2020

![GPU board(s)](image)

**PERFORMANCE COMPARISON**
Germline End-to-End Secondary Analysis

<table>
<thead>
<tr>
<th></th>
<th>CPU/GATK</th>
<th>8X T4</th>
<th>8X V100</th>
<th>8X A100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>1,200 min</td>
<td>52 min</td>
<td>35 min</td>
<td>23 min</td>
</tr>
</tbody>
</table>
NVIDIA Hopper GPU Architecture Accelerates Dynamic Programming Up to 40x Using New DPX Instructions

Dynamic programming algorithms are used in healthcare, robotics, quantum computing, data science and more.

March 22, 2022 by DION HARRIS

https://blogs.nvidia.com/blog/2022/03/22/nvidia-hopper-accelerates-dynamic-programming-using-dpx-instructions/
We are accelerating the transformation in how we analyze the human genome!

Bionano & NVIDIA:
Accelerating Analysis for Fast Time to Results

- Technological solution to support higher throughput
- New high-performance algorithms from Bionano
- Powered by NVIDIA RTX™ 6000 Ada Generation GPUs
- Analysis of highly complex cancer whole genomes in less than 2 hours
- Workflow tailored for a small lab and IT footprint
A Bright Future for Intelligent Genome Analysis

Mohammed Alser, Zülal Bingöl, Damla Senol Cali, Jeremie Kim, Saugata Ghose, Can Alkan, Onur Mutlu


Accelerating Genome Analysis: A Primer on an Ongoing Journey
DOI Bookmark: 10.1109/MM.2020.3013728

FPGA-Based Near-Memory Acceleration of Modern Data-Intensive Applications
DOI Bookmark: 10.1109/MM.2021.3088396

MinION from ONT

SmidgION from ONT
Computing is Still Bottlenecked by Data Movement
Adoption Challenges of Hardware Accelerators

- Accelerate the entire read mapping process rather than its individual steps (*Amdahl’s law*)

- Reduce the high amount of data movement
  - Working directly on compressed data
  - Filter out unlikely-reused data at the very first component of the compute system

- Develop flexible hardware architectures that do NOT conservatively limit the range of supported parameter values at design time

- Adapt existing genomic data formats for hardware accelerators or develop more efficient file formats
Adoption Challenges of Hardware Accelerators

- Maintaining the same (or better) accuracy/sensitivity of the output results of the software version
  - Using heuristic algorithms to gain speedup!

- High hardware cost

- Long development life-cycle for FPGA platforms
Did we Achieve Our Goal?

- **Fast** genome analysis in mere seconds using **limited computational resources** (i.e., personal computer or small hardware).

1997

2015
Open Questions

How and where to enable fast, accurate, cheap, privacy-preserving, and exabyte scale analysis of genomic data?
Pushing Towards New Architectures

Modern systems

FPGAs

Heterogeneous Processors and Accelerators

(General Purpose) GPUs

Hybrid Main Memory

Persistent Memory/Storage

Sequencing Machine

Modern systems?
Cerebras’s Wafer Scale Engine (2019)

- The largest ML accelerator chip (2019)
- 400,000 cores

**Cerebras WSE**
- 1.2 Trillion transistors
- 46,225 mm²

**Largest GPU**
- 21.1 Billion transistors
- 815 mm²

https://www.cerebras.net/cerebras-wafer-scale-engine-why-we-need-big-chips-for-deep-learning/
Cerebras’s Wafer Scale Engine (2021)

- The largest ML accelerator chip (2021)
- 850,000 cores

Cerebras WSE-2
2.6 Trillion transistors
46,225 mm²

Largest GPU
54.2 Billion transistors
826 mm²
NVIDIA Ampere GA100

https://www.anandtech.com/show/14758/hot-chips-31-live-blogs-cerebras-wafer-scale-deep-learning
https://www.cerebras.net/cerebras-wafer-scale-engine-why-we-need-big-chips-for-deep-learning/
NextSeq 2000 with Analysis Capability

NextSeq 1000/2000 Integrates DRAGEN Bio-IT Platform On-Board

DRAGEN Bio-IT platform:
- Fast
- Accurate
- Industry standard pipelines
- For both novice and expert users

Pipelines available on-board:
- DRAGEN Enrichment pipeline
- DRAGEN RNA pipeline
- DRAGEN Germline
- DRAGEN Single Cell RNA
- Generate FASTQ via BCL Convert
- Additional pipelines available in BaseSpace Sequence Hub

For Research Use Only.
Not for use in diagnostic procedures.
NVIDIA is claiming a 7x improvement in dynamic programming algorithm (DPX instructions) performance on a single H100 versus naïve execution on an A100.

The vision of BioPIM is the realization of cheap, ultra-fast and ultra-low energy mobile genomics that eliminates the current dependence of sequence analysis on large and power-hungry computing clusters/data-centers.
UPMEM Processing-in-DRAM Engine (2019)

- Processing in DRAM Engine
- Includes **standard DIMM modules**, with a **large number of DPU processors** combined with DRAM chips.

- Replaces **standard** DIMMs
  - DDR4 R-DIMM modules
    - 8GB+128 DPUs (16 PIM chips)
    - Standard 2x-nm DRAM process
  - **Large amounts of** compute & memory bandwidth


Onur Mutlu, *Computer Architecture Lecture 2b*, Fall 2019, ETH Zurich
Conclusion

- **System design for bioinformatics** is a critical problem
  - It has large scientific, medical, societal, personal implications

- This lecture is about accelerating **a key step in bioinformatics**: 
  genome sequence analysis
  - Especially techniques for **read mapping**

- We covered various **recent ideas to accelerate genome analysis**

- **Many future opportunities exist**
  - Especially with new sequencing technologies
  - Especially with new applications and use cases
Recommended Readings

Mohammed Alser, Zülal Bingöl, Damla Senol Cali, Jeremie Kim, Saugata Ghose, Can Alkan, Onur Mutlu

“Accelerating Genome Analysis: A Primer on an Ongoing Journey”

A Few Overview Readings (II)

Gagandeep Singh, Mohammed Alser, Damla Senol Cali, Dionysios Diamantopoulos, Juan Gomez-Luna, Henk Corporaal, Onur Mutlu,

“FPGA-Based Near-Memory Acceleration of Modern Data-Intensive Applications”
[Source Code]

FPGA-Based Near-Memory Acceleration of Modern Data-Intensive Applications
DOI Bookmark: 10.1109/MM.2021.3088396

Authors
Gagandeep Singh, ETH Zürich, Zürich, Switzerland
Mohammed Alser, ETH Zürich, Zürich, Switzerland
Damla Senol Cali, Carnegie Mellon University, Pittsburgh, PA, USA
Dionysios Diamantopoulos, Zürich Lab, IBM Research Europe, Rüschlikon, Switzerland
Juan Gomez-Luna, ETH Zürich, Zürich, Switzerland
Henk Corporaal, Eindhoven University of Technology, Eindhoven, The Netherlands
Onur Mutlu, ETH Zürich, Zürich, Switzerland

Source Code
Mohammed Alser, Joel Lindegger, Can Firtina, Nour Almadhoun, Haiyu Mao, Gagandeep Singh, Juan Gomez-Luna, Onur Mutlu
“From Molecules to Genomic Variations: Intelligent Algorithms and Architectures for Intelligent Genome Analysis”
Computational and Structural Biotechnology Journal, 2022
[Source code]

Review

From molecules to genomic variations: Accelerating genome analysis via intelligent algorithms and architectures

Mohammed Alser *, Joel Lindegger, Can Firtina, Nour Almadhoun, Haiyu Mao, Gagandeep Singh, Juan Gomez-Luna, Onur Mutlu *

ETH Zurich, Gloriastrasse 35, 8092 Zürich, Switzerland

Onur Mutlu and Can Firtina,
"Accelerating Genome Analysis via Algorithm-Architecture Co-Design"
[Slides (pptx) (pdf)]
[Talk Video (38 minutes, including Q&A)]
[Related Invited Paper]
[arXiv version]

Accelerating Genome Analysis via Algorithm-Architecture Co-Design

Onur Mutlu  Can Firtina

ETH Zürich

Sequence Alignment on Real PIM Systems

- Safaa Diab, Amir Nassereldine, Mohammed Alser, Juan Gómez Luna, Onur Mutlu, and Izzat El Hajj,

"A Framework for High-throughput Sequence Alignment using Real Processing-in-Memory Systems"


[Online link at Bioinformatics Journal]

[arXiv preprint]

[AiM Source Code]
April 14, 2023

BIO-Arch: Workshop on Hardware Acceleration of Bioinformatics Workloads

About

BIO-Arch is a new forum for presenting and discussing new ideas in accelerating bioinformatics workloads with the co-design of hardware & software and the use of new computer architectures. Our goal is to discuss new system designs tailored for bioinformatics. BIO-Arch aims to bring together researchers in the bioinformatics, computational biology, and computer architecture communities to strengthen the progress in accelerating bioinformatics analysis (e.g., genome analysis) with efficient system designs that include hardware acceleration and software systems tailored for new hardware technologies.

Venue

BIO-Arch will be held in The Social Facilities of Istanbul Technical University on April 14. Detailed information about how to arrive at the venue location with various transportation options can be found on the RECOMB website.

Our panel discussion will be held in conjunction with the main RECOMB conference. The panel discussion will be held in Marriott Şişli on April 17 at 17:00. You can find

https://www.youtube.com/watch?v=2rCsb4-nLmg

https://safari.ethz.ch/recomb23-arch-workshop/
Accelerating Genome Analysis

How Large is a Genome?

Prime Tower, Zurich

~3.2 billion genomic bases

Livestream - Seminar in Computer Architecture - ETH Zürich (Spring 2022)
Seminar in Computer Arch. - Lecture 5: Accelerating Genome Analysis (Spring 2022)

https://www.youtube.com/watch?v=qPIiiwUVFug
More on Accelerating Genome Analysis...

- Can Firtina,
  "Enabling Accurate, Fast, and Memory-Efficient Genome Analysis via Efficient and Intelligent Algorithms"
  Talk at UC Berkeley, Berkeley, CA, United States, May 27, 2022.
  [Slides (pptx) (pdf)]
  [Talk Video (1 hour 6 minutes)]
More on Accelerating Genome Analysis...

- Mohammed Alser,
  "Accelerating Genome Analysis: A Primer on an Ongoing Journey"
  Talk at RECOMB 2021, Virtual, August 30, 2021.
  [Slides (pptx) (pdf)]
  [Talk Video (27 minutes)]
  [Related Invited Paper (at IEEE Micro, 2020)]
More on Intelligent Genome Analysis…

- Mohammed Alser,
  "Computer Architecture - Lecture 10: Intelligent Genome Analysis"
  ETH Zurich, Computer Architecture Course, Fall 2021, Lecture 10, Virtual, 29 October 2021.
  [Slides (pptx) (pdf)]
  [Talk Video (3 hour 2 minutes, including Q&A)]
  [Related Invited Paper (at IEEE Micro, 2020)]
More on Intelligent Genome Analysis…

- Mohammed Alser,
  "Computer Architecture - Lecture 8: Intelligent Genome Analysis"
  ETH Zurich, Computer Architecture Course, Lecture 8, Virtual, 15 October 2021.
  [Slides (pptx) (pdf)]
  [Talk Video (2 hour 54 minutes, including Q&A)]
  [Related Invited Paper (at IEEE Micro, 2020)]
More on Fast Genome Analysis…

- Onur Mutlu,
  "Accelerating Genome Analysis: A Primer on an Ongoing Journey"
  [Slides (pptx) (pdf)]
  [Talk Video (1 hour 37 minutes, including Q&A)]
  [Related Invited Paper (at IEEE Micro, 2020)]
More on Fast Genome Analysis…

- Onur Mutlu,
  "Accelerating Genome Analysis"
  
  Invited Talk at the Barcelona Supercomputing Center (BSC), Barcelona, Spain, 6 September 2022.
  
  [Slides (pptx) (pdf)]
  [Talk Video (1 hour 35 minutes, including Q&A)]
  [Related Invited Paper (at IEEE Micro, 2020)]
  [Related Invited Paper (at Computational and Structural Biology Journal, 2022)]
Accelerating Genome Analysis [DAC 2023]

- Onur Mutlu,
  "Accelerating Genome Analysis via Algorithm-Architecture Co-Design"
  Invited Special Session Talk in Proceedings of the 60th Design Automation Conference (DAC), San Francisco, CA, USA, July 2023.
  [Slides (pptx) (pdf)]
  [Talk Video (38 minutes, including Q&A)]
  [Related Invited Paper]
Onur Mutlu and Can Firtina,
"Accelerating Genome Analysis via Algorithm-Architecture Co-Design"
[Slides (pptx) (pdf)]
[Talk Video (38 minutes, including Q&A)]
[Related Invited Paper]
[arXiv version]
More on Real-Time Genome Analysis

Can Firtina, 
"RawHash: Enabling Fast and Accurate Real-Time Analysis of Raw Nanopore Signals for Large Genomes"
[Slides (pptx) (pdf)]
[Talk Video (18 minutes)]
Genomics Course (Fall 2023)

- **Fall 2023 Edition:**

- **Spring 2023 Edition:**

- **Youtube Livestream (Fall 2023):**
  - [https://youtube.com/playlist?list=PL5Q2soXY2Zi_O0wyOjiMShG4t2QPZoeE3](https://youtube.com/playlist?list=PL5Q2soXY2Zi_O0wyOjiMShG4t2QPZoeE3)

- Project course
  - Taken by Bachelor’s/Master’s students
  - Genomics lectures
  - Hands-on research exploration
  - Many research readings

[https://www.youtube.com/onurmutlulectures](https://www.youtube.com/onurmutlulectures)
PIM Course (Fall 2022)

- **Fall 2022 Edition:**
  - [https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=processing_in_memory](https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=processing_in_memory)

- **Spring 2022 Edition:**
  - [https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=processing_in_memory](https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=processing_in_memory)

- **Youtube Livestream (Fall 2022):**
  - [https://www.youtube.com/watch?v=QLL0wQ9I4Dw&list=PL5Q2soXY2Zi8KzG2CQYRNQ0VD0GObnK](https://www.youtube.com/watch?v=QLL0wQ9I4Dw&list=PL5Q2soXY2Zi8KzG2CQYRNQ0VD0GObnK)

- **Youtube Livestream (Spring 2022):**
  - [https://www.youtube.com/watch?v=9e4Chnwdov0&list=PL5Q2soXY2Zi-841fUYUK9EsXKhQKRPvX](https://www.youtube.com/watch?v=9e4Chnwdov0&list=PL5Q2soXY2Zi-841fUYUK9EsXKhQKRPvX)

- **Project course**
  - Taken by Bachelor’s/Master’s students
  - Processing-in-Memory lectures
  - Hands-on research exploration
  - Many research readings

[https://www.youtube.com/onurmutlulectures](https://www.youtube.com/onurmutlulectures)
SSD Course (Spring 2023)

- **Spring 2023 Edition:**

- **Fall 2022 Edition:**

- **Youtube Livestream (Spring 2023):**
  - [https://www.youtube.com/watch?v=4VTwOMmsnJY&list=PL5Q2soXY2Zi_8qOM5Icpp8hB2SHtm4z57&pp=iAQB](https://www.youtube.com/watch?v=4VTwOMmsnJY&list=PL5Q2soXY2Zi_8qOM5Icpp8hB2SHtm4z57&pp=iAQB)

- **Youtube Livestream (Fall 2022):**
  - [https://www.youtube.com/watch?v=hqLrd-Uj0aU&list=PL5Q2soXY2Zi9BJhenUq4Jl5bwhAMpAp13&pp=iAQB](https://www.youtube.com/watch?v=hqLrd-Uj0aU&list=PL5Q2soXY2Zi9BJhenUq4Jl5bwhAMpAp13&pp=iAQB)

- **Project course**
  - Taken by Bachelor’s/Master’s students
  - SSD Basics and Advanced Topics
  - Hands-on research exploration
  - Many research readings

---

https://www.youtube.com/onurmutlulectures
Real PIM Tutorials [ISCA’23, ASPLOS’23, HPCA’23]

- June, March, Feb: Lectures + Hands-on labs + Invited talks

Real-world Processing-in-Memory Systems for Modern Workloads

Tutorial Description

Processing-in-Memory (PIM) is a computing paradigm that aims at overcoming the data movement bottleneck (i.e., the waste of execution cycles and energy resulting from the back-and-forth data movement between memory units and compute units) by making memory compute-capable.

Explored over several decades since the 1960s, PIM systems are becoming a reality with the advent of the first commercial products and prototypes.

A number of startups (e.g., UPMEM, Neuroblade) are already commercializing real PIM hardware, each with its own design approach and target applications. Several major vendors (e.g., Samsung, SK Hynix, Alibaba) have presented real PIM chip prototypes in the last two years. Most of these architectures have in common that they place compute units near the memory arrays. This type of PIM is called processing near memory (PNM).

2,560-DPU Processing-in-Memory System

PIM can provide large improvements in both performance and energy consumption for many modern applications, thereby enabling a commercially viable way of dealing with huge amounts of data that is bottlenecking our computing systems. Yet, it is critical to (1) study and understand the characteristics that make a workload suitable for a PIM architecture, (2) propose optimization strategies for PIM kernels, and (3) develop programming frameworks and tools that can lower the learning curve and ease the adoption of PIM.

This tutorial focuses on the latest advances in PIM technology, workload characterization for PIM, and programming and optimizing PIM kernels. We will (1) provide an introduction to PIM and taxonomy of PIM systems, (2) give an overview and a rigorous analysis of existing real-world PIM hardware, (3) conduct hands-on labs about important workloads (machine learning, sparse linear algebra, bioinformatics, etc.) using real PIM systems, and (4) shed light on how to improve future PIM systems for such workloads.

https://events.safari.ethz.ch/isca-pim-tutorial/
Real PIM Tutorial [ISCA 2023]

- June 18: Lectures + Hands-on labs + Invited talks

**ISCA 2023 Real-World PIM Tutorial**
Sunday, June 18, Orlando, Florida

Organizers: Juan Gómez Luna, Onur Mutlu, Ataberk Olguin
Program: https://events.safari.ethz.ch/isca-pim-tutorial/

**Real-world Processing-in-Memory Systems for Modern Workloads**

https://www.youtube.com/live/GIb5EgSrWk0

https://events.safari.ethz.ch/isca-pim-tutorial/
Real PIM Tutorial [ASPLOS 2023]

March 26: Lectures + Hands-on labs + Invited talks

https://www.youtube.com/watch?v=oYCaLcT0Kmo

https://events.safari.ethz.ch/asplos-pim-tutorial/
Real PIM Tutorial [HPCA 2023]

- February 26: Lectures + Hands-on labs + Invited Talks

https://www.youtube.com/watch?v=f5-nT1tbz5w

https://events.safari.ethz.ch/real-pim-tutorial/
Real-world Processing-in-Memory Systems for Modern Workloads

Table of Contents

- Real-world Processing-in-Memory Systems for Modern Workloads
- Tutorial Description
- Live streaming
- Organizations
- Agenda (Tentative, October 29, 2023)
- Lectures
- Learning Materials

Tutorial Description

Processing-in-Memory (PIM) is a computing paradigm that aims at overcoming the data motion bottleneck (i.e., the waste of execution cycles and energy resulting from the back-and-forth data movement between memory units and compute units) by making memory compute-capable.

Explored over several decades since the 1980s, PIM systems are becoming a reality with the advent of the first commercial products and prototypes.

A number of startups (e.g., UPMEM, Neuroblade) are already commercializing real PIM hardware, each with its own design approach and target applications. Several major vendors (e.g., Samsung, SK Hynix, Alibaba) have presented real PIM chip prototypes in the last two years. Most of these architectures have in common that they place compute units near the memory arrays. This type of PIM is called processing near memory (PNM).

PIM can provide large improvements in both performance and energy consumption for many modern applications, thereby enabling a commercially viable way of dealing with huge amounts of data that is bottlenecking our computing systems. Yet, it is critical to (1) study and understand the characteristics that make a workload suitable for a PIM architecture, (2) propose optimization strategies for PIM kernels, and (3) develop programming frameworks and tools that can lower the learning curve and ease the adoption of PIM.

This tutorial focuses on the latest advances in PIM technology, workload characterization for PIM, and programming and optimizing PIM kernels. We will (1) provide an introduction to PIM and taxonomy of PIM systems, (2) give an overview and a rigorous analysis of existing real-world PIM hardware, (3) conduct hand-on labs about important workloads (machine learning, sparse linear algebra, bioinformatics, etc.) using real PIM systems, and (4) shed light on how to improve future PIM systems for such workloads.

Livestream

- YouTube livestream

https://www.youtube.com/watch?v=ohU00NSIxEI

https://events.safari.ethz.ch/micro-pim-tutorial/doku.php?id=start

https://safari.ethz.ch/real-world-pim-tutorial-at-micro-2023/
Detailed Lectures on Genome Analysis

- **Computer Architecture, Fall 2020, Lecture 3a**
  - *Introduction to Genome Sequence Analysis* (ETH Zürich, Fall 2020)
  - [https://www.youtube.com/watch?v=CrRb32v7SJc&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=5](https://www.youtube.com/watch?v=CrRb32v7SJc&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=5)

- **Computer Architecture, Fall 2020, Lecture 8**
  - *Intelligent Genome Analysis* (ETH Zürich, Fall 2020)
  - [https://www.youtube.com/watch?v=ygmQpdDTL7o&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=14](https://www.youtube.com/watch?v=ygmQpdDTL7o&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=14)

- **Computer Architecture, Fall 2020, Lecture 9a**
  - *GenASM: Approx. String Matching Accelerator* (ETH Zürich, Fall 2020)
  - [https://www.youtube.com/watch?v=XoLpzmNPas&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=15](https://www.youtube.com/watch?v=XoLpzmNPas&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=15)

- **Accelerating Genomics Project Course, Fall 2020, Lecture 1**
  - *Accelerating Genomics* (ETH Zürich, Fall 2020)
  - [https://www.youtube.com/watch?v=rgjl8ZyLsAg&list=PL5Q2soXY2Zi9E2bBVAgCqLgwiDRQDtyId](https://www.youtube.com/watch?v=rgjl8ZyLsAg&list=PL5Q2soXY2Zi9E2bBVAgCqLgwiDRQDtyId)

*SAFARI [https://www.youtube.com/onurmutlulectures](https://www.youtube.com/onurmutlulectures)*
Challenges in Read Mapping

- Need to find many **mappings** of each read

- Need to **tolerate variances/sequencing errors** in each read

- Need to **map** each read **very fast** (i.e., performance is important, life critical in some cases)

- Need to **map** reads to both **forward and reverse strands**

![DNA sequence diagram](https://www.bioinformaticsalgorithms.org/bioinformatics-chapter-1)
Analysis is Bottlenecked in Read Mapping!!

48 Human whole genomes at 30× coverage in about 2 days

1 Human genome 32 CPU hours on a 48-core processor

Illumina NovaSeq 6000

A Tsunami of Sequencing Data

<table>
<thead>
<tr>
<th>A Tera-scale increase in sequencing production in the past 25 years</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Genes &amp; Operons</strong></td>
</tr>
<tr>
<td><strong>Bacterial genomes</strong></td>
</tr>
<tr>
<td><strong>Human genome</strong></td>
</tr>
<tr>
<td><strong>Human microbiome</strong></td>
</tr>
<tr>
<td><strong>50K Microbiomes</strong></td>
</tr>
</tbody>
</table>

**What is expected for the next 15 years? (a Giga?)**

| **200K Microbiomes** | 2020 | **Exa** = 1,000,000,000,000,000,000,000 |
| **1M Microbiomes** | 2025 | **Zetta** = 1,000,000,000,000,000,000,000,000 |
| **Earth Microbiome** | 2030 | **Yotta** = 1,000,000,000,000,000,000,000,000,000 |

Source: @kyrpides
Solving the Puzzle

Reference genome

Reads

FASTA file

FASTQ file

Obtaining the Human Reference Genome
n
n

n
n
n
n
n

GRCh38.p13
Description: Genome Reference Consortium Human Build 38
patch release 13 (GRCh38.p13)
Organism name: Homo sapiens (human)
Date: 2019/02/28
3,099,706,404 bases
Compressed .fna file (964.9 MB)

>NC_000001.11 Homo sapiens chromosome 1, GRCh38.p13 Primary Assembly
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
….

341


Obtaining .FASTQ Files


**ERX215261**: Whole Genome Sequencing of human TSI NA20754
1 ILLUMINA (Illumina HiSeq 2000) run: 4.1M spots, 818.7M bases, 387.2Mb downloads

**Design**: Illumina sequencing of library 6511095, constructed from sample accession SRS001721 for study accession SRP000540. This is part of an Illumina multiplexed sequencing run (9340_1). This submission includes reads tagged with the sequence TTAGGCAT.

**Submitted by**: The Wellcome Trust Sanger Institute (SC)

**Study**: Whole genome sequencing of (TSI) Toscani in Italia HapMap population

**Sample**: Coriell GM20754

**Library**:
- **Name**: 6511095
- **Instrument**: Illumina HiSeq 2000
- **Strategy**: WGS
- **Source**: GENOMIC
- **Selection**: RANDOM
- **Layout**: PAIRED
- **Construction protocol**: Standard

**Runs**: 1 run, 4.1M spots, 818.7M bases, 387.2Mb
Today’s Computing Systems

von Neumann model, 1945
where the CPU can access data stored in an off-chip main memory only through power-hungry bus

Burks, Goldstein, von Neumann, “Preliminary discussion of the logical design of an electronic computing instrument,” 1946.
Data analysis is performed far away from the data
Map reads to a known reference genome with some minor differences allowed.
Read Mapping Algorithms: Two Styles

- **Hash based seed-and-extend** (hash table, suffix array, suffix tree)
  - Index the “k-mers” in the genome into a hash table (pre-processing)
  - When searching a read, find the location of a k-mer in the read; then extend through alignment
  - More sensitive (can find all mapping locations), but slow
  - Requires large memory; this can be reduced with cost to run time

- **Burrows-Wheeler Transform & Ferragina-Manzini Index based aligners**
  - BWT is a compression method used to compress the genome index
  - Perfect matches can be found very quickly, memory lookup costs increase for imperfect matches
  - Reduced sensitivity
An Example of Hash Table Based Mappers

- Guaranteed to find \textit{all} mappings $\rightarrow$ very sensitive
- Can tolerate up to \(e\) errors

\begin{center}
\url{https://github.com/BilkentCompGen/mrfast}
\end{center}

Personalized copy number and segmental duplication maps using next-generation sequencing

Can Alkan\textsuperscript{1,2}, Jeffrey M Kidd\textsuperscript{1}, Tomas Marques-Bonet\textsuperscript{1,3}, Gozde Aksay\textsuperscript{1}, Francesca Antonacci\textsuperscript{1}, Fereydoun Hormozdiari\textsuperscript{4}, Jacob O Kitzman\textsuperscript{1}, Carl Baker\textsuperscript{1}, Maika Malig\textsuperscript{1}, Onur Mutlu\textsuperscript{5}, S Cenk Sahinalp\textsuperscript{4}, Richard A Gibbs\textsuperscript{6} & Evan E Eichler\textsuperscript{1,2}

Performance of Read Mapping

Alser+, "Technology dictates algorithms: Recent developments in read alignment", Genome Biology, 2021
The Need for Speed

Did we realize the need for faster genome analysis?

Alser+, "Technology dictates algorithms: Recent developments in read alignment", Genome Biology, 2021
### Sequence Alignment in Unavoidable

**Quadratic-time** dynamic-programming algorithm **WHY?!**

- Enumerating all possible prefixes

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>E</th>
<th>T</th>
<th>H</th>
<th>E</th>
<th>R</th>
<th>L</th>
<th>A</th>
<th>N</th>
<th>D</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>W</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>11</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

- **NETHERLANDS x SWITZERLAND**
- **NETHERLANDS x S**
- **NETHERLANDS x SW**
- **NETHERLANDS x SWI**
- **NETHERLANDS x SWIT**
- **NETHERLANDS x SWITZ**
- **NETHERLANDS x SWITZE**
- **NETHERLANDS x SWITZER**
- **NETHERLANDS x SWITZERL**
- **NETHERLANDS x SWITZERLA**
- **NETHERLANDS x SWITZERLAN**
- **NETHERLANDS x SWITZERLAND**
Sequence Alignment in Unavoidable

- **Quadratic-time** dynamic-programming algorithm
  
  Enumerating all possible prefixes

- **Data dependencies** limit the computation parallelism
  
  Processing row (or column) after another

- **Entire matrix** is computed even though strings can be dissimilar.
  
  Number of differences is computed only at the backtracking step.
Metagenomics Analysis

Reads from different unknown donors at sequencing time are mapped to many known reference genomes.

genetic material recovered directly from environmental samples

Reads “text format”

Reference Database
Genomics vs. Metagenomics
More on Metagenomic Profiling: Metalign

Nathan LaPierre, Mohammed Alser, Eleazar Eskin, David Koslicki, Serghei Mangul

“Metalign: efficient alignment-based metagenomic profiling via containment min hash” Genome Biology, September 2020.

[Talk Video] (7 minutes) at ISMB 2020

[Source code]
Key observation:
- Existing kmer-counting approaches provide inaccurate taxonomic profiles (with large number of false positives).
- Alignment-based approaches are often considered accurate yet computationally infeasible.

Key idea:
- Filter out reference genomes that don’t share enough number of regions (long kmers) with the input reads.
- Perform sequence alignment using the subset database.

Key result:
- 100x reduction in our comprehensive NCBI database (243 GB).
- 271x reduction in false positives and showing best balance between precision and recall tradeoff compared to Kraken 2.
- 8x less memory than that of Kraken 2 (325 GB) at the cost of 5x increase in execution time.
Accuracy Results of Metalign and Others

<table>
<thead>
<tr>
<th>Genus level</th>
<th>Precision</th>
<th>Recall</th>
<th>F1 Score</th>
<th>Scaled L1 error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metalign</td>
<td>0.82</td>
<td>0.84</td>
<td>0.83</td>
<td>0.73</td>
</tr>
<tr>
<td>Bracken+Kraken2</td>
<td>0.19</td>
<td>0.61</td>
<td>0.28</td>
<td>0.45</td>
</tr>
<tr>
<td>CLARK</td>
<td>0.38</td>
<td>0.53</td>
<td>0.42</td>
<td>0.46</td>
</tr>
<tr>
<td>GOTTCHA</td>
<td>0.59</td>
<td>0.23</td>
<td>0.33</td>
<td>0.14</td>
</tr>
<tr>
<td>MEGAN+DIAMOND</td>
<td>0.88</td>
<td>0.72</td>
<td>0.79</td>
<td>0.73</td>
</tr>
<tr>
<td>MetaBinG2</td>
<td>0.62</td>
<td>0.05</td>
<td>0.088</td>
<td>0.15</td>
</tr>
<tr>
<td>MetaPhlAn2</td>
<td>0.75</td>
<td>0.29</td>
<td>0.42</td>
<td>0.27</td>
</tr>
<tr>
<td>mOTUs2</td>
<td>0.84</td>
<td>0.79</td>
<td>0.82</td>
<td>0.77</td>
</tr>
</tbody>
</table>

Graph showing precision and recall for different methods.
**FastHASH**

- **Goal**: Reducing the number of seed (k-mer) locations.
  - **Heuristic** (limits the number of mapping locations for each seed).
  - Supports **exact** matches only.

_Xin et al. BMC Genomics 2013, 14(Suppl 1):S13_  
_http://www.biomedcentral.com/1471-2164/14/S1/S13_
Key Observations

- **Observation 1 (Adjacent k-mers)**
  - **Key insight:** Adjacent k-mers in the read should also be adjacent in the reference genome
  - **Key idea:** 1) sort the location list based on their number of locations and 2) search for adjacent locations in the k-mers’ location lists

![Diagram showing valid and invalid mappings](image-url)
Key Observations

- **Observation 1 (Adjacent k-mers)**
  - **Key insight:** Adjacent k-mers in the read should also be adjacent in the reference genome
  - **Key idea:** 1) sort the location list based on their number of locations and 2) search for adjacent locations in the k-mers’ location lists

- **Observation 2 (Cheap k-mers)**
  - **Key insight:** Some k-mers are cheaper to verify than others because they have shorter location lists (they occur less frequently in the reference genome)
  - **Key Idea:** Read mapper can choose the cheapest k-mers and verify their locations
Cheap K-mer Selection

- occurrence threshold = 500

Previous work needs to verify:
3004 locations

FastHASH verifies only:
8 locations
FastHASH Conclusion

- **Problem:** Existing read mappers perform poorly in mapping billions of short reads to the reference genome, in the presence of errors.

- **Observation:** Most of the verification calculations are unnecessary → filter them out.

- **Key Idea:** To reduce the cost of unnecessary verification
  - Select **Cheap** and **Adjacent** k-mers.

- **Key Result:** FastHASH obtains up to 19x speedup over the state-of-the-art mapper without losing valid mappings.
More on FastHASH

- Download source code and try for yourself
  - Download link to FastHASH

Xin et al. BMC Genomics 2013, 14(Suppl 1):S13
http://www.biomedcentral.com/1471-2164/14/S1/S13

Accelerating read mapping with FastHASH

Hongyi Xin¹, Donghyuk Lee¹, Farhad Hormozdiari², Samihan Yedkar¹, Onur Mutlu¹*, Can Alkan³*

From The Eleventh Asia Pacific Bioinformatics Conference (APBC 2013)
Vancouver, Canada. 21-24 January 2013
GateKeeper Conclusions

- FPGA-based pre-alignment greatly speeds up read mapping
  - 10x speedup of a state-of-the-art mapper (mrFAST)

- FPGA-based pre-alignment can be integrated with the sequencer
  - It can help to hide the complexity and details of the FPGA
  - Enables real-time filtering while sequencing
Can we improve the **accuracy**?
MAGNET (AACBB 2018, TIR 2017)

- **Key observation:**
  - Correct alignment always includes non-overlapping long identical subsequences.

- **Key idea:**
  - count the **consecutive zeros** in each mask and select the longest in a divide-and-conquer approach.

- **Key result:**
  - MAGNET is 74x - 460x **faster** than its CPU implementation.
  - Contains up to **2 or 8 filtering units**, each of which has **10 folds the footprint** of that of GateKeeper on the FPGA.
  - MAGNET is 3.5x to 25552x (as they stop filtering after E=4%[250bp] or 8%[100bp]) **more accurate** than GateKeeper and SHD.

- **Weaknesses:** Challenging to be implemented on FPGA due to random search.
MAGNET Walkthrough

Find the longest segment of consecutive zeros
Exclude the errors from the search space
Divide the problem into two subproblems and repeat

Total number of edits = number of 1’s in MAGNET bit-vector
**MAGNET Walkthrough**

- $E$ right-shift registers (length=ReadLength)
- $E$ left-shift registers (length=ReadLength)
- $(2E+1) \times \text{(ReadLength)}$ 2-XOR operations.

- $\log_2 \text{ReadLength}$-bit counter.

**Build Neighborhood Map**

**Identifying $E+1$ non-overlapping subsequences**

**ACCEPT iff number of ‘1’ ≤ Threshold**

---

**Pattern** $m$

**Reference** $m$

**Edit distance threshold** $m$

**Proceed to the next Iteration**
MAGNET Accelerator
More on MAGNET

- Download and test for yourself
  [https://github.com/BilkentCompGen/MAGNET](https://github.com/BilkentCompGen/MAGNET)

Sequence alignment

Shouji: a fast and efficient pre-alignment filter for sequence alignment

Mohammed Alser¹,²,³, *, Hasan Hassan¹, Akash Kumar², Onur Mutlu¹,³, * and Can Alkan³, *

¹Computer Science Department, ETH Zürich, Zürich 8092, Switzerland, ²Chair for Processor Design, Center For Advancing Electronics Dresden, Institute of Computer Engineering, Technische Universität Dresden, 01062 Dresden, Germany and ³Computer Engineering Department, Bilkent University, 06800 Ankara, Turkey

*To whom correspondence should be addressed.
Associate Editor: Inanc Birol
Received on September 13, 2018; revised on February 27, 2019; editorial decision on March 7, 2019; accepted on March 27, 2019

Shouji

- **Key observation:**
  - Correct alignment always includes long identical subsequences.
  - Processing the entire mapping at once is ineffective for hardware design.

- **Key idea:**
  - Use overlapping sliding window approach to quickly and accurately find all long segments of consecutive zeros.

- **Key result:**
  - Shouji on FPGA is up to three orders of magnitude faster than its CPU implementation.
  - Shouji accelerates best-performing CPU read aligner Edlib (Bioinformatics 2017) by up to 18.8x using 16 filtering units that work in parallel.
  - Shouji is 2.4x to 467x more accurate than GateKeeper (Bioinformatics 2017) and SHD (Bioinformatics 2015).
Shouji Walkthrough

Building the Neighborhood Map

Finding all common subsequences (diagonal segments of consecutive zeros) shared between two given sequences.

Storing it @ Shouji Bit-vector

### Building the Neighborhood Map

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>G</td>
<td>G</td>
<td>T</td>
<td>G</td>
<td>C</td>
<td>A</td>
<td>G</td>
<td>A</td>
<td>G</td>
<td>C</td>
<td>T</td>
<td>C</td>
</tr>
<tr>
<td>2</td>
<td>G</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>T</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>G</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>A</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>G</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>A</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>G</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>T</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>T</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>G</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>T</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Storing it @ Shouji Bit - vector

#### Finding all common subsequences
- Diagonal segments of consecutive zeros

**Shouji** is a fast and efficient pre-alignment filter for sequence alignment, *Bioinformatics* 2019, https://doi.org/10.1093/bioinformatics/btz234

Shouji Walkthrough

ACCEPT iff number of ‘1’ ≤ Threshold
The reason behind the selection of the window size is due to the minimal possible length of the identical subsequence that is a single match (e.g., such as `101').
• Counting is performed **concurrently** for all bit-vectors and all sliding windows in a single clock cycle using multiple 4-input LUTs.
More on Shouji

Download and test for yourself
https://github.com/CMU-SAFA/R/Shouji

Bioinformatics, 2019, 1–9
doi: 10.1093/bioinformatics/btz234
Advance Access Publication Date: 28 March 2019
Original Paper

Sequence alignment

**Shouji: a fast and efficient pre-alignment filter for sequence alignment**

Mohammed Alser¹,²,³,* , Hasan Hassan¹, Akash Kumar², Onur Mutlu¹,³,* and Can Alkan³,*

¹Computer Science Department, ETH Zürich, Zürich 8092, Switzerland, ²Chair for Processor Design, Center For Advancing Electronics Dresden, Institute of Computer Engineering, Technische Universität Dresden, 01062 Dresden, Germany and ³Computer Engineering Department, Bilkent University, 06800 Ankara, Turkey

*To whom correspondence should be addressed.
Associate Editor: Inanc Birol
Received on September 13, 2018; revised on February 27, 2019; editorial decision on March 7, 2019; accepted on March 27, 2019

Alser+, “Shouji: a fast and efficient pre-alignment filter for sequence alignment”, Bioinformatics 2019,
https://doi.org/10.1093/bioinformatics/btz234
The execution time of SneakySnake, Parasail, and SneakySnake integrated with Parasail using long reads, (a) Set_5 and (b) Set_6, and 40 CPU threads. The y-axis is on a logarithmic scale. For each edit distance threshold value, we provide the rate of accepted pairs (out of 100,000 pairs for Set_5 and out of 74,687 pairs for Set_6).
The execution time of SneakySnake, KSW2, and SneakySnake integrated with KSW2 using long reads, (a) Set_5 and (b) Set_6, and a single CPU thread. The y-axis is on a logarithmic scale. For each edit distance threshold value, we provide the rate of accepted pairs (out of 100,000 pairs for Set_5 and out of 74,687 pairs for Set_6) by SneakySnake that are passed to KSW2.
GenCache: Leveraging In-Cache Operators for Efficient Sequence Alignment

Anirban Nag
anirban@cs.utah.edu
University of Utah
Salt Lake City, Utah

C. N. Ramachandra
ramgowda@cs.utah.edu
University of Utah
Salt Lake City, Utah

Rajeev Balasubramonian
rajeev@cs.utah.edu
University of Utah
Salt Lake City, Utah

Ryan Stutsman
stutsman@cs.utah.edu
University of Utah
Salt Lake City, Utah

Edouard Giacomino
edouard.giacomin@utah.edu
University of Utah
Salt Lake City, Utah

Hari Kambalasubramanyam
hari.kambalasubramanyam@utah.edu
University of Utah
Salt Lake City, Utah

Pierre-Emmanuel Gaillardon
pierre-emmanuel.gaillardon@utah.edu
University of Utah
Salt Lake City, Utah

GenCache

- **Key observation:** State-of-the-art alignment accelerators are still bottlenecked by memory.

- **Key ideas:**
  - Performing in-cache alignment + pre-alignment filtering by enabling processing-in-cache using previous proposal, ComputeCache (HPCA’17).
  - Using different Pre-alignment filters depending on the selected edit distance threshold.

- **Results:**
  - GenCache on CPU is 1.36x faster than GenAx (ISCA 2018).
  - GenCache in cache is 5.26x faster than GenAx.
  - GenCache chip has 16.4% higher area, 34.7% higher peak power, and 15% higher average power than GenAx.
GenCache’s Four Phases

Figure 7: Four phases in the new alignment algorithm that exploits in-cache operators.
Throughput Results

Figure 9: Throughput improvement of GenCache (Hardware & Software).
Integrating GRIM-Filter into a Read Mapper

**INPUT:** Read Sequence

GAACCTTGCAG...GTATT

**1.** GRIM-Filter: Filter Bitmask Generator

...0001010...011010...

Seed Location Filter Bitmask

**INPUT:** All Potential Seed Locations

...020128 020131 414415...

**2.** GRIM-Filter: Seed Location Checker

KEEP

KEEP

DISCARD

**3.** Reference Segment Storage

reference segment @ 020131...

reference segment @ 414415

**4.** Read Mapper: Sequence Alignment

Edit-Distance Calculation

**OUTPUT:** Correct Mappings