Agenda for Today

- What is Genome Analysis?
- What is Intelligent Genome Analysis?
- How we Analyze Genome?
- What is Read Mapping?
- What Makes Read Mapper Slow?

- Algorithmic & Hardware Acceleration
  - Seed Filtering Technique
  - Pre-alignment Filtering Technique
  - Read Alignment Acceleration

- Where is Read Mapping Going Next?
Agenda for Today

- **What is Genome Analysis?**
- What is Intelligent Genome Analysis?
- How we Analyze Genome?
- What is Read Mapping?
- What Makes Read Mapper Slow?
- Algorithmic & Hardware Acceleration
  - Seed Filtering Technique
  - Pre-alignment Filtering Technique
  - Read Alignment Acceleration
- Where is Read Mapping Going Next?
What is Data Analysis?

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

Richard Hamming
What is Genome Analysis?

https://onlinelearning.hms.harvard.edu/hmx/courses/genetic-testing/
https://www.nature.com/subjects/genomic-analysis
What is Genome Analysis?

Genomic analysis is the identification, measurement or comparison of genomic features such as DNA sequence, structural variation, gene expression, or regulatory and functional element annotation at a genomic scale. Methods for genomic analysis typically require high-throughput sequencing or microarray hybridization and bioinformatics.
DNA Testing

Health + Ancestry Service

$199

- Includes everything in Ancestry + Traits Service
- PLUS
  - 10+ Health Predisposition reports*
  - 5+ Wellness reports
  - 40+ Carrier Status reports*

SAFARI  
https://www.myheritage.ch/dna  
https://www.23andme.com/
Human Chromosomes (23 Pairs)

Autosomes

From mom

From dad

Sex chromosomes

XX  or  XY
Human Chromosomes (23 Pairs)

Autosomes

From mom

From dad

= Adenine

= Thymine

= Cytosine

= Guanine

= Phosphate backbone

Sex chromosomes

XX or XY

SAFARI
Finding SNPs Associated with Complex Trait

<table>
<thead>
<tr>
<th>Individual #1</th>
<th>SNP1</th>
<th>SNP2</th>
<th>Blood Pressure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>...ACATG</td>
<td></td>
<td>180</td>
</tr>
<tr>
<td></td>
<td>CCGACATTTTCATA</td>
<td></td>
<td>175</td>
</tr>
<tr>
<td>Individual #2</td>
<td></td>
<td></td>
<td>170</td>
</tr>
<tr>
<td></td>
<td>...ACATG</td>
<td></td>
<td>165</td>
</tr>
<tr>
<td>Individual #3</td>
<td></td>
<td></td>
<td>160</td>
</tr>
<tr>
<td></td>
<td>CCGACATTTTCATA</td>
<td></td>
<td>145</td>
</tr>
<tr>
<td>Individual #4</td>
<td></td>
<td></td>
<td>140</td>
</tr>
<tr>
<td></td>
<td>...ACATG</td>
<td></td>
<td>130</td>
</tr>
<tr>
<td>Individual #5</td>
<td></td>
<td></td>
<td>120</td>
</tr>
<tr>
<td></td>
<td>CCGACATTTTCATA</td>
<td></td>
<td>120</td>
</tr>
<tr>
<td>Individual #6</td>
<td></td>
<td></td>
<td>115</td>
</tr>
<tr>
<td>Individual #7</td>
<td></td>
<td></td>
<td>110</td>
</tr>
<tr>
<td>Individual #8</td>
<td></td>
<td></td>
<td>110</td>
</tr>
<tr>
<td>Individual #9</td>
<td></td>
<td></td>
<td>105</td>
</tr>
<tr>
<td>Individual #10</td>
<td></td>
<td></td>
<td>100</td>
</tr>
<tr>
<td>Individual #11</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Individual #12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Individual #13</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Individual #14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Individual #15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Individual #16</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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-Armstrong², Nicholas Mancuso³, Alvaro N. Barbeira⁴, David A. Knowles⁵,⁶, David Golan², Raili Ermel⁷, Arno Ruusalepp⁷,⁸, Thomas Quertermous⁹, Ke Hao¹⁰, Johan L. M. Björkegren¹⁰,¹¹,¹², Hae Kyung Im¹⁰,⁴, Bogdan Pasaniuc¹³,¹⁴, Manuel A. Rivas¹⁵, and Anshul Kundaje¹,²

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

**Basic Information**

<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>

**Links to SNPedia**

<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
Personalized Medicine for Critically Ill Infants

- **rWGS** can be performed in 2-day *(costly)* or 5-day time to interpretation.
- Diagnostic **rWGS** for infants
  - Avoids **morbidity**
  - Reduces **hospital stay length** by 6%-69%
  - Reduces **inpatient cost** by $800,000-$2,000,000.
“From 2019, all seriously ill children in UK will be offered whole genome sequencing as part of their care”
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

SAFARI
Agenda for Today

- What is Genome Analysis?
- **What is Intelligent Genome Analysis?**
- How we Analyze Genome?
- What is Read Mapping?
- What Makes Read Mapper Slow?

- Algorithmic & Hardware Acceleration
  - Seed Filtering Technique
  - Pre-alignment Filtering Technique
  - Read Alignment Acceleration

- Where is Read Mapping Going Next?
What is Intelligent Genome Analysis?

- Fast genome analysis
  - Real-time analysis?

- Population-scale genome analysis
  - Number of analyses per day!

- Using intelligent architectures
  - Small specialized HW with less data movement

- DNA is a valuable asset
  - Controlled-access analysis

- Avoiding erroneous analysis
  - E.g., your father is not your father
Does intelligent genome analysis really matter?
Fast Genome Analysis?

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

1997: *Gattaca*  
2015: *Tomorrowland*
Rapid Surveillance of Disease Outbreaks?

Figure 1: Deployment of the portable genome surveillance system in Guinea.

Scalable SARS-CoV-2 Testing

Swab-Seq: A high-throughput platform for massively scaled up SARS-CoV-2 testing


doi: https://doi.org/10.1101/2020.08.04.20167874

Bloom+, "Swab-Seq: A high-throughput platform for massively scaled up SARS-CoV-2 testing", medRxiv, 2020
Population-Scale Microbiome Profiling

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?

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

Afshinnekoo+, "Geospatial Resolution of Human and Bacterial Diversity with City-Scale Metagenomics", Cell Systems, 2015
Population-Scale Microbiome Profiling

Danko+, "A global metagenomic map of urban microbiomes and antimicrobial resistance", Cell, 2021
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 (Yersinia Pestis)

What Is It?

Published: December, 2018

Plague is caused by Yersinia pestis and has existed for over 2,000 years. Plague has caused skin sores that form in less than one-third of the population, leading to death within 6 to 9 days.

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

The New York Times

Bubonic Plague in the Subway System? Don’t Worry About It

In October, riders were not deterred after reports that an Ebola-infected man had ridden the subway just before he fell ill. Robert Stolarik for The New York Times

data. 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...
Intelligent Architecture?

Modern systems

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

SAFARI
Intelligent Architecture?

Modern systems

FPGAs

Heterogeneous Processors and Accelerators

(General Purpose) GPUs

Hybrid Main Memory

Persistent Memory/Storage

Sequencing Machine

https://nanoporetech.com/products/smidgion
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.
Can you Really Anonymize the Donors of Genomic Data in Today’s Digital World?

Mohammed Alser, Nour Almadhoun, Azita Nouri, Can Alkan, and Erman Ayday

Computer Engineering Department, Bilkent University, 06800 Bilkent, Ankara, Turkey

Abstract. The rapid progress in genome sequencing technologies leads to availability of high amounts of genomic data. Accelerating the pace of biomedical breakthroughs and discoveries necessitates not only collecting millions of genetic samples but also granting open access to genetic databases. However, one growing concern is the ability to protect the privacy of sensitive information and its owner. In this work, we survey a wide spectrum of cross-layer privacy breaching strategies to human genomic data (using both public genomic databases and other public non-genomic data). We outline the principles and outcomes of each technique, and assess its technological complexity and maturation. We then review potential privacy-preserving countermeasure mechanisms for each threat.

Keywords: Genomics, Privacy, Bioinformatics

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.

---

**30x Whole Genome Sequencing DNA Test**

$299

Normally $499
Save 70%

A genetic test that decodes 100% of your DNA with very high accuracy. 30x Whole Genome Sequencing offers the best value for money and is the best choice for most people.

---

**100x Whole Genome Sequencing DNA Test**

$999

Normally $1599
Save 70%

A genetic test that decodes 100% of your DNA with extremely high accuracy. 100x Whole Genome Sequencing is recommended for the discovery of rare genetic mutations.

---

[SAFARI](https://nebula.org/whole-genome-sequencing/)
Achieving Intelligent Genome Analysis?

How and where to enable fast, accurate, cheap, privacy-preserving, and exabyte scale analysis of genomic data?
Agenda for Today

- What is Genome Analysis?
- What is Intelligent Genome Analysis?

- How we Analyze Genome?
- What is Read Mapping?
- What Makes Read Mapper Slow?

- Algorithmic & Hardware Acceleration
  - Seed Filtering Technique
  - Pre-alignment Filtering Technique
  - Read Alignment Acceleration

- Where is Read Mapping Going Next?
NO machine can read the entire content of a genome.

>CCTCCTCAGTGCCACCCAGCCCACTGGCAGCTCCCAAACAGGCTCTTATTAAAACACCCTGTTCCCTGCCCCTTGGAGTGAGGTGTCAAGGACCTAAACTAAAAAAAAAAAAAGAAAAAGAAAAGAAAAAGAATTTAAAATTTAAGTAATTCTTTGAAAAAAACTAATTTCTAAGCTTCTT
CATGTCAAGGACCTAATGTGCTAAACAGCACTTTTTTGACCATTATTTTGGATCTGAAAGAAATCAAGAATAAATGAAGGACTTGATACATTGGAAGAGGAGAGTCAAGGACCTACAGAAAAAAAAAAAAAGAAAAAGAAAAGAAAAAGA

ACTAATTTCTAAGCTTCTTCATGTCAAGGACCTAATGTGCTTGCTGTTGCGAGTCTTTCTTGACATTTCCCTGTCAAAGAAAGAAAGAATTTAAAATTTAAGTAATTCTTTGAAAAAAACTAATTTCTAAGCTTCTTTCATGTCAAGGACCTAATGTGCTGTACTATACTGAACGTTATCTAGGGGAAAGATTGAAGGGGAGCTCTAAGGTCAACACACCACCACTTCCCAGAAAGCTTCTTCA……
Genome Analysis

No machine can read the entire content of a genome

Why?!
Next-generation sequencing library preparation: simultaneous fragmentation and tagging using *in vitro* transposition

Fraz Syed, Haiying Grunenwald & Nicholas Caruccio

*Nature Methods* 6, i–ii (2009) | Cite this article

16k Accesses | 4 Citations | 5 Altmetric | Metrics

https://www.nature.com/articles/nmeth.f.272
Suggested Readings

nature biotechnology

Published: 09 October 2008

Next-generation DNA sequencing

Jay Shendure & Hanlee Ji

Nature Biotechnology 26, 1135–1145 (2008) | Cite this article

149k Accesses | 2645 Citations | 79 Altmetric | Metrics

https://www.nature.com/articles/nbt1486
Genome Sequencer is a Chopper

Genome Analysis Pipeline

Genomic Sample → Sequencing Machine → Reads → Genomic Variants

Read Mapping

ACGT

1x10^{12} bases*

44 hours*

<1000 $

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

Alser+, "Technology dictates algorithms: Recent developments in read alignment", Genome Biology, 2021
High-Throughput Sequencers

Illumina MiSeq

Illumina NovaSeq 6000

Pacific Biosciences RS II

Oxford Nanopore PromethION

Pacific Biosciences Sequel II

Oxford Nanopore MinION

Oxford Nanopore SmidgION

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

<table>
<thead>
<tr>
<th></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)
## 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><strong>Run time</strong></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><strong>Max. reads per run</strong></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><strong>Max. read length</strong></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><strong>Max. output</strong></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><strong>Estimated price</strong></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>

https://www.illumina.com/systems/sequencing-platforms.html
How Does Illumina Machine Work?

[Diagram showing the process of sequencing using Illumina technology, with nucleotides T, C, A, and G being added one by one and an optical sensor detecting the sequence.]
How Does Illumina Machine Work?

**Glass flow cell surface**

**Optical Sensor**

**DNA fragment = Read**

Billions of Short Reads

- ATATATACGTACGTACGT
- TTTTAGTACGTACGT
- ATACGTACGTACGT
- TGGGACCTACGT
- TTAGTACGTACGT
- TTAGTACGTACGT
- TTAGTACGTACGT
- TTAGTACGTACGT
- TTAGTACGTACGT
How Does Illumina Machine Work?

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

DNA fragment = Read
How Does Nanopore Machine Work?

- **Nanopore** is a nano-scale hole (<20nm).
- In nanopore sequencers, an **ionic current** passes through the nanopores.
- When the DNA strand passes through the nanopore, the sequencer measures the change in current.
- This change is used to identify the bases in the strand with the help of different electrochemical structures of the different bases.

Figure is adapted from: [https://phys.org/news/2013-12-gene-sequencing-future.html](https://phys.org/news/2013-12-gene-sequencing-future.html)
How Does Nanopore Machine Work?

Nanopore is a nano-scale hole (<20nm).

In nanopore sequencers, an ionic current passes through the nanopores. When the DNA strand passes through the nanopore, the sequencer measures the change in current. This change is used to identify the bases in the strand with the help of different electrochemical structures of the different bases.

Check Nanopore virtual tour:

https://nanoporetech.com/resource-centre/minion-video

measures the change in current

This change is used to identify the bases in the strand with the help of different electrochemical structures of the different bases.

Figure is adapted from: https://phys.org/news/2013-12-gene-sequencing-future.html
Beyond sequencing: machine learning algorithms extract biology hidden in Nanopore signal data

Trends in Genetics, October 25, 2021

Yuk Kei Wan, Christopher Hendra, Ploy N. Pratanwanich, and Jonathan Göke
Regardless the sequencing machine, reads still lack information about their order and location (which part of genome they are originated from)
Solving the Puzzle

Reference genome

Reads

HTS 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%)

- 500-2M bp
- high error rate (~15%)

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/
Changes in sequencing technologies can render some read mapping algorithms irrelevant.
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]
Looking forward,
Will we be able to read
the entire genome sequence?
Agenda for Today

- What is Genome Analysis?
- What is Intelligent Genome Analysis?
- How we Analyze Genome?
- **What is Read Mapping?**
- What Makes Read Mapper Slow?
- Algorithmic & Hardware Acceleration
  - Seed Filtering Technique
  - Pre-alignment Filtering Technique
  - Read Alignment Acceleration
- Where is Read Mapping Going Next?
Map **reads** to a known reference genome with some minor differences allowed.
Solving the Puzzle

Reference genome

Reads

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.

...
Three Decades & Yet to be Complete!

The complete sequence of a human genome


doi: https://doi.org/10.1101/2021.05.26.445798
27 May 2021

COMPLETING THE HUMAN GENOME
Researchers have been filling in incompletely sequenced parts of the human reference genome for 20 years, and have now almost finished it, with 3.05 billion DNA base pairs.

200 million new bases

0.3% of sequence might still have errors. Includes X but not Y chromosome. Count excludes mitochondrial DNA.
Obtaining the Human Reference Genome

- **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)
How Long is DNA?

- Phi X174 virus: 5.386 Killo 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
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

**Organism**: Homo sapiens

**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

<table>
<thead>
<tr>
<th>Run</th>
<th># of Spots</th>
<th># of Bases</th>
<th>Size</th>
<th>Published</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERR240727</td>
<td>4,093,747</td>
<td>818.7M</td>
<td>387.2Mb</td>
<td>2013-03-22</td>
</tr>
</tbody>
</table>
Let’s learn how to map a read
Read Mapping: A Brute Force Algorithm

Very expensive!

\[ O(m^2 kn) \]

- \( m \): read length
- \( k \): no. of reads
- \( n \): reference genome length
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]
Feedback From Our Community!

James Ferguson
@Psy_Fer_

This is awesome! I've got my evening reading sorted.

Stéphane Le Crom
@slecrom

Very complete article on the evolution of read alignment algorithms. #NGS #genomics

Svetlana Gorokhova
@SGorokhova

An impressive overview of read alignment methods over the last three decades

BContrerasMoreira @BrunoContrerasM · Sep 10
Replying to @mealser @GenomeBiology and 3 others
Buen hilo de repaso sobre la evolución de los algoritmos de alineamiento de secuencias a medida que ha mejorado la tecnología de secuenciación

https://twitter.com/mealser/status/1435223377644503040
Mapping a read is similar to querying the yellow pages!
Similar to Searching Yellow Pages!

- Step 1: Get the page number from the book’s index using a small portion of the name (e.g., 1st letter).

- Step 2: Retrieve the page(s).

- Step 3: Match the full name & get the phone number.
Matching Each Read with Reference Genome

.FASTA file:

>NG_008679.1:5001-38170 Homo sapiens paired box 6 (PAX6)
ACCCCTCTTCATATATATCTCTCTTATATACTCTGGGCGAGG
GAACGCGCGCTGTCAGATCT
GCCACTTCCCTGCGGAGCGCGCGGGTGAGGATCTGGAACCGCGCTCACCTGACCCTCCCGCC
CCTCGCTCCAGGTAACCCCGCC
CCCGGGCCCAGCTCGGGGCCGCCGGGGCCCTCTCGCCTG
CCGAGCAGCTGGCTGCCCAATCAAGCCCGCCTCAAGTGCGCCCGGGCTTCTAGTTTGCTTTTAAG
GAGGCAATAAACGATGGAAAGCGAGTTACTGAGGGAGGAAGAAGGAGGGGTGGAGGGAGGGACTTGTCTT
TCGGAGTGTAAAAGTAGCAAGTTATACCTCAAGCTCCAGTCCCGCCCT
GAGCTGGAGTAGGGGGCGGGGTCTCTGTCTGTCTGTCTGTCTGCTAAAGCCACTCGGACCACGCCAATGCA
GGAGGTGGGACGCACTTTGCATCCAGACCTCTCTCTGACTCGCAGTTC
ACGCTTTGGGAAAG
TCCGTACCCCGCGCCT
AAGACACCCTGCGCGGCGGCGAGGTGCACGCAAAGTTTCCC
GCCGTTGCAAAGTGCAAGATGGCTGAGGGCTCTAGAGATGGGTGTCGTCTTCAGAAAGACGC

.FASTQ file:

@HWI-EAS209_0006_FC706VJ:5:58:5894:21141#ATCACG/1
T
AATAAATCTTTAGATN
+HWI-EAS209_0006_FC706VJ:5:58:5894:21141#ATCACG/1
efffffefffcefffefffefffeffdf`feed]\`_Ba^__[YBBB BBB BBBBBBRTT
Step 1: Indexing the Reference Genome
Hashing is the most popular indexing technique for read mapping since 1988.

Alser+, "Technology dictates algorithms: Recent developments in read alignment", Genome Biology, 2021
Step 1: Indexing the Reference Genome

Index the first seed at location 1

Seed = k-mer (string of length k)

Location list:
- 1 9 16 30
- 2 7 60
- 3 5 12
- 4 10 18 32
- 6 14

Seed location at the reference genome
Genome Index Properties

- The index is built only once for each reference.

- Seeds can be overlapping, non-overlapping, spaced, adjacent, 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\textcolor{red}{TAG}\textcolor{blue}{TAT}\textcolor{blue}{ATA}\textcolor{blue}{TAT}\textcolor{green}{ACT}\textcolor{green}{TAG}\textcolor{green}{TAC}\textcolor{green}{GT}\textcolor{green}{T}

read 2: TAT\textcolor{red}{TCT}\textcolor{green}{TACG}\textcolor{green}{TAGT}\textcolor{green}{AGTA}\textcolor{red}{CCG}\textcolor{green}{CCC}

read 3: GCG\textcolor{red}{TCT}\textcolor{green}{TAT}\textcolor{green}{TAT}\textcolor{green}{CCG}\textcolor{green}{TAGT}\textcolor{green}{AT}\textcolor{green}{TAT}\textcolor{green}{GGT}

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

read 1: C C T T A G T A T A T A T A C T A C G T A C G T T

read 2: T A T T C T T A C G T A C T A G T A C C G C C C

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

seed

<table>
<thead>
<tr>
<th>location list</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 9 16 30</td>
</tr>
<tr>
<td>2 7 60</td>
</tr>
<tr>
<td>3 5 12</td>
</tr>
<tr>
<td>4 10 18 32</td>
</tr>
<tr>
<td>6 14</td>
</tr>
</tbody>
</table>

seed location at the reference genome

reference genome

seed from read 1

location list from index data structure

Step 2: Query the Index Using Read Seeds

We can query the Hash table with substrings from reads to quickly find a list of possible mapping locations.
### Step 3: Sequence Alignment (Verification)

<table>
<thead>
<tr>
<th></th>
<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>
<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>
<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>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>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>
<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>
<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>
<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>
<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>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>
<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.

<table>
<thead>
<tr>
<th>Ref Read</th>
<th>Ref Read</th>
</tr>
</thead>
<tbody>
<tr>
<td>o - - r g a - n i z a - t i o n</td>
<td>o r g a n i z a - t i o n</td>
</tr>
<tr>
<td>o p e r - a - - - - - t i o n</td>
<td>t r - a n - s l a - t i o n</td>
</tr>
</tbody>
</table>

Edit distance = 7

<table>
<thead>
<tr>
<th>Ref Read</th>
<th>Ref Read</th>
</tr>
</thead>
<tbody>
<tr>
<td>o - - r g a - n i z a - t i o n</td>
<td>o r g a n i z a - t i o n</td>
</tr>
<tr>
<td>o p e r - a - - - - - t i o n</td>
<td>t r - a n - s l a - t i o n</td>
</tr>
</tbody>
</table>

Edit distance = 7

<table>
<thead>
<tr>
<th>Ref Read</th>
<th>Ref Read</th>
</tr>
</thead>
<tbody>
<tr>
<td>o - - r g a - n i z a - t i o n</td>
<td>o r g a n i z a - t i o n</td>
</tr>
<tr>
<td>o p e r - a - - - - - t i o n</td>
<td>t r - a n - s l a - t i o n</td>
</tr>
</tbody>
</table>

Edit distance = 4
Popular Algorithms for Sequence Alignment

Smith-Waterman remains the most popular algorithm since 1988

Hamming distance is the second most popular technique since 2008

Alser+, "Technology dictates algorithms: Recent developments in read alignment", Genome Biology, 2021
An Example of Hash Table Based Mappers

- Guaranteed to find all mappings → very sensitive
- Can tolerate up to $e$ errors

https://github.com/BilkentCompGen/mrfast

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

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

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
Read Mapping

Map reads to a known reference genome with some minor differences allowed.
Metagenomics Analysis

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

Genomics

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]
Check Also CAMI II Paper

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”

bioRxiv, 2021

[Source Code]

Critical Assessment of Metagenome Interpretation - the second round of challenges


doi: https://doi.org/10.1101/2021.07.12.451567
Check Also MiCoP

Nathan LaPierre, Serghei Mangul, Mohammed Alser, Igor Mandric, Nicholas C. Wu, David Koslicki & Eleazar Eskin

“MiCoP: microbial community profiling method for detecting viral and fungal organisms in metagenomic samples”

BMC Genomics, June 2019.

[Source code]
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
Revisiting the Puzzle

“African pan-genome contains ~10% more DNA bases than the current human reference genome”
“Switching to a consensus reference would offer important advantages over the continued use of the current reference with few disadvantages”
Analysis is Bottlenecked in Read Mapping!!

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

Illumina NovaSeq 6000

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

Agenda for Today

- What is Genome Analysis?
- What is Intelligent Genome Analysis?
- How we Analyze Genome?
- What is Read Mapping?
- **What Makes Read Mapper Slow?**
- Algorithmic & Hardware Acceleration
  - Seed Filtering Technique
  - Pre-alignment Filtering Technique
  - Read Alignment Acceleration
- Where is Read Mapping Going Next?
What makes read mapping a bottleneck?
A Tsunami of Sequencing Data

A Tera-scale increase in sequencing production in the past 25 years

<table>
<thead>
<tr>
<th></th>
<th>Year</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Genes &amp; Operons</td>
<td>1990</td>
<td>Kilo = 1,000</td>
</tr>
<tr>
<td>Bacterial genomes</td>
<td>1995</td>
<td>Mega = 1,000,000</td>
</tr>
<tr>
<td>Human genome</td>
<td>2000</td>
<td>Giga = 1,000,000,000</td>
</tr>
<tr>
<td>Human microbiome</td>
<td>2005</td>
<td>Tera = 1,000,000,000,000,000</td>
</tr>
<tr>
<td>50K Microbiomes</td>
<td>2015</td>
<td>Peta = 1,000,000,000,000,000,000</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th></th>
<th>Year</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>200K Microbiomes</td>
<td>2020</td>
<td>Exa = 1,000,000,000,000,000,000,000</td>
</tr>
<tr>
<td>1M Microbiomes</td>
<td>2025</td>
<td>Zetta = 1,000,000,000,000,000,000,000</td>
</tr>
<tr>
<td>Earth Microbiome</td>
<td>2030</td>
<td>Yotta = 1,000,000,000,000,000,000,000,000</td>
</tr>
</tbody>
</table>

Source: @kyrpides
Lack of Specialized Compute Capability

Specialized Machine for Sequencing

General-Purpose Machine for Analysis

FAST

SLOW
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.
The Problem

Data analysis is performed far away from the data
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
* Pandiyar 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
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
**Sequence Alignment in Unavoidable**

- **Quadratic-time** dynamic-programming algorithm

```
WHY?!
```

Enumerating all possible prefixes

- 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

---

**SAFARI**
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.
Computational Cost is Mathematically Proven

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 $M^{O(1)2^{(1-\epsilon)N}}$ for a constant $\epsilon > 0$. The latter result would violate the Strong Exponential Time Hypothesis, which postulates that such algorithms do not exist.

https://arxiv.org/abs/1412.0348
Large Search Space for Mapping Location

98% of candidate locations have high dissimilarity with a given read

Computing System

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

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.

Image source: https://science.sciencemag.org/content/368/6495/eaam9744
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
Agenda for Today

- What is Genome Analysis?
- What is Intelligent Genome Analysis?
- How we Analyze Genome?
- What is Read Mapping?
- What Makes Read Mapper Slow?

- Algorithmic & Hardware Acceleration
  - Seed Filtering Technique
  - Pre-alignment Filtering Technique
  - Read Alignment Acceleration

- Where is Read Mapping Going Next?
Accelerating Read Mapping

Ongoing Directions

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

- **Pre-alignment Filtering Technique:**
  - **Goal:** Reducing the number of *invalid mappings* (>E).
    - Supports both *exact* and *inexact* matches.
    - Provides some *falsely-accepted* mappings.

- **Read Alignment Acceleration:**
  - **Goal:** Performing read alignment at scale.
    - Limits the *numeric range* of each cell in the DP table and hence supports *limited scoring* function.
    - May not support *backtracking* step due to random memory accesses.
Our Contributions

- **Near-memory/In-memory Pre-alignment Filtering**
  - GRIM-Filter [BMC Genomics’18]
  - GenASM [MICRO 2020]
  - SneakySnake [IEEE Micro’21]

- **Near-memory Sequence Alignment**
  - GenASM [MICRO 2020]

- **Specialized Pre-alignment Filtering Accelerators (GPU, FPGA)**
  - GateKeeper [Bioinformatics’17]
  - MAGNET [AACBB’18]
  - Shouji [Bioinformatics’19]
  - GateKeeper-GPU [arXiv’21]
  - SneakySnake [Bioinformatics’20]
Ongoing Directions

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

- **Pre-alignment Filtering Technique:**
  - **Goal:** Reducing the number of invalid mappings ($>E$).
  - Supports both **exact and inexact** matches.
  - Provides some **falsely-accepted** mappings.

- **Read Alignment Acceleration:**
  - **Goal:** Performing read alignment at scale.
    - Limits the **numeric range** of each cell in the DP table and hence supports **limited scoring** function.
    - May not support **backtracking** step due to random memory accesses.
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 mapping of reads to the reference genome.](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
Ongoing Directions

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

- **Pre-alignment Filtering Technique:**
  - **Goal:** Reducing the number of *invalid mappings* (>E).
    - Supports both *exact and inexact* matches.
    - Provides some *falsely-accepted* mappings.

- **Read Alignment Acceleration:**
  - **Goal:** Performing read alignment at scale.
    - Limits the *numeric range* of each cell in the DP table and hence supports *limited scoring* function.
    - May not support *backtracking* step due to random memory accesses.
Pre-alignment Filtering Technique

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!
Ideal Filtering Algorithm

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

Alser+, "GateKeeper: A New Hardware Architecture for Accelerating Pre-Alignment in DNA Short Read Mapping", Bioinformatics, 2017.
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 ($\sum \Theta$)

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

AND

Count 1’s

0 0 0 1 1 1 1 1

1 1 1 0 0 0 0 0

0 0 0 1 0 0 0 0

7 matches 1 mismatches

Edit = 1 Deletion
Our goal to track the diagonally consecutive matches in the neighborhood map.
Our goal is to track the diagonally consecutive matches in the neighborhood map.
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>-1</td>
<td>-2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>-1</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>-2</td>
<td>-1</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>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>1</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.

x10^{12} mappings

Billions of Short Reads

1. High throughput DNA sequencing (HTS) technologies

2. Read Pre-Alignment Filtering
   Fast & Low False Positive Rate

3. Read Alignment
   Slow & Zero False Positives

x10^{3} mappings
GateKeeper Walkthrough (cont’d)

- **Generate 2E+1 masks**
- **Amend random zeros:** 101 → 111 & 1001 → 1111
- **AND all masks, ACCEPT iff number of ’1’ ≤ Threshold**

• E right-shift registers (length=ReadLength)
• E left-shift registers (length=ReadLength)
• (2E+1) * (ReadLength) 2-XOR operations.

• (2E) *(ReadLength) 2-AND operations.
• (ReadLength/4) 5-input LUT.
• \( \log_2 \text{ReadLength} \)-bit counter.

**Hamming mask**

01001000110100010101100111100010010

**Hamming mask after amending**

0111100011100011111111111111100011110

• (2E+1)* (ReadLength) 5-input LUT.
The LUTs in 7 series FPGAs can be configured as either a 6-input LUT with one output, or as two 5-input LUTs with separate outputs.

**Figure 1-1:** Arrangement of Slices within the CLB

**Table 2-1:** Logic Resources in One CLB

<table>
<thead>
<tr>
<th>Slices</th>
<th>LUTs</th>
<th>Flip-Flops</th>
<th>Arithmetic and Carry Chains</th>
<th>Distributed RAM$^{(1)}$</th>
<th>Shift Registers$^{(1)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>8</td>
<td>16</td>
<td>2</td>
<td>256 bits</td>
<td>128 bits</td>
</tr>
</tbody>
</table>
**GateKeeper Accelerator Architecture**

- **Maximum data throughput** = \( \sim 13.3 \) billion bases/sec
- Can examine **8 (300 bp) or 16 (100 bp) mappings concurrently** at 250 MHz
- **Occupies 50\%** (100 bp) to **91\%** (300 bp) of the FPGA slice LUTs and registers
FPGA Chip Layout

GateKeeper: 17.6%, PCIe Controller, RIFFA, and IO: 5%

300 bp

E=15
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
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
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 Xin\textsuperscript{1,*}, John Greth\textsuperscript{2}, John Emmons\textsuperscript{2}, Gennady Pekhimenko\textsuperscript{1}, Carl Kingsford\textsuperscript{3}, Can Alkan\textsuperscript{4,*} and Onur Mutlu\textsuperscript{2,*}
More on GateKeeper

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

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](https://doi.org/10.1093/bioinformatics/btx342)

**Published:** 31 May 2017

Can we do better? Scalability?
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
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).
Building the Neighborhood Map

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

Storing it @ Shouji Bit-vector

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

#### Building the Neighborhood Map

Storing it @ Shouji Bit - vector

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

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

---

**Shouji: a fast and efficient pre-alignment filter for sequence alignment**, *Bioinformatics* 2019, [https://doi.org/10.1093/bioinformatics/btz234](https://doi.org/10.1093/bioinformatics/btz234)
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').
Hardware Implementation

- Counting is performed **concurrently** for *all* bit-vectors and all sliding windows in a single clock cycle using **multiple 4-input LUTs**.

**SAFARI**

164
More on Shouji

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

Sequence alignment

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

Mohammed Alser\textsuperscript{1,2,3,*}, Hasan Hassan\textsuperscript{1}, Akash Kumar\textsuperscript{2}, Onur Mutlu\textsuperscript{1,3,*} and Can Alkan\textsuperscript{3,*}

\textsuperscript{1}Computer Science Department, ETH Zürich, Zürich 8092, Switzerland, \textsuperscript{2}Chair for Processor Design, Center For Advancing Electronics Dresden, Institute of Computer Engineering, Technische Universität Dresden, 01062 Dresden, Germany and \textsuperscript{3}Computer 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
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)
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:

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

<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>$3^{rd}$ 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>0</td>
<td>1</td>
</tr>
<tr>
<td>$2^{nd}$ 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>1</td>
<td>0</td>
</tr>
<tr>
<td>$1^{st}$ 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>$1^{st}$ 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>1</td>
<td>0</td>
</tr>
<tr>
<td>$2^{nd}$ 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>$3^{rd}$ 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

\[ E = 3 \]
SneakySnake Walkthrough

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

171
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 $37.7\times$ and $43.9\times$ (>12× on average), on CPUs
  - up to $413\times$ and $689\times$ (>400× on average) with FPGA/GPU acceleration

- Using long reads, SneakySnake accelerates Parasail and KSW2 by $140.1\times$ and $17.1\times$ on average, respectively, on CPUs
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**
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

ETH ZURICH D-ITET
Computer Architecture - Lecture 6: Processing using Memory (Fall 2021)
721 views • Streamed live on Oct 15, 2021

Onur Mutlu Lectures
19.6K subscribers

https://www.youtube.com/watch?v=HNd4skQrt6I
Computer Architecture
Lecture 7: Processing using Memory II

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

https://www.youtube.com/watch?v=k56x2qcaXWY
Processing Near Memory

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

Computer Architecture
Lecture 9a: Real PIM Systems: UPMEM Case Study

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

Computer Architecture - Lecture 9: Real PIM Systems: UPMEM Case Study (Fall 2021)

137 views • Streamed live 5 hours ago

Onur Mutlu Lectures
19.6K subscribers

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 $27.4 \times$ and $133 \times$, respectively, over a 16-core (64 hardware threads) IBM POWER9 CPU.

SAFARI
More on SneakySnake [Bioinformatics 2020]

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]

SneakySnake: a fast and accurate universal genome pre-alignment filter for CPUs, GPUs and FPGAs

Mohammed Alser, Taha Shahroodi, Juan Gómez-Luna, Can Alkan, Onur Mutlu

Bioinformatics, btaa1015, https://doi.org/10.1093/bioinformatics/btaa1015
Published: 26 December 2020 Article history ▼
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)
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 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
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 $t = \text{number of bins}$ and $n = \text{token length}$.

For bin size $\sim 200$, and $n = 5$, memory footprint $\sim 3.8 \text{ GB}$.
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 $r$

```
GAACTTGGAGTCTA ... CGAG
```

1. **Get tokens**

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

3. **Match tokens to bitvector**

4. **Sum**

5. **Compare**

![Diagram showing the process of GRIM-Filter](image)

- **≥ Threshold?**
  - NO: **Discard**
  - YES: **Send to Read Mapper for Sequence Alignment**
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](https://arxiv.org) (pdf)
GenCache

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 Giacomini
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.
Figure 7: Four phases in the new alignment algorithm that exploits in-cache operators.
Figure 9: Throughput improvement of GenCache (Hardware & Software).
Ongoing Directions

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

- **Pre-alignment Filtering Technique:**
  - **Goal:** Reducing the number of *invalid mappings* (\(>E\)).
  - Supports both *exact* and *inexact* matches.
  - Provides some *falsely-accepted* mappings.

- **Read Alignment Acceleration:**
  - **Goal:** Performing read alignment at scale.
    - Limits the *numeric range* of each cell in the DP table and hence supports *limited scoring* function.
    - May not support *backtracking* step due to random memory accesses.
GenASM Framework [MICRO 2020]

- Damla Senol Cali, Gurpreet S. Kalsi, Zulal Bingol, Can Firtina, Lavanya Subramanian, Jeremie S. Kim, Rachata Ausavarungnirun, Mohammed Alser, Juan Gomez-Luna, Amirali Boroumand, Anant Nori, Allison Scibisz, Sreenivas Subramoney, Can Alkan, Saugata Ghose, and Onur Mutlu,

"GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework for Genome Sequence Analysis"

[Lightning Talk Video (1.5 minutes)]
[Lightning Talk Slides (pptx) (pdf)]
[Talk Video (18 minutes)]
[Slides (pptx) (pdf)]
Near-memory GenASM Framework

- **Our goal:** Accelerate approximate string matching (ASM) by designing a fast and flexible framework, which can accelerate multiple steps of genome sequence analysis.

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

- Modify and extend Bitap\(^1,2\), ASM algorithm with fast and simple bitwise operations, such that it now:
  - Supports long reads
  - Supports traceback
  - Is highly parallelizable

- **Co-design** of our modified scalable and memory-efficient algorithms with low-power and area-efficient hardware accelerators

---

Key Results of the GenASM Framework

(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)
Conclusion on Our Contributions

Near-memory/In-memory Pre-alignment Filtering
- GRIM-Filter [BMC Genomics‘18]
- GenASM [MICRO 2020]
- SneakySnake [IEEE Micro‘21]

Near-memory Sequence Alignment
- GenASM [MICRO 2020]

Specialized Pre-alignment Filtering Accelerators (GPU, FPGA)
- GateKeeper [Bioinformatics‘17]
- MAGNET [AACBB‘18]
- Shouji [Bioinformatics‘19]
- GateKeeper-GPU [arXiv‘21]
- SneakySnake [Bioinformatics‘20]
Conclusion on Ongoing Directions

- Read alignment can be **substantially accelerated** using computationally inexpensive and **accurate** pre-alignment **filtering** algorithms designed for specialized hardware.

- All the **three directions are used** by mappers today, but filtering has replaced alignment as the bottleneck.

- Pre-alignment filtering does **not** sacrifice any of the aligner capabilities, as it does **not** modify or replace the alignment step.
What else can be done?
What if we got a new version of the reference genome?

AirLift [Kim+, arXiv 2021]

Jeremie S. Kim, Can Firtina, Meryem Banu Cavlak, Damla Senol Cali, Mohammed Alser, Nastaran Hajinazar, Can Alkan, Onur Mutlu

[Source Code]
[Online link at arXiv]
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.
More Details on AirLift

Jeremie S. Kim, Can Firtina, Meryem Banu Cavlak, Damla Senol Cali, Mohammed Alser, Nastaran Hajinazar, Can Alkan, Onur Mutlu
[Source Code]
[Online link at arXiv]
Agenda for Today

- What is Genome Analysis?
- What is Intelligent Genome Analysis?
- How we Analyze Genome?
- What is Read Mapping?
- What Makes Read Mapper Slow?
- Algorithmic & Hardware Acceleration
  - Seed Filtering Technique
  - Pre-alignment Filtering Technique
  - Read Alignment Acceleration

- Where is Read Mapping Going Next?
Adoption of hardware accelerators in genome analysis
I have a major concern with the work that is actually not a problem with the manuscript at all. Specifically, I have the concern that there has been little to no adoption of previous specialized hardware solutions related to improving the speed of alignment. While there has been considerable work in this area (which the authors do an admirable job of citing), it does not seem that these hardware-based solutions have gained any type of real traction in the community, as the vast majority of alignment is still performed on “regular” CPUs, where the extent of hardware acceleration is the adoption of specific SIMD or vectorized instructions. While I don’t think that this practical concern should preclude publication of the current work, it is something worth considering (what, if any, of the proposed improvements to the SHD filter could be “back-ported” to a software-only solution).
Our Response

We see the reviewer’s point, but we do not believe this should be held against the research in the area of FPGA-based acceleration of read mapping in particular or genomics in general. It always takes time to adopt a “new” or “different” hardware technology since it requires investment into the hardware infrastructure. The main challenges/barriers that limit the popularity of FPGAs in the genomics field are the high cost, design effort, and development time. Due to the fact that the deliverable of such projects is normally a hardware product, researchers tend to commercialize their research with startup companies and engage themselves with industrial collaborators, as we describe below. Today, the cost structure of FPGAs is changing because major cloud infrastructures (e.g., by Microsoft Azure and Amazon AWS) offer FPGAs as core engines of the infrastructure. Therefore, we believe the benefits of FPGA-based acceleration has become available to many more folks in the community, especially with the open-source release of such FPGA-accelerated solutions. To increase adoption, we have decided to release our source code for GateKeeper. It is available on https://github.com/BilkentCompGen/GateKeeper.

Some examples of the research groups that commercialize their research and promote FPGA-based or even cloud-based products for genomics are as follows:
http://www.timelogic.com/catalog/775
http://www.edicogenome.com/dragen_bioit_platform/the-dragen-engine-2/
http://www.bcgsc.ca/platform/bioinfo/software/XpressAlign/releases/1.0
https://www.sevenbridges.com/amazon/
It is also important to emphasize that the necessity of designing a mapper on hardware is currently steering the field towards more personalized medicine. Hardware-accelerated mappers (using various platforms such as SIMD, GPUs, and FPGAs) are becoming increasingly popular as they can be potentially directly integrated into sequencing machines (the Illumina sequencer, for example, includes an FPGA chip inside it [https://support.illumina.com/content/dam/illumina-support/documents/downloads/software/hiseq/hcs_2-0-12/installnotes_hcs2-0-12.pdf]), such that we have a single machine that can perform both sequencing and mapping (Lindner, et al., Bioinformatics 2016). This approach has two benefits. First, it can hide the complexity and details of the underlying hardware from users who are not necessarily aware about FPGAs (e.g., biologists and mathematicians). Second, it allows a significant reduction in total genome analysis time by starting read mapping while still sequencing. Hence, an end user or researcher in genomics might not directly deal with the “pre-alignment on FPGA” or “mapper on FPGA”, but they might purchase a sequencer that performs pre-alignment and alignment using FPGAs inside. As such, one potential target of our research is to influence the design of more intelligent sequencing machines by integrating GateKeeper inside them.

In fact, we believe GateKeeper is very suitable to be used as part of a sequencer as it provides a complete pre-alignment system that includes many processing cores, where all processing cores work in parallel to provide extremely fast filtering. We believe such a fast approach can make sequencers more intelligent and attractive.
Dream and, they will come

Computing landscape is very different from 10-20 years ago
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’s startup in 2018 and joined NVIDIA in 2020

(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>1,200 minutes</td>
<td>52 minutes</td>
<td>35 minutes</td>
<td>23 minutes</td>
<td></td>
</tr>
</tbody>
</table>
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
- Sequencing Machine
- Persistent Memory/Storage
Cerebras’s Wafer Scale Engine (2019)

- The largest ML accelerator chip
- 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/
TESLA Full Self-Driving Computer (2019)

- ML accelerator: 260 mm², 6 billion transistors, 600 GFLOPS GPU, 12 ARM 2.2 GHz CPUs.
- Two redundant chips for better safety.

https://youtu.be/Ucp0TTmvqOE?t=4236

Onur Mutlu, Computer Architecture Lecture 2b, Fall 2019, ETH Zurich
Where is Read Mapping Going Next?

Will 100% accurate genome-long reads alleviate/eliminate the need for read mapping?

Think about metagenomics, pan-genomics, ...
Lecture 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
  - In particular, **read mapping**

- Many bottlenecks exist in accessing and manipulating **huge amounts of genomic data** during analysis

- We cover various **recent ideas to accelerate read mapping**
  - A journey since September 2006
Key Takeaways

- **Population-scale analyses** are not an easy task

- You need to consider **many** things in designing a new system + have good **intuition/insight into ideas/tradeoffs**

- But, it is fun and can be **very rewarding/impactful**

- And, enables a great future
  - It has large scientific, medical, societal, personal implications

- **Very hot topic for graduate studies and research!**
Key Conclusion

Most speedup comes from parallelism enabled by novel architectures and algorithms.
Acknowledgments

- Many colleagues and collaborators
  - Damla Senol Cali, Jeremie Kim, Hasan Hassan, Can Firtina, Juan Gómez Luna, Hongyi Xin, ...

- Funders:
  - NIH and Industrial Partners (Alibaba, AMD, Google, Facebook, HP Labs, Huawei, IBM, Intel, Microsoft, Nvidia, Oracle, Qualcomm, Rambus, Samsung, Seagate, VMware)

- All papers, source code, and more are at:
  - https://people.inf.ethz.ch/omutlu/projects.htm
Work With Us

- If you are already a student at ETH and are interested in doing research with SAFARI research group on similar topics, **Talk to me:**

  - ALSERM@ethz.ch
Openings @ SAFARI

- We are hiring enthusiastic and motivated students and researchers at all levels.

- Join us now: safari.ethz.ch/apply
Recommended Readings

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

"Accelerating Genome Analysis: A Primer on an Ongoing Journey"

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]
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)]

Our Contributions

- Near-memory/In-memory Pre-alignment Filtering
  - GRIM-Filter [BMC Genomics’18]
  - SneakySnake [IEEE Micro’21]
  - GenASM [MICRO 2020]

- Specialized Pre-alignment Filtering Accelerators (GPU, FPGA)
  - GateKeeper [Bioinformatics’17]
  - MAGNET [AACBB’18]
  - Shouji [Bioinformatics’19]
  - GateKeeper-GPU [arXiv’21]
  - SneakySnake [Bioinformatics’20]

Accelerating Genome Analysis: A Primer on an Ongoing Journey - RECOMB 2021 talk by Mohammed Alser
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"
  Invited Lecture at Technion, Virtual, 26 January 2021.
  [Slides (pptx) (pdf)]
  [Talk Video (1 hour 37 minutes, including Q&A)]
  [Related Invited Paper (at IEEE Micro, 2020)]
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=XoLpzmN-Pas&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=15](https://www.youtube.com/watch?v=XoLpzmN-Pas&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=PL5Q2soXY2Zi9E2bBVAgCgLgwiDRQDTyId](https://www.youtube.com/watch?v=rgjl8ZyLsAg&list=PL5Q2soXY2Zi9E2bBVAgCgLgwiDRQDTyId)

SAFARI

[https://www.youtube.com/onurmutlulectures](https://www.youtube.com/onurmutlulectures)
Prior Research on Genome Analysis (1/2)


- Alser + "SneakySnake: A Fast and Accurate Universal Genome Pre-Alignment Filter for CPUs, GPUs, and FPGAs.", *Bioinformatics*, 2020.


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

Alser+, "MAGNET: understanding and improving the accuracy of genome pre-alignment filtering”, IPSI Transaction, 2017.
Computer Architecture

Lecture 10:
Intelligent Genome Analysis

Dr. Mohammed Alser
@mealser

ETH Zurich
Fall 2021
29 October 2021