

SquiggleFilter

An Accelerator for Portable Virus Detection

Tim Dunn, Harisankar Sadasivan, Jack Wadden, Kush Goliya,
Kuan-Yu Chen, David Blaauw, Reetuparna Das, and Satish Narayanasamy

University of Michigan

MICRO '21

Presented by: Joël Lindegger
joel.lindegger@inf.ethz.ch

23/12/2021

Executive Summary

Motivation: Portable nanopore sequencers enable **virus detection early** in a pandemic, before PCR or antigen tests are available

Problem:

- Sequencing output must be **labeled as “viral” or “non-viral”** in real-time
- Existing solutions are **too slow by $\geq 2x$** and **will not scale** for future sequencers
- Most of the work done by existing solutions is **unnecessary**

Goal:

- Accelerate the labelling process to keep up even with future sequencers

Challenge:

- *Basecalling* is a necessary step in the pipeline and the main **bottleneck**
- It is **data-dependent**, tools may become invalid every 6 months, thus people are discouraged to improve it

Key Ideas:

- **Filter** data before the basecalling step (in raw signal space)
- accelerate the filter in **hardware**

Results:

- 274x more throughput
- 2x less power usage

(baseline: Guppy-lite on NVIDIA Jetson AGX Xavier)

Outline

1. Background and Motivation
 1. Virus detection
 2. Nanopore sequencing
2. SquiggleFilter
 1. Pipeline
 2. Accelerator
3. Results
4. Strengths/Weaknesses
5. Discussion

Outline

1. Background and Motivation

1. Virus detection
2. Nanopore sequencing

2. SquiggleFilter

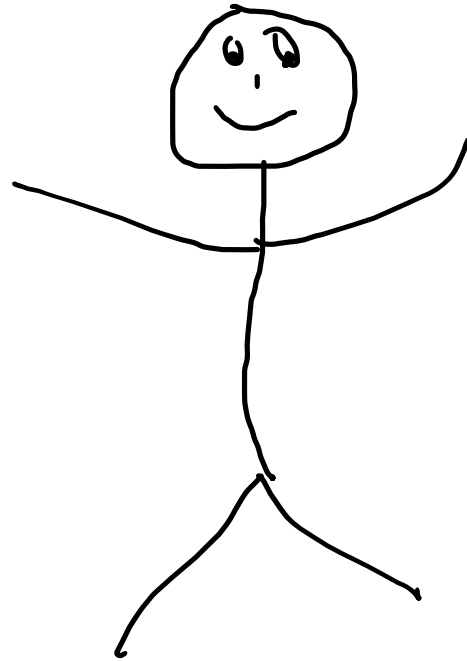
1. Pipeline
2. Accelerator

3. Results

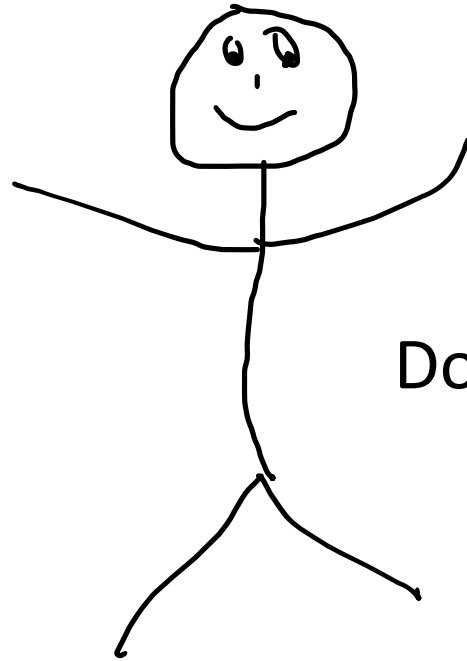
4. Strengths/Weaknesses

5. Discussion

Virus Detectors



Virus Detectors



Does he have COVID-19?

Virus Detectors

Non-Programmable

Programmable

Virus Detectors

Non-Programmable

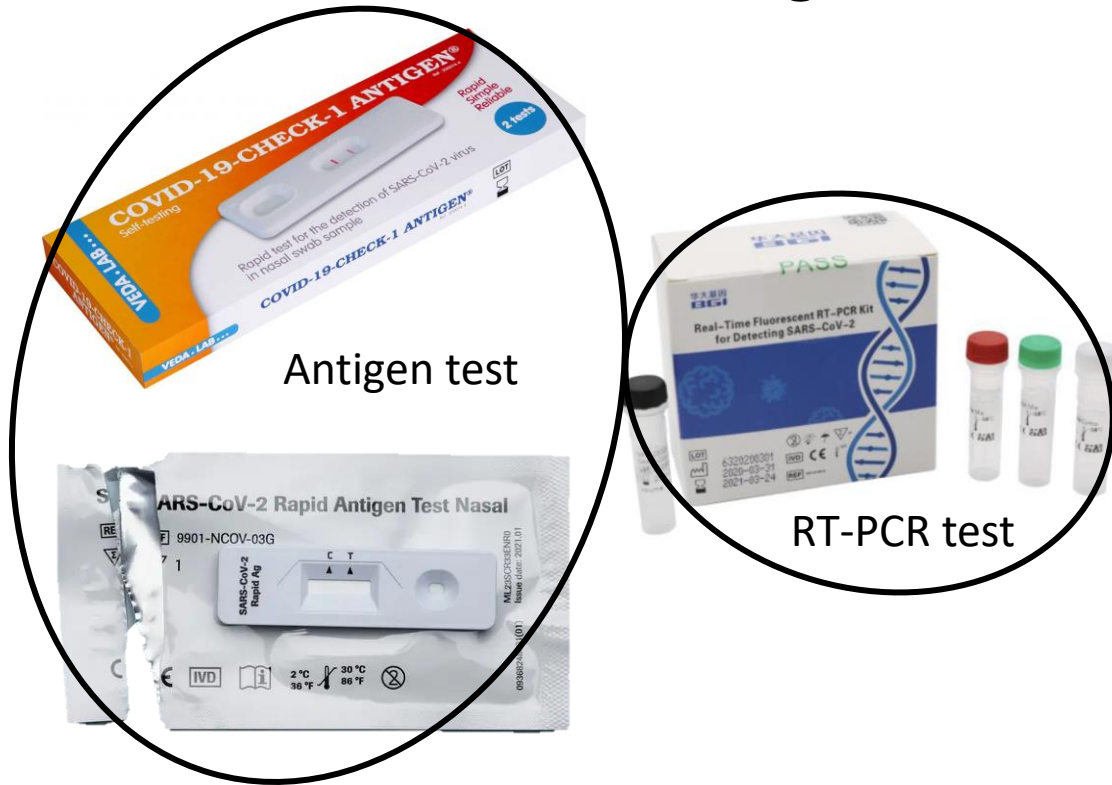
Programmable



Virus Detectors

Non-Programmable

Programmable

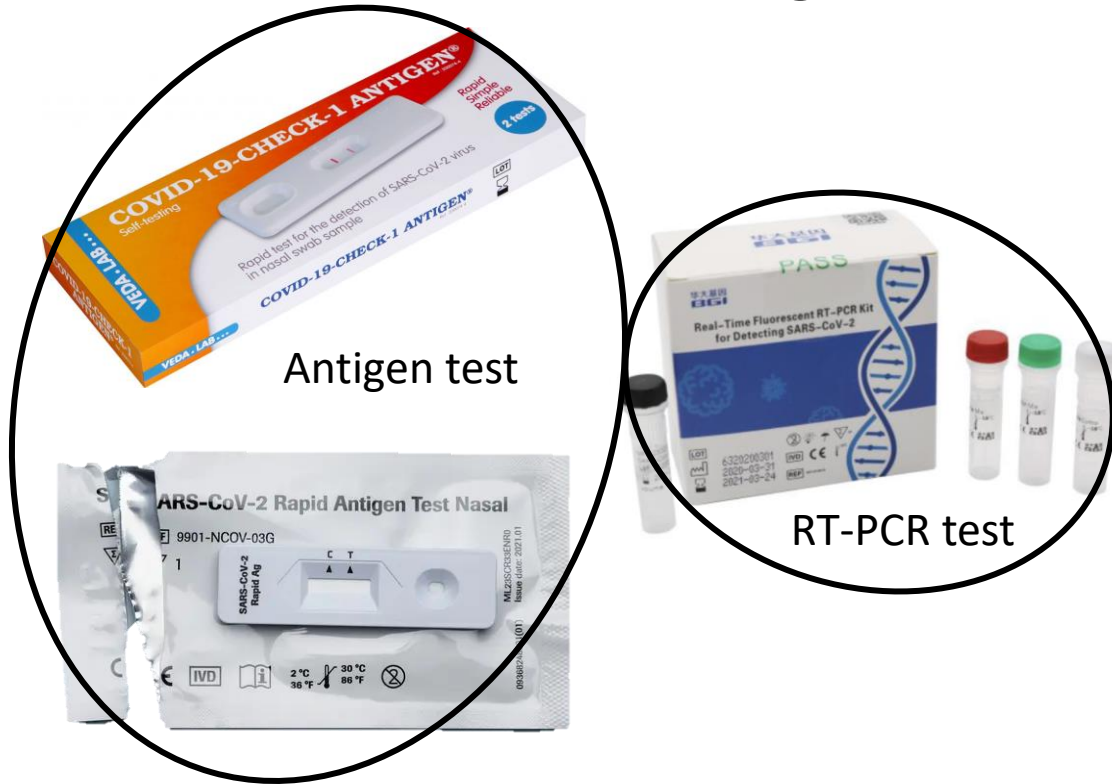


Antigen test

RT-PCR test

Virus Detectors

Non-Programmable



Programmable

ONT MinION Sequencer



Illumina Sequencer



Motivation

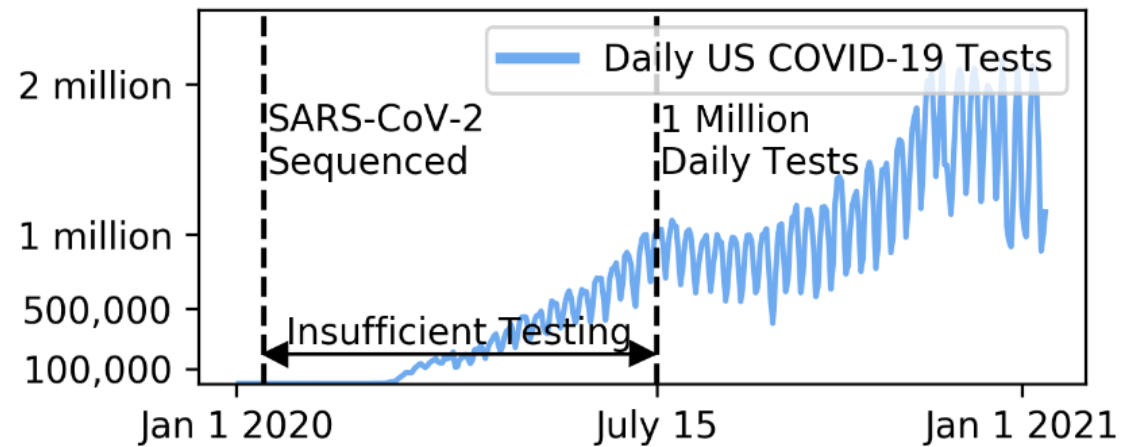


Figure 2: Progression of US COVID-19 testing [15]

Motivation

Non-programmable virus detectors are **slow to deploy**.

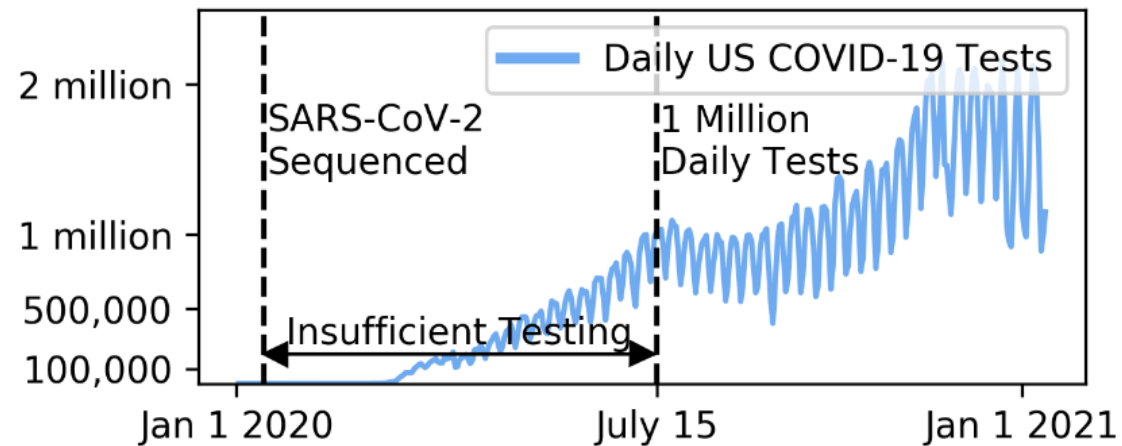


Figure 2: Progression of US COVID-19 testing [15]

Motivation

Non-programmable virus detectors are **slow to deploy**.

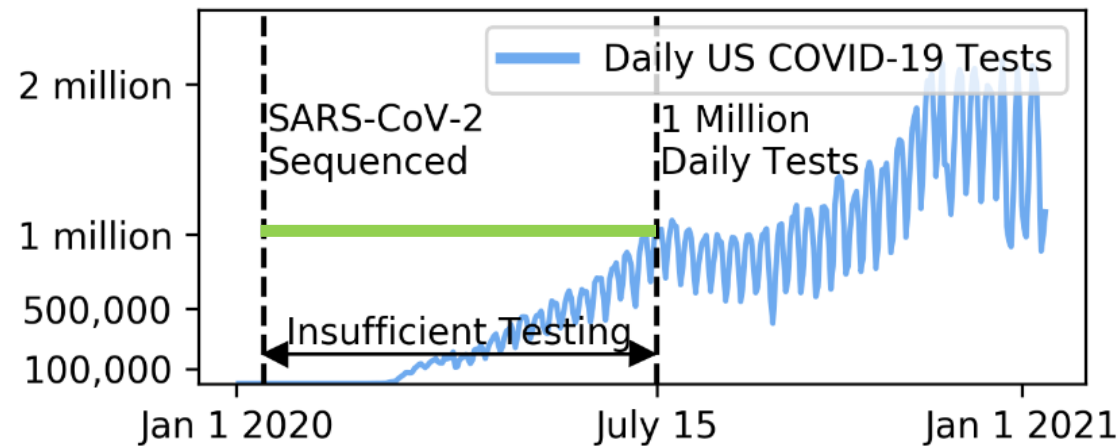


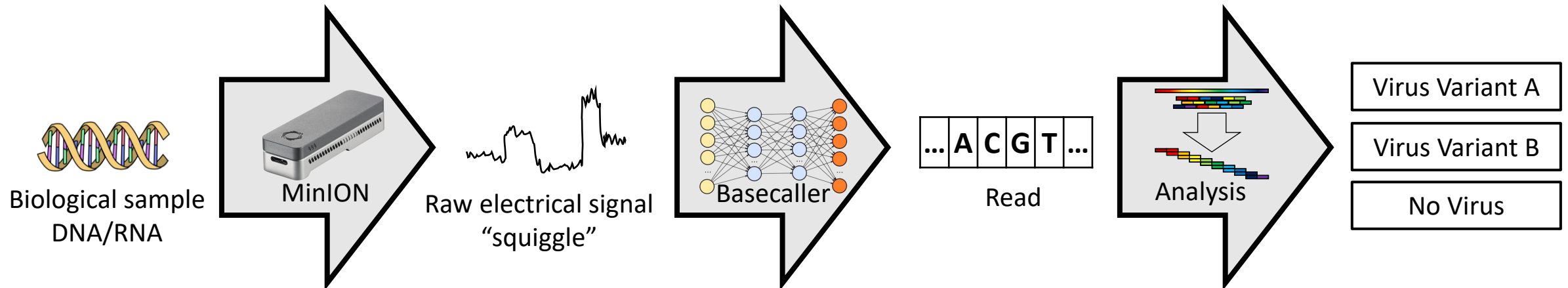
Figure 2: Progression of US COVID-19 testing [15]

Programmable virus detectors can help bridge the gap early in a pandemic.

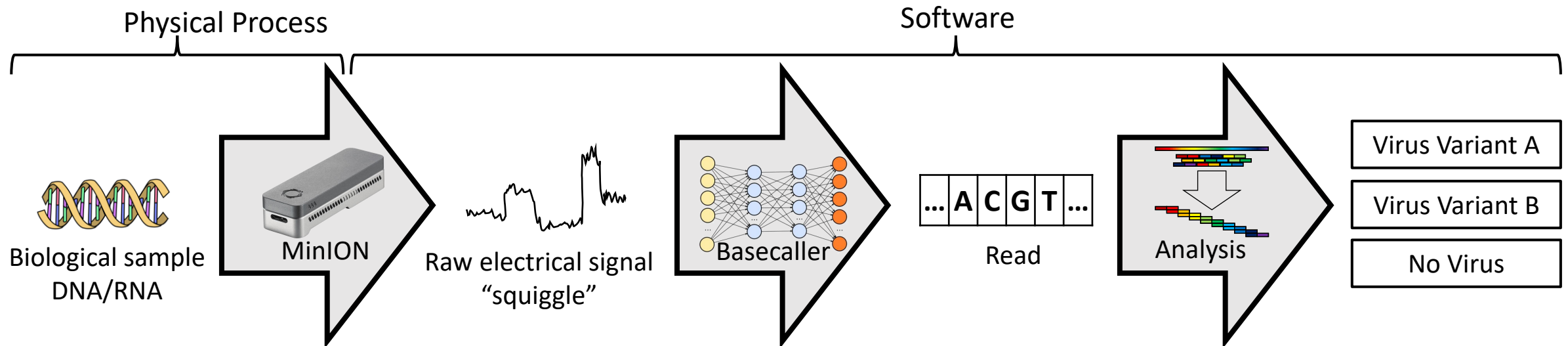
Outline

1. Background and Motivation
 1. Virus detection
 2. Nanopore sequencing
2. SquiggleFilter
 1. Pipeline
 2. Accelerator
3. Results
4. Strengths/Weaknesses
5. Discussion

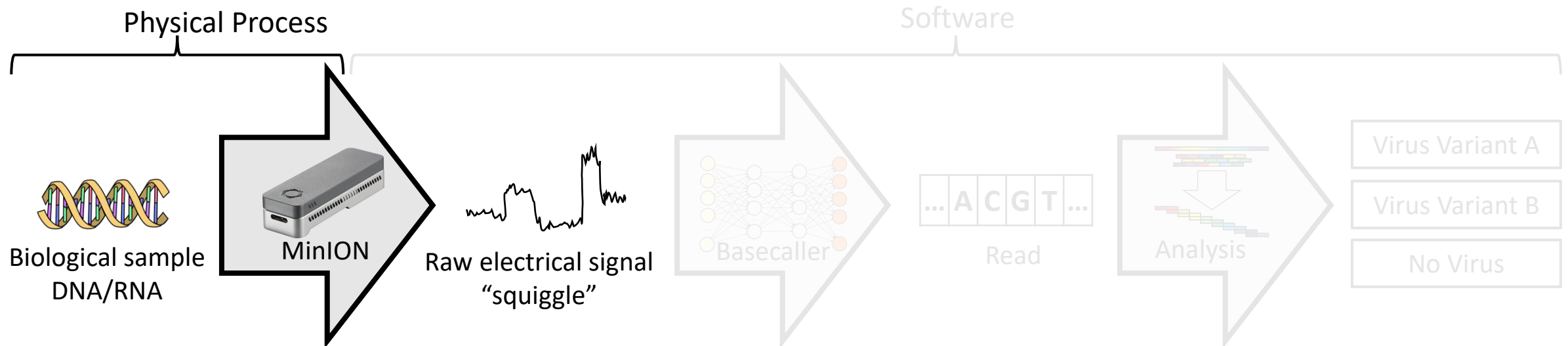
Nanopore Sequencing Overview



Nanopore Sequencing Overview



Nanopore Sequencing Overview



MinION



MinION



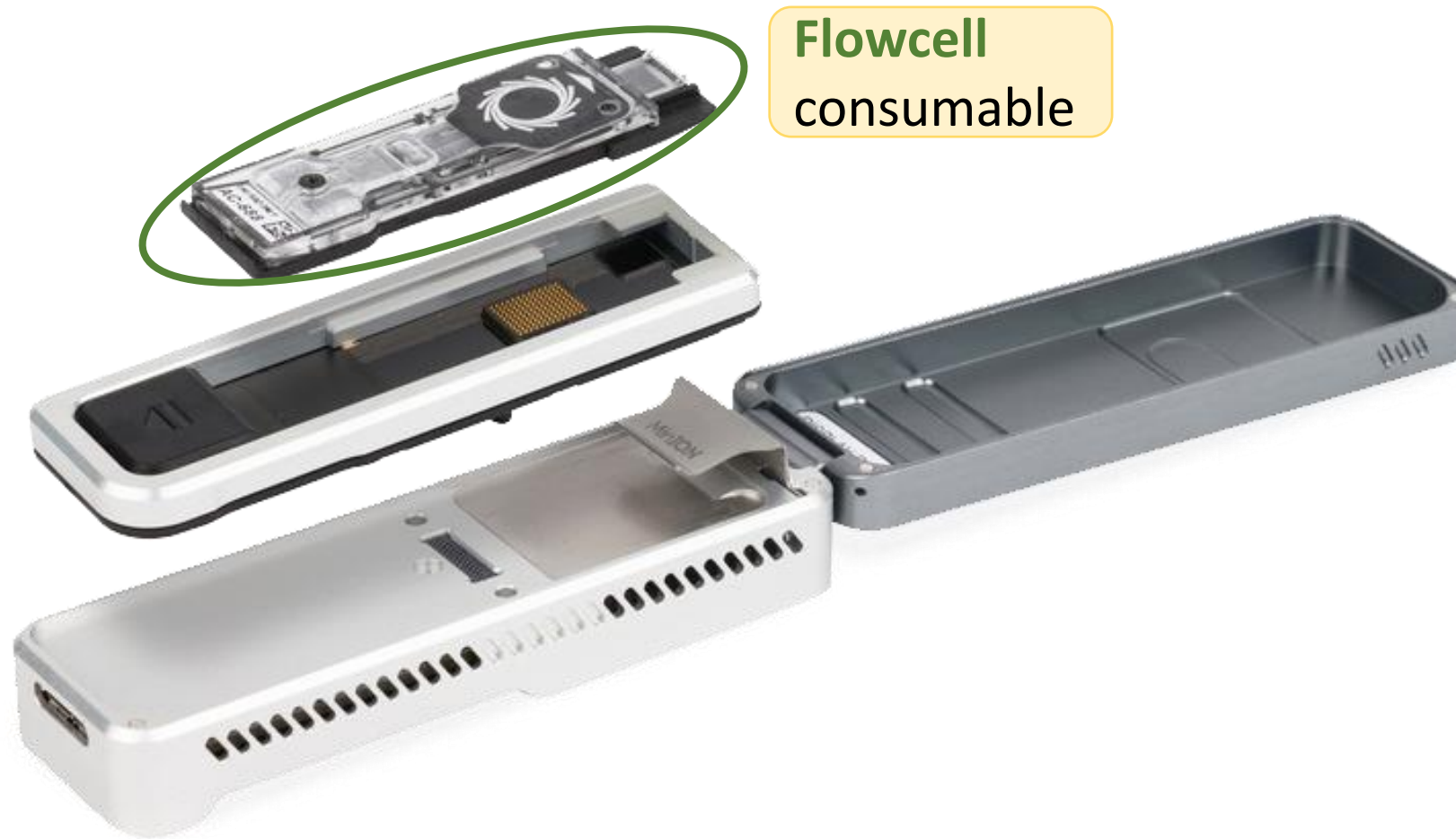
MinION is

- cheap (\$1000 + \$hundreds for consumables)
- portable
- and supports ReadUntil

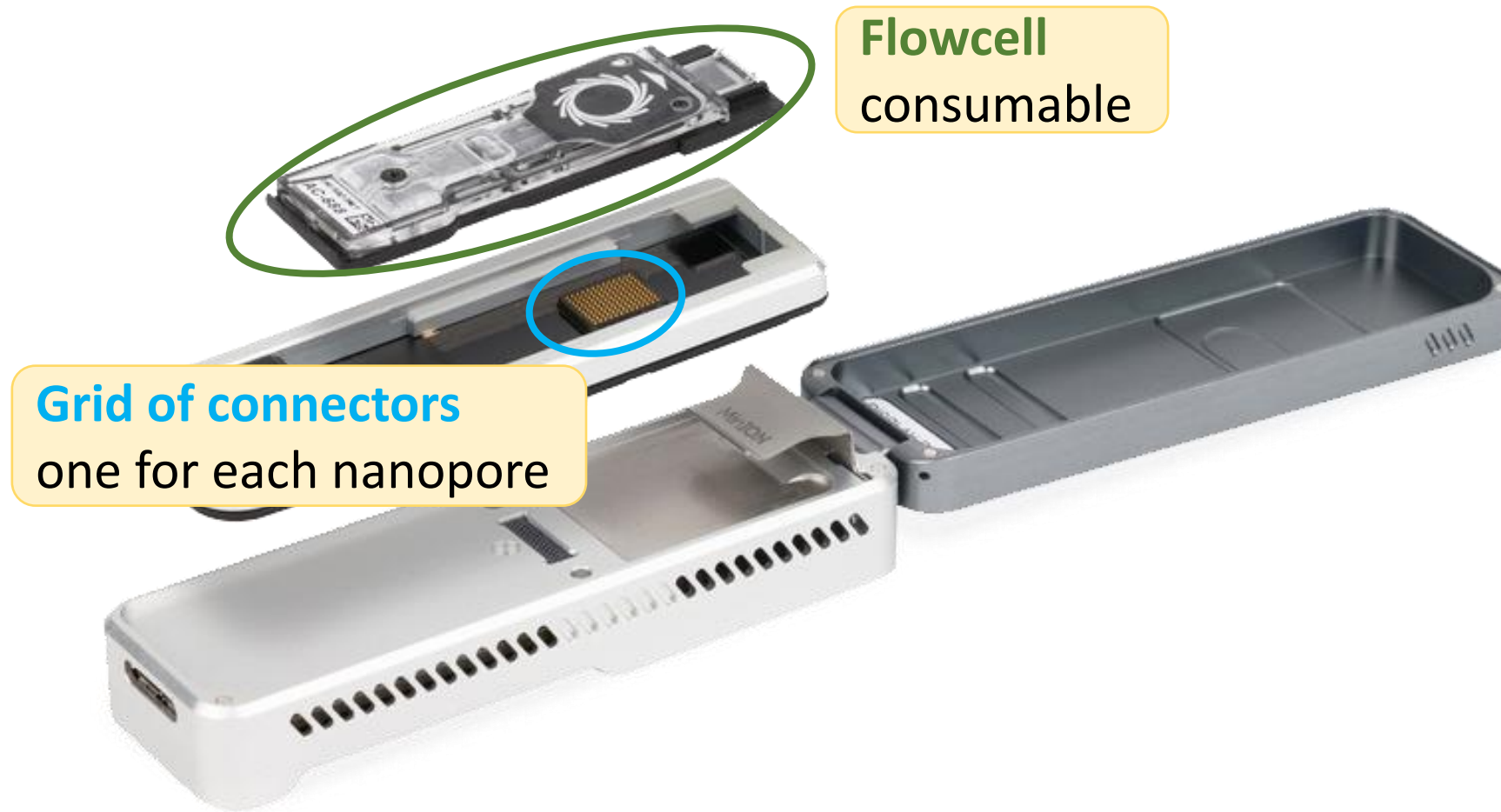
MinION



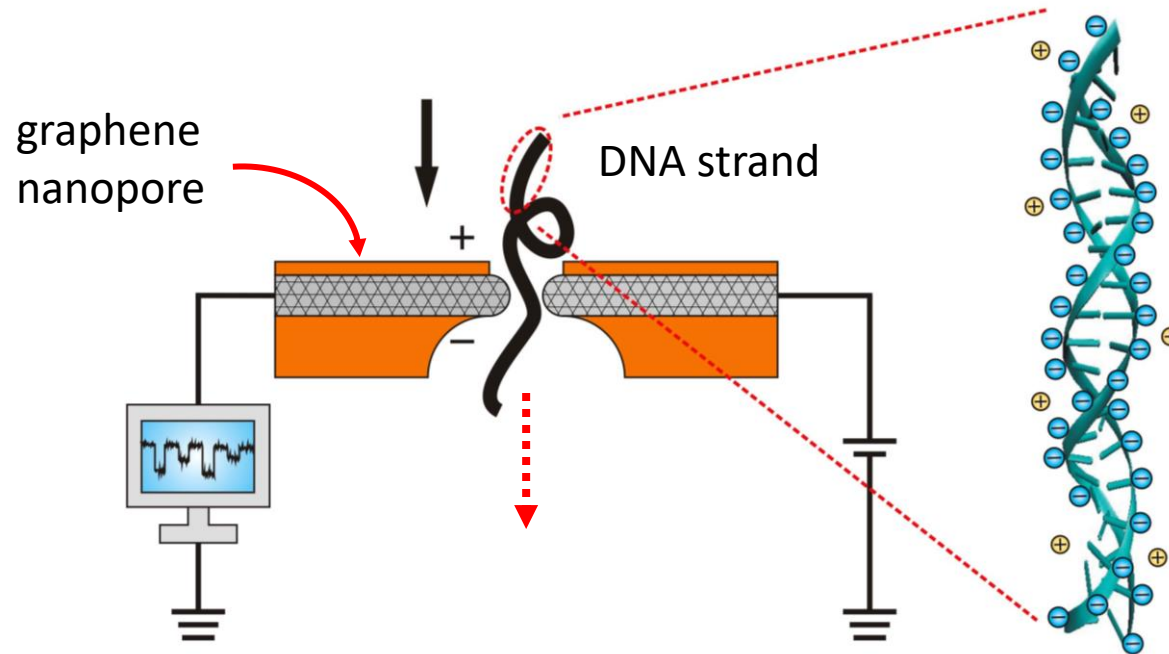
MinION



MinION



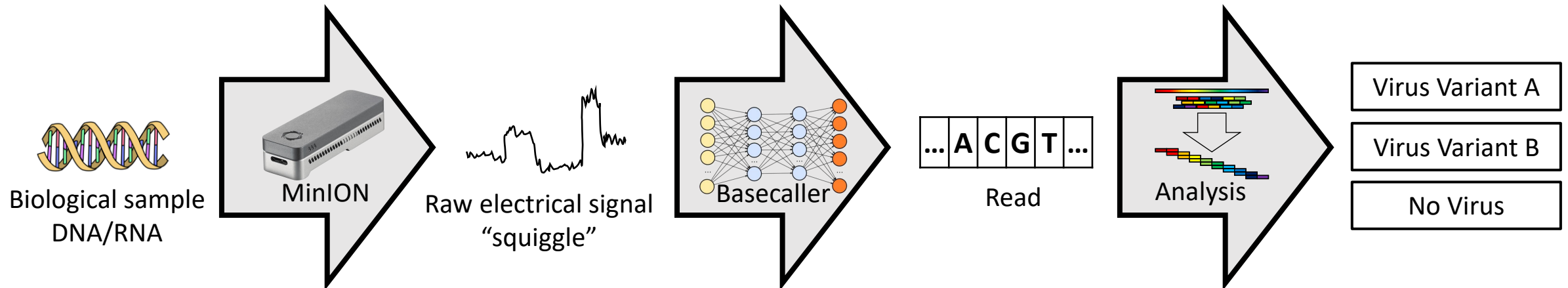
How Do Nanopores Work?



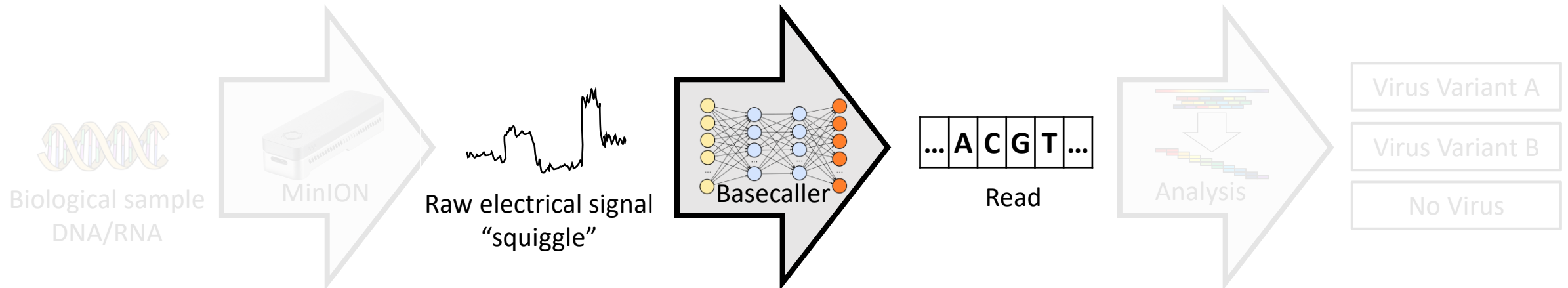
- **Nanopore** is a nano-scale hole (<20nm).
- In nanopore sequencers, an **ionic current** passes through the nanopores
- When the DNA strand passes through the nanopore, the sequencer measures the **change in current**
- This change is used to identify the bases in the strand with the help of **different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

Figure is adapted from: <https://phys.org/news/2013-12-gene-sequencing-future.html>

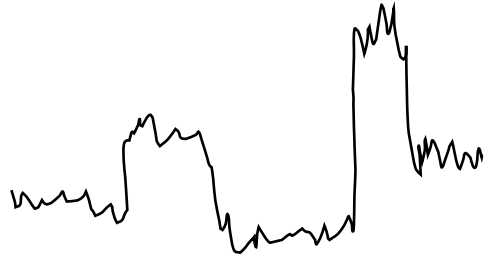
Nanopore Sequencing Overview



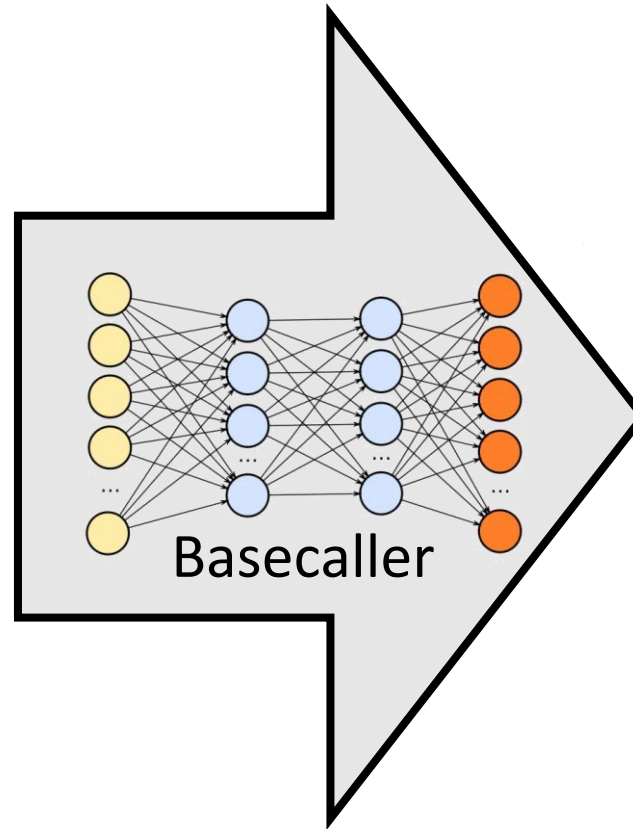
Nanopore Sequencing Overview



Basecalling



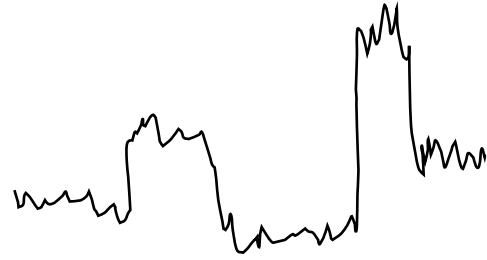
Raw electrical signal
“squiggle”



...	A	C	G	T	...
-----	---	---	---	---	-----

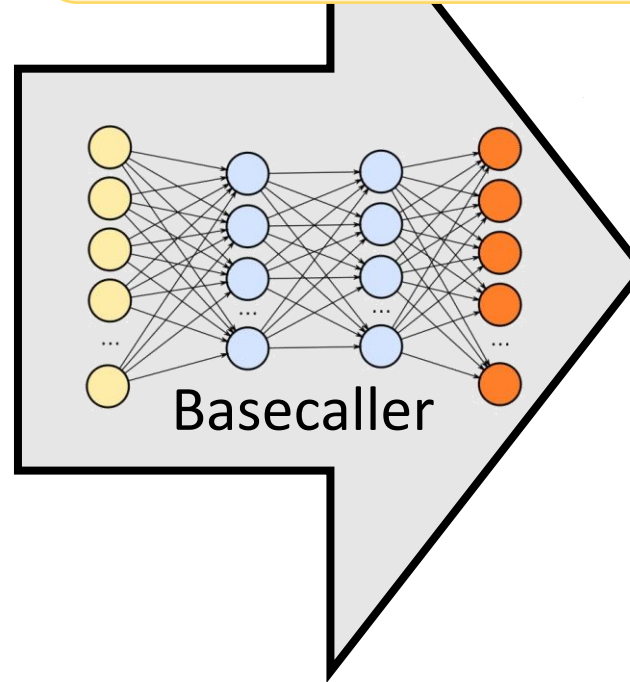
Read

Basecalling



Raw electrical signal
“squiggle”

State-of-the-art basecallers use computationally expensive **deep neural networks** to convert the electrical signal to nucleotide bases.



...	A	C	G	T	...
-----	---	---	---	---	-----

Read

Basecalling Cost

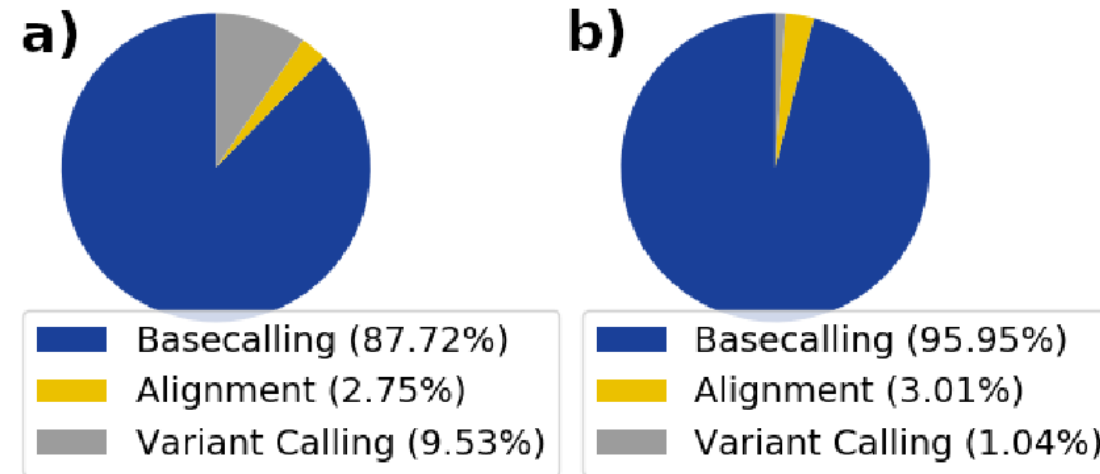


Figure 5: Basecalling is the bottleneck in a Read Until assembly of a SARS-CoV2 genome from specimens with a) 1%, and b) 0.1% viral reads.

Basecalling Cost

Basecalling is the **sole bottleneck** of the computational steps.

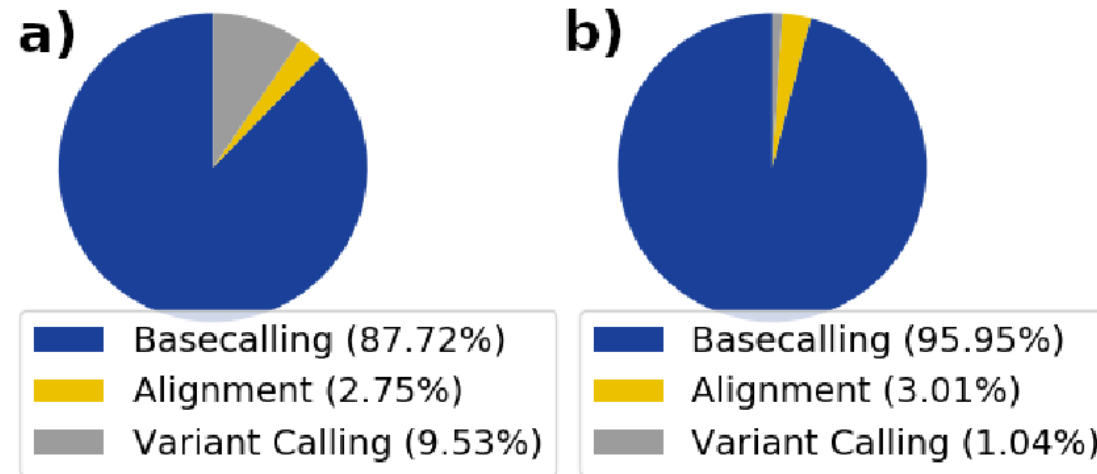
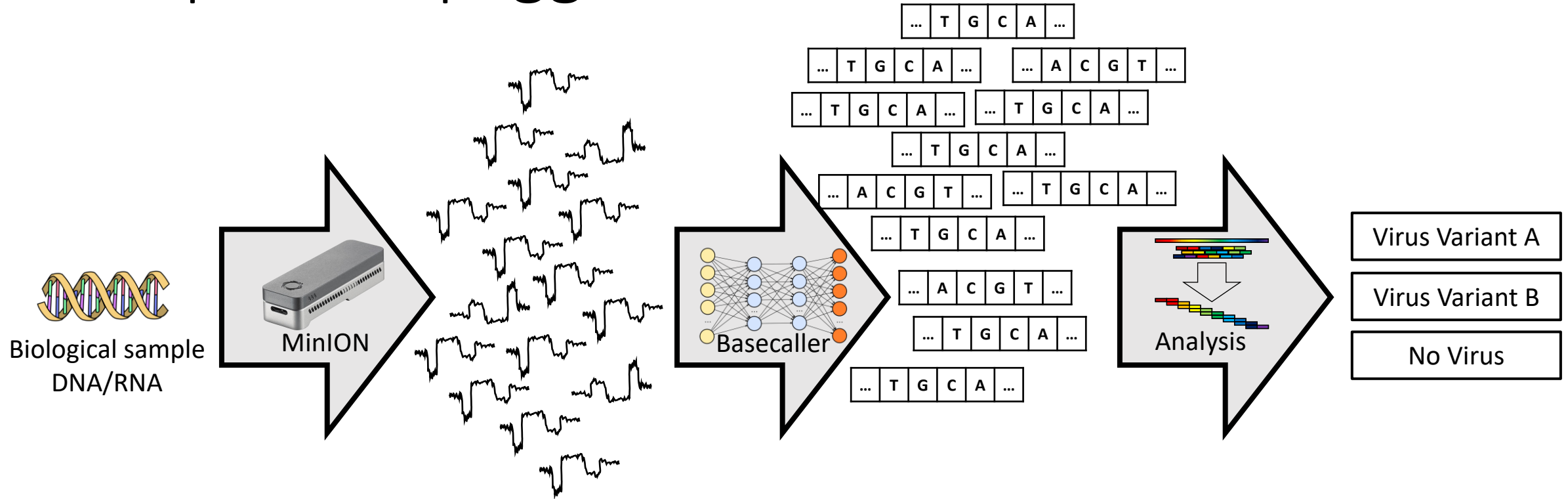
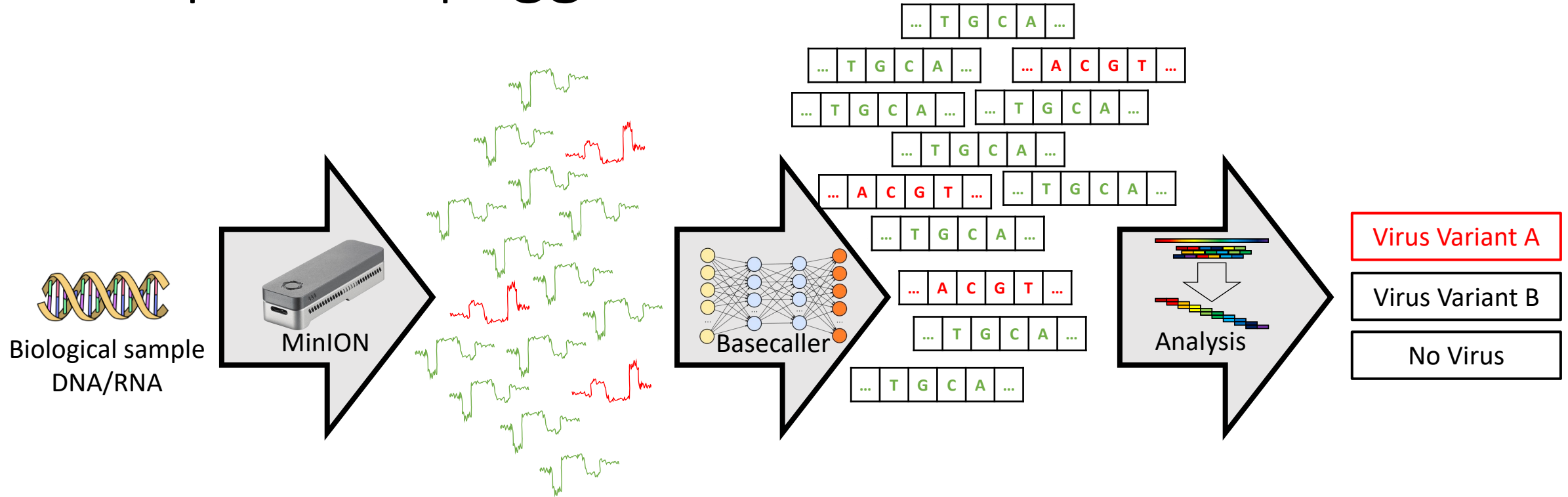


Figure 5: Basecalling is the bottleneck in a Read Until assembly of a SARS-CoV2 genome from specimens with a) 1%, and b) 0.1% viral reads.

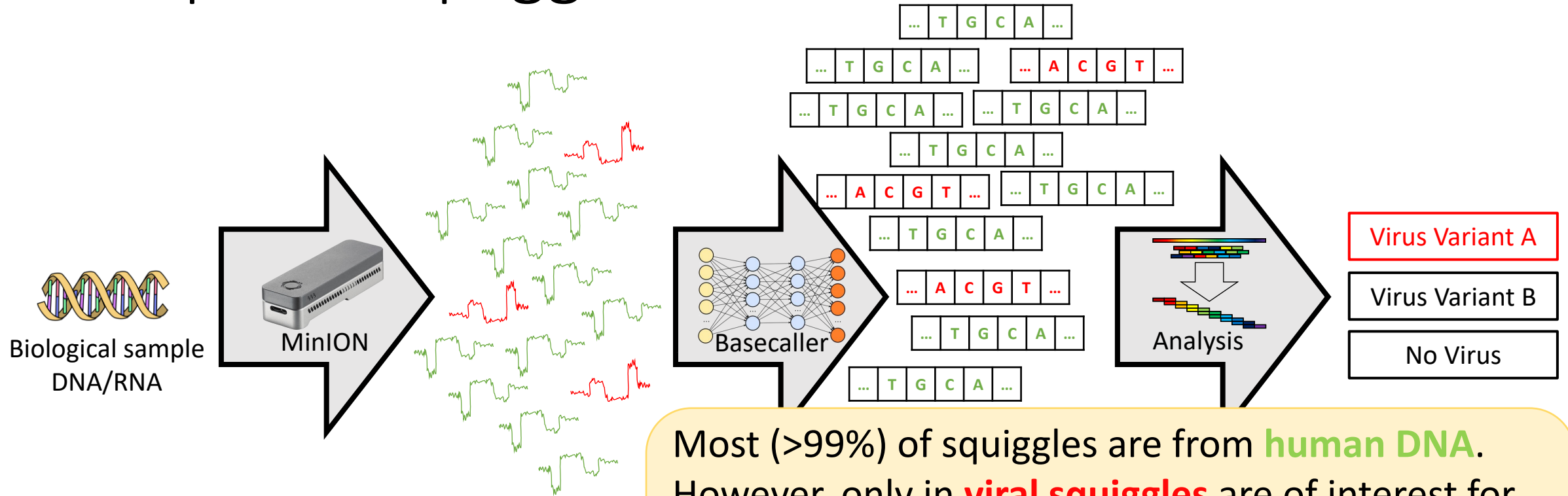
Required Squiggles



Required Squiggles



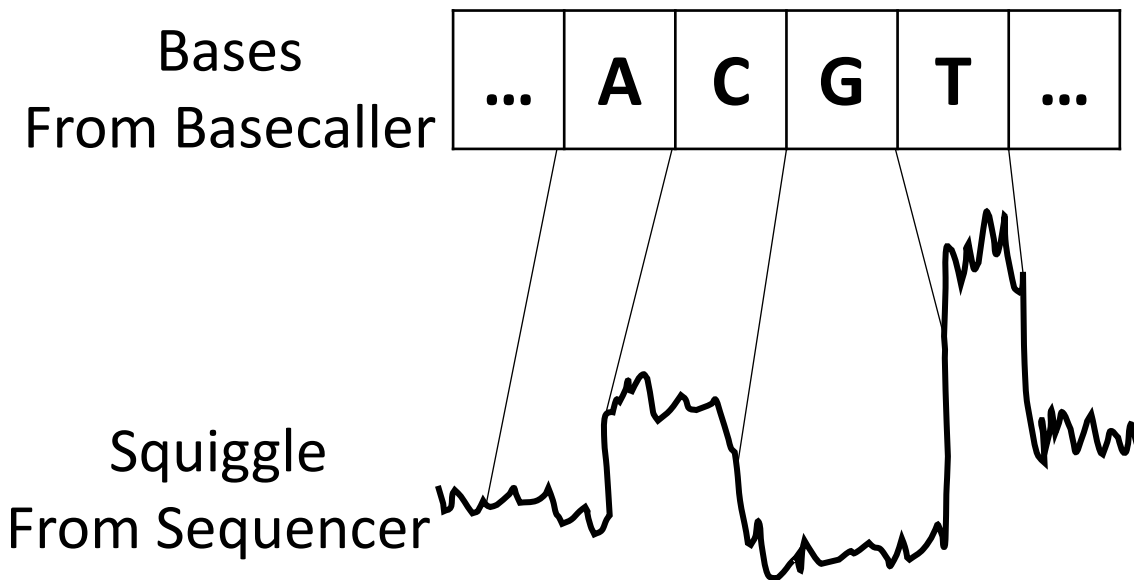
Required Squiggles



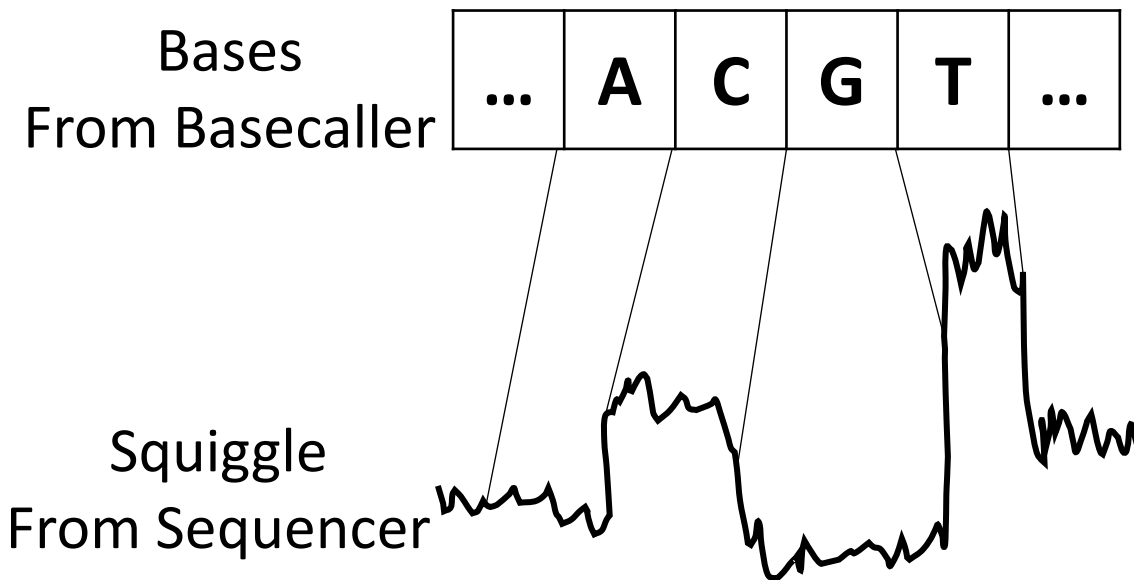
Most (>99%) of squiggles are from **human DNA**. However, only in **viral squiggles** are of interest for virus detection.

Thus, most of the basecaller's work is **unnecessary**.

Without ReadUntil

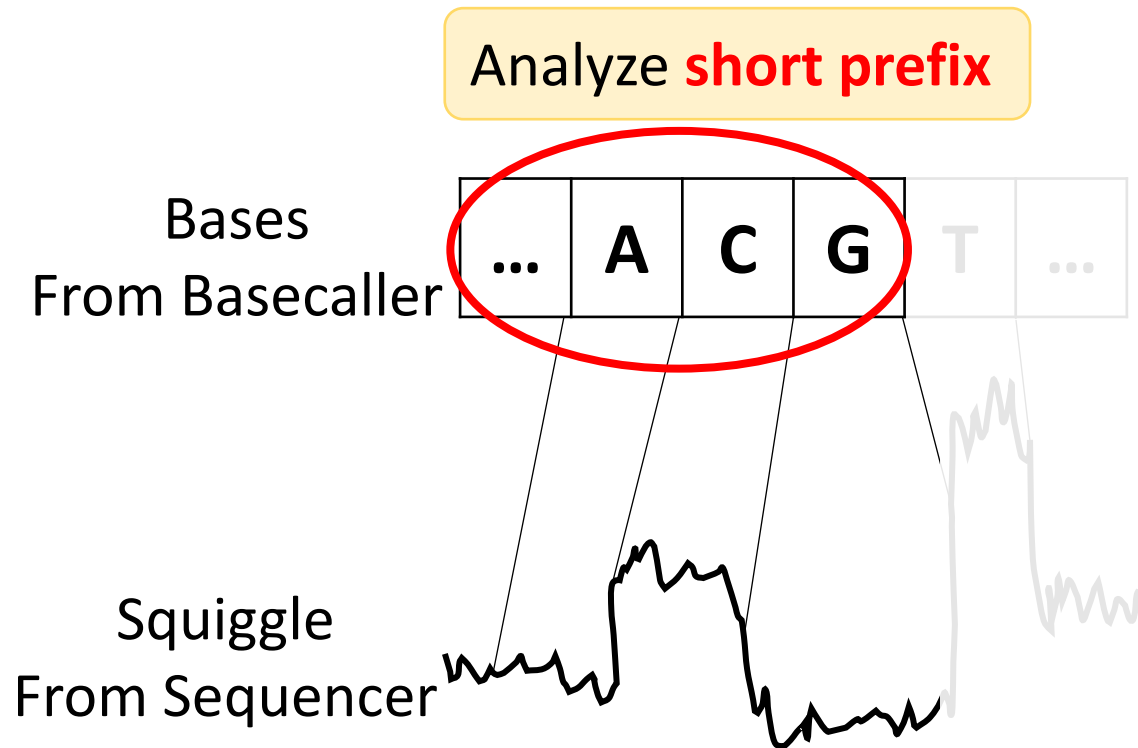


Without ReadUntil

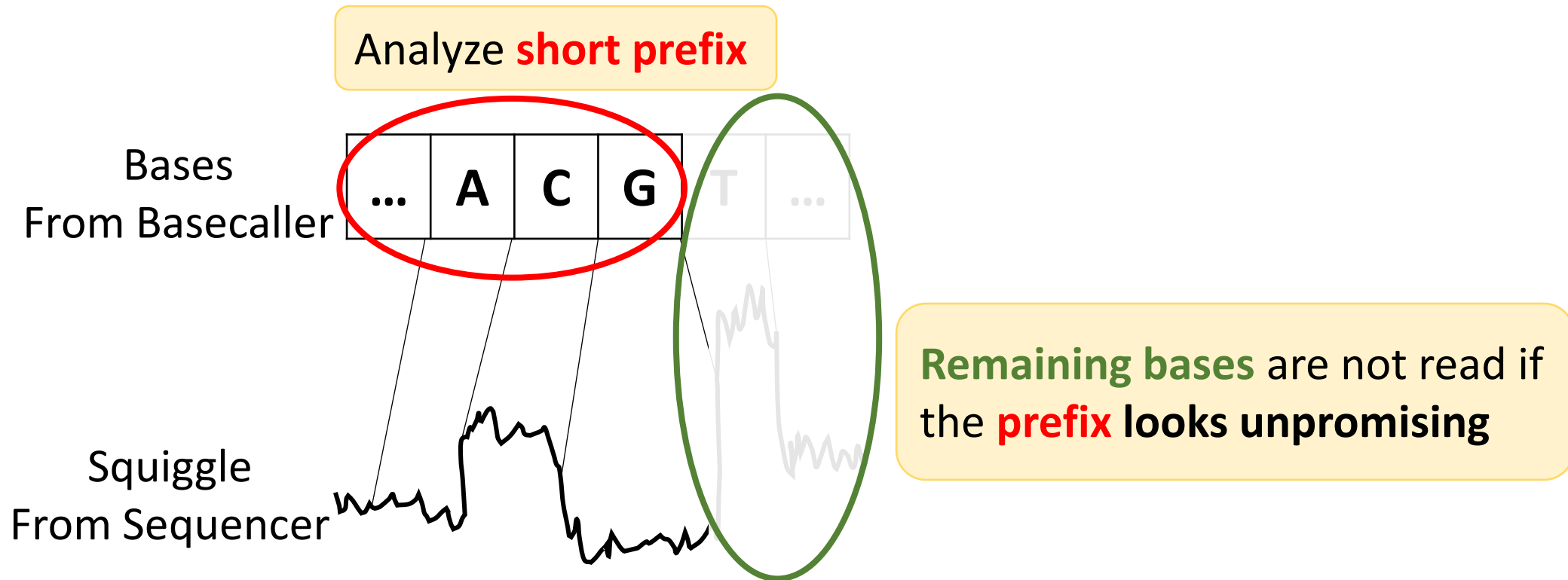


Without ReadUntil, the **entire squiggle** is processed by the machine.

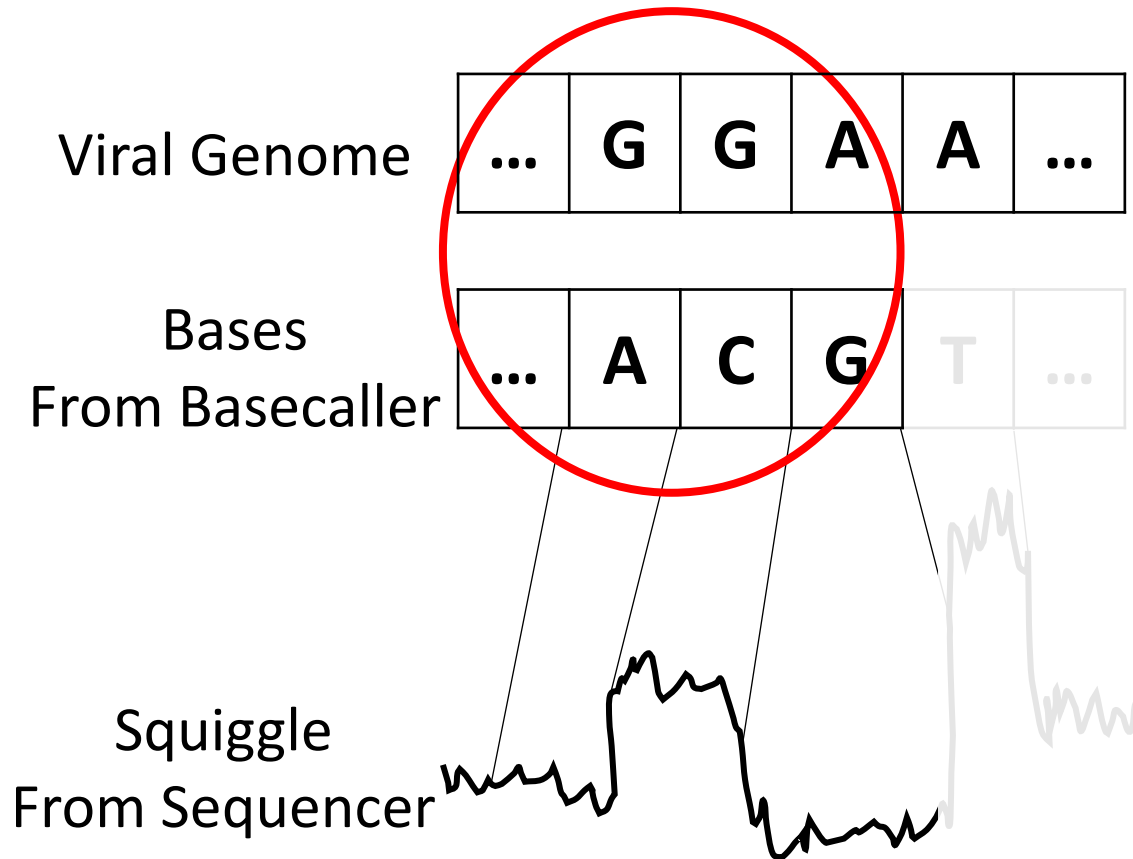
Without ReadUntil



With ReadUntil

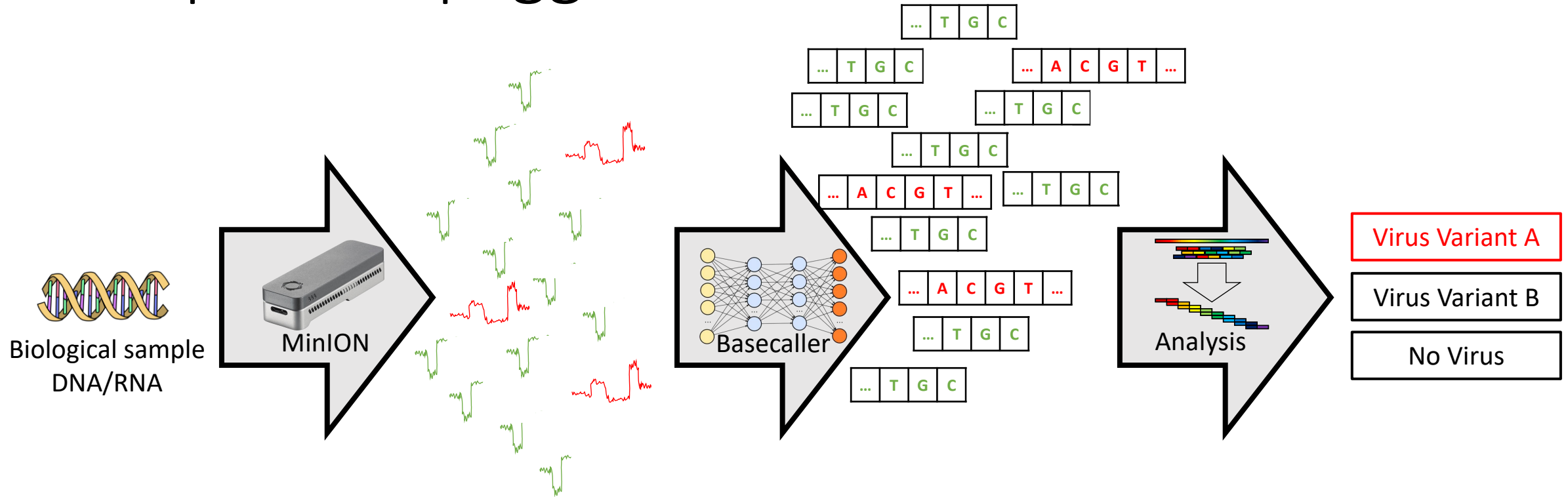


With ReadUntil



For example, **compare the prefix** to the virus genome we are looking for

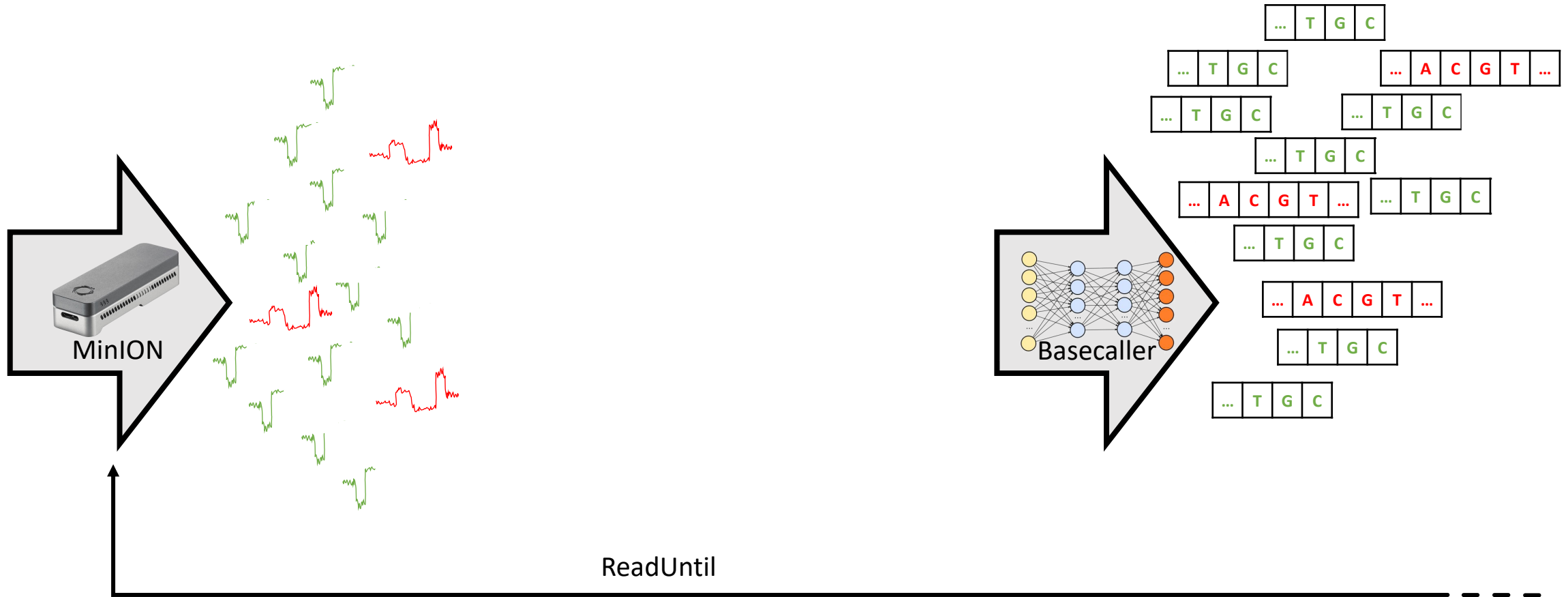
Required Squiggles



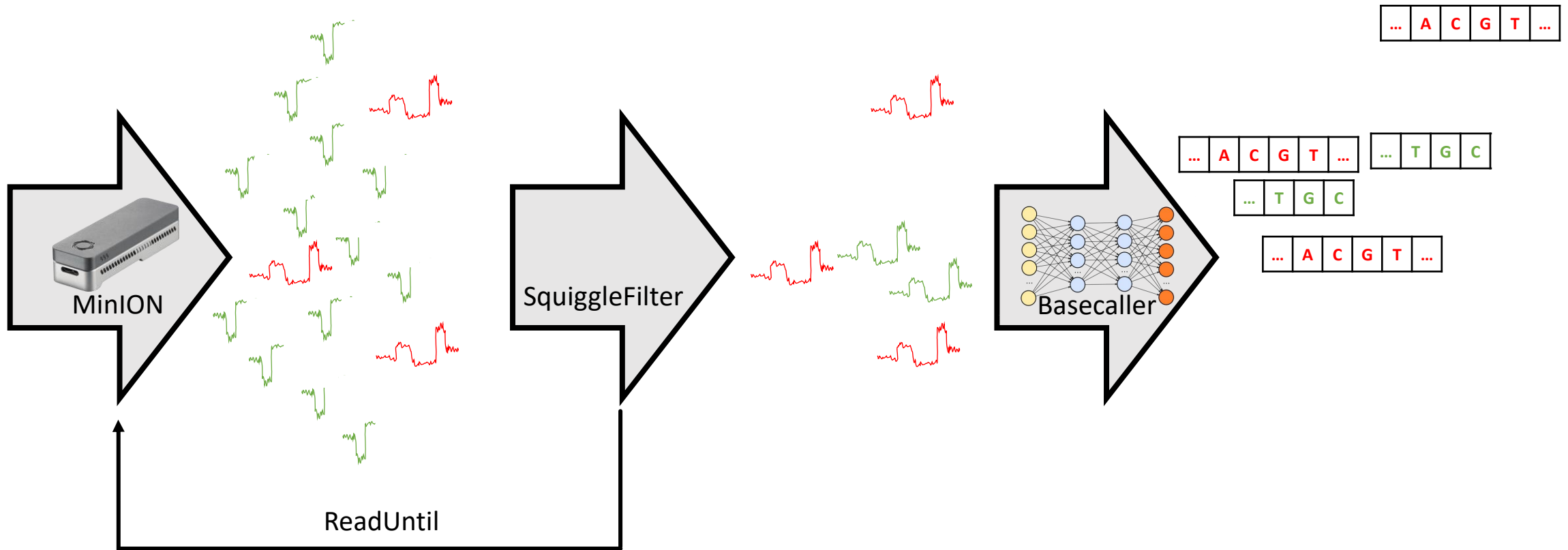
Outline

1. Background and Motivation
 1. Virus detection
 2. Nanopore sequencing
2. SquiggleFilter
 1. Pipeline
 2. Accelerator
3. Results
4. Strengths/Weaknesses
5. Discussion

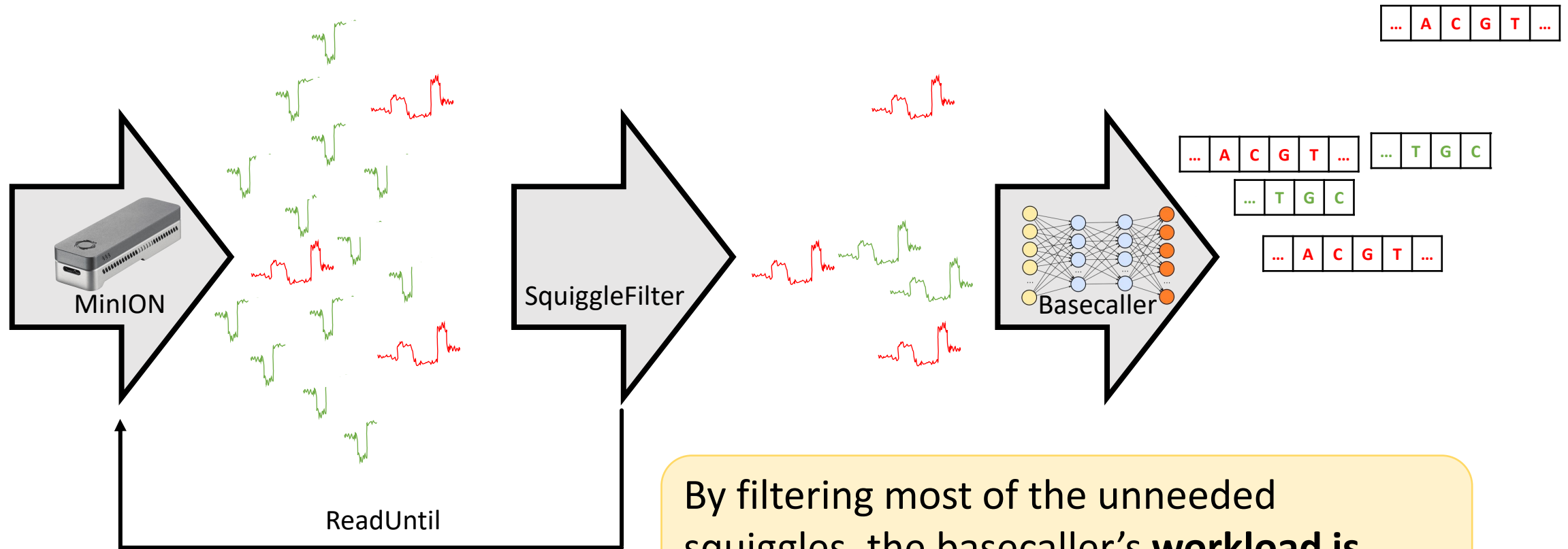
Before SquiggleFilter



With SquiggleFilter



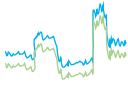


With SquiggleFilter




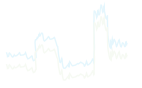
By filtering most of the unneeded squiggles, the basecaller's **workload is reduced** significantly.

SquiggleFilter

1. Translate **reference** genome to **expected squiggle**
2. Normalize **query squiggle**  → 
3. **Compare** input and reference signal using **dynamic time warping** 
4. If they match: keep analyzing
If not: reject immediately



SquiggleFilter

1. Translate **reference** genome to **expected squiggle**
2. Normalize **query squiggle** 
3. **Compare** input and reference signal using **dynamic time warping** 
4. If they match: keep analyzing
If not: reject immediately



Expected Squiggle

6-mer	current
AAAAAA	86.486
AAAAAC	83.949
AAAAAG	85.475
AAAAAT	84.424
AAACA	77.097

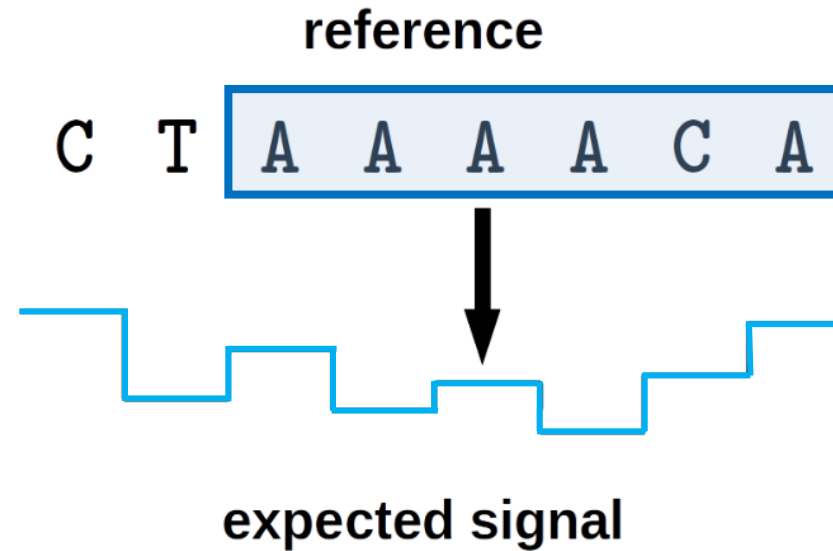


Figure 7: Aligning reference bases to expected currents.

Expected Squiggle

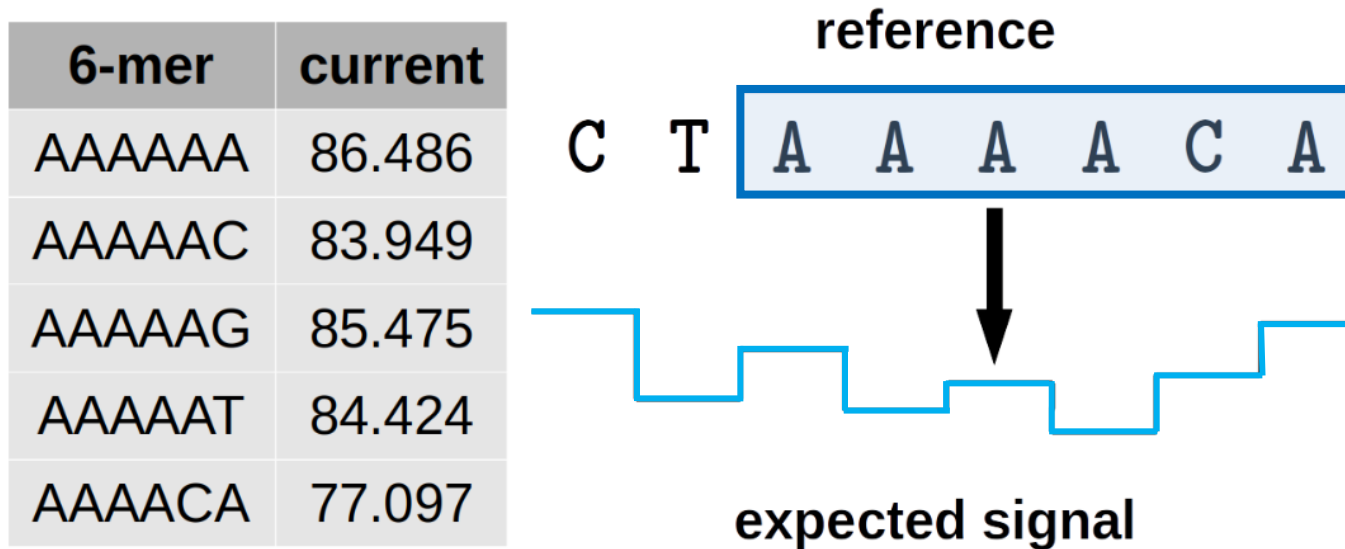
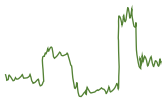

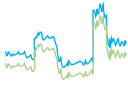
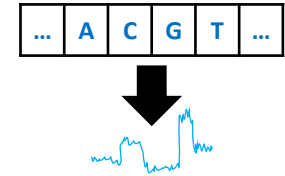


Figure 7: Aligning reference bases to expected currents.


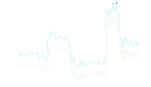
Each **reference** 6-mer corresponds to a current level, the mapping is provided by the manufacturer. The **expected signal** can be obtained through **lookups from the table**.

SquiggleFilter

1. Translate **reference** genome to **expected squiggle**
2. Normalize **query squiggle**  → 
3. **Compare** input and reference signal using **dynamic time warping** 
4. If they match: keep analyzing
If not: reject immediately

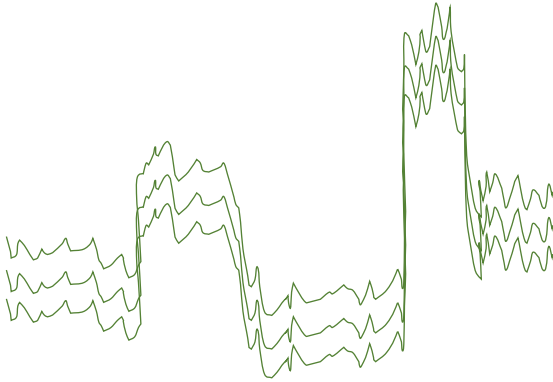


SquiggleFilter

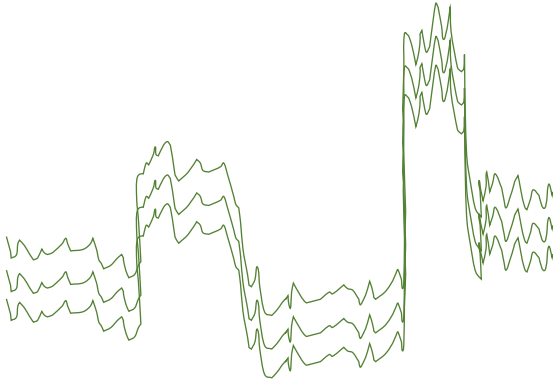
1. Translate **reference** genome to **expected squiggle**
2. Normalize **query squiggle** 
3. Compare input and reference signal using **dynamic time warping** 
4. If they match: keep analyzing
If not: reject immediately



Normalizing Query Squiggles

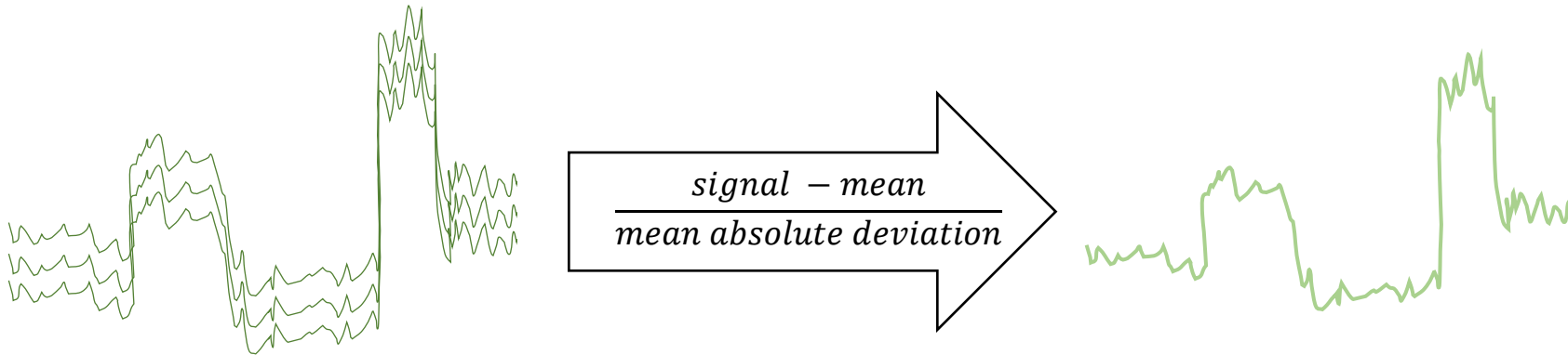


Normalizing Query Squiggles



Measured currents in query squiggles can **vary** due to process variation, among other things.

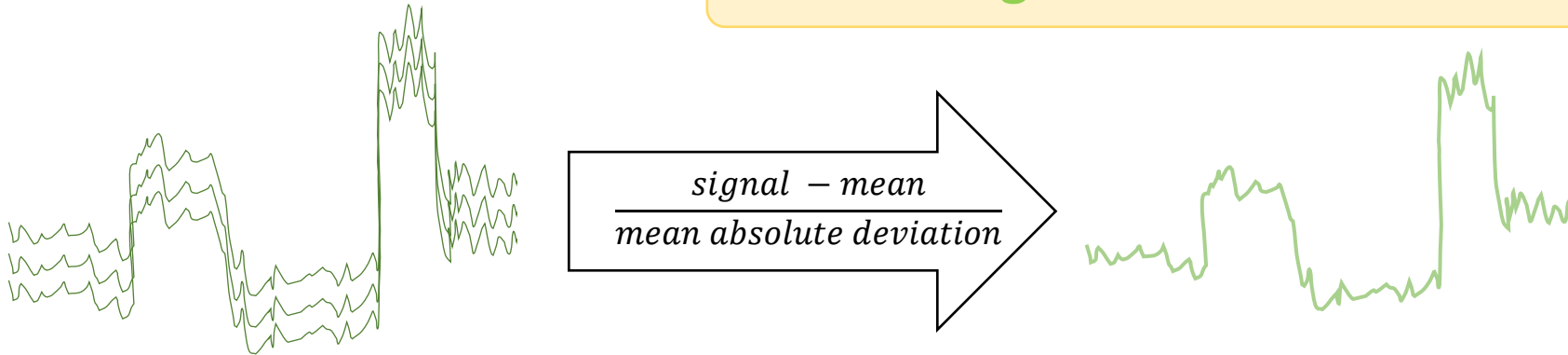
Normalizing Query Squiggles



Measured currents in query squiggles can **vary** due to process variation, among other things.


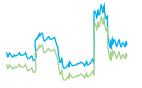
Normalizing Query Squiggles

Normalizing removes these differences.



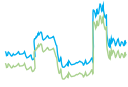
Measured currents in query squiggles can **vary** due to process variation, among other things.

SquiggleFilter

1. Translate **reference** genome to **expected squiggle**
2. Normalize **query squiggle** 
3. **Compare** input and reference signal using **dynamic time warping** 
4. If they match: keep analyzing
If not: reject immediately



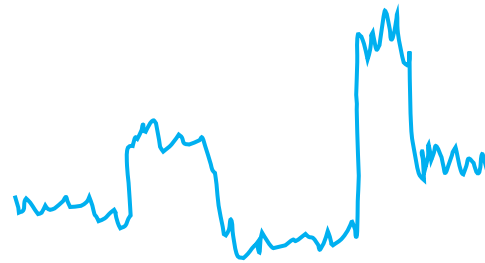
SquiggleFilter

1. Translate **reference** genome to **expected squiggle**
2. Normalize **query squiggle** 
3. **Compare** input and reference signal using **dynamic time warping** 
4. If they match: keep analyzing
If not: reject immediately



Dynamic Time Warping (prior work)

Reference Squiggle



Query Squiggle



Vintsyuk, T.K. [Speech discrimination by dynamic programming](#).
Cybernetics and System Analysis **4**, 52–57 (1968).

Dynamic Time Warping (prior work)

Reference Squiggle



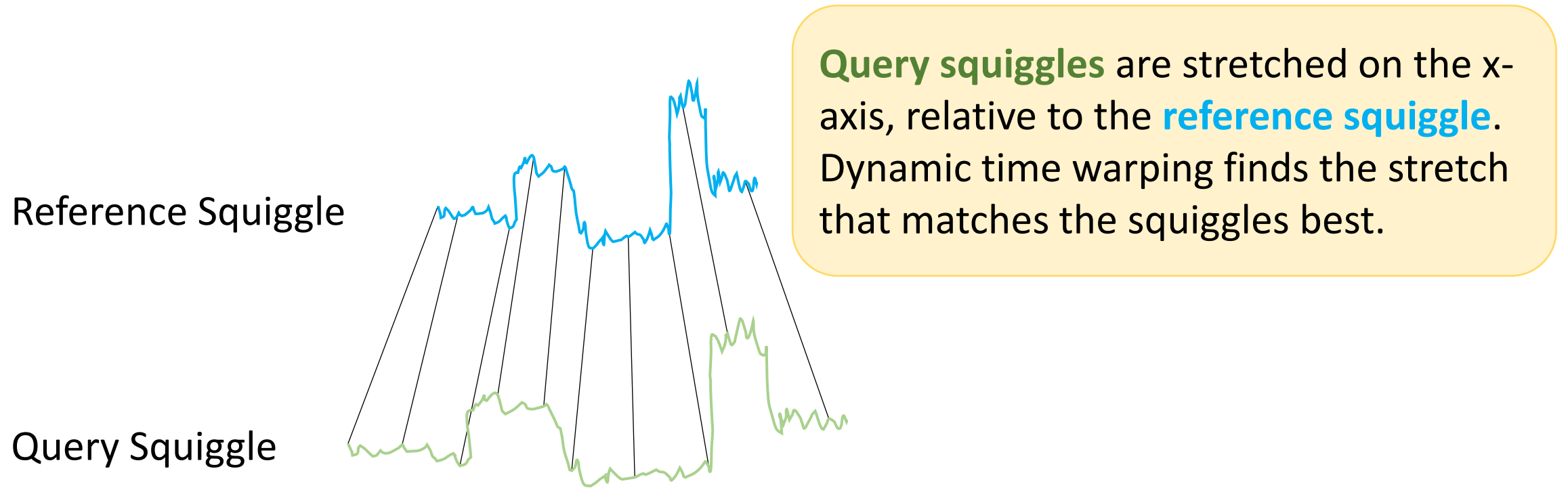
Query Squiggle



Query squiggles are stretched on the x-axis, relative to the **reference squiggle**.

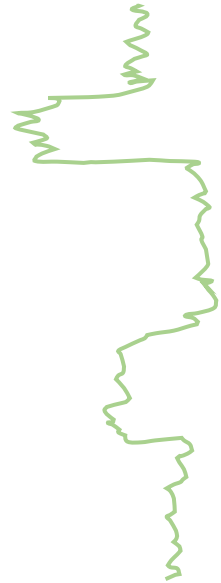
Vintsyuk, T.K. [Speech discrimination by dynamic programming](#).
Cybernetics and System Analysis **4**, 52–57 (1968).

Dynamic Time Warping (prior work)



Vintsyuk, T.K. [Speech discrimination by dynamic programming](#).
Cybernetics and System Analysis **4**, 52–57 (1968).

DTW Dynamic Programming (prior work)



0	4	7	8	11	16	20	22	23	26	30	33	34	39	44	47	49	50	51	54
4	7	8	11	16	20	22	23	26	30	33	34	39	44	47	49	50	51	54	57
7	8	11																	

DTW Dynamic Programming (prior work)

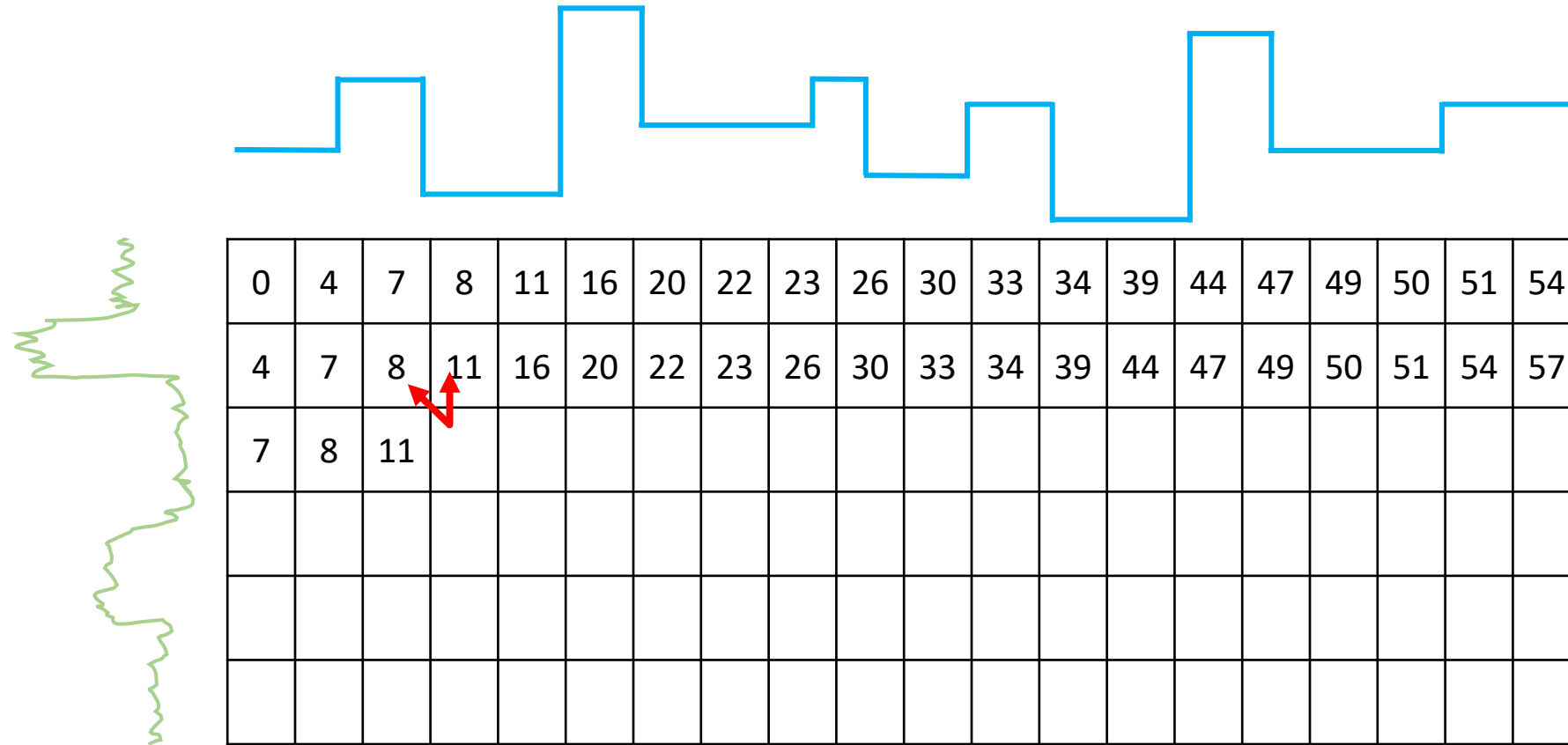


0	4	7	8	11	16	20	22	23	26	30	33	34	39	44	47	49	50	51	54
4	7	8	11	16	20	22	23	26	30	33	34	39	44	47	49	50	51	54	57
7	8	11																	

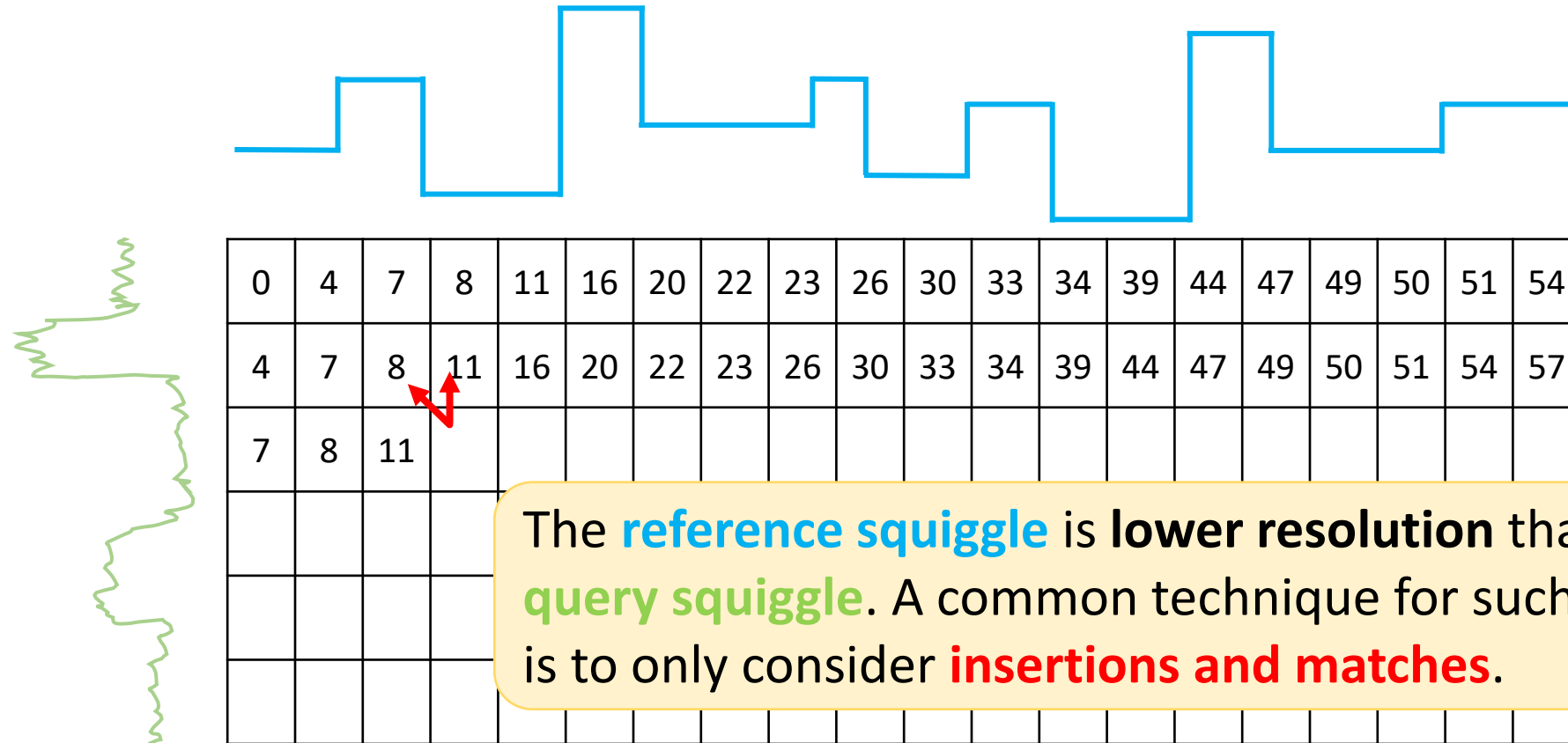
Update rule:

$$dp[i,j] = \min(\text{neighbors}) + \text{abs}(a[i] - b[j])$$

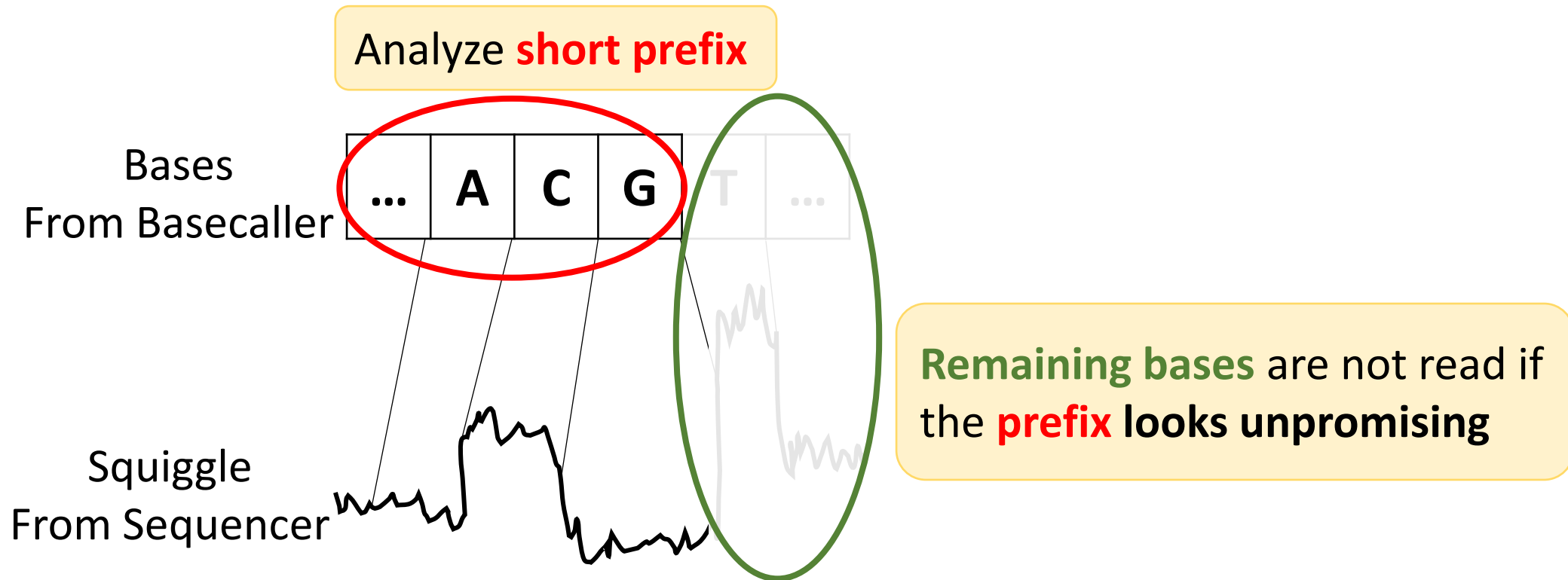
DTW Dynamic Programming (prior work)



DTW Dynamic Programming (prior work)



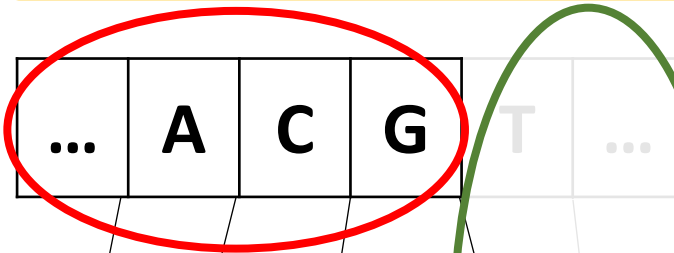
With ReadUntil



With ReadUntil

Analyze **short prefix**. But **how short**?

Bases
From Basecaller



Squiggle
From Sequencer



Remaining bases are not read if
the **prefix** looks unpromising

Query Prefix Size

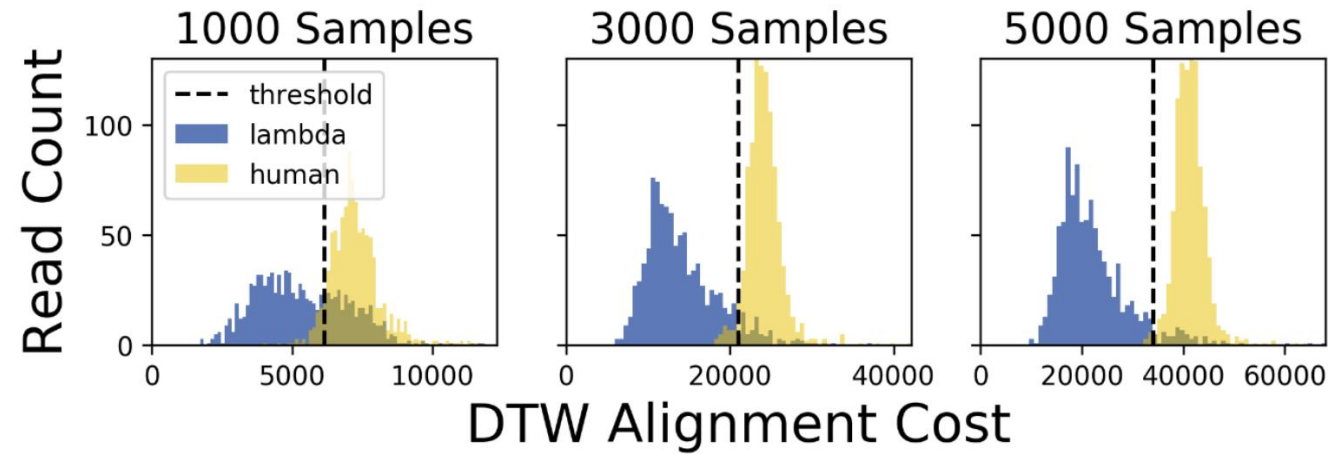


Figure 11: sDTW cost distributions for reads of 3 prefix lengths, aligned to the lambda phage genome.

Query Prefix Size

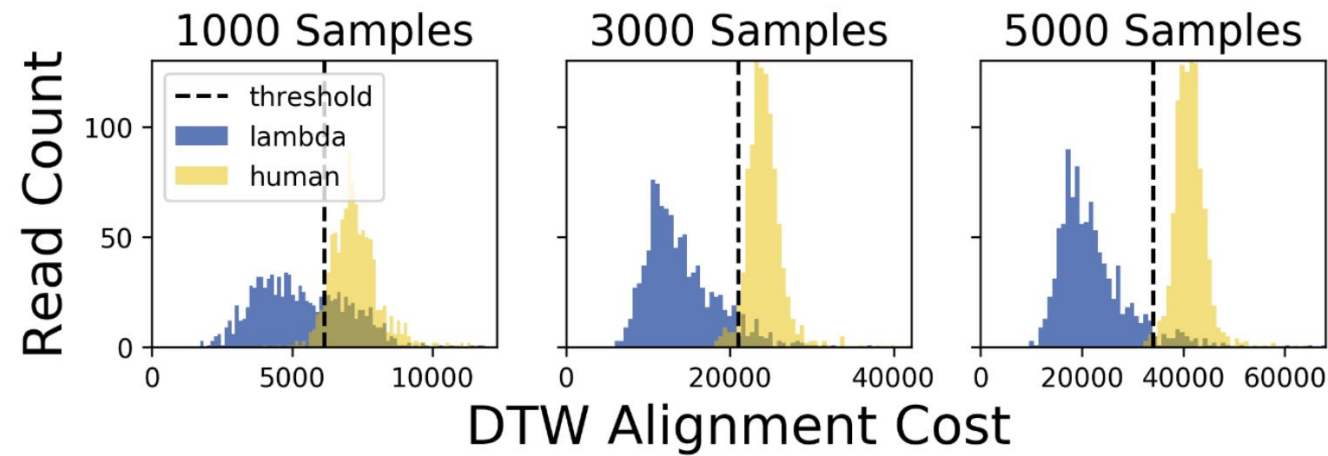
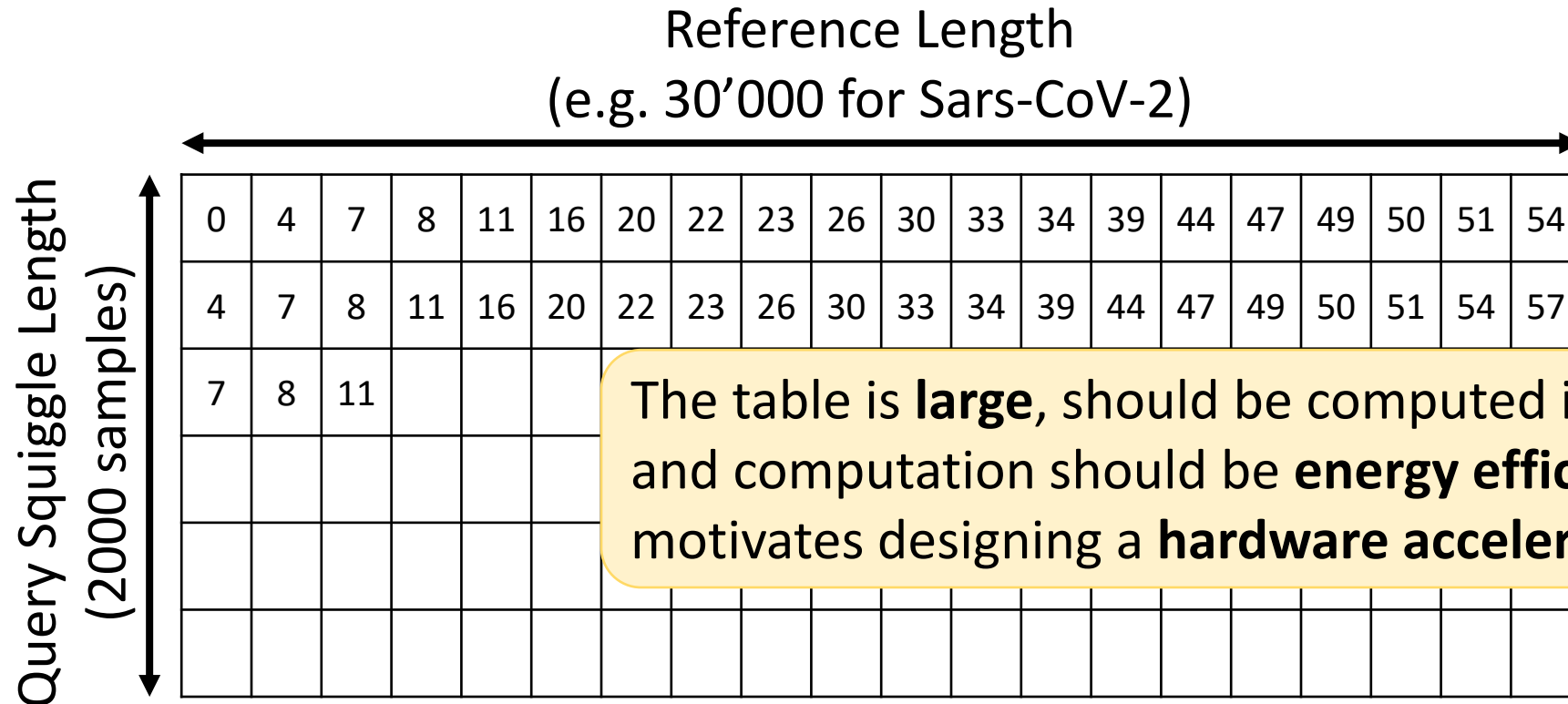


Figure 11: sDTW Alignment Cost for different sample sizes. The authors determine **2000 samples** to be a good **middle ground** between accuracy and performance.

DTW Dynamic Programming (prior work)



The table is **large**, should be computed in **real-time**, and computation should be **energy efficient**. This motivates designing a **hardware accelerator**.

Outline

1. Background and Motivation
 1. Virus detection
 2. Nanopore sequencing
2. SquiggleFilter
 1. Pipeline
 2. Accelerator
3. Results
4. Strengths/Weaknesses
5. Discussion

Accelerator Overview

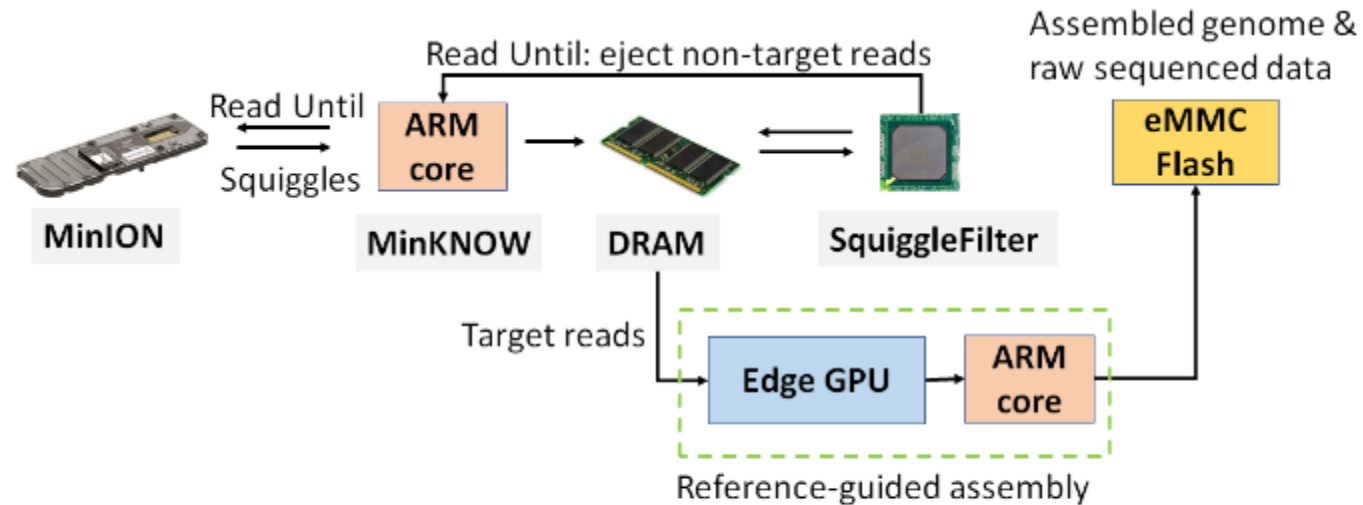


Figure 12: System-on-Chip design with the accelerated hardware filter on ASIC integrated with NVIDIA GPU and 8-core ARM v8.2 64-bit CPU

Accelerator Systolic Array

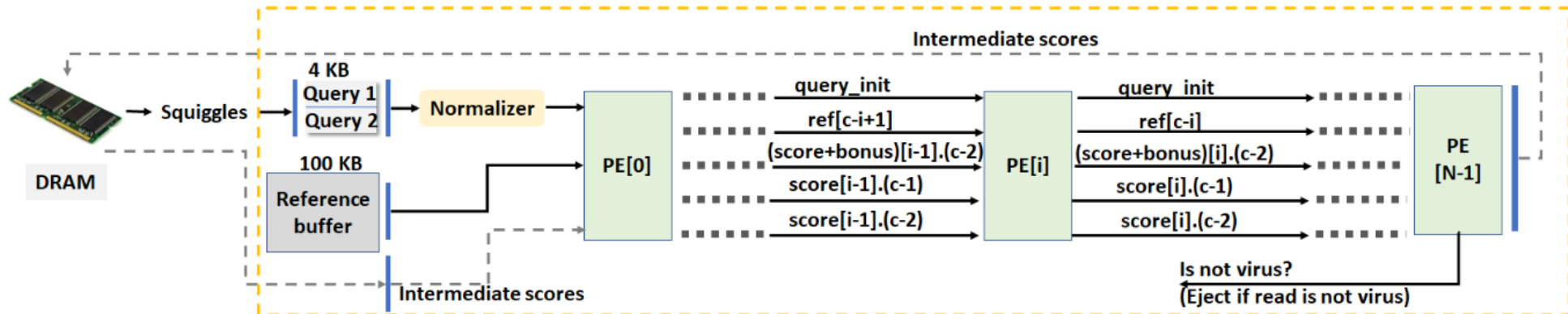


Figure 13: SquiggleFilter Tile. $N=2000$ PEs are connected with streaming inputs and outputs. The last PE determines the classification by comparing its cost to a threshold every cycle. c is the cycle and i is the PE index.

Accelerator Systolic Array

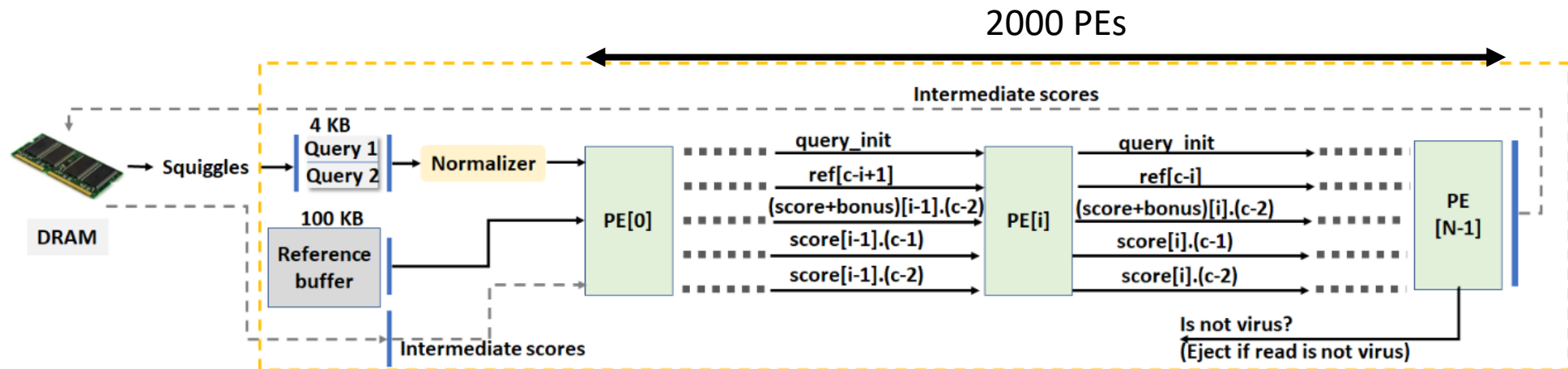


Figure 13: SquiggleFilter Tile. $N=2000$ PEs are connected with streaming inputs and outputs. The last PE determines the classification by comparing its cost to a threshold every cycle. c is the cycle and i is the PE index.

The DTW accelerator is implemented as a **systolic array** of 2000 PEs.

Outline

1. Background and Motivation

1. Virus detection
2. Nanopore sequencing

2. SquiggleFilter

1. Pipeline
2. Accelerator

3. Results

4. Strengths/Weaknesses

5. Discussion

Results – Computational Speedup

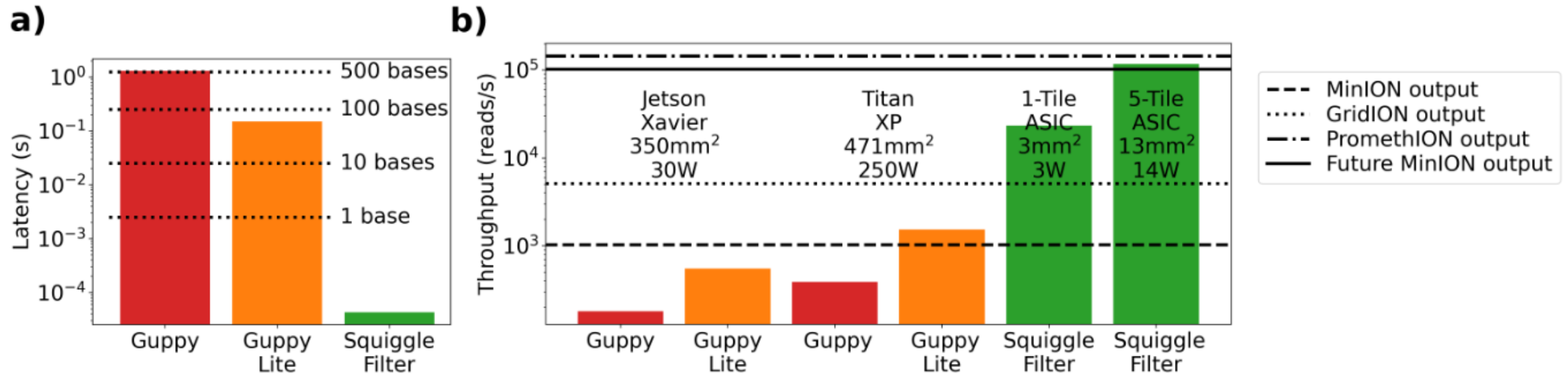


Figure 16: a) Latency, and b) throughput of Guppy, Guppy-lite and SquiggleFilter during Read Until.

Results - Computational Speedup

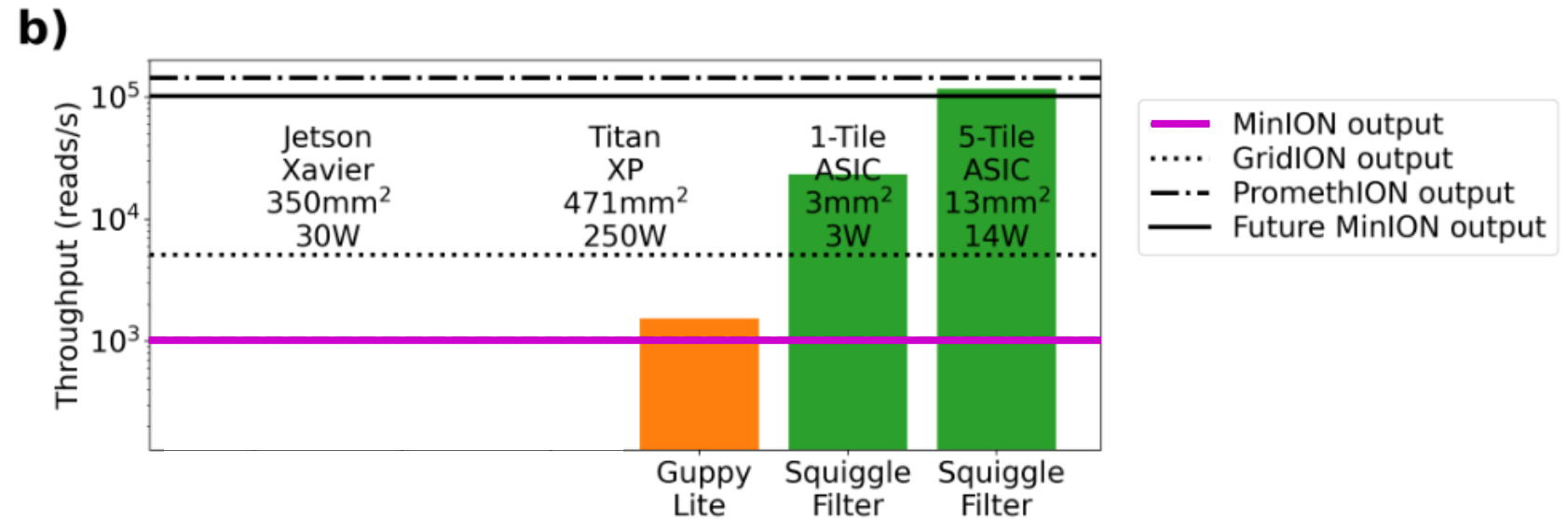


Figure 16: a) Latency, and b) throughput of Guppy, Guppy-lite and SquiggleFilter during Read Until.

Only the **most inaccurate model** on a **server-class GPU** or **SquiggleFilter** can keep up with even the **slowest sequencer**.

Results - Computational Speedup

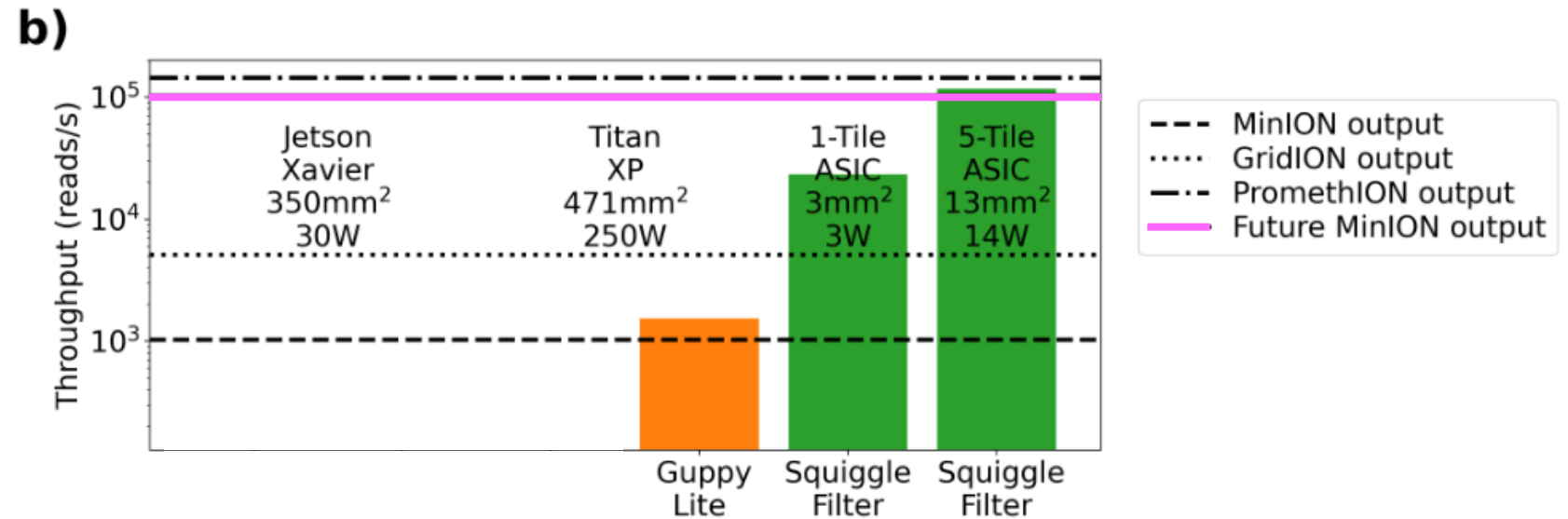
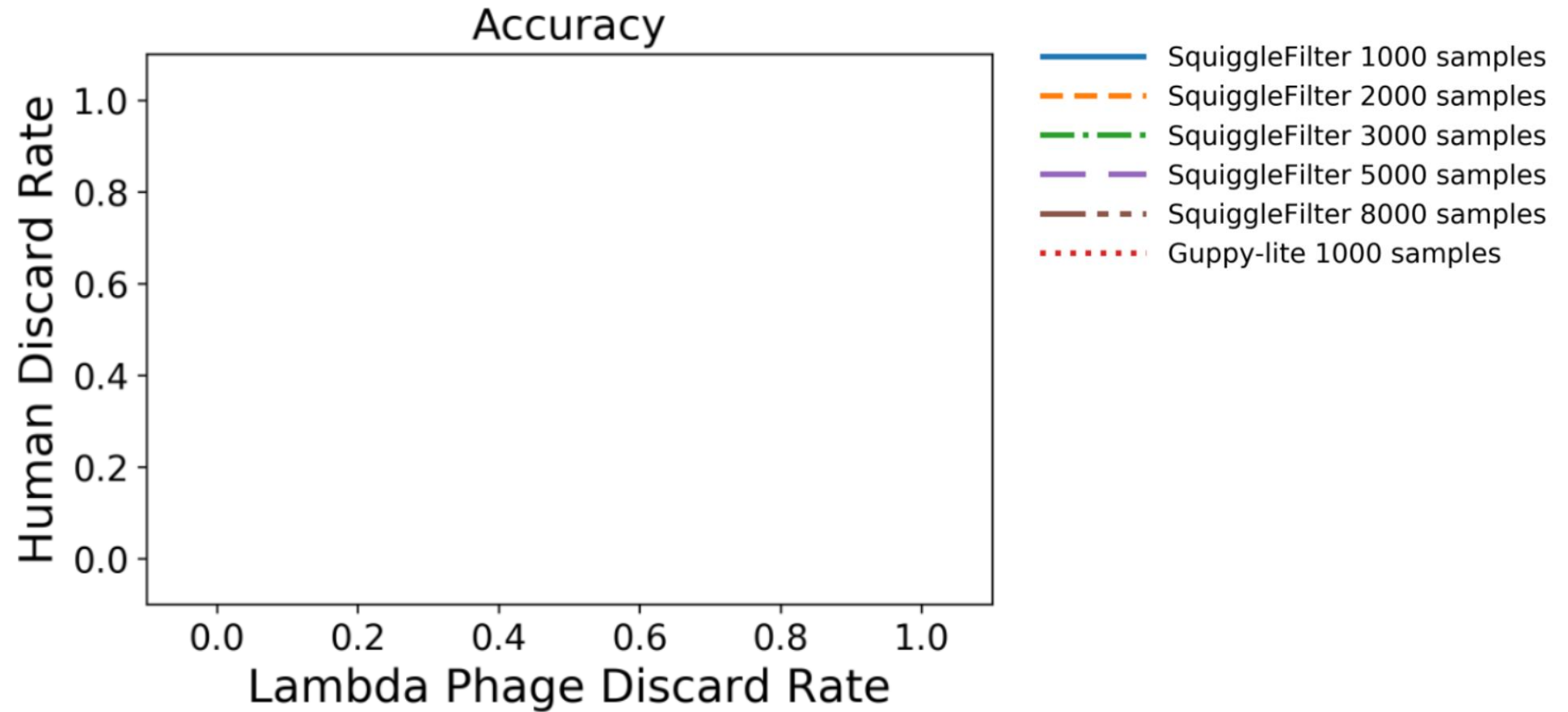


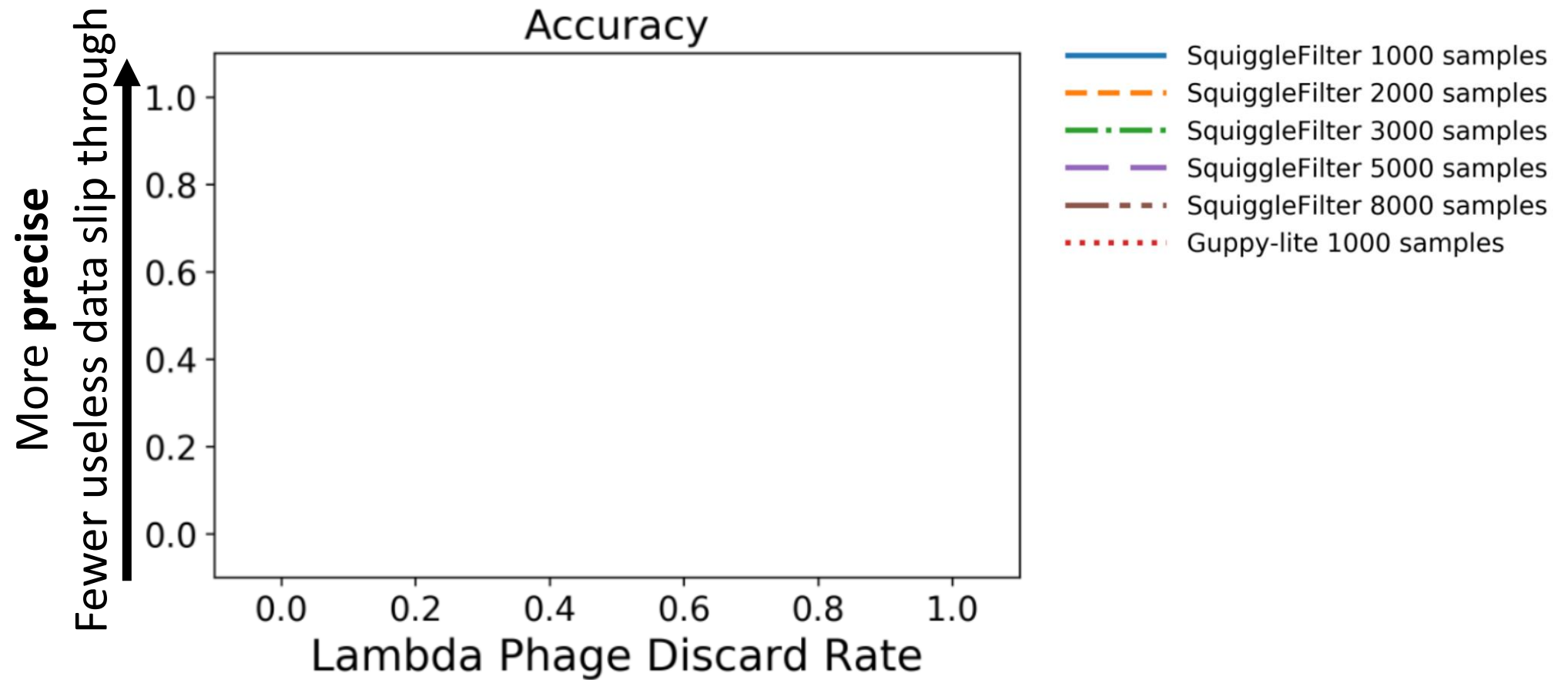
Figure 16: a) Latency, and b) throughput of Guppy, Guppy-lite and SquiggleFilter during Read Until.

Only **SquiggleFilter** can keep up with a projected **future version** of the **slowest sequencer**.

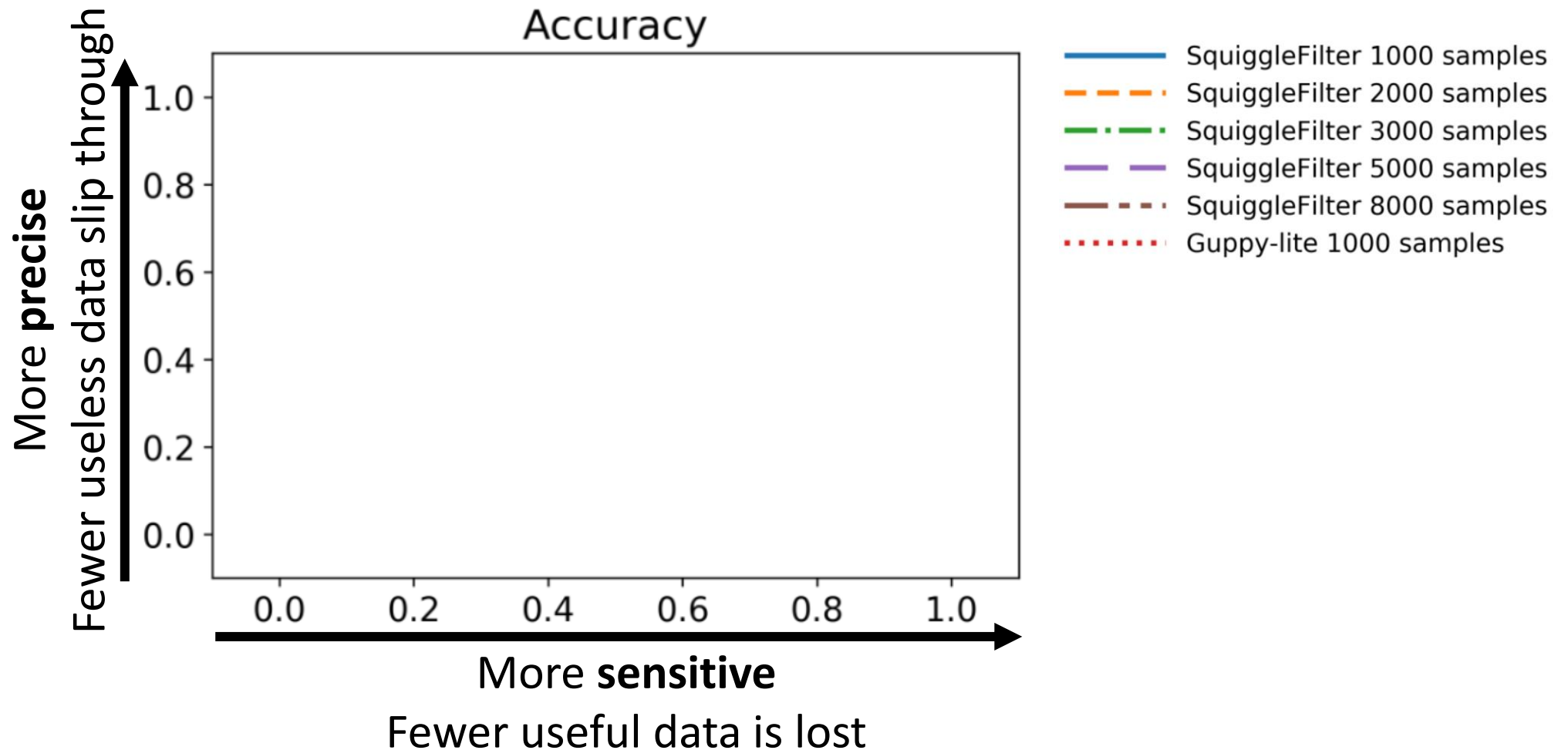
Results - Accuracy



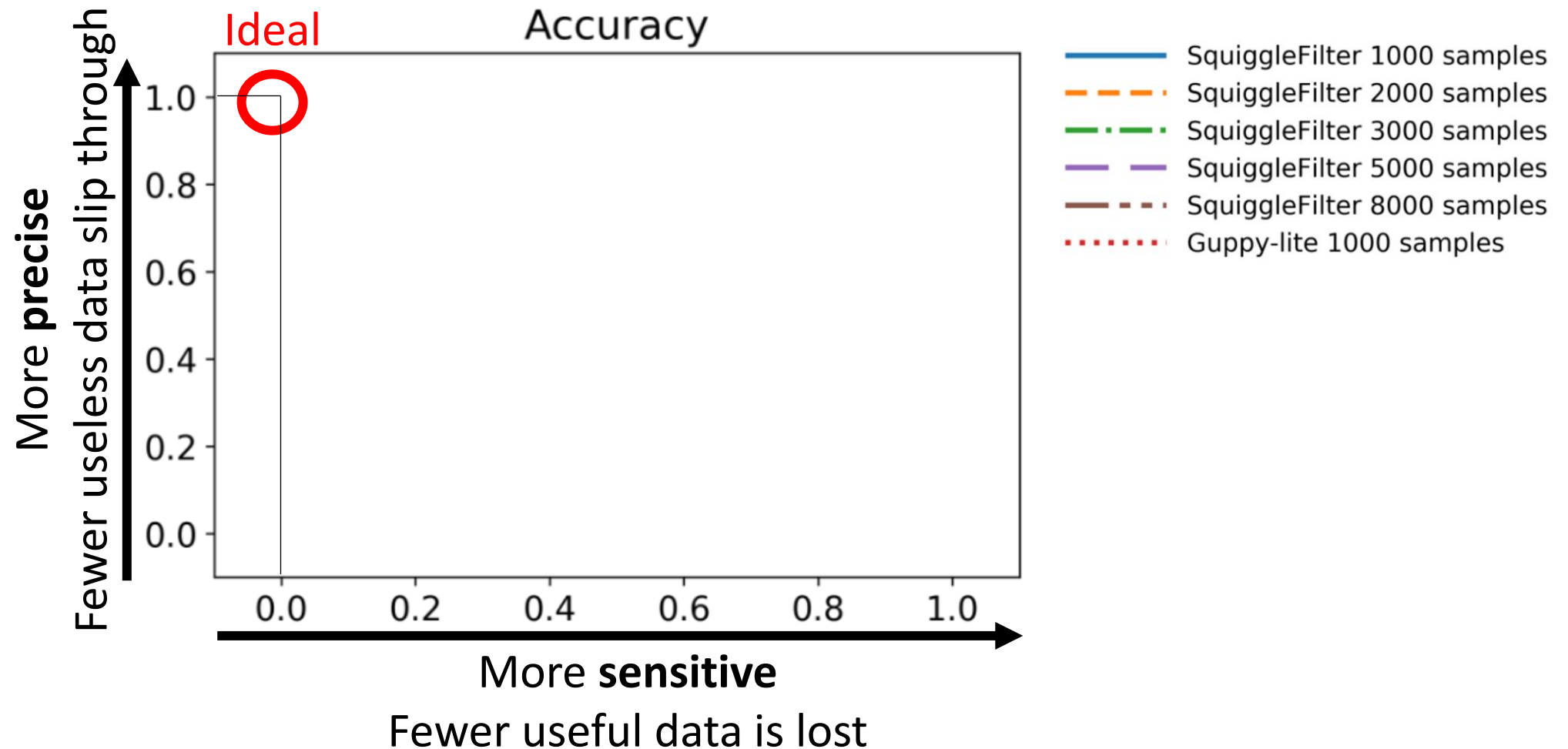
Results - Accuracy



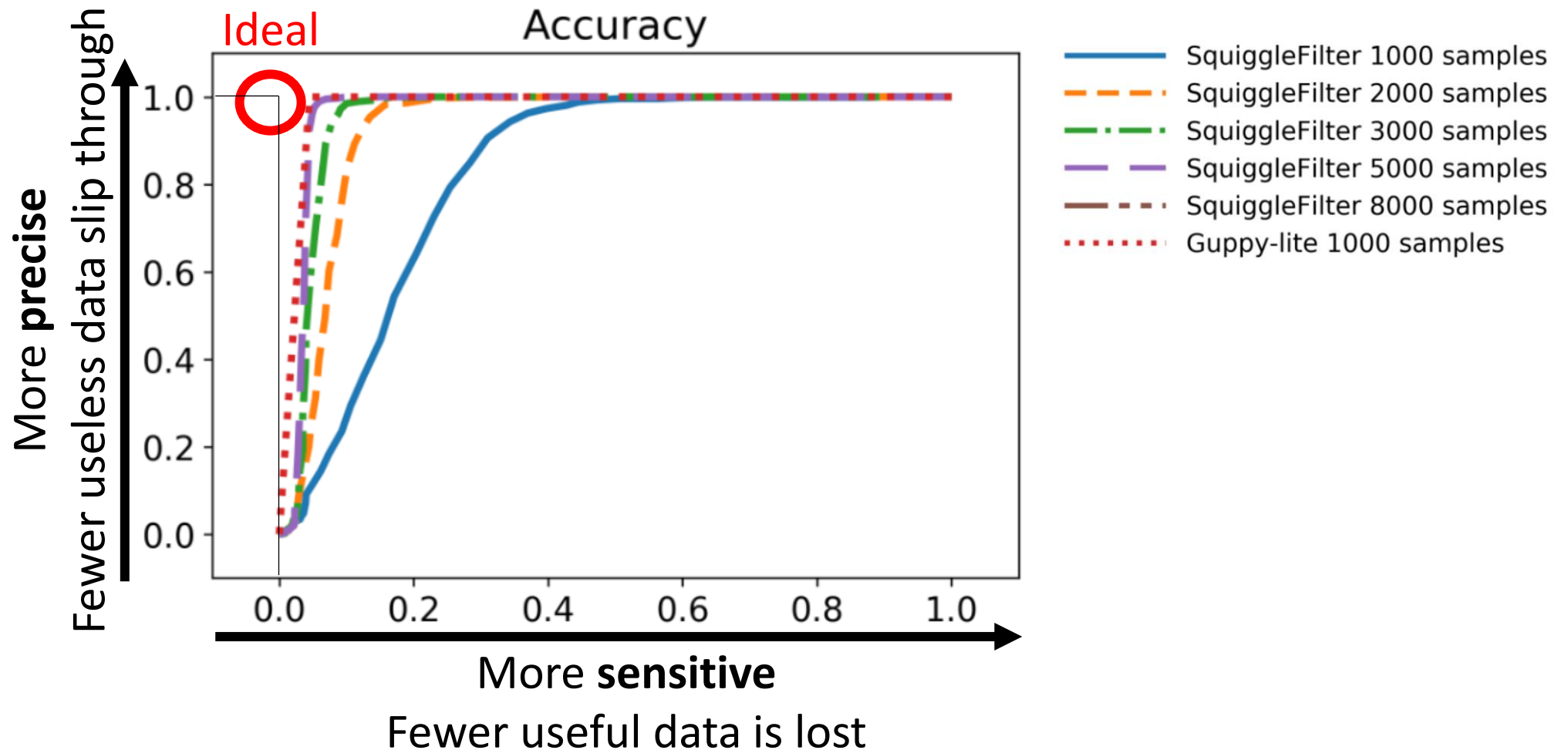
Results - Accuracy



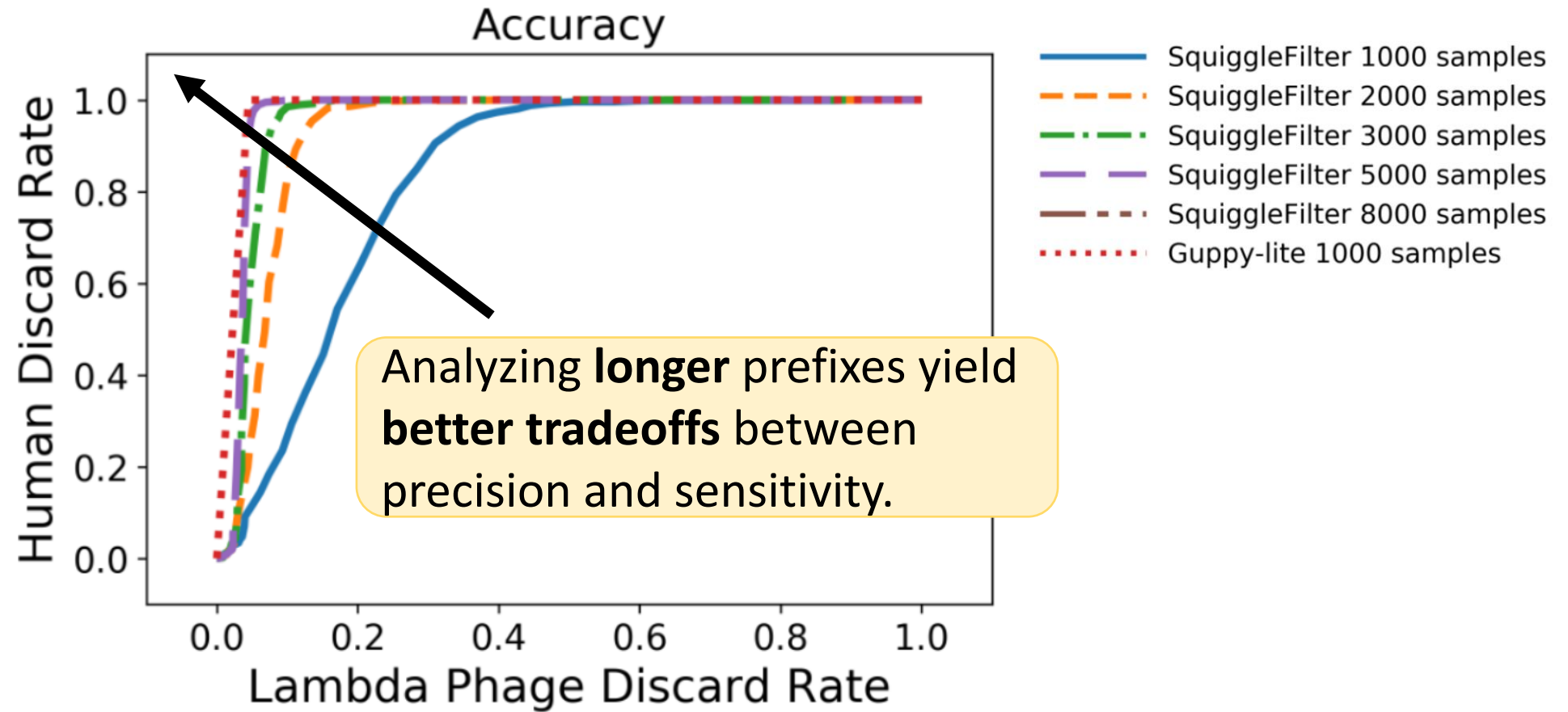
Results - Accuracy



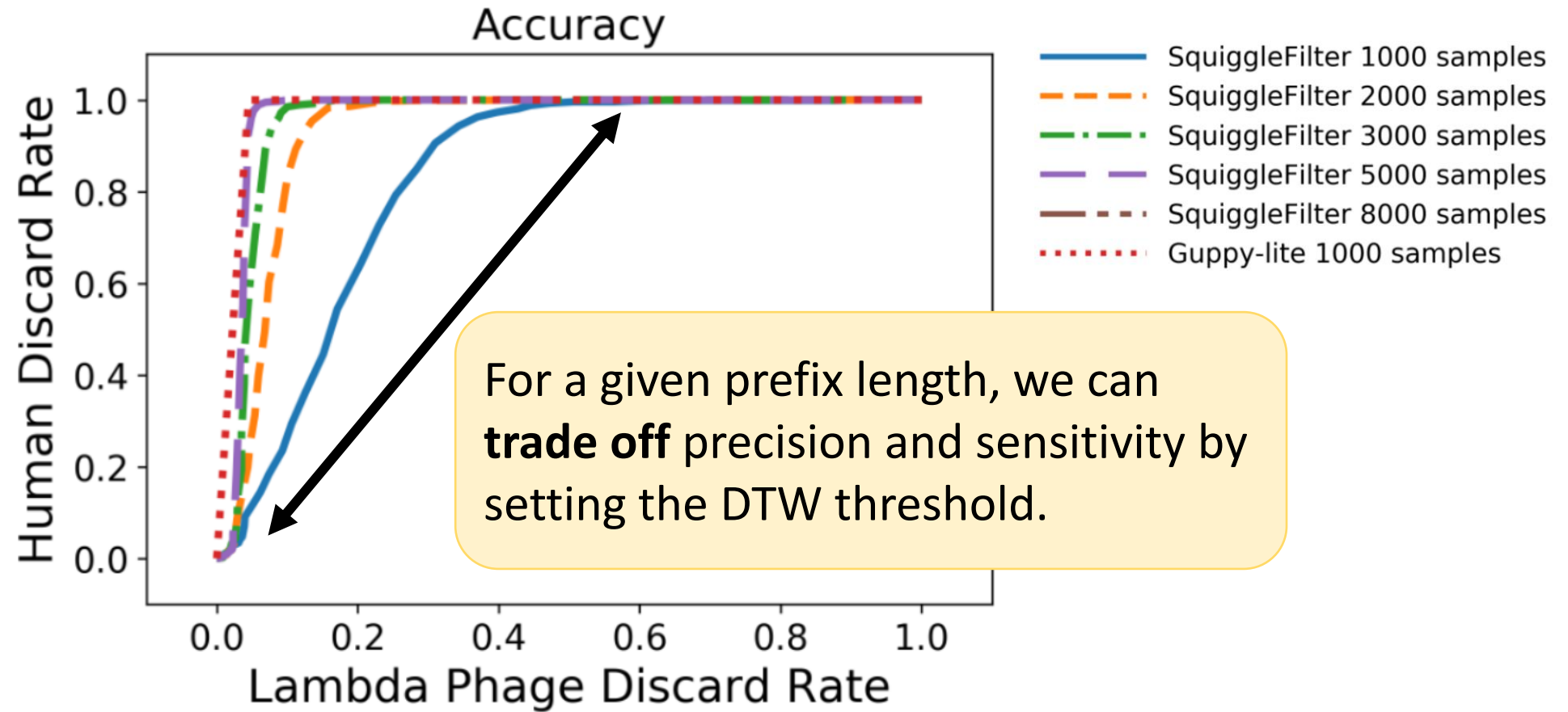
Results - Accuracy



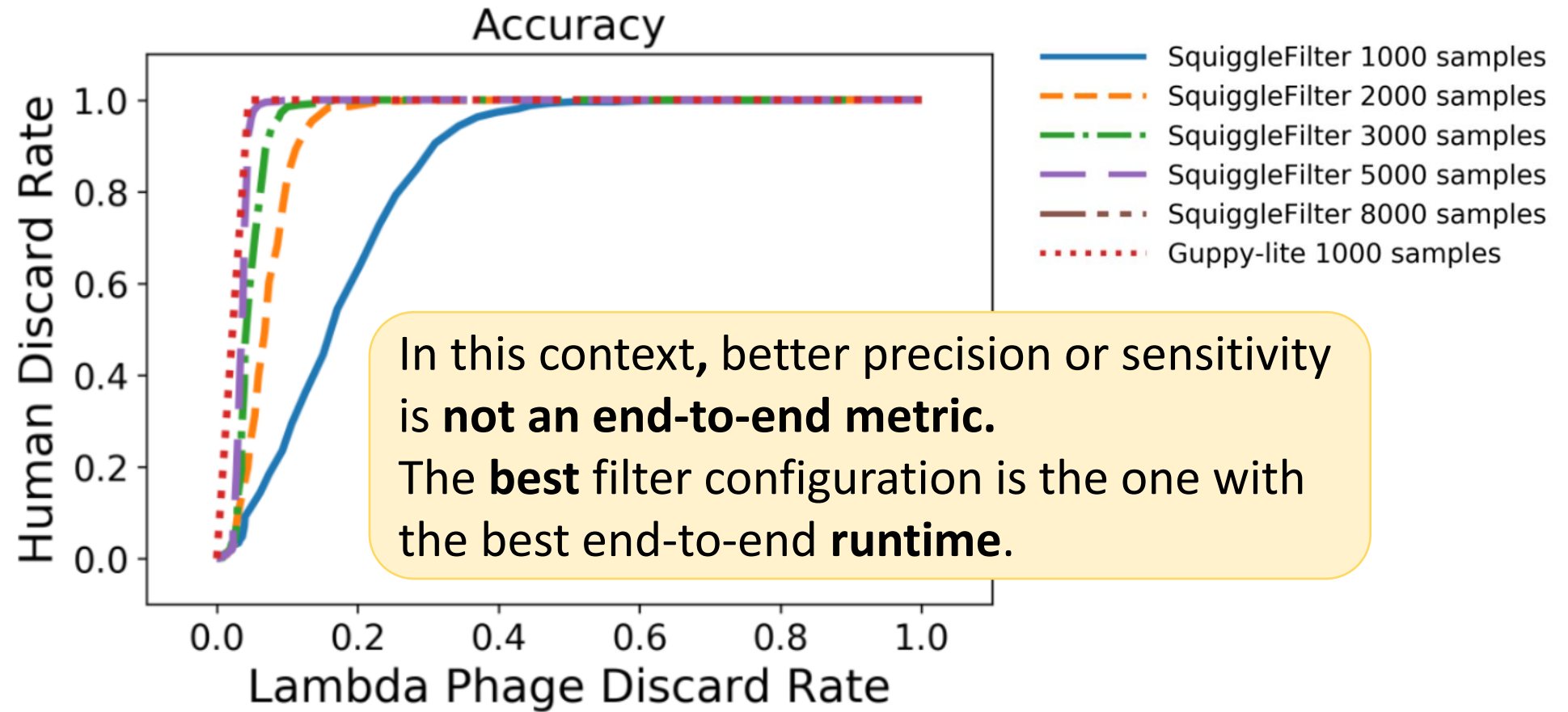
Results - Accuracy



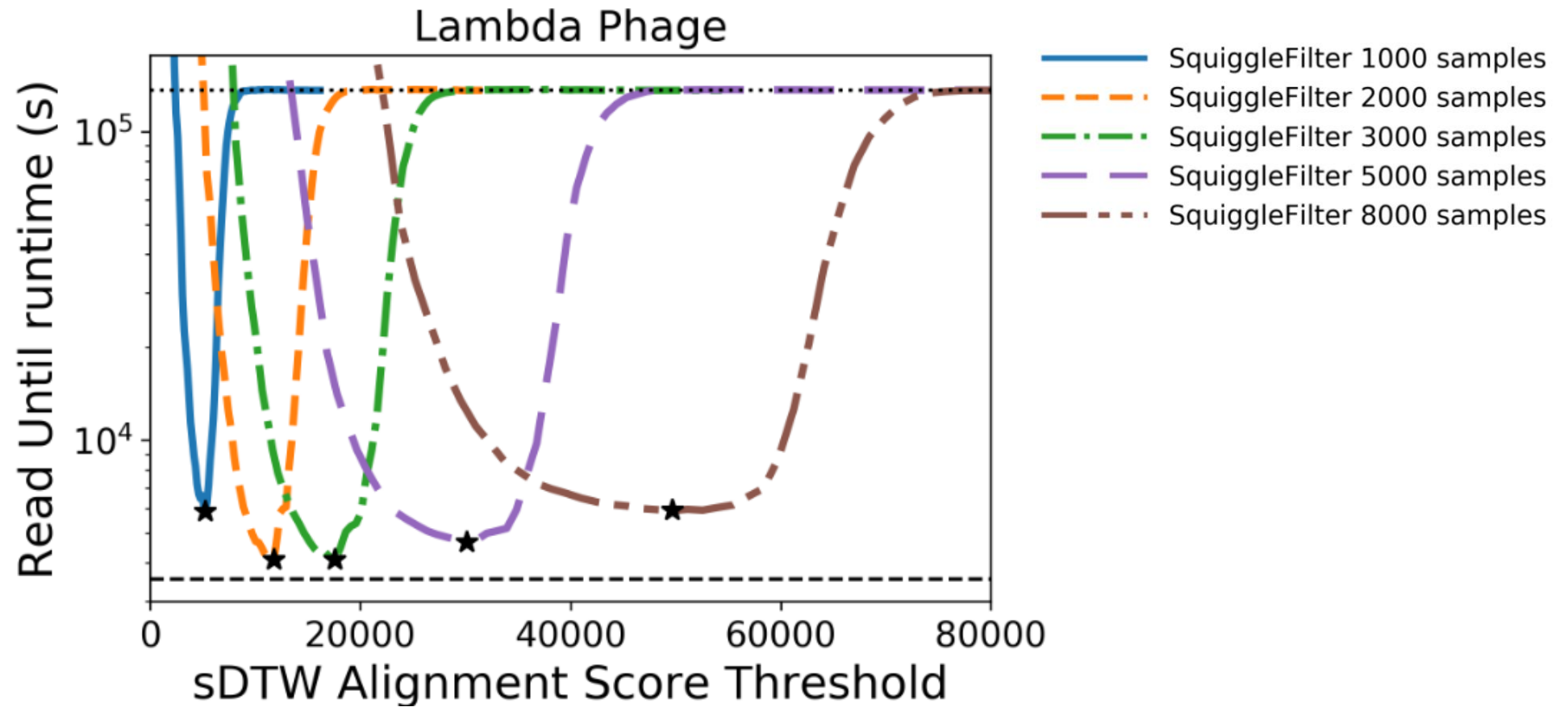
Results - Accuracy



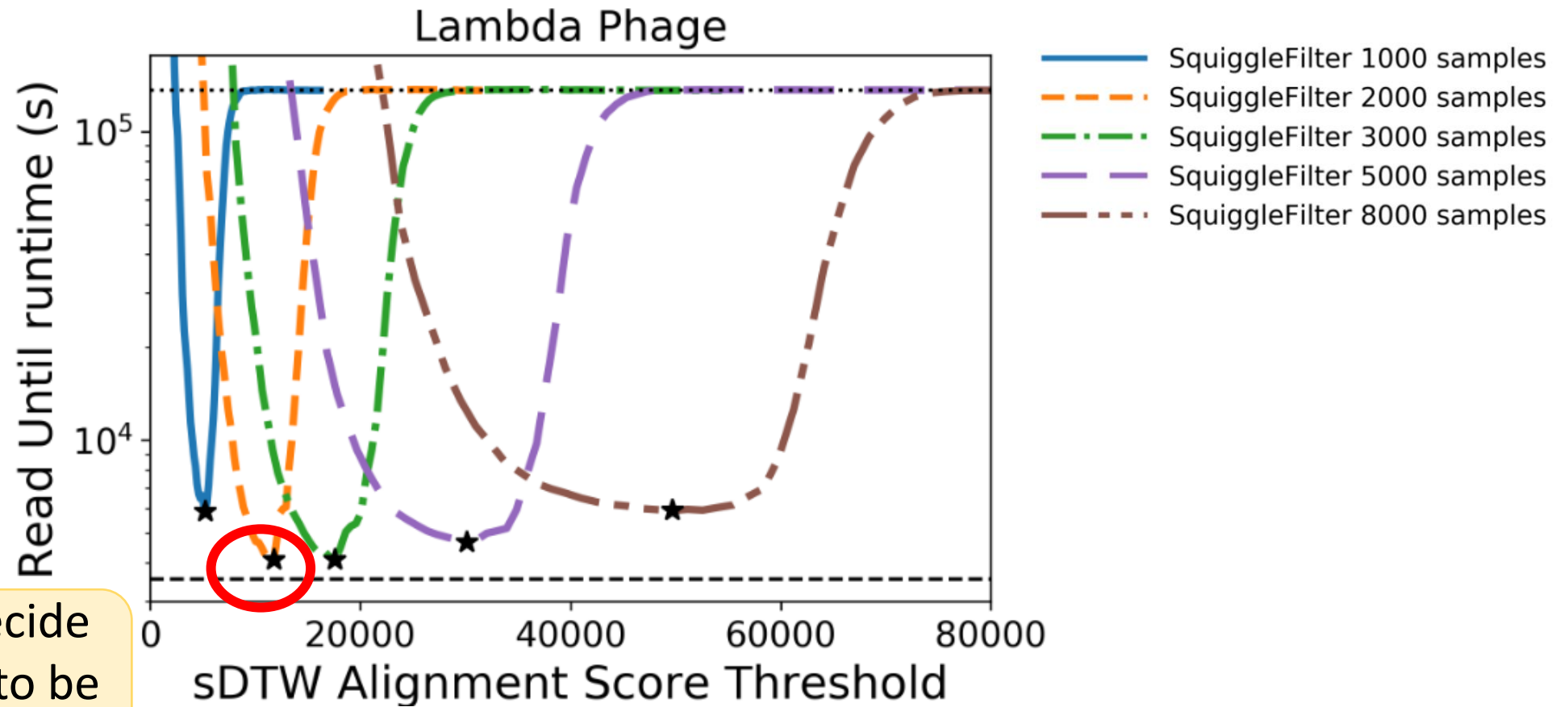
Results - Accuracy



Results - Accuracy



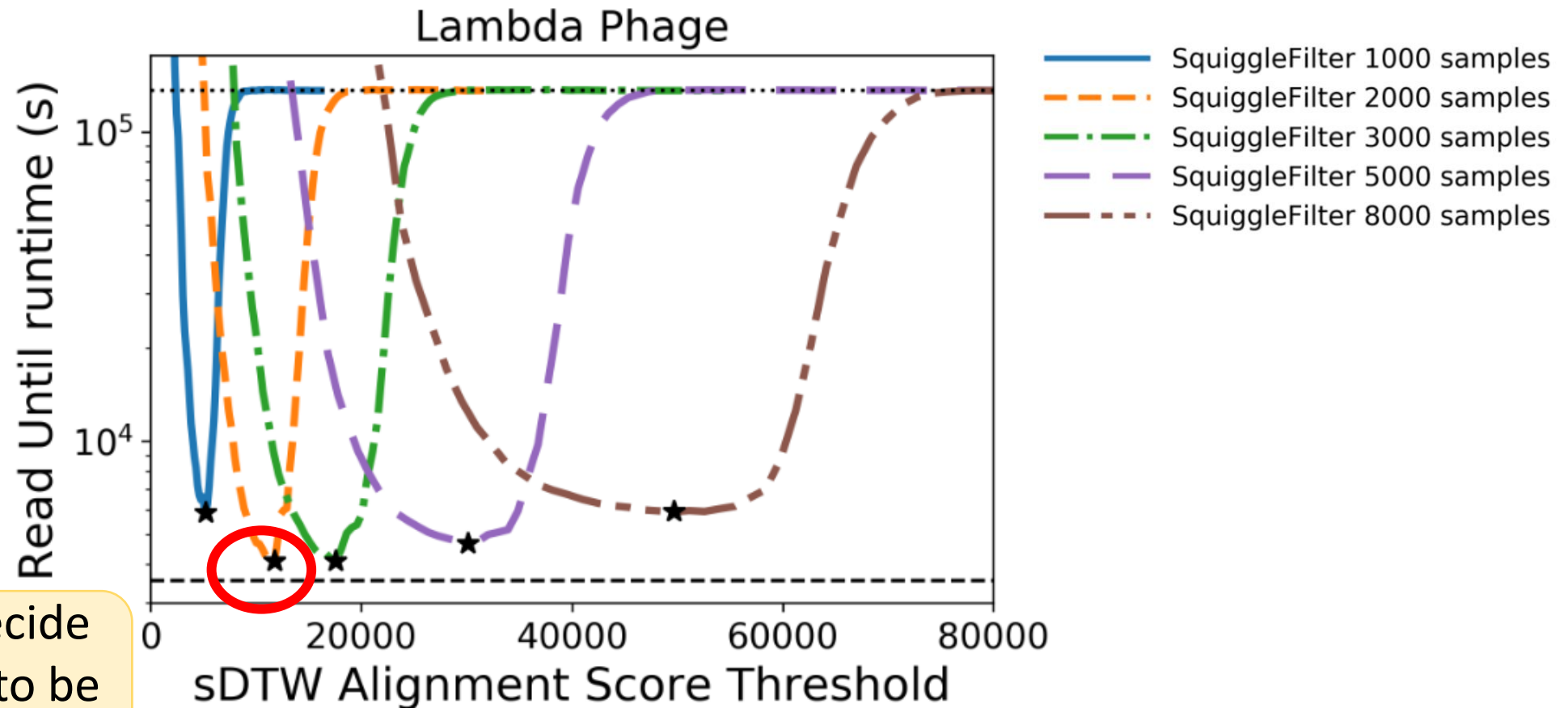
Results - Accuracy



The authors decide **2000 samples** to be the sweet-spot.

Results - Accuracy

SquiggleFilter achieves an **end-to-end** speedup over Guppy-lite of **12.9%**.
(Not drawn here)



The authors decide **2000 samples** to be the sweet-spot.

Results - Accuracy

$$F_1 = \frac{2}{\text{recall}^{-1} + \text{precision}^{-1}}$$

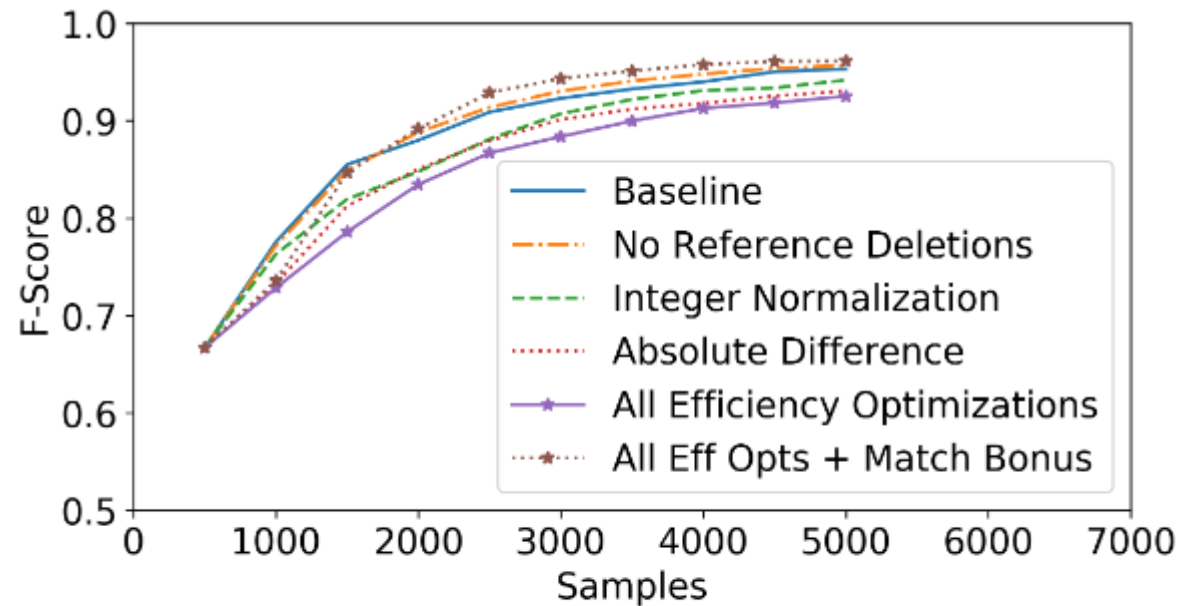


Figure 18: Accuracy results for modifications to the standard sDTW algorithm.

Results - Power

ASIC Element	Area (mm ²)	Power (W)
Normalizer	0.014	0.045
Processing Element	0.001	0.002
Tile (1×2000 PEs)	2.423	2.780
Query buffer	0.023	0.009
Reference buffer	0.185	0.028
Complete 1-Tile ASIC	2.65	2.86
Complete 5-Tile ASIC	13.25	14.31

Table 4: SquiggleFilter ASIC synthesis results.

Synthesized for **28nm** TSMC HPC @2.5GHz

Results - Power

SquiggleFilter's **ASIC** is significantly more **power efficient than** a server-class GPU, sufficiently so it could be battery powered.

ASIC Element	Area (mm ²)	Power (W)
Normalizer	0.014	0.045
Processing Element	0.001	0.002
Tile (1×2000 PEs)	2.423	2.780
Query buffer	0.023	0.009
Reference buffer	0.185	0.028
Complete 1-Tile ASIC	2.65	2.86
Complete 5-Tile ASIC	13.25	14.31

Table 4: SquiggleFilter ASIC synthesis results.

Synthesized for **28nm** TSMC HPC @2.5GHz

Outline

1. Background and Motivation
 1. Virus detection
 2. Nanopore sequencing
2. SquiggleFilter
 1. Pipeline
 2. Accelerator
3. Results
4. Strengths/Weaknesses
5. Discussion

Strengths

- Extremely **important** problem

Strengths

- Extremely **important** problem
- The solution **works** well and is **reasonable** to implement

Strengths

- Extremely **important** problem
- The solution **works** well and is **reasonable** to implement
- **Future proof** (can tolerate much faster sequencers)

Strengths

- Extremely **important** problem
- The solution **works** well and is **reasonable** to implement
- **Future proof** (can tolerate much faster sequencers)
- They claim to be the **first** proposal of using **squiggle alignment for enriching** low-concentration viral specimen with ReadUntil

Weaknesses

- **No comparison to a CPU/GPU** implementation of their algorithm. It's unclear how much of the speedup comes from the algorithm, and how much from HW acceleration

Weaknesses

- **No comparison to a CPU/GPU** implementation of their algorithm. It's unclear how much of the speedup comes from the algorithm, and how much from HW acceleration
- **No comparison** to other DTW accelerators

Weaknesses

- **No comparison to a CPU/GPU** implementation of their algorithm. It's unclear how much of the speedup comes from the algorithm, and how much from HW acceleration
- **No comparison** to other DTW accelerators
- **No coverage** accuracy measurements

Weaknesses

- **No comparison to a CPU/GPU** implementation of their algorithm. It's unclear how much of the speedup comes from the algorithm, and how much from HW acceleration
- **No comparison** to other DTW **accelerators**
- **No coverage** accuracy measurements
- The work is **limited** to relatively **small viral genomes**. Their proposal would not work for viruses with a larger genome, such as Smallpox and Herpes Simplex

Weakness – Limited to Small Genomes

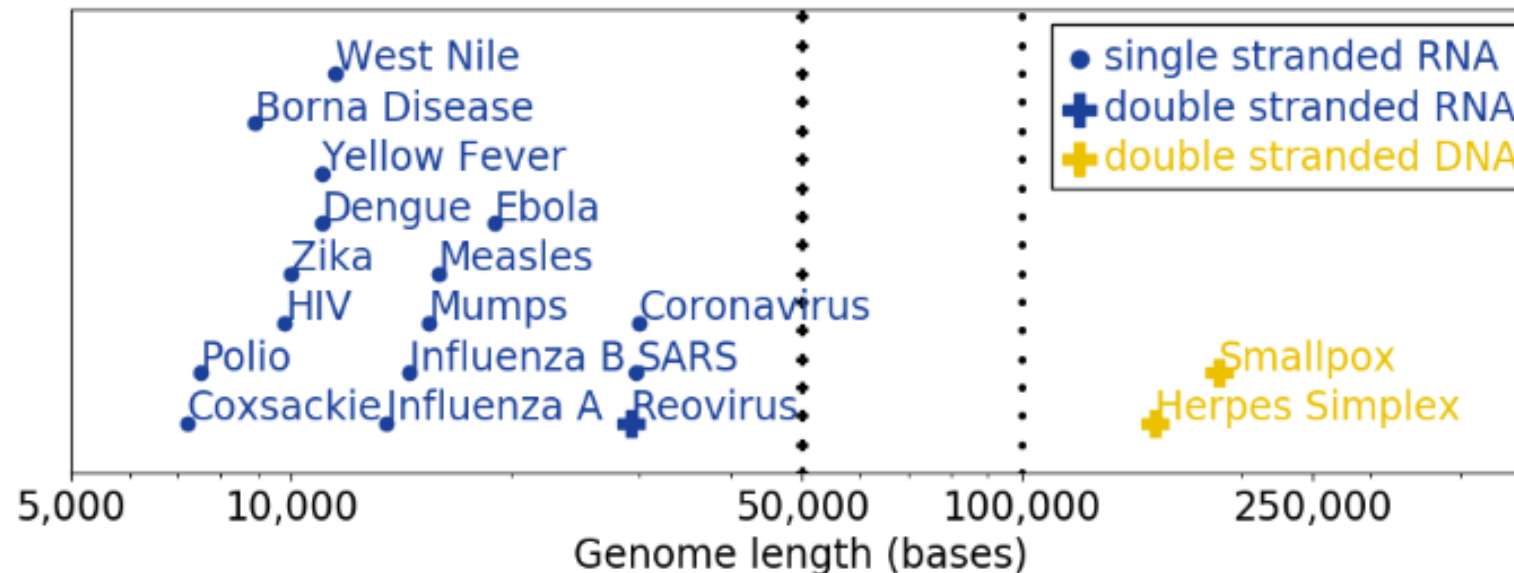
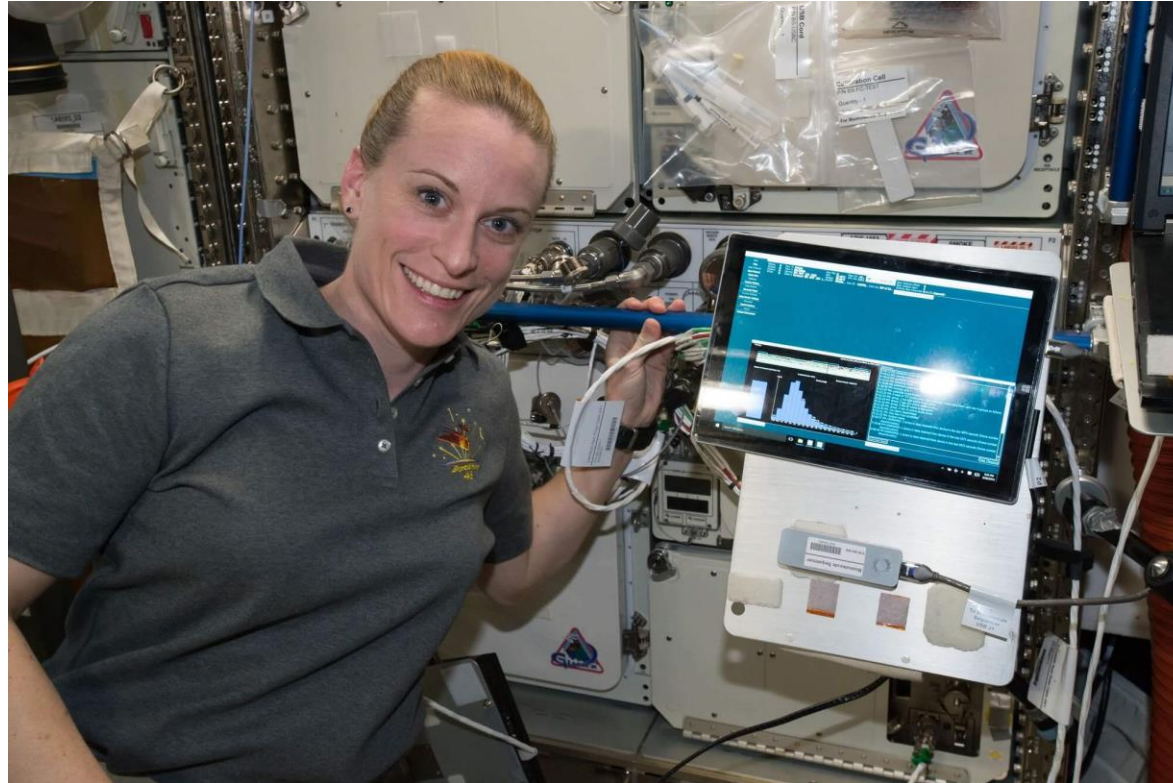


Figure 10: Epidemic virus genome lengths.

Outline

1. Background and Motivation
 1. Virus detection
 2. Nanopore sequencing
2. SquiggleFilter
 1. Pipeline
 2. Accelerator
3. Results
4. Strengths/Weaknesses
5. Discussion

What are challenges of genome analysis in space?



<https://www.universetoday.com/135327/whats-strange-glowing-mold-astronauts-will-soon-be-able-to-sequence-unknown-space-organisms/>

What are challenges of genome analysis in space?

Wallace offered support to Whitson, a biochemist, as she used the MinION device (developed by [Oxford Nanopore Technologies](#)) to sequence the amplified DNA.

The **data were downlinked** to the team in Houston for **analysis and identification**.

“Once we actually got the data on the ground we were able to turn it around and start analyzing it,” said Aaron Burton, NASA biochemist and the project’s co-investigator. “You get all these **squiggle plots** and you have to turn that into As, Gs, Cs and Ts.”

https://www.nasa.gov/mission_pages/station/research/news/b4h-3rd/ge-gis-3-identifies-unknown-microbes

What are challenges of genome analysis in space?

- Low communication bandwidth with base station
- Devices must be low energy, lightweight, portable
- Devices should be reconfigurable

Signal comparison without DTW?

Signal comparison without DTW?

- Fourier Transform, then compare in frequency space?

Signal comparison without DTW?

- Fourier Transform, then compare in frequency space?
- Compare based on handcrafted metrics?
 - Number of peaks and valleys
 - Bin oscillations and compare distributions
 - ...

Signal comparison without DTW?

- Fourier Transform, then compare in frequency space?
- Compare based on handcrafted metrics?
 - Number of peaks and valleys
 - Bin oscillations and compare distributions
 - ...
- Fortunately, being inaccurate only costs time, so we are free to experiment wildly in this application!

Can we apply heuristics to DTW?

Can we apply heuristics to DTW?

- Search less entries of the $O(n*m)$ matrix
- Several techniques exist for the (related) **approximate string matching** problem
 - Banded diagonals, such as when calculating edit distance
[Algorithms for approximate string matching](#), E. Ukkonen, 1985
 - Greedy
[GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework for Genome Sequence Analysis](#), Senol Cali et al., 2020
 - Seeding
[Targeted nanopore sequencing by real-time mapping of raw electrical signal with UNCALLED](#), Kovaka et al., 2021

SquiggleFilter

An Accelerator for Portable Virus Detection

Tim Dunn, Harisankar Sadasivan, Jack Wadden, Kush Goliya,
Kuan-Yu Chen, David Blaauw, Reetuparna Das, and Satish Narayanasamy

University of Michigan

MICRO '21

Presented by: Joël Lindegger
joel.lindegger@inf.ethz.ch

23/12/2021

Backup Slides

Virus Detectors



Non-Programmable

Programmable



Virus Detectors



Non-Programmable

- Tailor chemistry to the given virus
- Cheaply mass-manufacture chemistry
- Kits must be physically distributed

Programmable



Virus Detectors



Non-Programmable

- Tailor chemistry to the given virus
- Cheaply mass-manufacture chemistry
- Kits must be physically distributed



Programmable

- Distribute viral genome (Software)
- Sequence DNA/RNA reads
- Compare to viral genome (Software)

Virus Detectors



Non-Programmable

- Tailor chemistry to the given virus
- Cheaply mass-manufacture chemistry
- Kits must be physically distributed



Programmable

“Programmable”

- Distribute viral genome (Software)
- Sequence DNA/RNA reads
- Compare to viral genome (Software)

Virus Detectors



Non-Programmable

- Tailor chemistry to the given virus
- Cheaply mass-manufacture chemistry
- Kits must be physically distributed
- **Inflexible**



Programmable

“Programmable”

- **Distribute viral genome (Software)**
- Sequence DNA/RNA reads
- Compare to viral genome (Software)
- **Flexible**

Virus Detectors



Non-Programmable

- Tailor chemistry to the given virus
- Cheaply mass-manufacture chemistry
- Kits must be physically distributed
- Inflexible
- Unavailable at beginning of pandemic



Programmable

“Programmable”

- Distribute viral genome (Software)
- Sequence DNA/RNA reads
- Compare to viral genome (Software)
- Flexible
- Available immediately

Virus Detectors



Non-Programmable

- Tailor chemistry to the given virus
- Cheaply mass-manufacture chemistry
- Kits must be physically distributed
- Inflexible
- Unavailable at beginning of pandemic
- Low diagnostic power (virus/no virus)



Programmable

“Programmable”

- Distribute viral genome (Software)
- Sequence DNA/RNA reads
- Compare to viral genome (Software)
- Flexible
- Available immediately
- High diagnostic power (e.g. virus strain)

Virus Detectors



Non-Programmable

- Tailor chemistry to the given virus
- Cheaply mass-manufacture chemistry
- Kits must be physically distributed
- Inflexible
- Unavailable at beginning of pandemic
- Low diagnostic power (virus/no virus)
- Cheap



Programmable

“Programmable”

- Distribute viral genome (Software)
- Sequence DNA/RNA reads
- Compare to viral genome (Software)
- Flexible
- Available immediately
- High diagnostic power (e.g. virus strain)
- Expensive

Virus Detectors



Non-Programmable

- Tailor chemistry to the given virus
- Cheaply mass-manufacture chemistry
- Kits must be physically distributed
- Inflexible
- Unavailable at beginning of pandemic
- Low diagnostic power (virus/no virus)
- Cheap
- Fast



Programmable

“Programmable”

- Distribute viral genome (Software)
- Sequence DNA/RNA reads
- Compare to viral genome (Software)
- Flexible
- Available immediately
- High diagnostic power (e.g. virus strain)
- Expensive
- Slow

Related work

Do signal comparison algorithms/accelerators from other domains already exist?

NATSA: A Near-Data Processing Accelerator for Time Series Analysis

Ivan Fernandez[§]

Ricardo Quisiant[§]

Christina Giannoula[†]

Mohammed Alser[‡]

Juan Gómez-Luna[‡]

Eladio Gutiérrez[§]

Oscar Plata[§]

Onur Mutlu[‡]

[§]University of Malaga

[†]National Technical University of Athens

[‡]ETH Zürich

Time series analysis is a key technique for extracting and predicting events in domains as diverse as epidemiology, genomics, neuroscience, environmental sciences, economics, and more. *Matrix profile*, the state-of-the-art algorithm to perform time series analysis, computes the most similar subsequence for a given query subsequence within a sliced time series. *Matrix profile* has low arithmetic intensity, but it typically operates on large amounts of time series data. In current computing systems, this data needs to be moved between the off-chip memory units and the on-chip computation units for

tection, which cannot be tolerated by many applications (e.g., vehicle safety systems [85]). Unlike approximate algorithms, exact algorithms [67] do not yield false positives or discordant dismissals, but can be very time-consuming on large time series data. Thus, *anytime* versions (aka interruptible algorithms) of exact algorithms are proposed to provide approximate solutions quickly [108, 112] and can return a valid result even if the user stops their execution early.

The state-of-the-art exact *anytime* method for motif and discord discovery is *matrix profile* [108], which is based on Eu-

ICCD 2020, pp. 120-129


Do raw signals contain extra information?

Trends in
Genetics

CellPress

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 Göke ^{1,6,*}

Nanopore sequencing provides signal data corresponding to the nucleotide motifs sequenced. Through machine learning-based methods, these signals are translated into long-read sequences that overcome the read size limit of short-read sequencing. However, analyzing the raw nanopore signal data provides many more opportunities beyond just sequencing genomes and transcriptomes: algorithms that use machine learning approaches to extract biological information from these signals allow the detection of DNA and RNA

Highlights

Nanopore sequencing accuracy has increased to 98.3% as new-generation base callers replace early generation hidden Markov model basecalling algorithms with neural network algorithms.

Machine learning methods can classify

[https://www.cell.com/trends/genetics/fulltext/S0168-9525\(21\)00257-2](https://www.cell.com/trends/genetics/fulltext/S0168-9525(21)00257-2)

Which sequencing technology is best for virus detection?

nature
biomedical engineering

ARTICLES

<https://doi.org/10.1038/s41551-021-00754-5>

 Check for updates

Massively scaled-up testing for SARS-CoV-2 RNA via next-generation sequencing of pooled and barcoded nasal and saliva samples

Joshua S. Bloom^{1,2,3}  , Laila Sathe⁴ , Chetan Munugala^{1,2}, Eric M. Jones³, Molly Gasperini³, Nathan B. Lubock³, Fauna Yarza³ , Erin M. Thompson³, Kyle M. Kovary³, Jimin Park³, Dawn Marquette⁵ , Stephania Kay⁵, Mark Lucas⁵, TreQuan Love⁵, A. Sina Booeshaghi⁶, Oliver F. Brandenburg^{1,2,7}, Longhua Guo^{1,2,7}, James Boocock^{1,2,7}, Myles Hochman³, Scott W. Simpkins³ , Isabella Lin^{1,4}, Nathan LaPierre⁸, Duke Hong⁵, Yi Zhang¹, Gabriel Oland⁹ , Bianca Judy Choe¹⁰, Sukantha Chandrasekaran⁴, Evann E. Hilt⁴, Manish J. Butte^{11,12} , Robert Damoiseaux^{13,14,15}, Clifford Kravitz¹⁶, Aaron R. Cooper³, Yi Yin¹, Lior Pachter¹⁷, Omai B. Garner⁴, Jonathan Flint^{1,18}, Eleazar Eskin^{1,5,8}, Chongyuan Luo¹, Sriram Kosuri^{3,19}  , Leonid Kruglyak^{1,2,7}   and Valerie A. Arboleda^{1,4}  

Nature Biomedical Engineering 5, 657–665 (2021).

Is early “viral”/”non-viral” labelling possible with Illumina as well?

