P&S Genomics

Lecture 13b: TargetCall

Meryem Banu Cavlak

ETH Zürich

Spring 2023

1 June 2023

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

Meryem Banu Cavlak, Gagandeep Singh, Mohammed Alser,
Can Firtina, Joel Lindegger, Mohammad Sadrosadati,
Nika Mansouri Ghiasi, Can Alkan, Onur Mutlu

bioRxiv

Code

ETH zürich

Bilkent University

SAFARI

TargetCall Summary

Motivation: Basecalling consumes up to 84.2% of total execution time and bottlenecks the genome analysis pipeline

Problem: The majority of the reads do not match the reference genome (i.e., useless reads) and thus are discarded after basecalling, wasting the basecalling computation

Goal: Eliminating the wasted computation in basecalling while maintaining high accuracy, scalability and adaptability

Key Idea: Filter out useless reads **before basecalling** with a highly accurate and high-performance pre-basecalling filter

TargetCall: **New pre-basecalling filter**

- **LightCall:** A *light-weight* basecaller that computes **noisy reads** with *high performance*
- **Similarity Check:** Computes the similarity of the noisy read to the reference genome

Results:

- Improves the *end-to-end performance* of basecalling by 3.3× over the state-of-the-art basecaller by filtering out 94.7% of the useless reads
- Achieves better *performance*, *throughput, recall* and *precision* than the state-of-the-art targeted sequencing approaches.

TargetCall Outline

Background and Motivation

TargetCall: Pre-Basecalling Filter

Use Cases

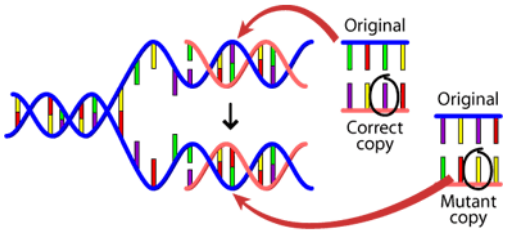
Evaluation

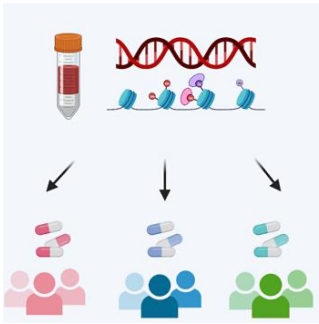
Conclusion

Genome Sequence Analysis

Genome Sequencing: Enables us to determine the order of the DNA sequence in an organism's genome

- Plays a pivotal role in:
  - Precision medicine
  - Outbreak tracing
  - Understanding of evolution



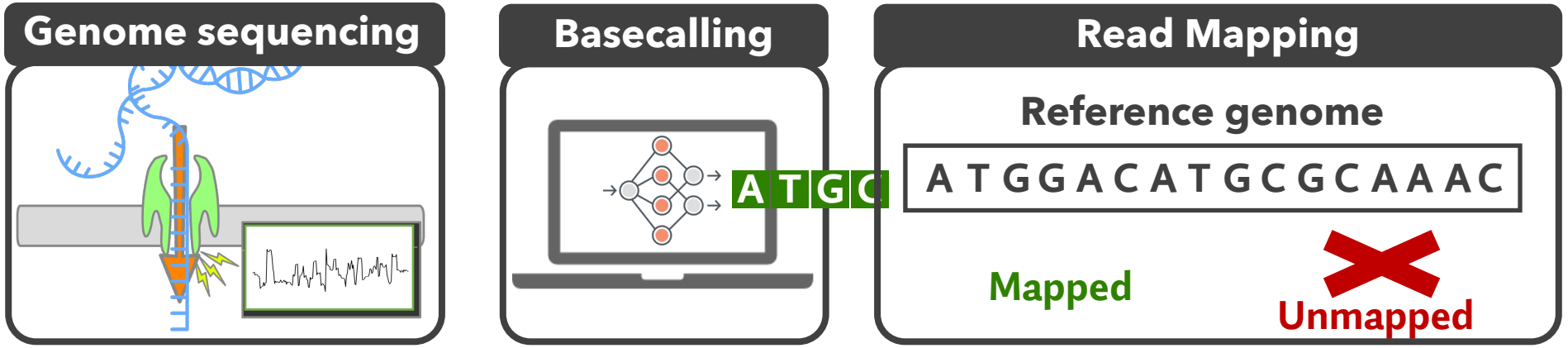


Nanopore Sequencing: a **widely used** sequencing technology that

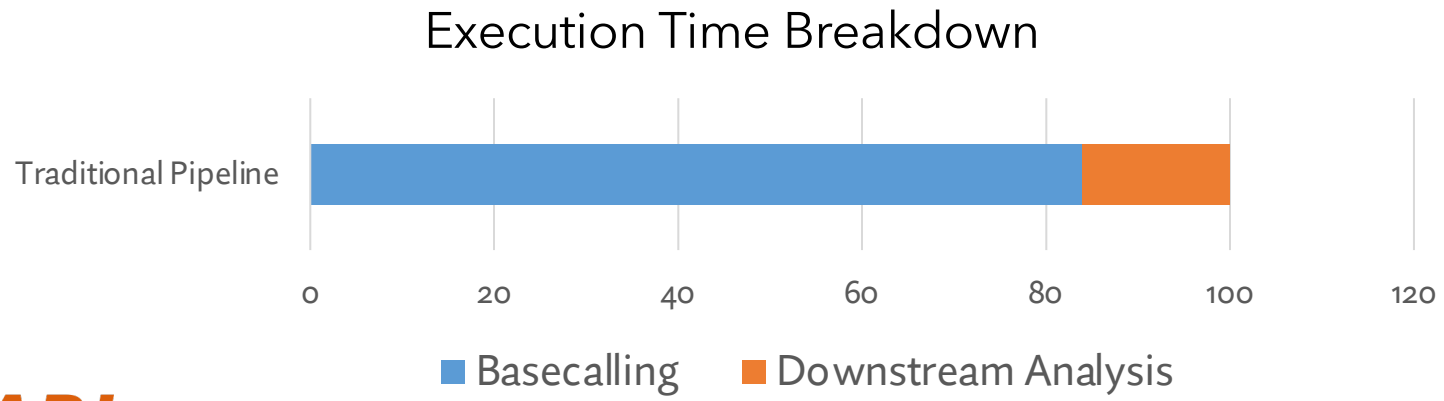
- Produces long reads (i.e., 10Kbp-100Mbp)
- Has high throughput
- Is low cost

Option 1: Traditional Pipeline

Traditional Nanopore Sequence Analysis Pipeline

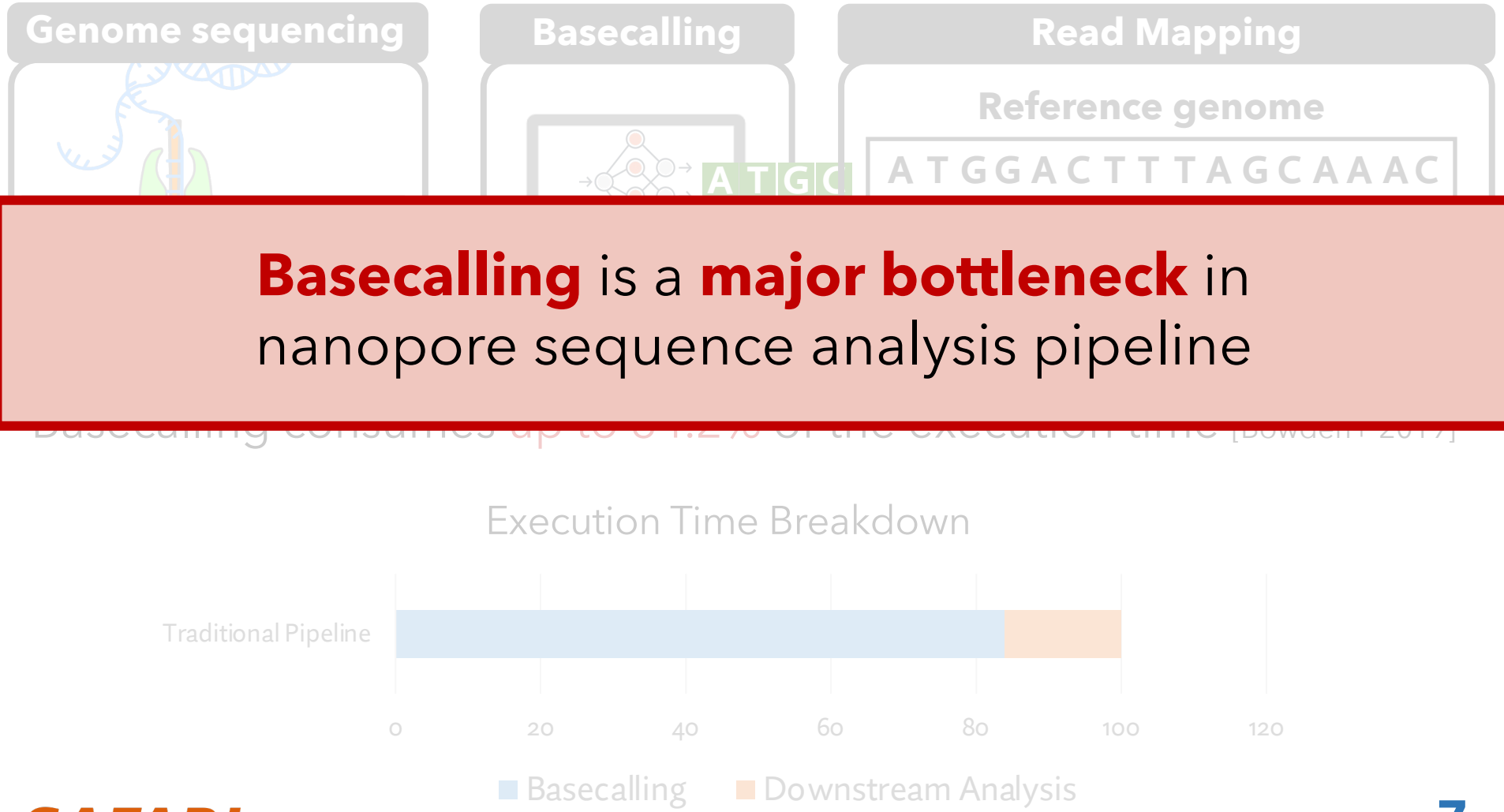


Basecalling consumes up to 84.2% of the execution time [Bowden+ 2019]



Option 1: Traditional Pipeline

Traditional Nanopore Sequence Analysis Pipeline

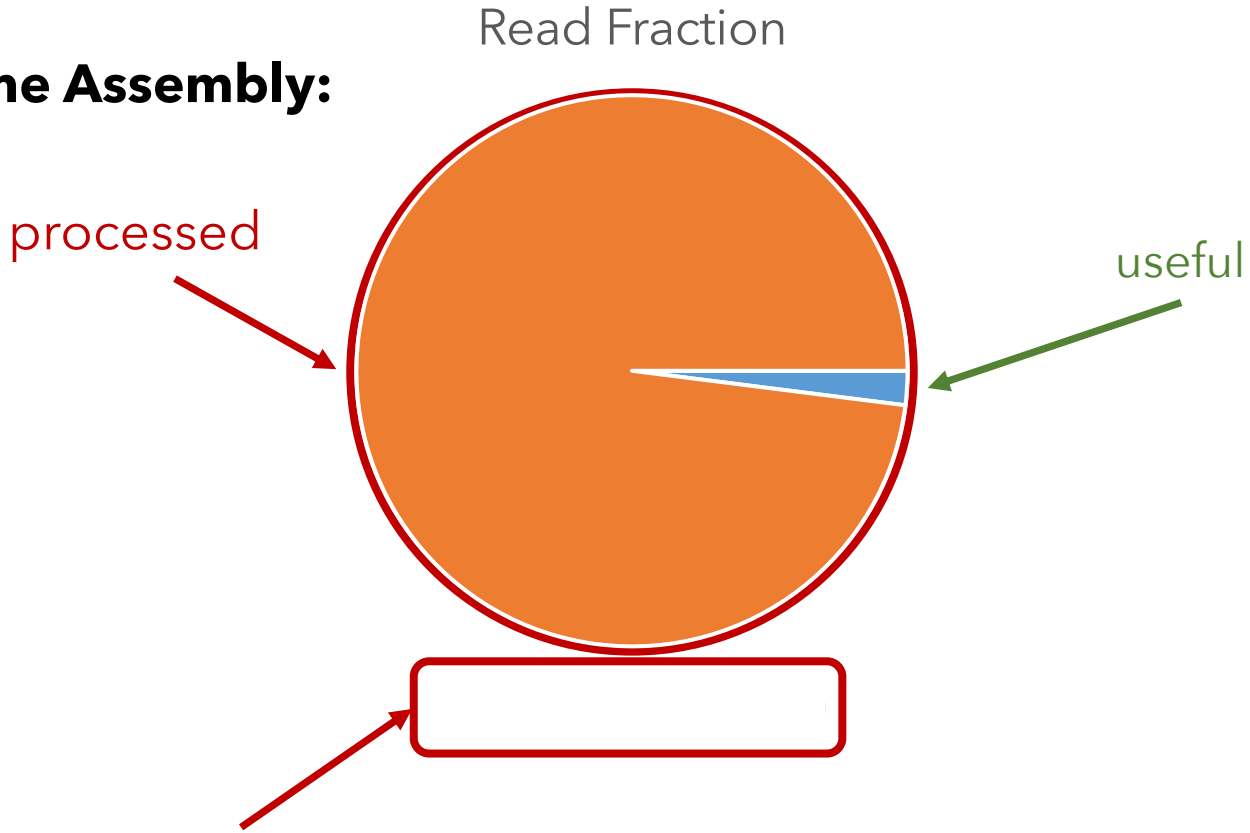


Basecalling is a **major bottleneck** in nanopore sequence analysis pipeline

Key Observation

Majority of the basecalled reads are discarded in the later downstream analysis

SARS-CoV-2 Genome Assembly:
[Dunn+, 2021]

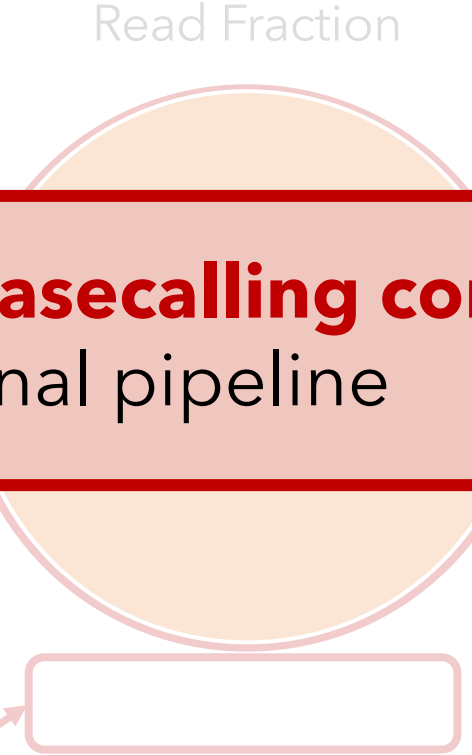


A read is **useless** if it does **not match** the reference genome

Key Observation

Majority of the basecalled reads are discarded in the later downstream analysis

SARS-CoV-2 Genome Assembly:
[Dunn+, 2021]



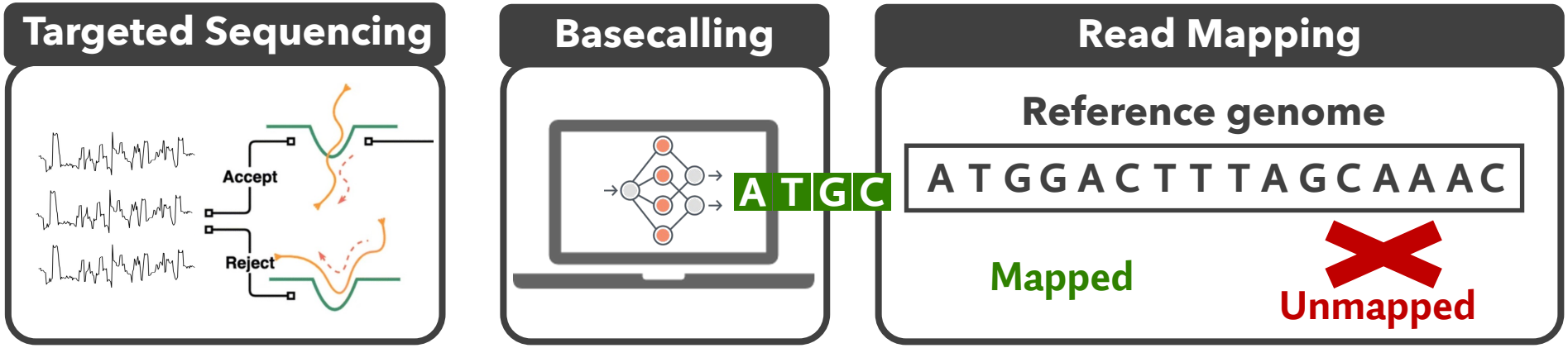
Useless reads **waste basecalling computation** in traditional pipeline

A read is **useless** if it does **not match** the reference genome

Option 2: Targeted Sequencing

Traditional Pipeline: Sequence all reads

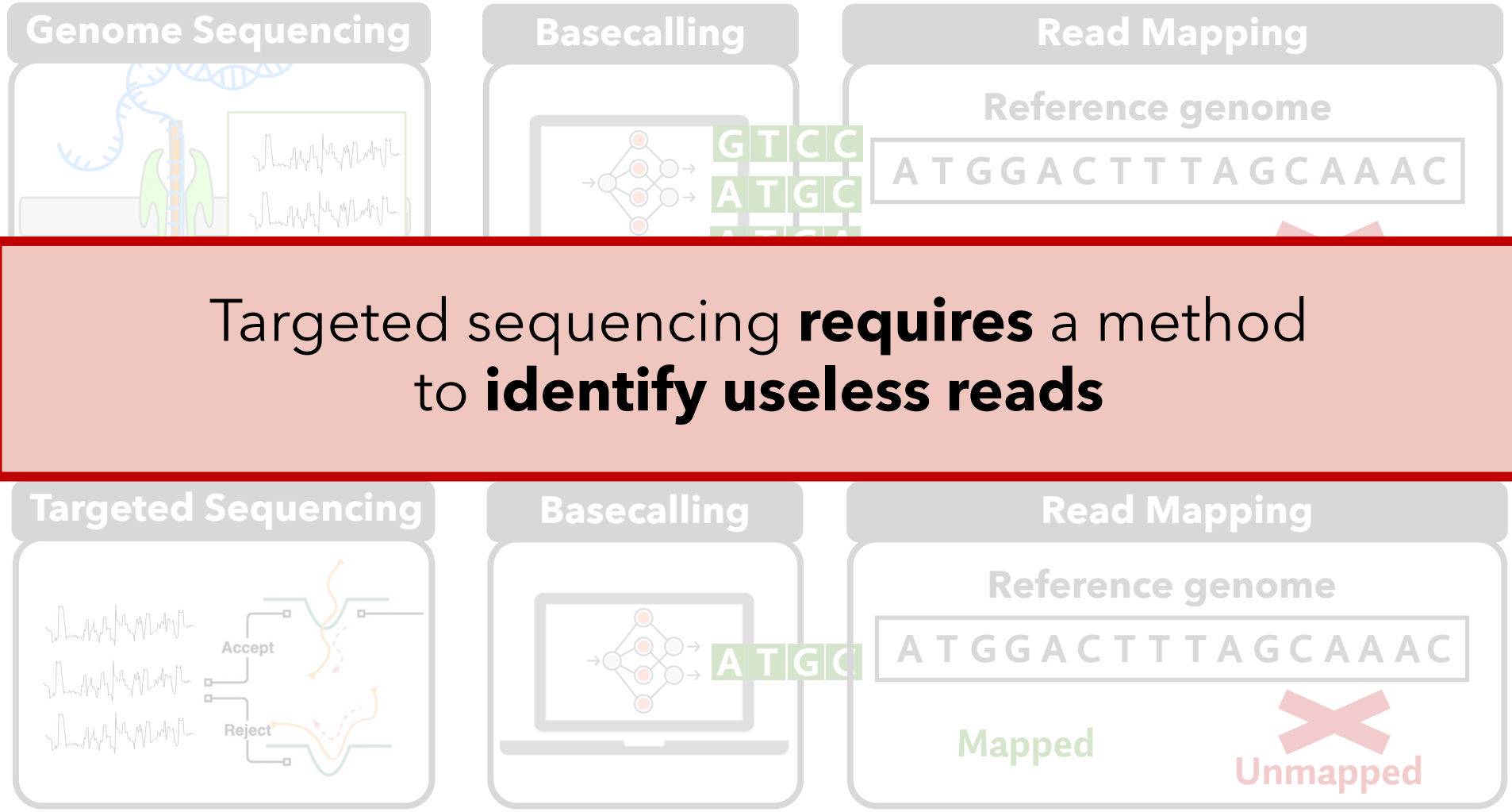
Targeted Sequencing: Selectively sequence useful reads



Nanopore sequencers can stop sequencing useless reads

Option 2: Targeted Sequencing

Traditional Sequencing: Sequence all reads



Nanopore sequencers can stop sequencing of useless reads

Targeted Sequencing Limitations

Current targeted sequencing approaches to identify useful reads suffer **at least** from *one of the following*:

Low Sensitivity

Falsely reject many useful reads

Poor Scalability

Have poor performance and accuracy for large reference genomes

Lack of Adaptability

Require neural network training for each genome sequencing use case

Targeted Sequencing Limitations

Current targeted sequencing approaches to identify useful reads suffer **at least** from *one of the following*:

Low Sensitivity

Targeted sequencing approaches are **not robust** for *eliminating the wasted computation* in basecalling

Lack of Adaptability

Require neural network training for each genome sequencing use case

Our Goal

Eliminate the wasted computation in basecalling
while maintaining high sensitivity in keeping *useful* reads,
scalability to large reference genomes and adaptability.

TargetCall Outline

Background and Motivation

TargetCall: Pre-Basecalling Filter

Use Cases

Evaluation

Conclusion

Our Proposal: Pre-Basecalling Filter

Option 1 - Traditional Pipeline:

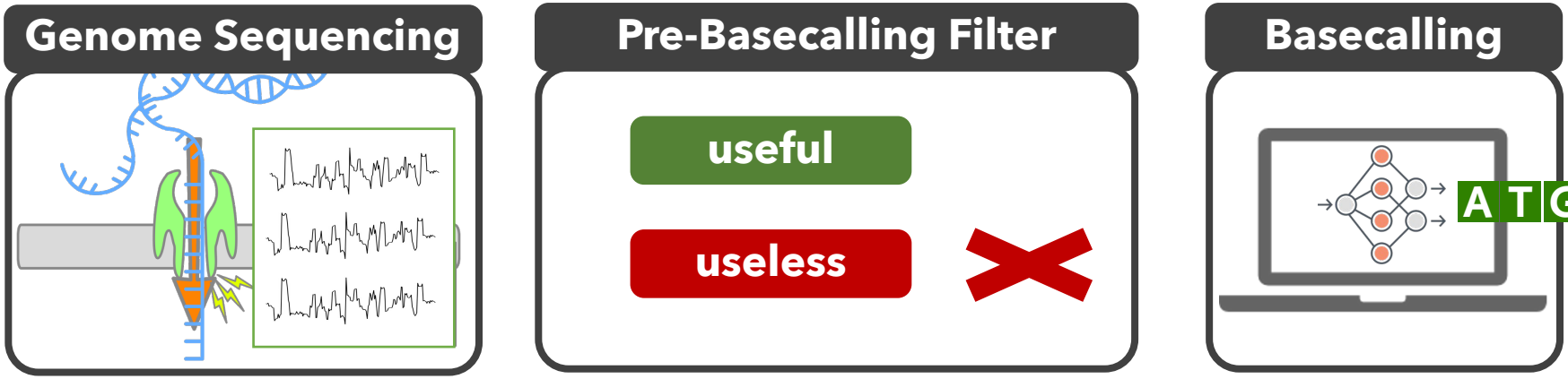
Sequence & basecall **all** reads

Option 2 – Targeted Sequencing:

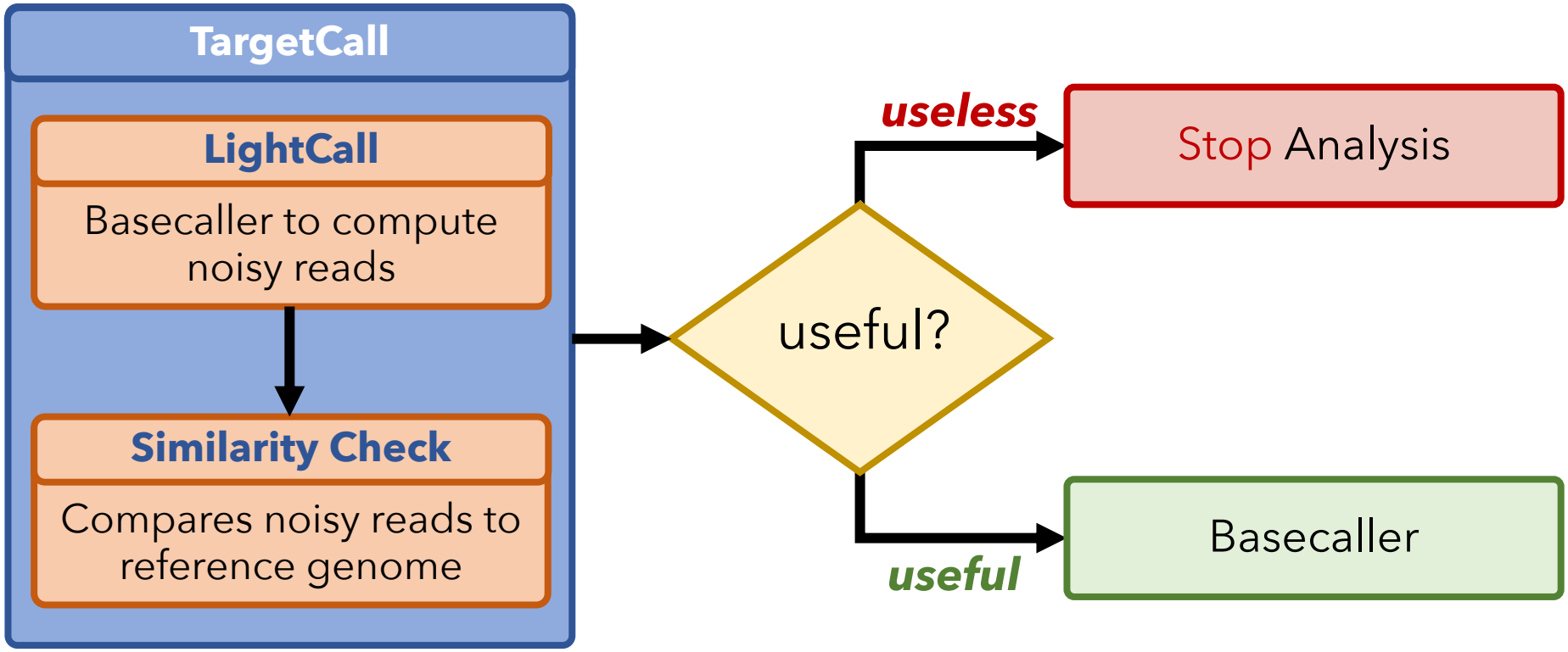
Sequence **selectively**

Our Proposal - Pre-Basecalling Filter:

Selectively basecall useful reads



TargetCall Overview



TargetCall Overview

TargetCall has:

High Sensitivity

with its highly accurate Similarity Check component

Good Scalability

with a reference genome size independent LightCall model

Adaptability

with a generic LightCall model that does not require NN retraining

TargetCall Overview

TargetCall has:

High Sensitivity

with its highly accurate Similarity Check component

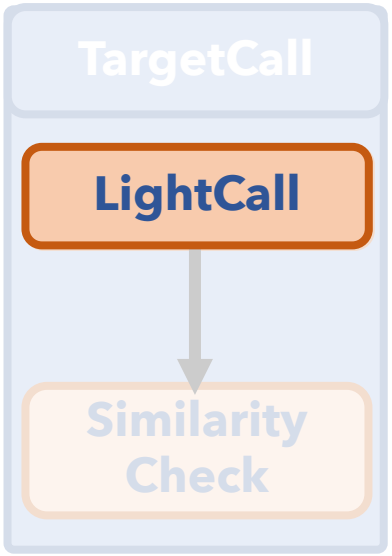
TargetCall is a **robust** solution for
eliminating the wasted computation in basecalling

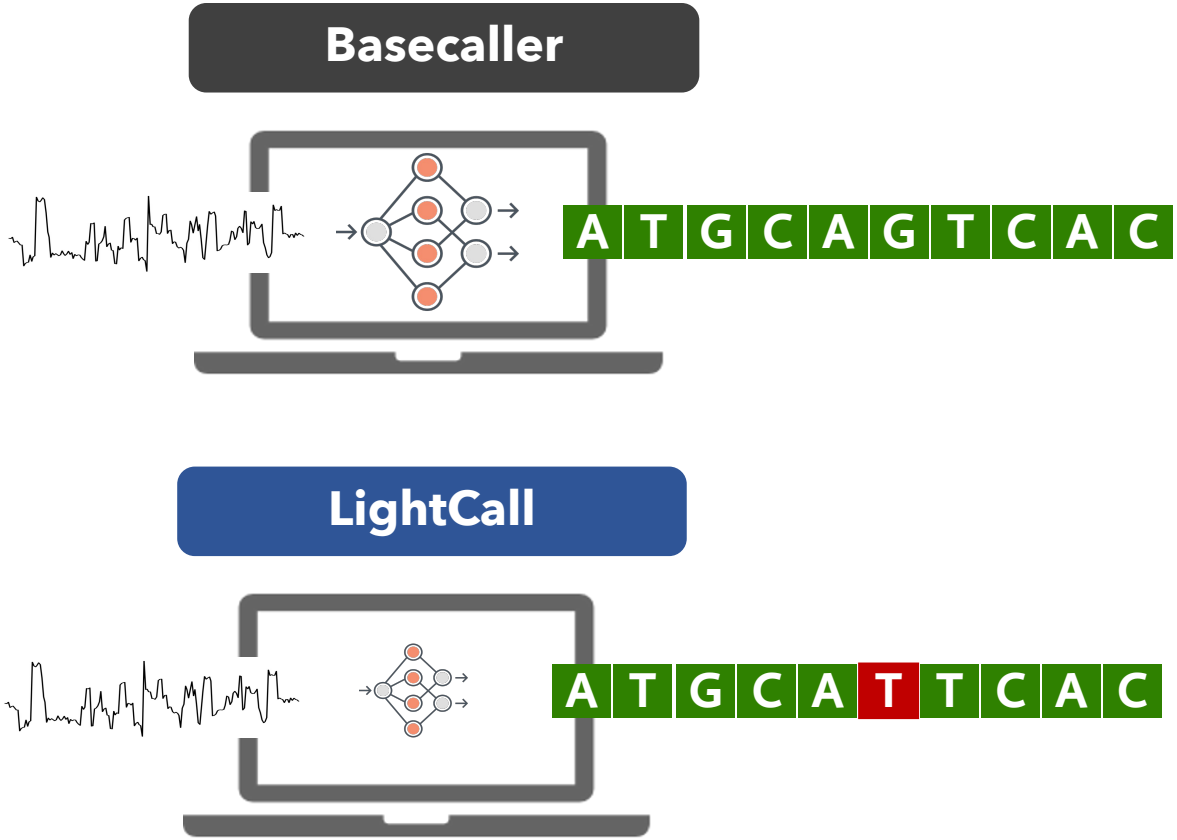
Adaptability

with a generic LightCall model that does not require NN retraining

LightCall

A light-weight basecaller that produces noisy reads

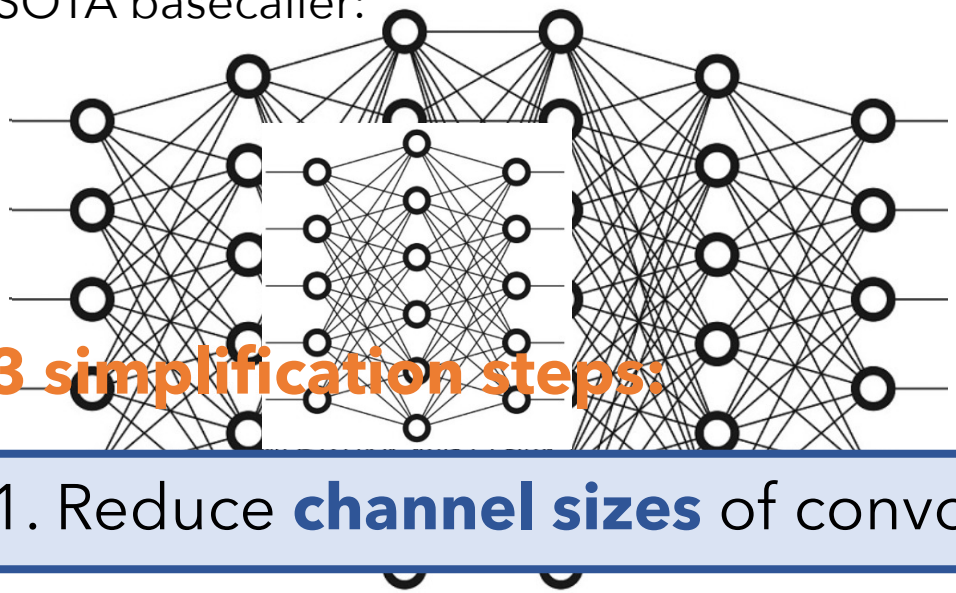




LightCall

Design based on by simplifying the state-of-the-art **basecaller** model architectures while maintaining most accuracy benefits

SOTA basecaller:



LightCall:

3 simplification steps:

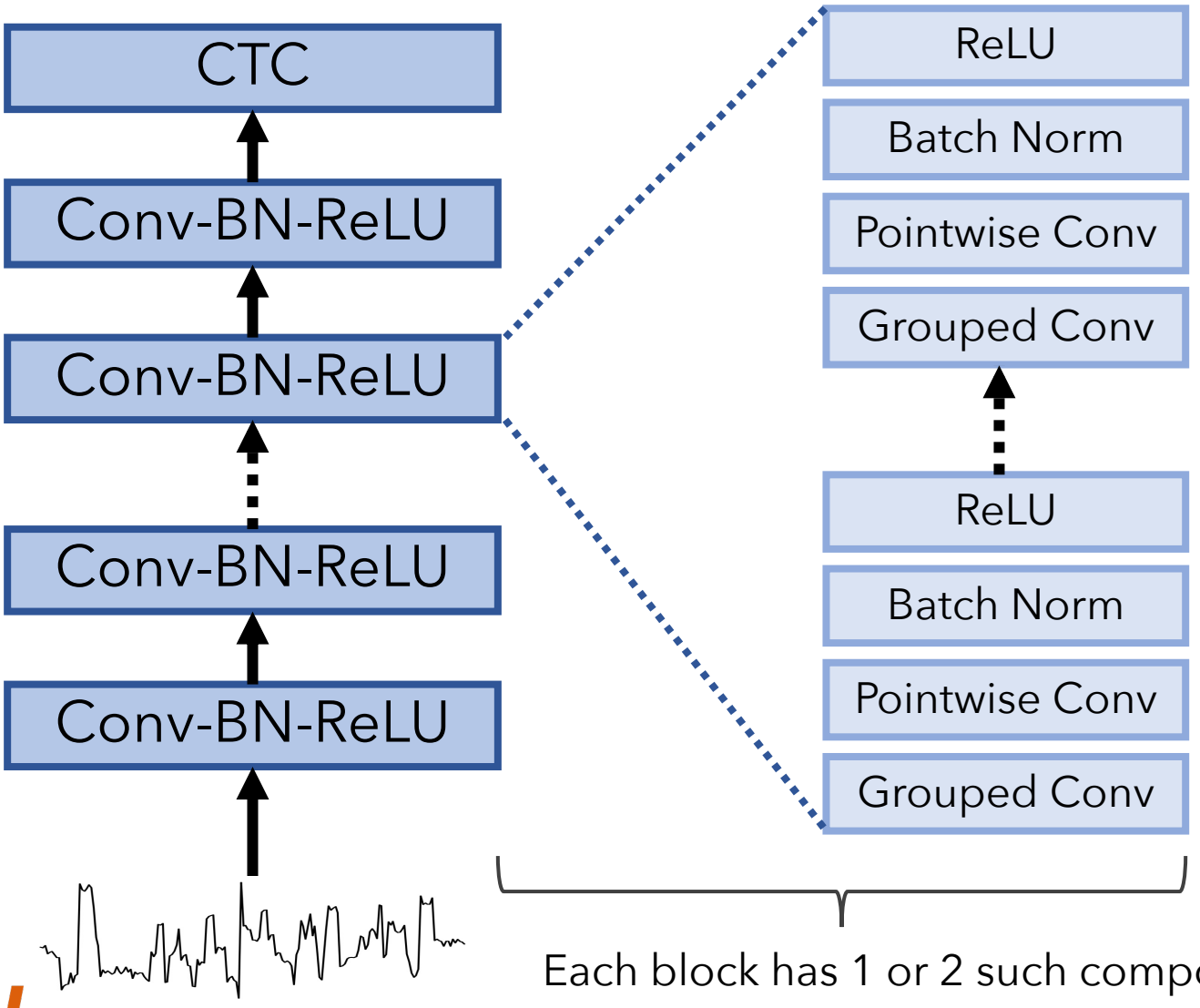
1. Reduce **channel sizes** of convolution layers

2. Remove **skip connections**

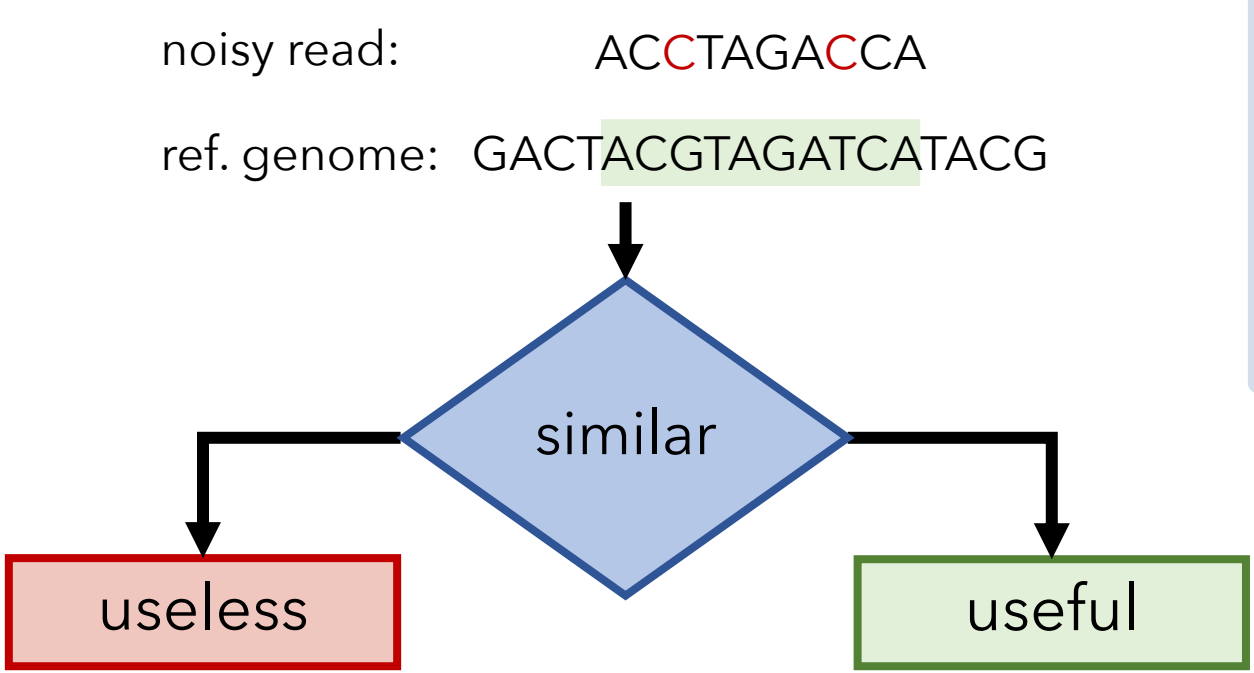
3. Reduce number **basic convolutional blocks**

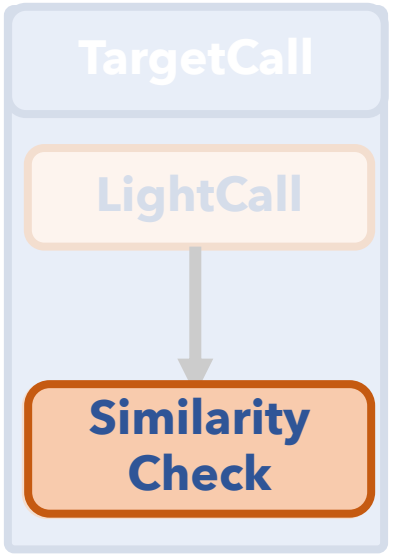
LightCall

LightCall is a series of convolutional blocks



Similarity Check





- Similarity Check module: minimap2

- **LightCall + Similarity Check:**
  Up to 99.45% sensitive in keeping useful reads
  - 0.55% can be tolerated via **sequencing-depth-of-coverage**

TargetCall Outline

Background and Motivation

TargetCall: Pre-Basecalling Filter

Use Cases

Evaluation

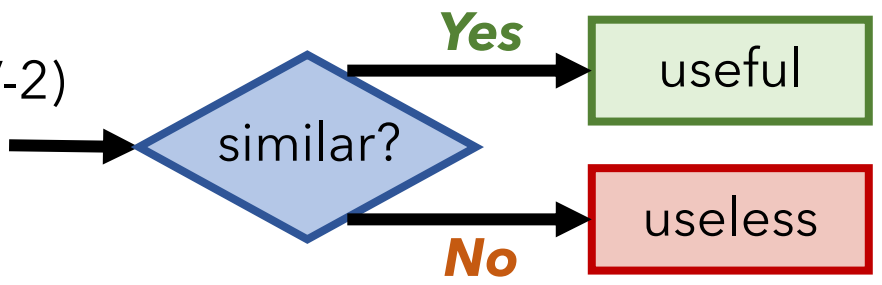
Conclusion

TargetCall Use Cases

Show the scalability and adaptability of TargetCall:

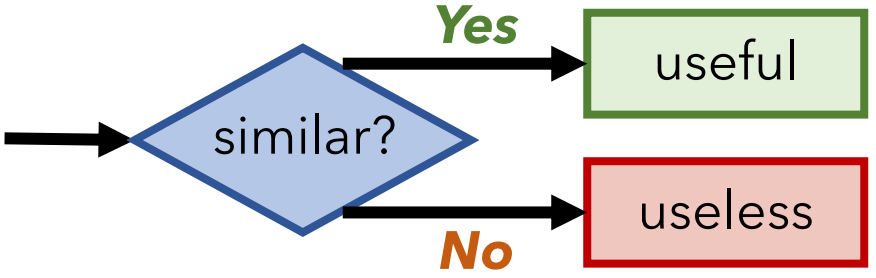
1. SARS-CoV-2 Detection

Reference Genome: **Small** (SARS-CoV-2)
Reads: SARS-CoV-2 & Human



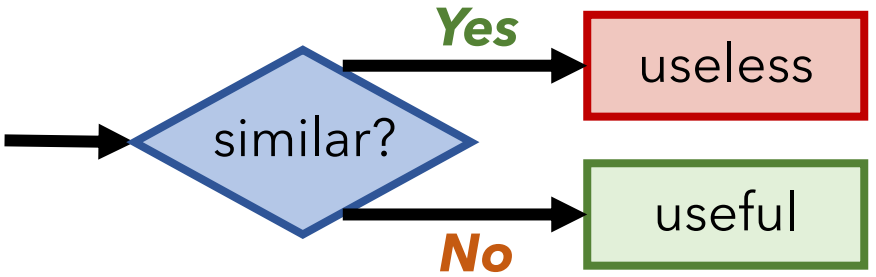
2. Viral Detection

Reference Genome : **Complex** (Viral)
Reads: Bacterial & Viral



3. Sepsis Detection

Reference Genome : **Large** (Human)
Reads: Bacterial & Human



TargetCall Outline

Background and Motivation

TargetCall: Pre-Basecalling Filter

Use Cases

Evaluation

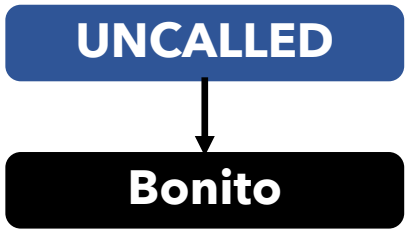
Conclusion

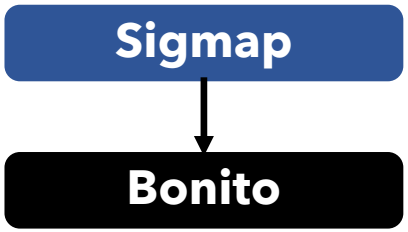
Evaluation Methodology - Experiments

1. Benefits of Pre-Basecalling Filtering
   - **Baseline:** Bonito
   - **Methodology:** Compare 

and 

   - **Evaluation Metric:** Basecalling speedup
2. Comparison against Targeted Sequencing
   - **Baseline:** UNCALLED [Kovaka+, 2020] & Sigmap [Zhang+, 2021]
   - **Methodology:** **Repurpose** labelling mechanism of the targeted sequencing approaches as pre-basecalling filters, compare:





   - **Evaluation Metric:** Execution time, recall and precision

Evaluation Methodology - Datasets

Read Sets:

- 5 different read sets from various organisms
  - 4 read sets are sampled from prior work [Wick+ 2019, Zook+ 2019, CADDE 2020]
  - 1 simulated read set using DeepSimulator
- We open source the datasets

Reference Genomes:

- 4 different reference genomes with various
  - Reference genome size
  - Ratio of useful reads

Evaluation Methodology - System

We evaluate **TargetCall** using:

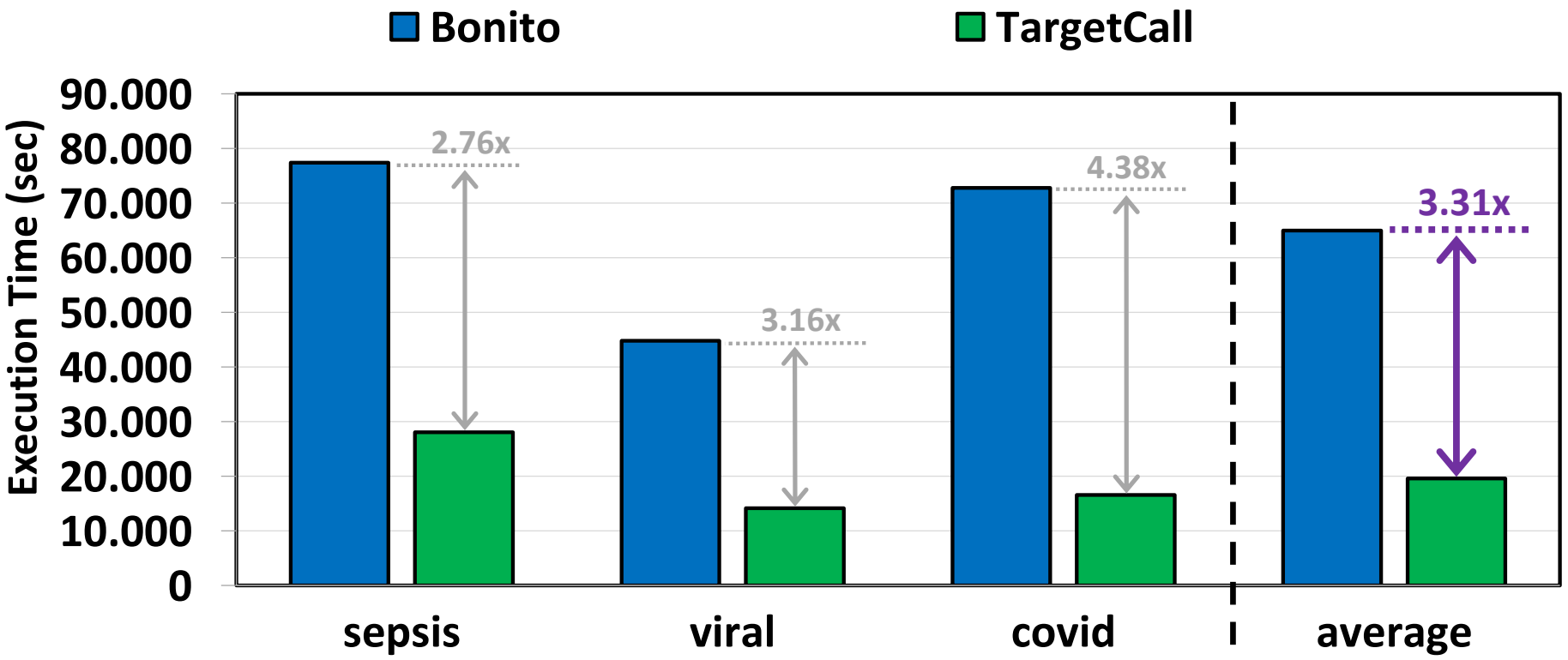
- NVIDIA A100 and TITAN V GPU for LightCall
- AMD EPYC 7742 CPU with **~0.2 TB DDR4 DRAM** for Similarity Check

We evaluate **Sigmap** and **UNCALLED** using:

- AMD EPYC 7742 CPU with **~1 TB DDR4 DRAM**

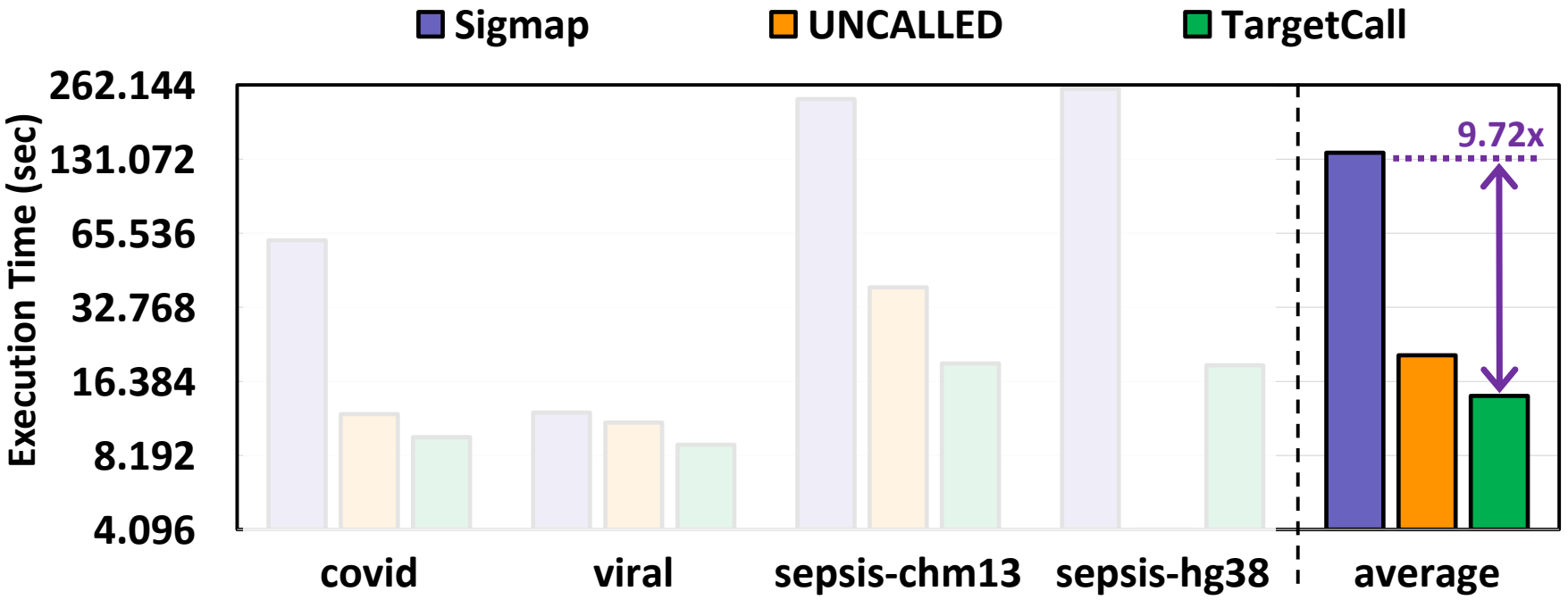
Sigmap and UNCALLED **require more than 0.2 TB of DRAM** for **large** reference genomes

TargetCall – Basecalling Speedup



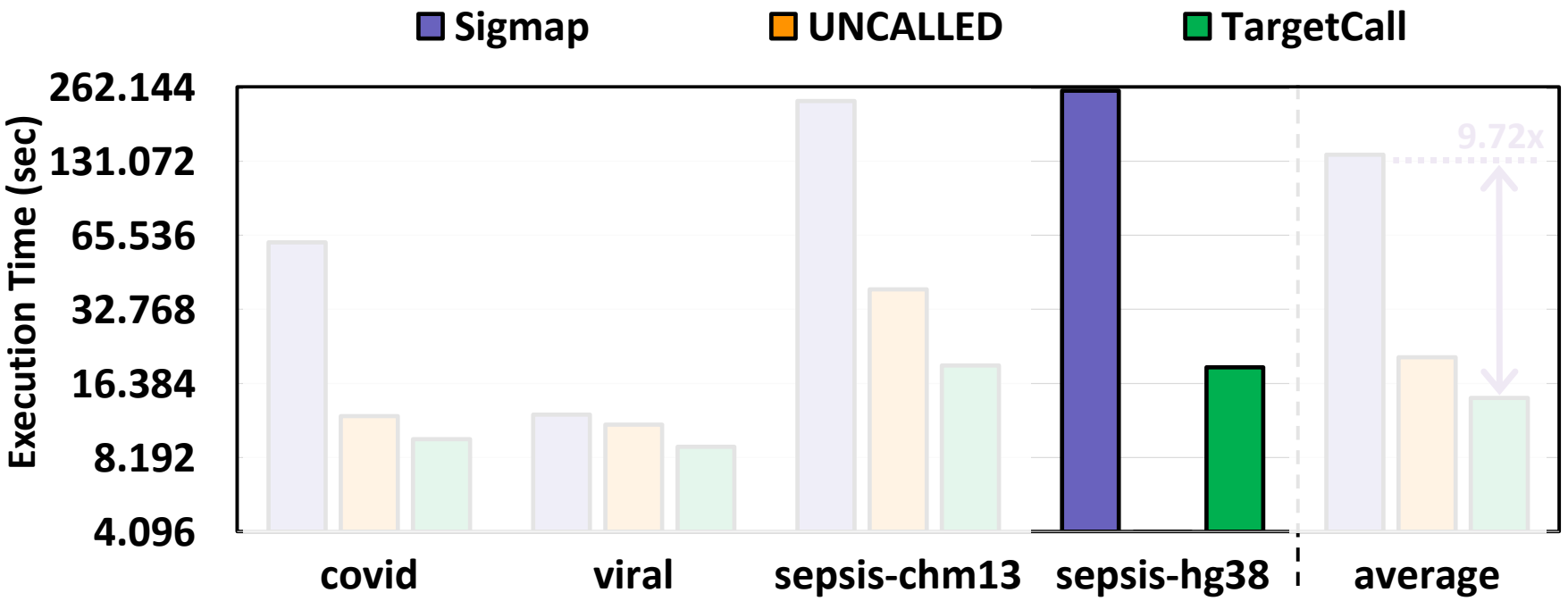
On average, TargetCall provides **3.3x basecalling speedup** over Bonito

Comparison to SOTA - Performance (1/3)



TargetCall provides **1.5x**/**9.7x** speedup
over UNCALLED/Sigmap

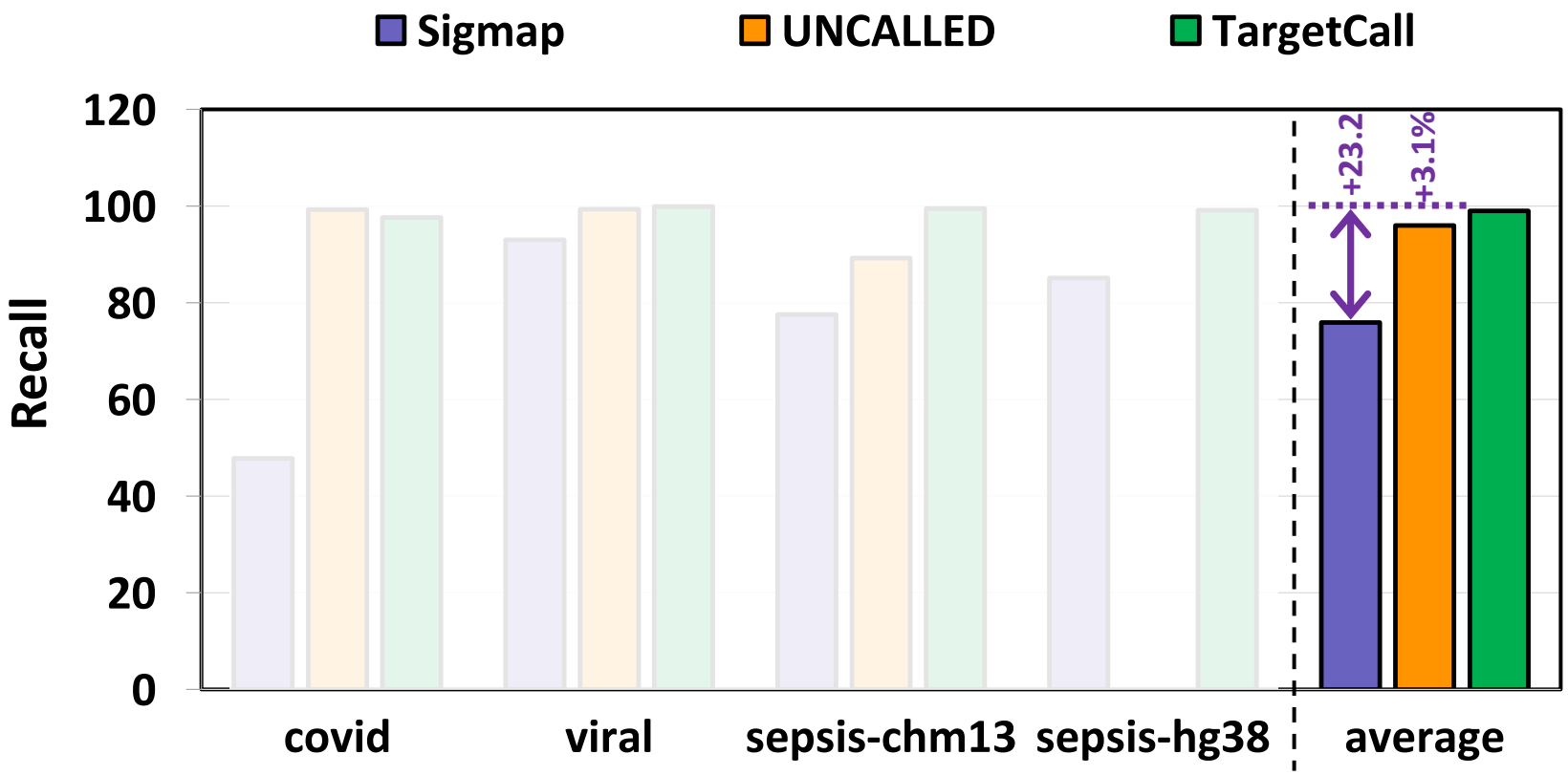
Comparison to SOTA - Performance (1/3)



TargetCall provides **higher** speedup improvement with **a large reference genome:**

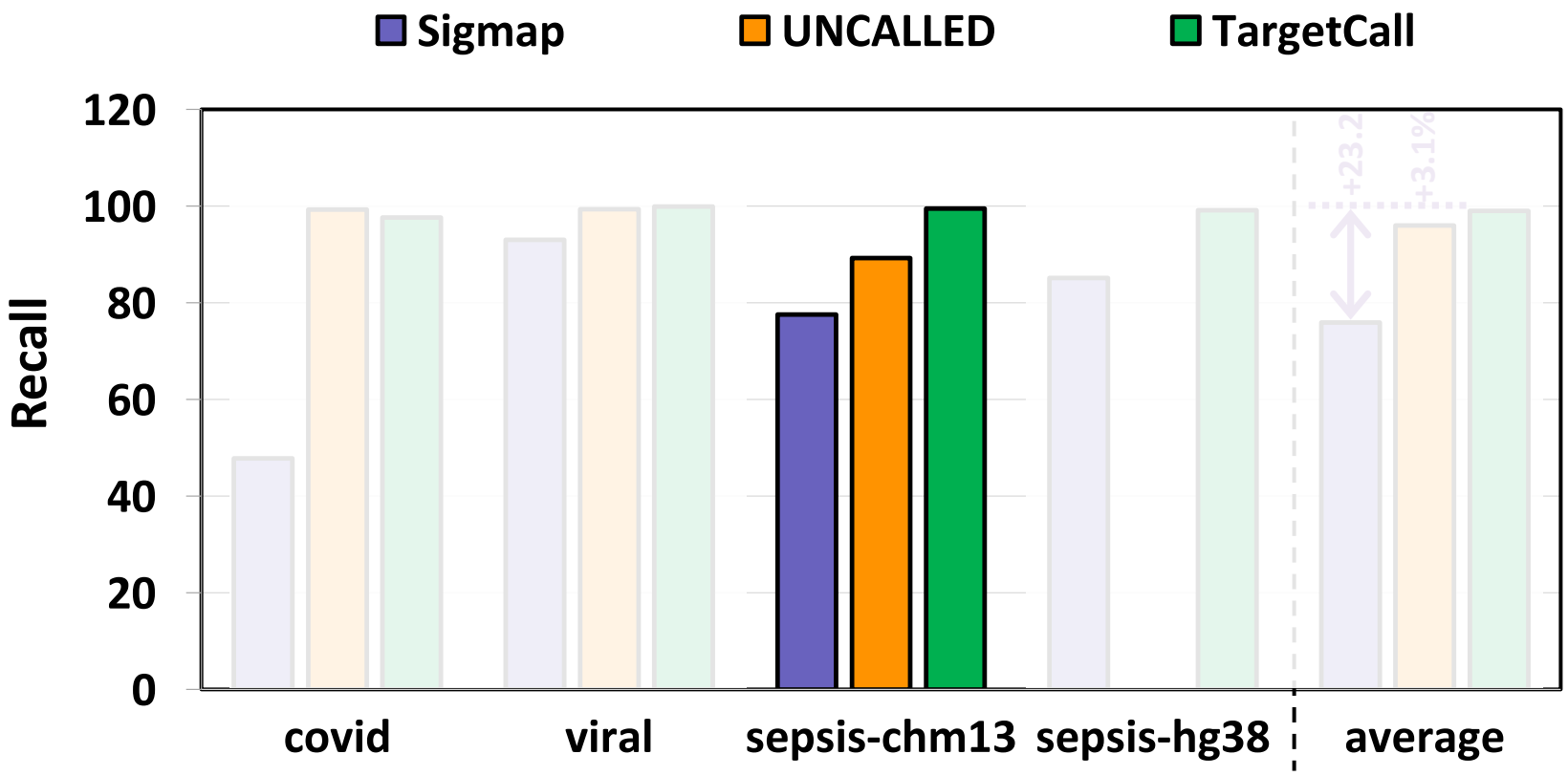
- **13.3x speedup** over Sigmap
- UNCALLED is **inapplicable**: **cannot** generate the index

Comparison to SOTA - Recall (2/3)



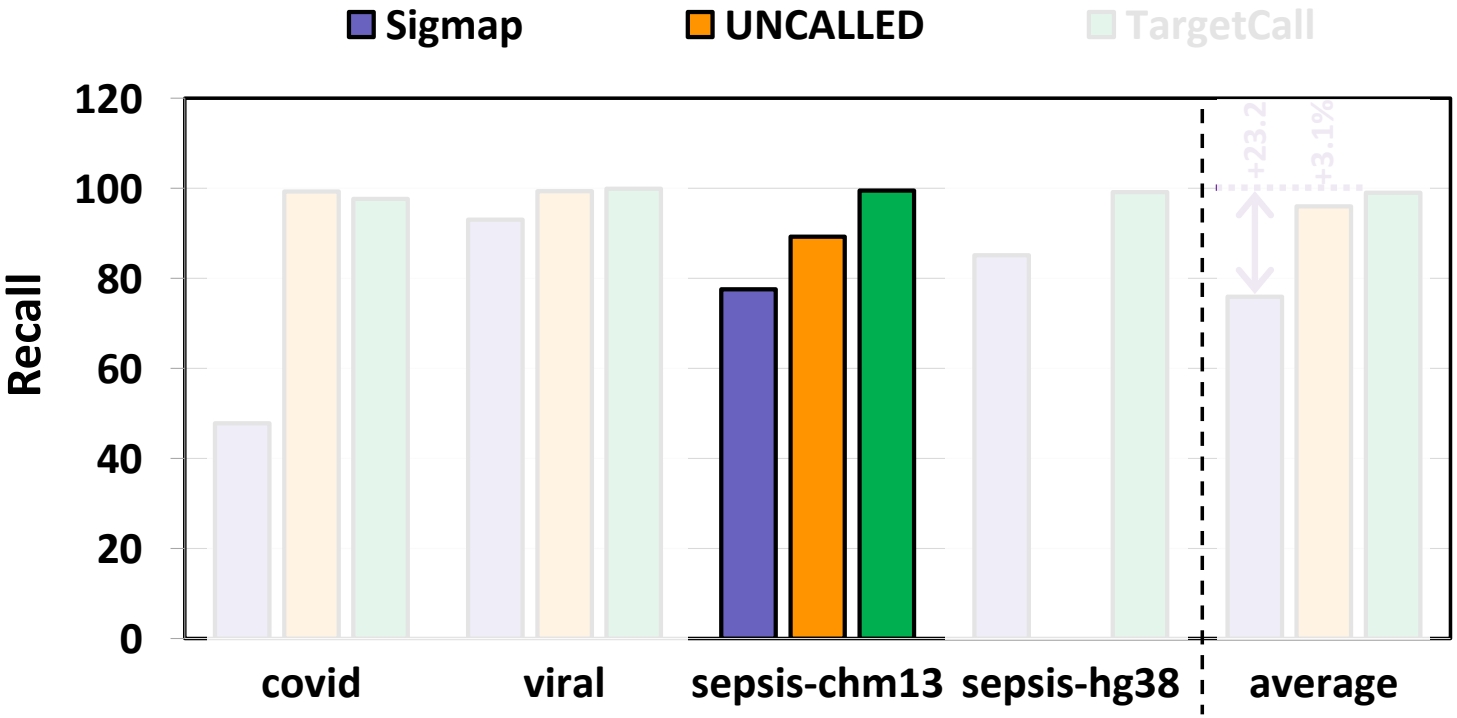
On average, TargetCall's **recall** is **99.1%**

Comparison to SOTA - Recall (2/3)



TargetCall's recall benefits improve (**21.9%-10.3%**) with **increasing** reference genome size

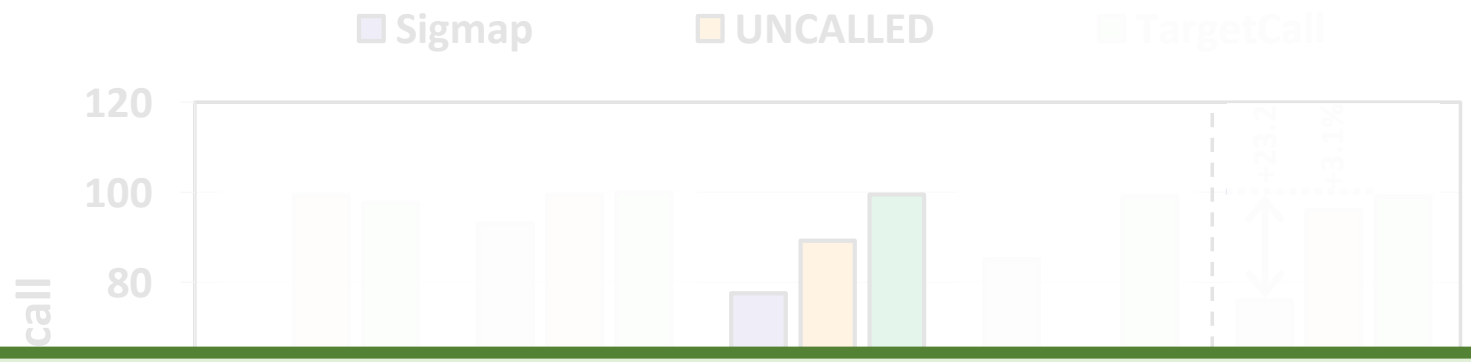
Comparison to SOTA - Recall (2/3)



Sepsis use case doesn't measure the recall for finding human reads

The recall for finding human reads:
- Sigmap/UNCALLED: **40.6%**/**53.7%**
- TargetCall: **96.2%**

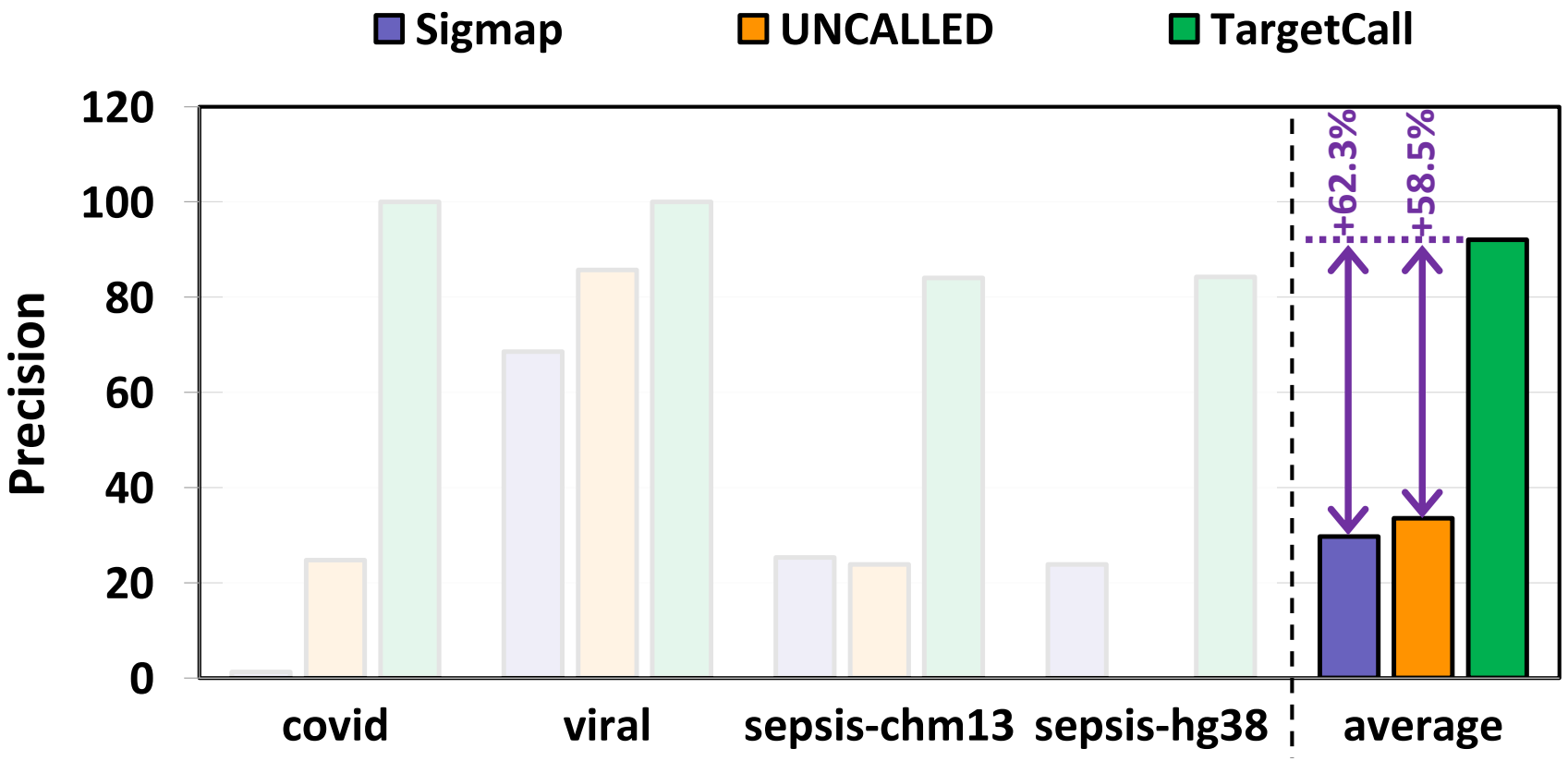
Comparison to SOTA - Recall (2/3)



TargetCall consistently provides high recall for **all** reference genome sizes tested

The recall for finding human reads:
- Sigmap/UNCALLED: **40.6%**/**53.7%**
- TargetCall: **96.2%**

Comparison to SOTA - Precision (3/3)



On average, TargetCall's **precision** is **92.1%**

More Details in the Paper

- Details of targeted sequencing
- Details of LightCall design
- More details on evaluation methodology
- More evaluation results
  - **Basecalling speedup, recall** and **precision** of **different LightCall** architectures to finalize the TargetCall design
  - **End-to-end accuracy analysis** using relative abundances
  - **End-to-end performance** results including variant calling
  - **Throughput** comparison: **42x**/**1124x** over Sigmap/UNCALLED
  - **Peak memory** discussion

More Details in the Paper

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

Meryem Banu Cavlak[1] Gagandeep Singh[1] Mohammed Alser[1] Can Firtina[1] Joël Lindegger[1]
Mohammad Sadrosadati[1] Nika Mansouri Ghiasi[1] Can Alkan[2] Onur Mutlu[1]
[1]*ETH Zürich* [2]*Bilkent University*

bioRxiv

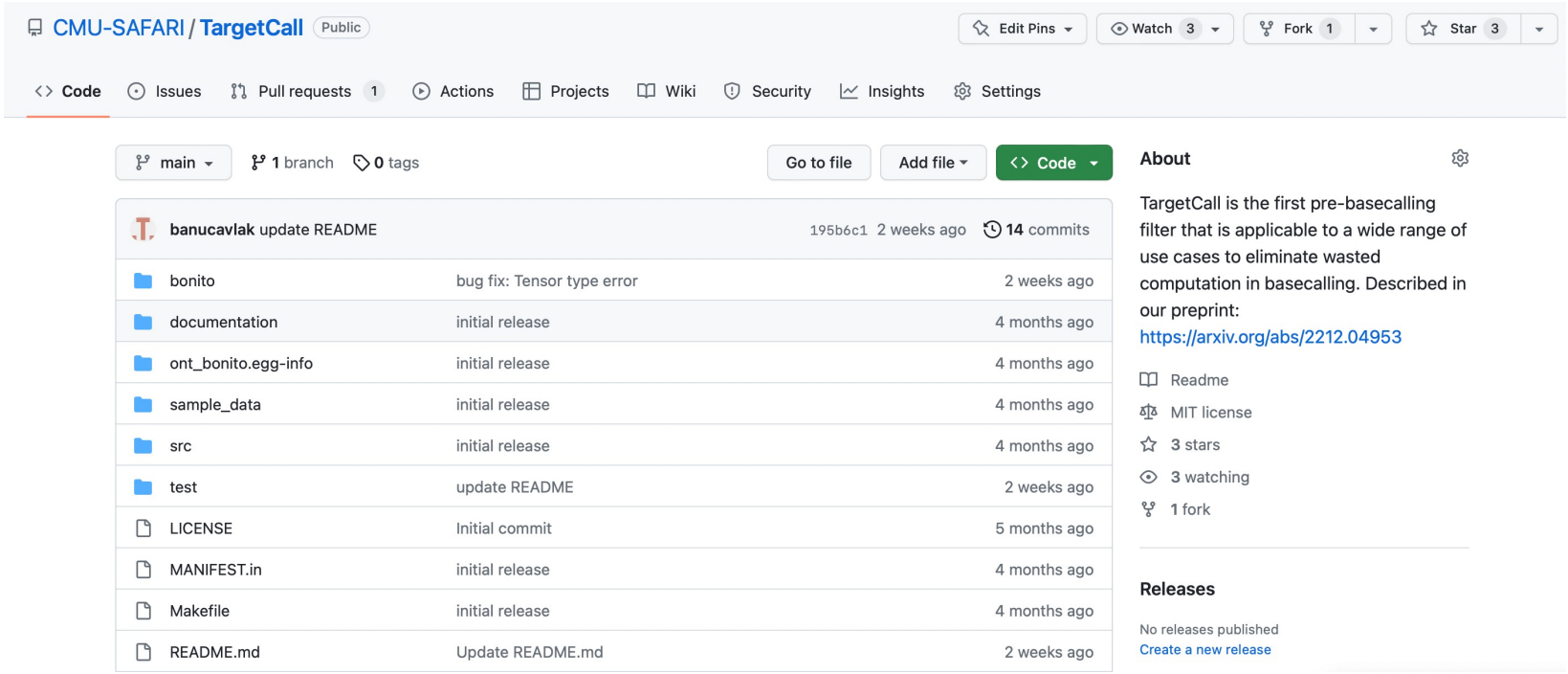
Code

TargetCall – GitHub Page

Artifacts are **open-sourced**

DOI 10.5281/zenodo.7335545

https://github.com/CMU-SAFARI/TargetCall



TargetCall Outline

Background and Motivation

TargetCall: Pre-Basecalling Filter

Use Cases

Evaluation

Conclusion

TargetCall Summary

TargetCall: An accurate, scalable and adaptable **pre-basecalling filter:**

- **LightCall:** A *light-weight* basecaller that computes noisy reads with *high performance*
- **Similarity Check:** Computes the similarity of the noisy read to the reference genome

Results:

- TargetCall significantly improves basecalling performance for **three sample use cases** by filtering out majority of the useless reads
- Achieves better recall, precision, performance and throughput than the state-of-the-art targeted sequencing approaches **repurposed** as pre-basecalling filters

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

Meryem Banu Cavlak, Gagandeep Singh, Mohammed Alser,
Can Firtina, Joel Lindegger, Mohammad Sadrosadati,
Nika Mansouri Ghiasi, Can Alkan, Onur Mutlu

bioRxiv

Code

ETH zürich

Bilkent University

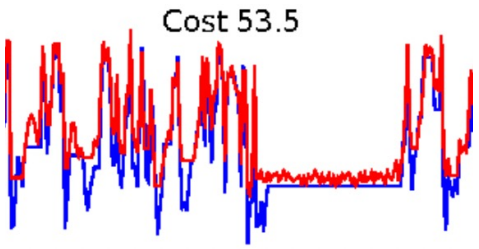
SAFARI

Backup Slides

Targeted Sequencing

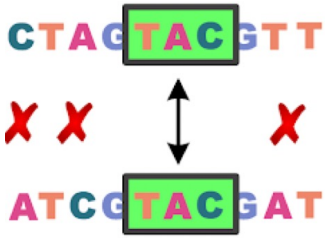
Set of techniques to discard off-target reads during *sequencing*

Raw Signal Comparison



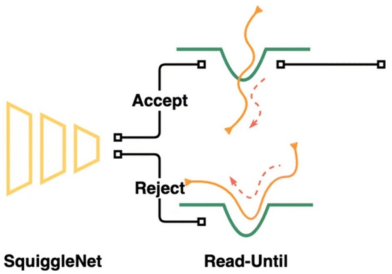
Sigmap
SquiggleFilter

Sequence Comparison



ReadFish
UNCALLED

Machine Learning



SquiggleNet
BaseLess

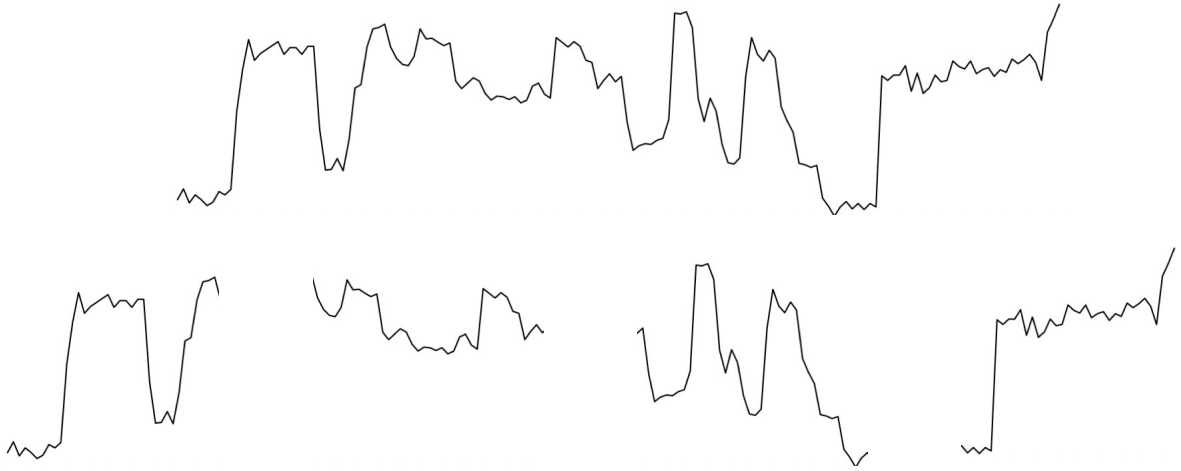
Some of these works can be **partially repurposed** as pre-basecalling filters.

Basecalling

Basecallers use complex DNN models

1

They split the raw signals into fixed length chunks



2

They basecall chunks independently

ACGTA GAGGC TCTC GACTCA

3

They stitch the chunks back

ACGTAGAGGCTCTCGACTCA

Evaluation – System Configuration

TargetCall evaluation system:

CPU	AMD EPYC 7742 [70] @2.25GHz, 4-way SMT [71]
Cache-Hierarchy	32×32 KiB L1-I/D, 512 KiB L2, 256 MiB L3
System Memory	4×32GiB RDIMM DDR4 2666 MHz [72] PCIe 4.0 ×128
OS details	Ubuntu 21.04 Hirsute Hippo [73], GNU Compiler Collection (GCC) version 10.3.0 [74]
GPU	NVIDIA TITAN V [75] 5120 CUDA Cores@1.2GHz, 12GiB HBM2 NVIDIA System Management Interface (NVIDIA-SMI) version 510.47.03 [76] NVIDIA CUDA Compiler Driver (NVCC) version 11.1.105 [77]

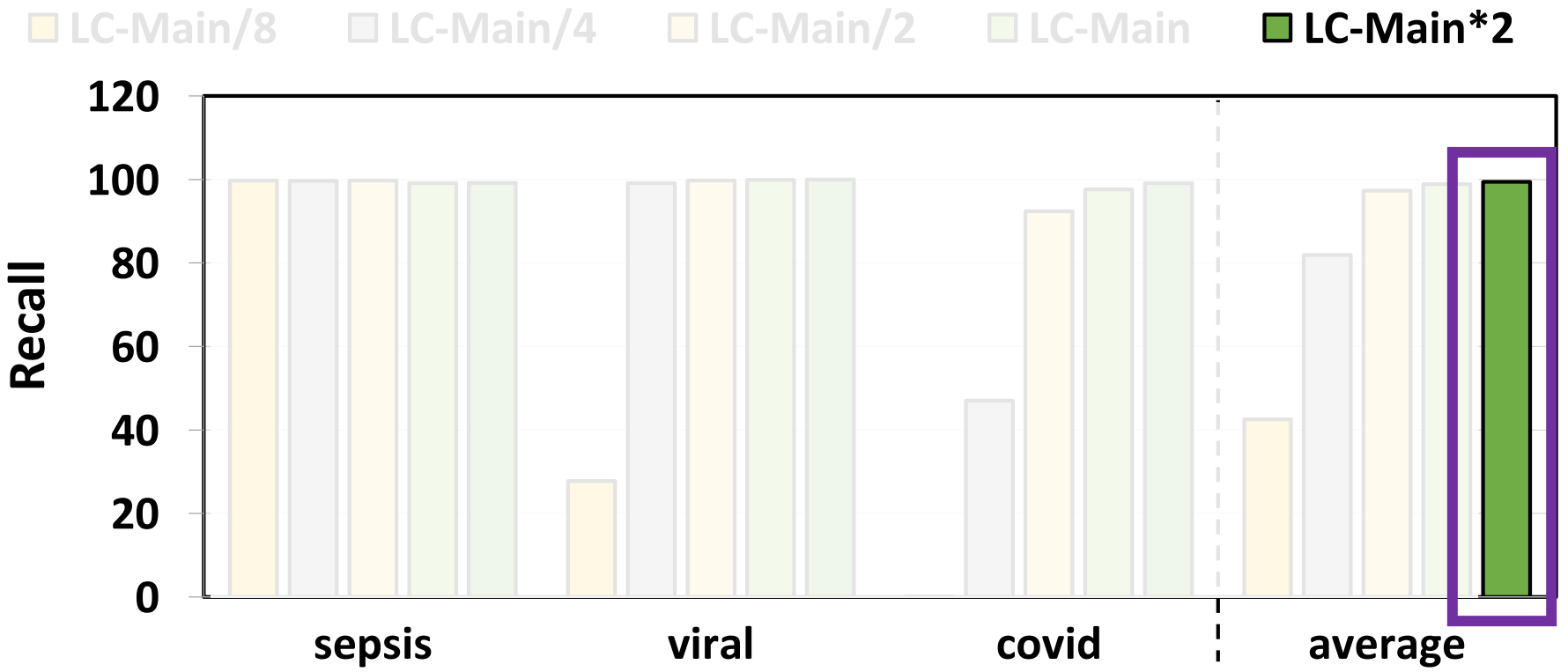
Evaluation – Training Setting

Dataset for training and validation: publicly available ONT dataset sequenced using MinION Flow Cell (R9.4.1)

Optimizer: Adam with

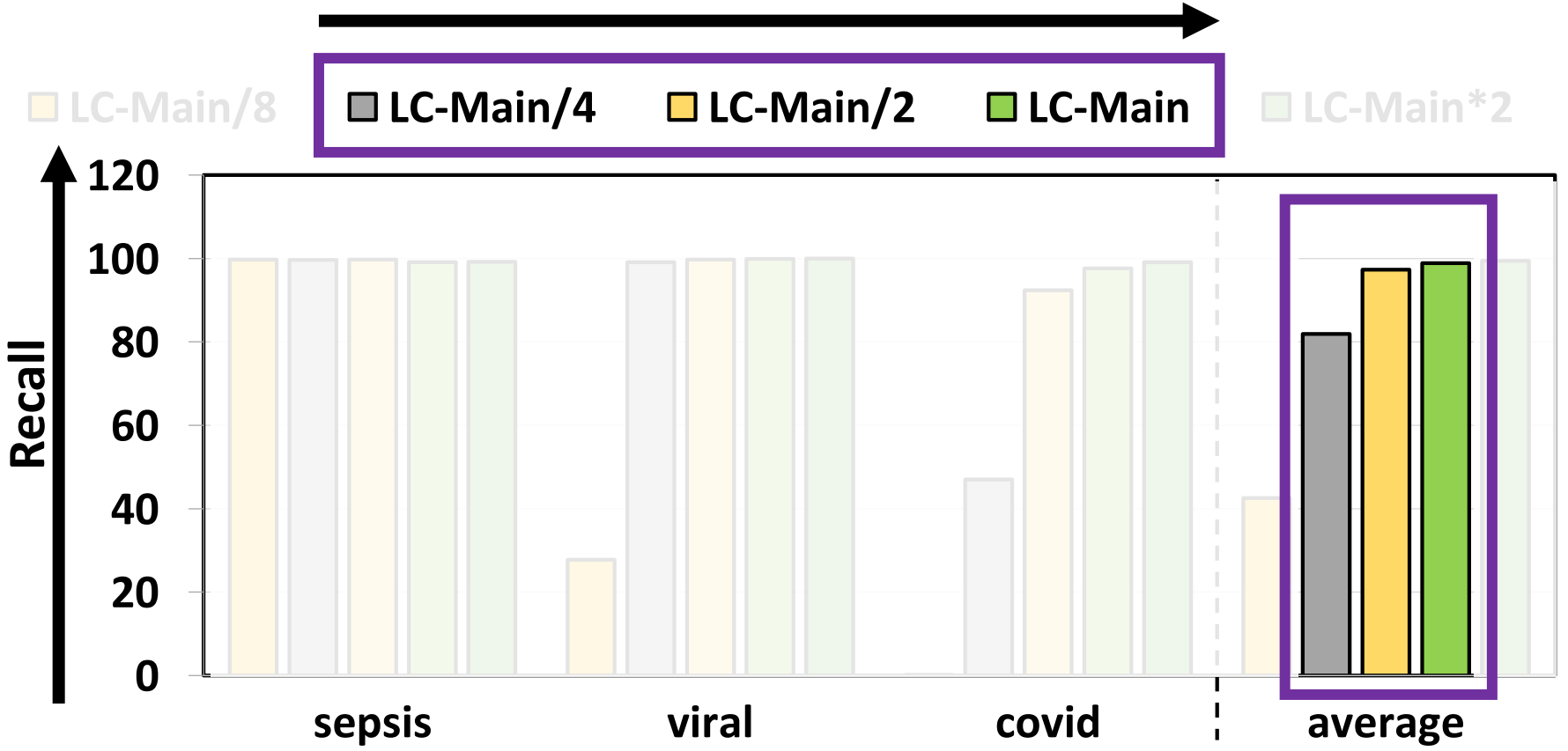
- learning rate: 2e–3
- beta value: 0.999
- weight decay: 0.01
- epsilon:1e–8

TargetCall Design – Recall (1/6)



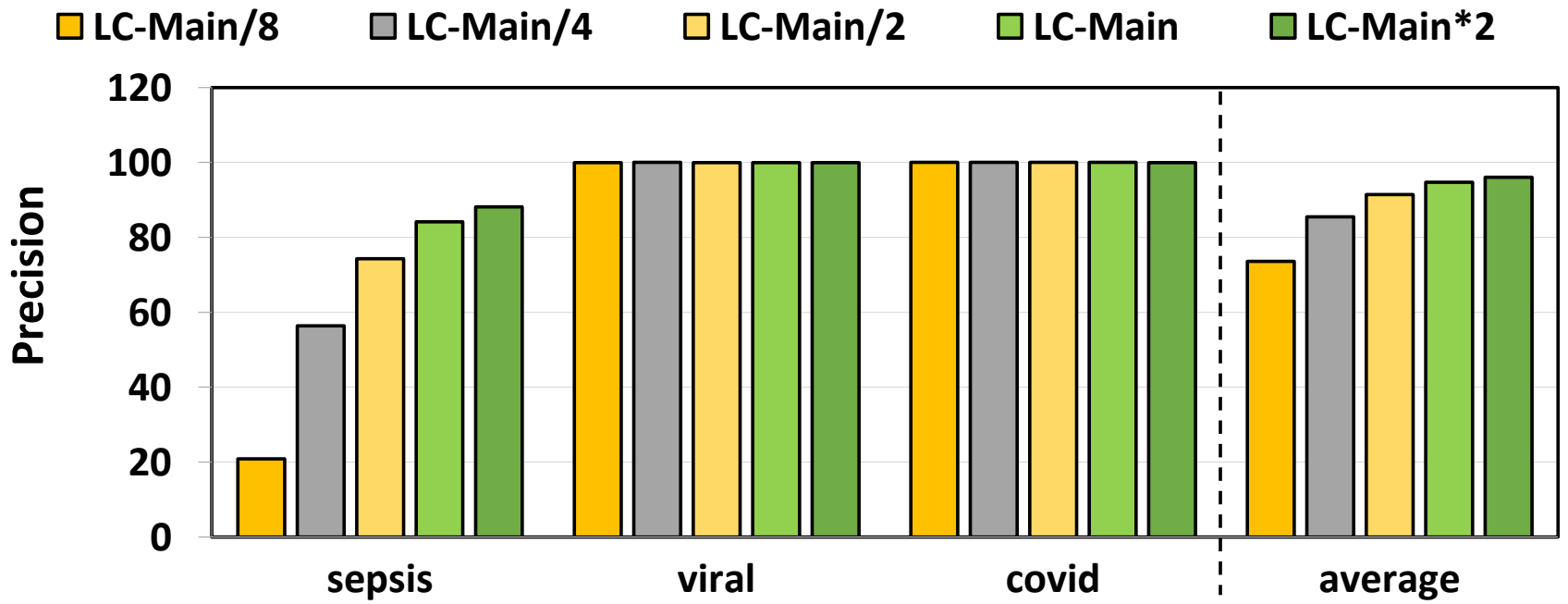
TargetCall provides up to **99.45%** recall.

TargetCall Design – Recall (1/6)



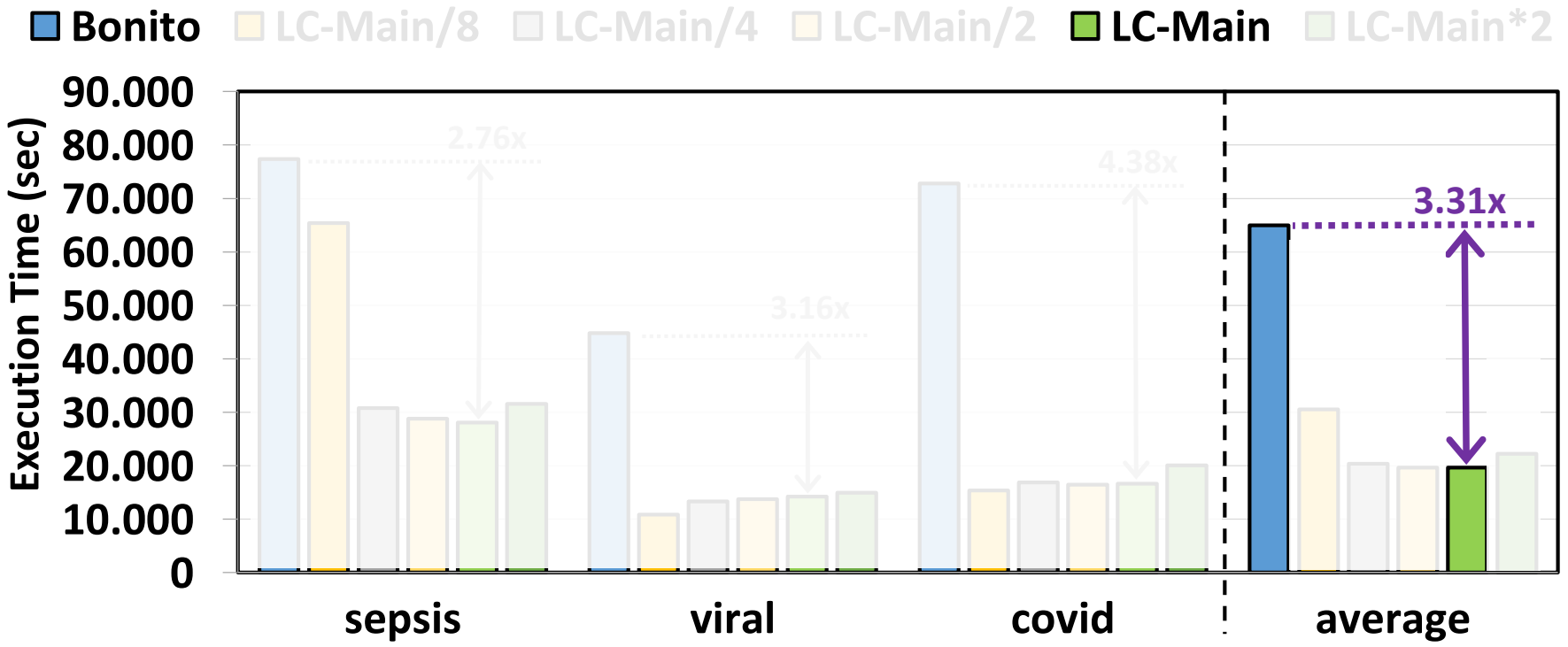
TargetCall's recall **improves** as
the **model complexity of LightCall increases**

TargetCall Design – Precision (2/6)



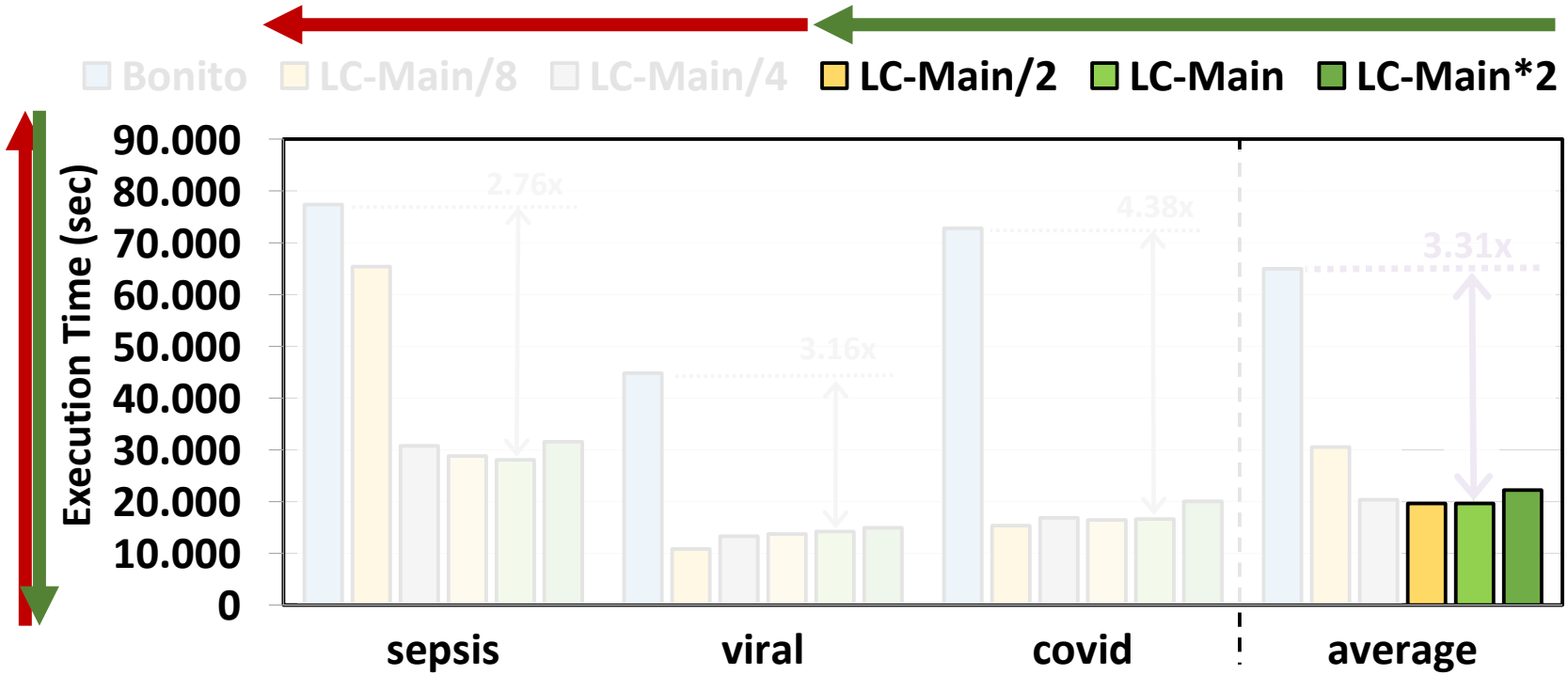
TargetCall can filter out up to **96.03%** of **useless** reads

TargetCall Design – Performance (3/6)



TargetCall provides up to **3.31x** basecalling speedup.

TargetCall Design – Performance (3/6)



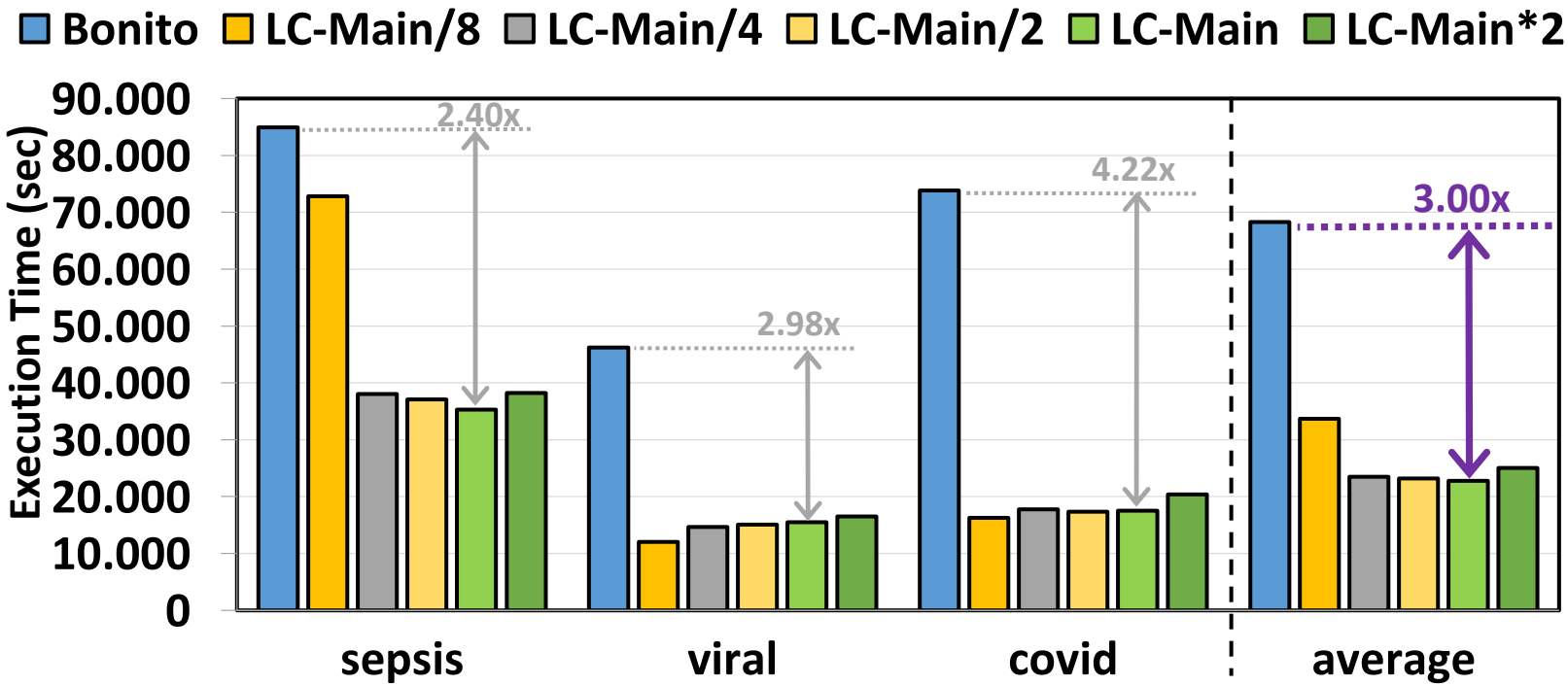
TargetCall's performance **improves** with
decreasing LightCall complexity
until the **filtering precision** is **too low**

TargetCall Design – EtE Accuracy (4/6)

Model Name	Average RA Deviation
$LC_{Main \times 2}$	0.03%
LC_{Main}	0.08%
$LC_{Main/2}$	0.23%
$LC_{Main/4}$	0.91%
$LC_{Main/8}$	72.19%

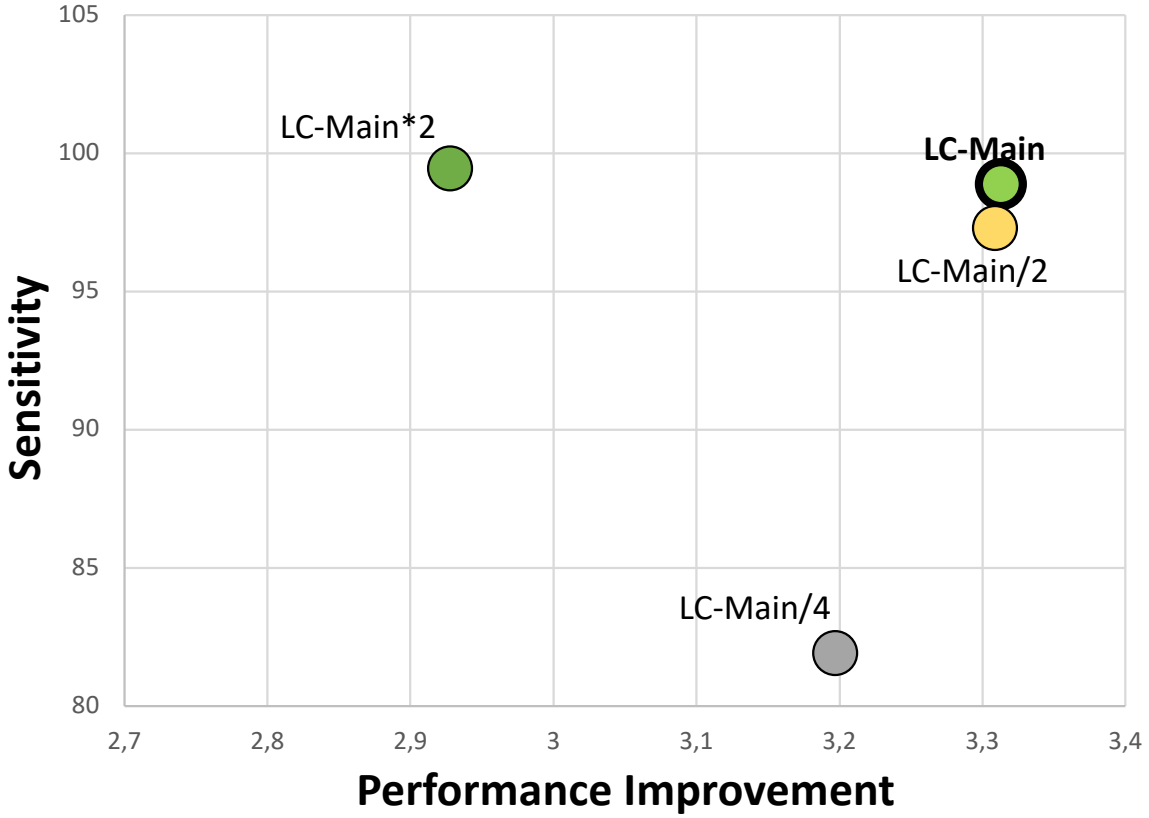
TargetCall affects the relative abundances **slightly**.

TargetCall Design – EtE Performance (5/6)



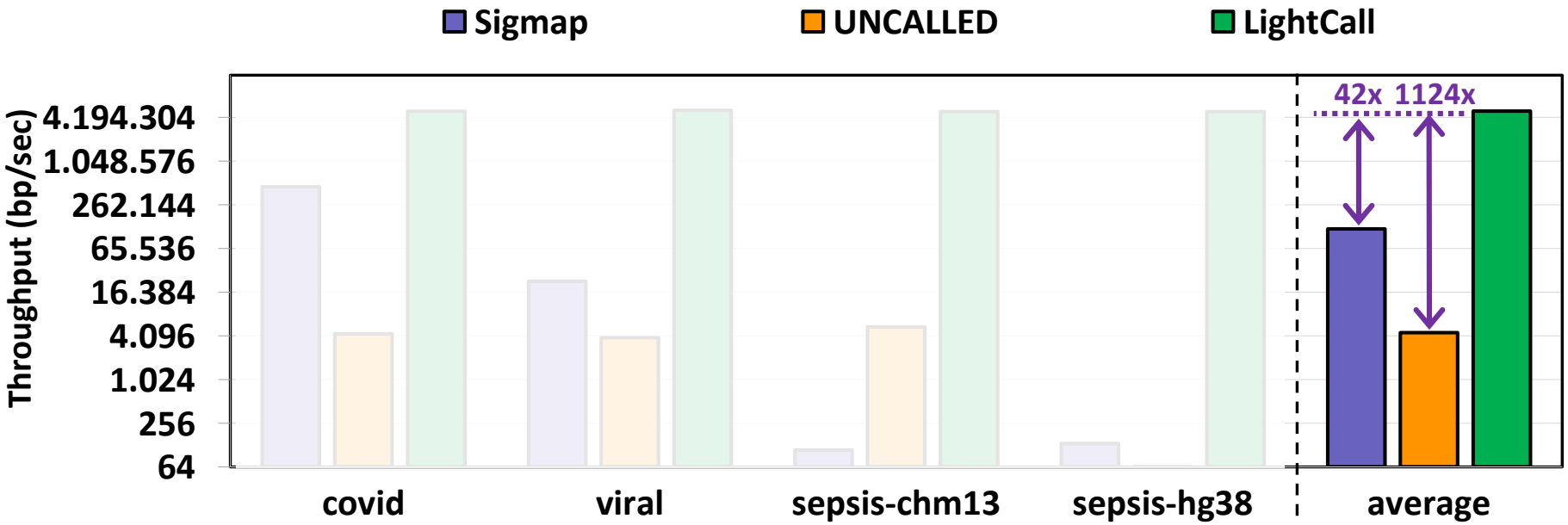
TargetCall provides up to 3x end-to-end speedup over the entire genome analysis pipeline including variant calling

TargetCall Design – Best Model (6/6)



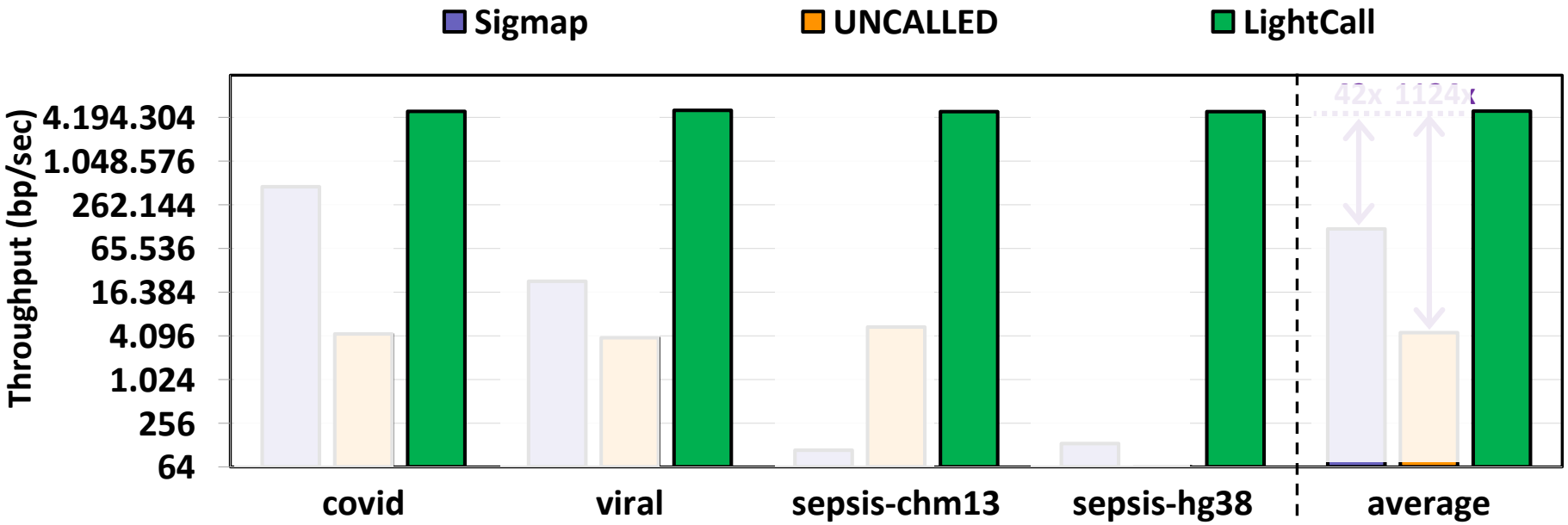
LC-main provides the best recall-performance trade-off

Comparison to SOTA - Throughput (4/4)



LightCall provides **42x**/**1124x** more throughput over Sigmap/ UNCALLED

Comparison to SOTA - Throughput (4/4)



LightCall provides **42x**/**1124x** more throughput over Sigmap/ UNCALLED

LightCall's throughput is **consistently high** for **all reference genome sizes** tested

Comparison to SOTA - Throughput (4/4)

LightCall's high throughput is not reflected to performance:

1. TargetCall processes entire read
2. LightCall and Similarity Check are not pipelined

TargetCall's benefits can be further amplified by

1. Chunk based early filtering
2. Pipelining LightCall and Similarity Check