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

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

1: Problem
- Basecalling consumes 84.2% of total execution time, bottlenecking the genome analysis pipeline
- 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
- Targeted sequencing approaches cannot be applied as general purpose pre-basecalling filters since they have low sensitivity or poor scalability to large target references or lack of adaptability to different applications

2: Our Goal
Eliminate the wasted computation in basecalling while maintaining high accuracy, scalability and adaptability

3: Key Observation & Idea
Key Observation: Typical reason for discarding basecalled reads (i.e., useless reads) is that they do not match some reference genome
Key Idea: Filter out useless reads before basecalling with a highly accurate and high-performance pre-basecalling filter

4: TargetCall
Mechanism: TargetCall consists of two components:
- LightCall: A light-weight basecaller that outputs noisy reads with high performance
- Similarity Check: Computes the similarity of the noisy reads to the reference genome

5: Evaluation Methodology
Baselines:
- Benefits of Pre-Basecalling Filtering: Bonito
- Comparison against Targeted Sequencing: UNCALLED & Sigmapi

Datasets:
- 5 different read sets from various organisms
- 4 different reference genomes with various sizes

Evaluation System:
- LightCall: NVIDIA A100 & TITAN V GPUs
- Similarity Check: AMD EPYC 7742 CPU with 196GB DRAM
- Sigmapi & UNCALLED: AMD EPYC 7742 CPU with 1TB DRAM

TargetCall improves the basecalling execution time by 3.31x by filtering out 94.71% of the useless reads with high accuracy (98.88%) in keeping the useful reads

Full Paper
Source Code

More Results in the Paper
TargetCall:
- Analysis of different LightCall architectures
- Comparison against SOTA:
  - TargetCall’s recall, throughput and peak memory against SOTA

6: Results
6.1: Basecalling Speedup
- TargetCall provides up to 3.31x basecalling speedup on average

6.2: Comparison against SOTA: Precision
- TargetCall provides +62.3% more precision in filtering out useless reads compared to Sigmapi/UNCALLED

6.3: Comparison against SOTA: Performance
- TargetCall provides 9.72x/1.46x better end-to-end basecalling performance over Sigmapi/UNCALLED
- TargetCall provides higher (11.85x/2.04x) speedup over Sigmapi/UNCALLED with a larger reference genome (chm13)