NGS read alignment

Given a read $R$ and a reference genome $G$, NGS read alignment aims to determine $R$'s likely point of origin with respect to $G$.

- Usually exert distance constraint (e.g. edit distance or Hamming distance)
- Usually assume that the true alignment has the optimal alignment score

Seed-and-extend heuristic is the most popular model for NGS read gapped alignment.

- Seed generation
- Extend and refine seeds
- Produce local/global/semi-global alignments around seeds

Gapped alignment on GPUs

Inter-task hybrid CPU-GPU parallelism model

- each CPU thread employs streaming SIMD extension (SSE) instructions
- an instance (i.e. a single process) supports only one GPU, and multi-GPU support can be realized by running multiple instances

Program workflow of CUSHAW2-GPU

- Seed generation
- Top seeds selection
- Final alignments generation

SOLiD read alignment

Hybrid seeding in CUSHAW3

- maximal exact match seeds
- exact k-mer seeds
- variable-length seeds derived from optimal local alignments

Weighed seed pairing heuristic

- $q_i$ is the optimal local alignment score between read $R_i$ (1s/2s) and the mapping region corresponding to a seed
- $m$ is the positive score for an alignment match
- A max-heap data structure is employed with $w$ as the key

CUSHAW Software Package: Harnessing CUDA-enabled GPUs for Next Generation Sequencing Read Alignment

Yongchao Liu and Bertil Schmidt

Abstract

Aligning next generation sequencing (NGS) reads to a genome is often essential to many applications of NGS and has motivated the development of several gapped read aligners as reads become longer. Although usually fast, existing gapped aligners still have difficulties in substantially meeting the requirements of ever-increasing sequence volume on a single multi-core CPU. In this regard, it becomes attractive to resort to accelerators like many-core GPUs.

We present CUSHAW2-GPU to accelerate the CUSHAW2 algorithm using CUDA-enabled GPUs. By aligning both simulated and real reads to the human genome, our aligner yields comparable or better performance compared to BWA-SW, Bowtie2 and GEM. Furthermore, CUSHAW2-GPU with a Tesla K20c GPU achieves significantly speedups over the multi-threaded CUSHAW2, BWA-SW, Bowtie2 and GEM on the 12 cores of a high-end CPU for both single-end and paired-end alignments. In addition, we have presented some features of CUSHAW3, an extension of CUSHAW2 to further improve the alignment quality of base-space reads and offer new support for color-space reads. For color-space alignment, CUSHAW3 is consistently one of the best aligners compared to SHRIMP and BFAST.

Performance Evaluation

CUSHAW2-GPU alignment quality on Illumina-like reads: (a) 100 bp; (b) 150 bp; and (c) 250 bp

References


Contact

Yongchao Liu: liuy@uni-mainz.de
Bertil Schmidt: bertil.schmidt@uni-mainz.de
Institute of Computer Science, University of Mainz
Germany

Alignment quality and runtimes on simulated color-space reads

Dataset | Measure | CUSHAW3 | SHRIMP2 | BFAST
--- | --- | --- | --- | ---
50-bp | Sensitivity | 92.13 | 91.55 | 88.94
Recall | 86.28 | 86.58 | 81.01
Recall$^+$ | 84.72 | 84.22 | 81.01
Time(min) | 41 | 227 | 160
75-bp | Sensitivity | 92.27 | 92.53 | 93.44
Recall | 91.16 | 91.24 | 86.14
Recall$^+$ | 89.33 | 88.15 | 86.14
Time(min) | 20 | 263 | 389

*means the recall is calculated from all reported alignments per read and $^+$ means the recall is calculated from the first alignment occurrence per read.

CUSHAW3 outperforms both SHRIMP2 and BFAST for the 50-bp dataset, while BFAST is the best for the 75-bp dataset, in terms of sensitivity.

CUSHAW3 achieves a speedup of 9.5 (and 11.9) over SHRIMP2 and BFAST on average.

For the 75-bp dataset, CUSHAW3 runs 13.5x and 18.9x faster than SHRIMP2 and BFAST, respectively.