Mercator Research Center Ruhr
Eine Initiative der Stiftung Mercator und der Universitätsallianz Metropole Ruhr

UA RUHR

Massively parallel read mapping on GPUs with the q-group index and PEANUT

Johannes Köster, Sven Rahmann
Genome Informatics, University of Duisburg-Essen

Introduction

We present the **q-group index**, a q-gram index variant with a particularly **small memory footprint**, and efficient parallel algorithms for building and querying the data structure. On top of the q-group index, we developed **PEANUT**, a **massively parallel GPU-based** solution to the read mapping problem. PEANUT is **2 to 10 times faster** than its competitors while maintaining a comparable accuracy. The software is available at `http://peanut.readthedocs.org`.

Read Mapping Problem

Occurs with next-generation sequencing of DNA or RNA:

- millions of small DNA or RNA reads are produced
- information about their origin in the genome is lost
- *read mapping problem:* find the likely origin of each read in a known reference genome
- *optimal solution:* calculating optimal alignments with Smith-Waterman algorithm (infeasible due to quadratic run-time)
- *state of the art:* use index data structures to find anchor points for alignment (BWT/FM-Index, q-gram index)

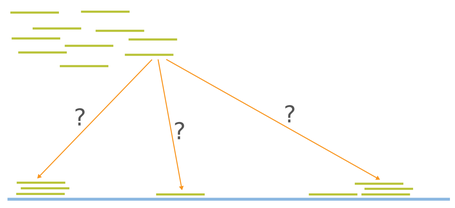

Fig. 1: The read mapping problem.

GPU Architecture

- partitioned into streaming multiprocessors (SM)
- each SM executes the same instruction of 32 threads in parallel
- branching (if-else) breaks parallelism
- memory small (e.g. 3GB) and rather slow (because of small caches)
- memory transfer between host and GPU slow

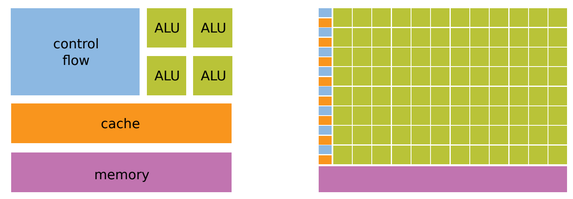

Fig. 2: CPU vs. GPU.

Q-Gram Index

Search for q-gram matches between reads and reference. A q-gram is a subsequence of length q.

- encode q-grams as integers:
  ACGT $= 11\ 10\ 01\ 00 = 228$
- for text T, q-gram index consists of arrays (S, O) such that k-th occurrence position of q-gram g is

$$O[S[g] + k].$$

address array
occurrence array

- size $\mathcal{O}(4^q + |T|)$

exceeds GPU memory

Q-Group Index

We introduce the q-group index, a variant of the q-gram index with small memory footprint.

- assign each q-gram g to a q-group $i = \lfloor g/w \rfloor$
- for text T, q-group index consists of arrays (I, S, S', O) such that k-th occurrence position of q-gram g is

$$O[S'[S[i] + popcount(I[i]\&(2^j - 1))] + k].$$

array of bit vectors
q-group address array
q-gram address array
occurrence array

- build and query in parallel using *population counts* and *prefix sums* (avoiding branching)
- worst case size: $2/w \cdot 4^q + \min\{4^q, |T|\} + |T|$

small memory footprint, almost branch-free queries

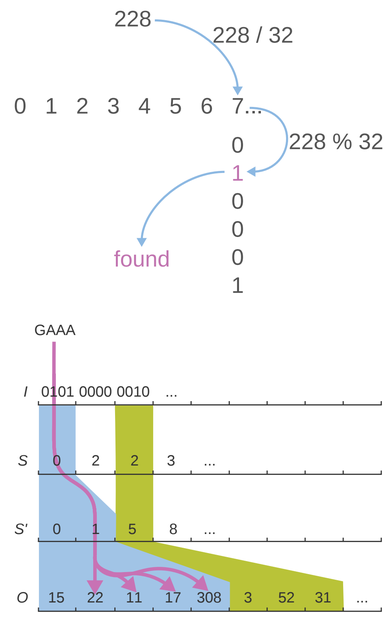

Fig. 3: Q-group index.

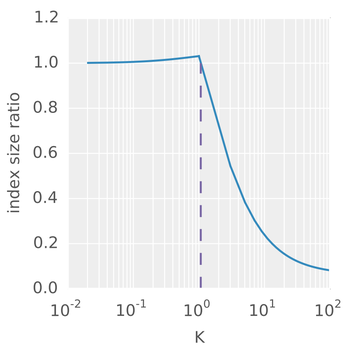

Fig. 4: Ratio between the size of the q-group index and the q-gram index for factor $K = \frac{4^q}{|T|}$ describing ratio between number of possible q-grams and text size.

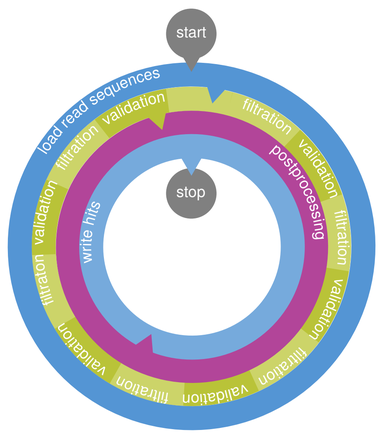

Fig. 5: PEANUT algorithm. Different layers run in parallel and are connected by queues.

PEANUT Algorithm

1. *load* reads into buffer
2. *build* a q-group index
3. *filter* hits between reference and index
4. *validate* hits with bit-parallel alignment algorithm (Myers G, 1999)
5. *post-process* hits
3. *write* hits

● IO ● GPU ● CPU

step 2 to 4: data resides entirely in GPU memory, no transfers needed

Occupancy

- occupancy measures the saturation of GPU cores
- high occupancy: ability to hide memory latency

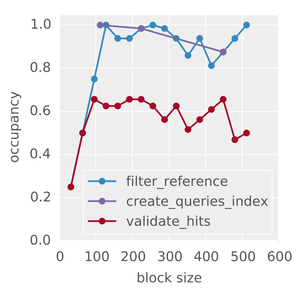

Fig. 6: Occupancy vs. thread block size.

Accuracy

PEANUT provides at least comparably good precision and recall as other mappers.

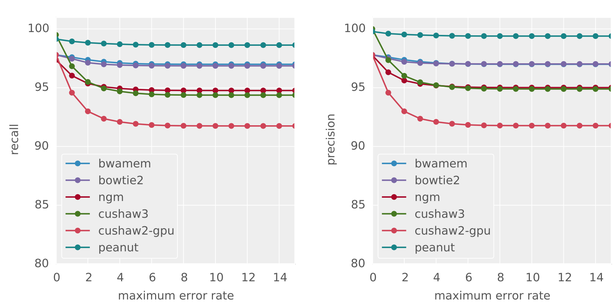

Fig. 7: Recall and precision vs. error rate.

Performance

- 50 million human paired-end Illumina HiSeq 2000 reads
- 4-core Intel Core i7, NVIDIA Geforce 780, 16GB RAM

mapper	mode	time [min:sec]		
PEANUT	best-stratum	18:22	18:36	18:31
BWA-MEM	best-hit	36:46	36:33	36:35
Bowtie 2	best-hit	54:38	54:22	55:51
CUSHAW3	best-hit	390:20	390:15	390:41
CUSHAW2-GPU	best-hit	30:23	30:30	30:34
PEANUT	all-hits	254:43	254:49	254:19
RazerS 3	all-hits	900:27	901:33	900:50

LaTeX TikZposter