Massively parallel read mapping with GPUs

Johannes Köster

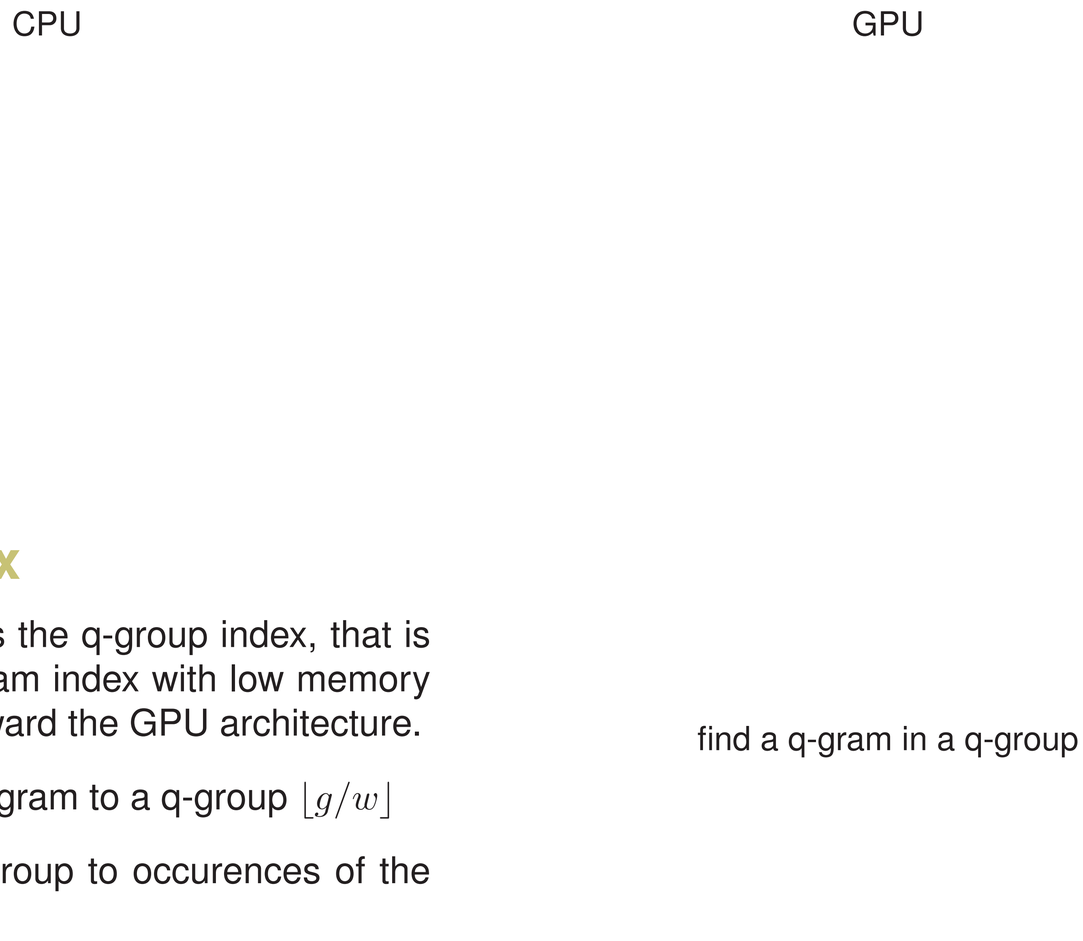
Genome Informatics, Institute of Human Genetics, University Duisburg-Essen
Paediatric Oncology, University Hospital Essen

Contribution

The **read mapping** problem is to find the origin of DNA or RNA sequencing reads – as they are produced by Next-Generation sequencers – in a reference genome. State-of-the art read mappers are fast at providing the best hit for each read. Here we present **PEANUT** (ParallEl AligNment UTility), a novel read mapper that exploits the massive parallelization capabilities of **graphics processors** (GPUs) to provide high sensitivity and a comprehensive enumeration of alternative hits at feasible running times. Thereby, we introduce the **q-group index**, a novel index datastructure, that allows to retreive potential hits between the sequence read and the reference in constant time, while providing a small memory footprint.

The GPU architecture

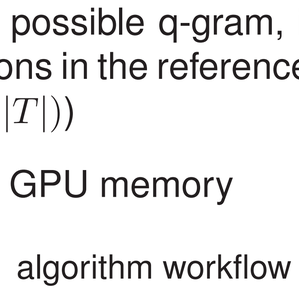
- partitioned into symmetric multiprocessors (SM)
- each SM executes the same instruction of 32 threads in parallel
- branching (if-else) breaks parallelism
- small memory and caches
- memory access slow (because of small caches)



Q-gram index

A common strategy of read mappers is to use a q-gram index to find 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 each possible q-gram, index provides all positions in the reference genome (size $\mathcal{O}(2^{2q} + |T|)$)
- exceeds GPU memory



Q-group index

This work introduces the q-group index, that is a variant of the q-gram index with low memory footprint, tailored toward the GPU architecture.

- assign each q-gram to a q-group $\lfloor g/w \rfloor$
- guide from q-group to occurences of the q-gram
- use population counts and prefix sums (avoiding branching)
- size $\mathcal{O}(2^{2q}/q + 2|T|)$

Algorithm

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

Sensitivity

Sensitivity was benchmarked with Rabema (Holtgrewe et al. 2011) by simulating reads on the *Saccharomyces cerevisiae* genome.

- 100% sensitivity for error rates below 7%
- at least 99.77% sensitivity for error rates up to 10%
- 98.94% sensitivity for error rate $\leq 20\%$

Occupancy

Occupancy measures the saturation of GPU cores. A higher occupancy corresponds to increased capability to hide memory latency.

- high occupancy for index building
- optimal occupancy for filtration
- medium occupancy for validation

Memory footprint

The q-group index allows to use larger q-grams while maintaining a small memory footprint. The following shows the difference in size between q-group index and q-gram index.

- choice of q is a tradeoff between specificity and size
- q-group index helps by reducing the size for larger q