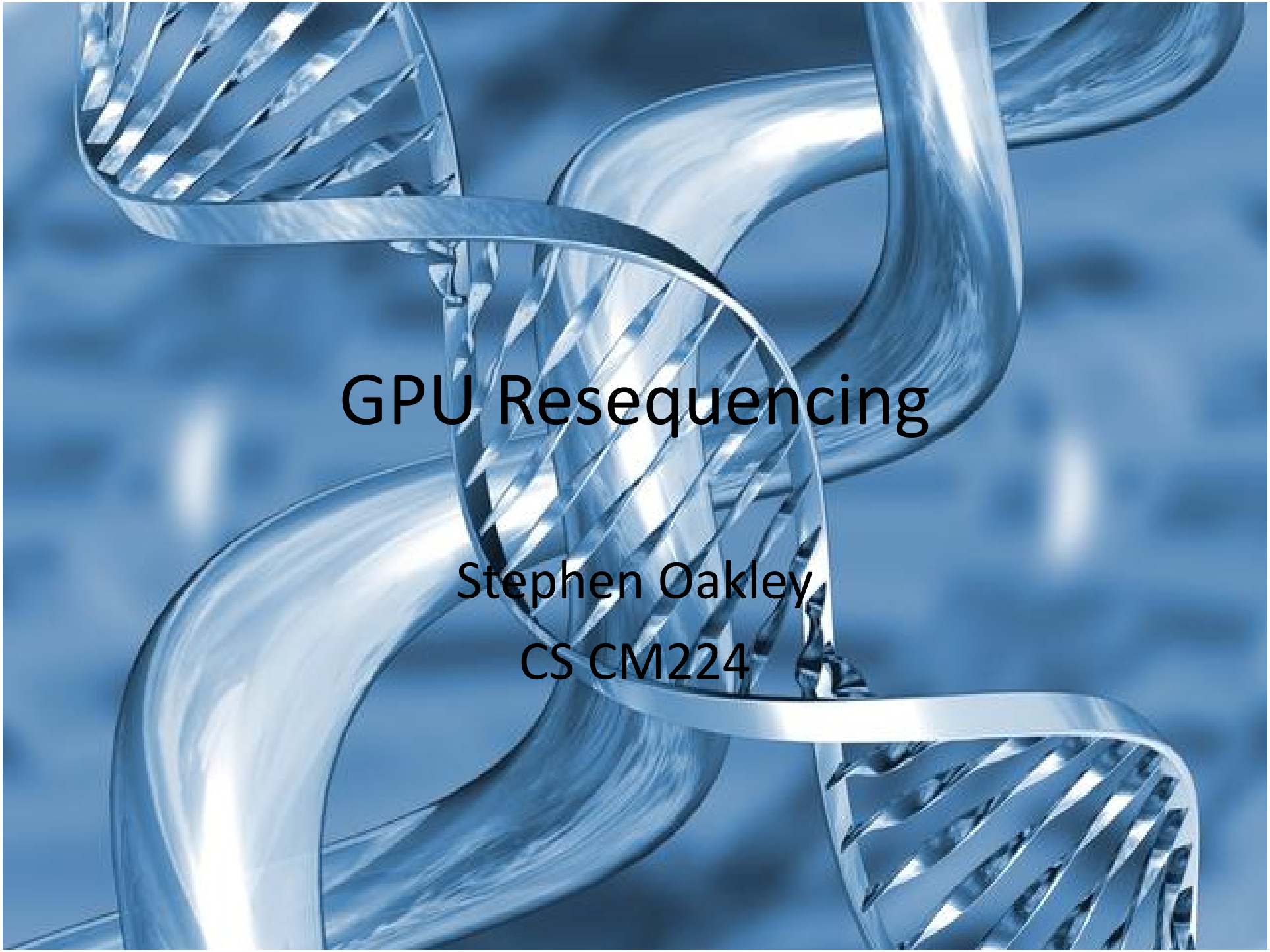


# GPU Resequencing

Stephen Oakley
CS CM224

# Motivation

- The advent of short read sequencing drastically reduces the cost of as compared to past technologies.
- The problem is that it produces random reads from the genome which must be aligned with a reference genome.
- This motivates the idea of resequencing against a reference genome.

# Motivation

- Resequencing works well because humans differ by ~ 0.1%.
- What if we want to quickly find regions of the genome which are similar to other species which vary by ~ 6%
- High throughput resequencing can be the answer.

# Goals

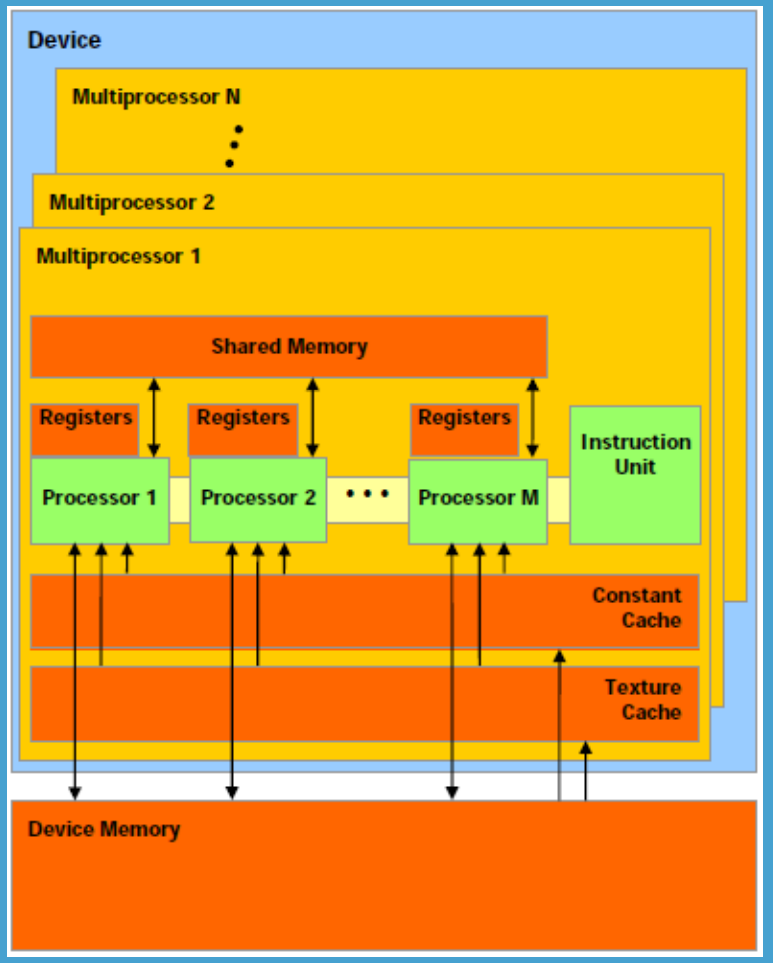
- Produce a high throughput resequencer.
- Leverage the parallelism of current GPU computational models.
  - Thousands of threads
  - Hierarchical memory model
- Develop a “fast” read mapper for long read lengths ( > 50 bp)
- Target high end machines (4+ GB RAM & GPUs)

# Assumptions

- The reference and sample genomes are very similar (currently < 5% variation)
- Any given read contains less than d SNPs
- A given read may map to more than one location in the reference genome (copy variations)

# GPU Overview

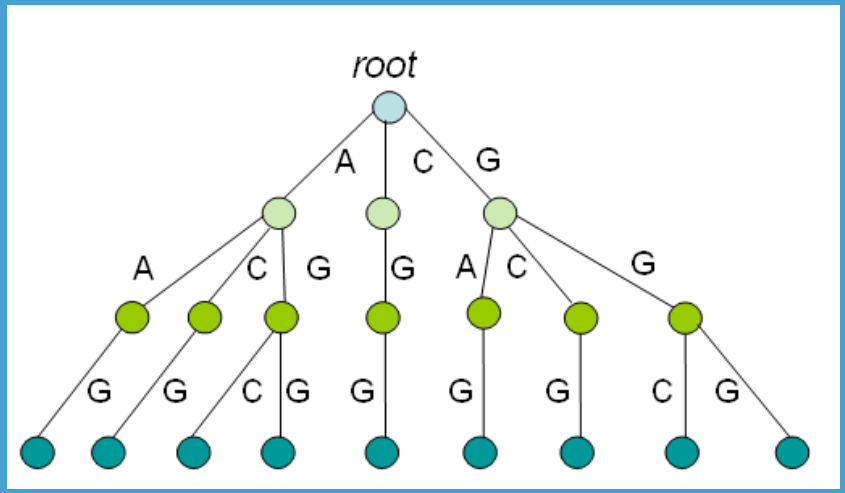
- NVIDIA GeForce 8800 GT
  - 112 cores, 14 multiprocessors, 512 threads per block
  - 512 MB DRAM, 16KB Shared Memory
- NVIDIA Tesla C1060
  - 240 cores
  - 4 GB DRAM



# Approach

- View the problem similar to virus scanning
  - Instead of indexing the reference, build a prefix tree over the set of reads
  - Stream through the reference and find all mappable reads for a given section of the genome.
- Prefix tree may be too large to reside on GPU

# Prefix Trees




**Number of Candidates per Prefix**

**Prefix Length**

| Base Pairs | 6 | 8 | 10 | 12 |
|---|---|---|---|---|
| 1.00E+05 | 25 | 2 | 1 | 1 |
| 1.00E+06 | 245 | 16 | 1 | 1 |
| 5.00E+08 | 122071 | 7630 | 477 | 30 |
| 1.00E+09 | 244141 | 15259 | 954 | 60 |
| 3.00E+09 | 732422 | 45777 | 2862 | 179 |

# Details

**CPU**

- Create Prefix Tree
- For every position in the reference genome
  - Find candidates
  - Send comparison task off to gpu
  - Collect results
- Maintain locality of candidate reads on the gpu

**GPU**

- Compare all candidates reads for a given block of the reference genome
- Report matches that meet the threshold criterion to the CPU

# When does this work?

- Performs better when…
  - Reads are long enough that the probability that the SNPs do not appear in the prefix is low.
  - Most reads map perfectly (reasonable if variations are equally distributed throughout the genome)