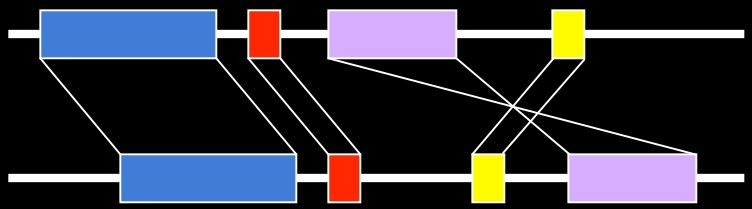
High-throughput sequence alignment using Graphics Processing Units

Michael C. Schatz and Cole Trapnell

September 20, 2007
CBCB Seminar

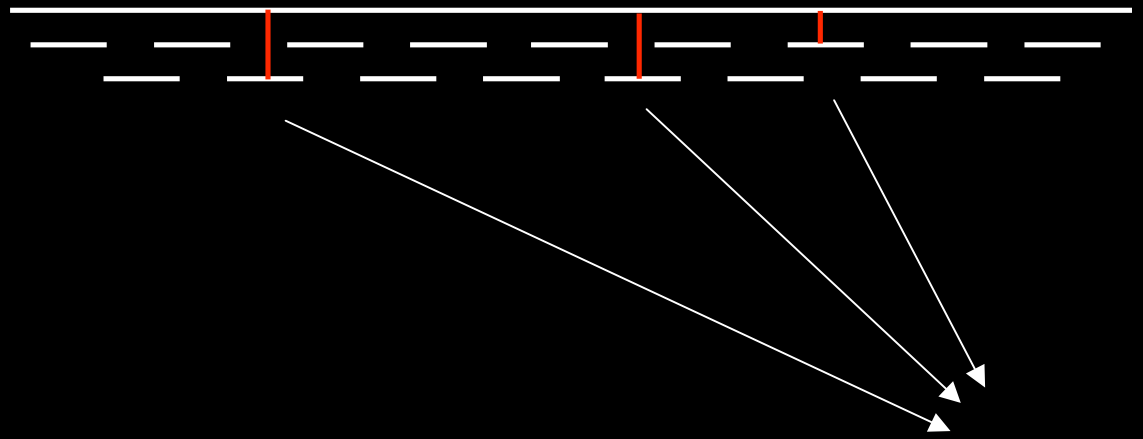
Sequence Alignment Applications

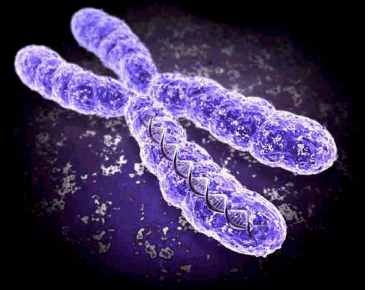
- A very common problem in computational biology is to find all occurrences (or approximate occurrences) of one sequence in another sequence
  - Genome Assembly
  - Gene Finding
  - Comparative Genomics
  - Functional analysis of proteins
  - Motif discovery
  - SNP analysis
  - Phylogenetic analysis
  - Primer Design
  - Personal Genomics
  - …

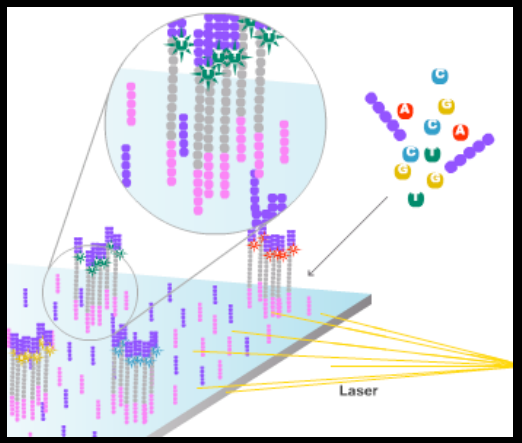


Personal Genomics

- How does your genome compare to Craig's?





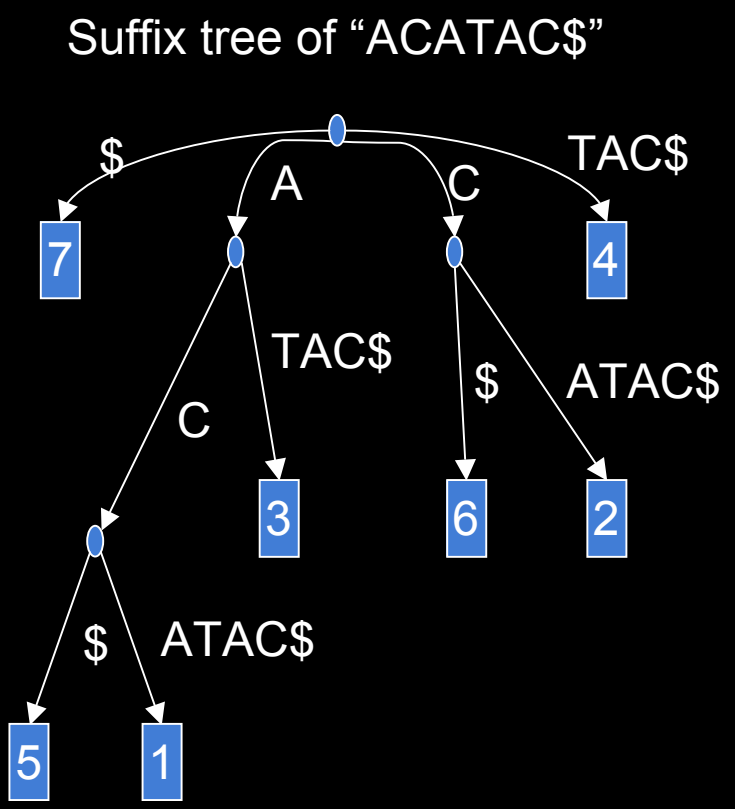


Heart Disease

Cancer

God Gene

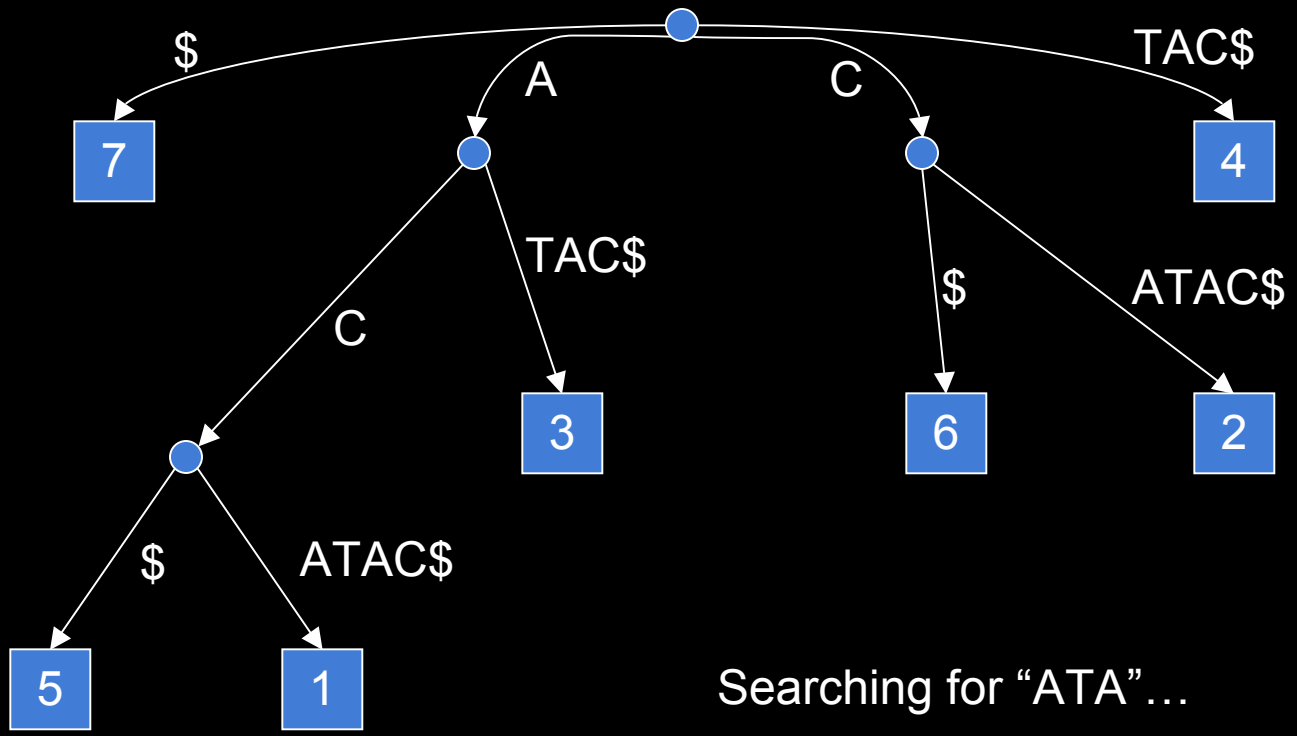
Suffix Trees to the Rescue



- Tree of all suffixes of string S
  - Suffix i encoded on path to leaf i
  - Nodes: positions where suffixes diverge
  - Edges: substrings of S
  - Leaves: starting position of suffix
  - Suffix Links: traverse to next suffix
- O(n) Construction
  - Ukkonen’s Algorithm
  - Exploits inter-suffix relationships and suffix links
- O(k) Substring Match
  - Every substring S[i,j] is a prefix of suffix i.
  - Walk from root following the characters in the query Q.
  - One leaf for each occurrence of Q in T.

*858E Algorithms for Biosequence Analysis

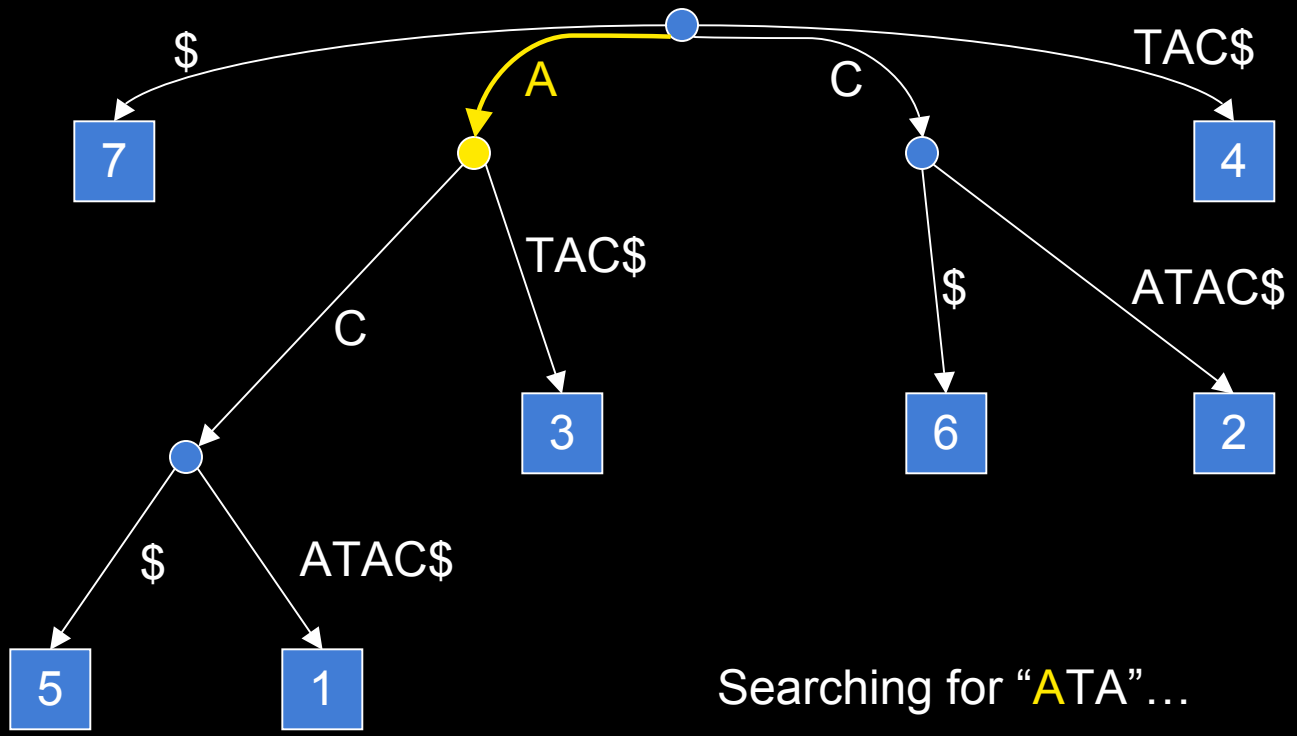
Suffix Tree Search



Searching for “ATA”…

Suffix tree of “ACATAC$”

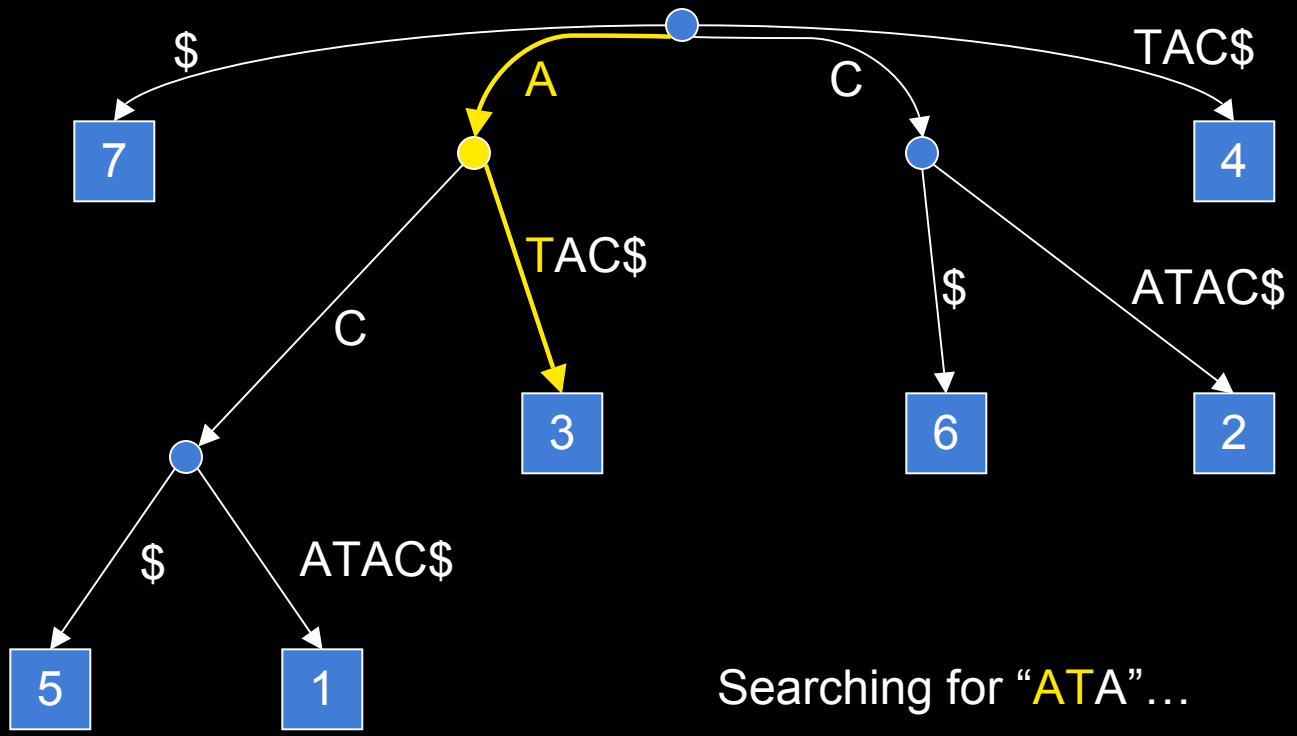
Suffix Tree Search



Searching for “ATA”…

Suffix tree of “ACATAC$”

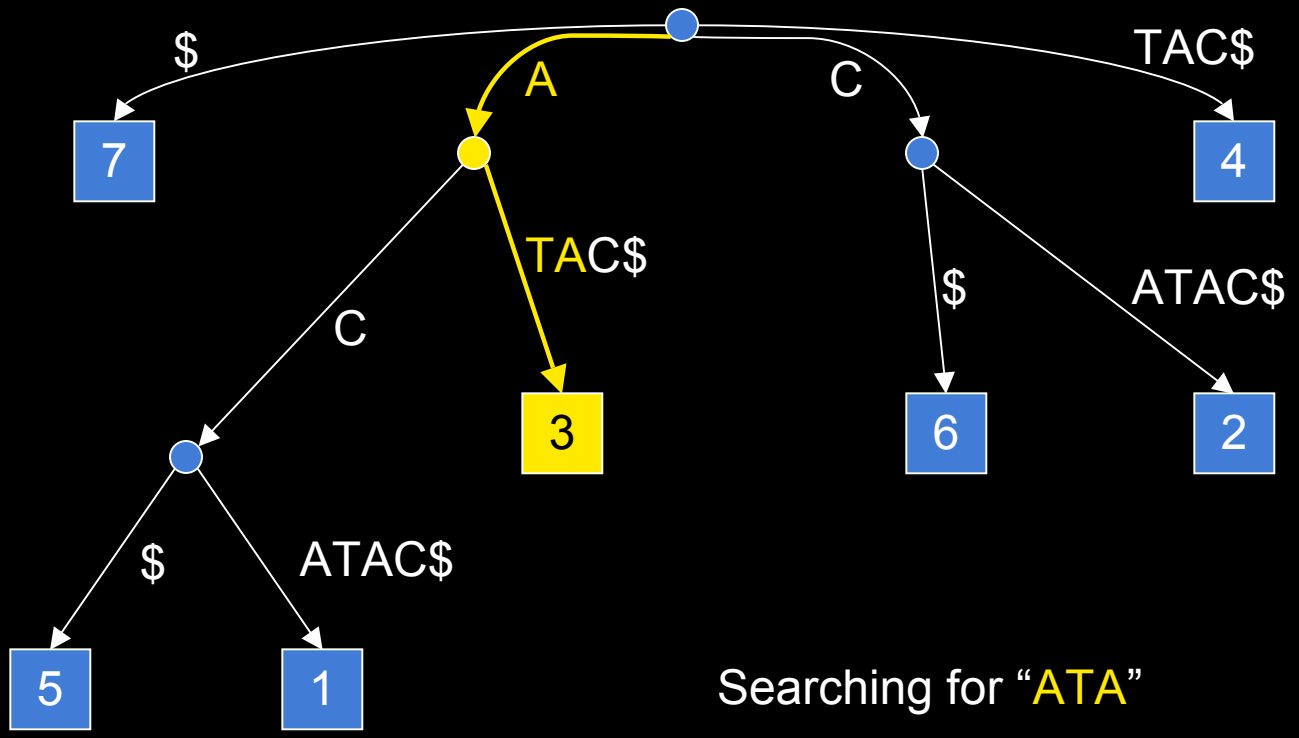
Suffix Tree Search



Searching for “ATA”…

Suffix tree of “ACATAC$”

Suffix Tree Search

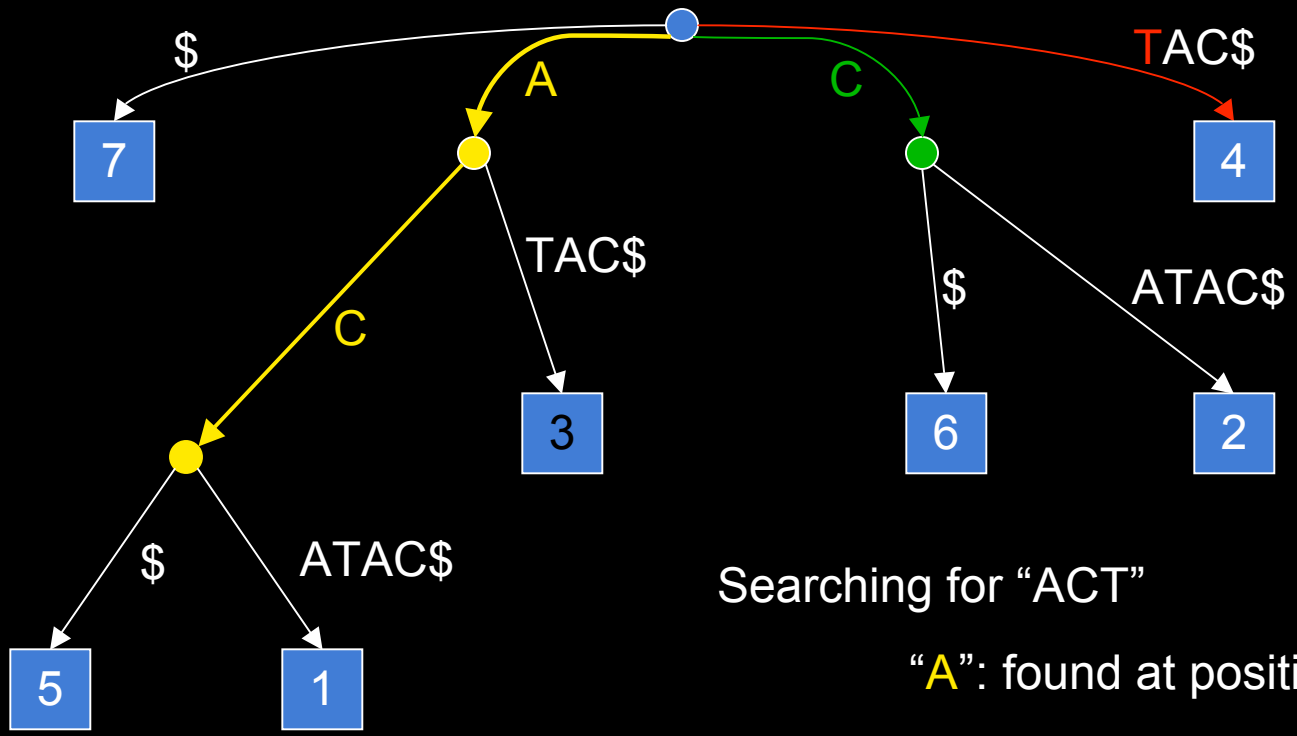


Searching for “ATA”

found at position 3!

Suffix tree of “ACATAC$”

Suffix Tree Search



Searching for “ACT”

“A”: found at positions 1, 3, & 5

“AC”: found at positions 1 & 5

“ACT”: falls off tree => Not in S

Can check next suffix without returning to root.

“C”: found at 2 & 6

“CT”: Not in S

“T”: Found at 4

MUMmer

- Widely used alignment program, developed for aligning whole genomes to each other.
  - Post-process exact alignment to seed longer inexact matches
- Uses MUMs as heuristic to filter less interesting results
  - Generally need to use –maxmatch for read alignment

1. Construct suffix tree S of human genome
2. For each read R
   1. Align R to S
   2. Output alignments

This is performed sequentially but is embarrassingly parallel.

Graphics Processing Units

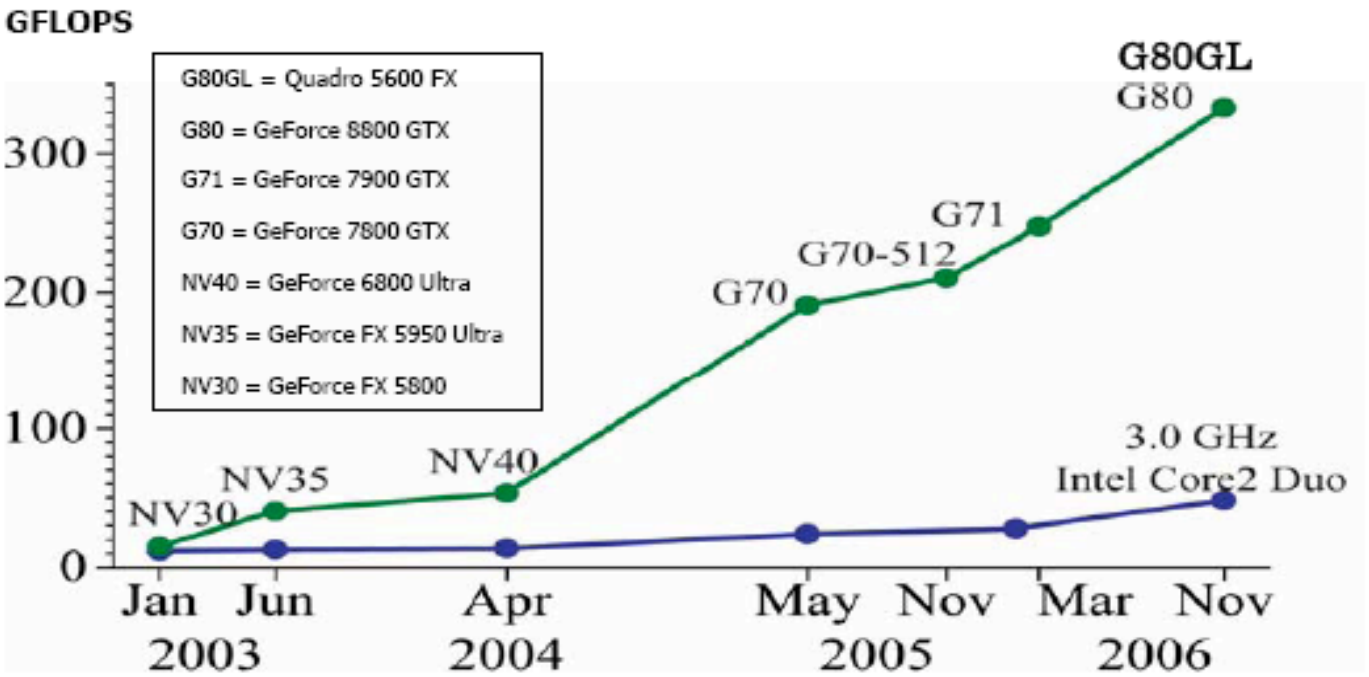
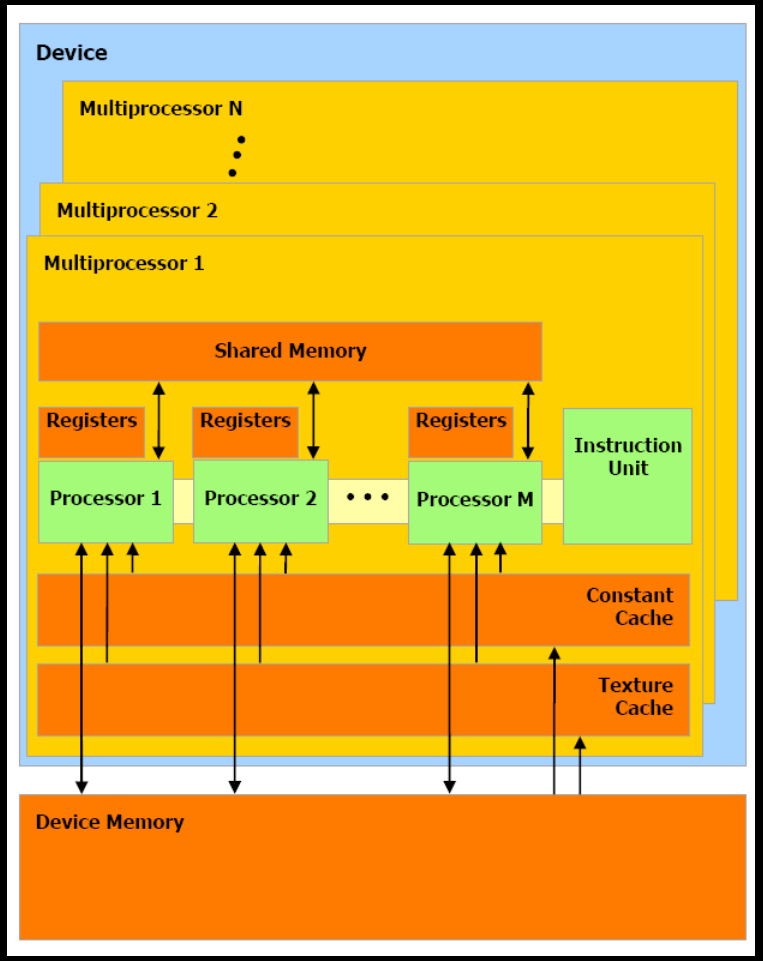


Figure 1-1. Floating-Point Operations per Second for the CPU and GPU

The processing power of highly parallel GPUs is growing faster than CPUs.

GPGPU Programming



- Utilize the highly parallel SPMD architecture of the GPU
  - Nominally used for in parallel triangle rendering, texture application
  - Each processor executes same kernel
  - Dramatic runtime improvement for scientific applications
- CUDA Architecture
  - API and runtime library to implement C style programming of stream processors
- nVidia GeForce 8800 GTX (G80)
  - 16 multiprocessors w/ 8 processors
    - 128 stream processors @ 1.35 GHz
  - 768 MB total on board RAM

*Image from CUDA Programming Guide

Host Programming

```
float* Bd;
size = wA * wB * sizeof(float);
cudaMalloc((void**)&Bd, size);
cudaMemcpy(Bd, B, size, cudaMemcpyHostToDevice);

// Allocate C on the device
float* Cd;
size = hA * wB * sizeof(float);
cudaMalloc((void**)&Cd, size);

// Compute the execution configuration assuming
// the matrix dimensions are multiples of BLOCK_SIZE
dim3 dimBlock(BLOCK_SIZE, BLOCK_SIZE);
dim3 dimGrid(wB / dimBlock.x, hA / dimBlock.y);

// Launch the device computation
Muld<<<dimGrid, dimBlock>>>(Ad, Bd, wA, wB, Cd);

// Read C from the device
cudaMemcpy(C, Cd, size, cudaMemcpyDeviceToHost);
```

Allocate and copy data to GPU

Allocate space for results

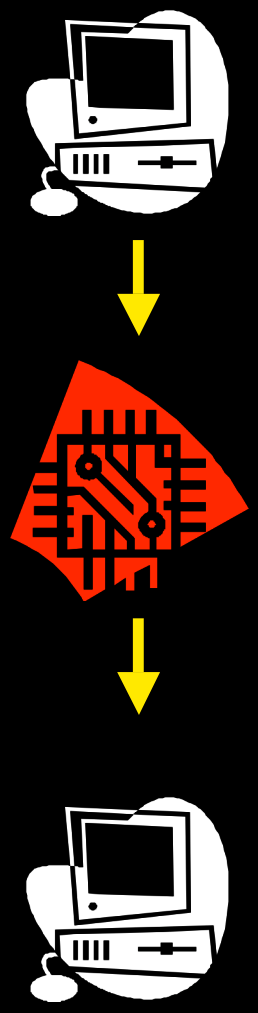
Execute Kernel

Read Results

Kernel Programming

- Restricted form of C
  - Loops & conditions allowed
  - No recursive calls, no stack
  - All storage must be pre-allocated from host
  - Very fast numerical functions: sin(), sqrt(), log()
  - Limited number of registers
- texfetch() to read memory from memory texture.
  - Uses hardware accelerated 2D cache for read-only memory
  - Non-cached reads and writes have high latency
- Threads execute independently
  - Synchronization primitives and atomic functions available
  - Small per-multiprocessor shared memory also available.

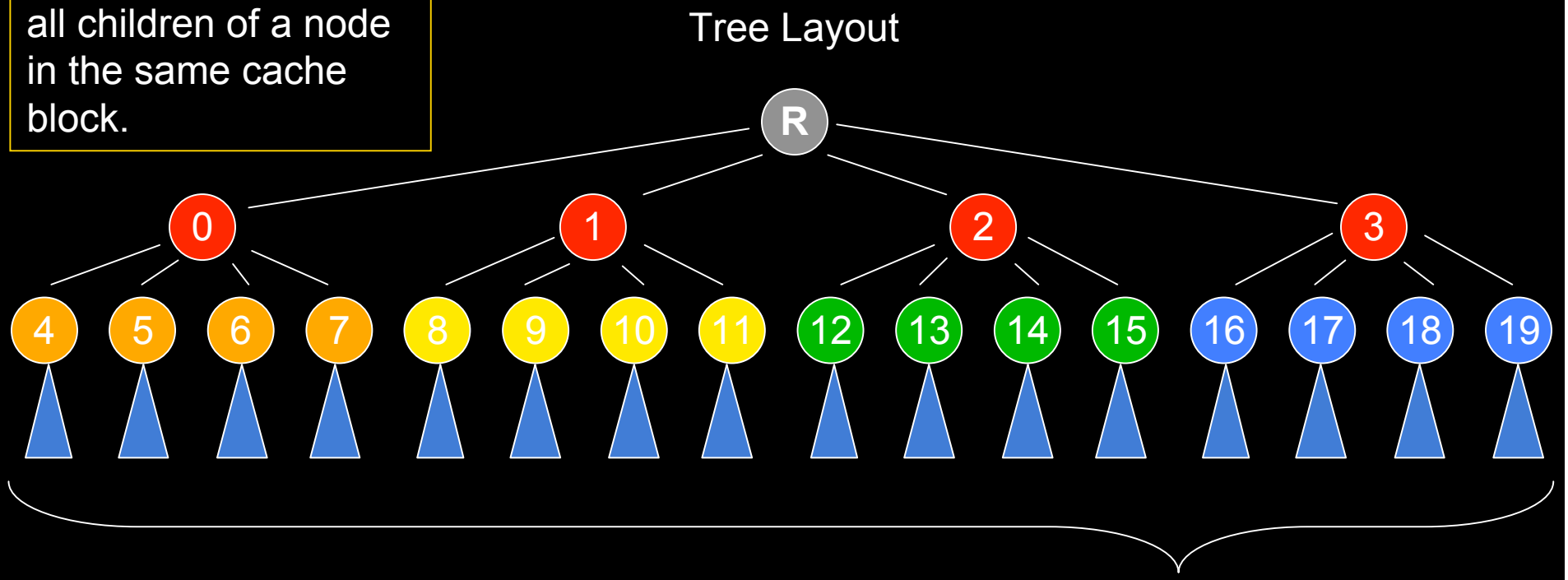
MUMmerGPU Algorithm



1. Load Reference String
2. Create Suffix Tree
3. Reorder Tree Layout
4. Load Query Strings
5. Transfer data to GPU
6. Execute Query Kernel
   - Up to 128 simultaneous matches on GPU
7. Fetch Results from GPU
8. Output results

Suffix Tree Reordering

Near the root, place all children of a node in the same cache block.



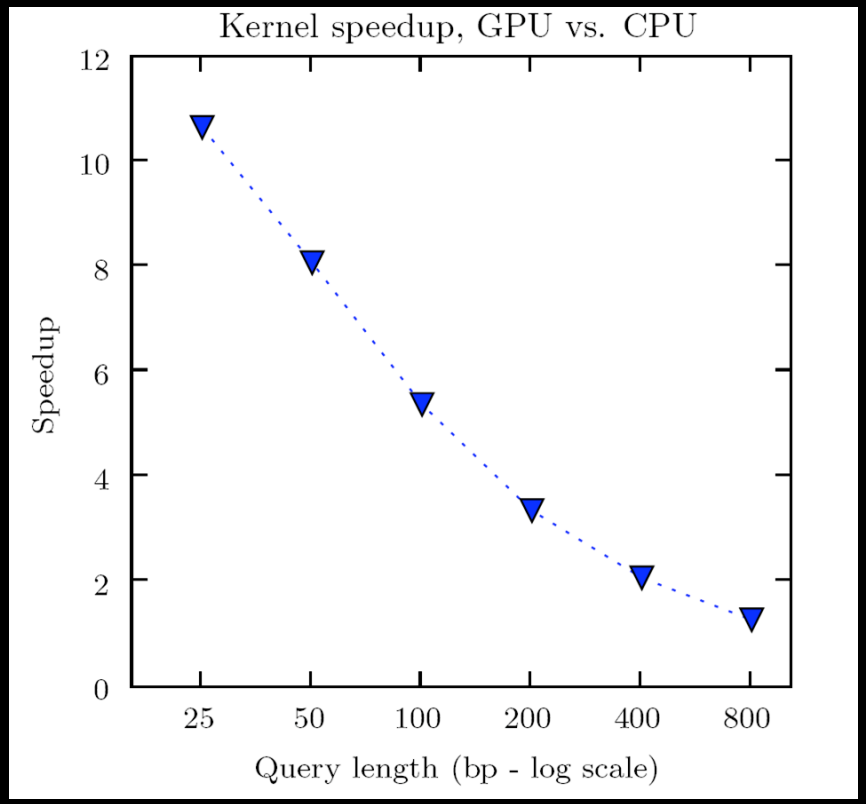
Cache Layout

0	2	4	6	8	10	12	14
1	3	5	7	9	11	13	15

Further down, place node and children in same cache block

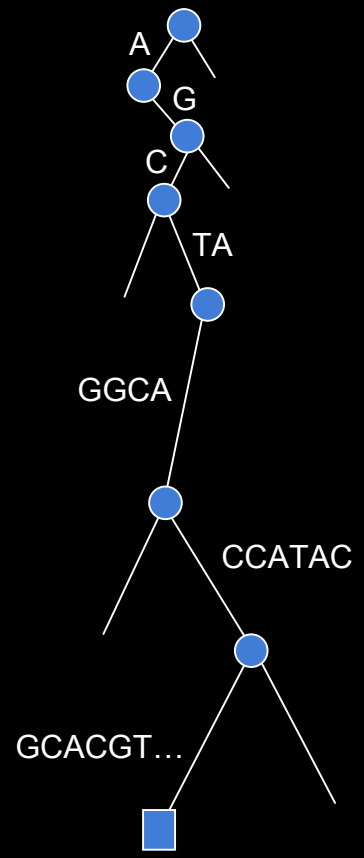
Synthetic Reads Results

- Aligned 50-, 100-, 200-, 400-, and 800-bp synthetically constructed reads to the *Bacillus anthracis* genome.
- Explore MUMmerGPU's performance in the absence of errors and over a wide variety of query lengths.
- Each test set contained exactly 250Mbp of query sequence divided evenly among all the reads in the set.



Long Read Slowdown

- Kernel walks down edges of tree until end of query or mismatch
  - Different edges may be different lengths
  - Typically short edges near root, long edges further down
- Thread Divergence
  - All threads on same multiprocessor must wait to reach end of longest tree edge
- Cache Performance
  - Longer reads will explore further into tree
  - Less opportunities for locality



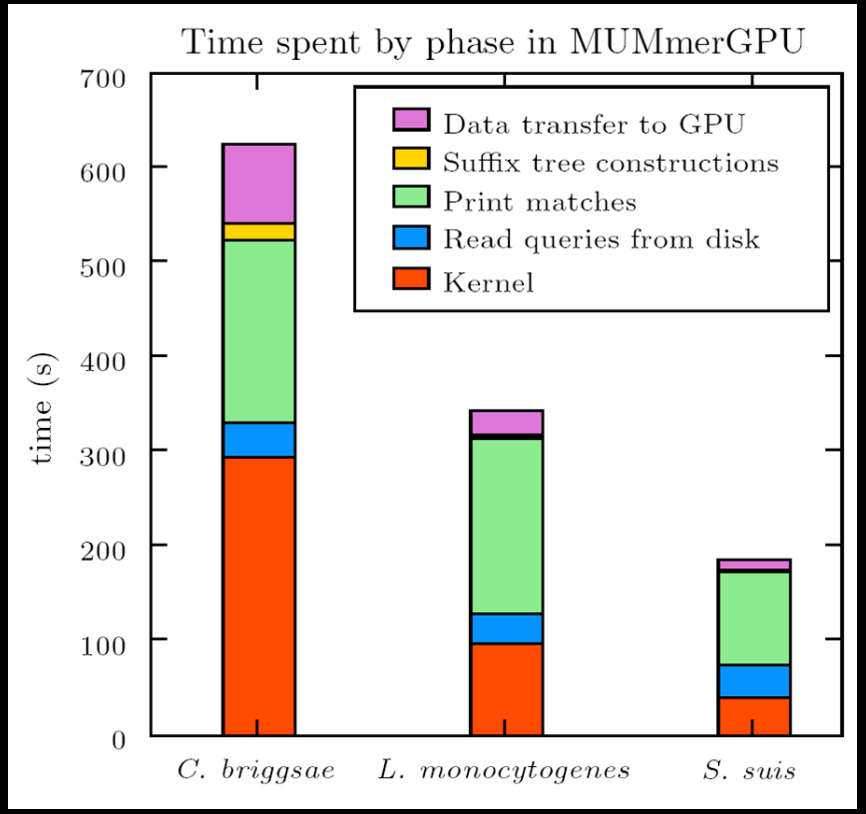
Genuine Reads Results

Reference	Reference Length (bp)	# of queries	Query length mean ± stdev	Min alignment length (l)	# of suffix trees (k)	Speedup
Caenorhabditis briggsae Sanger sequencing	13,163,117	2,357,666	717.84 ± 159.44	100	2	3.71
Listeria monocytogenes 454 pyrosequencing	2,944,528	6,620,471	200.54 ± 60.51	20	1	3.79
Streptococcus suis Illumina/Solexa sequencing	2,007,491	26,592,500	35.96 ± 0.27	20	1	3.47

- Aligned the reads against both strands of the chromosomal DNA for *L. monocytogenes* and *S. suis*, and against both strands of chromosome III of *C. briggsae*.
- Compare the end-to-end wall clock running time of MUMmerGPU versus MUMmer.

Genuine Reads Results

- Suffix tree construction is only a small fraction of total running time.
- MUMmerGPU execution time now dominated by serial IO.
- MUMmerGPU is within 2x of optimal speedup without parallelizing/compressing IO.



Conclusions

- We have reduced the computation processing time for short read resequencing & personal genomics from hours to minutes.
  - Make sure you have sufficient cooling available
- Low arithmetic intensity GPGPU programs can have dramatic performance improvements (10x) over CPU execution
  - Utilizing the texture cache with careful node placement and minimizing register use were essential to high performance
- A single GPU can supply same processing power as a small computer cluster at a fraction of the cost
  - Installing GPUs into an existing cluster can provide an order of magnitude increase in computing capacity.
- More information:
  - http://mummergpu.souceforge.net

Acknowledgements

Art Delcher

Amitabh Varshney

Steven Salzberg

Mihai Pop

UMIACS Staff

CBCB
Center for Bioinformatics & Computational Biology

UMIACS
University of Maryland Institute for Advanced Computer Studies

NVIDIA