Swift: A GPU-based Smith-Waterman Sequence Alignment Program

Pankaj Gupta
Bioinformatics Application Developer
St. Jude Children's Research Hospital

GPU Technology Conference 2012
05/15/2012

Agenda

- Sequence alignment
- Existing GPU-based aligners
- Swift: a new GPU-based aligner
- Method
- Results
- Problems faced
- Conclusion
- Future work

Source: Wikipedia

Sequence alignment

- Sequence alignment is an important component of bioinformatics
- New sequences are aligned to known sequences to
  - Find location of the new sequence
  - Find variants, insertions, and deletions
  - Infer functional, structural and evolutionary relationship
- Example

```
Position        0     5     10    15    20
Reference       CCTTAACGTCCGTTACGTAATCG
                     ||.|||.|||.|||
Read                 ACTTCCTTTAGGTA
```

Sequence alignment programs

- Existing programs
  - BWA
  - BFAST
  - Mosaik
  - BLAST
  - Etc.
- Problem with existing programs is that they are **slow, less accurate, and/or require large memory**
- Expensive hardware is required to run these programs
- Cheaper hardware is more desirable
- GPUs are a good alternative

GPU-based sequence aligners

NVIDIA

Search NVIDIA | USA - United States

DOWNLOAD DRIVERS | COOL STUFF | SHOP | PRODUCTS | TECHNOLOGIES | COMMUNITIES | SUPPORT

TESLA

NVIDIA Home > Products > High Performance Computing > Tesla Bio Workbench

Share this page

APPLICATIONS

- ACEMD
- AMBER
- CUDA-BLASTP
- CUDA-EC
- CUDA-MEME
- CUDASW++ (Smith-Waterman)
- DNADist
- GPU Blast
- GPU-HMMER
- HOOMD
- LAMMPS
- MUMmerGPU
- MUMmerGPU++
- NAMD
- SeqNFind
- TeraChem
- UGENE
- VMD

GPU SOLUTIONS

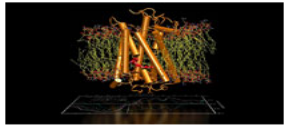
Tesla GPU Computing Overview
- Workstations
- Data Centers

RESOURCES FOR GPU COMPUTING

Tesla Bio Workbench - Enabling New Science

The NVIDIA® Tesla™ Bio Workbench enables biophysicists and computational chemists to push the boundaries of biochemical research. It turns a standard PC into a "computational laboratory" capable of running complex bioscience codes, in fields such as drug discovery and DNA sequencing, more than 10-20 times faster through the use of NVIDIA Tesla GPUs.

GPU TEST DRIVE



AMBER and NAMD 5x Faster

Run AMBER and NAMD 5x faster. Try it now on a free, remotely-hosted cluster and see how easy it is to reduce simulation time from days to hours.

Take a Free and Easy Test Drive Today

APPLICATIONS

Molecular Dynamics & Quantum Chemistry
- ACEMD
- AMBER
- BigDFT (ABINIT) (news)
- GROMACS
- HOOMD
- LAMMPS
- NAMD
- TeraChem (Quantum Chemistry)
- VMD

Bio Informatics
- CUDA-BLASTP
- CUDA-EC
- CUDA-MEME
- CUDASW++ (Smith-Waterman)
- DNADist
- GPU Blast
- GPU-HMMER
- HEX Protein Docking
- Jacket (MATLAB Plugin)
- MUMmerGPU
- MUMmerGPU++
- SARUMAN
- SeqNFind
- UGENE

Complex molecular simulations that had been only possible using supercomputing resources can now be run on an individual workstation, optimizing the scientific workflow and accelerating the pace of research.

Features we need

- Align millions of Illumina reads to the human genome
- Gapped alignment
- Fast
- High accuracy

Existing GPU-based aligners fall short

	Aligner	Drawback
Protein sequence aligners	CUDA-BLASTP	Protein sequence alignment only
	CUDASW++	Protein sequence alignment only
	GPU-BLAST	Protein sequence alignment only
	GPU-HMMER	Protein sequence alignment only
DNA sequence aligners	MUMmerGPU	• Exact matching • No gapped alignment
	MUMmerGPU++	• Exact matching • No gapped alignment
	UGENE	• Allows up to 3 mismatches • Doesn't seem to perform gapped alignments
	SeqNFind	• Commercial • Need to buy along with hardware
	SARUMAN	• Available in binary format only • Uses Needleman-Wunsch global alignment algorithm • More suitable for microbial sized genomes

Swift: a new GPU-based aligner

Swift: a new GPU-based aligner

- GPU-based DNA sequence alignment program
- Developed using C and CUDA
- Uses Smith-Waterman alignment algorithm
- Gapped alignment
- Aligns millions of Illumina reads to the human genome
- Outputs the best scoring alignment
- Paired-end alignment*
- SAM output*
- Run from command-line
- Currently works on Linux only

* Code needs to be updated

Hardware/software requirements

- Linux OS
- 4 GB of system memory
- GPU
  - CUDA compatible GPU card
  - CUDA toolkit 3.0+
  - 1 GB of global memory
  - 16 KB of shared memory

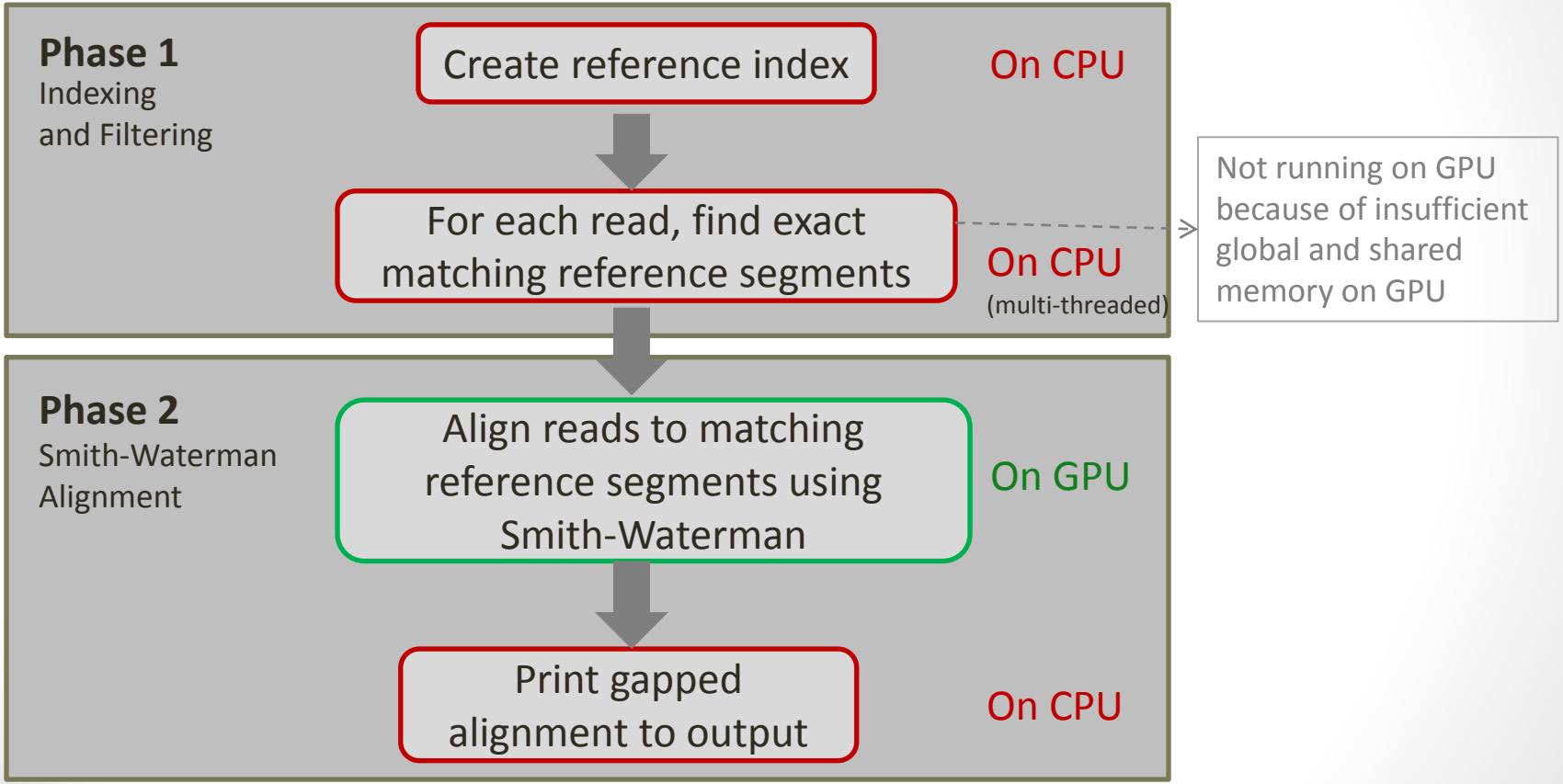
Installation

1. Download the tarball from the Sourceforge website (https://sourceforge.net/projects/swiftseqaligner/)
2. Untar the tarball
   - `$ tar -xvzf swift-0.11.1.tar.gz`
3. Change to the directory containing the source code
   - `$ cd /path/to/swift-0.11.1`
4. Compile the code
   - `$ make`
5. Run it
   - `$ ./bin/swift -q <queryFile> -r <refFile> -o <outFile>`

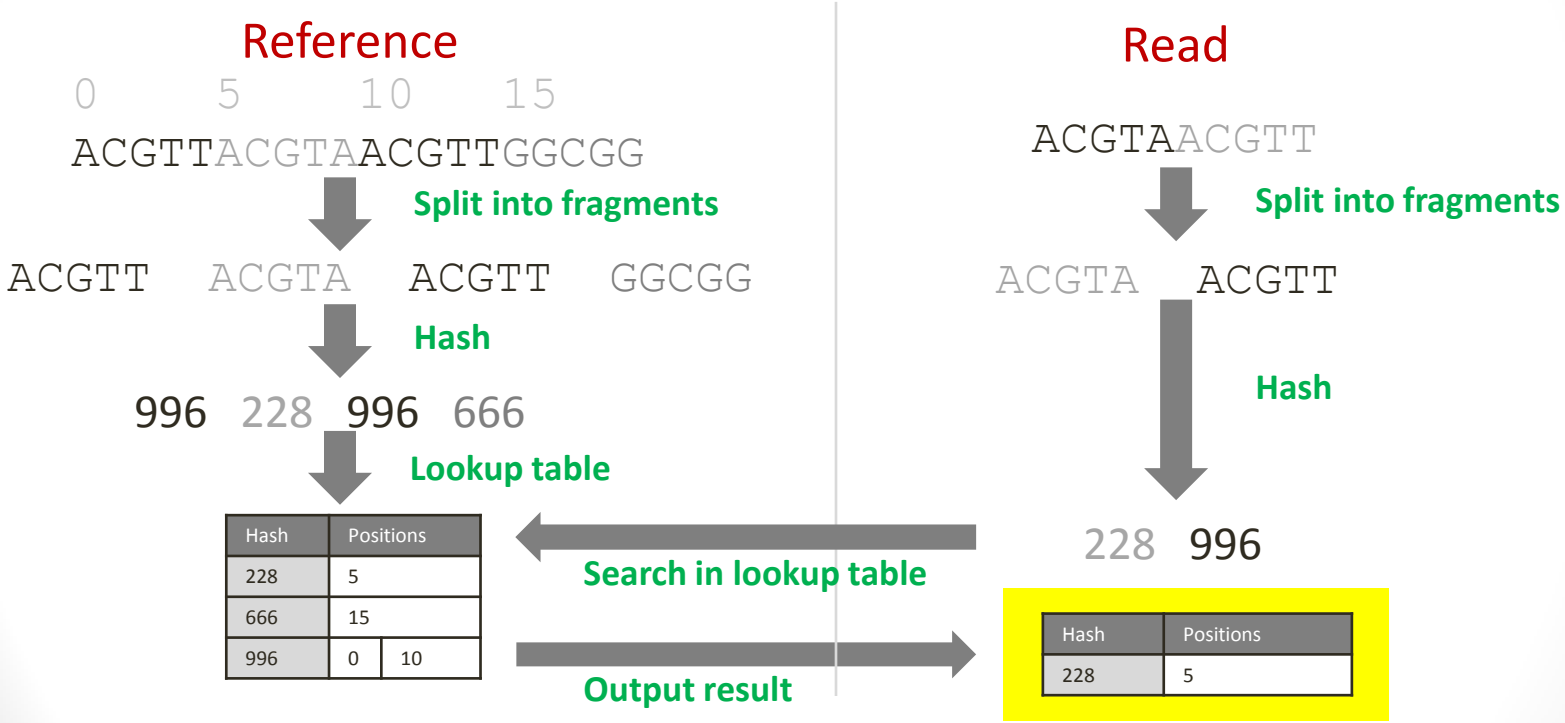
Usage

```
USAGE:
   /path/to/swift -q <query fasta file> -r <reference fasta file> -o <output file> [optional parameters]

DESCRIPTION:
   Aligns multiple query sequences to multiple reference sequences using the Smith-Waterman algorithm.

PARAMETERS:
   -q     Query fasta file (required)
   -q2    Paired query fasta file
   -r     Reference fasta file (required)
   -o     Output file (required)
   -s     Query sequence size
   -n     Number of queries
   -S     Maximum reference sequence size
   -N     Number of references
   -l     Length of a seed (Default: 12)
   -m     Match score (Default: 2)
   -M     Mismatch score (Default: -1)
   -O     Gap open penalty (Default: -10)
   -E     Gap extension penalty (Default: -1)
   -ms    Minimum sequence fragment size (Default: 200)
   -MS    Maximum sequence fragment size (Default: 400)
   -t     Threshold value used to ignore reference tuples
   -cpu   Run program on CPU only
   -f     Output format
          0 - Default program output format. Output includes alignment, score, positions, and length.
          1 - SAM format
   -v     Print program version
   -h     Print usage
```

Method

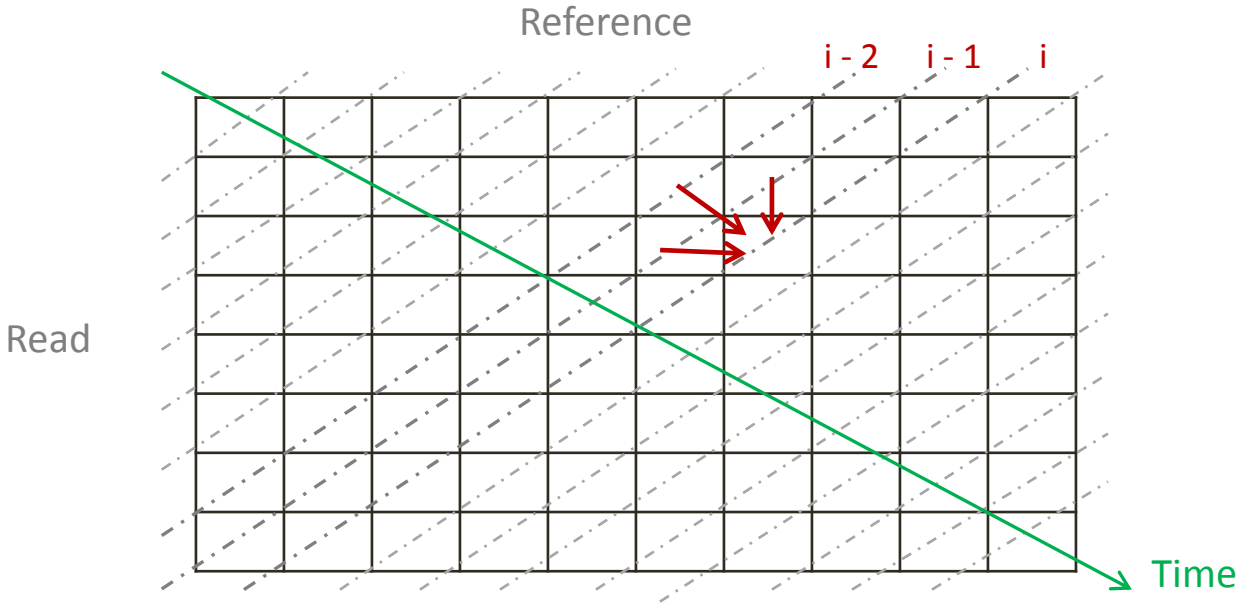

Phase 1
Indexing
and Filtering
Create reference index
On CPU
For each read, find exact matching reference segments
On CPU
(multi-threaded)
Not running on GPU because of insufficient global and shared memory on GPU
Phase 2
Smith-Waterman
Alignment
Align reads to matching reference segments using Smith-Waterman
On GPU
Print gapped alignment to output
On CPU

Phase 1: Indexing and Filtering (on CPU)



Note: This implementation is based on SSAHA (Ning et al. 2001)

Phase 2: Smith-Waterman Alignment (on GPU)



Input and Output

- Input
  - Reference sequence file (FASTA)
  - “Read” sequence file (FASTA)
- Output
  - Alignment
  - Alignment score
  - Alignment positions

```
Qry: Chr1:11021-11254/1 | Ref: Chr1 | Score: 200.0 | AlignStart: 11022 | AlignEnd: 11121 | AlignLength: 100
R: gggggcgtgtgttgcaggagcaaagtcgcacggcgccgggctggggcgggggagggtggcgccgtgcacgcgcagaaactcacgtcacggtggcgcggc
   ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Q: GGGGGCGTGTGTTGCAGGAGCAAAGTCGCACGGCGCCGGGCTGGGGCGGGGGGAGGGTGGCGCCGTGCACGCGCAGAAACTCACGTCACGGTGGCGCGGC
```

Results

- **Test data set**

Reads (simulated)	14,306,494 single-end, 100 bases each
Reference	Human genome (>3 billion base pairs)

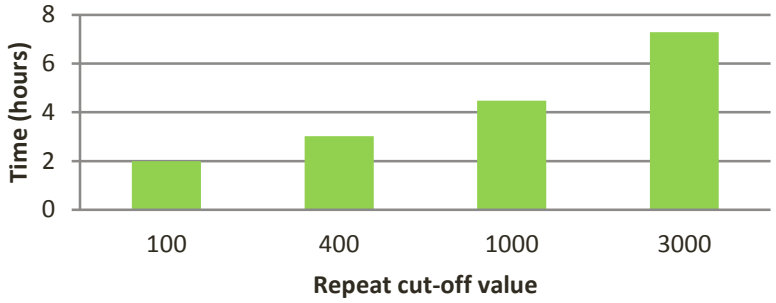
- **Machine**

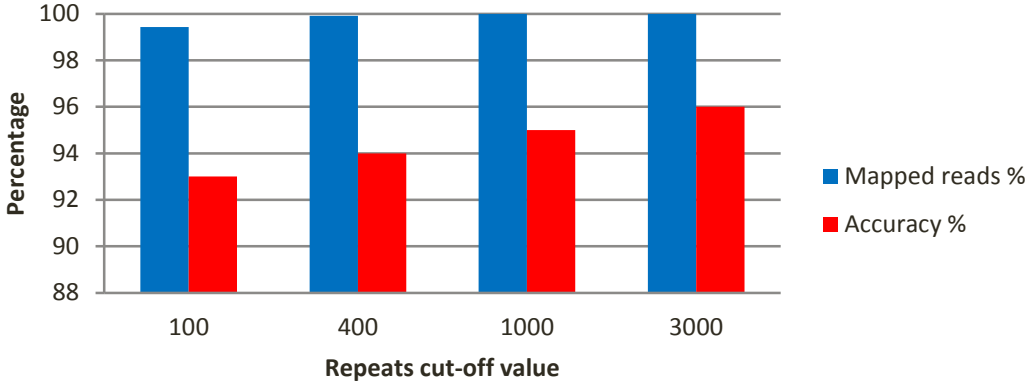
CPU	8 core Intel Xeon; 48 GB RAM
GPU	4 Tesla C2050 GPUs; 2.8 GB global memory; 48 KB shared memory

- **Results**

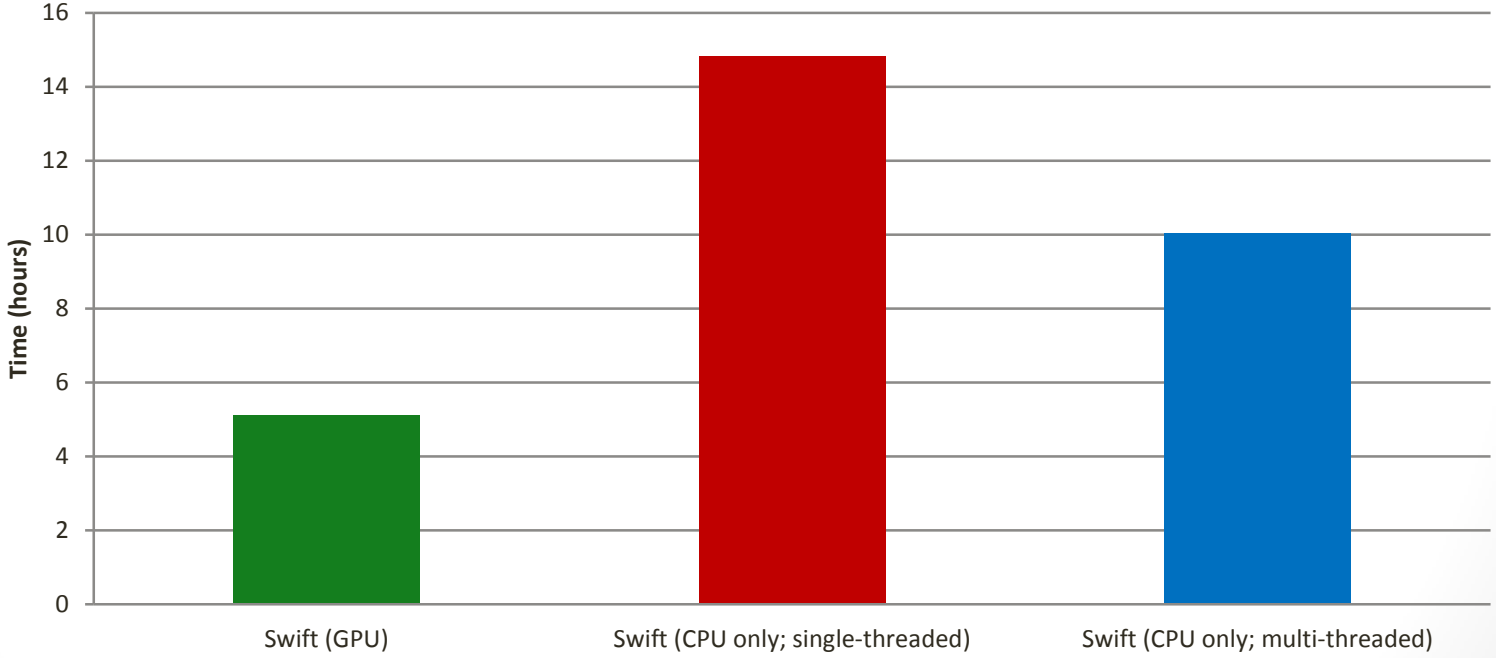
Time	**4h 28m**
Mapped reads	**99.99%**
Reads mapped correctly	**95%**

Effect of cut-off value on time and accuracy





Speed comparison to CPU-based implementation



BWA: a brief introduction

- A fast CPU-based sequence alignment program
- Developed by Heng Li and Richard Durbin at the Sanger Institute
- Uses Burrows-Wheeler Transform
- Uses prefix trie string matching
- Gapped alignment

BFAST: a brief introduction

- A CPU-based sequence alignment program
- Developed by Homer et al. at UCLA
- Uses Smith-Waterman
- Has at least 4 steps:
  1. FASTA to binary conversion
  2. Indexing
  3. Matching
  4. Alignment using Smith-Waterman

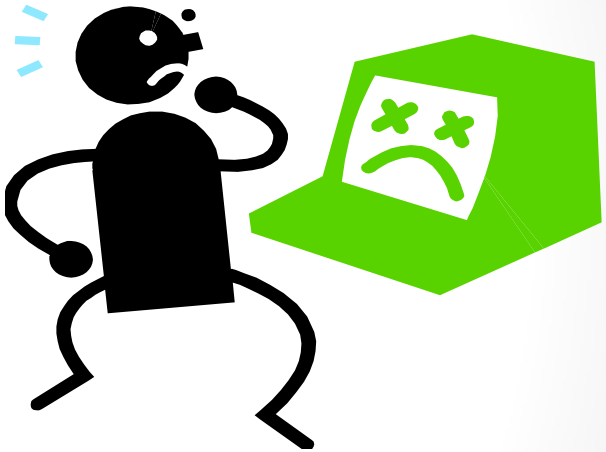
Comparison to other programs

Program	Repeats cut-off value	Time	Mapped reads	Correct fragment
Swift	100	2h	99.43%	93%
	400	3h 1m	99.91%	94%
	1000	4h 28m	99.99%	95%
	3000	7h 17m	100%	96%
BWA	-	2h	100%	97%
BFAST	-	3h 38m*	96%*	-

* localalign step kept aborting and could not be run

Problems faced

- **High number of repeats** in the genome is a major bottleneck
- **Not enough GPU memory** to perform Phase 1 on GPU
- **Lot of experimentation**
- **Long design-code-test cycle**
- **Debugging GPU programs is not easy**



Conclusion

- Swift is a GPU-based DNA sequence alignment program
- Gapped alignment using Smith-Waterman
- Repeats in the genome is a major cause of slow performance
- Accuracy can be improved by increasing the cut-off value for allowed number of repeat segments in reference
- Currently, not as fast as BWA
- Has potential to be faster and more accurate

Conclusion (contd.)

- **Bottom-line: Swift is a better GPU-based DNA sequence alignment program because**:
  - Uses Smith-Waterman
  - Gapped alignment
  - Uses affine gap penalty
  - No limit on the number of mismatches
  - Performs paired-end alignment*
  - Supports output in SAM format*
  - Free (GPL license)

* Code needs to be updated

Future work

- Port Phase 1 to higher memory GPU card (Tesla C2075)
- Use multiple GPUs
- Input reads in FASTQ format

Acknowledgements

- John Obenauer, PhD

  Bioinformatics Group Leader

  St. Jude Children's Research Hospital

- St. Jude Children's Research Hospital

References

- Li H., and Durbin R. 2009. Fast and accurate short read alignment with Burrows-Wheeler Transform. *Bioinformatics*. **25:** 1754-60.
- Homer, N., Merriman, B., Nelson, S.F. 2009. BFAST: An Alignment Tool for Large Scale Genome Resequencing. *PLoS ONE.* **4(11):** e7767.
- Mosaik: http://bioinformatics.bc.edu/marthlab/Mosaik
- Altschul, S., Gish, W., Miller, W., Myers, E., Lipman, D. 1990. Basic local alignment search tool. *Journal of Molecular Biology*. **215(3):** 403–410.
- Tesla Bio Workbench: http://www.nvidia.com/object/tesla_bio_workbench.html
- Ning, Z., Cox, A.J., and Mullikin, J.C. 2001. SSAHA: A Fast Search Method for Large DNA Databases. *Genome Research*. **11:** 1725-1729.
- Smith, T.F., and Waterman, M.S. 1981. Identification of Common Molecular Subsequences. *Journal of Molecular Biology*. **147:** 195–197.
- CUDA-BLASTP: http://sites.google.com/site/liuweiguohome/software
- Liu, Y., Maskell, D., Schmidt, B. 2009. CUDASW++: optimizing Smith-Waterman sequence database searches for CUDA-enabled graphics processing units. *BMC Research Notes*. **2:** 73.

References (contd.)

- Vouzis, P.D., and Sahinidis, N.V. 2011. GPU-BLAST: using graphics processors to accelerate protein sequence alignment. *Bioinformatics*. **27(2):** 182-188.
- Walters, J.P., Balu, V., Kompalli, S., and Chaudhary, V. 2009. Evaluating the use of GPUs in Liver Image Segmentation and HMMER Database Searches. *International Parallel and Distributed Processing Symposium (IPDPS).* Rome, Italy.
- Trapnell, C., Schatz, M. 2009. Optimizing data intensive GPGPU computations for DNA sequence alignment. *Parallel Computing*. **35:** 429-440.
- Gharaibeh, A., and Ripeanu, M. 2010. Size Matters: Space/Time Tradeoffs to Improve GPGPU Applications Performance. *IEEE/ACM International Conference for High Performance Computing, Networking, Storage, and Analysis (SC 2010).* New Orleans, LA.
- Blom, J., Jakobi, T., Doppmeier, D., Jaenicke, S., Kalinowski, J., Stoye, J., and Goesmann, A. 2011. Exact and complete short read alignment to microbial genomes using GPU programming. *Bioinformatics*. **27:** 1351-1358.
- Carr, D.A., Paszko, C., Kolva, D. 2011. SeqNFind: A GPU Accelerated Sequence Analysis Toolset Facilitates Bioinformatics. *Nature Methods*.
- Okonechinkov, K., Grekhov, G., Stepanyuk, K., and Fursov, M. Application of GPGPU for Acceleration of Short DNA Sequence Alignment in Unipro UGENE Project.

Thank you!

Pankaj.Gupta@StJude.org