GPU Accelerated Genetic Variation Discovery

SNP Detection & MAF Computation

Mian Lu, PhD Student
E-mail: lumian@cse.ust.hk
Hong Kong University of Science and Technology
BGI-NVIDIA Joint Innovation Lab

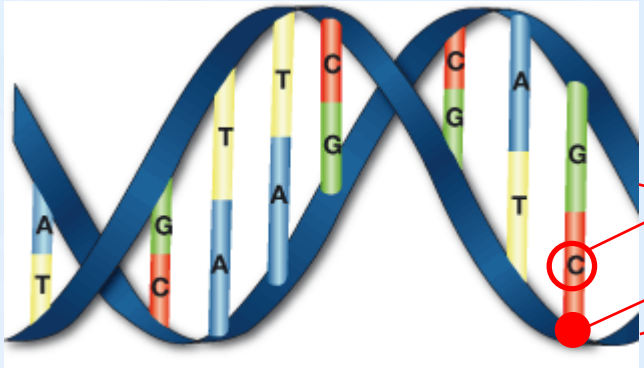
Overview

- Genetic variation discovery.
  - Diseases and drug research.
- Two categories of variation detection software.
  - Detection based on a data set of an individual.
  - Detection based on a data set of a population.

	CPU	GPU
Individual	**SOAPsnp**	**GSNP**
Population	**realSFS**	**GAMA**

Single-Nucleotide Polymorphism (SNP)

- DNA bases, sites, and sequences.

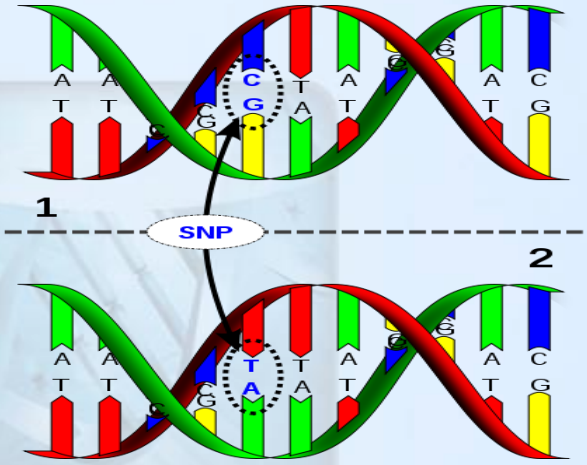


Base: A, T, C, G

Site: the position storing a base

Sequence: two strands

- What is a SNP?

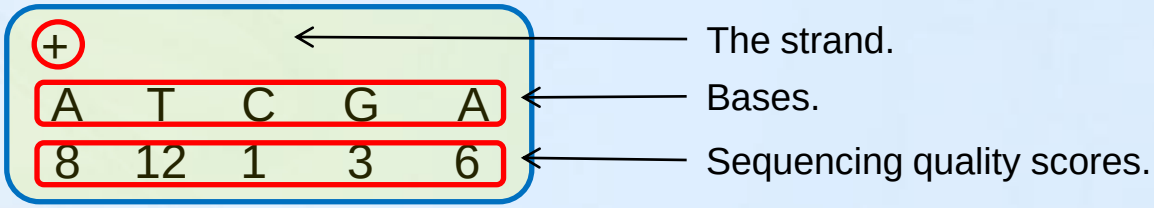


DNA variation on a single nucleotide.

(from Wikipedia: http://en.wikipedia.org/wiki/Single-nucleotide_polymorphism)

SNP Detection on Short Reads

- A short *read*: a DNA fragment.



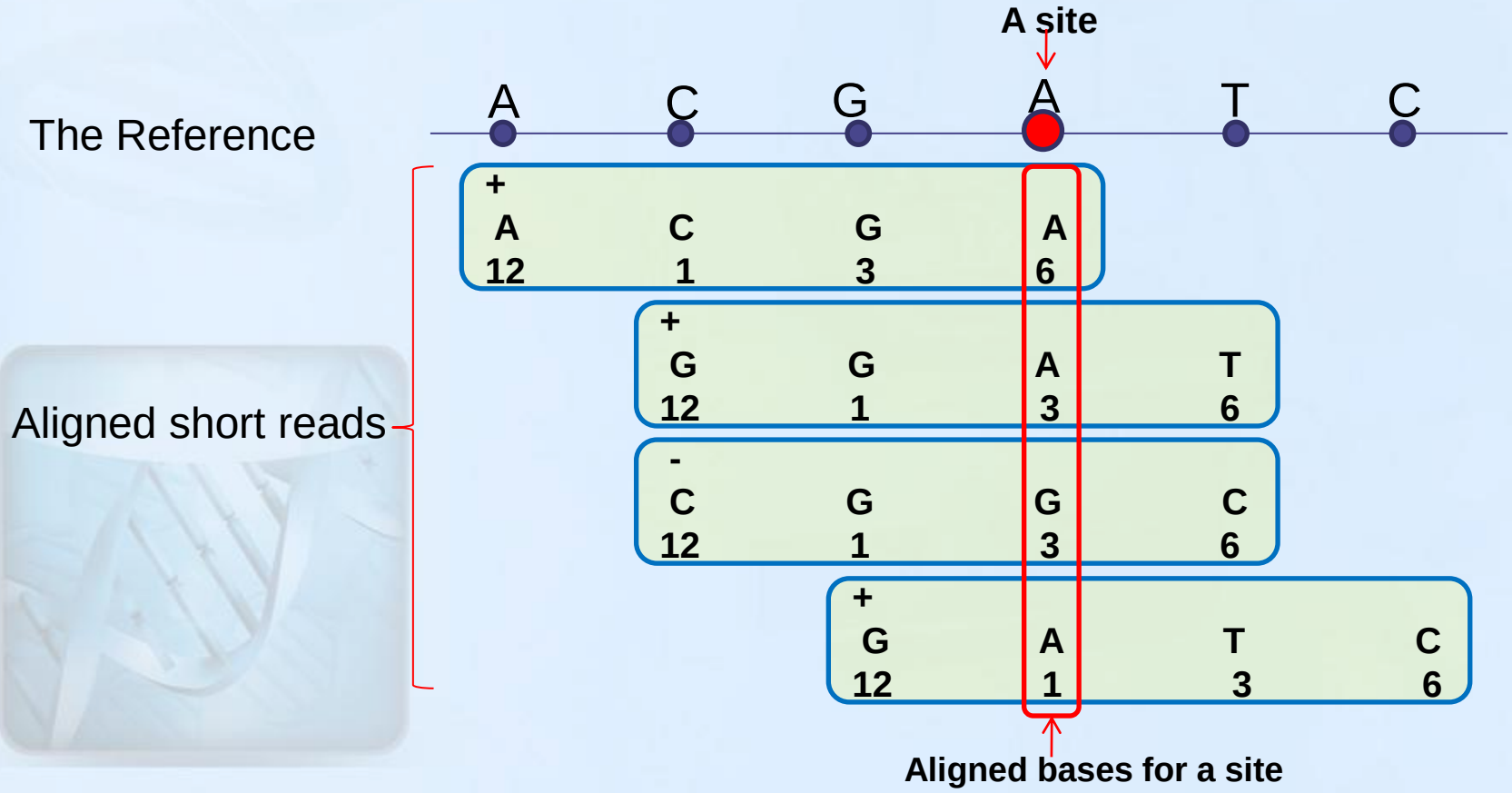
- A popular SNP detection tool for short reads of an individual: **SOAPsnp**.

华大基因 BGI

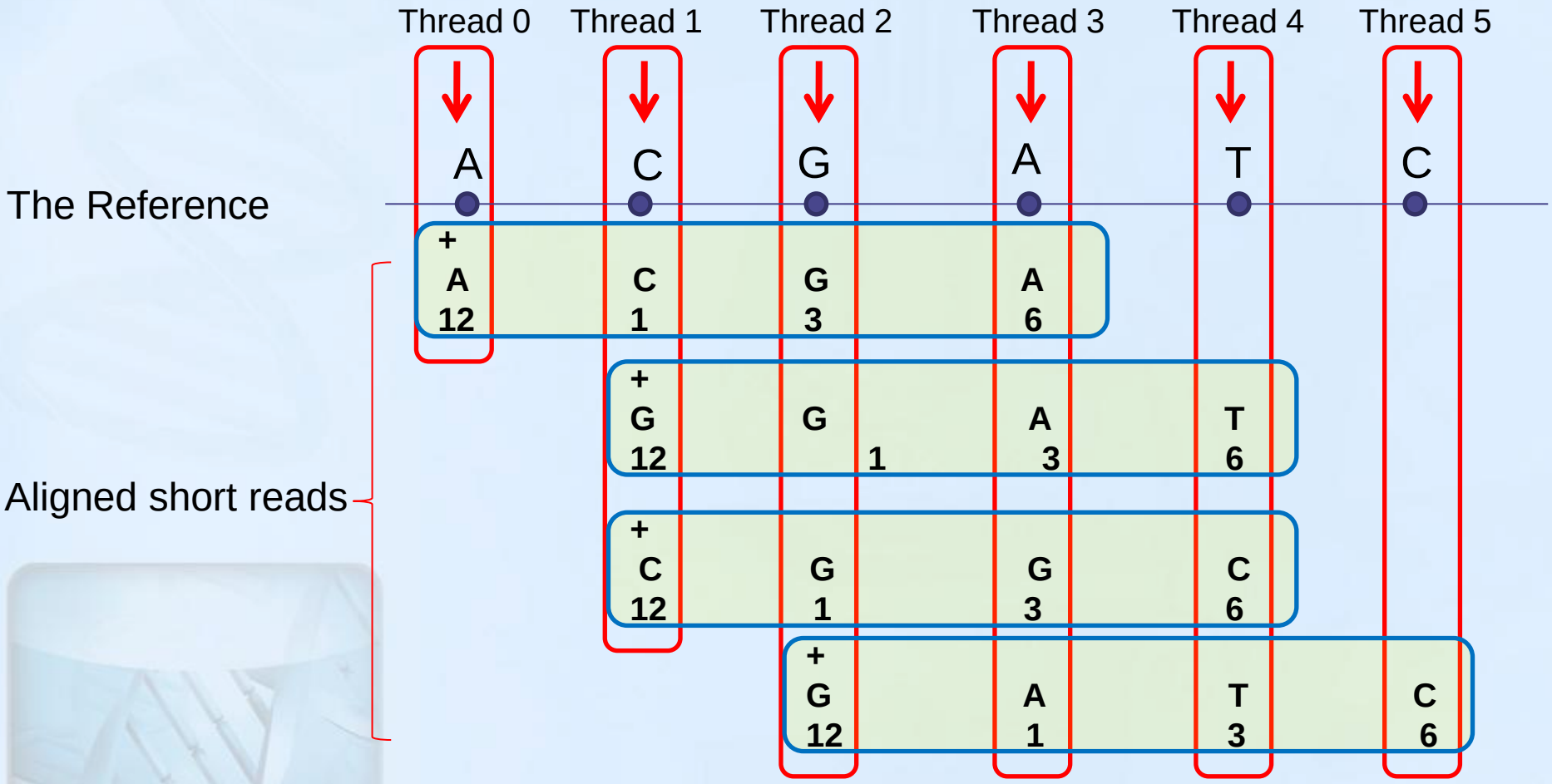
SOAP Short Oligonucleotide Analysis Package

Input Data of SOAPsnp

- The DNA reference sequence.
  - Billions of bases (human genome).
- A large number of *aligned* short reads from an individual.
  - Several billions of short reads (tens of bases in each read).



GPU-Based Parallelization of SOAPsnp



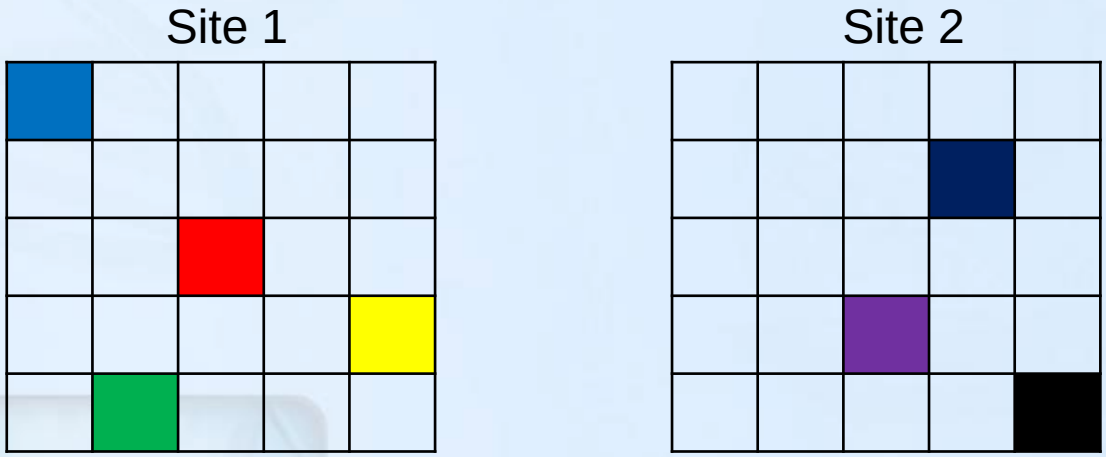
The parallelization strategy on the GPU: one thread handles one site

Optimization Techniques of GSNP

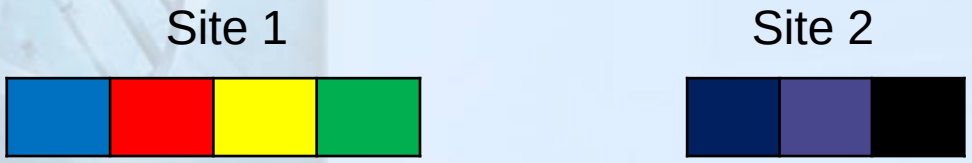
- A sparse data representation format for aligned bases.
  - Reduce memory overhead and branch divergence.
- A preprocessing step for sorting a large number of small arrays.
  - Balance workloads.
- A pre-calculated table storing results of logarithm.
  - The Consistency of GPU and CPU Results.
- Customized compression algorithms for output.
  - Reduce I/O cost.

The Sparse Representation of Aligned Bases

- A matrix to store aligned bases for one site.
- Likelihood computation only for non-zero elements.
- SOAPsnp



- The measured non-zero%: ~0.1%
- GSNP



The Consistency of GPU and CPU Results

- The inconsistency result of the GPU and CPU.

```
1  float a = func();
2  int b = int(a);
3  int result = h[b];
```

h is a lookup table storing scores to indicate whether it is a SNP

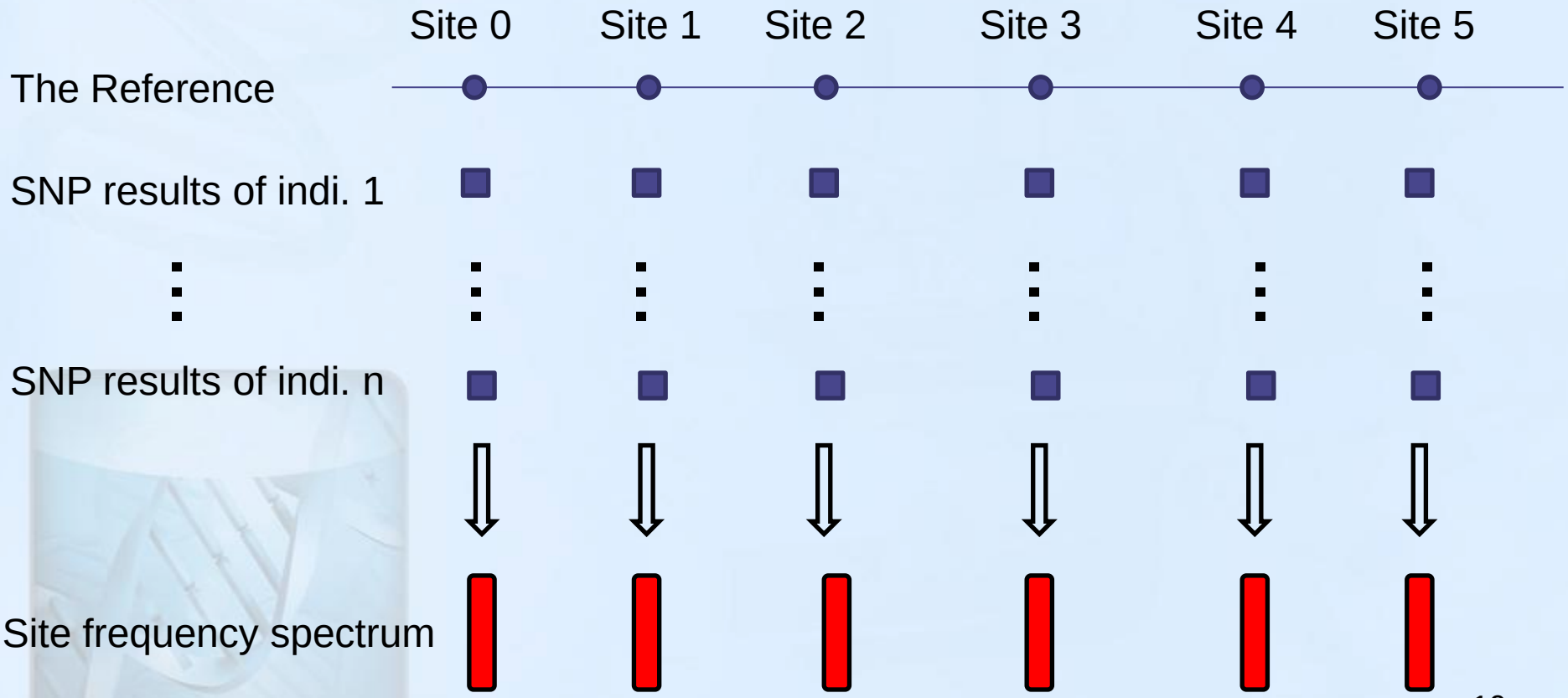
- Example

```
int(1.99999999999999) = 1
int(2.00000000000000) = 2
```

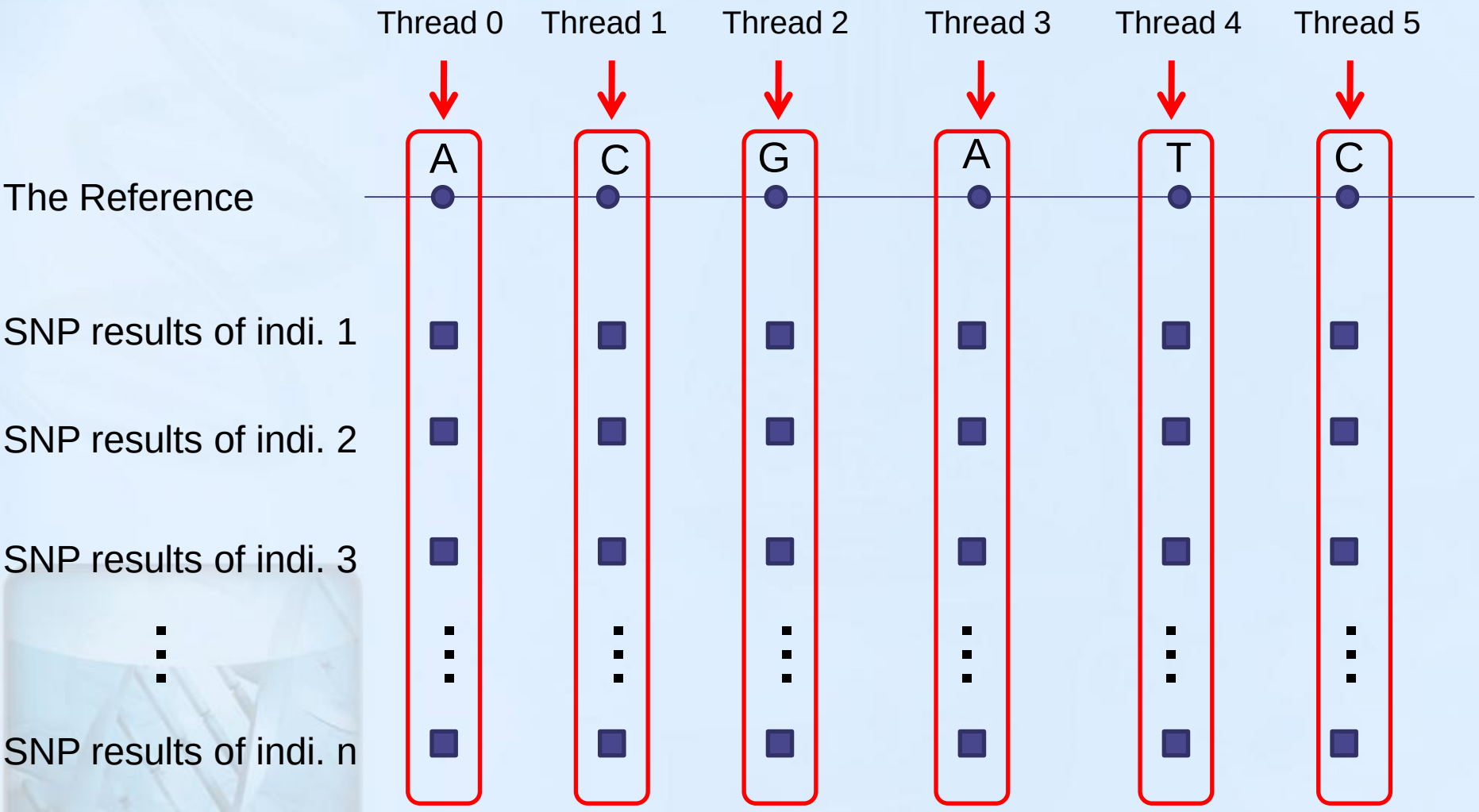
- Solution
  - The inconsistency is generated from a logarithm function.
  - The logarithm is only applied to 64 quality scores.
  - We calculate a table containing 64 logarithm results once on the CPU, and store it in the *constant memory* on the GPU.

MAF Computation (GAMA)

- Minor Allele Frequency (MAF) computation for a population.
- Input: SNP detection results for individuals.



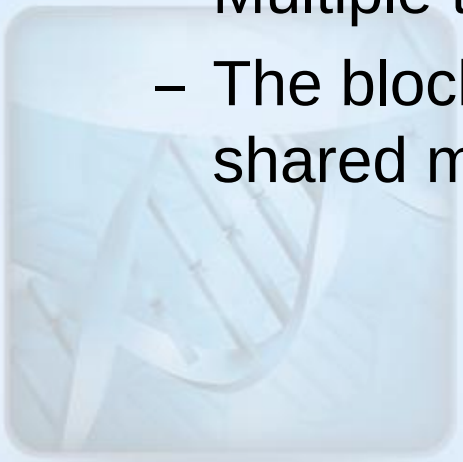
The Parallelization for MAF Computation



Is this a good solution?
Bad for SFS construction.

Issues of Per-Thread-Per-Site SFS Construction

- Large shared memory consumption for a single thread.
  - For 1024 individuals, around 16 KB per thread.
- Very low occupancy of the multiprocessor.
- Our solution:
  - Convert the iterative-update based algorithm to matrix multiplication based algorithm.
  - Multiple threads handle a site.
  - The block-nested loop implementation to utilize the shared memory.



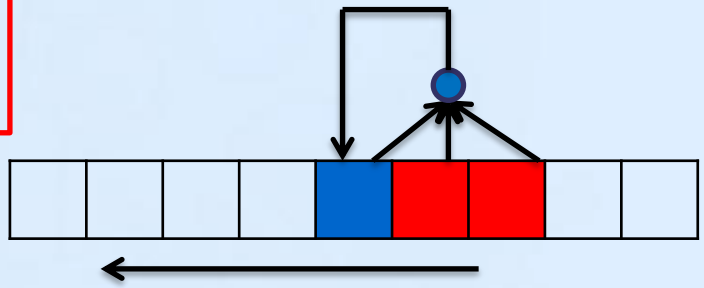
The Parallelization of SFS Construction

$h[0] = P_{aa}h[0]$

$h[1] = P_{Aa}h[0] + P_{aa}h[1]$

for (j = 2(N + 1); j > 1; j--)

$h[j] = P_{AA}\,h[j-2] + P_{Aa}\,h[j-1] + P_{aa}\,h[j]$



The iterative update on *h* from right to left.

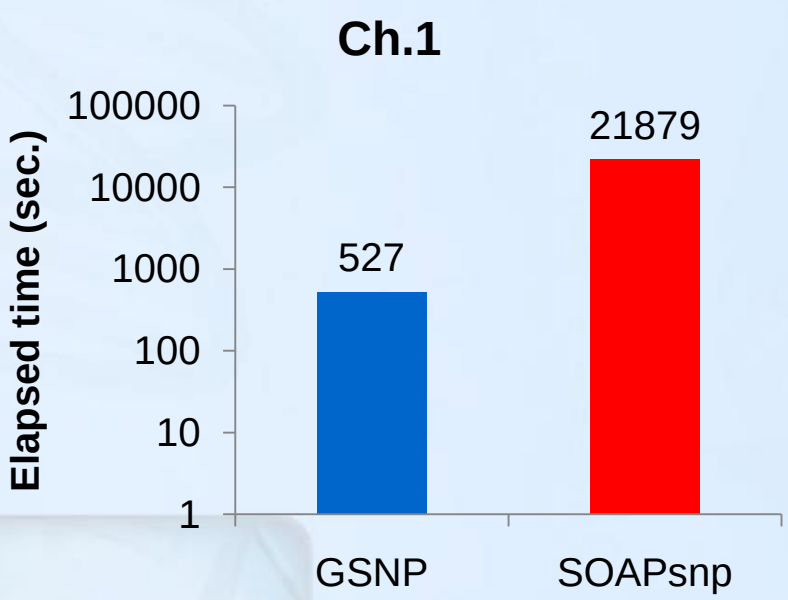
$$\begin{bmatrix} P_{aa}^N & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ P_{Aa}^N & P_{aa}^N & 0 & 0 & 0 & 0 & \cdots & 0 \\ P_{AA}^N & P_{Aa}^N & P_{aa}^N & 0 & 0 & 0 & \cdots & 0 \\ 0 & P_{AA}^N & P_{Aa}^N & P_{aa}^N & 0 & 0 & \cdots & 0 \\ 0 & 0 & P_{AA}^N & P_{Aa}^N & P_{aa}^N & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & P_{AA}^N & P_{Aa}^N & P_{aa}^N & 0 \\ 0 & 0 & 0 & \cdots & 0 & P_{AA}^N & P_{Aa}^N & P_{aa}^N \end{bmatrix}_{(2N+1)(2N+1)} \begin{bmatrix} h[0] \\ h[1] \\ h[2] \\ h[3] \\ \vdots \\ h[2N-1] \\ 0 \\ 0 \end{bmatrix}_{2N+1} = \begin{bmatrix} h[0] \\ h[1] \\ h[2] \\ h[3] \\ \vdots \\ h[2N-1] \\ h[2N] \\ h[2N+1] \end{bmatrix}$$

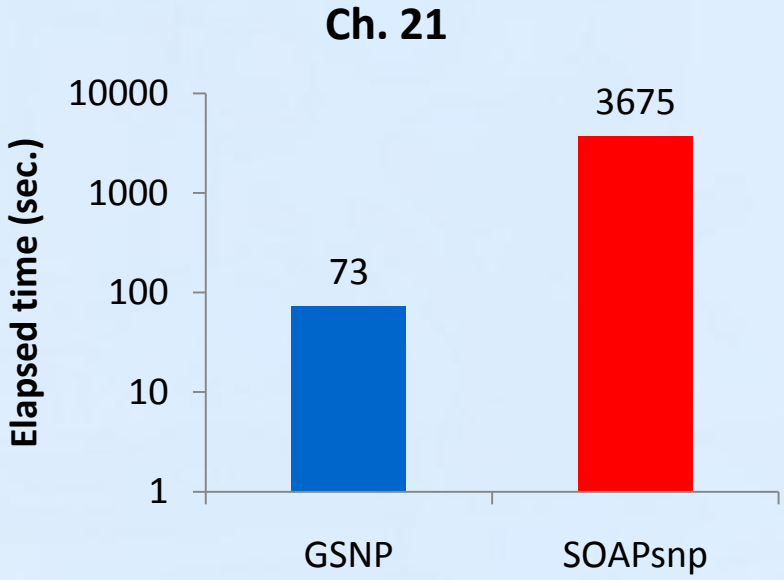
Experimental Setup for GSNP

- Hardware
  - CPU: Intel Xeon E5630 2.53 GHz, 64 GB memory.
  - GPU: NVIDIA Tesla M2050, 448 cores, 3 GB.
- CPU counterpart: SOAPsnp 1.03 (single-thread).
- Data sets: human chromosome 1 and 21.

	#sites	Seq. depth	#reads	Input	Output
Ch. 1	247 M	11X	44 M	12 GB	17 GB
Ch. 21	47 M	9.6X	6 M	2 GB	3 GB

End-to-End Performance Comparison of GSNP





The elapsed time of all components are included.
GSNP is around 50X faster than the single-thread CPU-based SOAPsnp.

Computation Performance of GAMA

- Single-thread realSFS (optimized) vs. GAMA
  - 2.7 million sampled sites

realSFS	GAMA	Speedup
1.7 hour	2.2 mins.	47X

- Estimated performance for the whole human genome (3 billion sites).
  - realSFS: 2.4 months
  - GAMA: 1.5 day

CPU: Intel Xeon E5630 2.53 GHz, 64 GB memory
GPU: NVIDIA Tesla C2070, 448 cores, 6 GB

Conclusion

- GPU-accelerated analysis tools for genetic variation discovery.
  - A single individual: GSNP.
  - Multiple individuals: GAMA.
- More detail:
  - Software:
    http://www.cse.ust.hk/gallop/
    http://jil.genomics.org.cn/
  - Reference:
    Mian Lu, Jiuxin Zhao, Qiong Luo, Bingqiang Wang, Shaohua Fu, Zhe Lin. *GSNP: A DNA Single-Nucleotide Polymorphism Detection System with GPU Acceleration*. 2011 International Conference on Parallel Processing (ICPP-2011).