# Ramethy: Reconfigurable Acceleration of Bisulfite Sequence Alignment

James Arram, Wayne Luk (Imperial College)
Peiyong Jiang (Chinese University of Hong Kong)

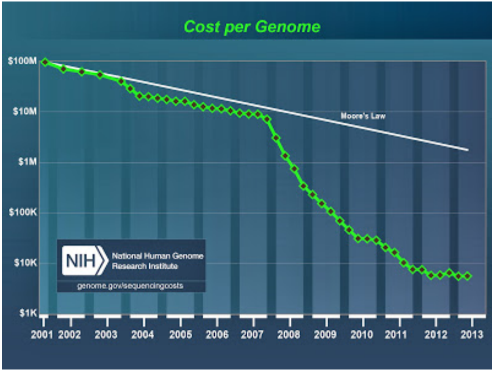
February 24, 2015

# Contributions

- A run-time reconfigurable architecture for accelerating sequence alignment using FPGAs
- An application of this architecture to accelerate bisulfite sequence alignment
- Optimisations which improve the performance of the FM-index search operation
- Implementation of our design: Ramethy on a Maxeler node
  - 14.9x speed of multicore CPU (up to 88.4x)
  - 3.8x speed of GPU (up to 22x)

# Motivation

- Cost of DNA sequencing decreasing faster than Moore's Law
  - Amount of sequenced data increasing
  - Current compute infrastructures struggling
  - Slow analysis = slow diagnosis



# Bisulfite Sequencing

- Particular DNA sequencing used to find methylation status of DNA
  - Mechanism which controls gene expression
- Studies link Methylation to certain illnesses

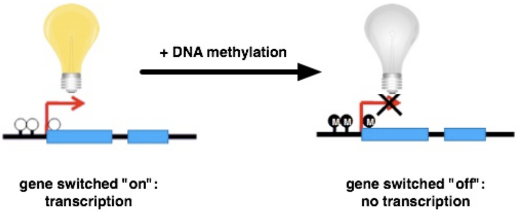


Figure: Image by Karina Zillner

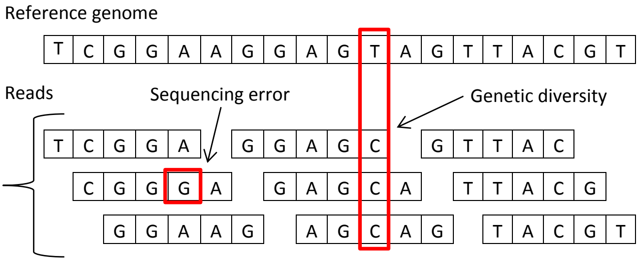
# Bisulfite Sequencing Applications

- Exciting diagnosis methods developed by CUHK:

1. Cancer diagnosis
    - General diagnosis: targets most cancers
    - Early stage detection
2. Prenatal diagnosis
    - Test baby for illness before birth
    - non-invasive: safe for mother and baby

# Bisulfite Sequencing Analysis

- Latest bisulfite sequencing analysis tool: Methy-Pipe (developed by CUHK)
  - Fully integrated, but still slow (∼1 day to perform analysis)
  - Sequence alignment is bottleneck (over 50% of run-time)



# Alignment Programs

- Multiple software programs for sequence alignment
    - Soap2, BWA, Bowtie, etc.
- Several GPU-based alignment programs
    - Soap3-dp, CUSHAW
    - up to 10x faster than CPU-based alignment programs
- FPGA based alignment programs have been proposed
    - Typically accelerate Smith-Waterman alignment algorithm
    - Good speed-up, but often compromise functionality and accuracy

# FM-index

- Given Methy-Pipe alignment parameters, we target FM-index search operation for acceleration
  - Performs alignment using an index of the reference genome
  - Closely related to suffix arrays (SA)
- FM-index search operation:
  - Update SA interval for each symbol in read (backwards search)
  - number updates (iterations) = number symbols in read

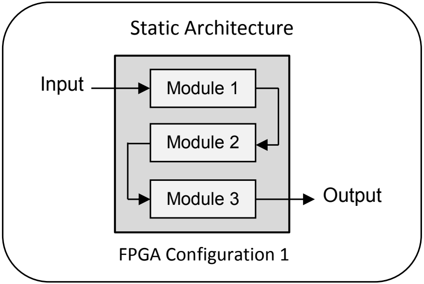
```
initialise SA interval
for each symbol in read (last to first) do
    update SA interval
end for
```

# FM-index Search Operation Analysis

- Extend search with backtracking for inexact alignment
  - Edit operations (sub, ins, del) performed on the read
- Algorithm bottlenecks:
  - Dynamic data access to index (2 per update)
  - Excessive backtracking in inexact alignment
- Optimise algorithm and alignment architecture

# Static Architecture

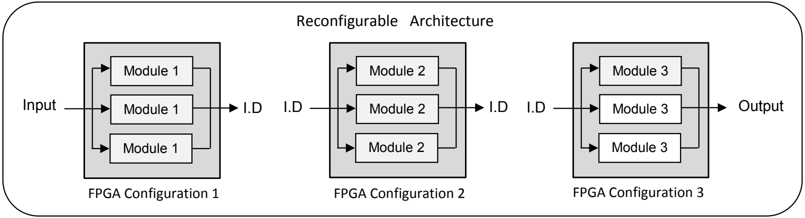
- FPGA is configured with a static alignment circuit



- Performance limited by features of alignment algorithms:
  1. Idle Modules
  2. Unbalanced pipelines
  3. Not enough resources
  4. Inflexible alignment

# Run-time Reconfigurable Architecture

- FPGA configurations *for each* module
  - Modules replicated as many times as possible
- Run-time reconfiguration used to switch configurations



I.D = intermediate data

# Run-time Reconfigurable Architecture Analysis

- Performance of run-time reconfigurable architecture:

$$T = \sum_i (T_i/N_i + t_r + t_t)$$

  - $T_i$: time for module to process input data
  - $N_i$: number of modules in configuration
  - Overheads: $t_r$ (reconfiguration time), $t_t$ (transfer time)
- Performance of static architecture

$$\text{best: } T = \max(T_1/N_1, ...) \qquad \text{worst: } T = \sum_i (T_i/N_i)$$

  - $N_{i:runtime} >> N_{i:static}$

# Run-time Reconfigurable Architecture Analysis

- Reconfiguration addresses limitations of static architecture:

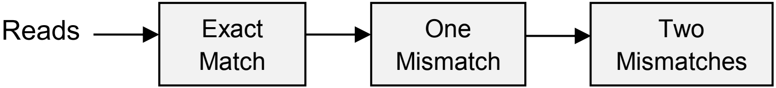
1. Configurations contain single type of module, data partitioned:
    - No idle modules
2. Modules are not interlinked:
    - No unbalanced pipeline of modules
3. Configurations for each stage in alignment algorithm:
    - Easier to map full alignment circuit to hardware
4. Run-time reconfiguration used to switch configurations:
    - Flexible alignment

# FM-index Optimisations

- Reduce dynamic data access: n-step FM-index
    - Old: SA interval updated for 1 symbol per iteration
    - New: SA interval updated for n symbols per iteration
    - Number of dynamic data accesses reduced by factor of $n$
- Reduce excessive backtracking: bi-directional search
    - Old: brute force search
    - New: Use bi-directional search and constrain edit position
    - Reduce search space and excessive backtracking
- More details in paper

# Ramethy: Implementation

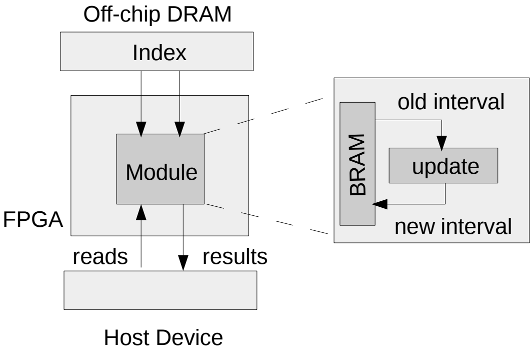
- Design goal: Match alignment parameters of Methy-Pipe
  - Permit up to two mismatches in reads
  - Report unique alignments only



- Implement Ramethy on a 1U Maxeler MPC-X1000 node
  - 8 Altera Stratix V FPGAs each with 48GB of off-chip DRAM
  - Design as data flow graph, then compiled into bitstream

# Module Designs

- Host CPU streams reads to modules, results streamed back
- Index stored in off-chip DRAM
- Module uses BRAM to store reads and SA intervals



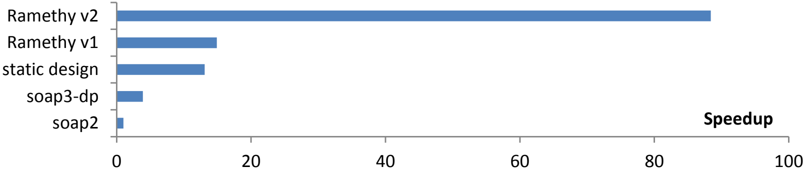
# Experimentation

- Experimental data:
    - Reference genome = chr22 of Human genome (Full 3.3GB)
    - 10M reads of 75 symbols
- Comparisons with:
  soap2 (CPU - 2 Intel Xeon: 32 cores)
  soap3-dp (GPU - 1 NVIDIA GTX 580, 512 cores)
    - Among fastest alignment programs available
    - soap2 used in Methy-Pipe, later versions may use soap3-dp
- Provide 3 performance values for Ramethy (8 FPGAs):
    - v1: measured values (1 module per Conf.)
    - v2: upper bound estimate (3 modules per Conf.)
    - static: best case static design 4 EM, 1 OM, 3 TM (1 module per Conf.)

# Performance: Bisulfite Sequence Alignment

| program | platform | clock freq. | device | Mbps | speedup |
|---|---|---|---|---|---|
| soap2 | Intel E5-2650 | 2000 | 2 | 4.5 | 1.0x |
| soap3-dp | NVIDIA GTX-580 | 772 | 1 | 17.4 | 3.9x |
| static design | MPC-X1000 | 150 | 8 | 58.5 | 13.1x |
| Ramethy v1 | MPC-X1000 | 150 | 8 | 66.4 | 14.9x |
| Ramethy v2 | MPC-X1000 | 150 | 8 | 395 | 88.4x |

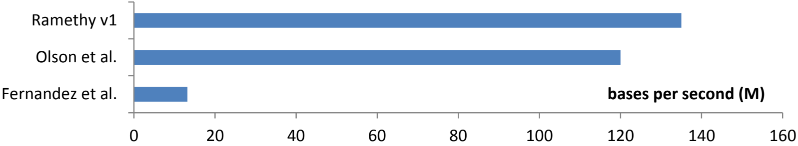
- 14.9x speed of soap2, 3.8x speed of soap3-dp
- Same accuracy as software



# Performance: Standard DNA Sequence Alignment

| program | platform | clock freq. | device | Mbps |
|---|---|---|---|---|
| Fernandez et al. | Convey HC-1 | 150 | 4 | 13.2 |
| Olson et al. | Pico M-503 | 250 | 8 | 120 |
| Ramethy v1 | MPC-X1000 | 150 | 8 | 135 |

- Fernandez et al.: Does not test all edit positions
- Olson et al.: Backtracing?
- Ramethy: Identical to software



# Energy Consumption

- Measure energy consumption from run-time device power (obtained from MaxOS)

| Program | Device Power (W) | Energy Consumption (kJ) |
|---|---|---|
| soap2 | 190 | 31.9 |
| soap3-dp | 244 | 10.5 |
| Ramethy v1 | 72 | 1.5 |

- Almost an order of magnitude less energy consumed by FPGAs (short run-time plus low power)

# Future Work

- Further optimising Ramethy
  - Other ways of reducing dynamic data access?
  - Pre-filer reads for quick discarding
- Accelerating other parts of Methy-Pipe
- Study scientific and clinical impact

# Conclusion

- Accelerate bisulfite sequence alignment using FPGAs
- Run-time reconfigurable architecture for accelerating alignment algorithms
- Target FM-index for acceleration
  - Reduce dynamic data access and excessive backtracking
- Ramethy 14.9x speed of soap2, 3.8x speed of soap3-dp
  - Identical alignment accuracy to software
  - Almost an order of magnitude less energy consumed
- Faster alignment = improved science and healthcare