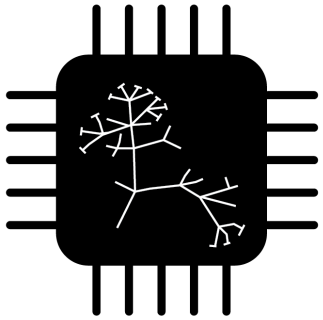


Darwin: A Genomic Co-processor Provides up to 15,000X acceleration on long read assembly

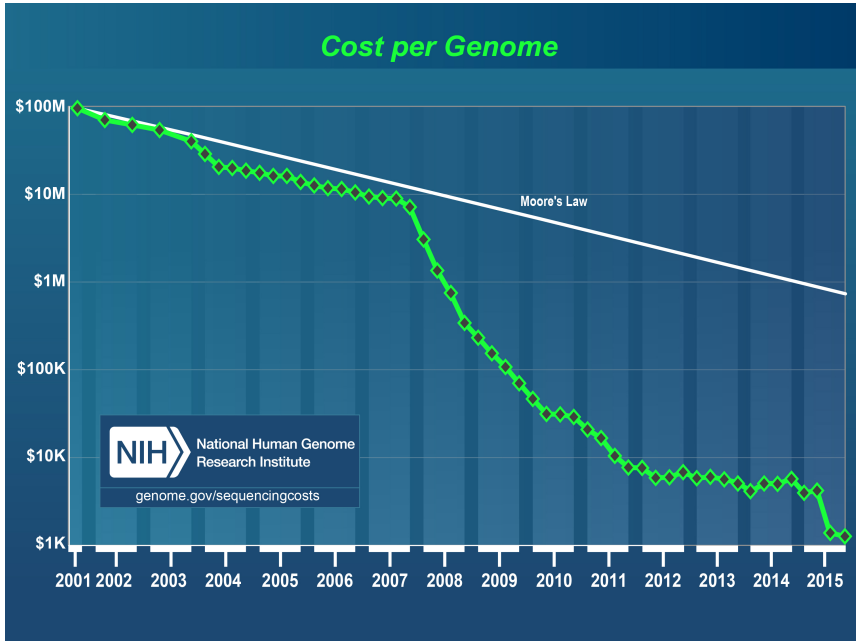
Yatish Turakhia
Prof. Gill Bejerano
Prof. William J. Dally

Stanford University

Sections

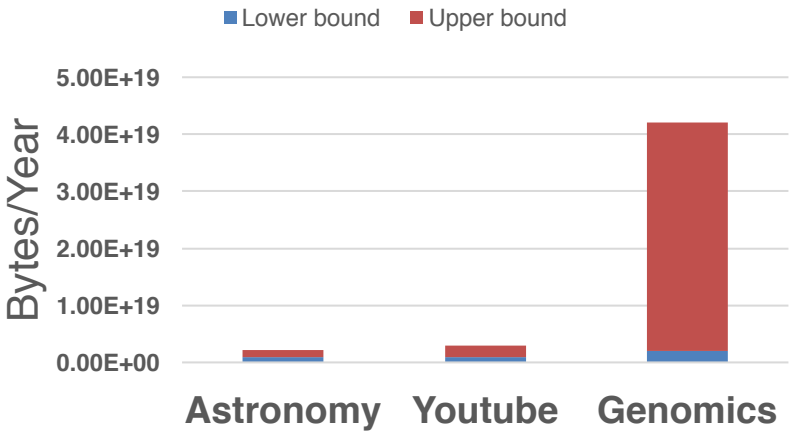
1. **Introduction: DNA Sequencing, Alignment and Long Read Assembly**
2. Darwin: Framework Overview and Design Philosophy
3. D-SOFT: Algorithm and Hardware-Acceleration
4. GACT: Algorithm and Hardware-Acceleration
5. Experimental Methodology and Results

Genomics: Exabyte scale by 2025



[NIH, NHGRI]

Data volume (2025)

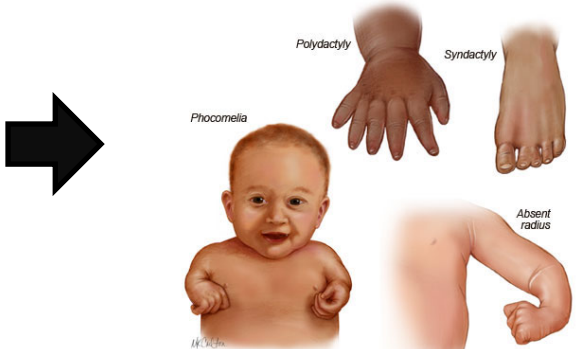


[Astronomical or Genomical, PLOS Bio 2015]

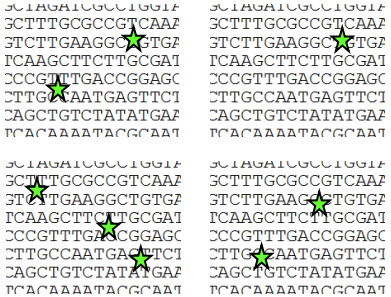
Power of Genomic Data

What genomic mutations predispose us to disease?

Power of Genomic Data

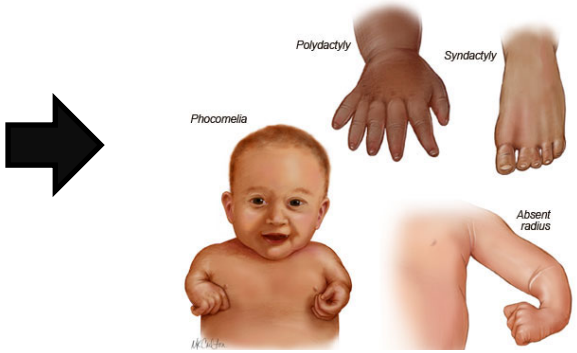


What genomic mutations predispose us to disease?

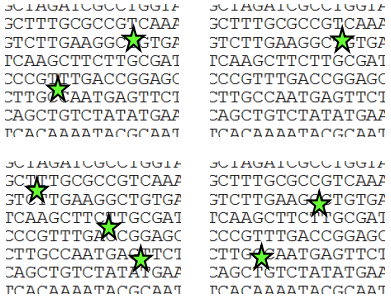


Where did we come from? How are we different from each other?

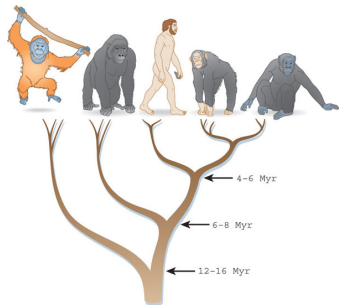
Power of Genomic Data



What genomic mutations predispose us to disease?



Where did we come from? How are we different from each other?



What in our genomes make us different from other species?

DNA Sequencing, Errors and Alignment

Genome AGCTTTACCTACGTAGCTGCATCTATTTCTCGTATTTAGC

Sequencer

[Image: www.malaghan.org]

DNA Sequencing, Errors and Alignment

Genome AGCTTTACCTACGTAGCTGCATCTATTTCTCGTATTTAGC

Sequencer

Read GTGCTTGGATATA

[Image: www.malaghan.org]

DNA Sequencing, Errors and Alignment

Genome AGCTTTACCTACGTAGCTGCATCTATTTCTCGTATTTAGC

Sequencer

Read GTGCTTGGATATA

Sequence Alignment

```
GTAGCT-GCATCTA
||  |||  |x|||x||
GT-GCTTGGATATA
```

[Image: www.malaghan.org]

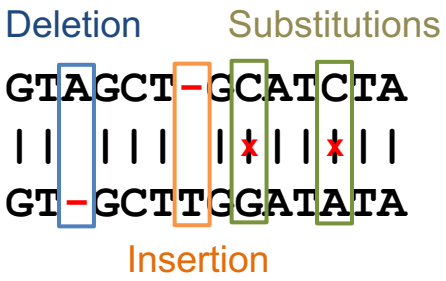
DNA Sequencing, Errors and Alignment

Genome AGCTTTACCTACGTAGCTGCATCTATTTCTCGTATTTAGC

Sequencer

Read GTGCTTGGATATA

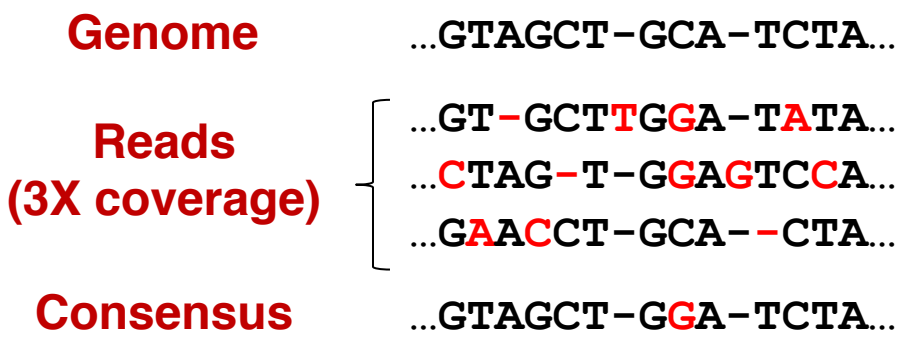
Sequence Alignment



[Image: www.malaghan.org]

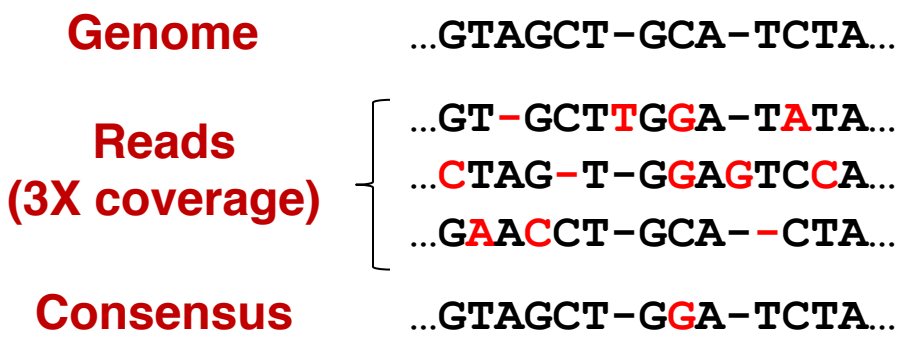
Consensus lowers error-rate in read assembly

- Consensus of multiple reads reduces assembly errors



Long read assembly: Up to 60K CPU hours

- Consensus of multiple reads reduces assembly errors

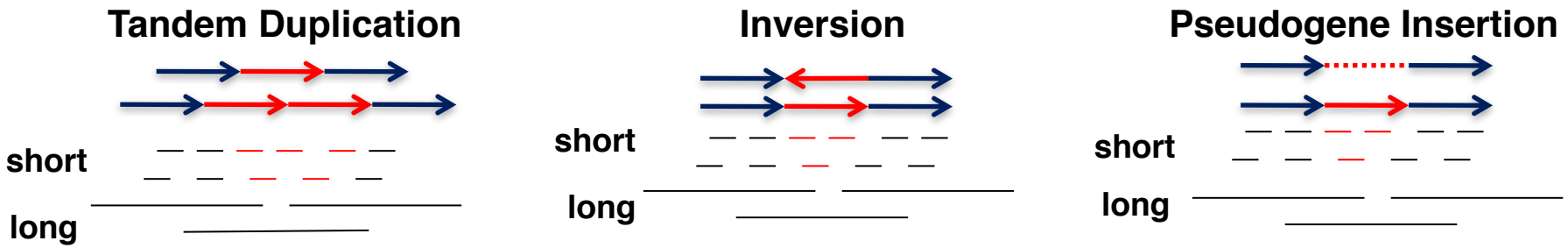


- Noisy long reads have significantly higher compute requirements

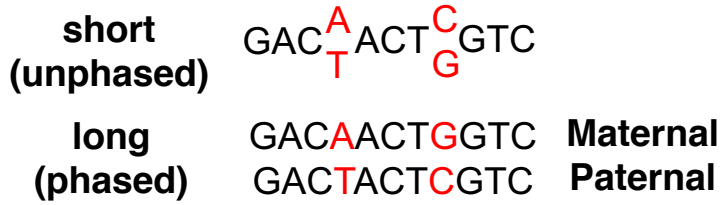
	Reference-guided assembly (54X human)	De novo assembly (54X human)
Short reads (~100bp, 0-2% error rate)	Up to 200 CPU hours	Up to 2,000 CPU hours
Long reads (~10Kbp, 15-40% error rate)	Up to 5,000 CPU hours	Up to 60,000 CPU hours

Long Reads: Holy Grail of Sequencing

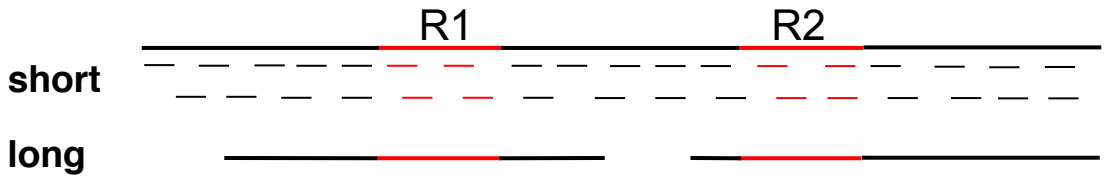
- Discovering structural variants



- Haplotype phasing



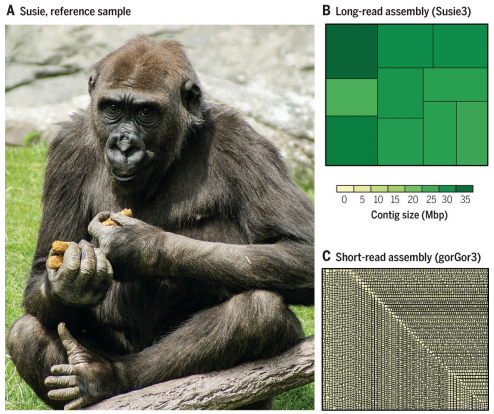
- Resolving repeats



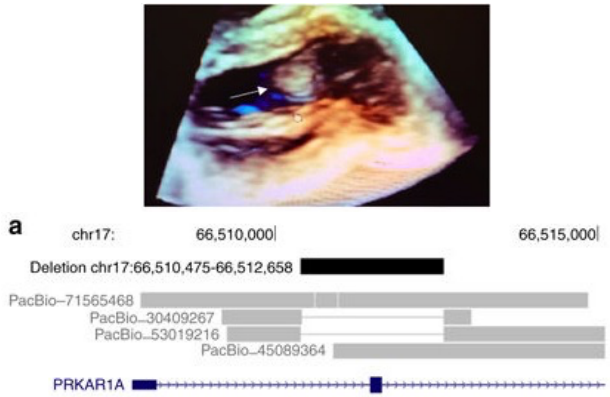
Recent Breakthroughs using Long Read Sequencing

Pacific Biosciences
RS II

Gorilla genome improved 800X
[Science 2016]



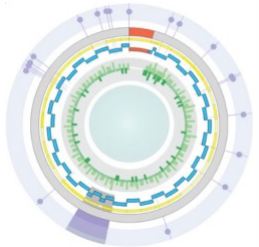
Structural variant in Mendelian disease
[Genetics in Medicine 2017]



Oxford Nanopore
MinIon

First sequencing in
Space [NASA 2017]

Human centromere
[Nat Biotech 2018]



Extremophile sequencing
Antarctica [J Biomol Tech 2017]

Smith-Waterman Algorithm

1 Smith-waterman equations:

$$H(i,j) = \max \begin{cases} H(i-1,j-1) + W(r_i, q_j) \\ H(i-1,j) + gap \\ H(i,j-1) + gap \end{cases}$$

2 Scoring Parameters:

$W =$

	A	C	G	T
A	2	-1	-1	-1
C	-1	2	-1	-1
G	-1	-1	2	-1
T	-1	-1	-1	2

gap = 1

Smith-Waterman Algorithm

1 Smith-waterman equations:

$$H(i,j) = \max \begin{cases} H(i-1,j-1) + W(r_i, q_j) \\ H(i-1,j) + gap \\ H(i,j-1) + gap \end{cases}$$

2 Scoring Parameters:

$W =$

	A	C	G	T
A	2	-1	-1	-1
C	-1	2	-1	-1
G	-1	-1	2	-1
T	-1	-1	-1	2

gap = 1

3 Matrix Fill:

Reference

Query

	*	G	C	G	A	C	T	T	T
*	0	0	0	0	0	0	0	0	0
G	0	2	1	2	1	0	0	0	0
T	0	1	1	1	1	0	2	2	2
C	0	0	3	2	1	3	2	1	1
G	0	2	2	5	4	3	2	1	0
T	0	1	1	4	4	3	5	4	3
T	0	0	0	3	3	3	5	7	6
T	0	0	0	2	2	2	5	7	9

Smith-Waterman Algorithm

1 Smith-waterman equations:

$$H(i,j) = \max \begin{cases} H(i-1,j-1) + W(r_i, q_j) \\ H(i-1,j) + gap \\ H(i,j-1) + gap \end{cases}$$

2 Scoring Parameters:

$W =$

	A	C	G	T
A	2	-1	-1	-1
C	-1	2	-1	-1
G	-1	-1	2	-1
T	-1	-1	-1	2

gap = 1

3 Matrix Fill:

Reference

Query \ Reference	*	G	C	G	A	C	T	T	T
*	0	0	0	0	0	0	0	0	0
G	0	2	1	2	1	0	0	0	0
T	0	1	1	1	1	0	2	2	2
C	0	0	3	2	1	3	2	1	1
G	0	2	2	5	4	3	2	1	0
T	0	1	1	4	4	3	5	4	3
T	0	0	0	3	3	3	5	7	6
T	0	0	0	2	2	2	5	7	9

4 Trace-back:

Alignment

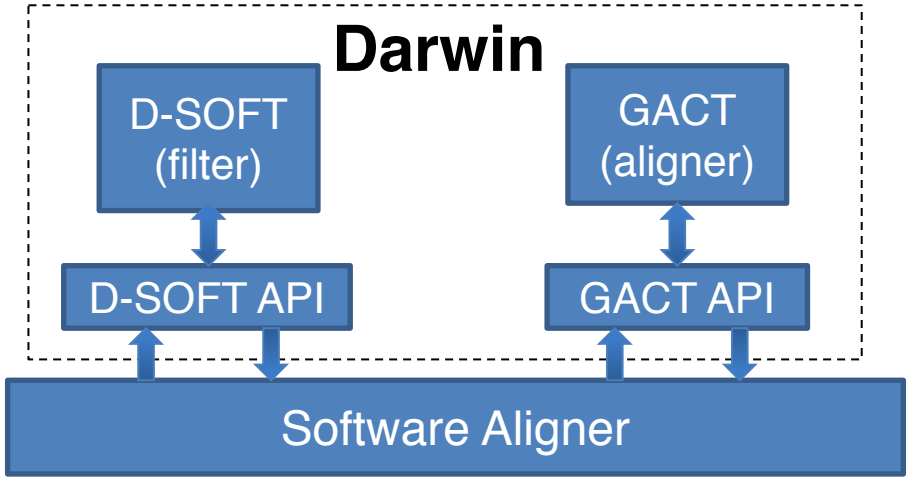
G	-	C	G	A	C	T	T	T
G	T	C	G	-	-	T	T	T

Score = 9

Sections

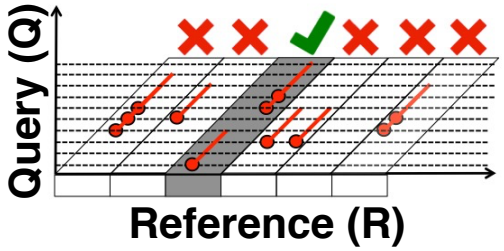
1. Introduction: DNA Sequencing, Alignment and Long Read Assembly
2. **Darwin: Framework Overview and Design Philosophy**
3. D-SOFT: Algorithm and Hardware-Acceleration
4. GACT: Algorithm and Hardware-Acceleration
5. Experimental Methodology and Results

Darwin: Overview

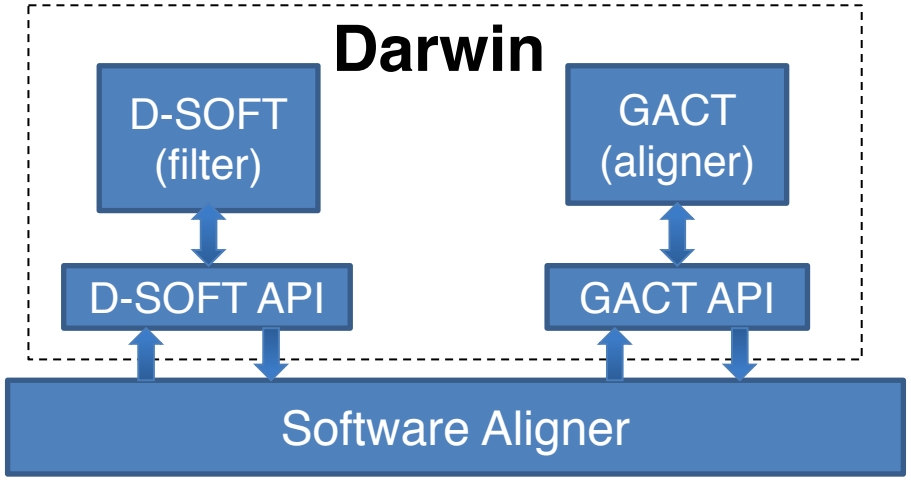


Darwin: Overview

D-SOFT

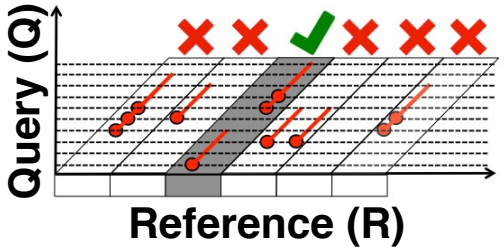


Up to **million fold** reduction in search space

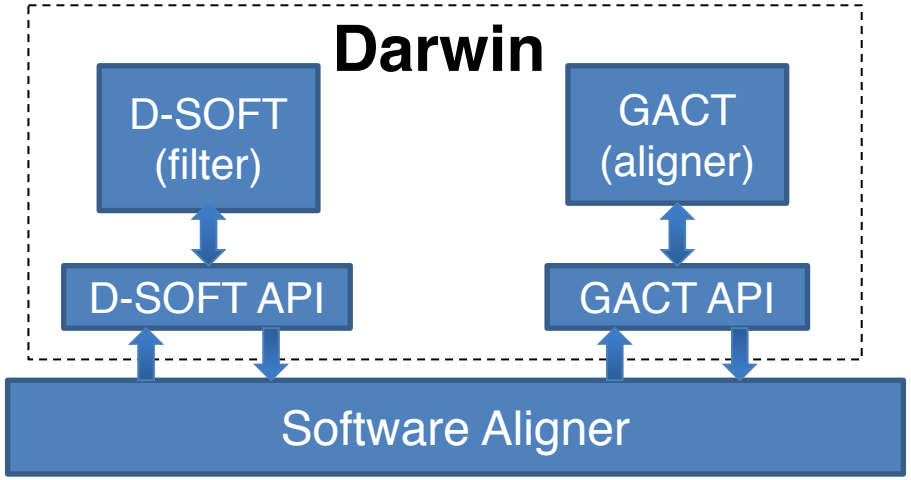


Darwin: Overview

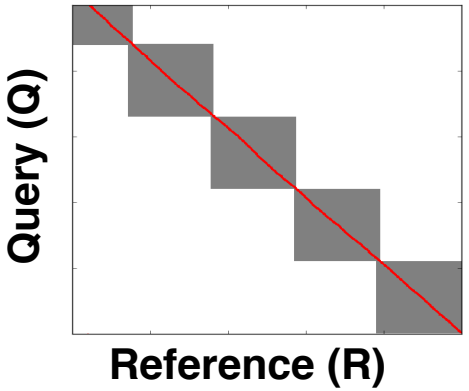
D-SOFT



Darwin

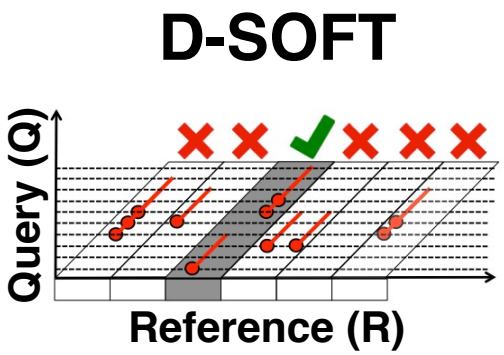


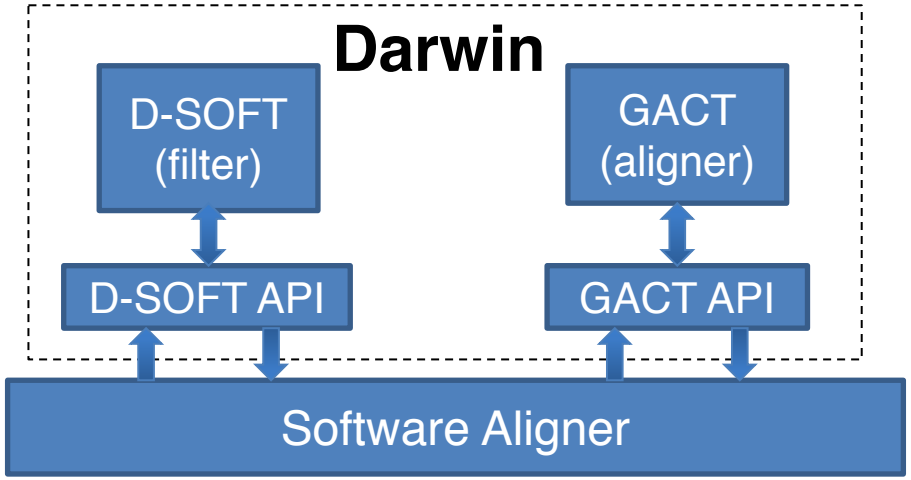
GACT

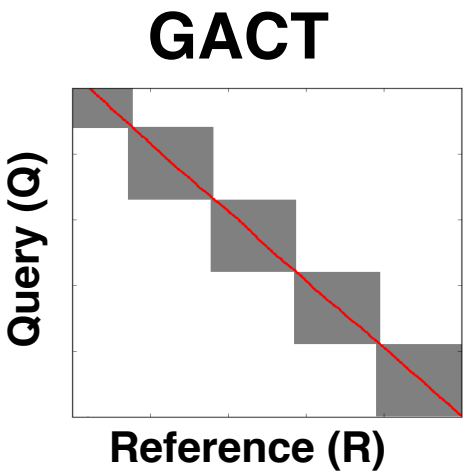


O(1) (time and space for compute-intensive step)
heuristic to Smith-Waterman

Darwin: Overview







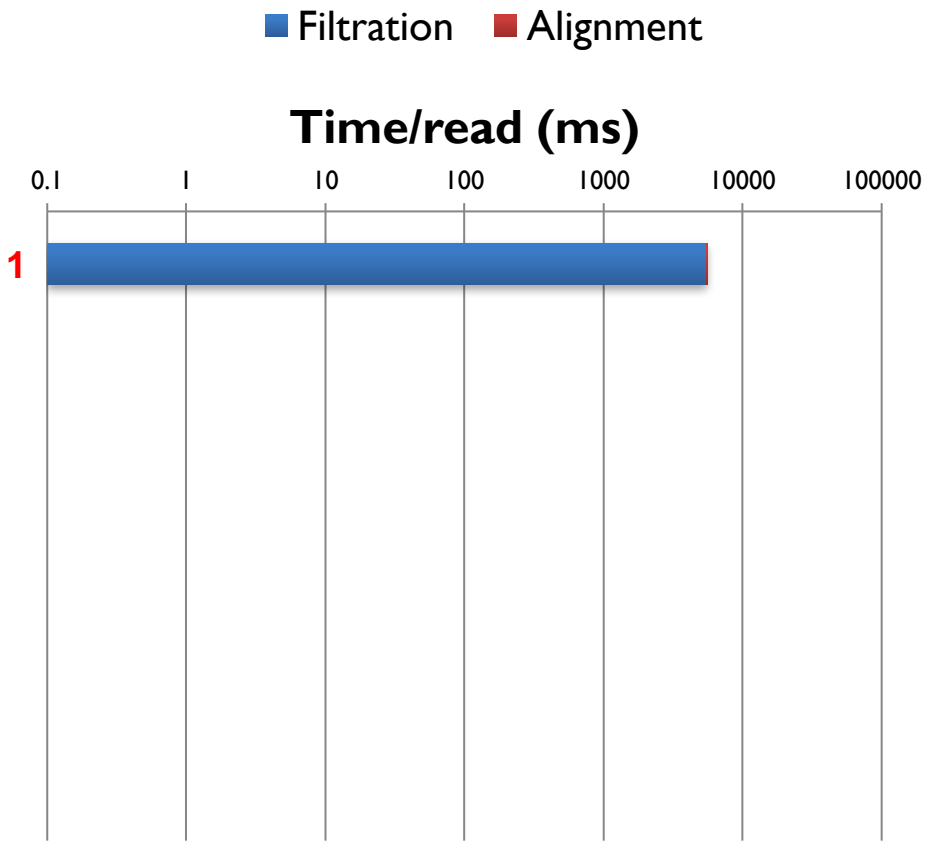
D-SOFT (Diagonal-band Seed Overlap based Filtration Technique):

- Tunable speed/precision to match any error profile
- Up to 85X speedup on ASIC

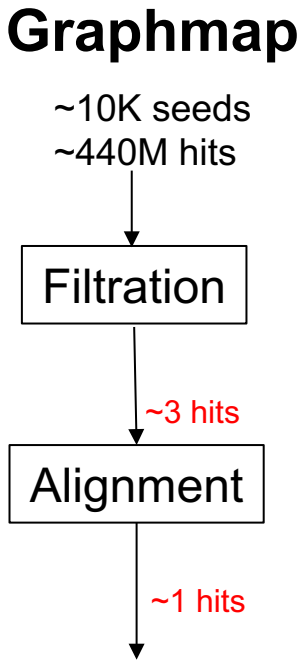
GACT (Genome Alignment using Constant-Memory Trace-back):

- Arbitrarily long alignments using constant on-chip memory (novel)
- Empirically optimal
- Up to 80,000X speedup on ASIC

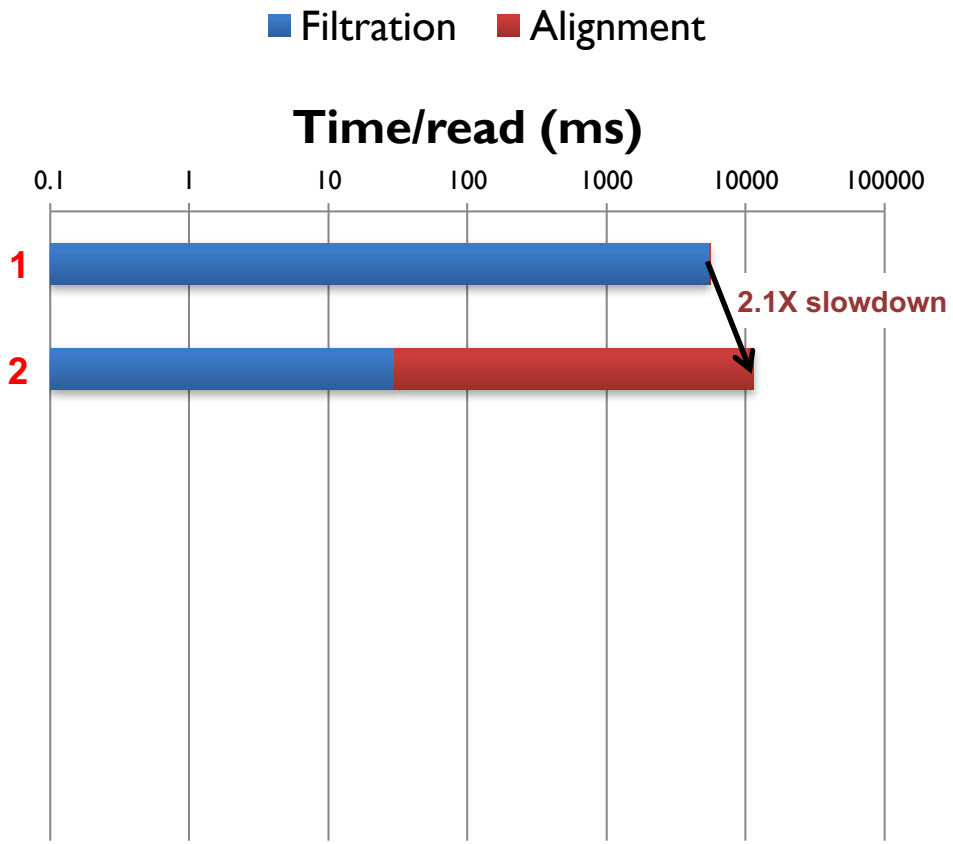
Evolution of Darwin (ASIC) from Graphmap



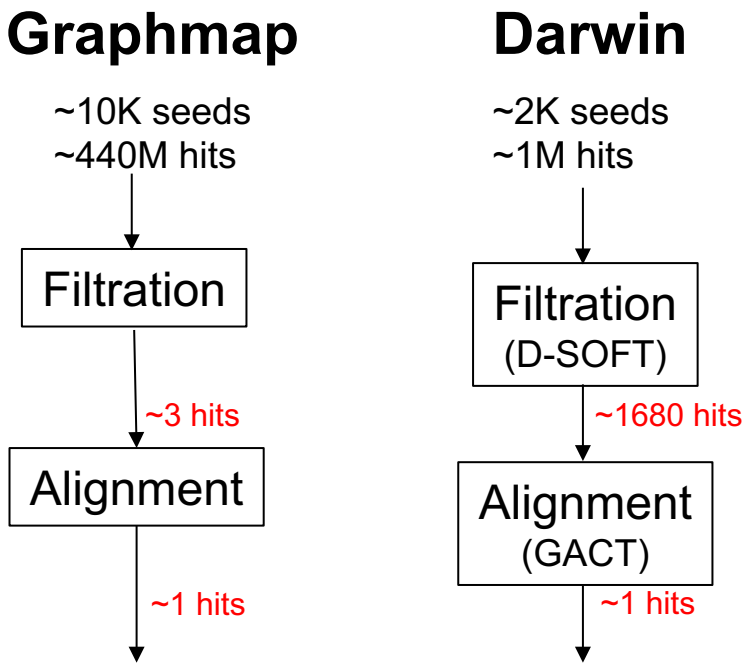
1. Graphmap (software)



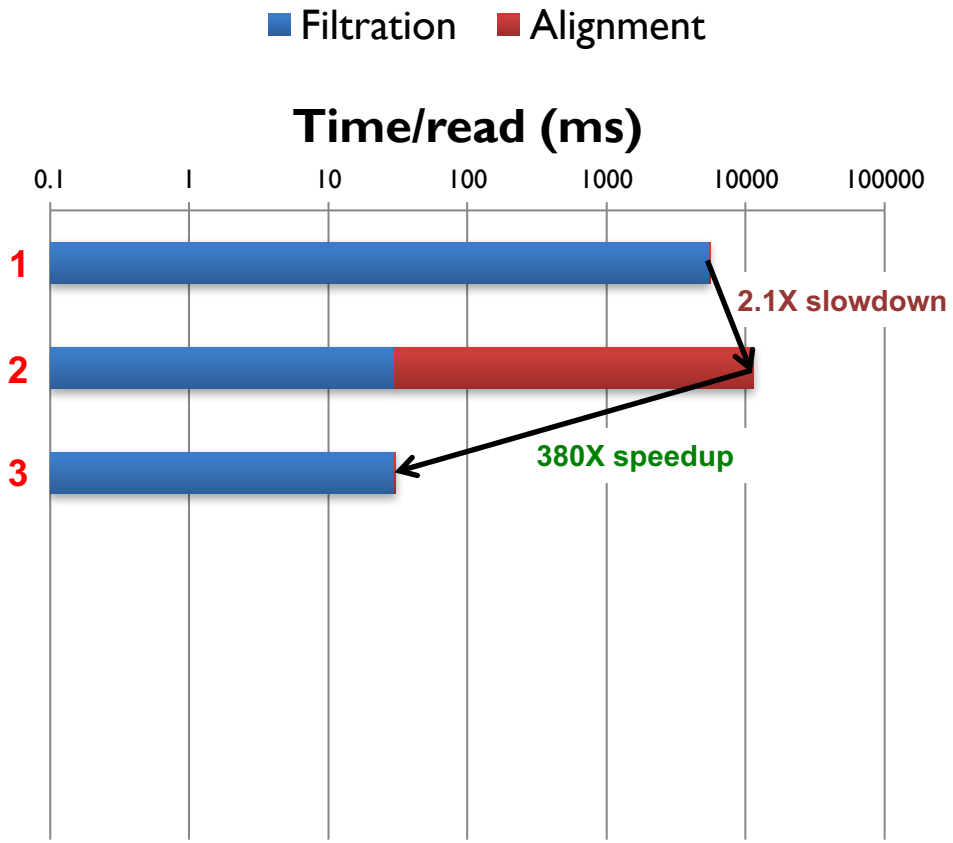
Evolution of Darwin (ASIC) from Graphmap



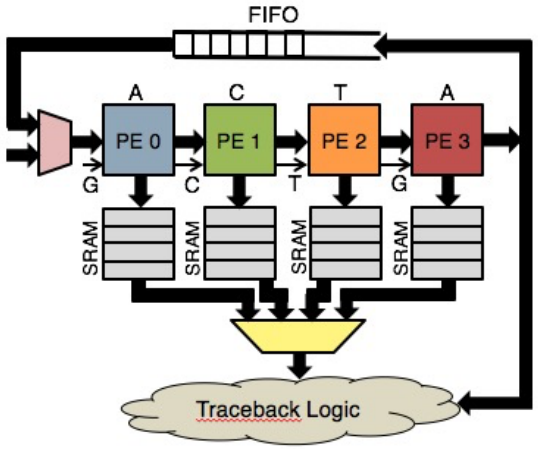
1. Graphmap (software)
2. Replace by D-SOFT and GACT (software)



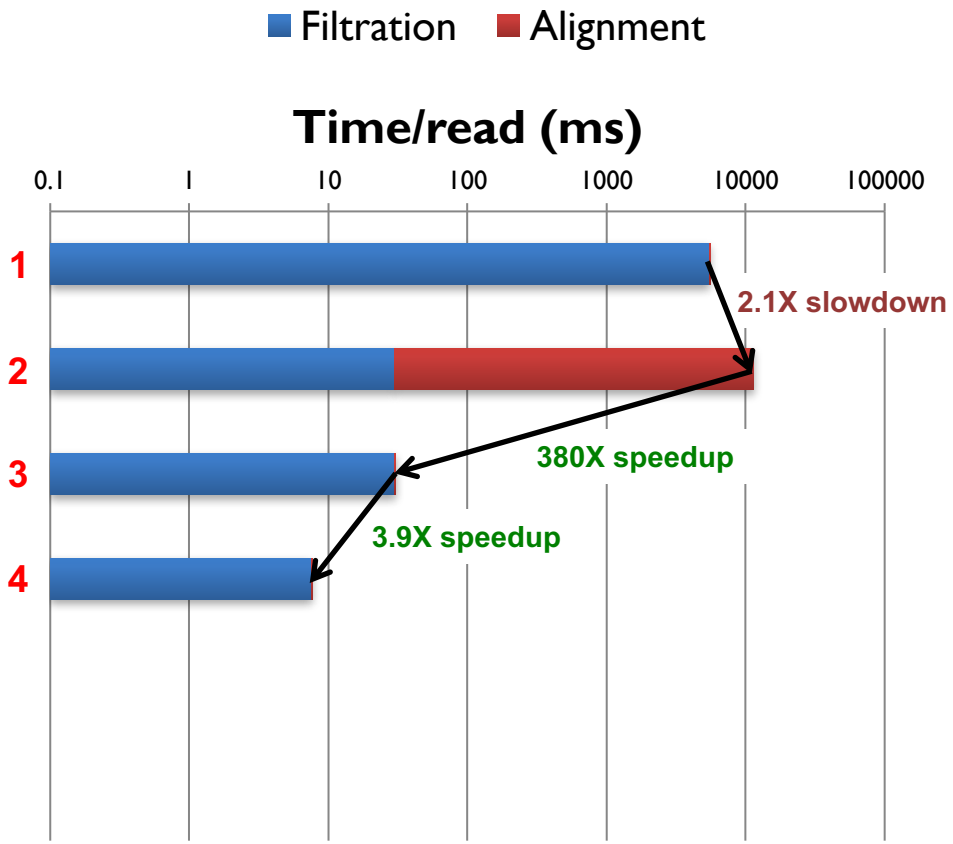
Evolution of Darwin (ASIC) from Graphmap



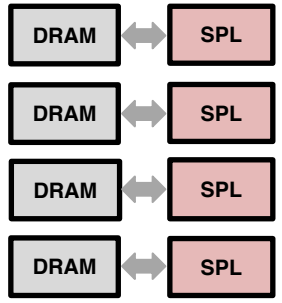
1. Graphmap (software)
2. Replace by D-SOFT and GACT (software)
3. GACT hardware-acceleration



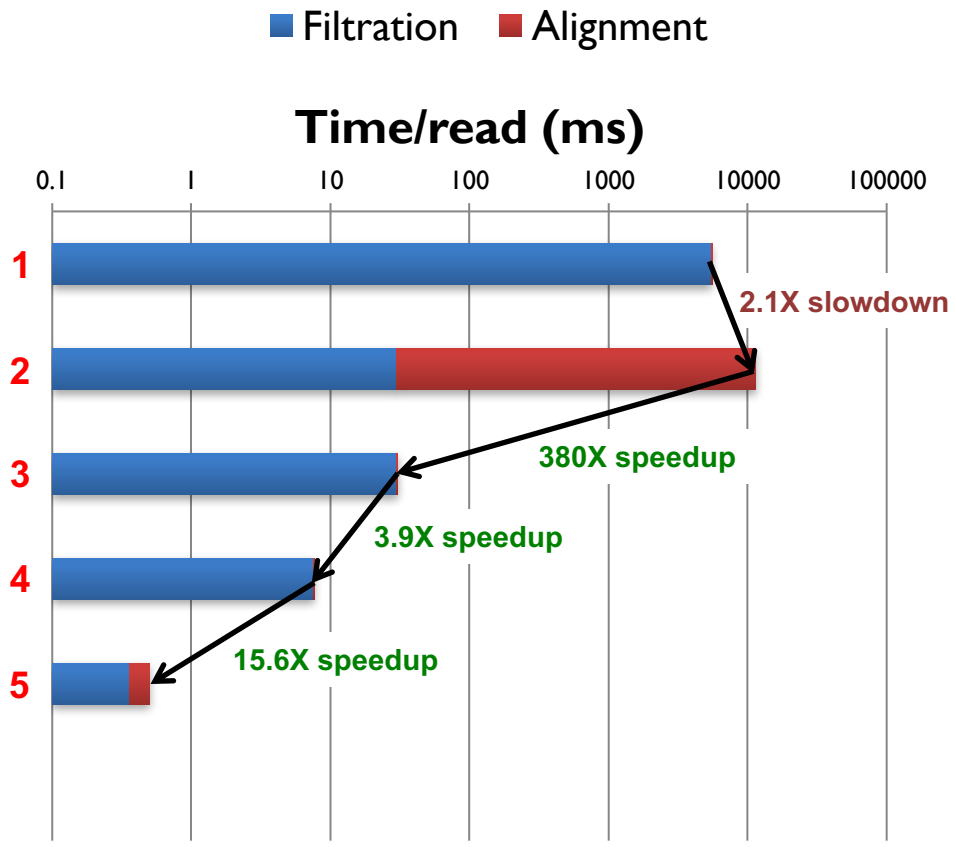
Evolution of Darwin (ASIC) from Graphmap



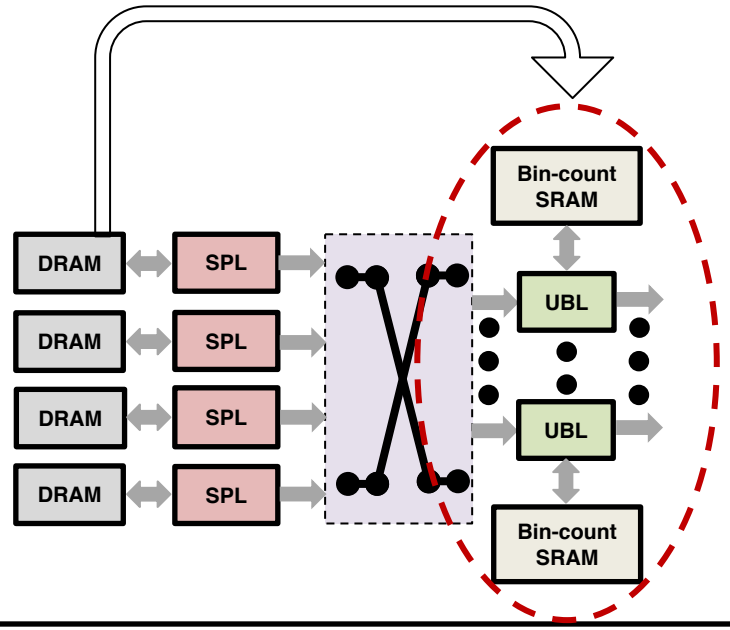
1. Graphmap (software)
2. Replace by D-SOFT and GACT (software)
3. GACT hardware-acceleration
4. Four DRAM channels for D-SOFT



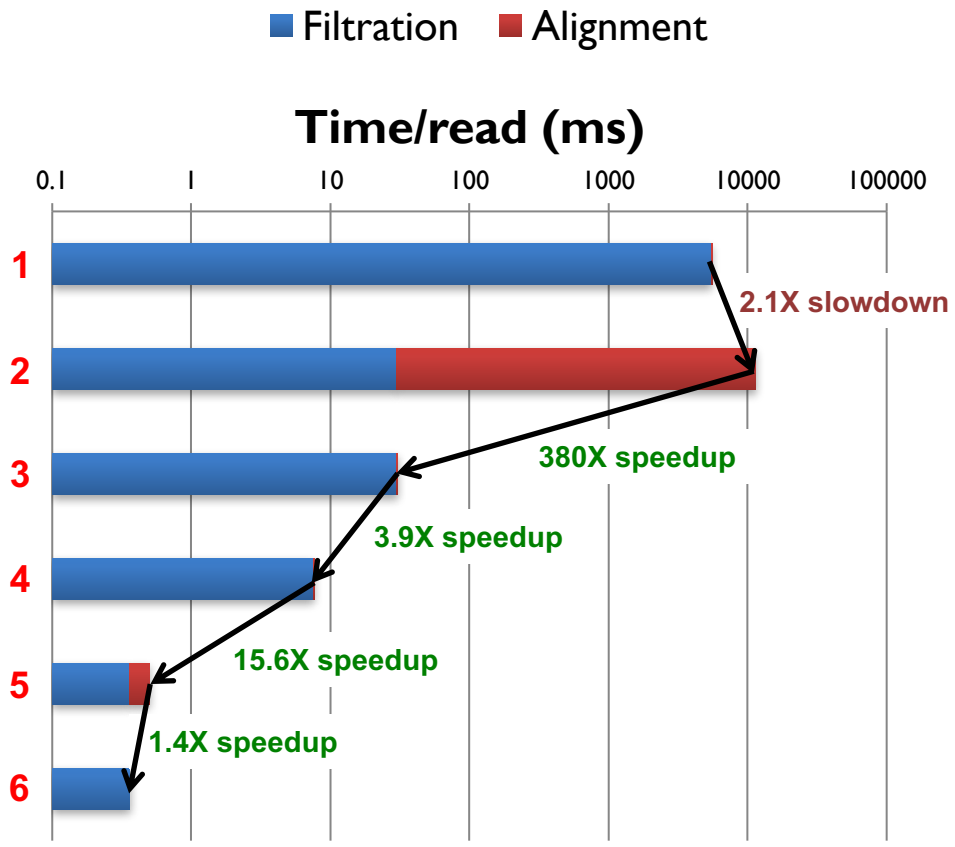
Evolution of Darwin (ASIC) from Graphmap



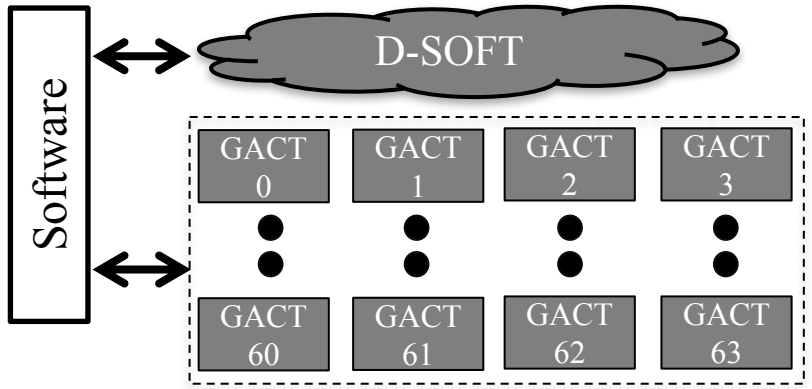
1. Graphmap (software)
2. Replace by D-SOFT and GACT (software)
3. GACT hardware-acceleration
4. Four DRAM channels for D-SOFT
5. Use on-chip memory to store random access data structures in D-SOFT



Evolution of Darwin (ASIC) from Graphmap



1. Graphmap (software)
2. Replace by D-SOFT and GACT (software)
3. GACT hardware-acceleration
4. Four DRAM channels for D-SOFT
5. Use on-chip memory to store random access data structures in D-SOFT
6. Pipeline D-SOFT and GACT



Sections

1. Introduction: DNA Sequencing, Alignment and Long Read Assembly
2. Darwin: Framework Overview and Design Philosophy
3. **D-SOFT: Algorithm and Hardware-Acceleration**
4. GACT: Algorithm and Hardware-Acceleration
5. Experimental Methodology and Results

D-SOFT: Data Structures

```
0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39
A  G  C  T  T  T  A  C  C  T  A  C  G  T  A  G  C  T  G  C  A  T  C  T  A  T  T  T  C  T  C  G  T  A  T  T  T  A  G  C
```

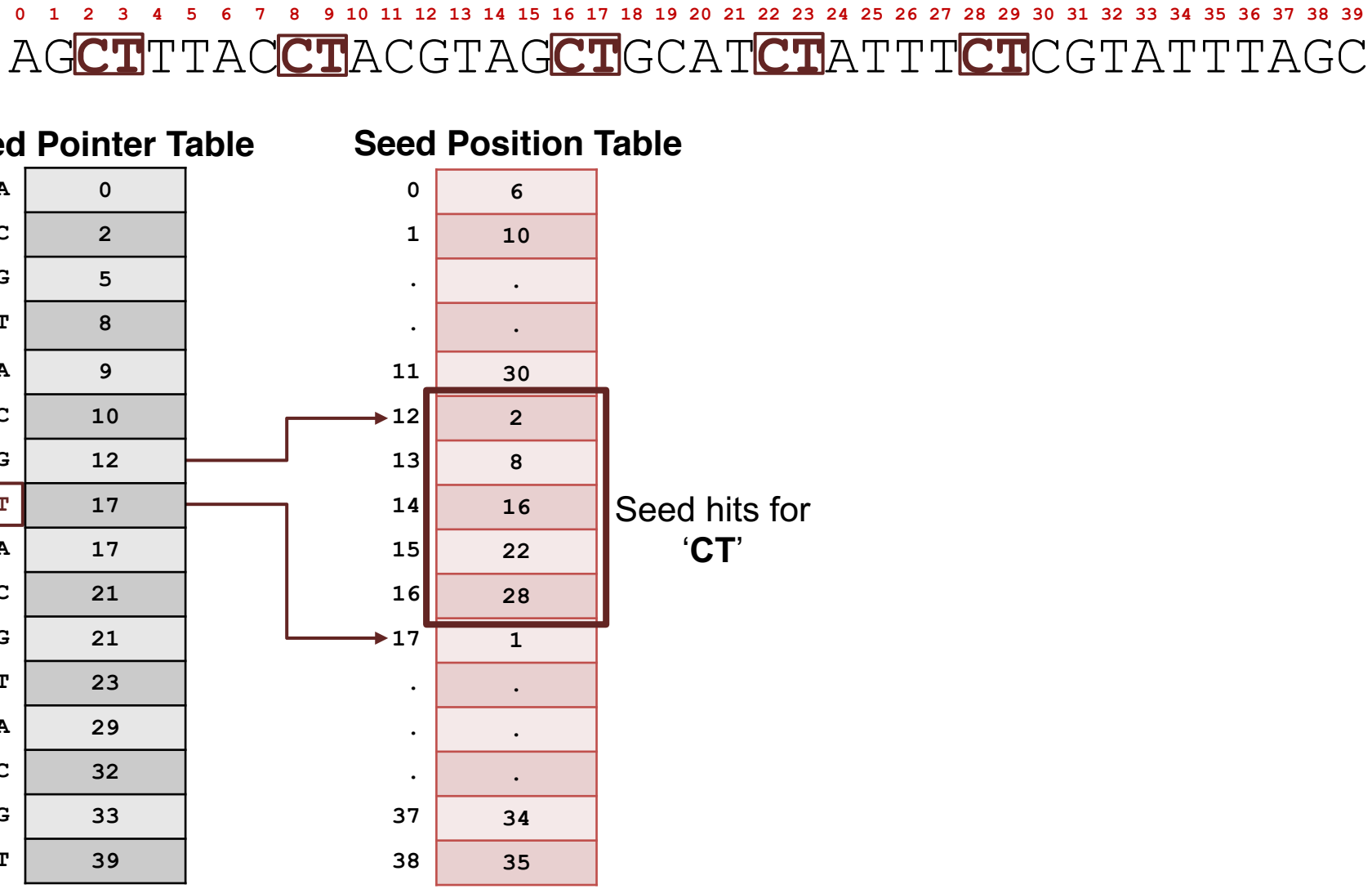
Seed Pointer Table

AA	0
AC	2
AG	5
AT	8
CA	9
CC	10
CG	12
CT	17
GA	17
GC	21
GG	21
GT	23
TA	29
TC	32
TG	33
TT	39

Seed Position Table

0	6
1	10
.	.
.	.
11	30
12	2
13	8
14	16
15	22
16	28
17	1
.	.
.	.
.	.
37	34
38	35

D-SOFT: Data Structures



D-SOFT: Data Structures

Seed Pointer Table

AA	0
AC	2
AG	5
AT	8
CA	9
CC	10
CG	12
CT	17
GA	17
GC	21
GG	21
GT	23
TA	29
TC	32
TG	33
TT	39

Seed Position Table

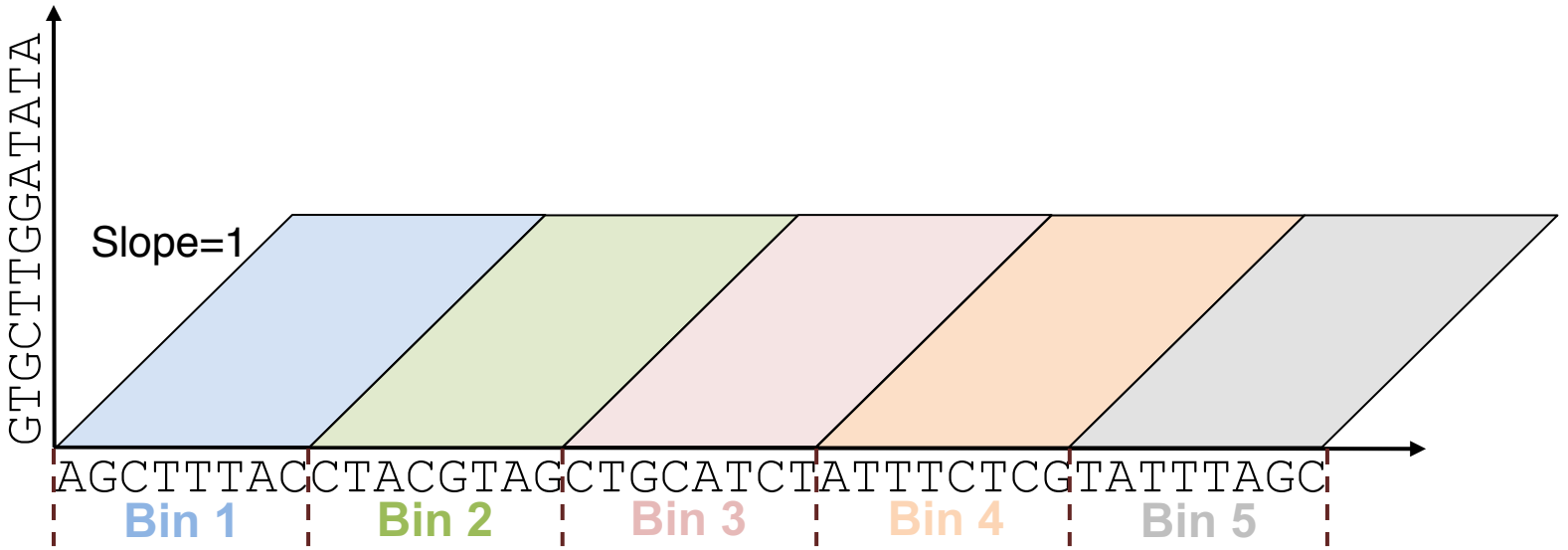
0	6
1	10
.	.
.	.
11	30
12	2
13	8
14	16
15	22
16	28
17	1
.	.
.	.
.	.
37	34
38	35

Seed hits for '**CT**'

3-step construction:

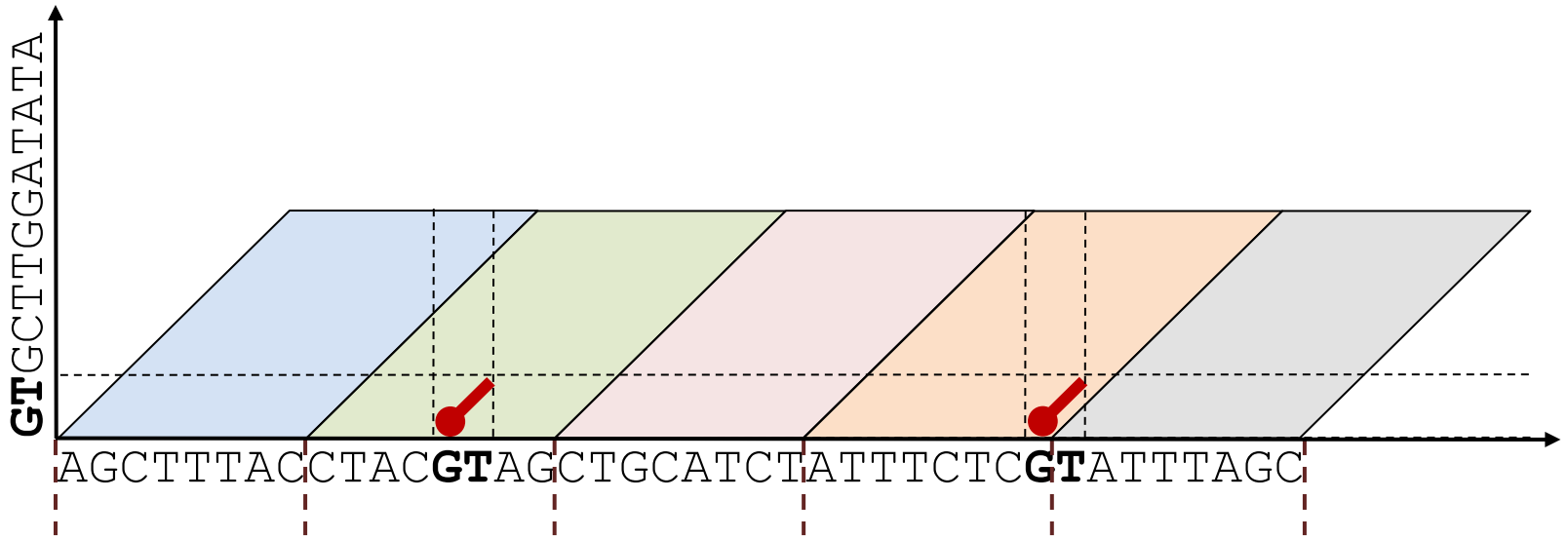
1. First pass on reference to record seed occurrence counts in seed pointer table
2. Cumulative add occurrence count to determine start index of each seed in position table
3. Second pass on reference to fill position table

D-SOFT: Algorithm Overview



Bin count (bases)	Last hit offset
0	-inf
0	-inf
0	-inf
0	-inf
0	-inf

D-SOFT: Algorithm Overview



Pointer Table

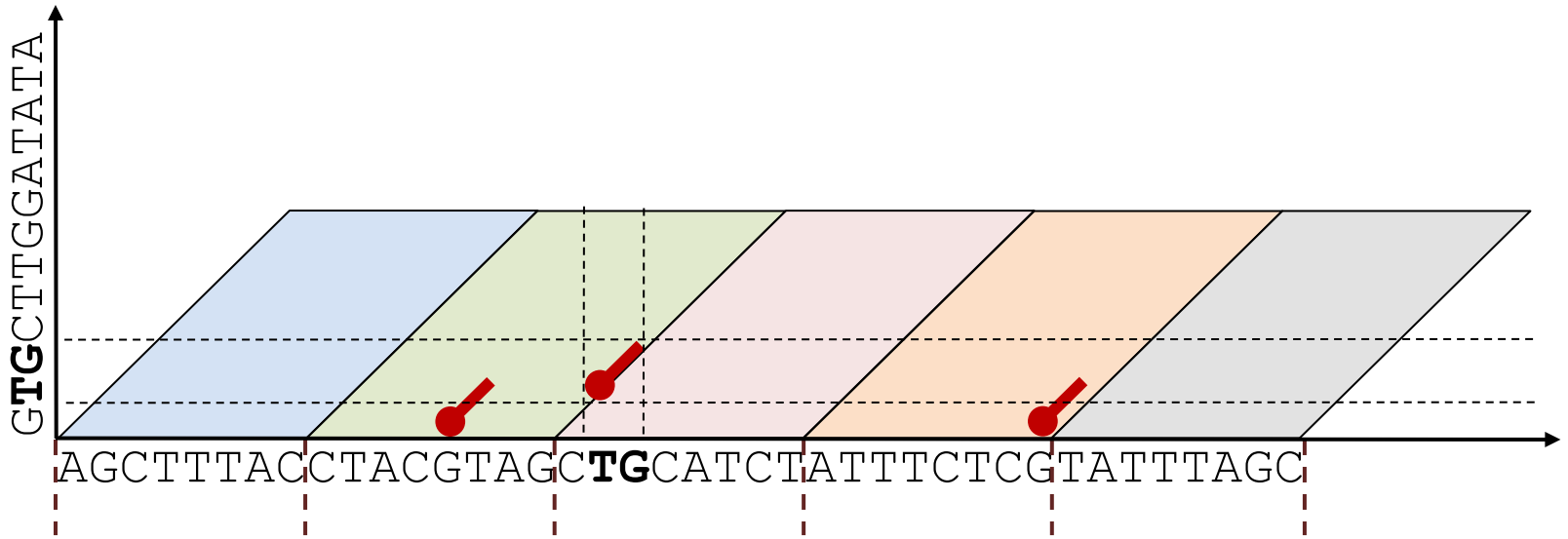
.	.
.	.
.	.
GG	21
GT	23
TA	29
.	.
.	.

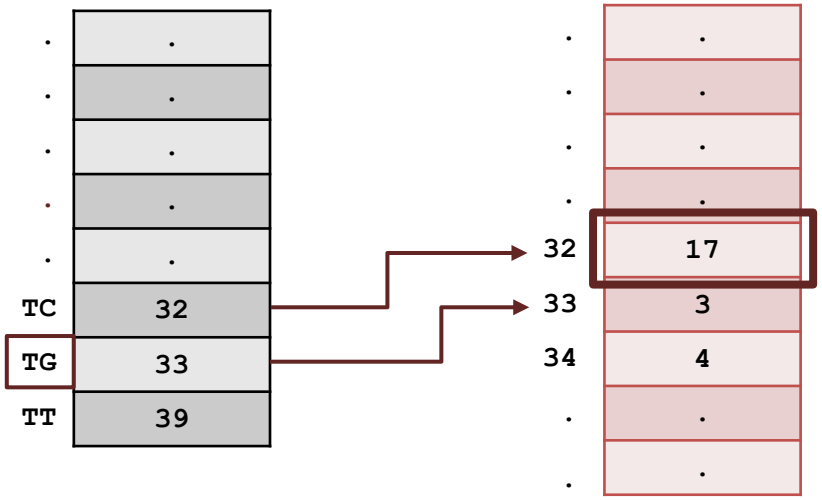
Position Table

.	.
.	.
20	38
21	12
22	31
23	5
.	.
.	.
.	.

Bin count (bases)	Last hit offset
0	-inf
2	0
0	-inf
2	0
0	-inf

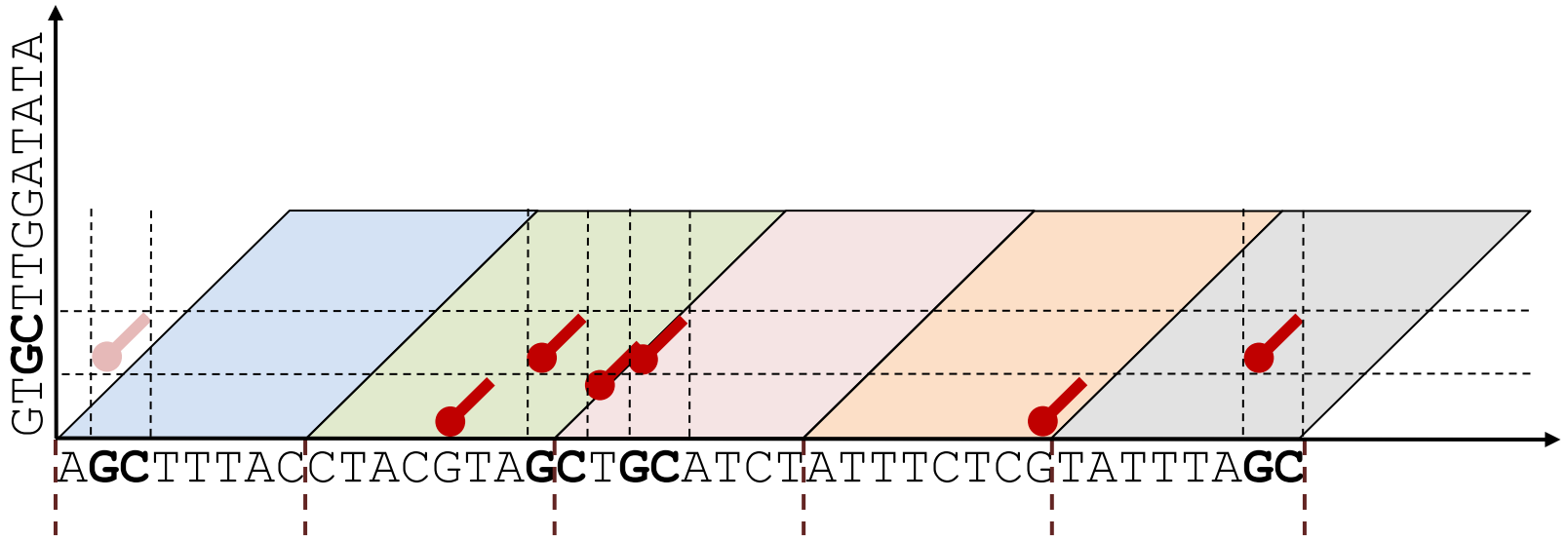
D-SOFT: Algorithm Overview





Bin count (bases)	Last hit offset
0	-inf
3	1
0	-inf
2	0
0	-inf

D-SOFT: Algorithm Overview



Pointer Table

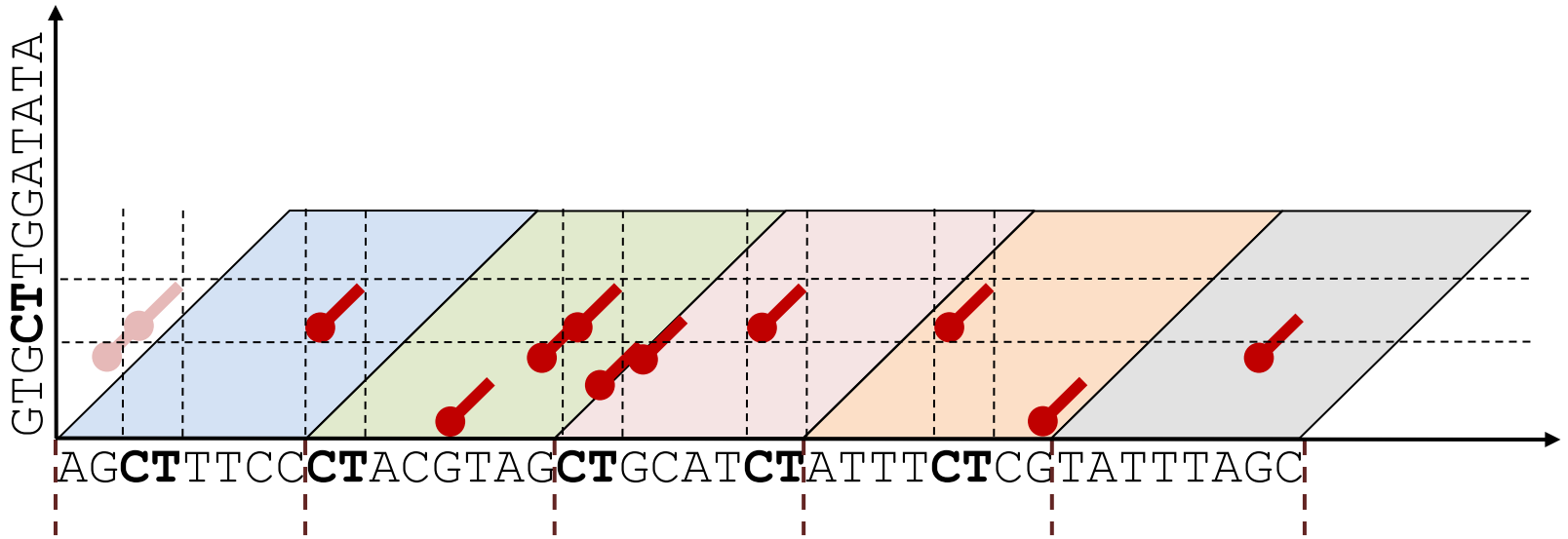
.	.
.	.
.	.
GA	17
GC	21
GG	21
.	.
.	.

Position Table

.	.
.	.
17	1
18	15
19	18
20	38
21	.
.	.
.	.

Bin count (bases)	Last hit offset
0	-inf
4	2
2	2
2	0
2	2

D-SOFT: Algorithm Overview



Pointer Table

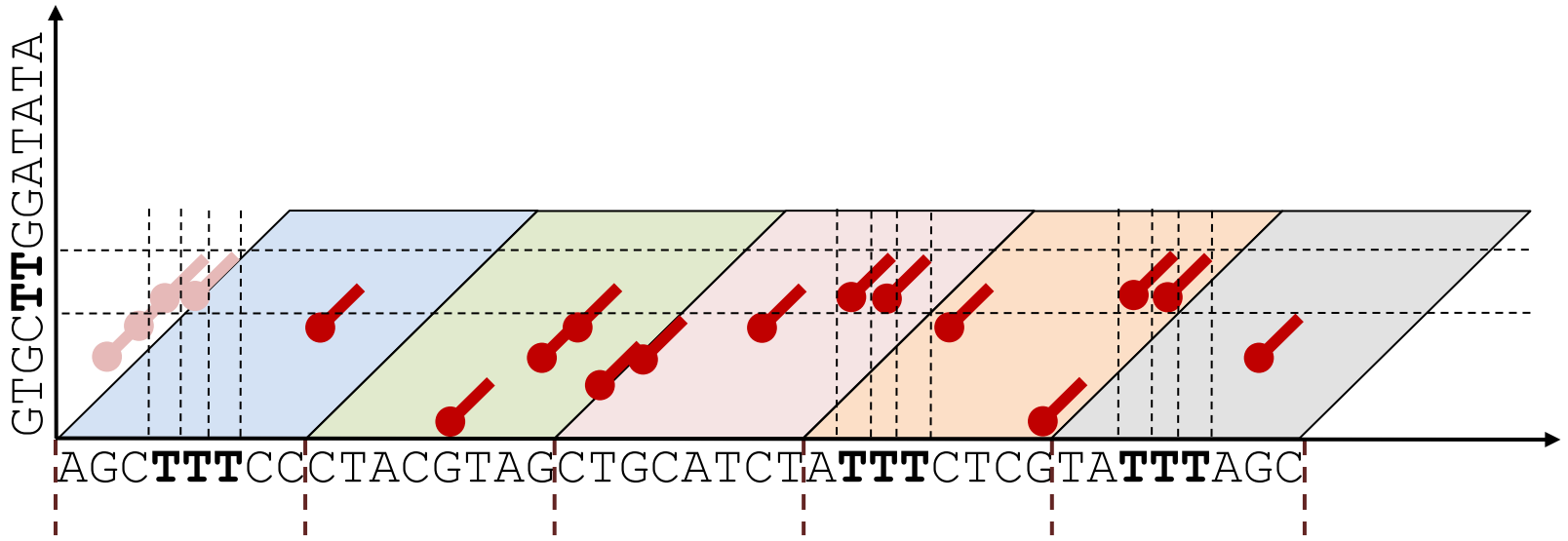
.	.
.	.
CG	12
CT	17
GA	17
.	.
.	.
.	.

Position Table

.	.
12	2
13	8
14	16
15	22
16	28
17	1
.	.
.	.

Bin count (bases)	Last hit offset
2	3
5	3
3	3
4	3
2	2

D-SOFT: Algorithm Overview



Pointer Table

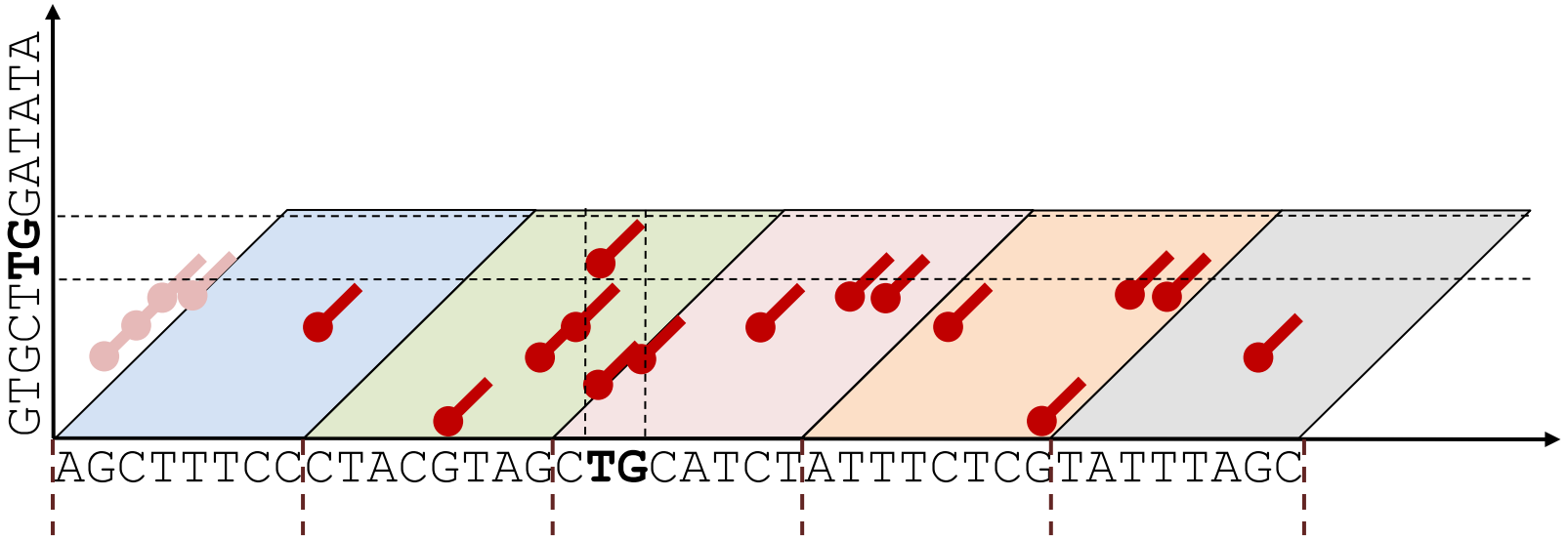
.	.
.	.
.	.
.	.
.	.
TC	32
TG	33
TT	39

Position Table

.	.
.	.
.	.
33	3
34	4
35	25
36	26
37	34
38	35

Bin count (bases)	Last hit offset
2	3
5	3
4	4
5	4
2	2

D-SOFT: Algorithm Overview



Pointer Table

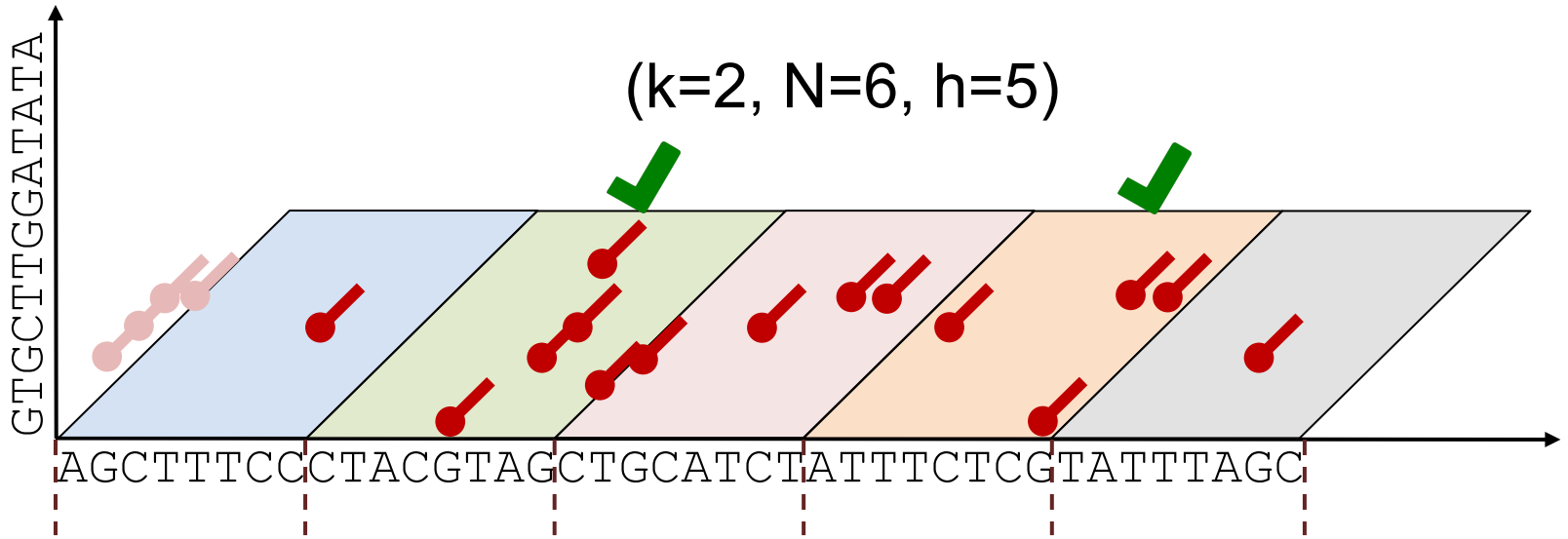
.	.
.	.
.	.
.	.
.	.
TC	32
TG	33
TT	39

Position Table

.	.
.	.
.	.
.	.
32	17
33	3
34	4
.	.
.	.

Bin count (bases)	Last hit offset
2	3
7	5
4	4
5	4
2	2

D-SOFT: Algorithm Overview



Parameters:

k: seed size

N: number of seeds

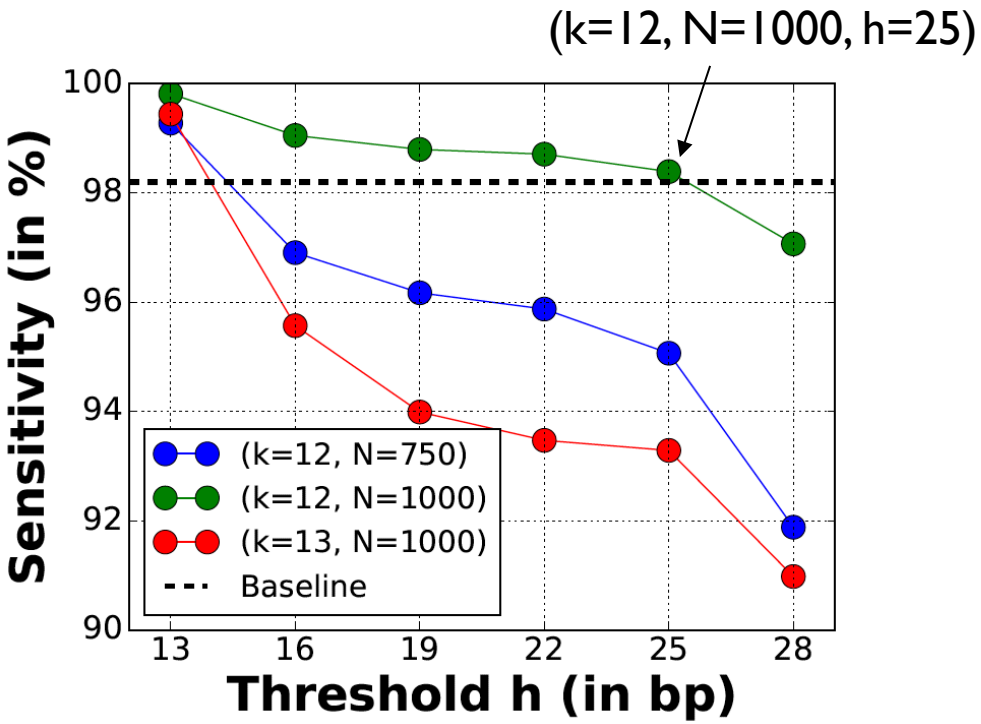
h: threshold on non-overlapping bases

B: bin size (number of bases, fixed to 128 in Darwin)

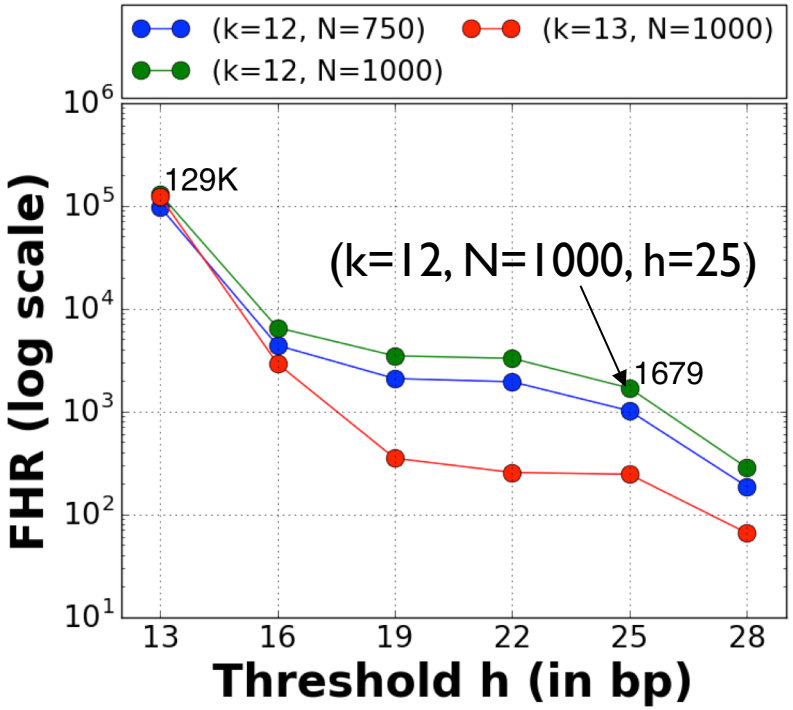
Bin count (bases)	Last hit offset
2	3
7	5
4	4
5	4
2	2

D-SOFT has tunable Speed/Precision

Sensitivity



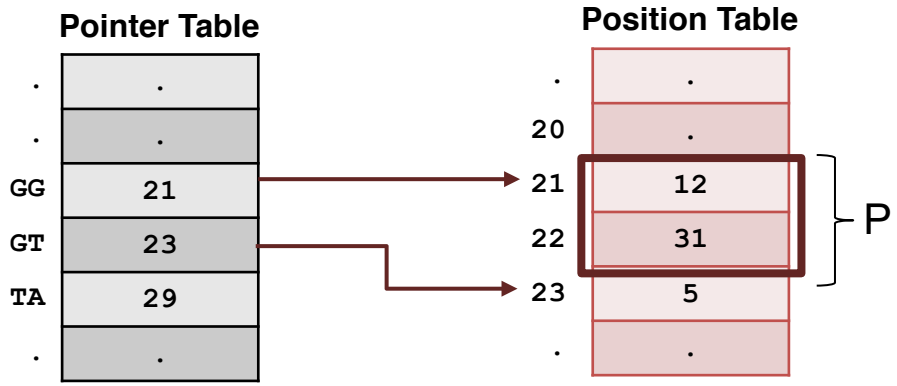
False Hit Rate



Data shown for ONT_2D reads with GraphMap used for Baseline sensitivity

D-SOFT Algorithm: Key Points

1. Seed tables are large (~16GB for human genome) and have mostly sequential memory accesses (fast)
   - ➢ Use multiple DRAM channels for ASIC

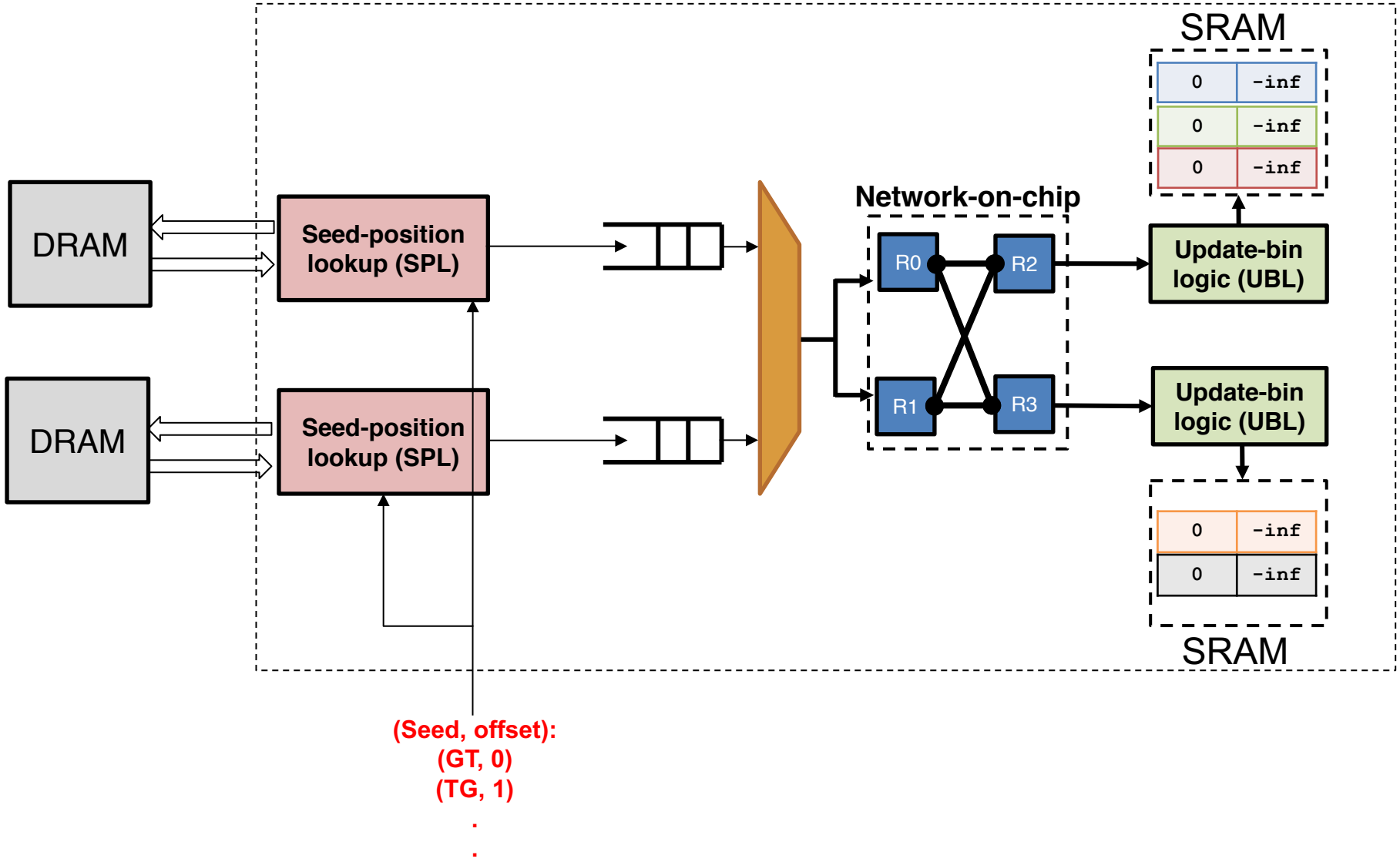


2. Bin-offset table is relatively small (~50-64MB for human genome) but have random access pattern (slow)
   - ➢ Can use on-chip memory for speedup

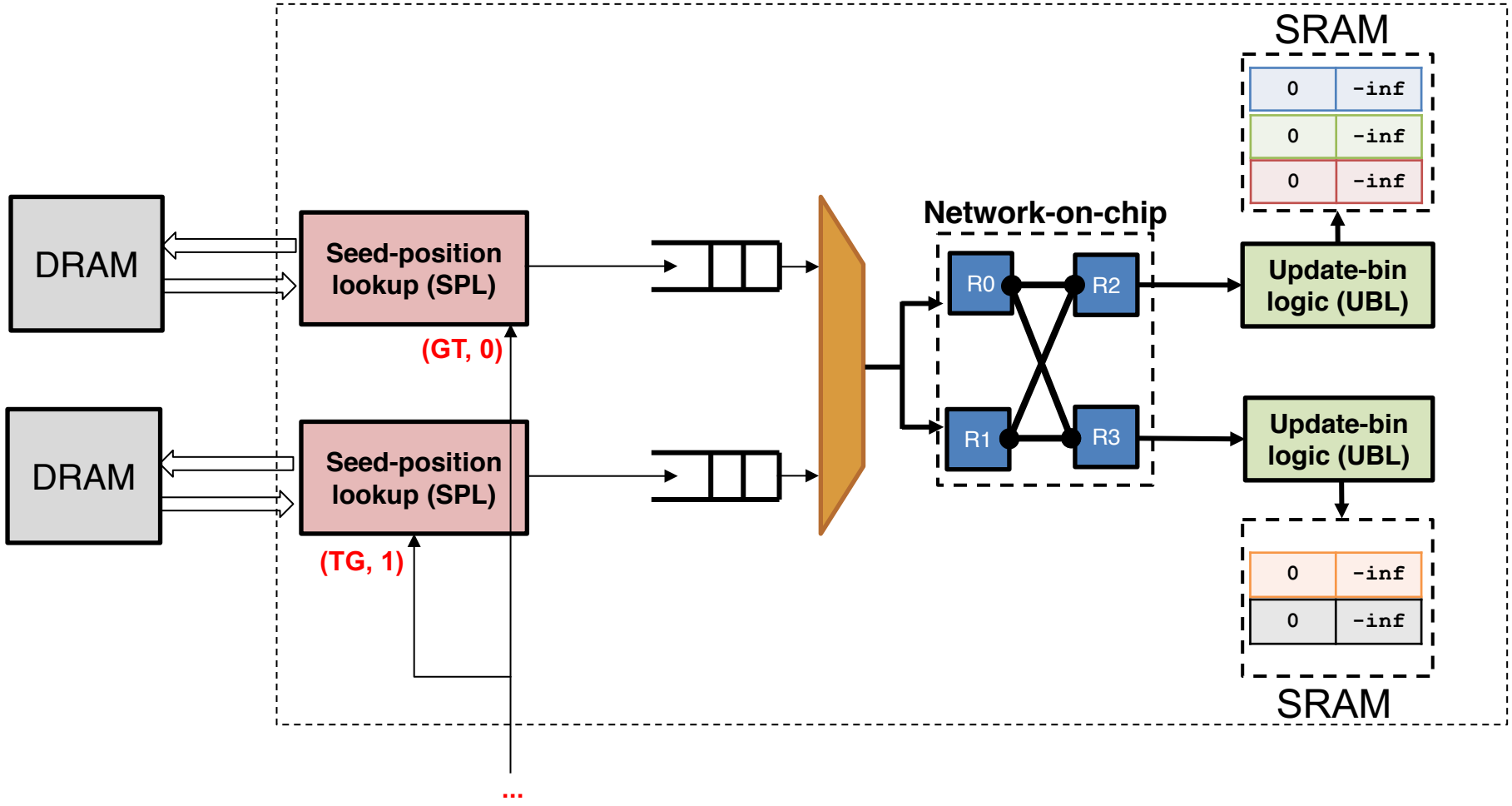
2P random memory accesses (read+write)

Bin count (bases)	Last hit offset
0	-inf
2	0
0	-inf
2	0
0	-inf

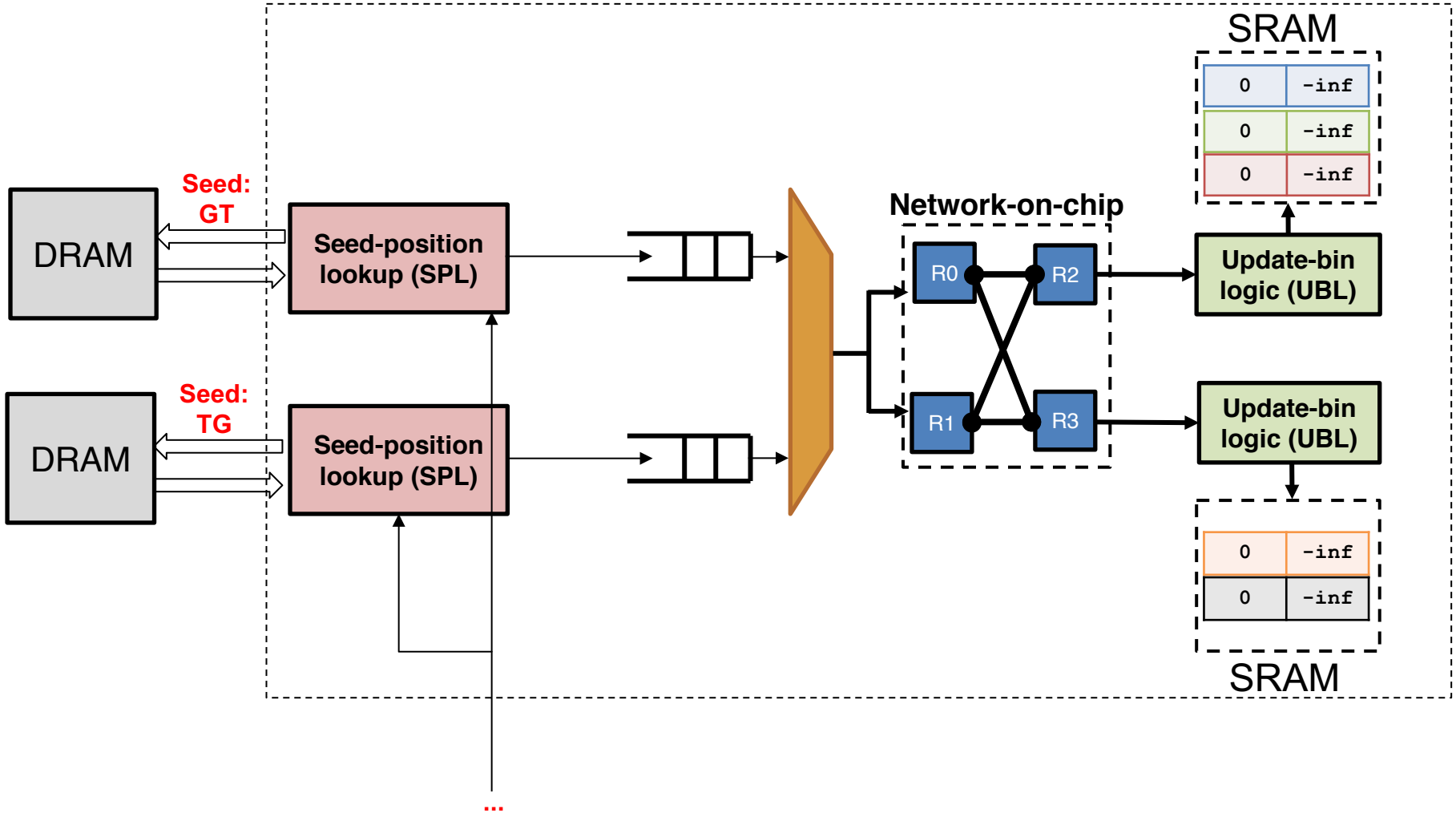
D-SOFT: Hardware-acceleration



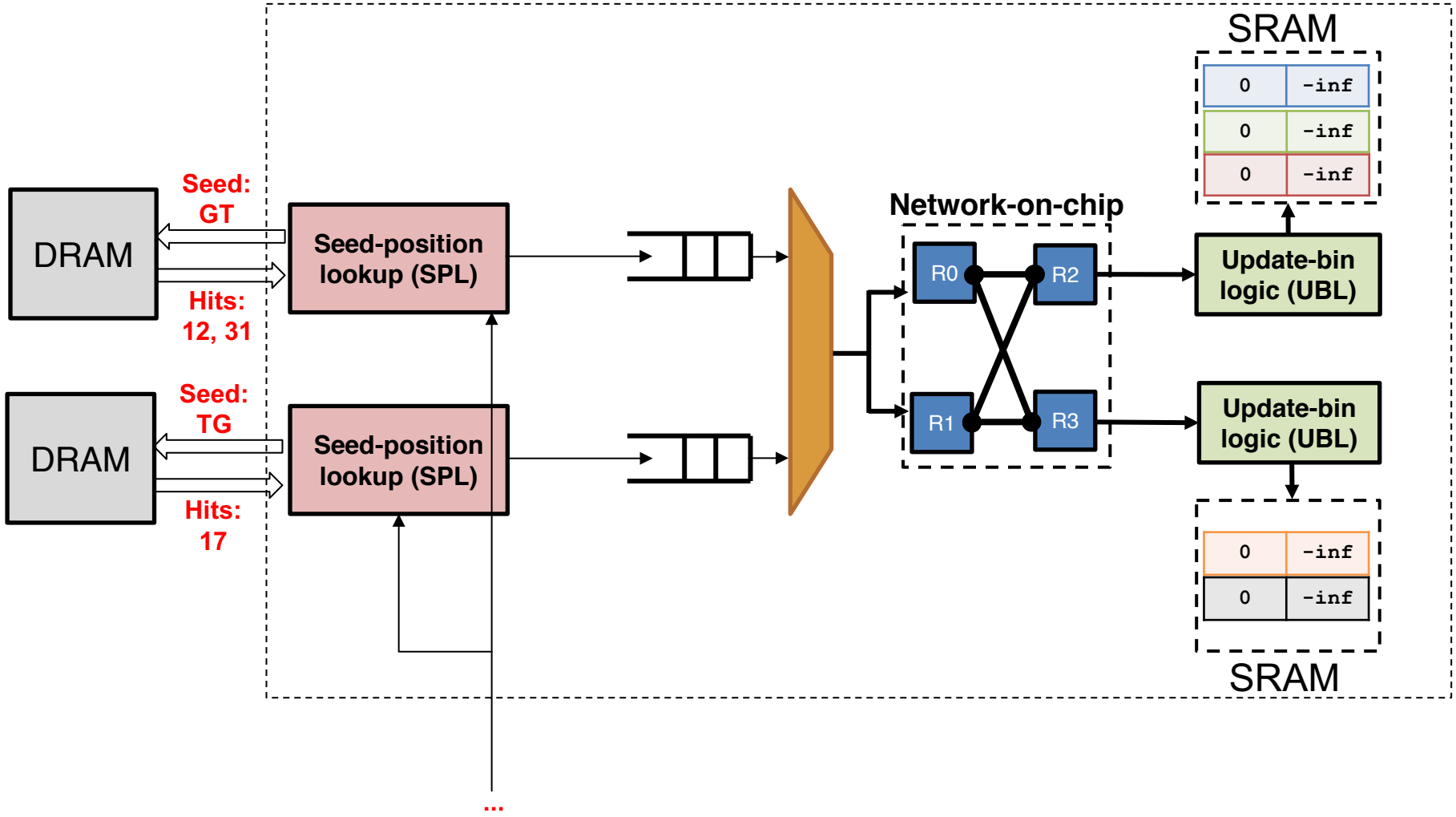
D-SOFT: Hardware-acceleration



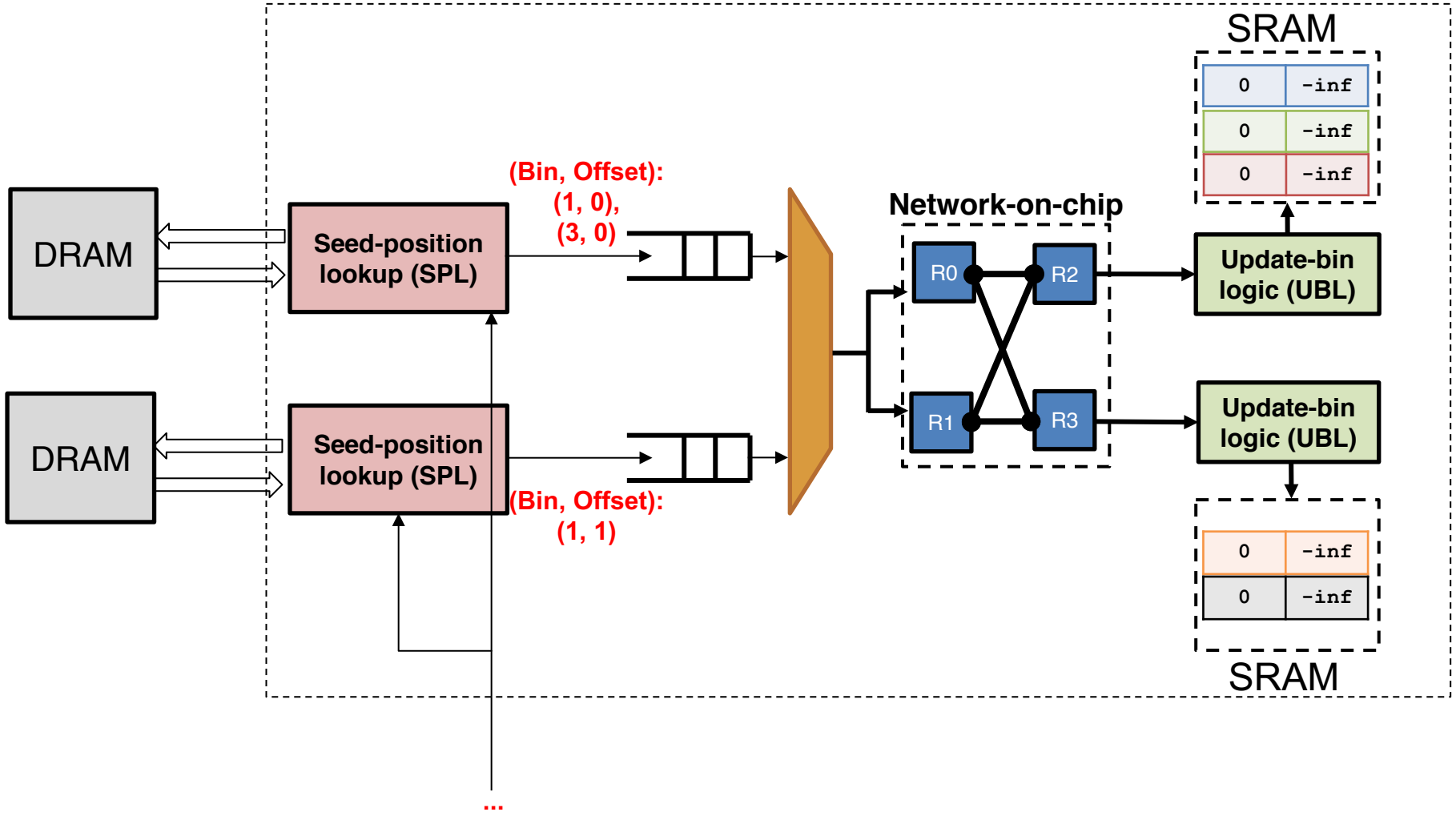
D-SOFT: Hardware-acceleration



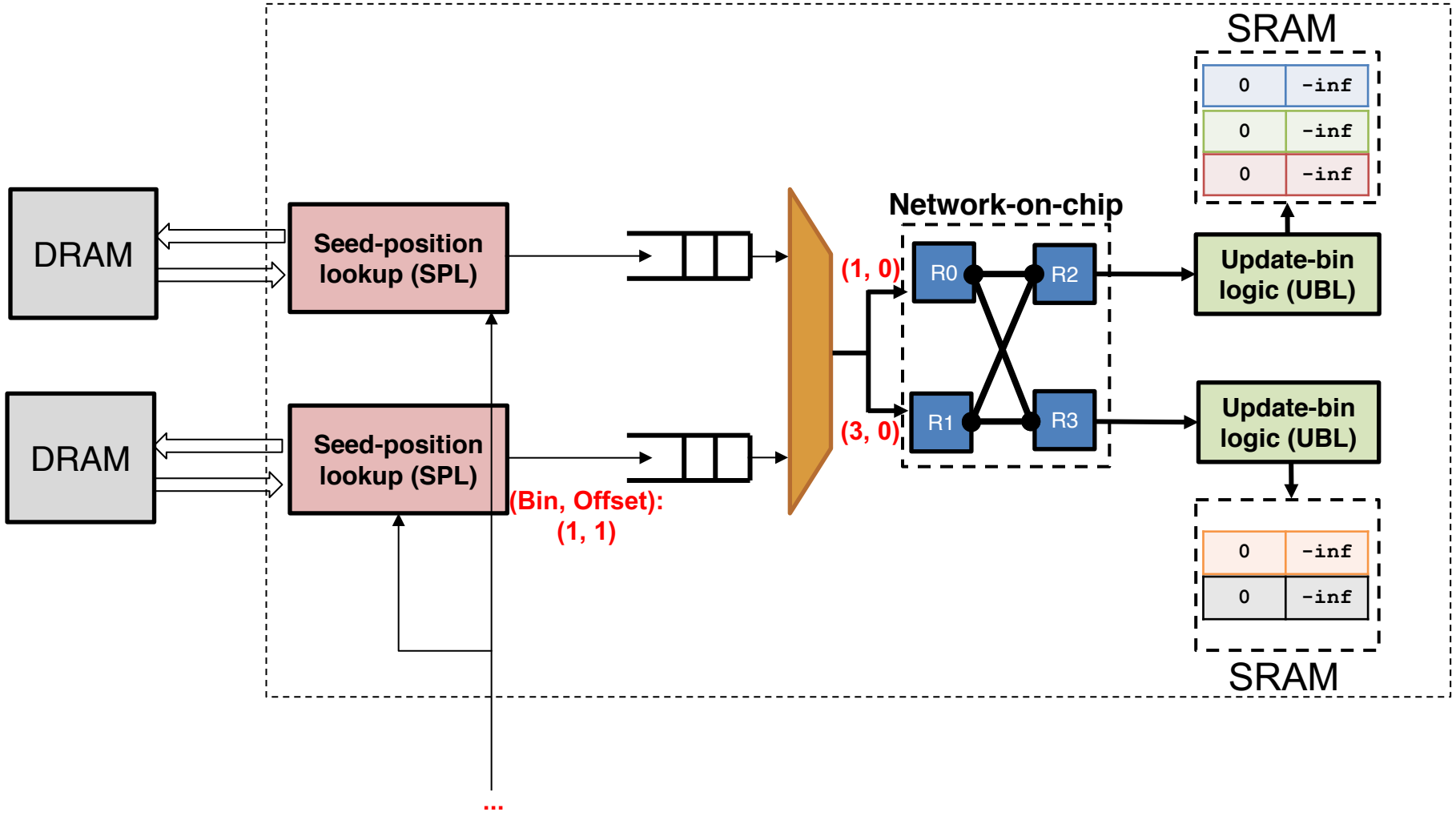
D-SOFT: Hardware-acceleration



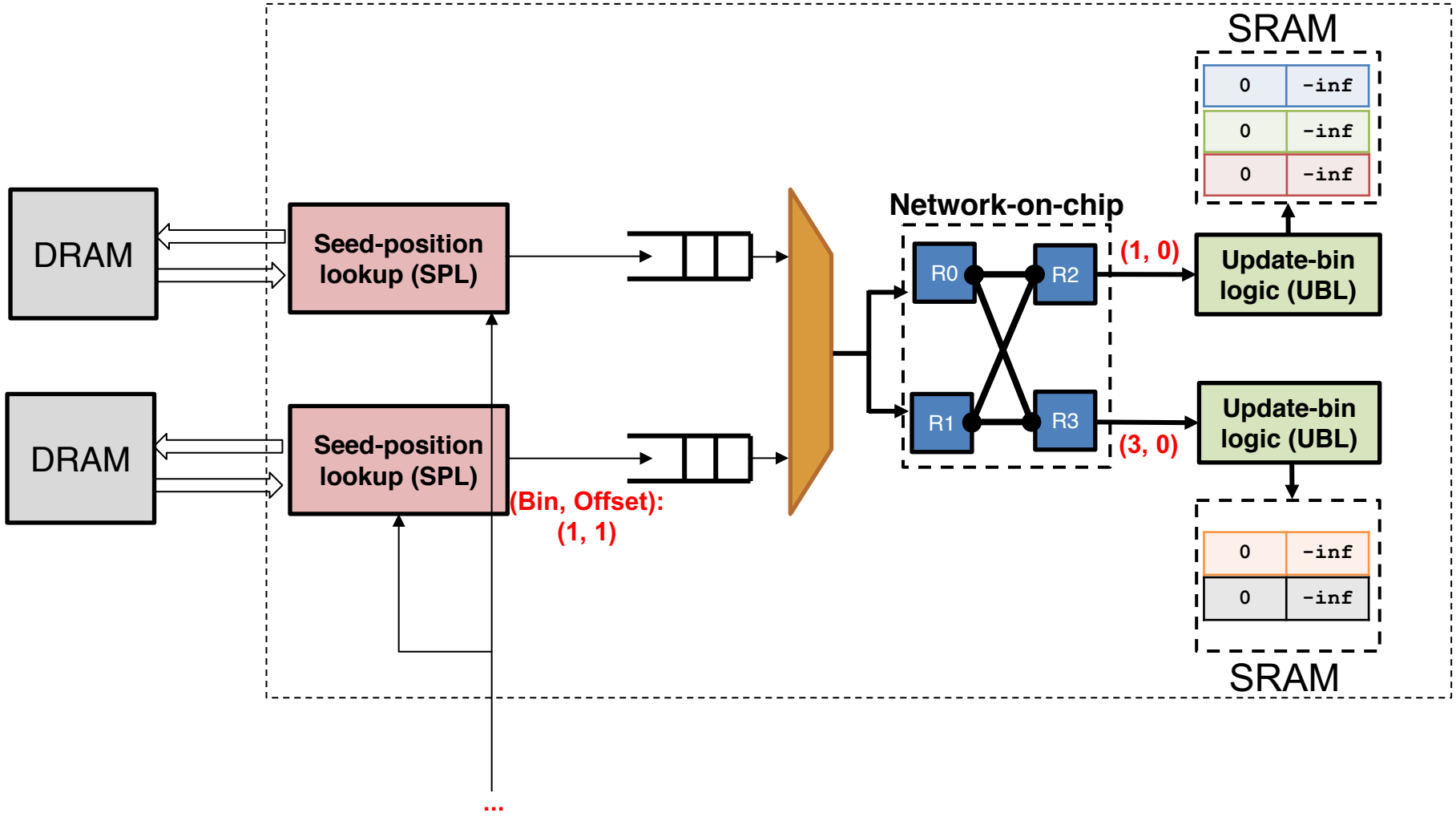
D-SOFT: Hardware-acceleration



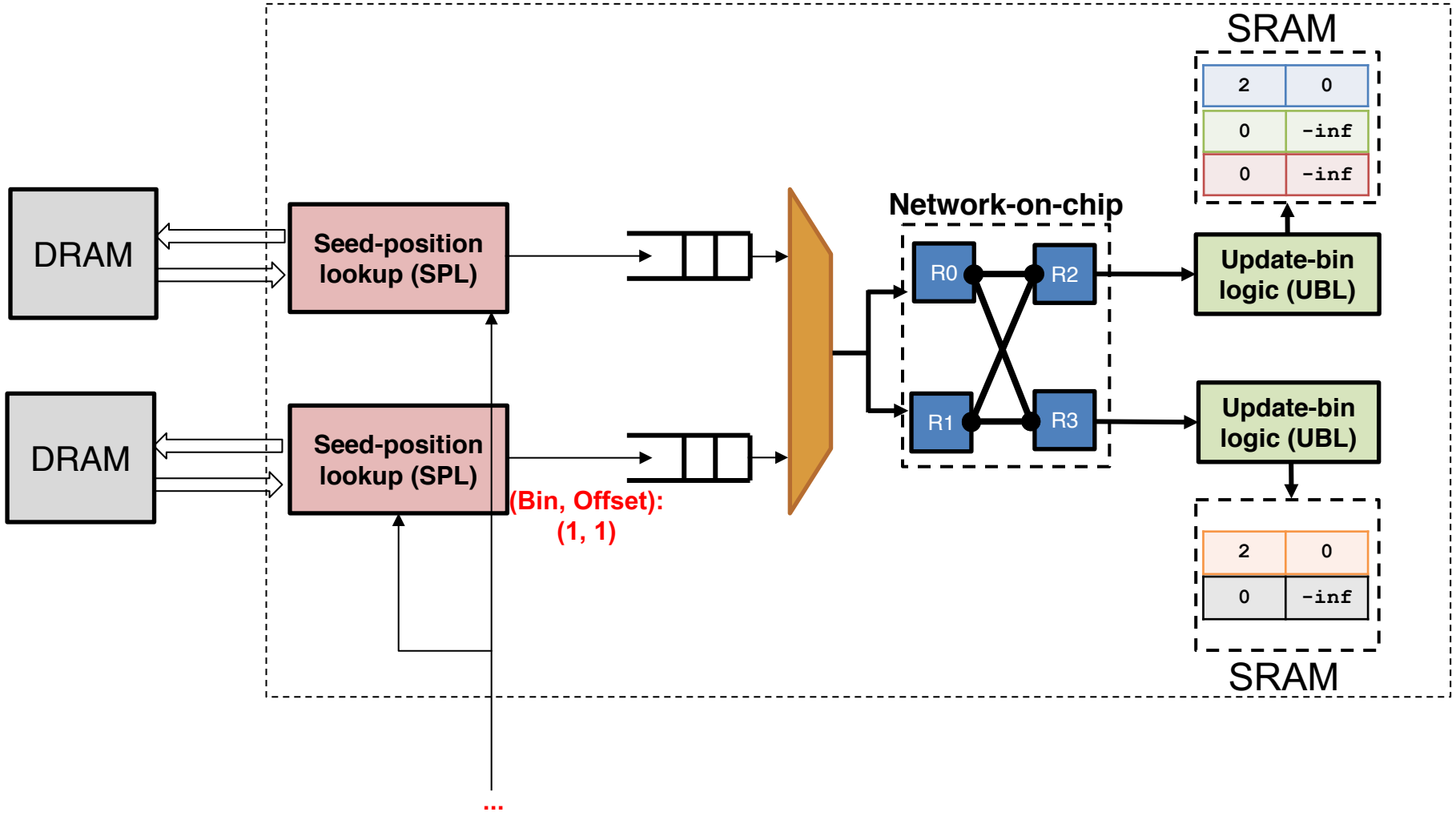
D-SOFT: Hardware-acceleration



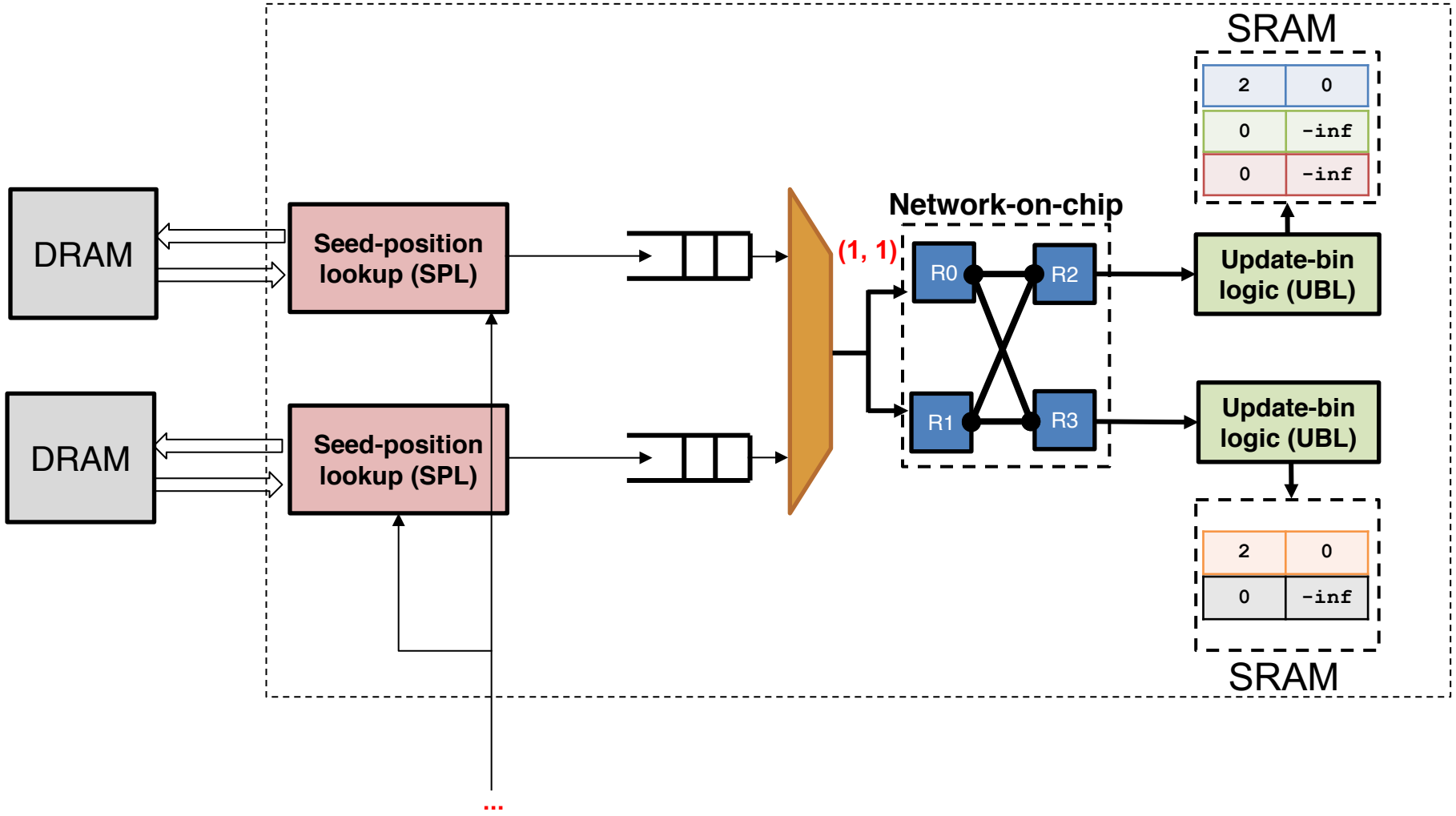
D-SOFT: Hardware-acceleration



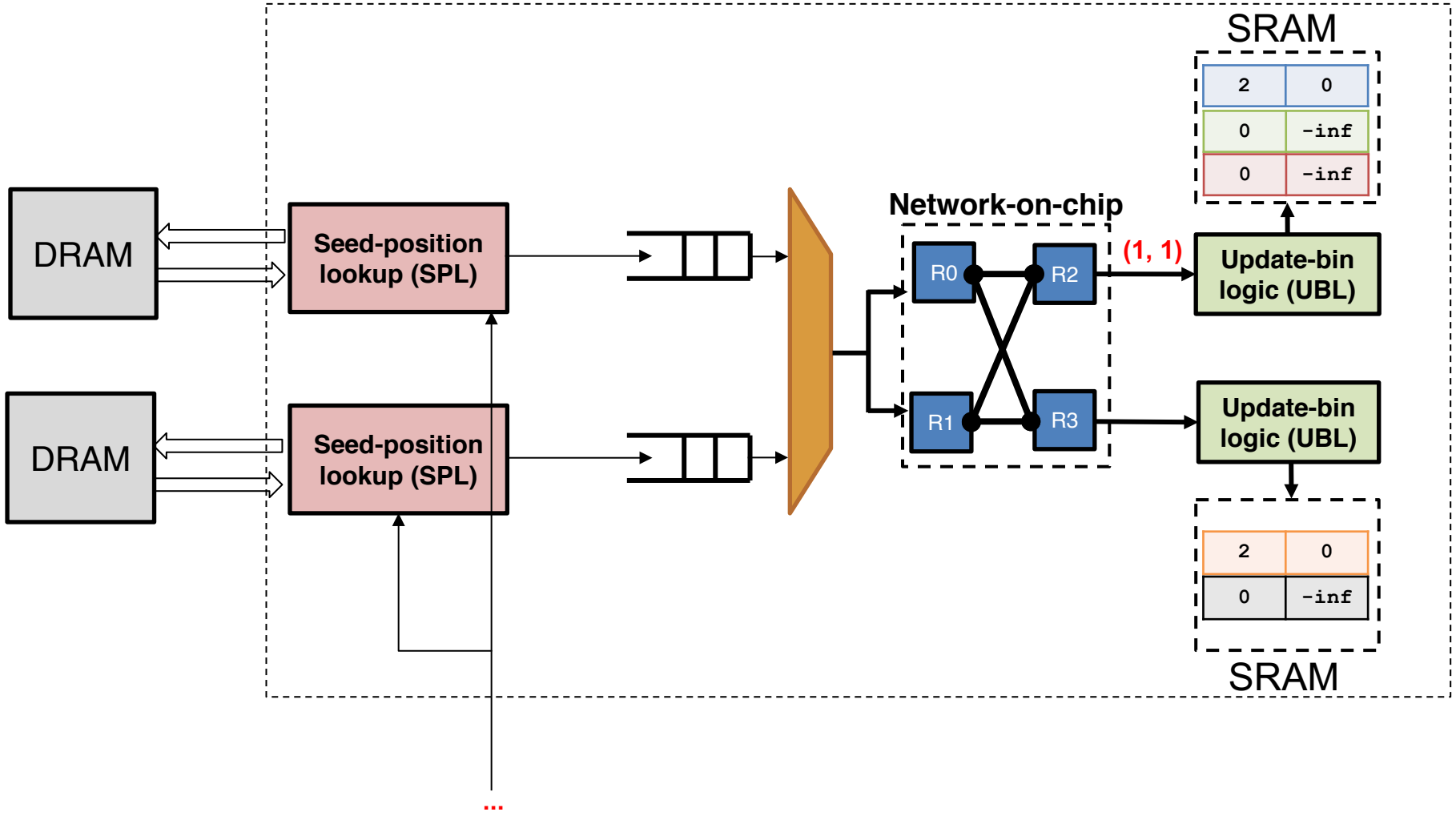
D-SOFT: Hardware-acceleration



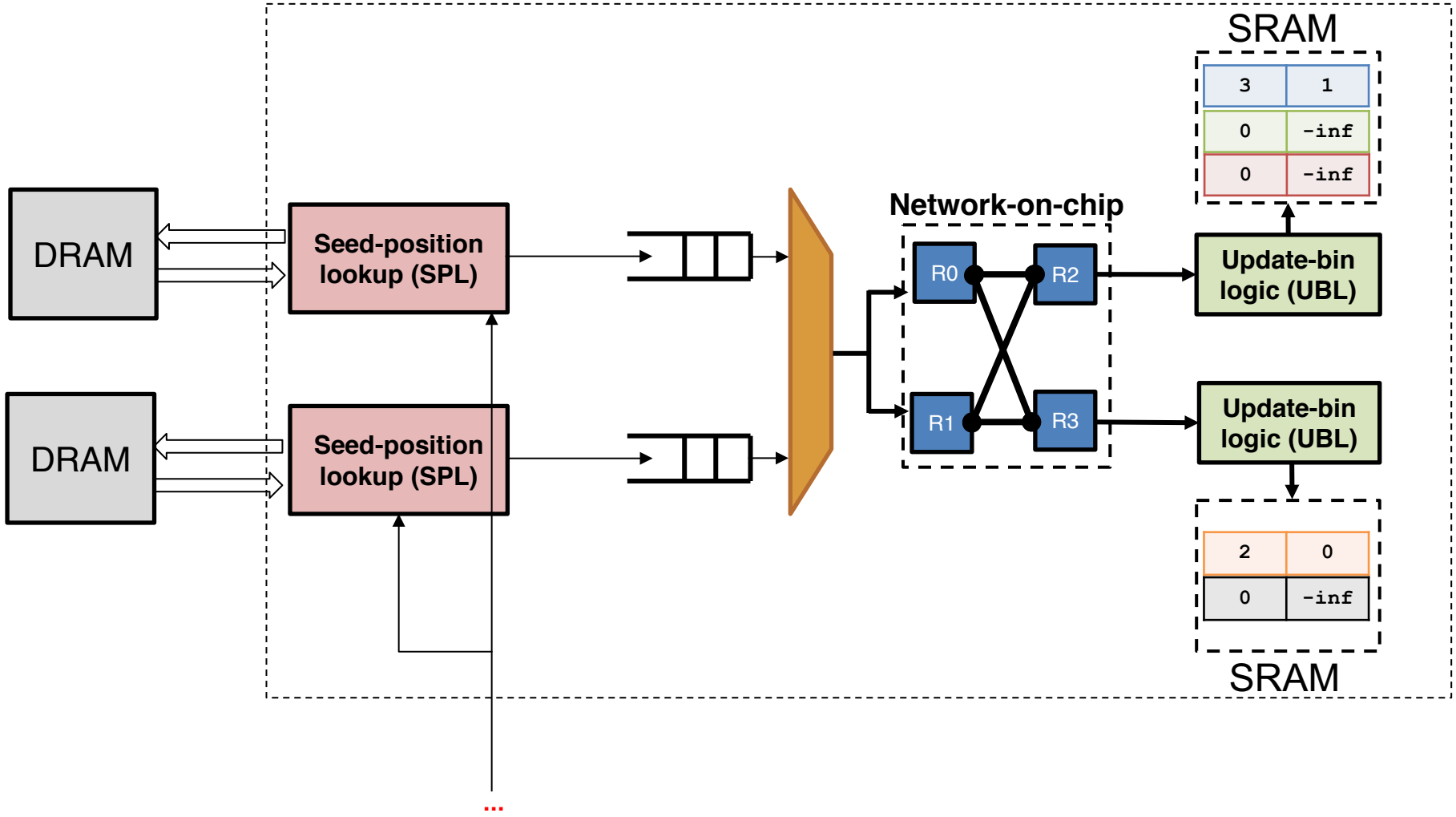
D-SOFT: Hardware-acceleration



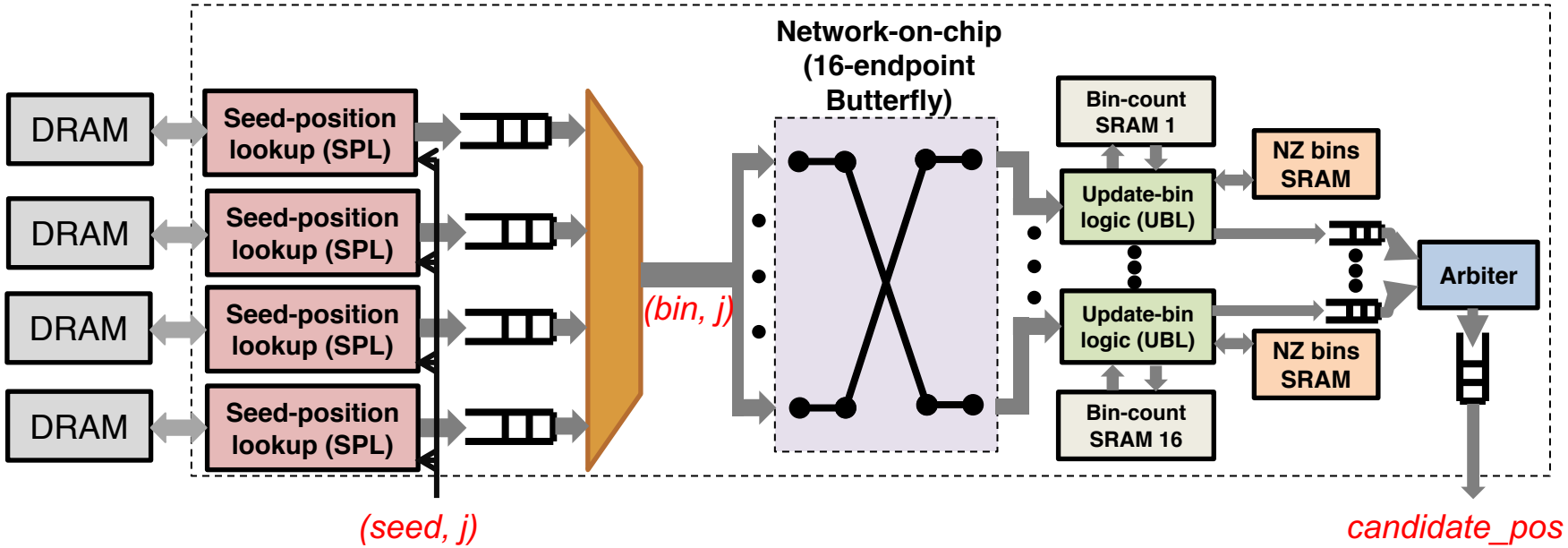
D-SOFT: Hardware-acceleration

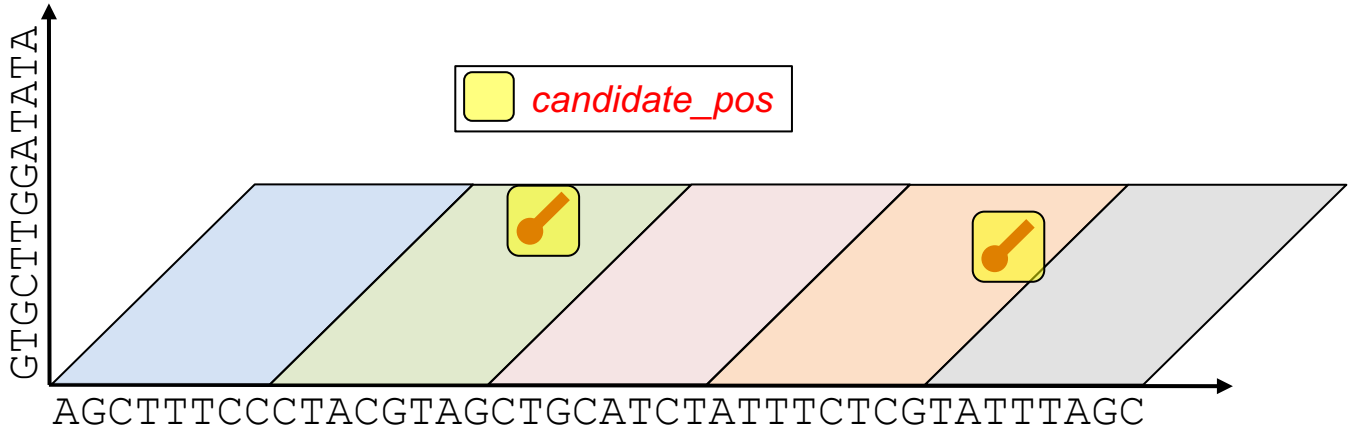


D-SOFT: Hardware-acceleration



D-SOFT: Hardware-acceleration

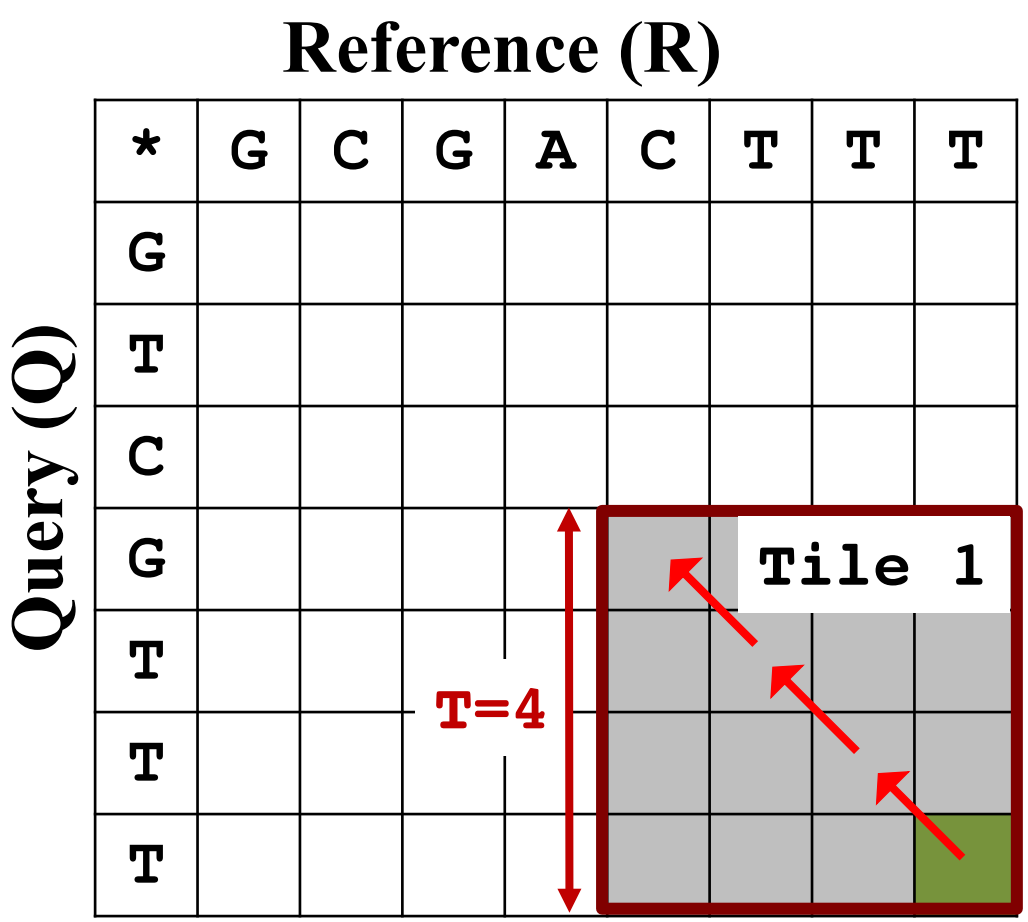




Sections

1. Introduction: DNA Sequencing, Alignment and Long Read Assembly
2. Darwin: Framework Overview and Design Philosophy
3. D-SOFT: Algorithm and Hardware-Acceleration
4. **GACT: Algorithm and Hardware-Acceleration**
5. Experimental Methodology and Results

GACT: Algorithm Overview

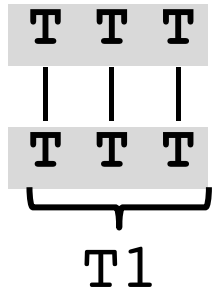


Tile Size (T) = 4, Tile Overlap (O) = 1

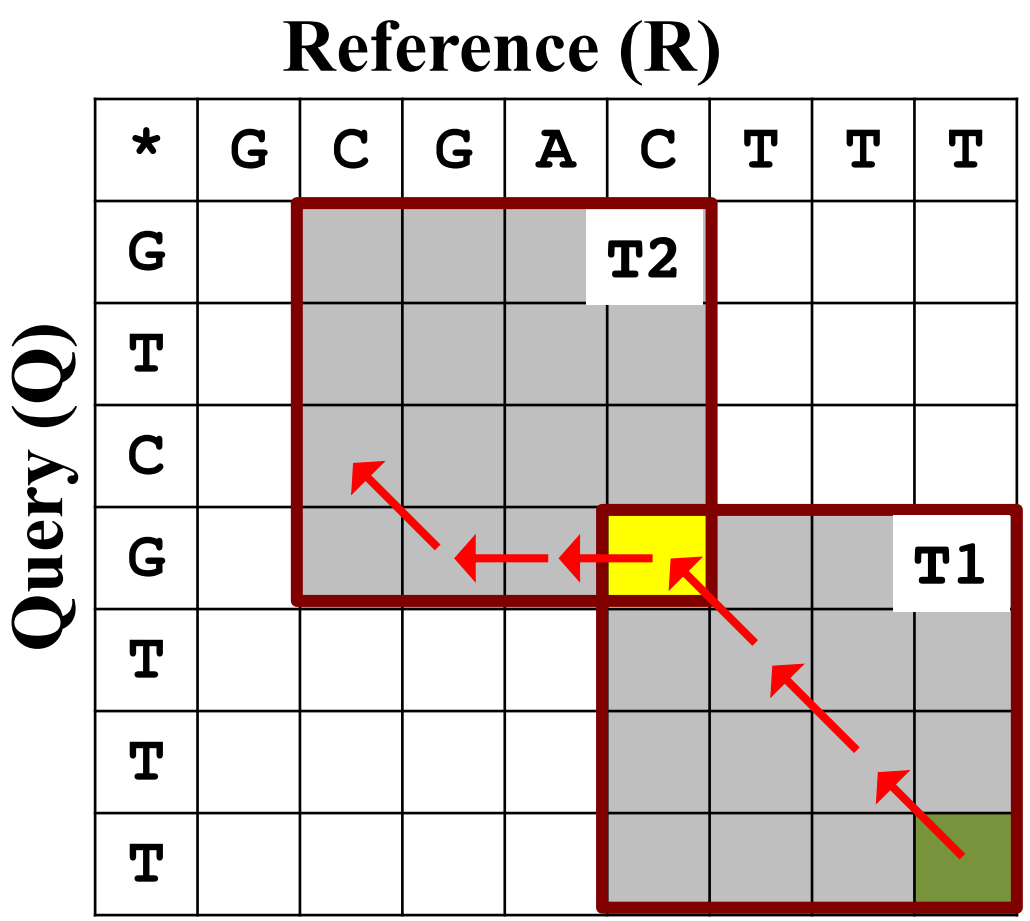
T1

	C	T	T	T
G	0	0	0	0
T	0	2	2	2
T	0	2	4	4
T	0	2	4	6

(i_{off}=3, j_{off}=3)



GACT: Algorithm Overview

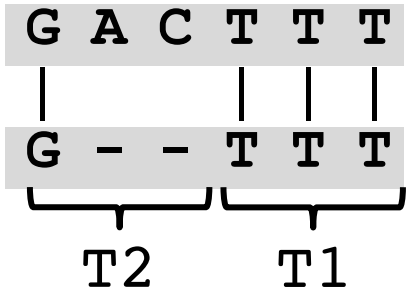


Tile Size (T) = 4, Tile Overlap (O) = 1

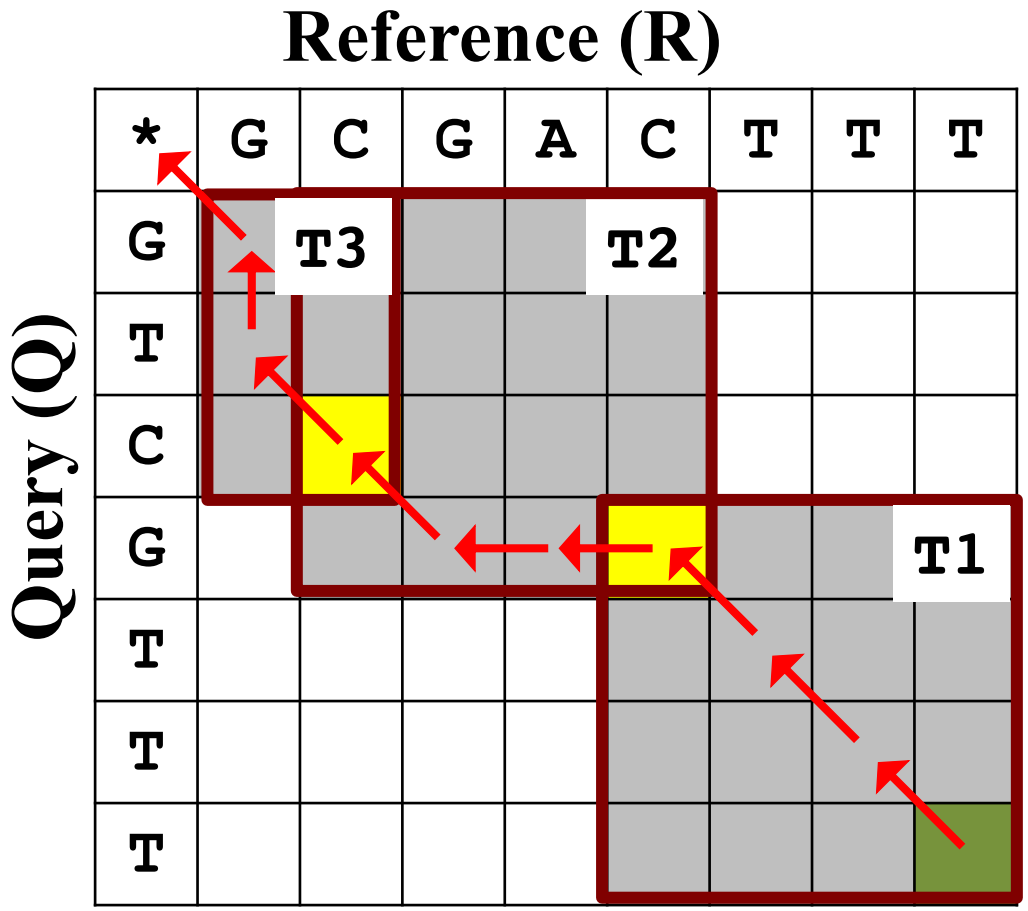
T2

		C	G	A	C
	G	0	2	0	0
	T	0	0	0	0
	C	2	0	0	2
	G	1	4	3	2

(i_{off}=3, j_{off}=1)



GACT: Algorithm Overview



Tile Size (T) = 4, Tile Overlap (O) = 1

T3

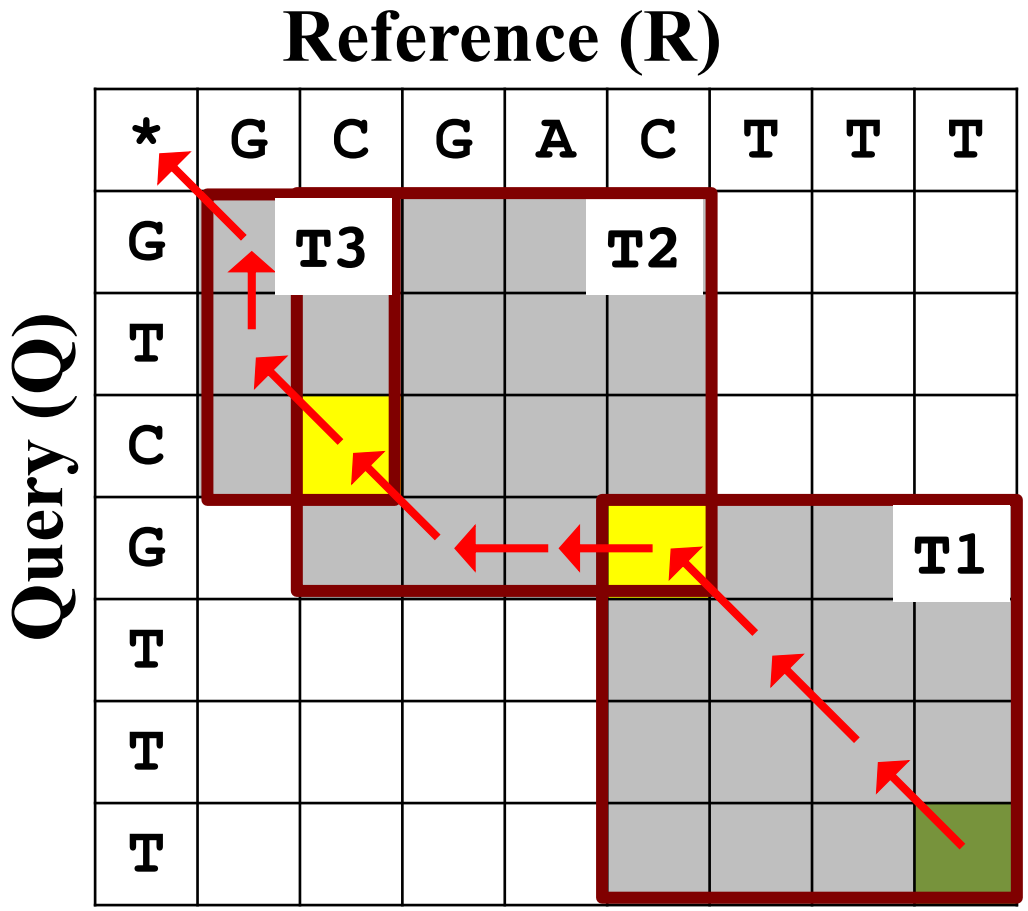
		G	C
G		2	0
T		1	0
C		0	3

(i_{off}=2, j_{off}=3)

Alignment

```
G - C G A C T T T
|   |   |     |   |
G T C G - - T T T
T3    T2    T1
```

GACT: Algorithm Overview



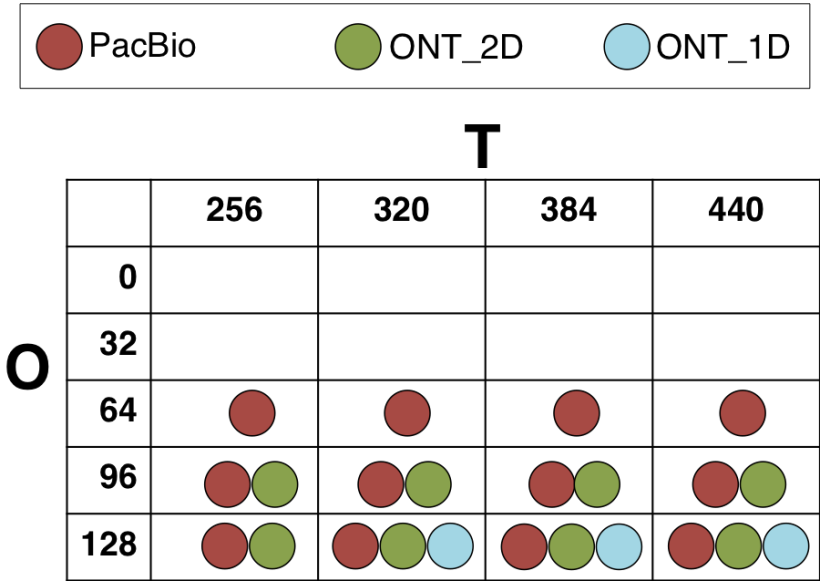
Tile Size (T) = 4, Tile Overlap (O) = 1

Pointers per tile = T^2 = 16

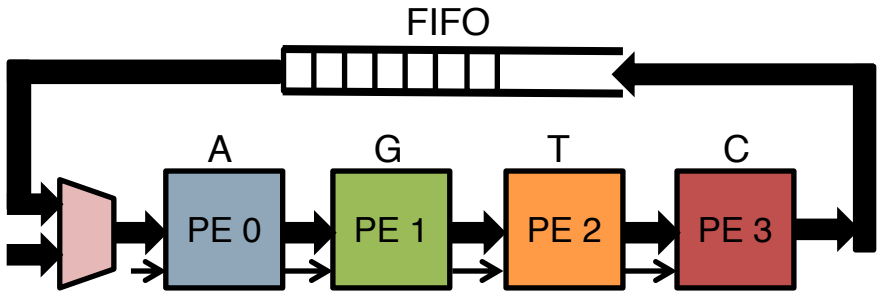
GACT provides empirically optimal alignments

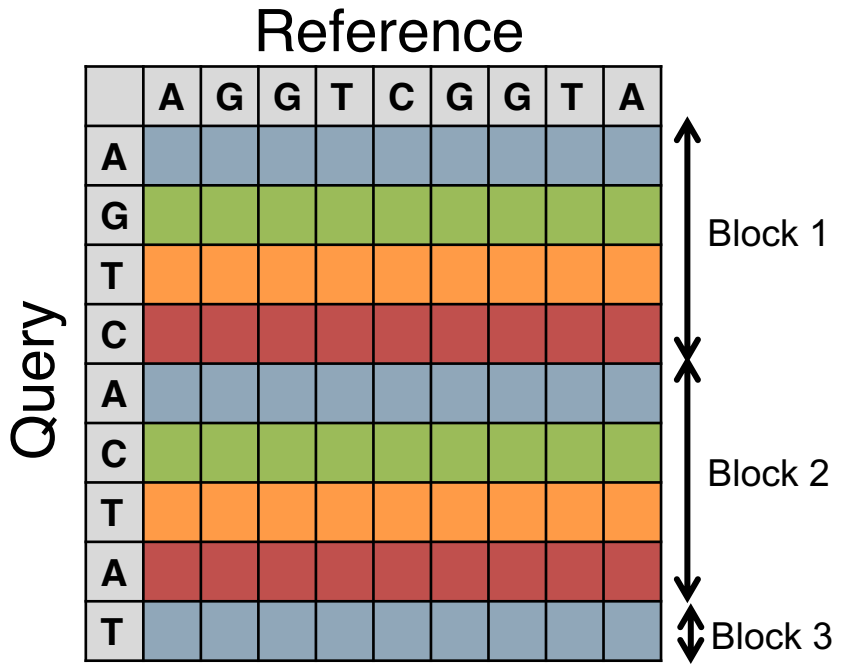
- GACT compared to optimal Smith-Waterman for 200,000 10Kbp sequences with 3 error profiles: PacBio (15% error), ONT_2D (30% error) and ONT_1D (40% error)
- Simple scoring (match: +1, mismatch: -1, gap: -1)
- At T=320, O=128, all observed alignments were **optimal**

Optimal (T, O) settings



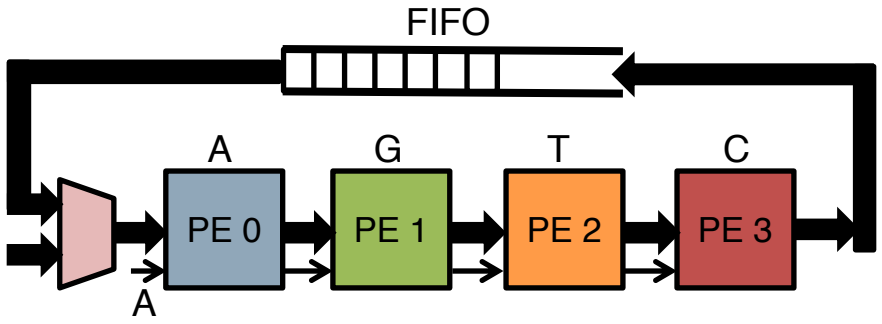
GACT: Hardware-acceleration

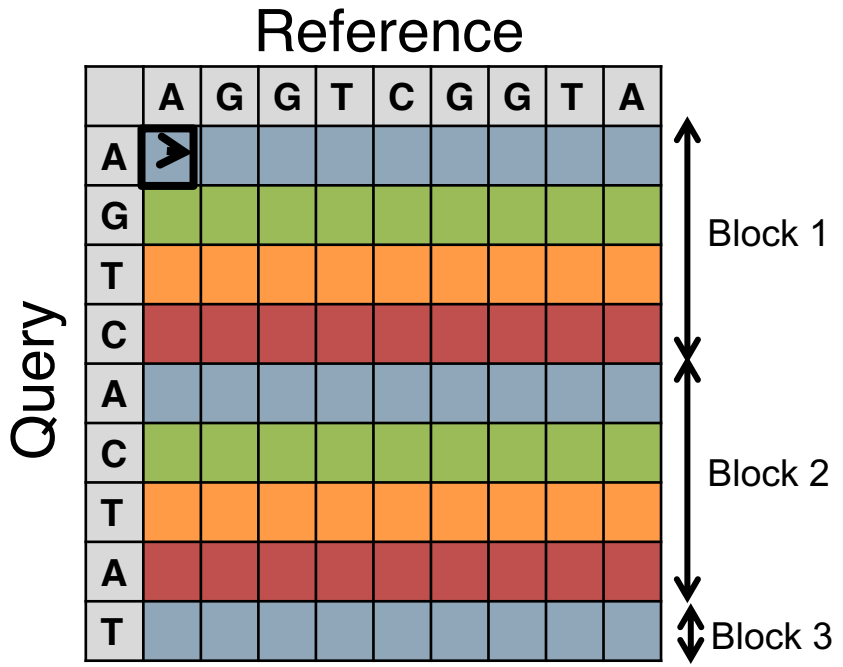




Tile Size (T) = 9

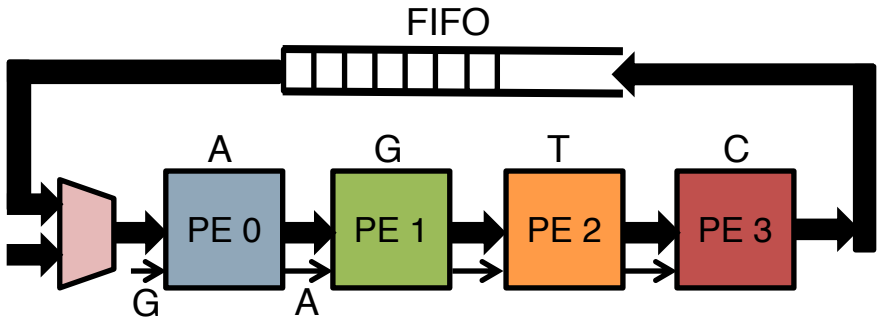
GACT: Hardware-acceleration

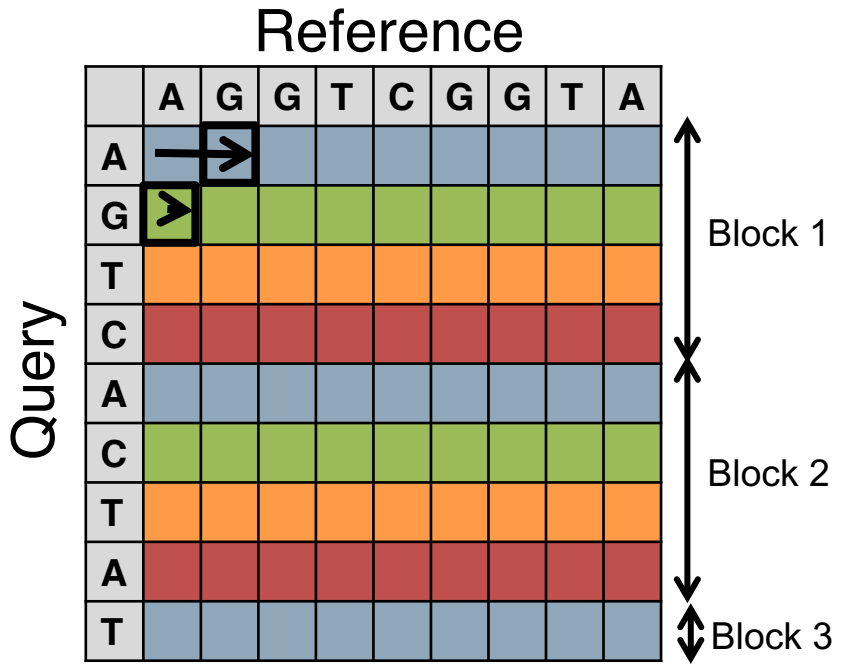




Tile Size (T) = 9

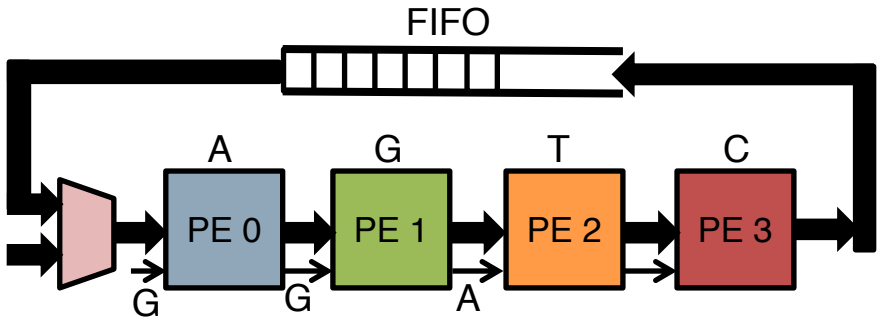
GACT: Hardware-acceleration

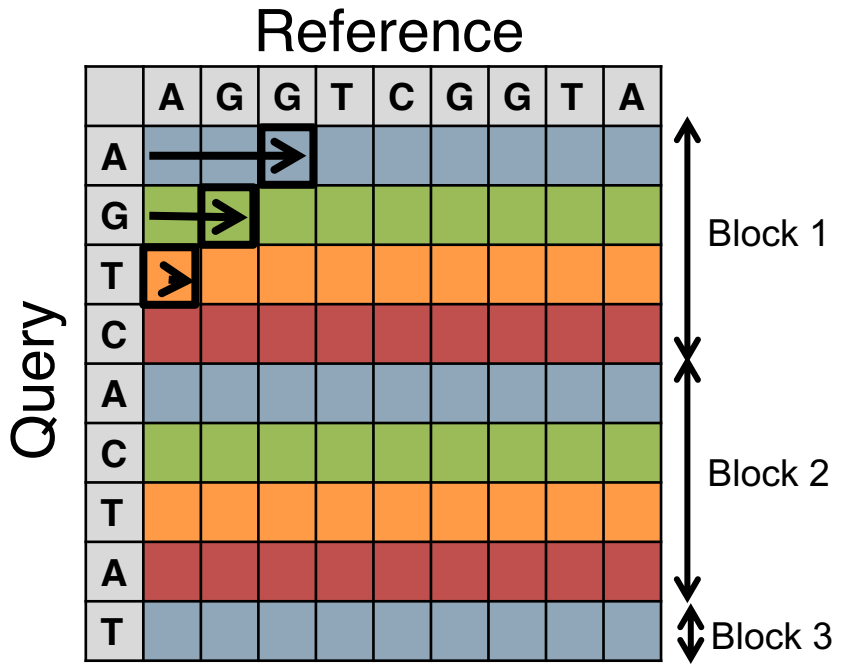




Tile Size (T) = 9

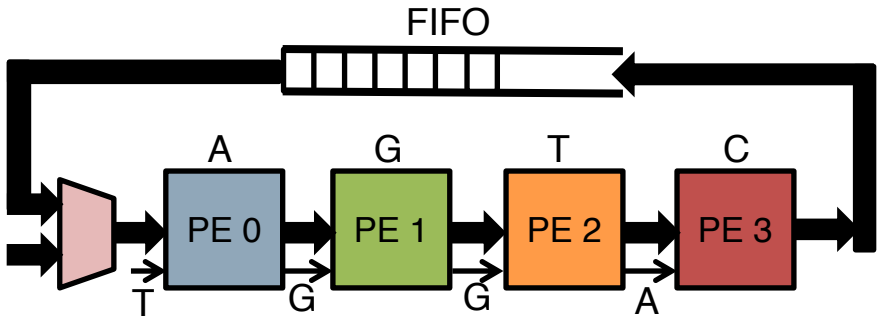
GACT: Hardware-acceleration

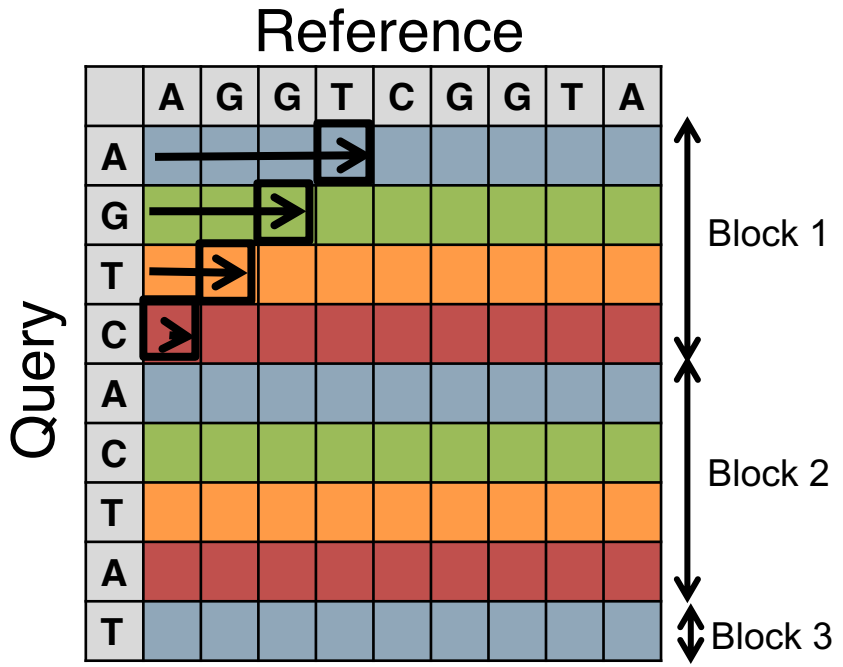




Tile Size (T) = 9

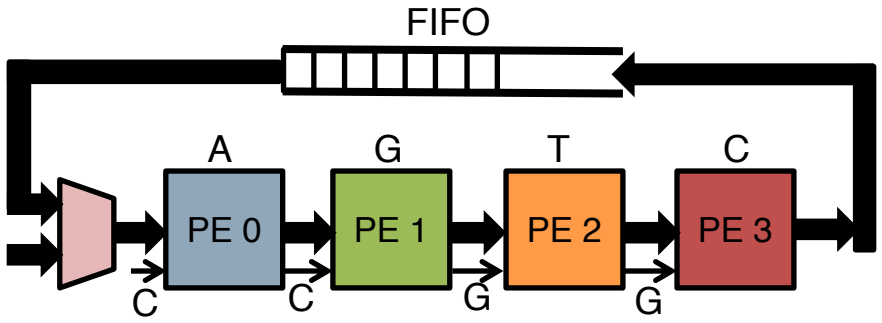
GACT: Hardware-acceleration

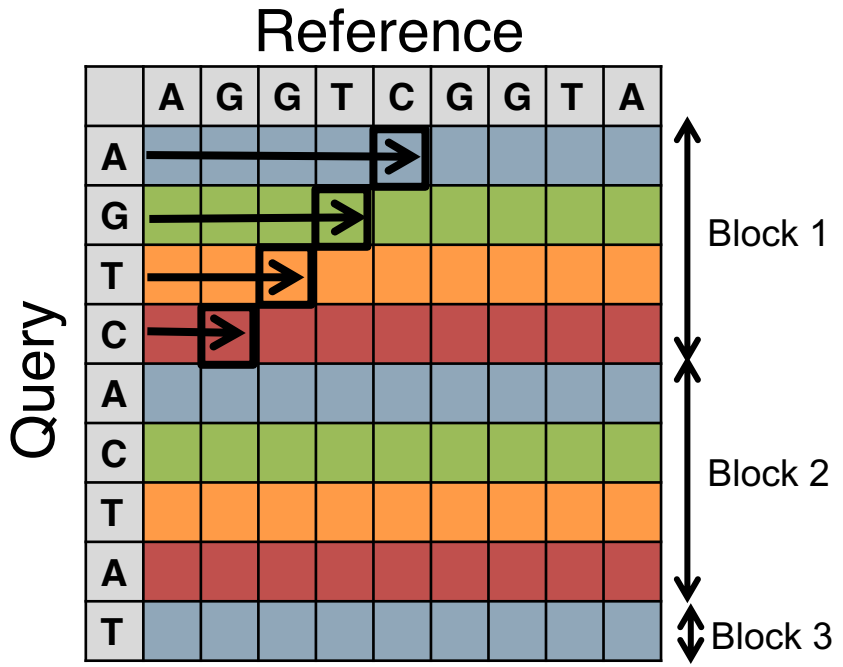




Tile Size (T) = 9

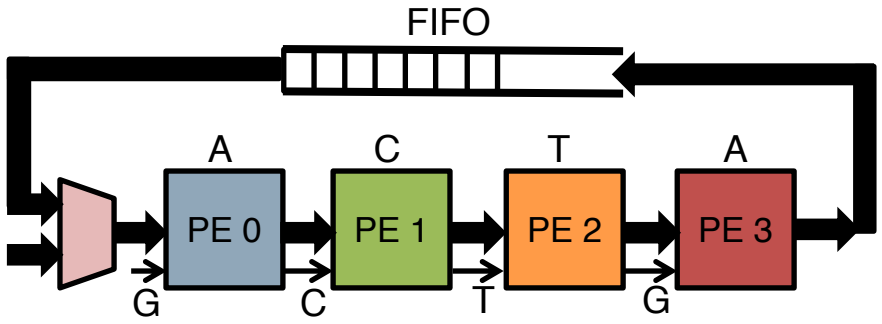
GACT: Hardware-acceleration

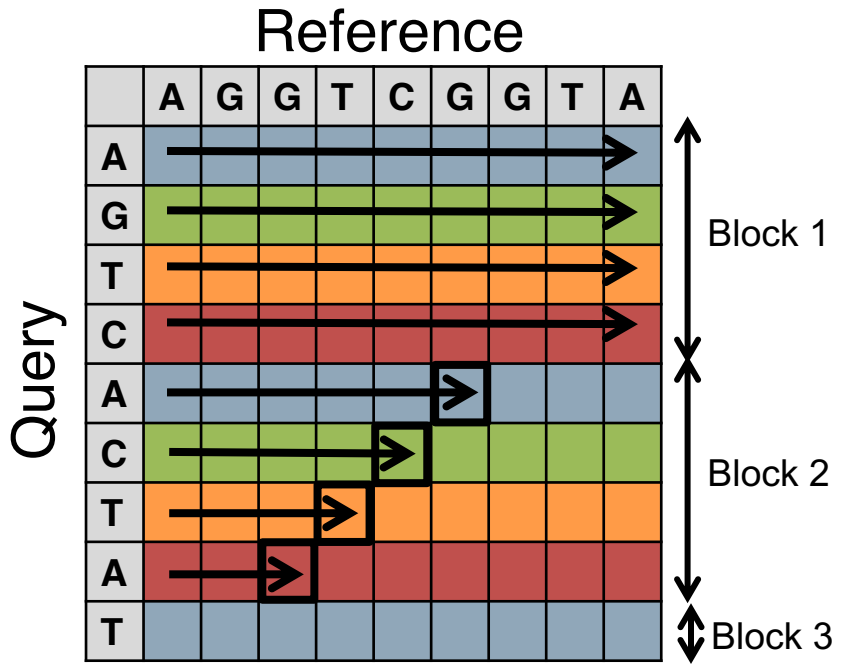




Tile Size (T) = 9

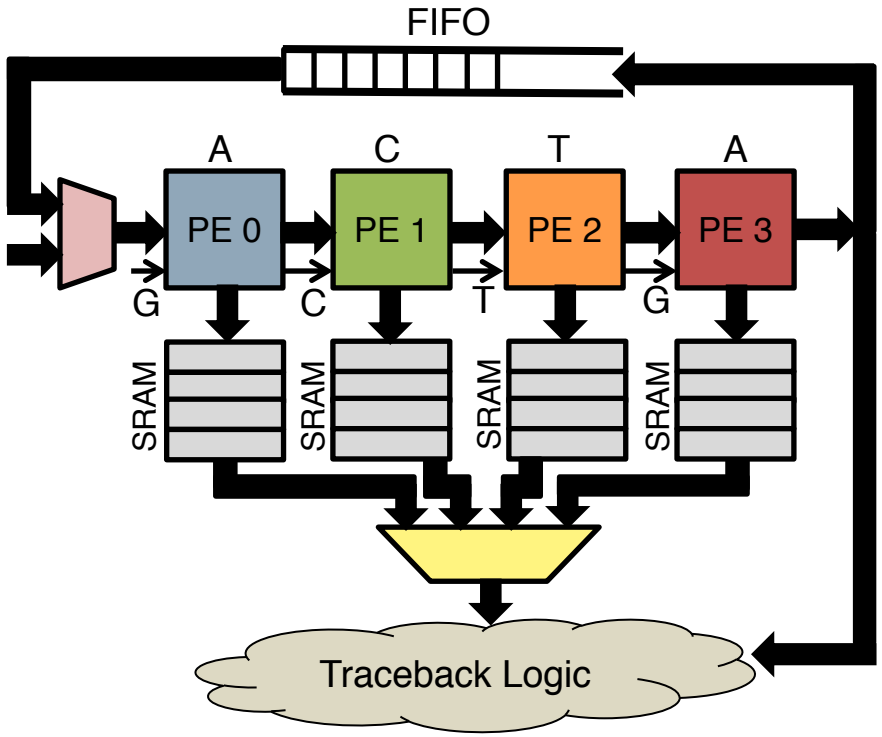
GACT: Hardware-acceleration

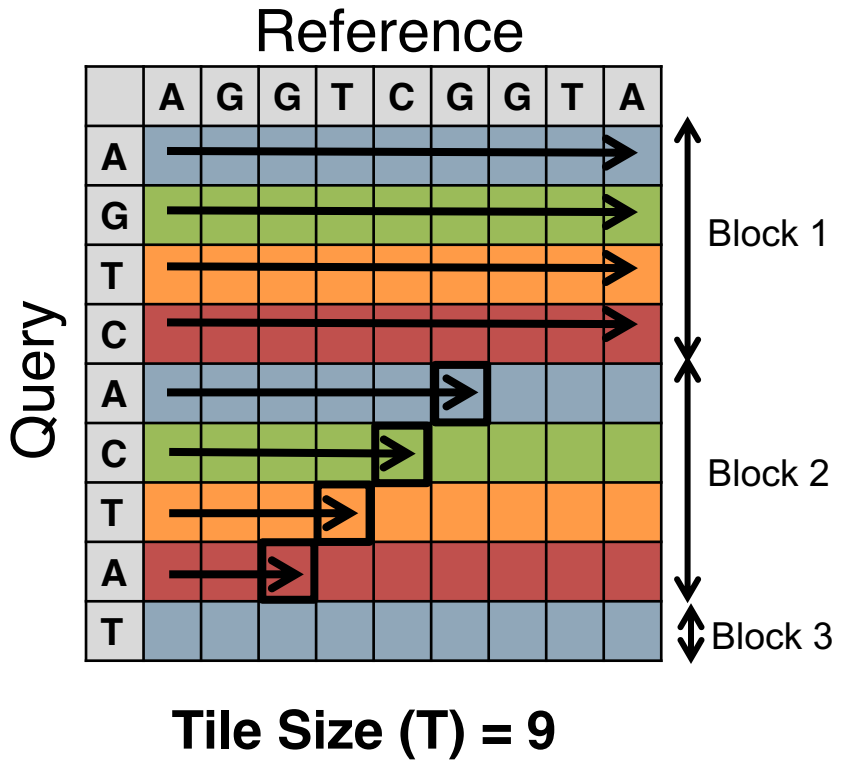




Tile Size (T) = 9

GACT: Hardware-acceleration





In Darwin, T_{max} = 512, requiring only 128KB memory

Sections

1. Introduction: DNA Sequencing, Alignment and Long Read Assembly
2. Darwin: Framework Overview and Design Philosophy
3. D-SOFT: Algorithm and Hardware-Acceleration
4. GACT: Algorithm and Hardware-Acceleration
5. **Experimental Methodology and Results**

Darwin: ASIC overview

Darwin

		Configuration	Area (mm^2) (40nm TSMC)	Power (W) (40nm TSMC)
GACT	Logic	64 x (64PE array)	17.6	1.04
	Memory	64 x (64PE x 2KB/PE)	68.0	3.36
D-SOFT	Logic	2xSPL + NoC + 16xUBL	6.2	0.41
	Bin-count SRAM	16 banks x 4MB/bank	300.8	7.84
	NZ-bin SRAM	16 x 256KB	19.5	0.96
DRAM	LPDDR4-2400	4 x 32GB	-	1.64
TOTAL			**412.1**	**15.25**

Baseline

- 1 thread, Intel Xeon E5-2658 at 2.2GHz
- 128GB DDR4-2133
- ~10W power (best iso-power point after accounting for technology node)

Darwin: ASIC overview

Darwin

		Configuration	Area (mm^2) (40nm TSMC)	Power (W) (40nm TSMC)
GACT	Logic	64 x (64PE array)	17.6	1.04
	Memory	64 x (64PE x 2KB/PE)	68.0	3.36
D-SOFT	Logic	2xSPL + NoC + 16xUBL	6.2	0.41
	Bin-count SRAM	16 banks x 4MB/bank	300.8	7.84
	NZ-bin SRAM	16 x 256KB	19.5	0.96
DRAM	LPDDR4-2400	4 x 32GB	-	1.64
TOTAL			**412.1**	**15.25**

Baseline

- 1 thread, Intel Xeon E5-2658 at 2.2GHz
- 128GB DDR4-2133
- ~10W power (best iso-power point after accounting for technology node)

D-SOFT: 40-85X speedup on ASIC

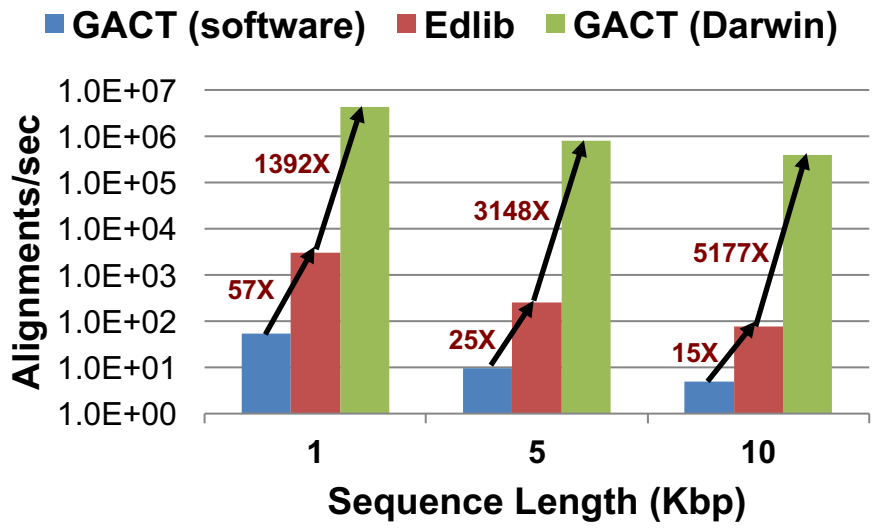
k	Avg. hits per seed (Human Genome)	Software Throughput (10^3 seeds/sec)	Darwin speedup
11	1866.1	16.6	**85X**
12	491.6	66.2	**82X**
13	127.3	259.3	**73X**
14	33.4	869.5	**63X**
15	8.7	2,257.1	**40X**

Why is D-SOFT ASIC so fast?

Memory Parallelism and Localization

1. Four DRAM channels (**~4X**)
2. Fewer DRAM accesses by on-chip bin updates (**~3X**)
3. Changing random DRAM access pattern to sequential (**~3-8X**)

GACT: ~80,000X speedup on ASIC



Why is GACT ASIC so fast?

Massive Parallelism

1. 4096 Processing Elements (PEs)
2. ~10 operations/PE/cycle @ 847MHz

~34.6 TeraOps/sec

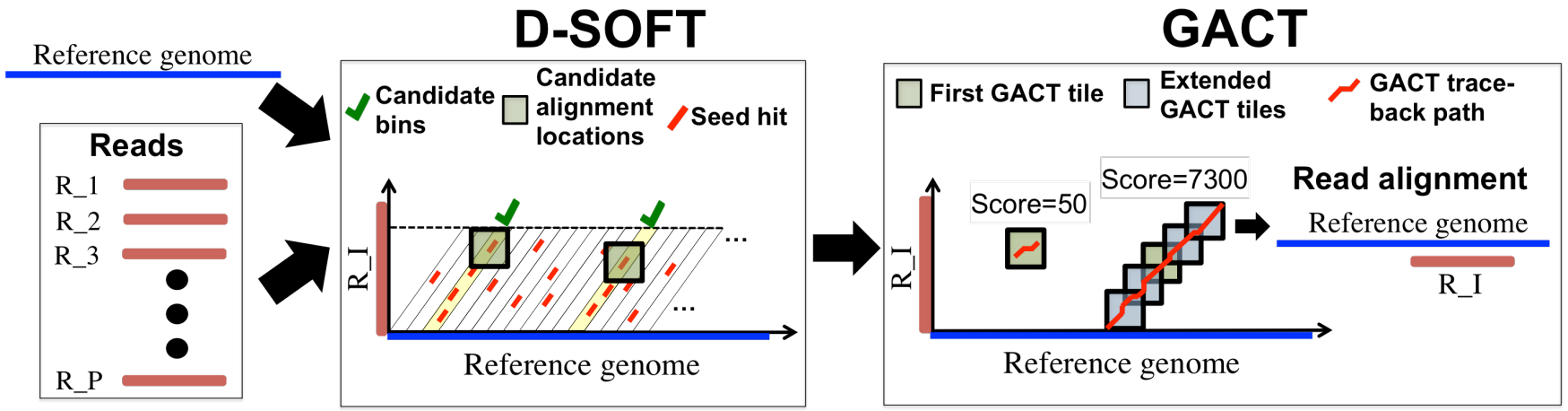
Small memory footprint => enormous bandwidth

1. 4096 Processing Elements (PEs)
2. ~4b write/PE/cycle @ 847MHz

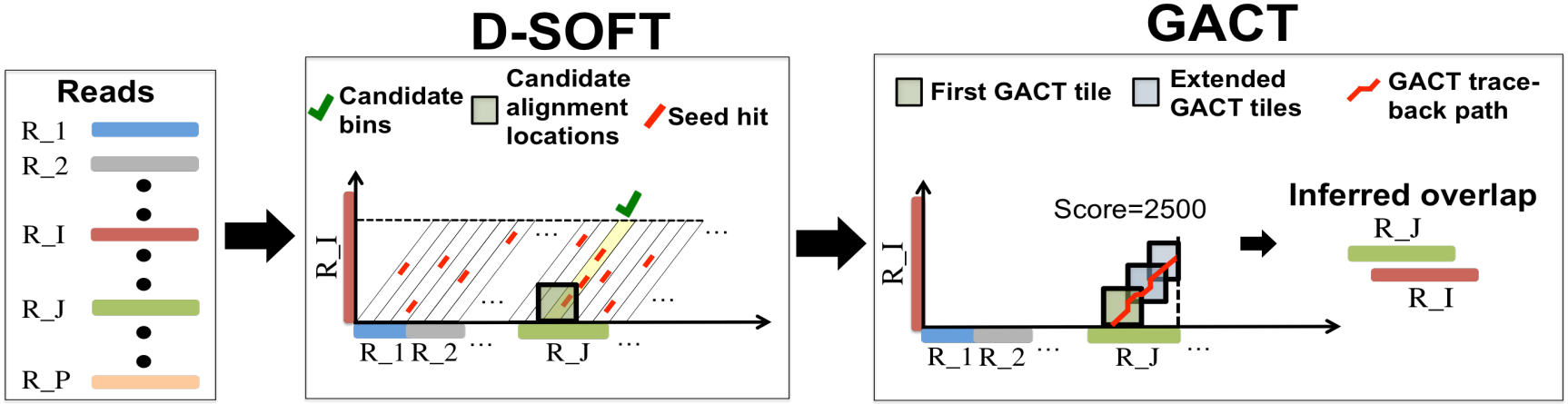
~1.7 TeraBytes/sec

Long read assembly using Darwin

Reference-guided



De novo (Overlap step)



Darwin: Assembly results

Reference-guided (54X human genome)

Read Error Rate	D-SOFT settings (k, N, h)	Sensitivity		Speedup
		Baseline	Darwin	
15%	(14, 750, 24)	95.95%	99.91%	**9,916X**
30%	(12, 1000, 25)	98.11%	98.40%	**15,062X**
40%	(11, 1300, 22)	97.10%	97.40%	**1,244X**

Baseline: BWA-MEM (15%), GraphMap (30%, 40%)

De novo (54X human genome)

Read Error Rate	D-SOFT settings (k, N, h)	Sensitivity		Speedup (Bottleneck)
		Baseline	Darwin	
15%	(14, 1300, 24)	99.80%	99.89%	**710X**

Baseline: DALIGNER

Thank You!