SegAlign

A Scalable GPU-Based Whole Genome Aligner

Sneha D. Goenka[+*]   Yatish Turakhia[#*]   Benedict Paten[#]   Mark Horowitz[+]

[+] Stanford University

[#] UCSC Genomics Institute

*equal contribution

> $5 Billion to sequence all species on Earth

EARTH BIOGENOME PROJECT
sequencing life for the future of life
$4.8B

VGP VERTEBRATE GENOMES PROJECT
A PROJECT OF THE G10K CONSORTIUM
$600M

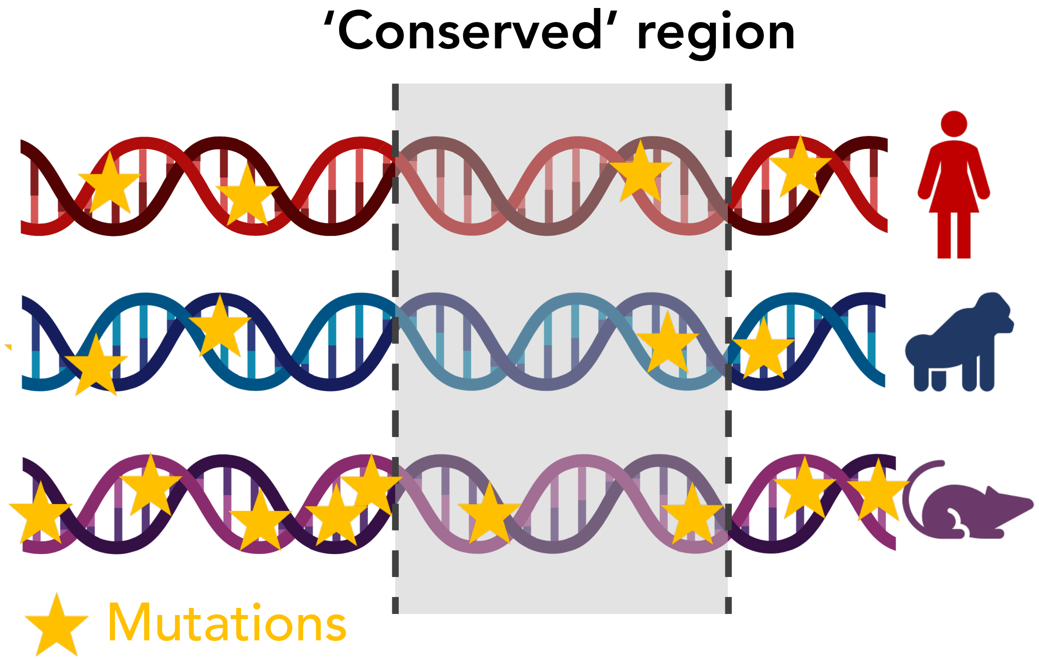
Darwin TREE of LIFE
$130M

B10K
$50M

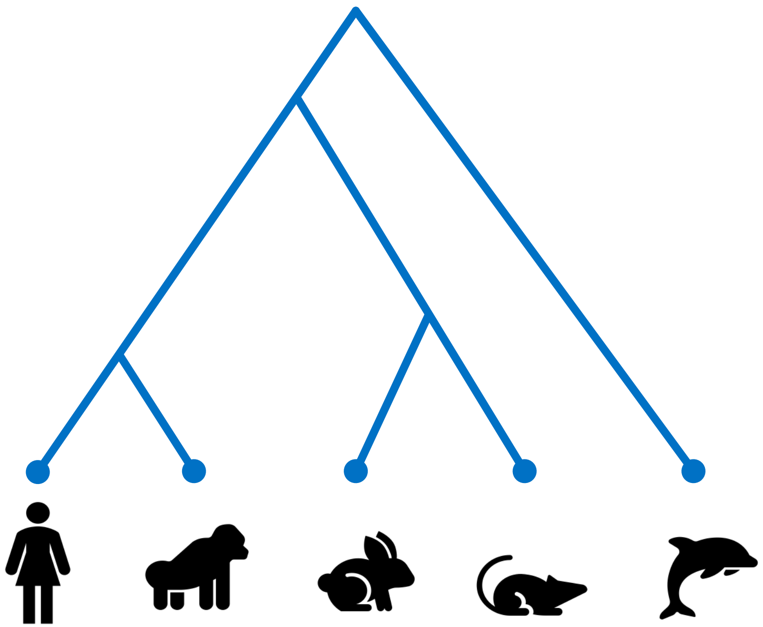
BAT 1K
$30M

https://www.earlham.ac.uk/newsroom/earlham-institute-branches-out-darwin-tree-life
https://www.genomeweb.com/genetic-research/vertebrate-genomes-project-plans-combine-technologies-near-gapless-assemblies

Whole Genome Alignments (WGA): first step in comparative genomics

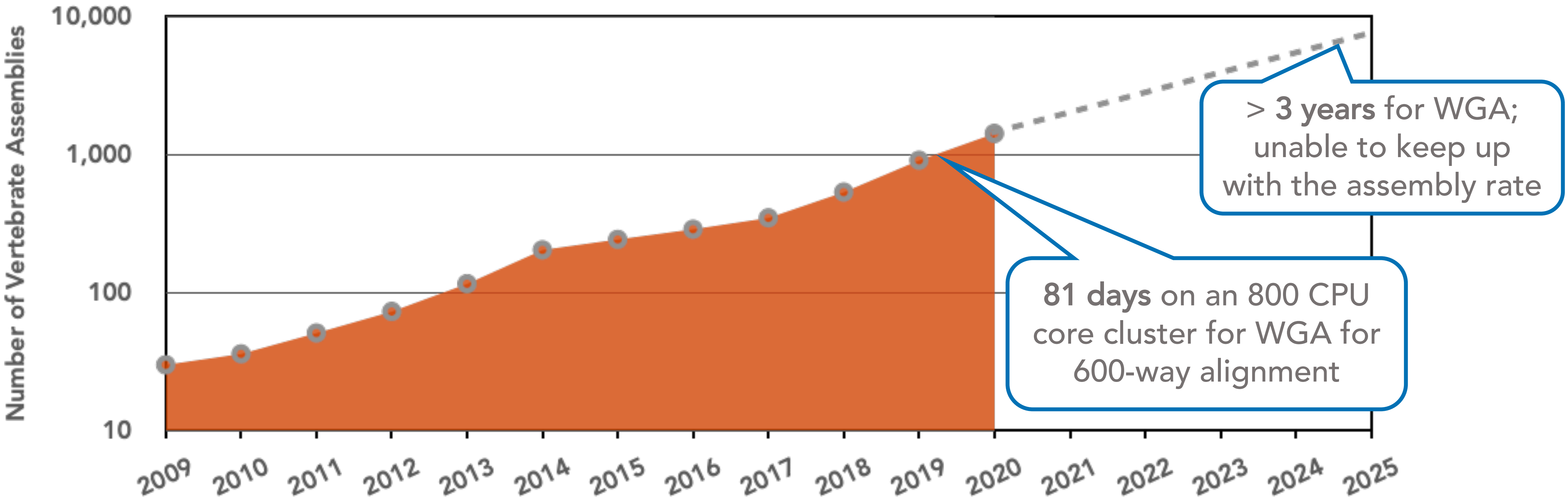


Prediction of functional elements



Phylogenetics

We have already entered the thousand-genome era

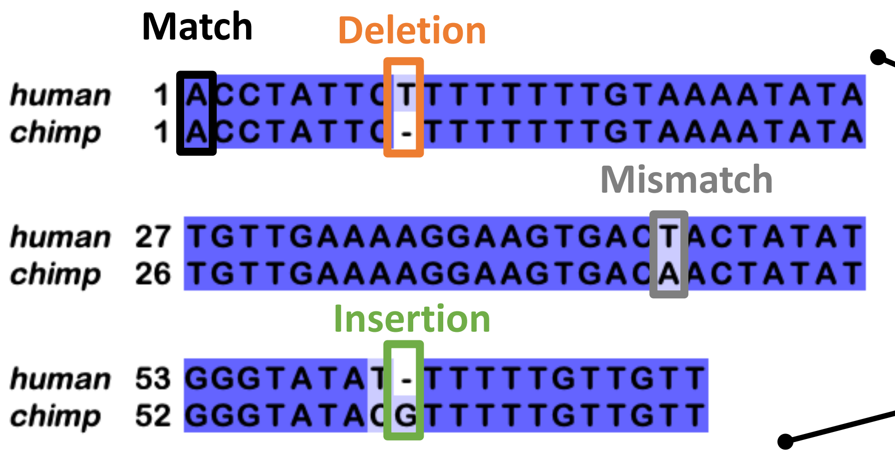


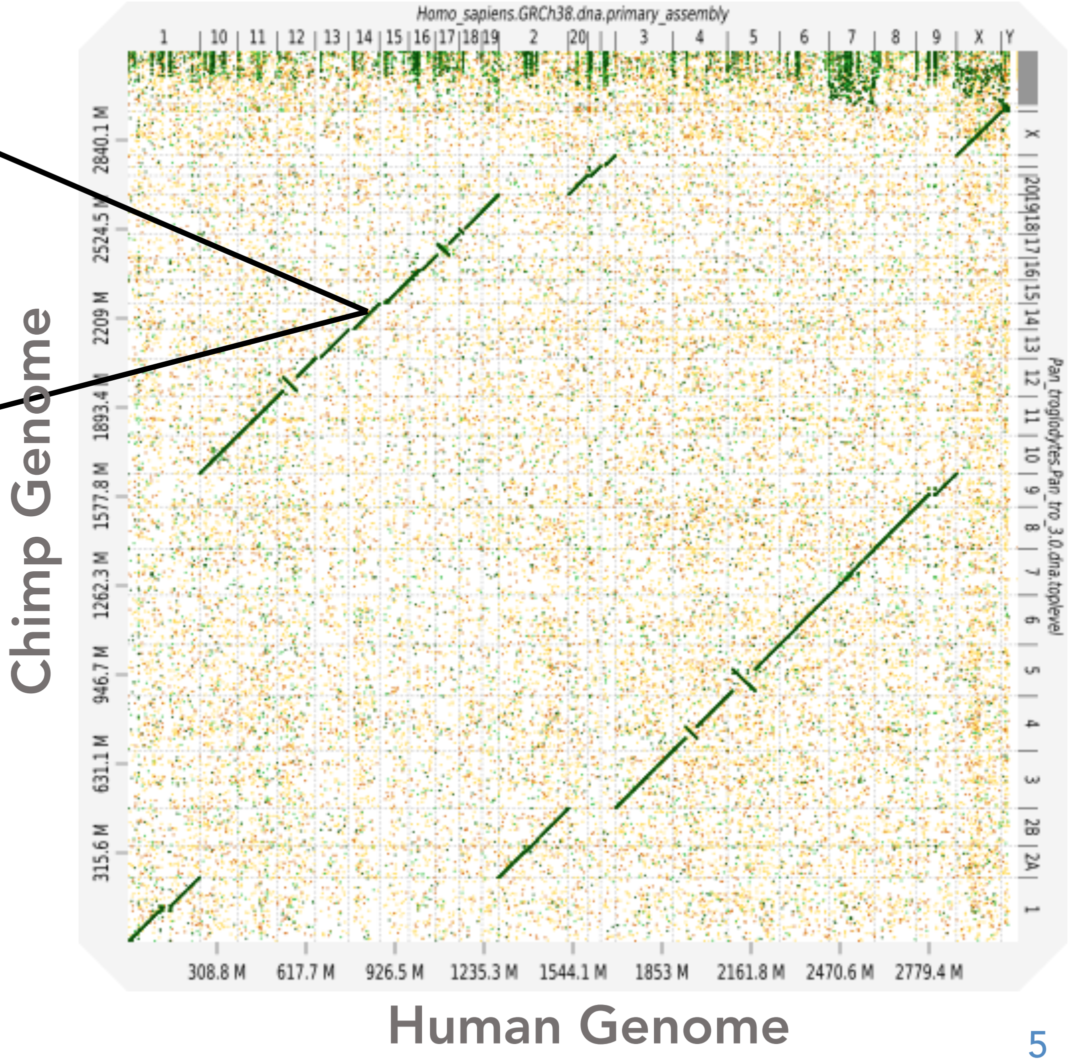
> **3 years** for WGA; unable to keep up with the assembly rate

81 days on an 800 CPU core cluster for WGA for 600-way alignment

NCBI Genome Database
Armstrong et al. *bioRxiv* (2019)

Dot plot for human-chimp WGA

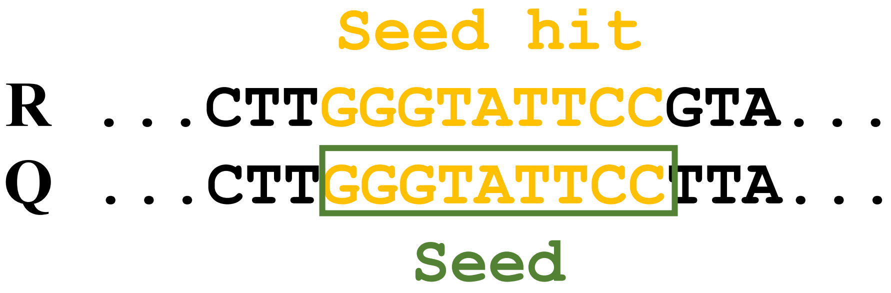

Match
Deletion
human 1 ACCTATTCTTTTTTTTGTAAAATATA
chimp 1 ACCTATTC-TTTTTTTGTAAAATATA
Mismatch
human 27 TGTTGAAAAGGAAGTGACTACTATAT
chimp 26 TGTTGAAAAGGAAGTGACAACTATAT
Insertion
human 53 GGGTATAT-TTTTTGTTGTT
chimp 52 GGGTATACGTTTTTGTTGTT



LASTZ is the state-of-the art whole genome aligner, based on the *seed-filter-extend* algorithm

Cabanettes et al. PeerJ6:e4958(2018)

Seeding finds small, local matching base-pairs

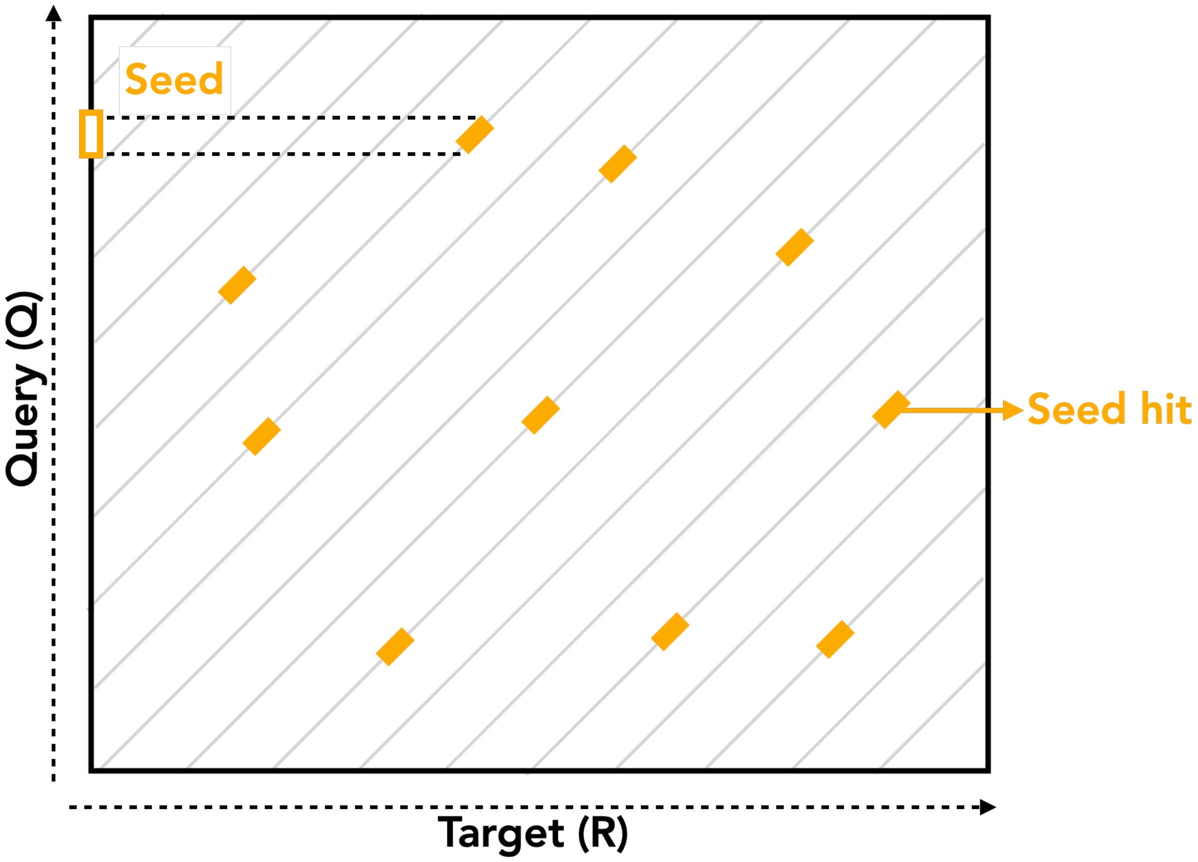


Seed hit

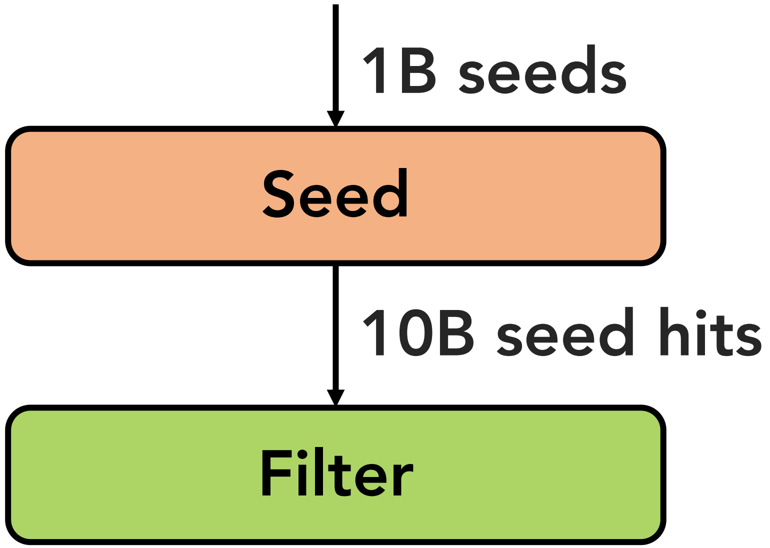
R ...CTTGGGTATTCCGTA...

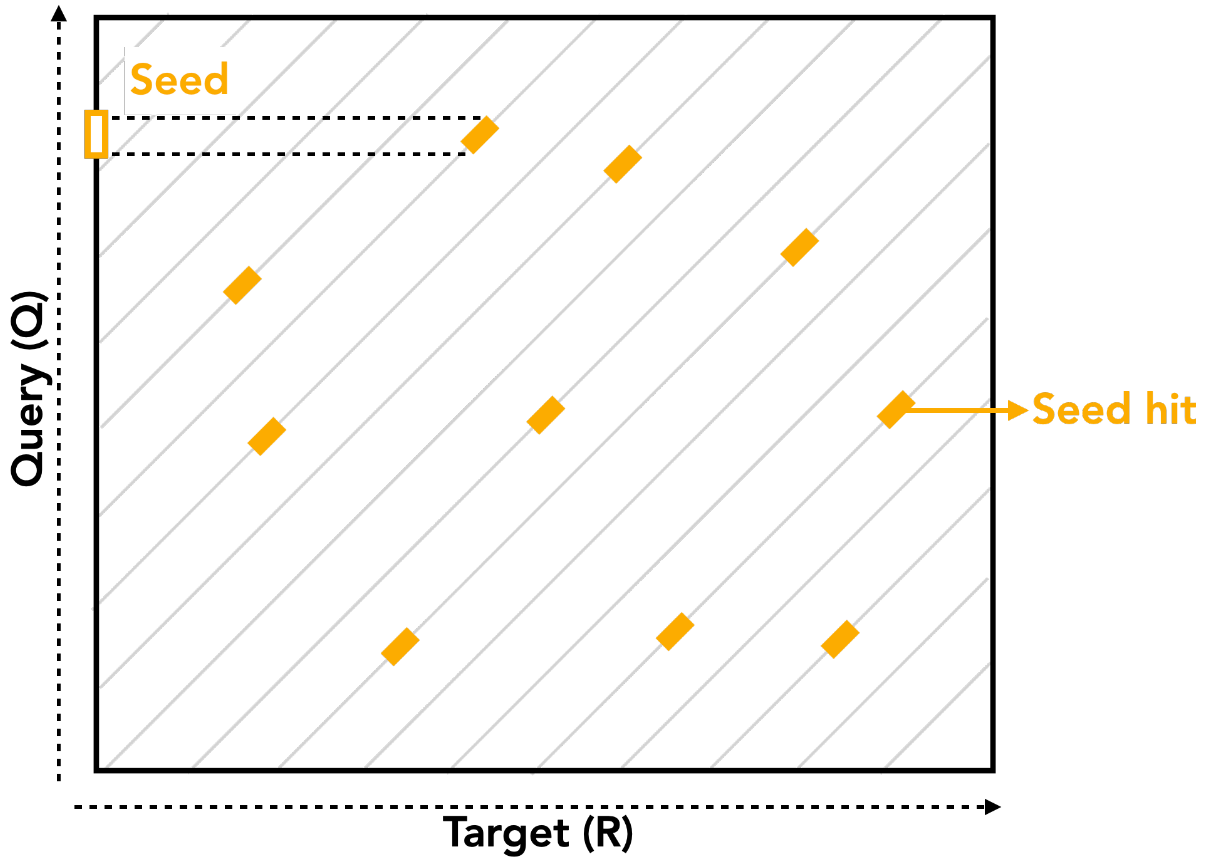
Q ...CTTGGGTATTCCTTA...

Seed



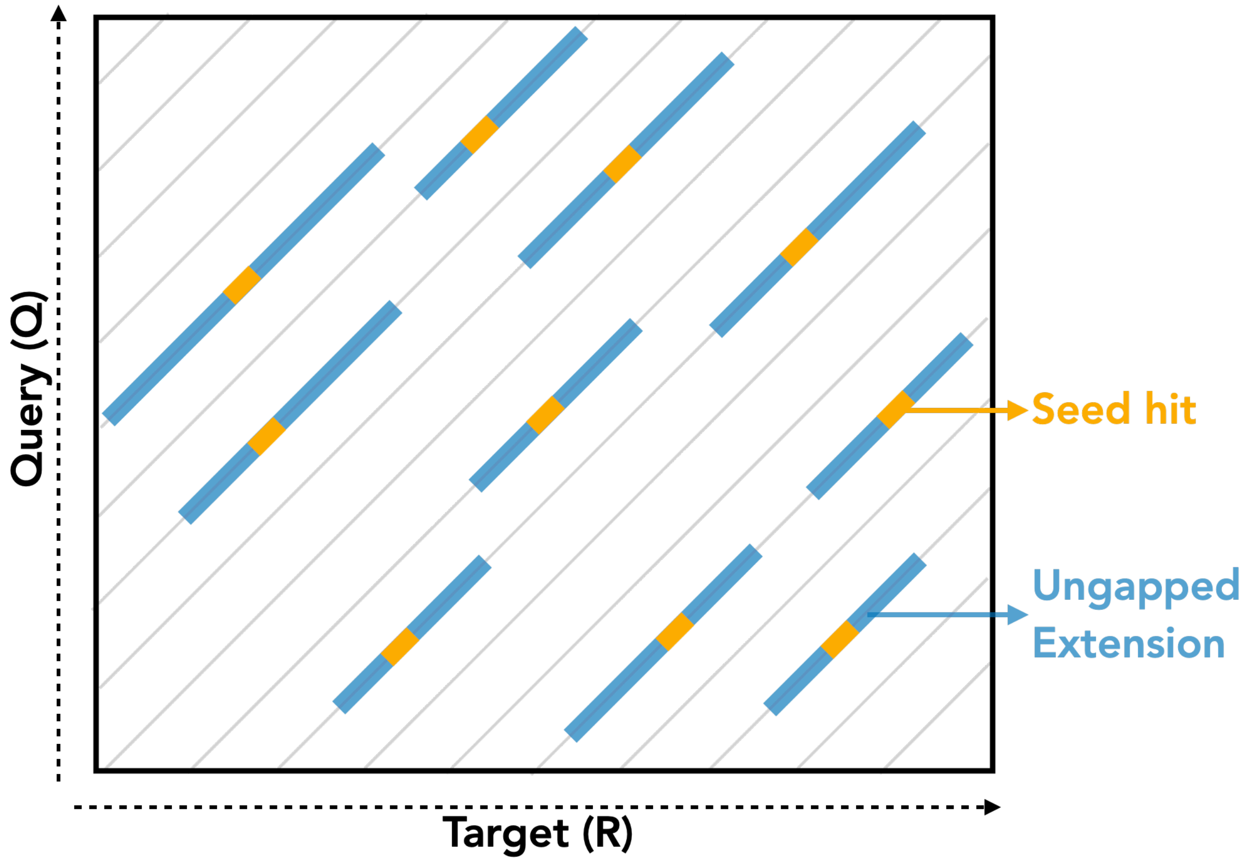
Seeding finds small, local matching base-pairs





Filtering aligns ~100bp around seed hits

R	A	A	G	T	C	A	A	T
Q	A	T	G	T	A	T	T	C
	2	-1	2	2	-1	-1	-1	-1
Cumulative Score	2	1	3	5	4	3	2	1
Max Score	2	2	3	5	5	5	5	5
Score Difference	0	1	0	0	1	2	3	4



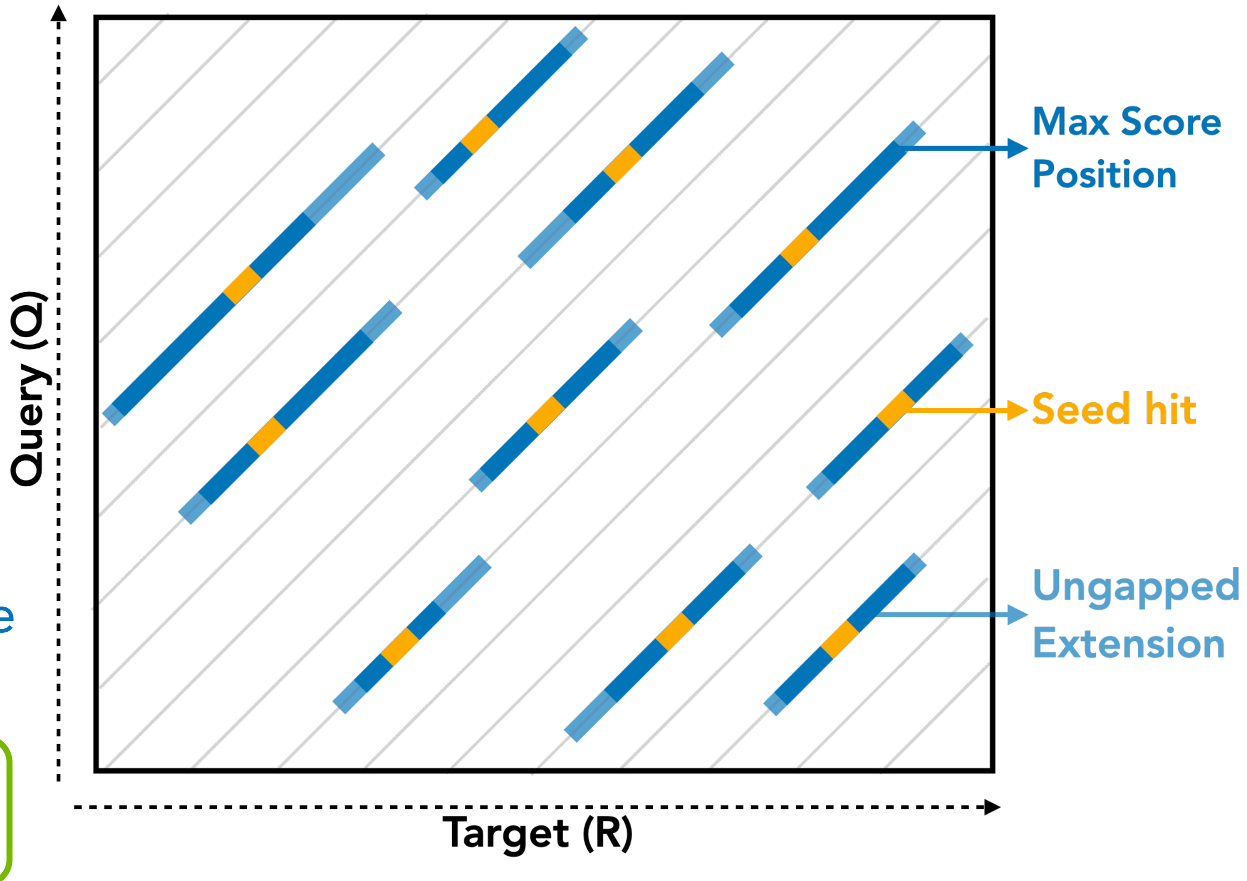
Filtering aligns ~100bp around seed hits

R	A	A	G	T	C	A	A	T
Q	A	T	G	T	A	T	T	C
	2	-1	2	2	-1	-1	-1	-1
Cumulative Score	2	1	3	5	4	3	2	1
Max Score	2	2	3	5	5	5	5	5
Score Difference	0	1	0	0	1	2	3	4

Max Score Position

Terminate Position

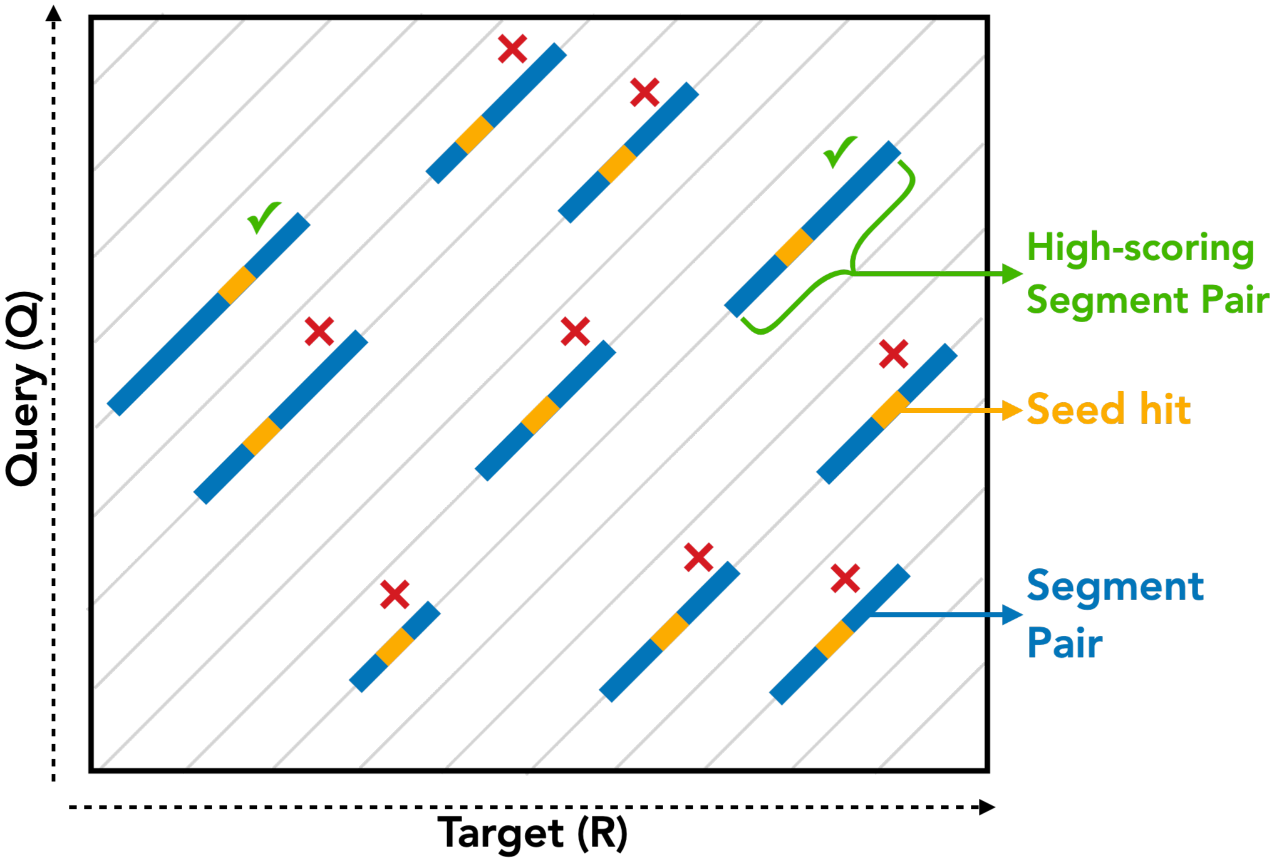
X-drop Condition:
Score Difference >= $H_x(4)$



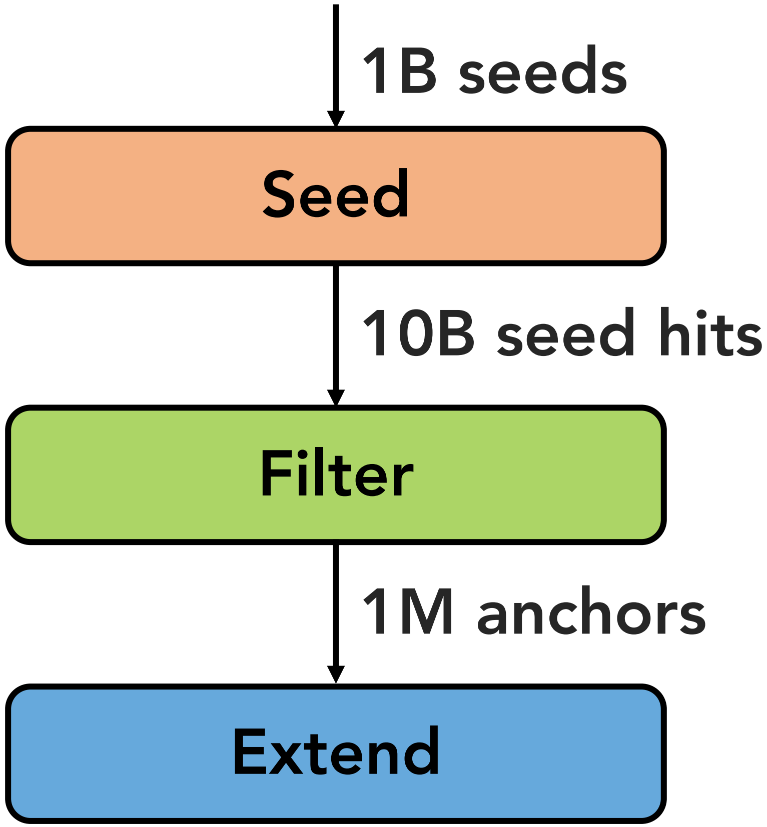
Filtering aligns ~100bp around seed hits

Right Segment Pair

R	A	A	G	T	C	A	A	T
Q	A	T	G	T	A	T	T	C
	2	-1	2	2	-1	-1	-1	-1
Cumulative Score	2	1	3	5	4	3	2	1
Max Score	2	2	3	5	5	5	5	5
Score Difference	0	1	0	0	1	2	3	4



High-scoring Segment Pair reduced to Anchor



1B seeds

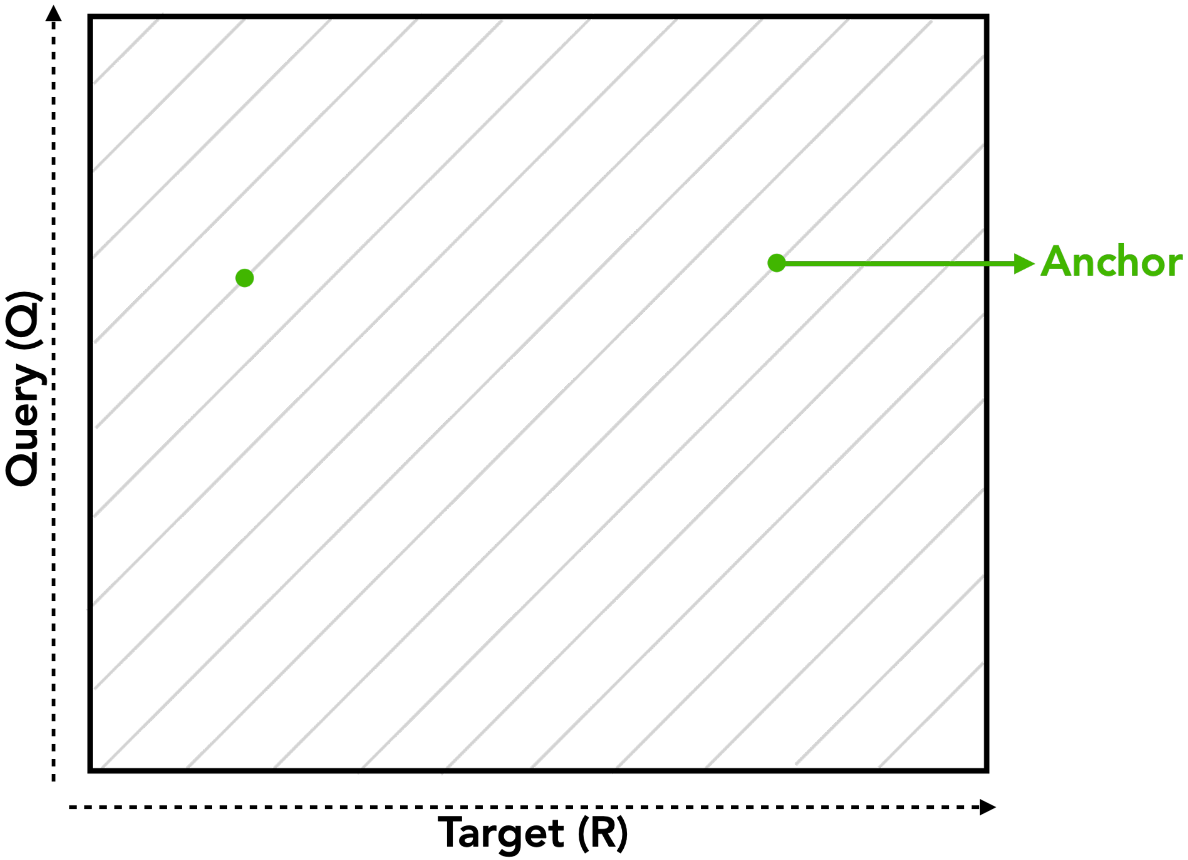
Seed

10B seed hits

Filter

1M anchors

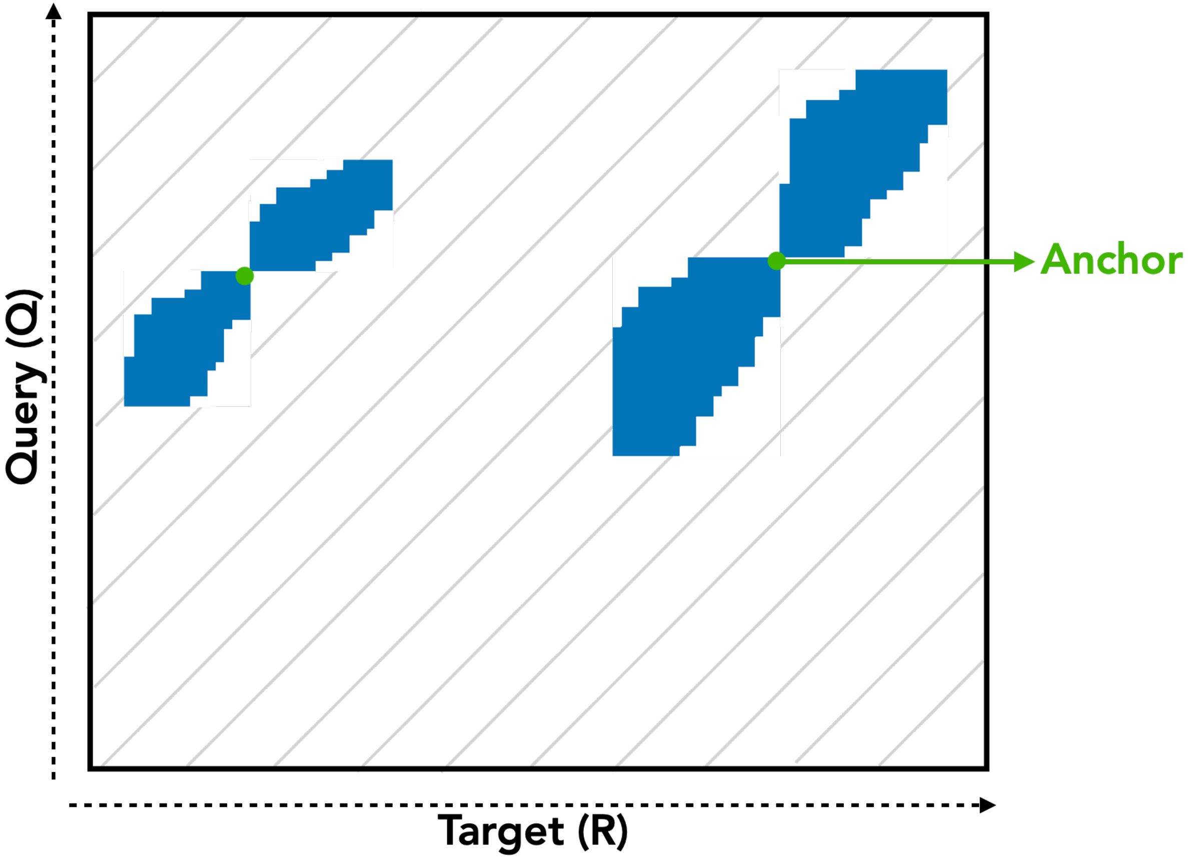
Extend



Extension results in the final alignments

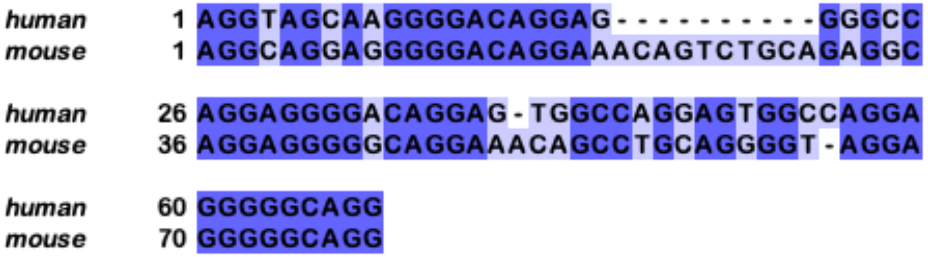
Dynamic Programming Equations

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

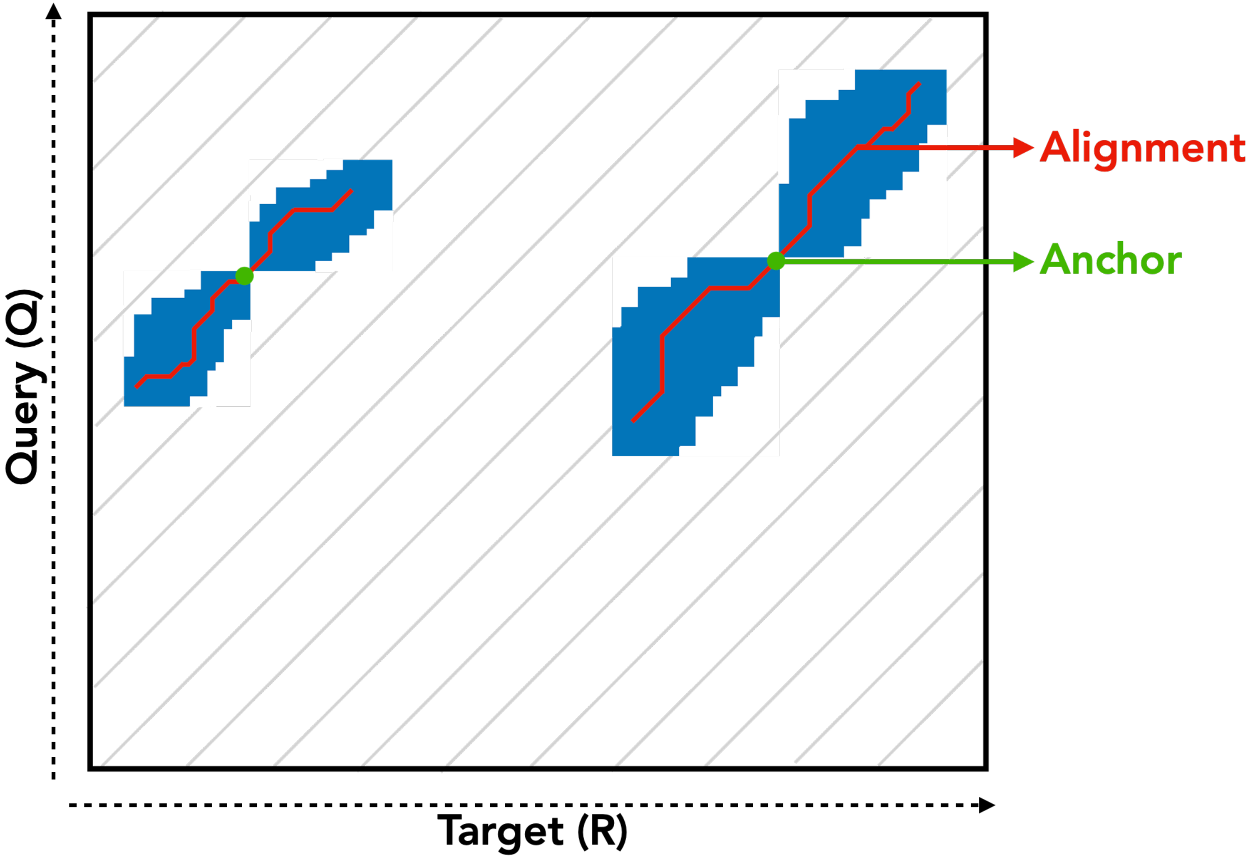


Extension results in the final alignments

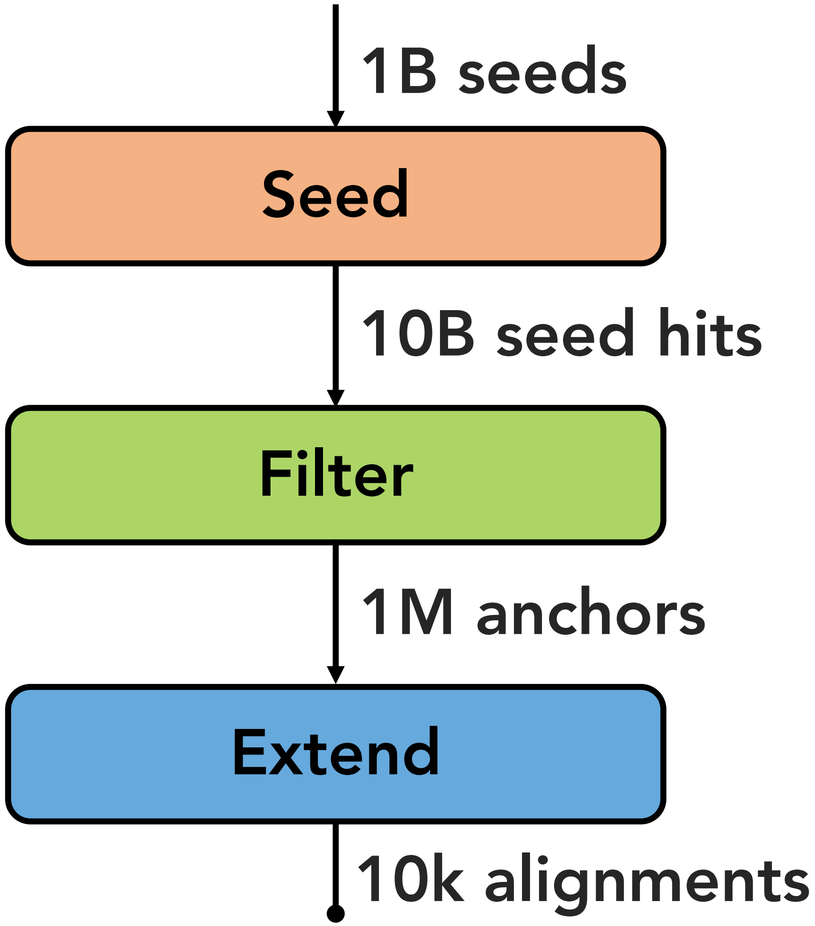
Alignment

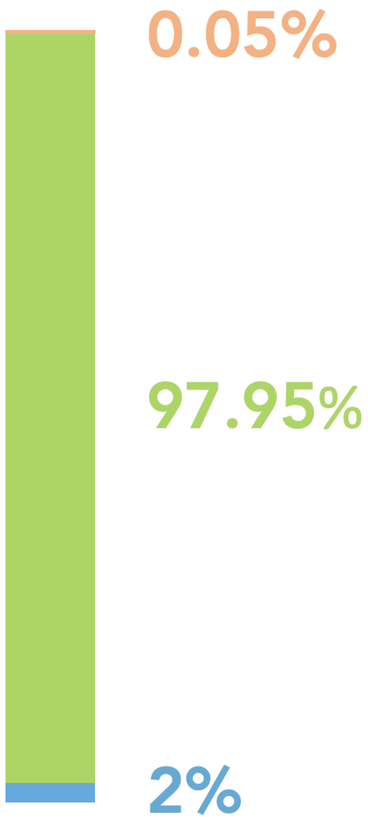


```
human    1 AGGTAGCAAGGGGACAGGAG----------GGGCC
mouse    1 AGGCAGGAGGGGGACAGGAAACAGTCTGCAGAGGC

human   26 AGGAGGGGACAGGAG-TGGCCAGGAGTGGCCAGGA
mouse   36 AGGAGGGGGCAGGAAACAGCCTGCAGGGGT-AGGA

human   60 GGGGGCAGG
mouse   70 GGGGGCAGG
```



Filtering stage dominates the runtime

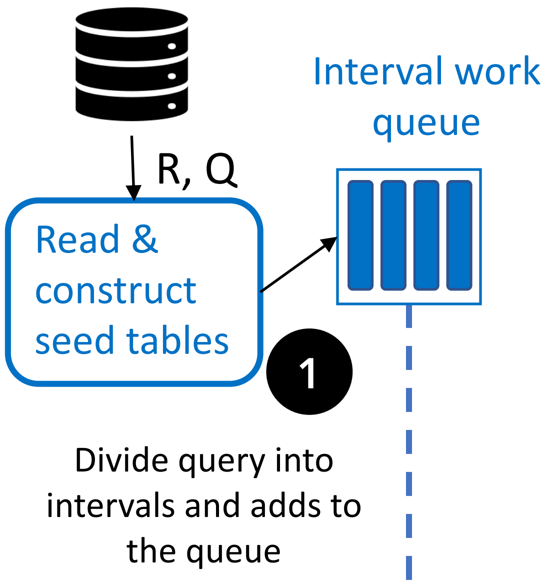




Runtime distribution per stage

System Overview – Genome Sequence to Query intervals

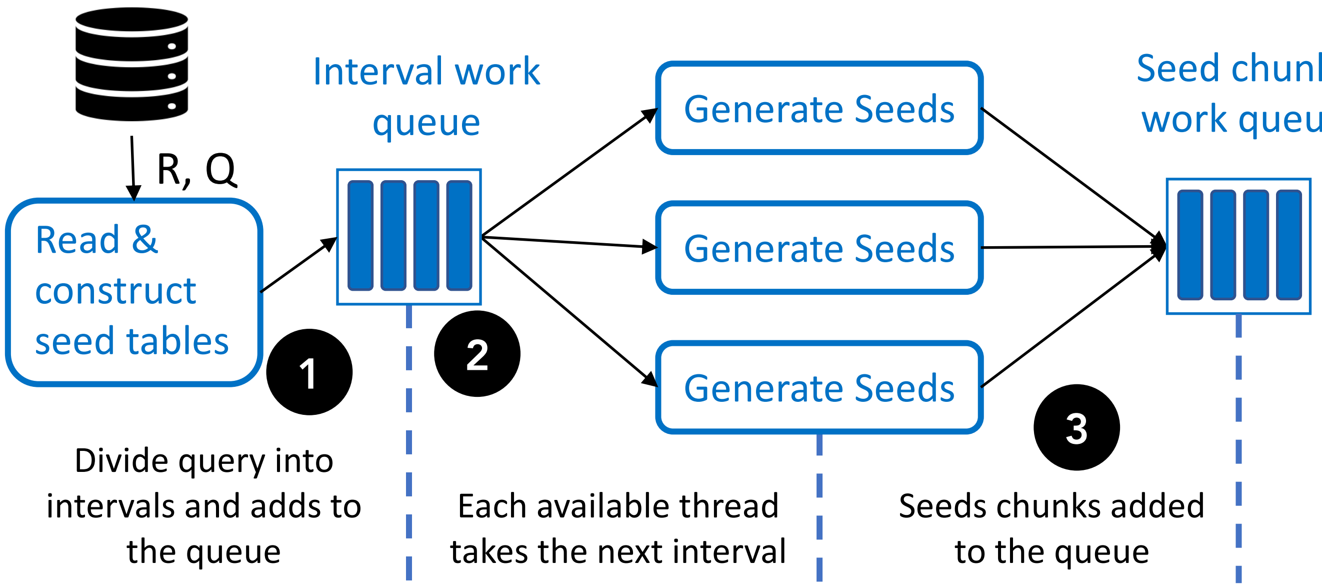
□ CPU □ GPU



Divide query into intervals and adds to the queue

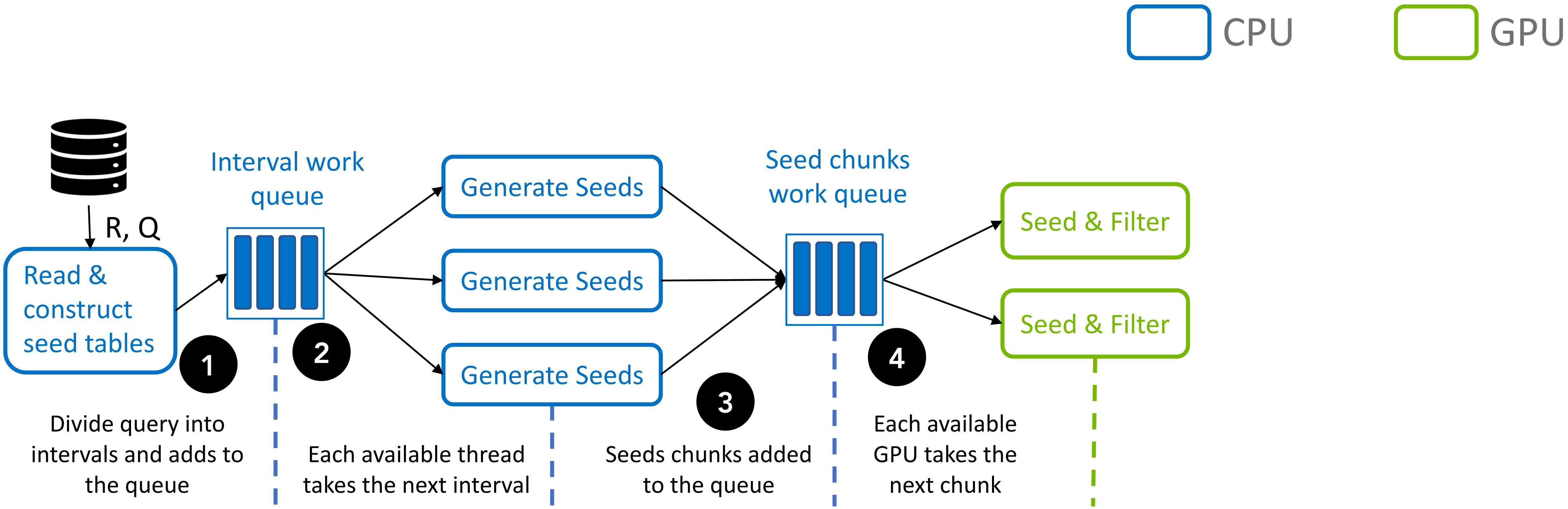
System Overview - Query intervals to Seed chunks

□ CPU □ GPU



1. Divide query into intervals and adds to the queue
2. Each available thread takes the next interval
3. Seeds chunks added to the queue

System Overview



1. Divide query into intervals and adds to the queue

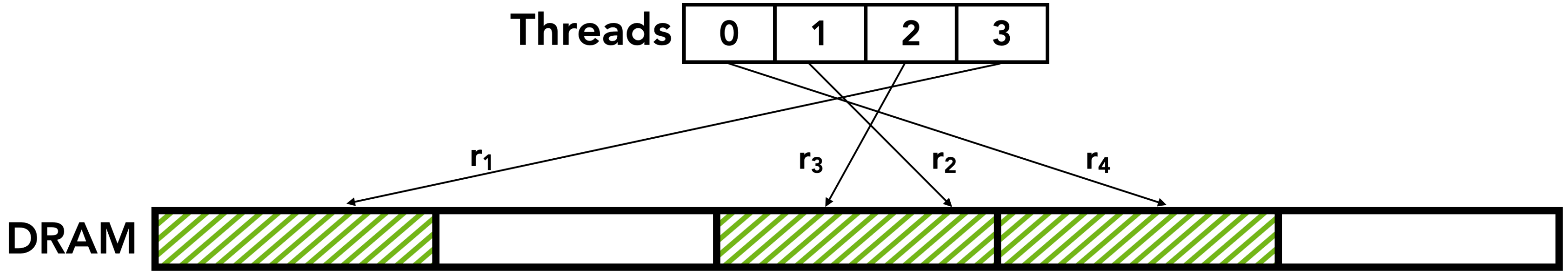
2. Each available thread takes the next interval

3. Seeds chunks added to the queue

4. Each available GPU takes the next chunk

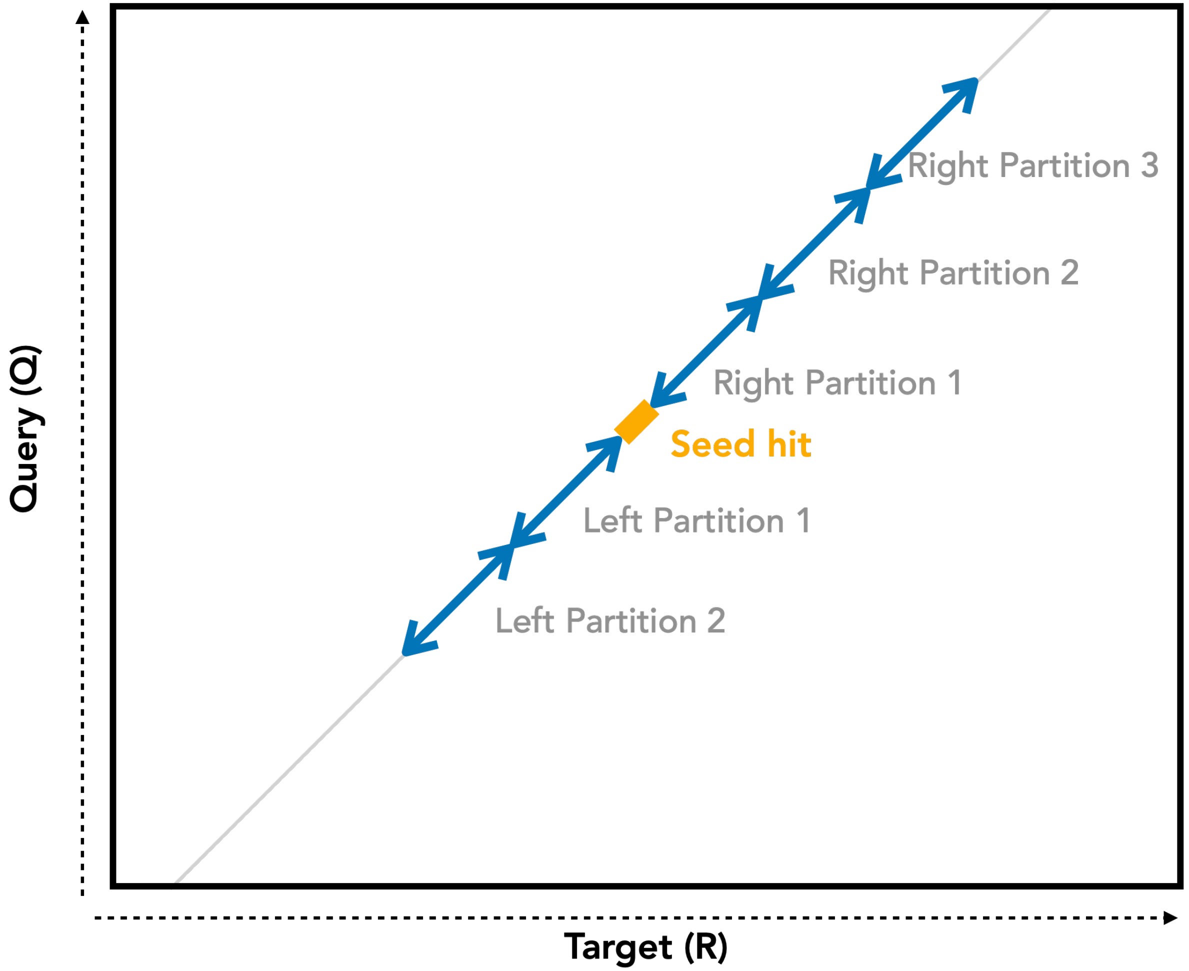
Naïve approach allocates 1 seed hit per thread

1. Considerably varying seed hit positions cause inefficient uncoalesced memory accesses



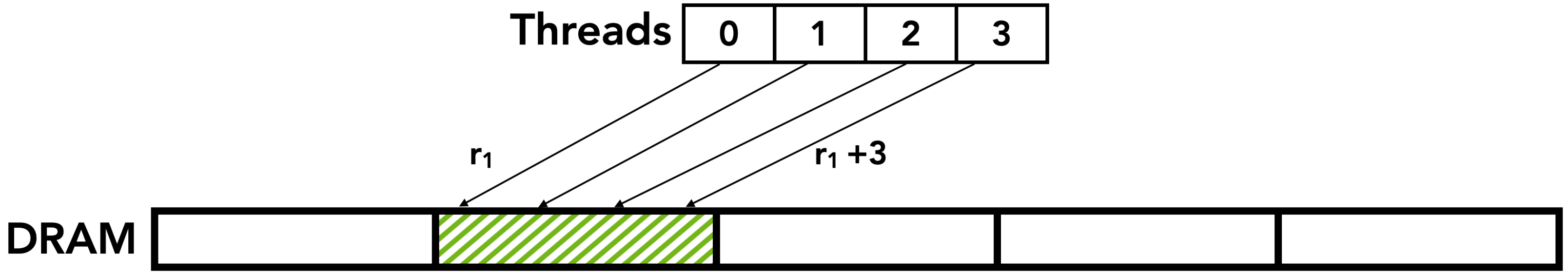
2. Divergent branches within a warp due to the dynamic X-drop condition for each thread

SegAlign allocates 1 seed hit per thread warp



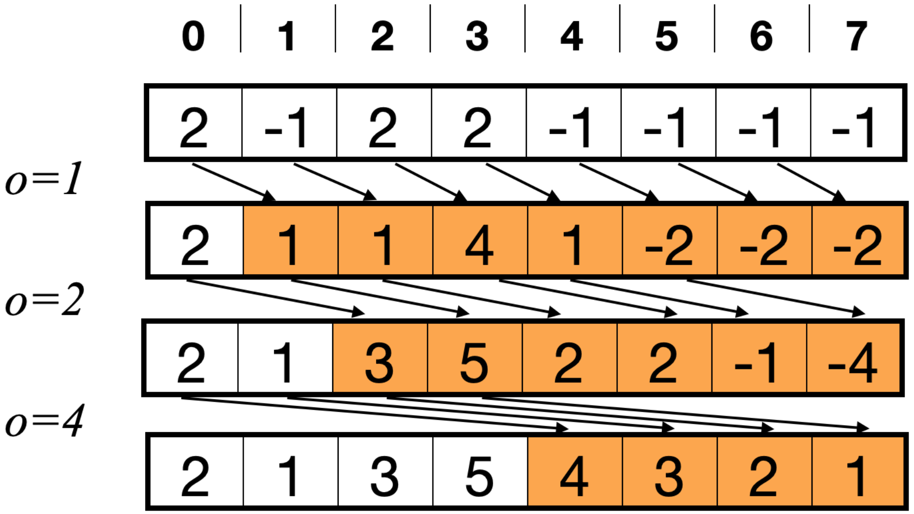
1 seed hit per thread warp results in high GPU DRAM bandwidth efficiency

- Efficient bandwidth gains with coalesced memory accesses



Exploiting data locality within each partition

R	A	A	G	T	C	A	A	T
Q	A	T	G	T	A	T	T	C
Score	2	-1	2	2	-1	-1	-1	-1



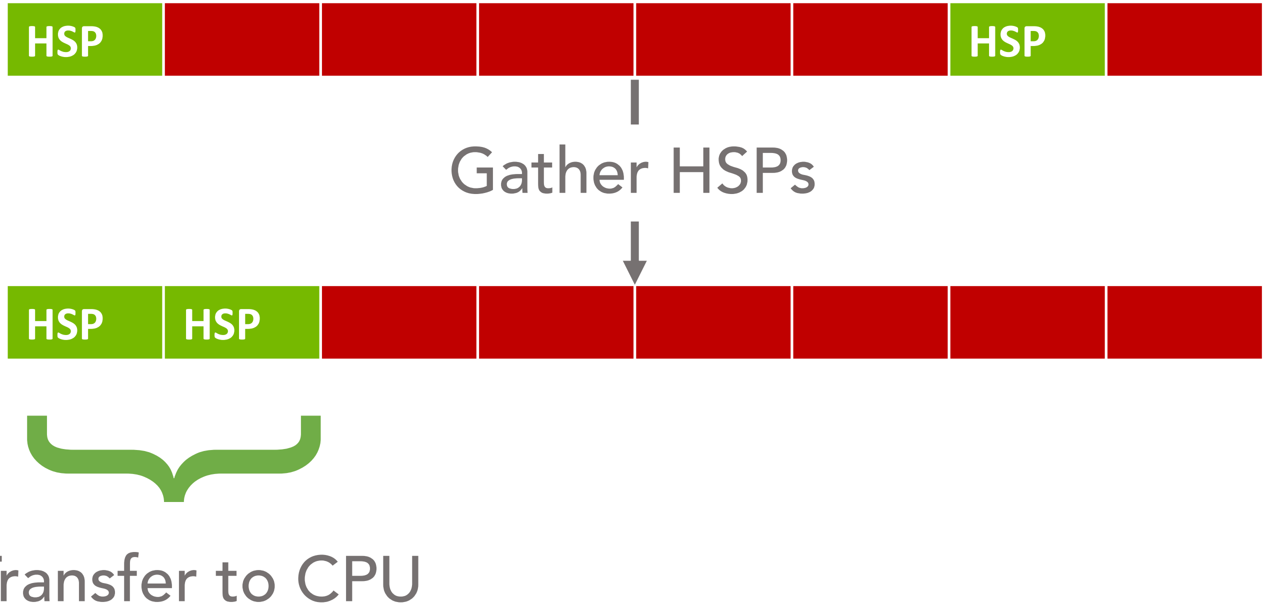
	0	1	2	3	4	5	6	7
	2	-1	2	2	-1	-1	-1	-1
o=1	2	1	1	4	1	-2	-2	-2
o=2	2	1	3	5	2	2	-1	-4
o=4	2	1	3	5	4	3	2	1

Hillis and Steele (1986)

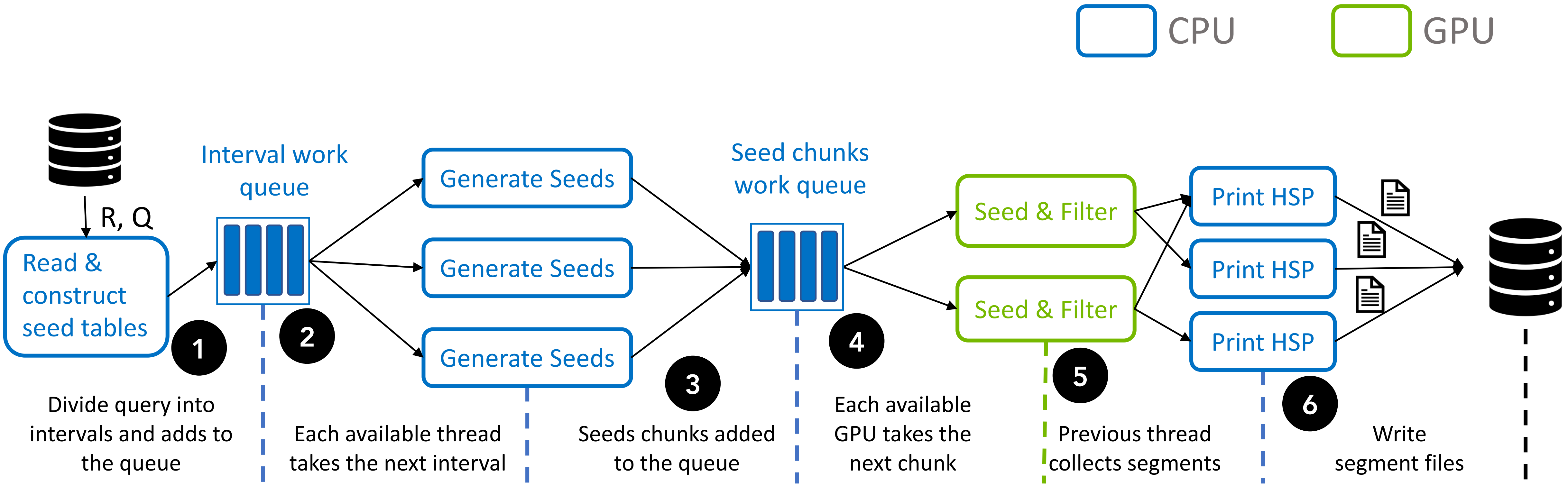
Reducing GPU-CPU communication time

- 1 in 10,000 segment pairs qualify for extension

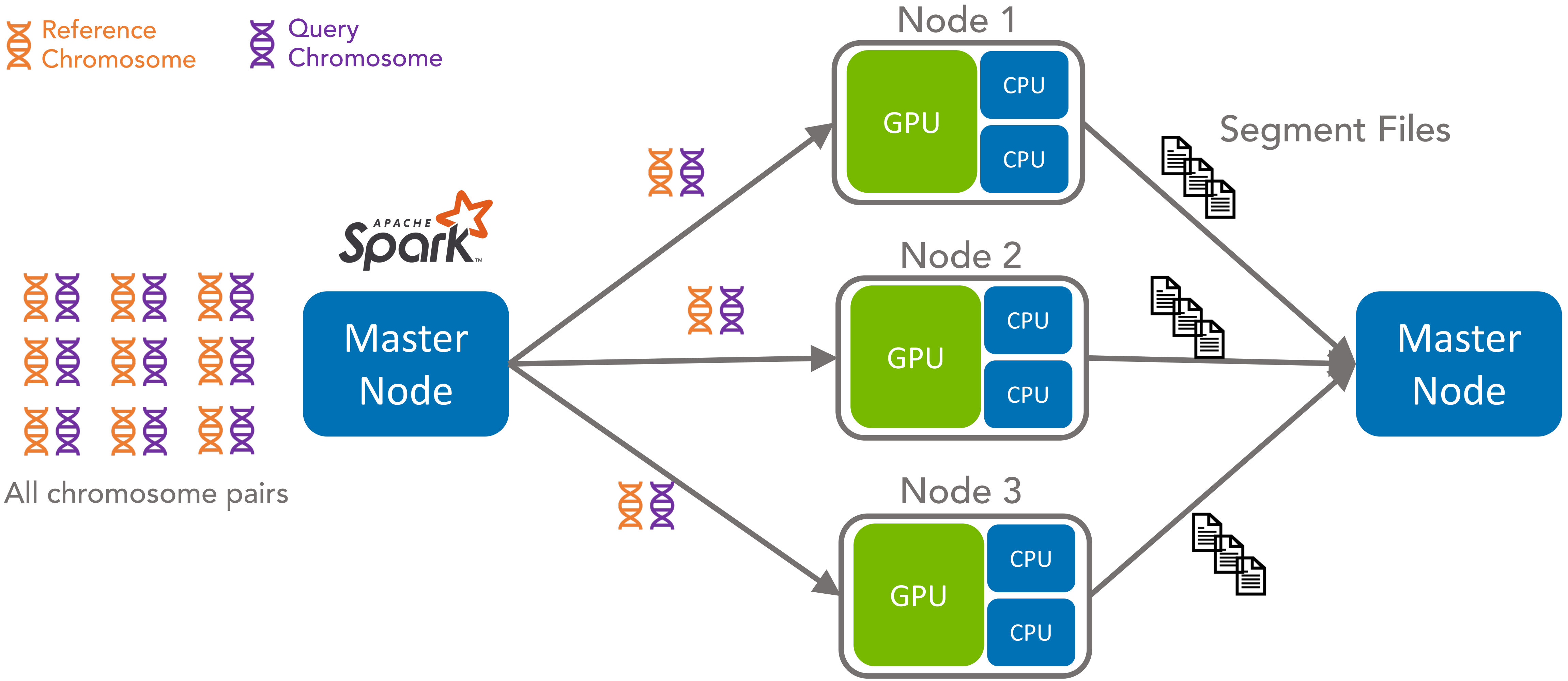
- HSPs are gathered in contiguous memory



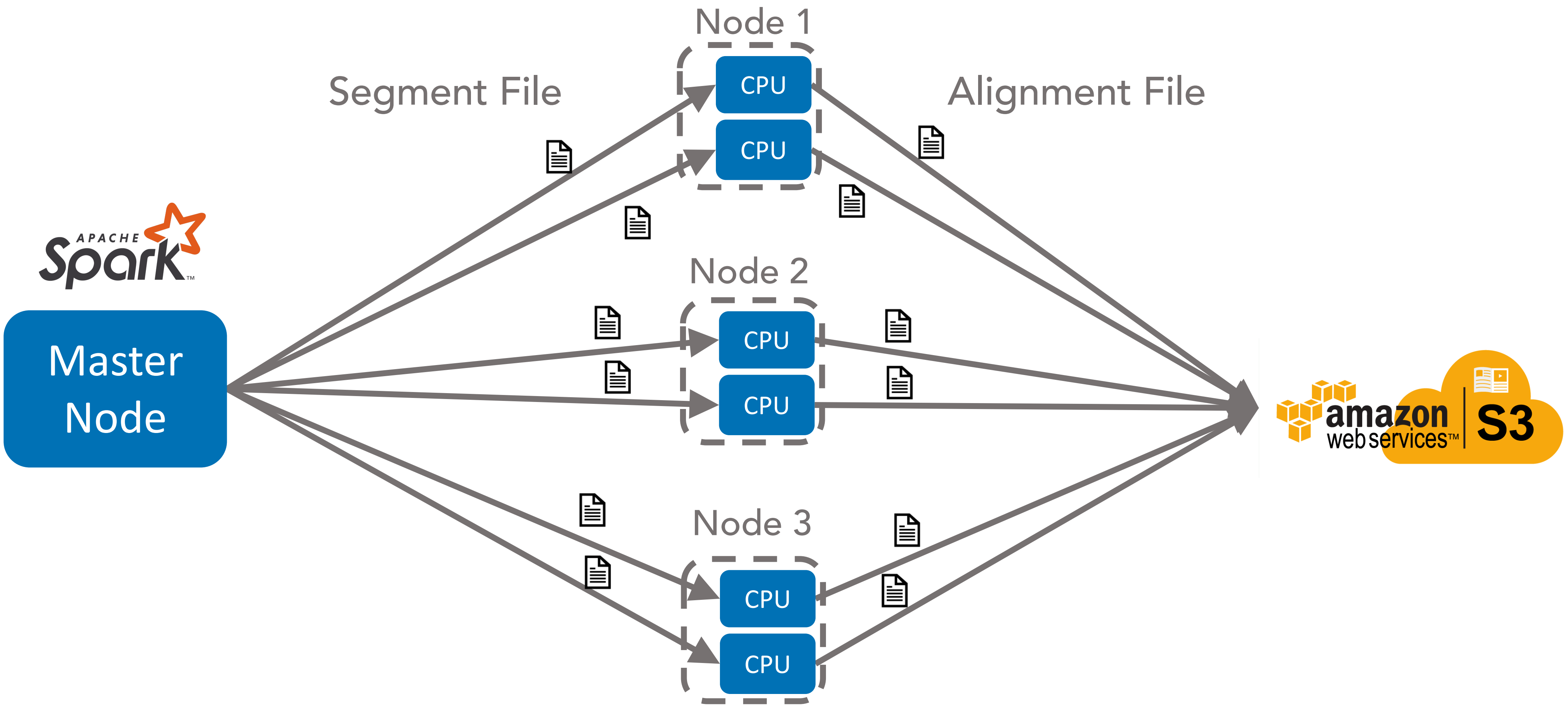
System Overview – HSP to final alignments


CPU
GPU
R, Q
Read & construct seed tables
Interval work queue
Generate Seeds
Generate Seeds
Generate Seeds
Seed chunks work queue
Seed & Filter
Seed & Filter
Print HSP
Print HSP
Print HSP
1
Divide query into intervals and adds to the queue
2
Each available thread takes the next interval
3
Seeds chunks added to the queue
4
Each available GPU takes the next chunk
5
Previous thread collects segments
6
Write segment files

Multi-node version: Seed-and-Filter phase



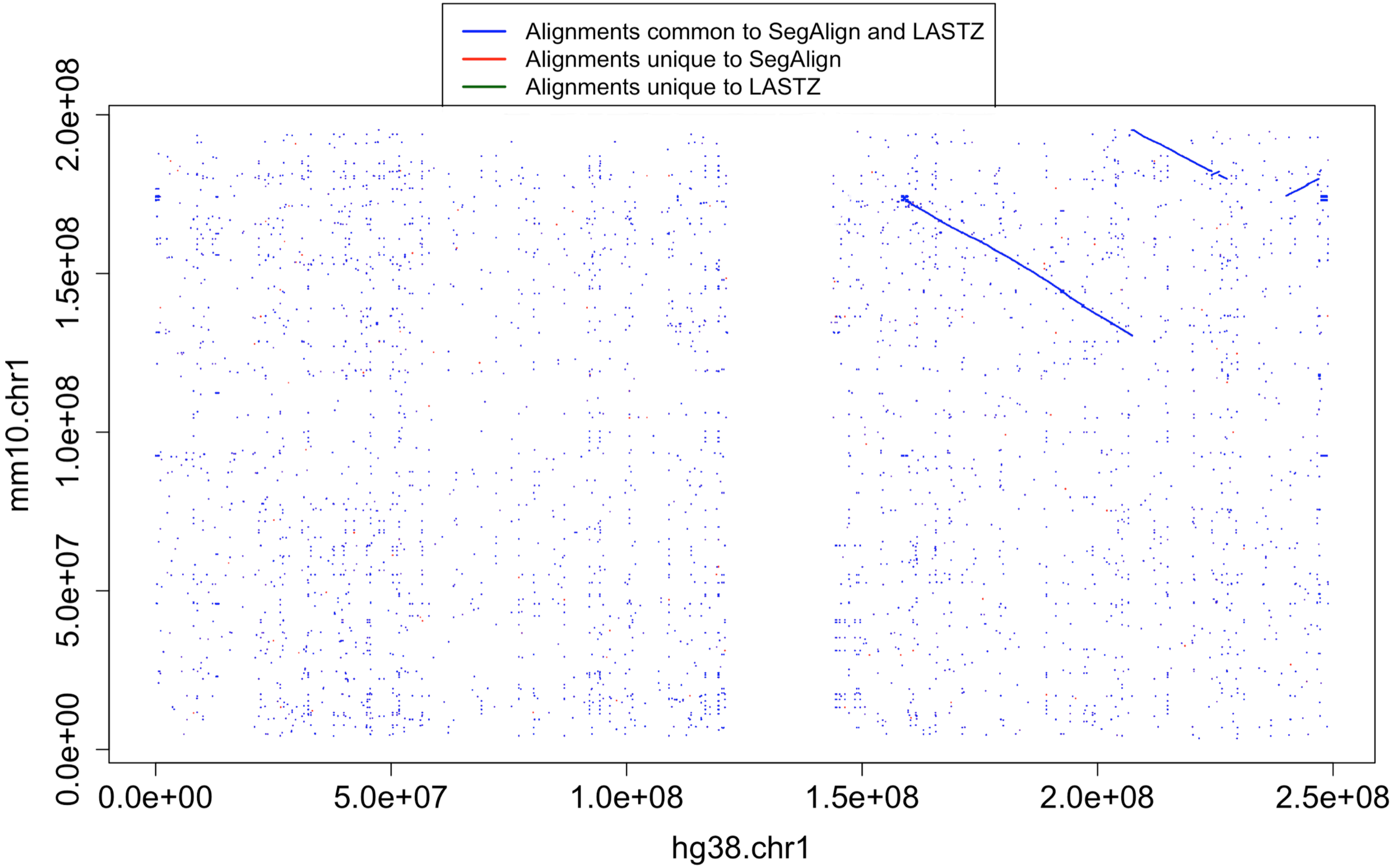
Multi-node version: Extension phase



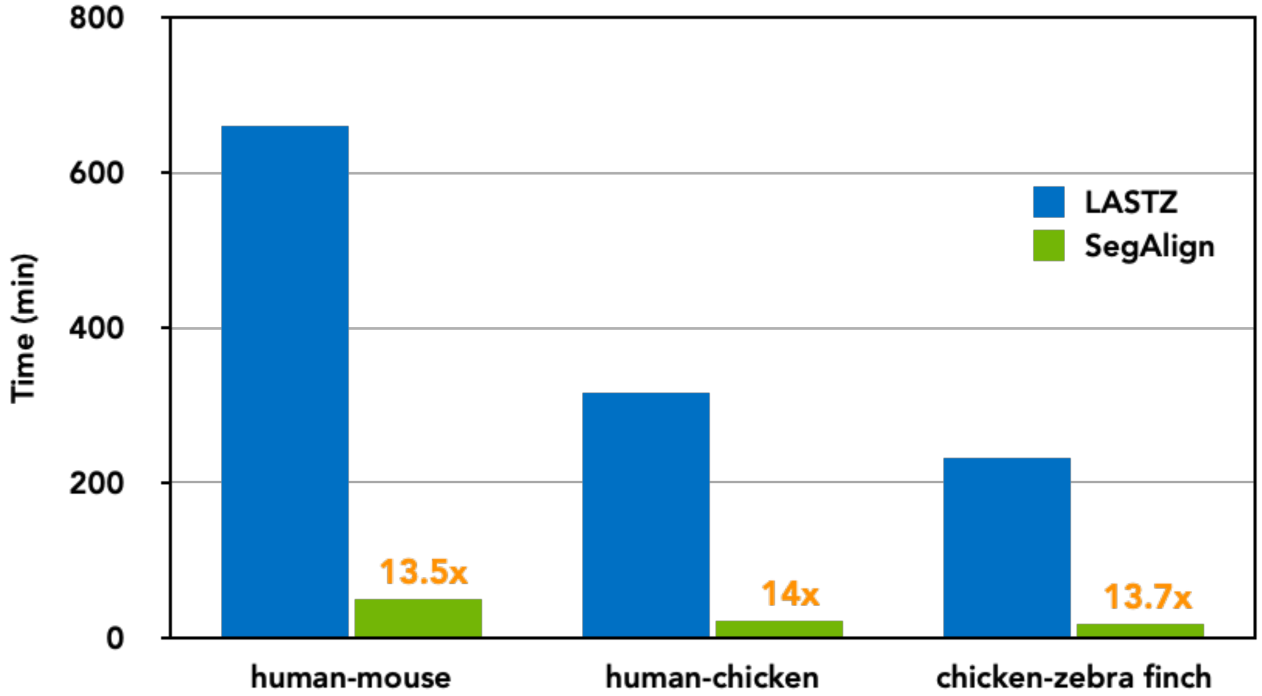
SegAlign generates all the LASTZ alignments, and more…

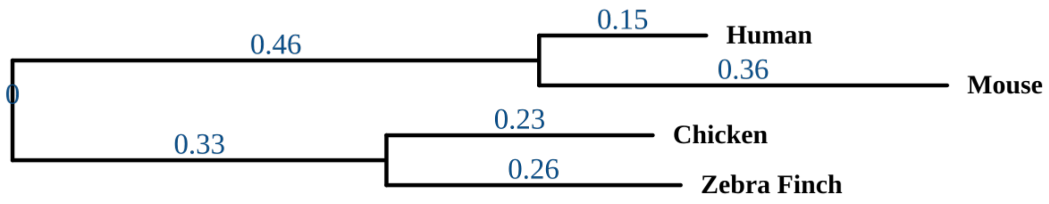
Few alignments unique to SegAlign

No alignments unique to LASTZ



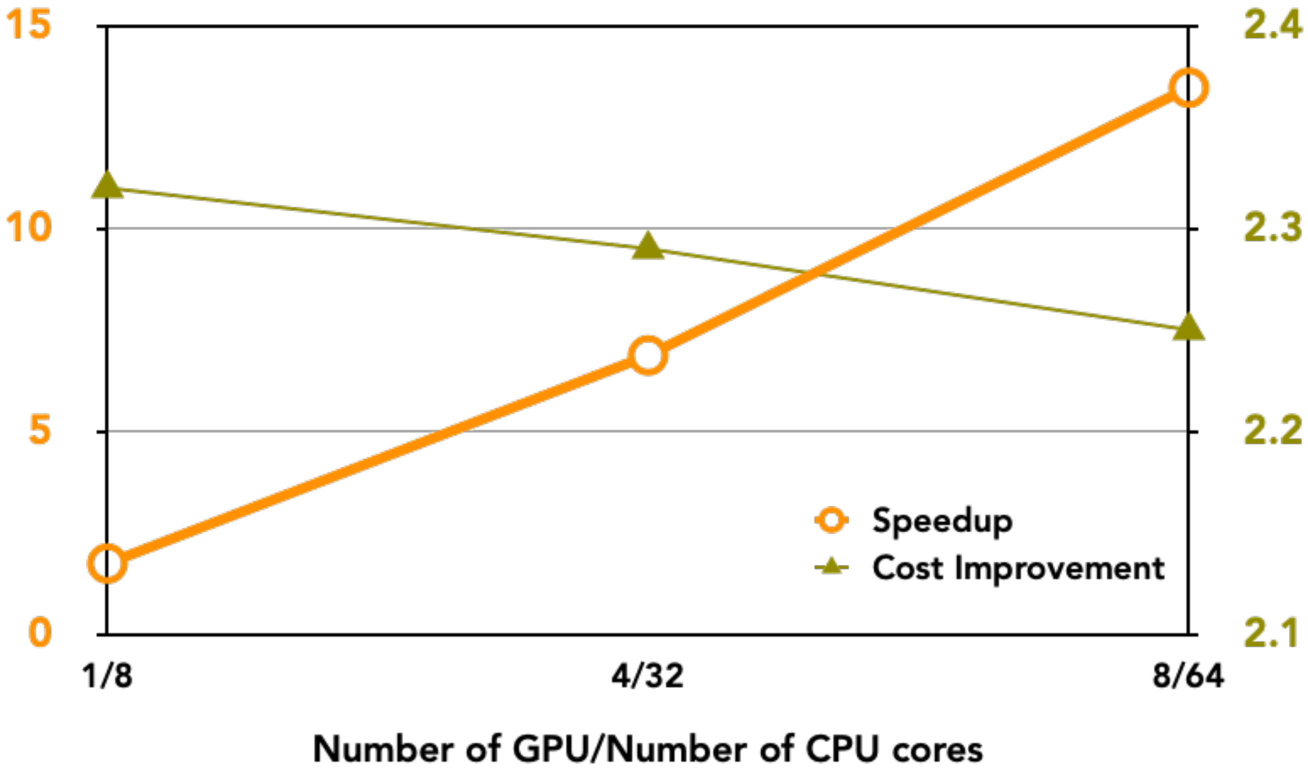
13x-14x speedup across different species pairs



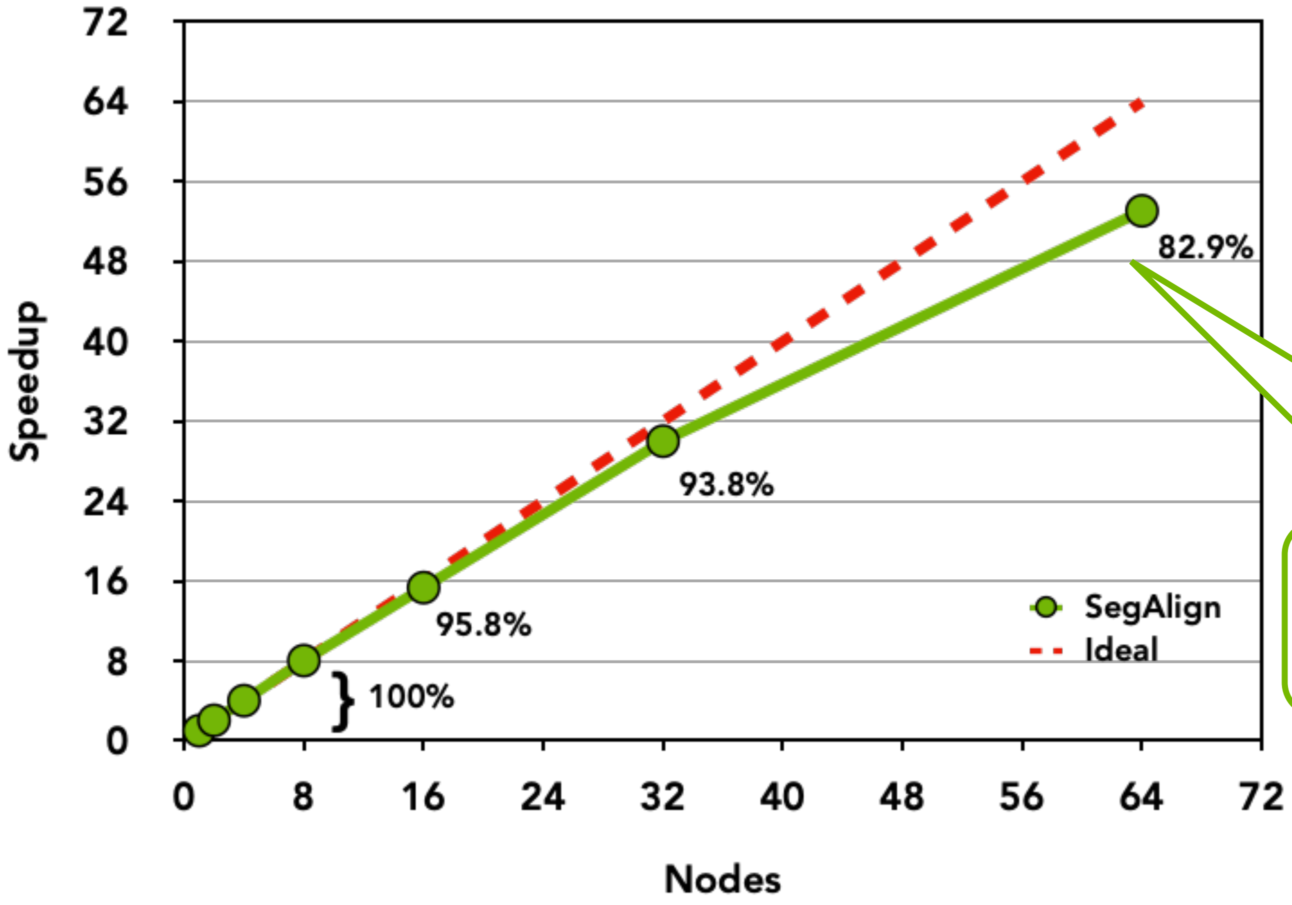


	HW config	AWS Instance
LASTZ	96 CPU cores	c5.24xlarge
SegAlign	8 V100 GPU 96 CPU cores	p3.16xlarge

Runtime and Cost Comparison for human-mouse WGA



Strong scaling efficiency of 93.8%



Parallel slack starts dominating

Each node consists of 1 V100 GPU + 8 cores

Weak scaling efficiency of 97.9%

Genome Size (Mbp)	#nodes	Time	Efficiency
195	1	44m 25s	100%
390	2	44m 27s	99.9%
780	4	44m 43s	99.3%
1560	8	45m 0s	98.7%
3120	16	45m 20s	98.0%
6240	32	45m 23s	97.9%
12480	64	46m 5s	96.4%

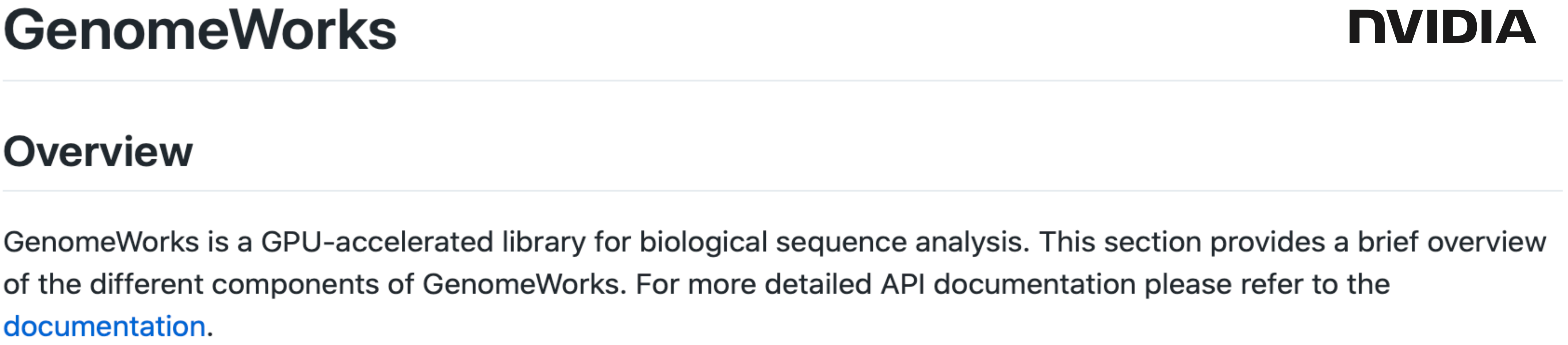
Each node consists of 1 V100 GPU + 8 cores

Communication delays start dominating

SegAlign's Ungapped extension kernel now in NVIDIA GenomeWorks library

https://github.com/clara-parabricks/GenomeWorks

NVIDIA



GenomeWorks

Overview

GenomeWorks is a GPU-accelerated library for biological sequence analysis. This section provides a brief overview of the different components of GenomeWorks. For more detailed API documentation please refer to the documentation.

NVIDIA team: Joyjit Daw, Ashutosh Tadkase, Andreas Hahn, Johnny Israeli, George Vacek

SegAlign for 1000+ way vertebrate alignment

SegAlign-integrated Cactus multiple genome aligner will be used to generate the pairwise alignments for the **1000+ vertebrate multiple alignment** at UCSC, and reduce the compute time from **months to days**

Progressive alignment with Cactus: a multiple-genome aligner for the thousand-genome era

To appear in Nature soon

Joel Armstrong, Glenn Hickey, Mark Diekhans, Alden Deran, Qi Fang, Duo Xie, Shaohong Feng, Josefin Stiller, Diane Genereux, Jeremy Johnson, Voichita Dana Marinescu, David Haussler, Jessica Alföldi, Kerstin Lindblad-Toh, Elinor Karlsson, Guojie Zhang, Benedict Paten

doi: https://doi.org/10.1101/730531

Acknowledgements: Glenn Hickey, Bob Harris, Mark Diekhans

Armstrong et al. *bioRxiv* (2019)
To appear in Nature soon

Conclusion

- SegAlign is a GPU-based system for pairwise whole genome alignment that
  - can serve as a **drop-in replacement** for LASTZ
  - provides **14x** improvement in speed over LASTZ
  - provides **2.2x** improvement in cost

- SegAlign's multi-node implementation has strong scaling efficiency of **93.8%** and a weak scaling efficiency of **97.9%**

https://github.com/gsneha26/SegAlign