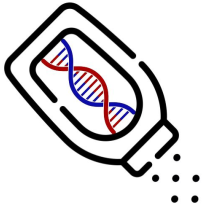
UC San Diego
Electrical and Computer Engineering
JACOBS SCHOOL OF ENGINEERING

TURAKHIA LAB



TALCO: Tiling Genome Sequence Alignment using Convergence of Traceback Pointers

Sumit Walia, Cheng Ye, Arkid Bera, Dhruvi Lodhavia and Yatish Turakhia
University of California San Diego

Best Paper Nominee, HPCA’24

Outline

- Current genome sequence alignment **algorithms**, **accelerators** and their **limitations**
- **TALCO**: A tiling technique based on convergence of traceback pointers for genome sequence alignment
- **Key Contributions** and **Results**

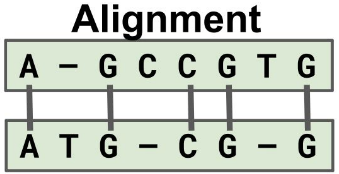
Outline

- Current genome sequence alignment **algorithms**, **accelerators** and their **limitations**
- TALCO: A tiling technique based on convergence of traceback pointers for genome sequence alignment
- Key Contributions and Results

Genome Sequence Alignment

- **Sequence alignment:** Given two sequences, Reference (R) and Query (Q), assign gaps (“**–**”) in R and Q to produce a valid alignment that maximizes the alignment score

R: AGCCGTG
Q: ATGCGG



Broad Classification of Alignment Algorithms

Classical Dynamic Programming (DP) Algorithms

Ex: *Needleman-Wunsch, Smith-Waterman*

Non-Classical Algorithms

Ex: *WFA, Myers' O(ND)*

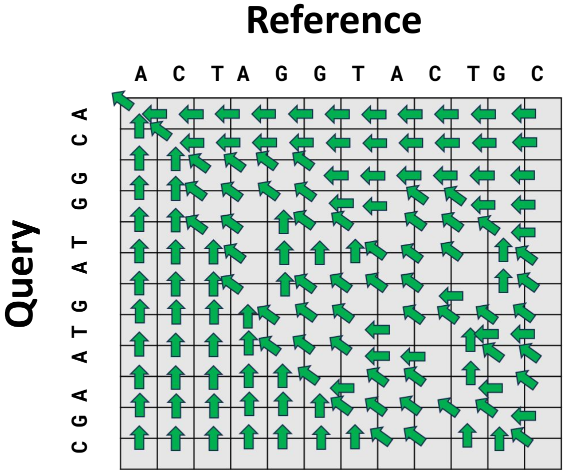
Classical DP based Alignment Algorithms

Classical Dynamic Programming (DP) Algorithms

Non-Classical Algorithms

Ex: *Needleman-Wunsch, Smith-Waterman*

Ex: *WFA, Myers' O(ND)*



1. **Matrix Fill (Store traceback pointers)**

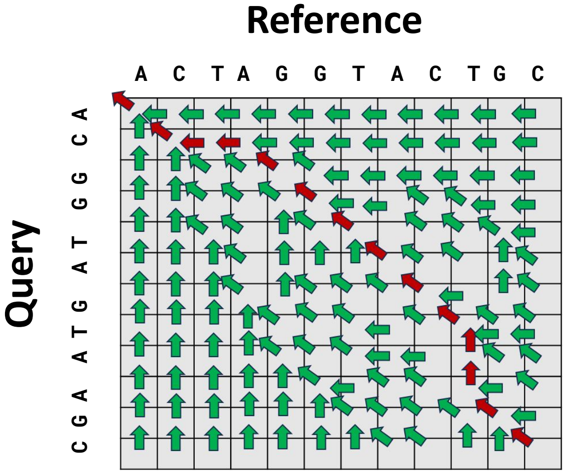
Classical DP based Alignment Algorithms

Classical Dynamic Programming (DP) Algorithms

Ex: *Needleman-Wunsch, Smith-Waterman*

Non-Classical Algorithms

Ex: *WFA, Myers’ O(ND)*



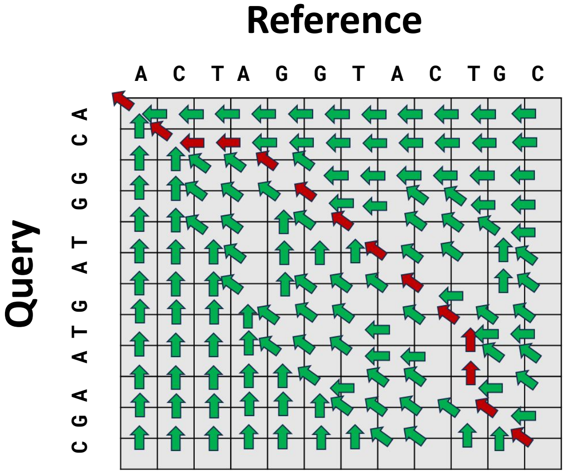
```
A C T A G G T A C T - - G C
A C - - G G T A G T A A G C
```

1. Matrix Fill (Store traceback pointers)
2. **Optimal traceback path**

Non-Classical Alignment Algorithms

Classical Dynamic Programming (DP) Algorithms

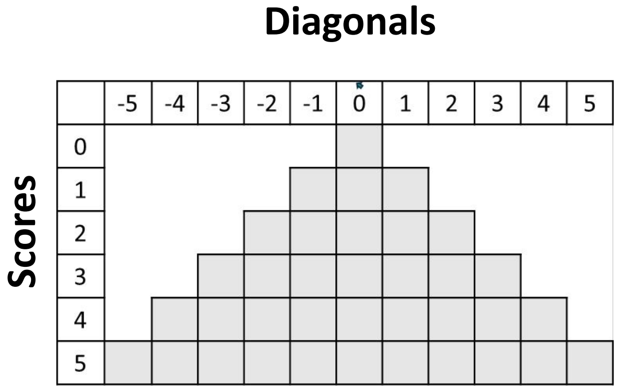
Ex: *Needleman-Wunsch, Smith-Waterman*



1. Matrix Fill (Store traceback pointers)
2. Optimal traceback path

Non-Classical Algorithms

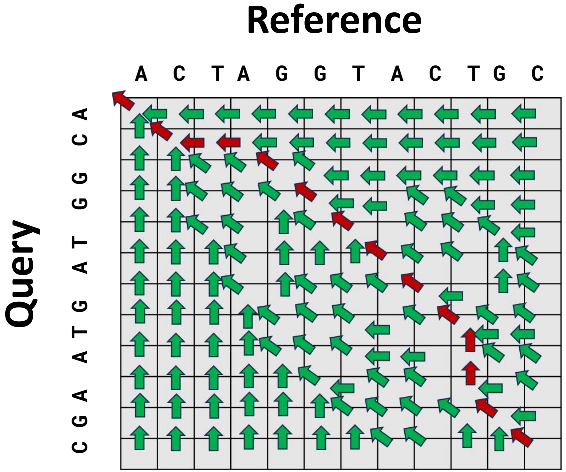
Ex: *WFA, Myers' O(ND)*



Non-Classical Alignment Algorithms

Classical Dynamic Programming (DP) Algorithms

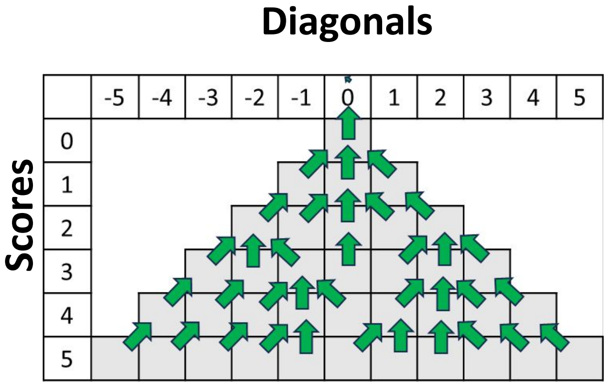
Ex: *Needleman-Wunsch, Smith-Waterman*



1. Matrix Fill (Store traceback pointers)
2. Optimal traceback path

Non-Classical Algorithms

Ex: *WFA, Myers' O(ND)*



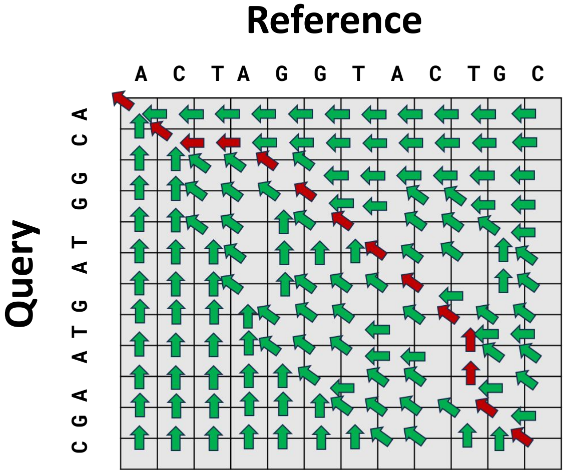
Penalties: mismatch=1, gap=1

1. **Matrix Fill (Store traceback pointers)**

Non-Classical Alignment Algorithms

Classical Dynamic Programming (DP) Algorithms

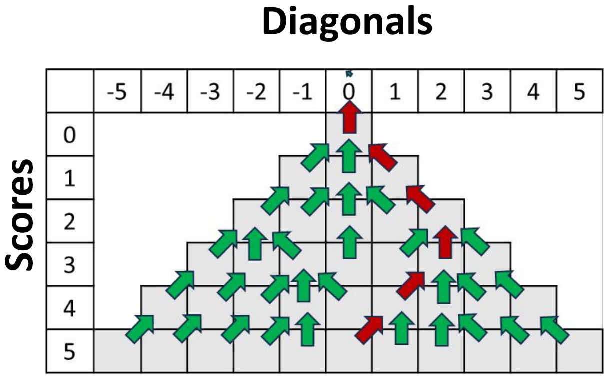
Ex: *Needleman-Wunsch, Smith-Waterman*



1. Matrix Fill (Store traceback pointers)
2. Optimal traceback path

Non-Classical Algorithms

Ex: *WFA, Myers' O(ND)*



Penalties: mismatch=1, gap=1

1. Matrix Fill (Store traceback pointers)
2. **Optimal traceback path**

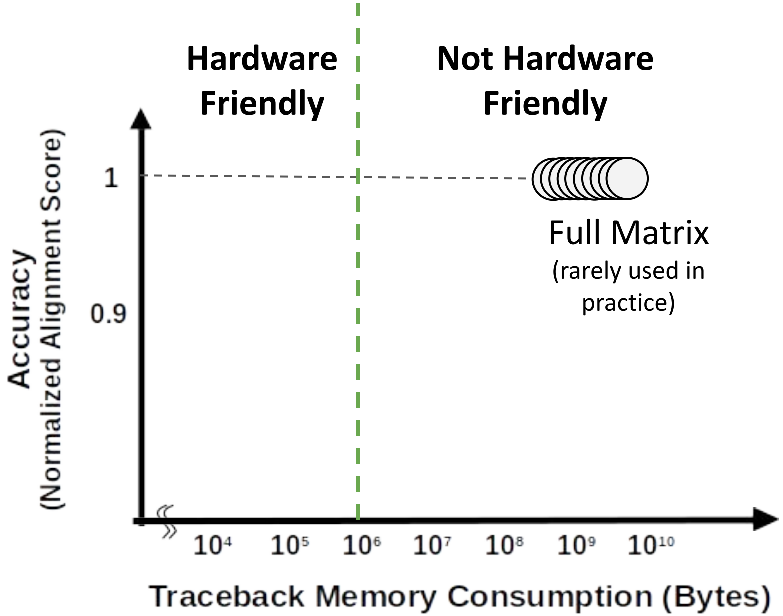
Comparison: Classical-DP and Non-Classical

Classical Dynamic Programming (DP) Algorithms	**Non-Classical Algorithms**
Ex: *Needleman-Wunsch, Smith-Waterman*	Ex: *WFA, Myers' O(ND)*

Both categories of algorithms produce optimal alignments

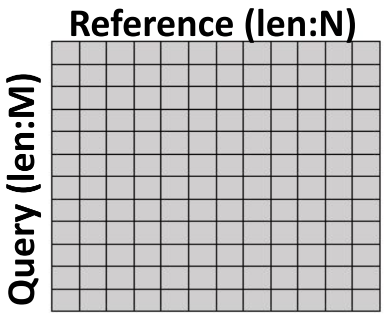
Uniform dependencies **Easier to accelerate**	Non-Uniform dependencies **Harder to accelerate**
More popular	Very Fast for similar sequences

Full Matrix Sequence Alignment Algorithms



Classical-DP

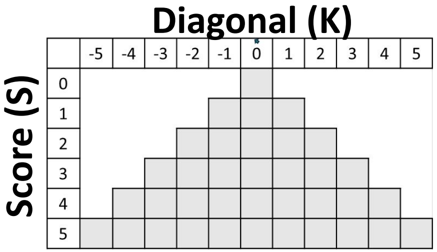
▨ Computed cells
☐ Uncomputed cells



Space Complexity O(NM)

Ex: *Needleman-Wunsch*

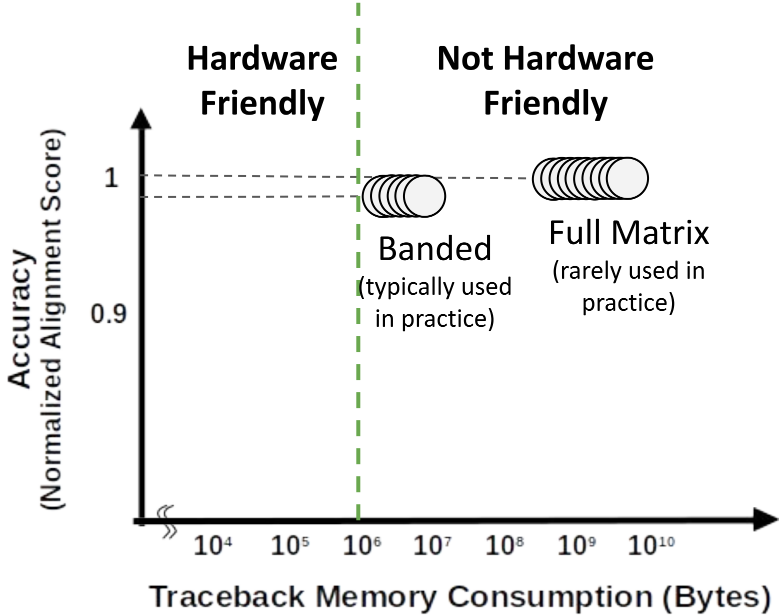
Non Classical



Space Complexity $O(S^2)$

Ex: *WFA*

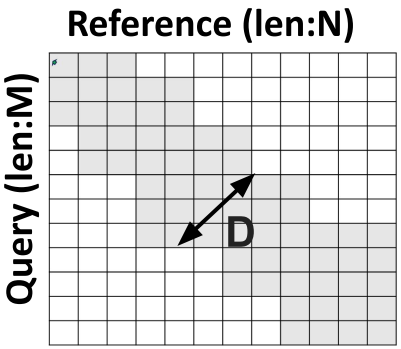
Banded Sequence Alignment Algorithms



Classical-DP

▨ Computed cells

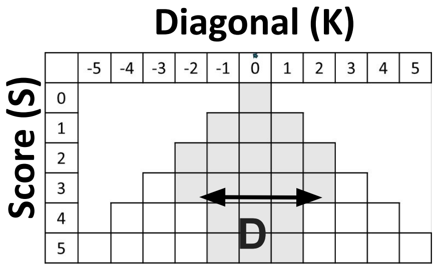
☐ Uncomputed cells



Space Complexity O(ND)

Ex: X-Drop

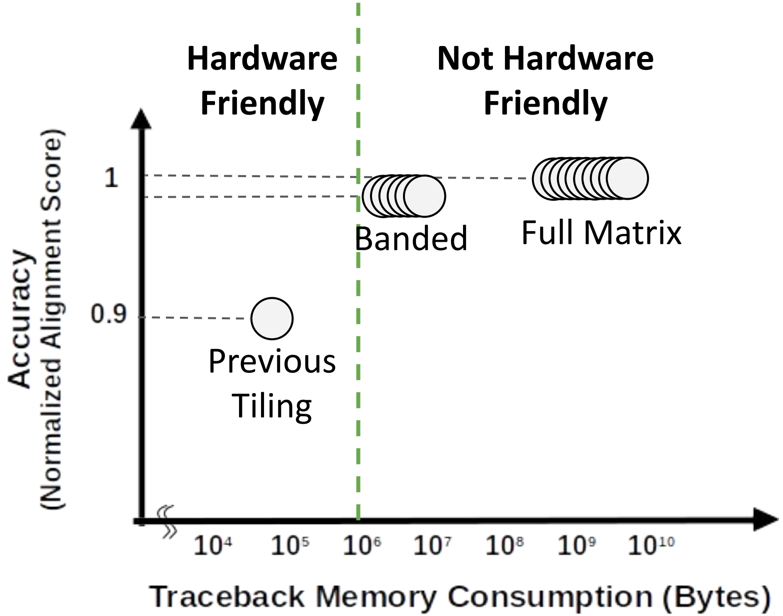
Non Classical



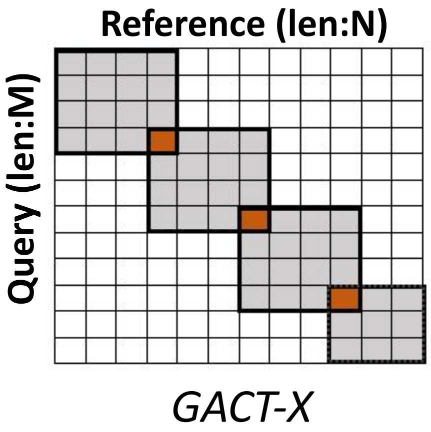
Space Complexity O(SD)

Ex: WFA-Adapt

Tiling heuristic



Classical-DP



▨ **Computed cells**

☐ **Uncomputed cells**

T: Tile size

Space Complexity $O(T^2)$

↑

Independent of sequence length!

Non Classical

Tiling is **never applied** to **non-classical sequence alignment algorithm**

Architecture Papers Adopting Tiling Heuristics

- **GACT** – Darwin: A Genomics Co-processor Provides up to 15,000X Acceleration on Long Read Assembly (**ASPLOS 2018 Best Paper Award**)
- **GACT-X** – Darwin-WGA: A Co-processor Provides Increased Sensitivity in Whole Genome Alignments with High Speedup

Lower accuracy imposes challenges for tiling-based accelerators to be adopted in critical real-world applications (e.g. medical diagnoses)

Van der Auwera, Geraldine A., et al. "From FastQ data to high-confidence variant calls: the genome analysis toolkit best practices pipeline." Current protocols in bioinformatics 43.1 (2013)

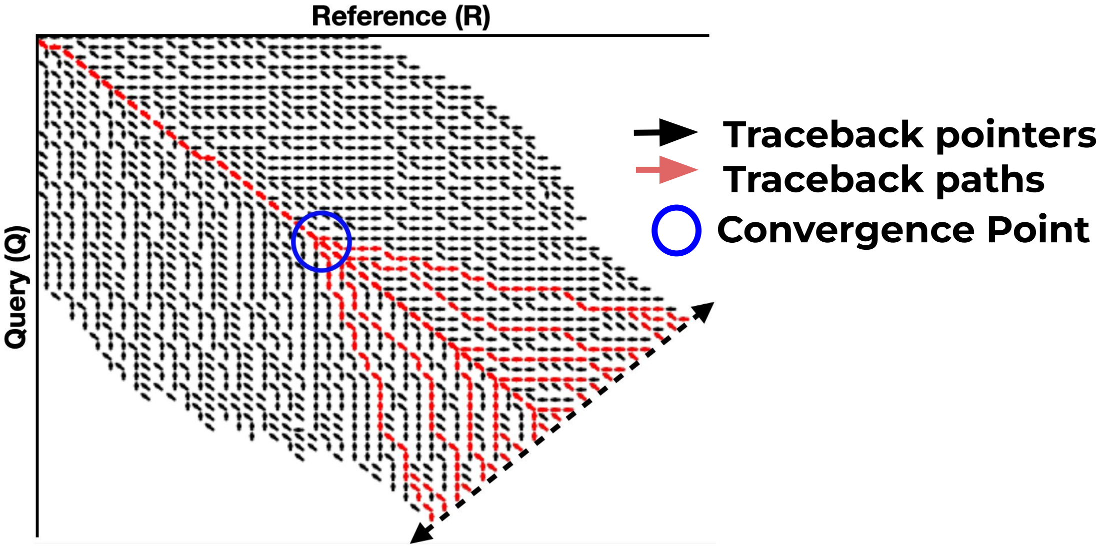
- **RAPIDx**: High-performance ReRAM processing in-memory accelerator for sequence alignment (**TCAD 2023**)
- **GMX: I**nstruction Set Extensions for Fast, Scalable, and Efficient Genome Sequence Alignment **(MICRO 2023)**
- **Scrooge:** a fast and memory-frugal genomic sequence aligner for CPUs, GPUs, and ASICs **(Bioinformatics 2023)**

Outline

- Current genome sequence alignment algorithms, accelerators and their limitations
- **TALCO**: A tiling technique based on convergence of traceback pointers for genome sequence alignment
- Key Contributions and Results

Key Insight: Convergence of Traceback Paths

TALCO (Tiling Long Genome **Al**ignment using **Co**nvergence of Traceback Pointers) is based on the following observation:



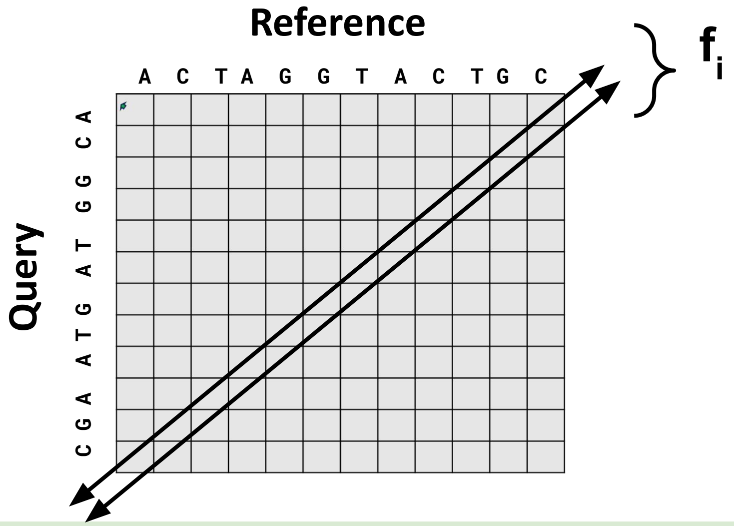
Experiment: Pairwise sequence alignment using Needleman-Wunsch with X-Drop banding

TALCO: Tiling technique for long genome alignment

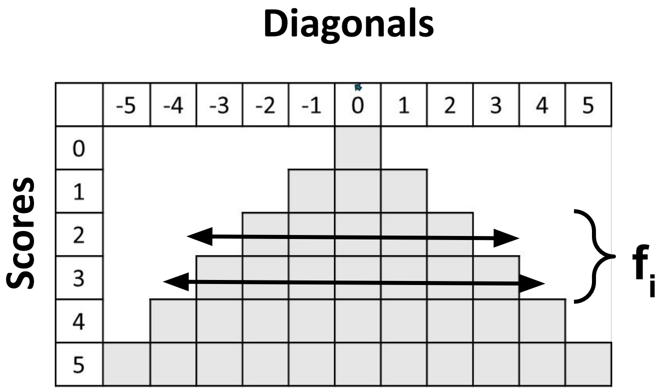
Frontier: The ith frontier (f_i) is a set of cells in the traceback matrix such that the scores and traceback pointers for cells in f_i depend only on the previous frontier, f_{i-1}.

Marker: A marker is a special frontier (f_M) in the TALCO algorithm that separates the two phases

Classical Dynamic Programming (DP) Algorithms



Non-Classical Algorithms



Penalties: mismatch=1, gap open=1, gap extend=1

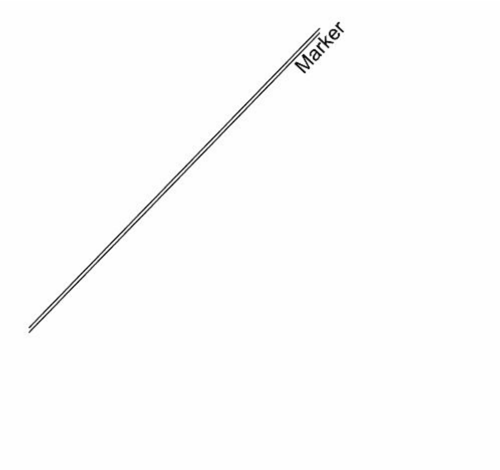
TALCO: Tiling technique for long genome alignment

Frontier: The i^{th} frontier (f_i) is a set of cells in the traceback matrix such that the scores and traceback pointers for cells in f_i depend only on the previous frontier, f_{i-1}.

Marker: A marker is a special frontier (f_M) in the TALCO algorithm that separates the two phases

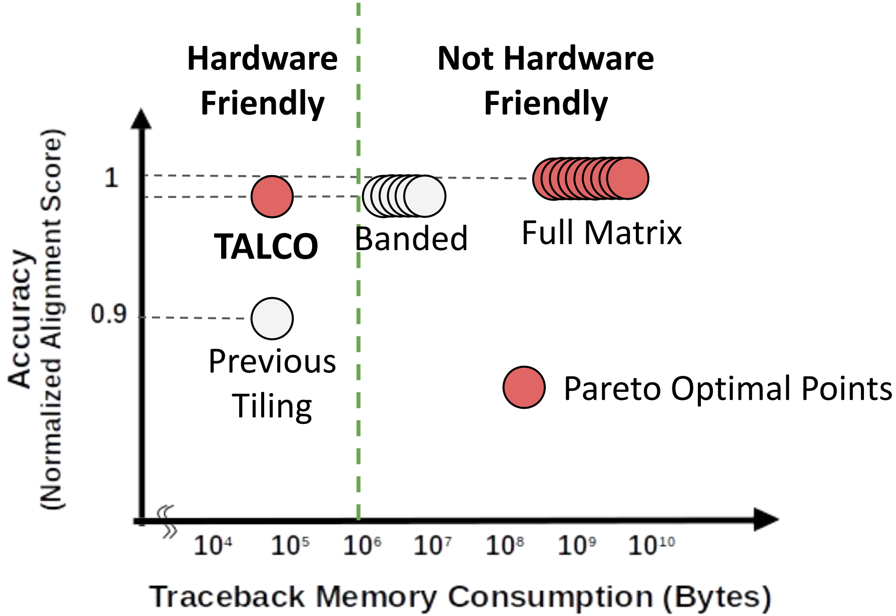
TALCO algorithm has **two** phases:

1. Stores traceback pointers till the **Marker**
2. Find point of convergence of traceback pointers using **pointer-redirection**



TALCO applied to X-Drop Algorithm

TALCO is on the Pareto Optimal Frontier



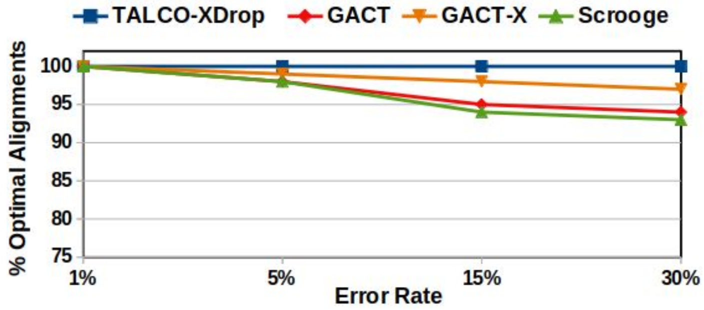
- Constant Space complexity
- Guarantees optimality under banding constraints

Outline

- Current genome sequence alignment algorithms, accelerators and their limitations
- TALCO: A tiling technique based on convergence of traceback pointers for genome sequence alignment
- **Key Contributions** and **Results**

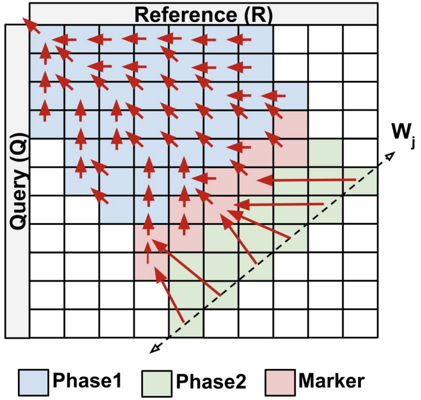
Key Contributions and Results

1. TALCO **guarantees optimality** under banding constraints

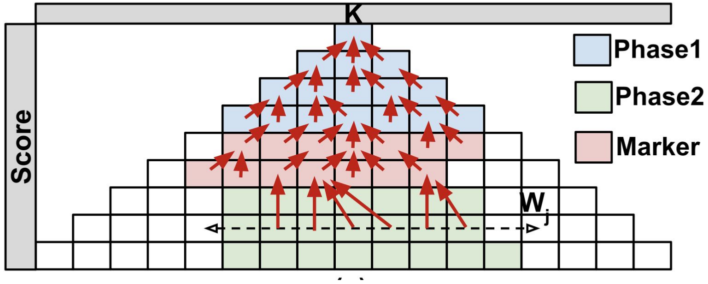


Key Contributions and Results

1. TALCO **guarantees optimality** under banding constraints
2. We applied TALCO to X-Drop (**TALCO- XDrop**) and WFA-Adapt (**TALCO-WFAA**)



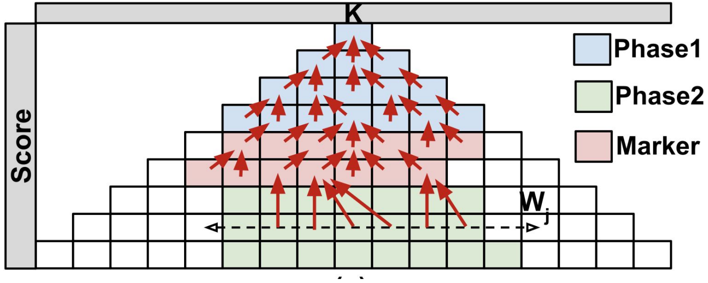
TALCO-XDrop



TALCO-WFAA

Key Contributions and Results

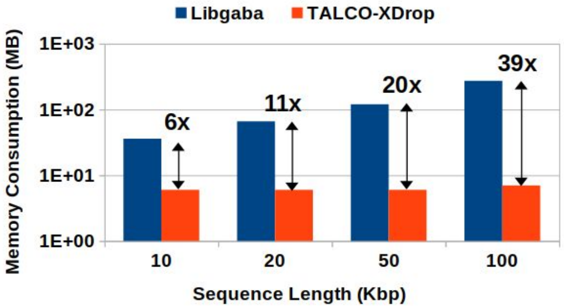
1. TALCO **guarantees optimality** under banding constraints
2. We applied TALCO to X-Drop (**TALCO- XDrop**) and WFA-Adapt (**TALCO-WFAA**)
3. TALCO-WFAA is the **first accelerator** based on the **WFA-Adapt algorithm** capable of performing arbitrary long sequence alignments

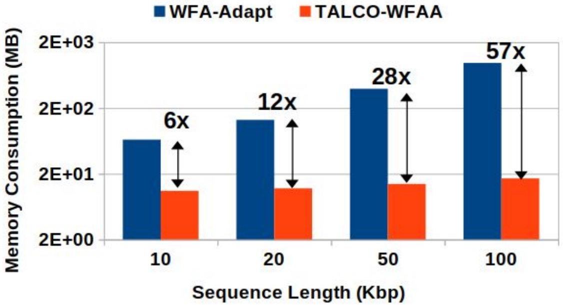


TALCO-WFAA

Key Contributions and Results

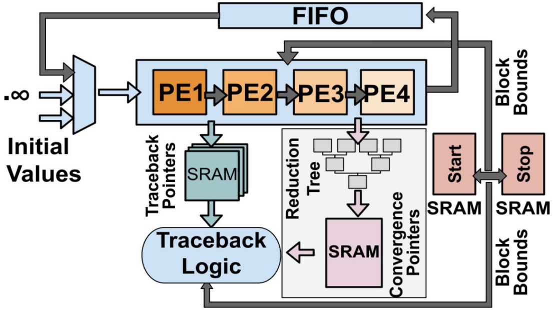
1. TALCO **guarantees optimality** under banding constraints
2. We applied TALCO to X-Drop (**TALCO- XDrop**) and WFA-Adapt (**TALCO-WFAA)**
3. TALCO-WFAA is the **first accelerator** based on the **WFA-Adapt algorithm** capable of performing arbitrary long sequence alignments
4. TALCO-XDrop and TALCO-WFAA (**software**) achieves up to **39x** and **57x** improvement in **memory footprint**, respectively, compared to software baselines



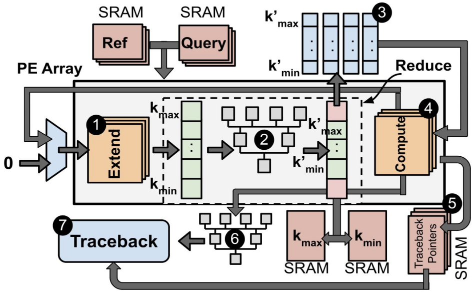


Key Contributions and Results

1. TALCO **guarantees optimality** under banding constraints
2. We applied TALCO to X-Drop (**TALCO- XDrop**) and WFA-Adapt (**TALCO-WFAA**)
3. TALCO-WFAA is the **first accelerator** based on the **WFA-Adapt algorithm** capable of performing arbitrary long sequence alignments
4. TALCO-XDrop and TALCO-WFAA (**software**) achieves up to **39x** and **57x** improvement in memory footprint, respectively, compared to software baselines
5. **Designed hardware accelerators** for TALCO-XDrop and TALCO-WFAA



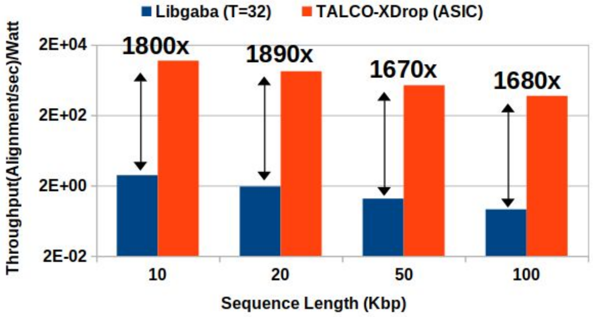
TALCO-XDrop hardware design

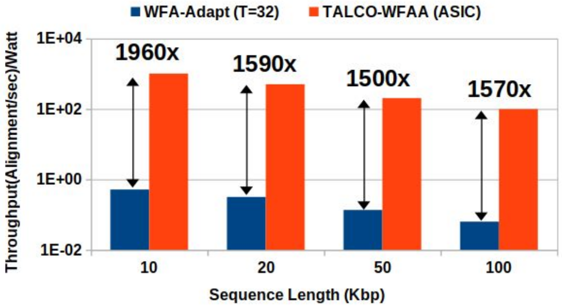


TALCO-WFAA hardware design

Key Contributions and Results

1. TALCO **guarantees optimality** under banding constraints
2. We applied TALCO to X-Drop (**TALCO- XDrop**) and WFA-Adapt (**TALCO-WFAA)**
3. TALCO-WFAA is the **first accelerator** based on the **WFA-Adapt algorithm** capable of performing arbitrary long sequence alignments
4. TALCO-XDrop and TALCO-WFAA (**software**) achieves up to **39x** and **57x** improvement in memory footprint, respectively, compared to software baselines
5. **Designed hardware accelerators** for TALCO-XDrop and TALCO-WFAA
6. TALCO-XDrop and TALCO-WFAA (**ASIC**) achieves up to **~1,900X** and **~2,000X**, respectively, improvement in **alignment throughput/watt** over software baselines





Key Contributions and Results

1. TALCO **guarantees optimality** under banding constraints
2. We applied TALCO to X-Drop (**TALCO- XDrop**) and WFA-Adapt (**TALCO-WFAA)**
3. TALCO-WFAA is the **first accelerator** based on the **WFA-Adapt algorithm** capable of performing arbitrary long sequence alignments
4. TALCO-XDrop and TALCO-WFAA (**software**) achieves up to **39x** and **57x** improvement in memory footprint, respectively, compared to software baselines
5. **Designed hardware accelerators** for TALCO-XDrop and TALCO-WFAA
6. TALCO-XDrop and TALCO-WFAA (**ASIC**) achieves up to **~1,900X** and **~2,000X**, respectively, improvement in **alignment throughput/watt** over software baselines
7. We **synthesized** TALCO-XDrop and TALCO-WFAA for **FPGAs** available on the **Amazon EC2 FPGA instances**

https://github.com/TurakhiaLab/TALCO/blob/main/hardware/README.md

Building on AWS EC2 F1 instance

Follow the below instructions to execute TALCO-XDrop and TALCO-WFAA on the AWS EC2 F1 instance, f1.2xlarge.

- Clone aws-fpga repository

```
git clone https://github.com/aws/aws-fpga
cd aws-fpga
source vitis_runtime_setup.sh
```

- Clone TALCO repository

```
git clone https://github.com/TurakhiaLab/TALCO.git
export TALCO_DIR=$PWD/TALCO
cd TALCO/hardware/TALCO-XDrop
```

- Steps for running on the EC2 F1 instance, f1.2xlarge (MODE-hw)

```
source $TALCO_DIR/hardware/scripts/run.sh
$TALCO_DIR/dataset/sequence.fa TALCO_XDrop.awsxclbin
```

Key Contributions and Results

1. TALCO **guarantees optimality** under banding constraints
2. We applied TALCO to X-Drop (**TALCO- XDrop**) and WFA-Adapt (**TALCO-WFAA)**
3. TALCO-WFAA is the **first accelerator** based on the **WFA-Adapt algorithm** capable of performing arbitrary long sequence alignments
4. TALCO-XDrop and TALCO-WFAA (**software**) achieves up to **39x** and **57x** improvement in memory footprint, respectively, compared to software baselines
5. **Designed hardware accelerators** for TALCO-XDrop and TALCO-WFAA
6. TALCO-XDrop and TALCO-WFAA (**ASIC**) achieves up to **~1,900X** and **~2,000X**, respectively, improvement in **alignment throughput/watt** over software baselines
7. We **synthesized** TALCO-XDrop and TALCO-WFAA for **FPGAs** available on the **Amazon EC2 FPGA instances**

https://github.com/TurakhiaLab/TALCO/

Available **Reviewed** **Reproduced**

HPCA Artifact Evaluation

UC San Diego
Electrical and Computer Engineering
JACOBS SCHOOL OF ENGINEERING

TURAKHIA LAB

TALCO: Tiling Genome Sequence Alignment using Convergence of Traceback Pointers

Sumit Walia
Ph.D student

Cheng Ye
MS student

Arkid Bera
MS student

Dhruvi L.
MS student

Yatish Turakhia
Assistant Professor, UCSD

Thank you!