Algorithmic Improvement and GPU Acceleration of the GenASM Algorithm

Joël Lindegger

Damla Senol Cali, Mohammed Alser, Juan Gómez-Luna, Onur Mutlu

jmlindegger@gmail.com

SAFARI Research Group

ETH zürich

30/05/2022

HiCOMB 2022

Background

Pairwise sequence alignment is a common stage in **genomic pipelines**, such as **read mapping**

Pairwise sequence alignment is computationally **expensive** and often the pipeline's **bottleneck**

GenASM is a **state-of-the-art sequence aligner** with **high throughput** and **energy efficiency**

[1] Senol Cali+, "GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework for Genome Sequence Analysis", MICRO 2020

Benefits of GenASM

GenASM uses only cheap **bitwise operations**

GenASM has **linear runtime complexity** with respect to the **sequence length**, due to its **windowing heuristic**

GenASM **supports traceback**, thus it can report the **CIGAR string**

GenASM provides **intra-task parallelism**

[1] Senol Cali+, "GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework for Genome Sequence Analysis", MICRO 2020

Problem – Inefficiencies in GenASM

GenASM has a **large** **memory footprint**

GenASM has a **low** **operational intensity**

GenASM does **unnecessary** **work**

Our Goal

Our goal is to enable **fast** and **efficient** implementations of GenASM for **CPUs**, **GPUs**, and **ASICs**

To this end we propose **Scrooge**, which

- **Reduces** the **memory footprint**
- **Reduces** the **number of bytes accessed**
- **Eliminates** **unnecessary work**

Scrooge

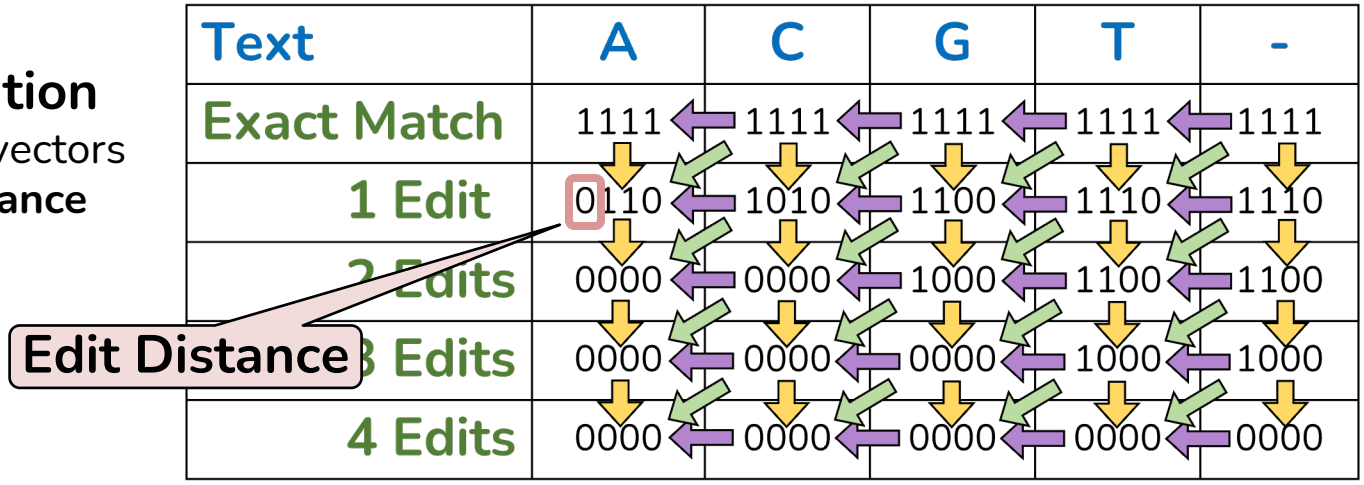
Three novel algorithmic improvements to the GenASM algorithm

High-performance CPU and **GPU** implementations of the improved algorithm

GenASM Algorithm

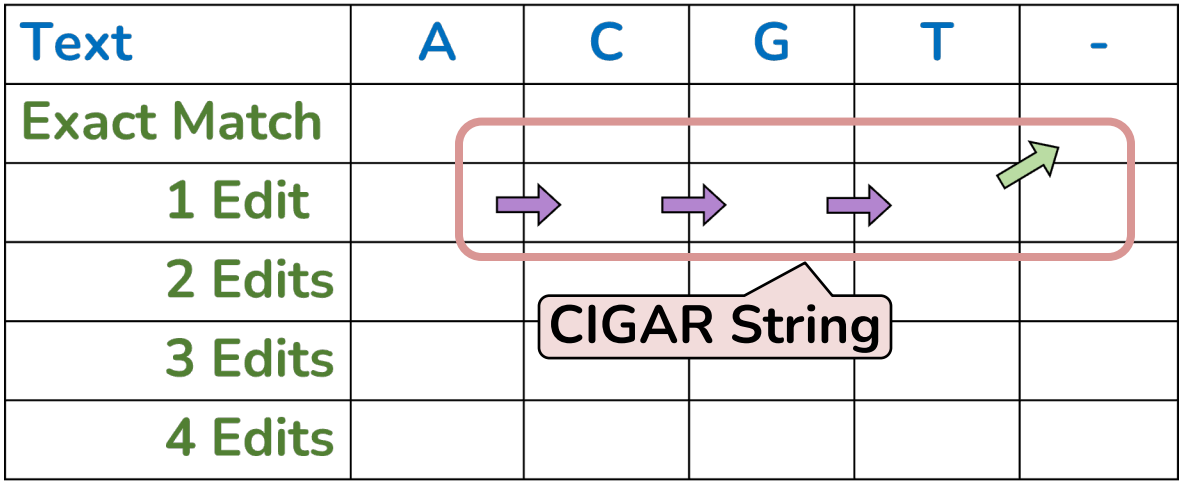
1 Distance Calculation

- Builds table of bitvectors
- Finds the **edit distance**



2 Traceback

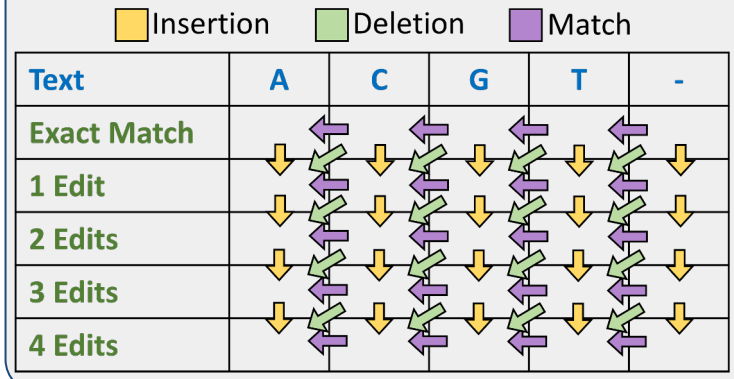
- Requires table of bitvectors
- Finds the **CIGAR string**



Scrooge Improvement 1:
Store Entries, Not Edges (SENE)

GenASM

Stores **3 edges** per table entry **for traceback**



SENE Improvement

Scrooge

Stores table **entries** directly, edges are **regenerated**

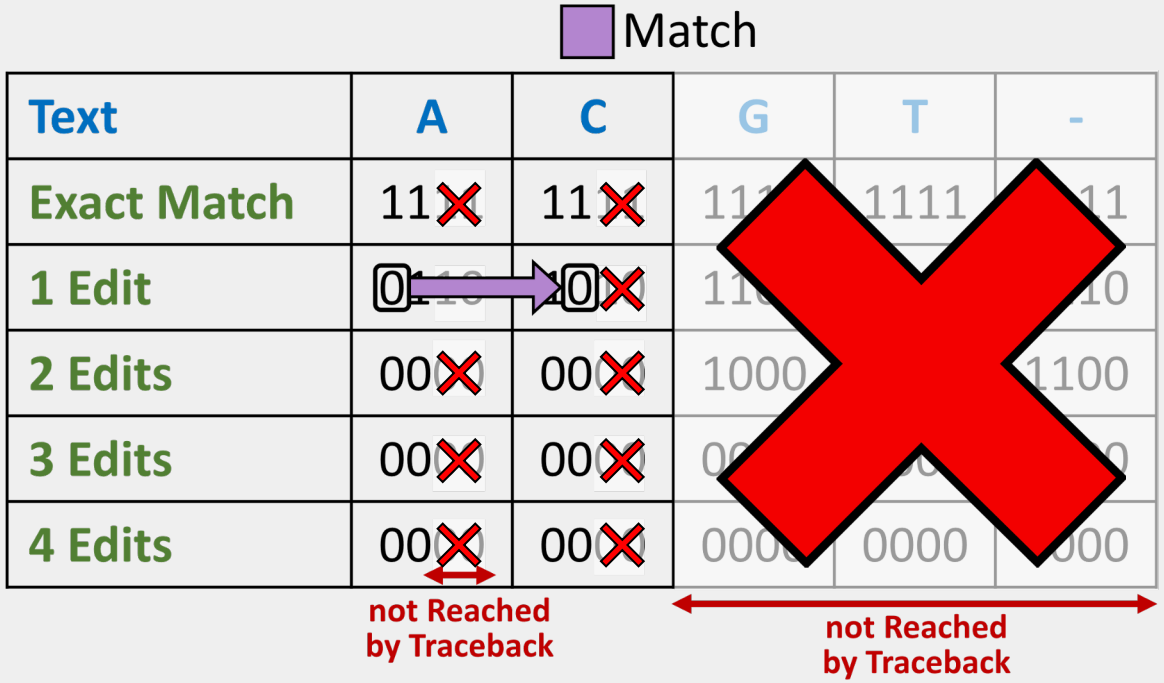
Table Entry 1111

Text	A	C	G	T	-
Exact Match	1111	1111	1111	1111	1111
1 Edit	0110	1010	1100	1110	1110
2 Edits	0000	0000	1000	1100	1100
3 Edits	0000	0000	0000	1000	1000
4 Edits	0000	0000	0000	0000	0000

SENE results in a **3x reduction** in **memory footprint** and **data movement**

Scrooge Improvement 2:
Discard Entries Not used by Traceback (DENT)

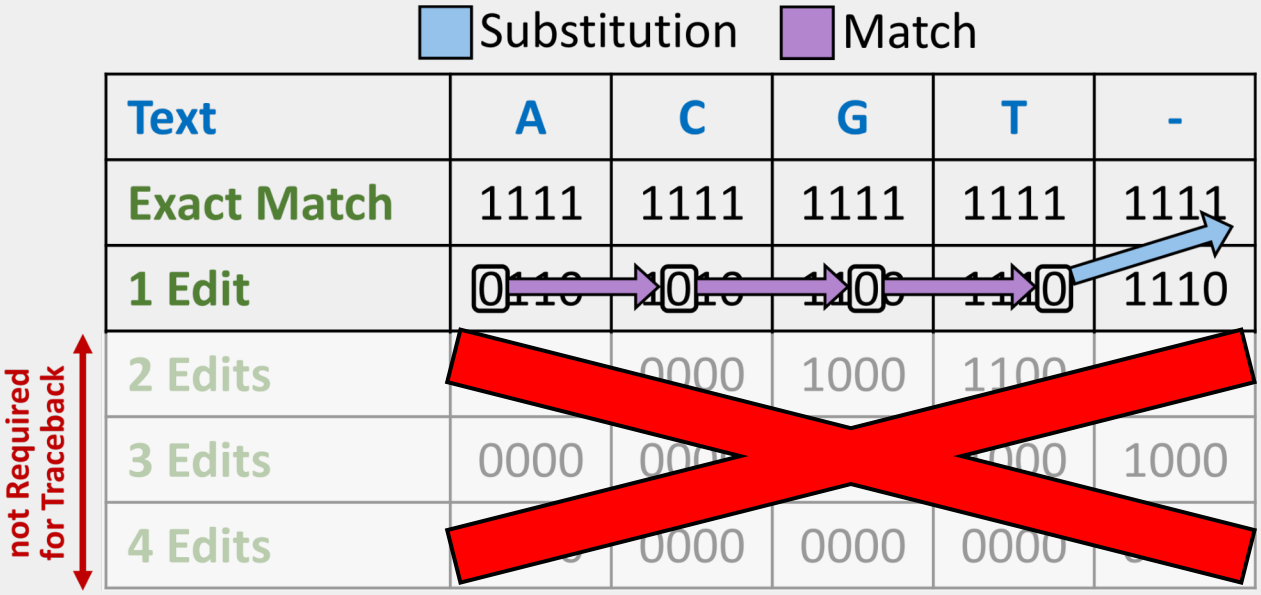
GenASM's **traceback** does **not cross** the **entire table**
Scrooge discards entries that are **never reached**



DENT results in a **4x reduction** in
memory footprint and **data movement**

Scrooge Improvement 3:
Early Termination (ET)

Rows **below** the **edit distance** do **not** contain **useful information**
Scrooge **stops the computation** when it finds the **edit distance**



ET **eliminates** **unnecessary work**
and **reduces** **data movement**

Methodology

Platforms:

- **CPU:** Dual Socket **Intel Xeon Gold 5118** (2× 24 logical cores at 3.2GHz with 196GB DDR4)
- **GPU: NVIDIA A6000**

Baseline Tools:

- **CPU:** KSW2 [2,3], Edlib [4]
- **GPU:** Darwin-GPU [5]

Dataset:

- We simulate **500 PacBio reads** from the human genome using PBSIM2 [6], each of length 10kb
- We obtain **138,929 read/candidate location pairs** from minimap2 [2]

Key Results

Scrooge is **faster** than **all baselines**

1.9x speedup over **GenASM** on a recent **CPU**

5.9x speedup over **GenASM** on a recent **GPU**

Discussion - ASIC

A **prior ASIC** implementation of **GenASM** [1] dedicates the **majority** of **chip area** and **power** to **SRAM**

[1] Senol Cali+, "GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework for Genome Sequence Analysis", MICRO 2020

Our **algorithmic improvements** can **reduce** the **SRAM size** by **12x**

Our **algorithmic improvements** could **reduce** GenASM's total ASIC **chip area** and **power** by up to **2.5x** and **2x**

Conclusion

Motivation

- **Pairwise sequence alignment** is a **bottleneck** in genomic pipelines (e.g., read mapping)
- **GenASM** is a **state-of-the-art** sequence aligner, its hardware implementation is up to **10,000x faster** than prior software aligners and draws **500x less power**

Problem

Three inefficiencies in the GenASM algorithm limit its throughput and energy efficiency:

1. It has a **large memory footprint**
2. It has a **high bandwidth pressure**
3. It does some **unnecessary work**

Goal

Enable **fast** and **efficient** implementations of GenASM for CPUs, GPUs, and ASICs

Scrooge

Three novel algorithmic improvements to the **GenASM algorithm**:

- ***SENE*** and ***DENT*** reduce the **memory footprint** and **data movement** of GenASM
- ***Early Termination*** eliminates **unnecessary work**

High-performance **CPU** and **GPU implementations**

Results

- **1.9x speedup** over GenASM on a recent CPU (Xeon Gold 5118)
- **5.9x speedup** over GenASM on a recent GPU (NVIDIA A6000)
- Similar improvements to be expected for ASICs

Links

Source Code
(Available Soon!)

Short Paper

Algorithmic Improvement and GPU Acceleration of the GenASM Algorithm

Joël Lindegger

Damla Senol Cali, Mohammed Alser, Juan Gómez-Luna, Onur Mutlu

jmlindegger@gmail.com

SAFARI Research Group

ETH zürich

30/05/2022

HiCOMB 2022

References

[1] Senol Cali+, “GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework for Genome Sequence Analysis”, MICRO 2020

[2] Li, “Minimap2: Pairwise Alignment for Nucleotide Sequences”, Bioinformatics 2018

[3] Suzuki+, “Introducing Difference Recurrence Relations for Faster Semi-Global Alignment of Long Sequences”, BMC Bioinformatics 2018

[4] Šošić+, “Edlib: A C/C++ Library for Fast, Exact Sequence Alignment Using Edit Distance”, Bioinformatics 2017

[5] Ahmed+, “GPU Acceleration of Darwin Read Overlapper for de Novo Assembly of Long DNA Reads”, BMC Bioinformatics 2020

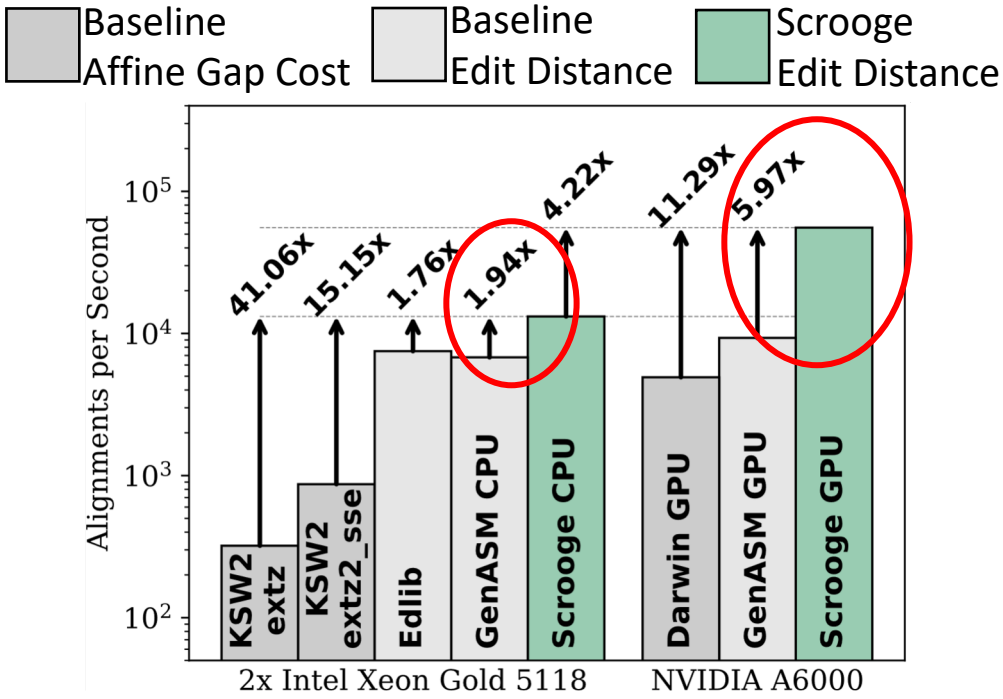
[6] Ono+, “PBSIM2: A Simulator for Long-Read Sequencers With a Novel Generative Model of Quality Scores”, Bioinformatics 2020

Backup Slides

Why “Scrooge”?

- **Our algorithm** has an extremely **small memory footprint**
- In other words, it is extremely **resource-saving**, a common attribute of characters called “Scrooge” in pop culture

Speedup



Scrooge with the **algorithmic improvements** is **significantly faster** than GenASM on **CPUs** and **GPUs**.

Result - Parameter Sweeps are Necessary

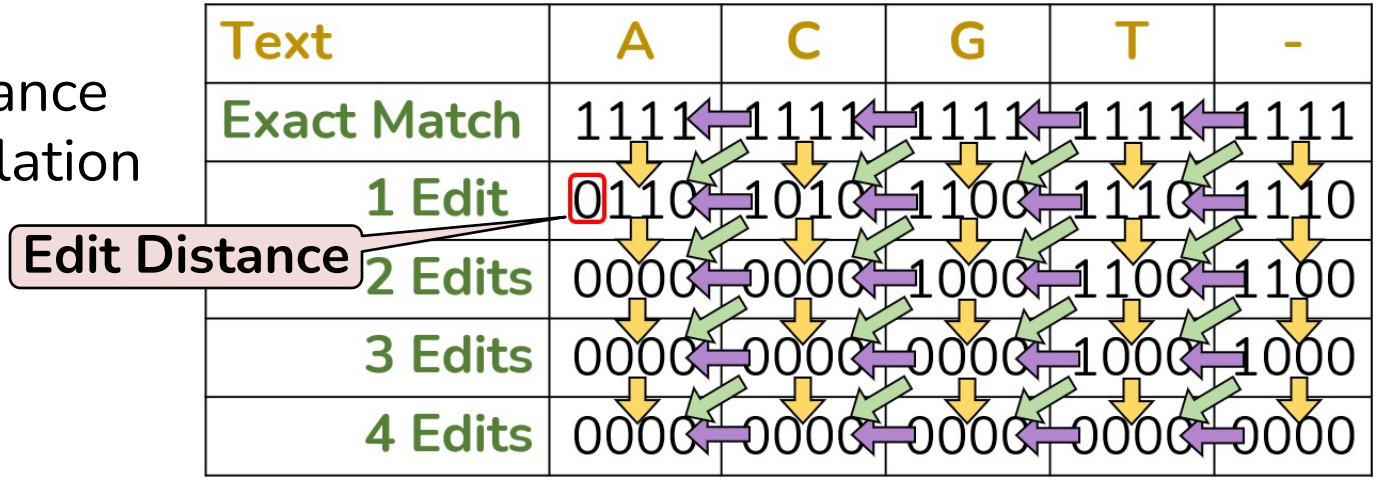
The **CPU** version of **Scrooge** **benefits most** from **ET** and **SENE**, while **DENT** should **not be used**.

The **GPU** version of **Scrooge** **benefits most** from **SENE** and **DENT**, while **ET** only yields **marginal speedups**.

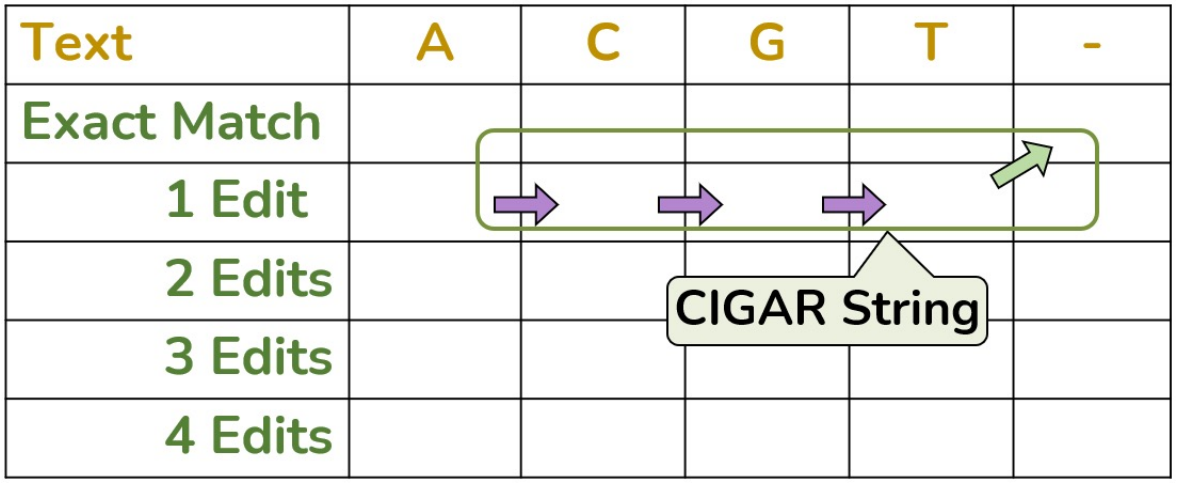
The **best combination** of **improvements** **depends** on the **computing platform**.

GenASM Algorithm

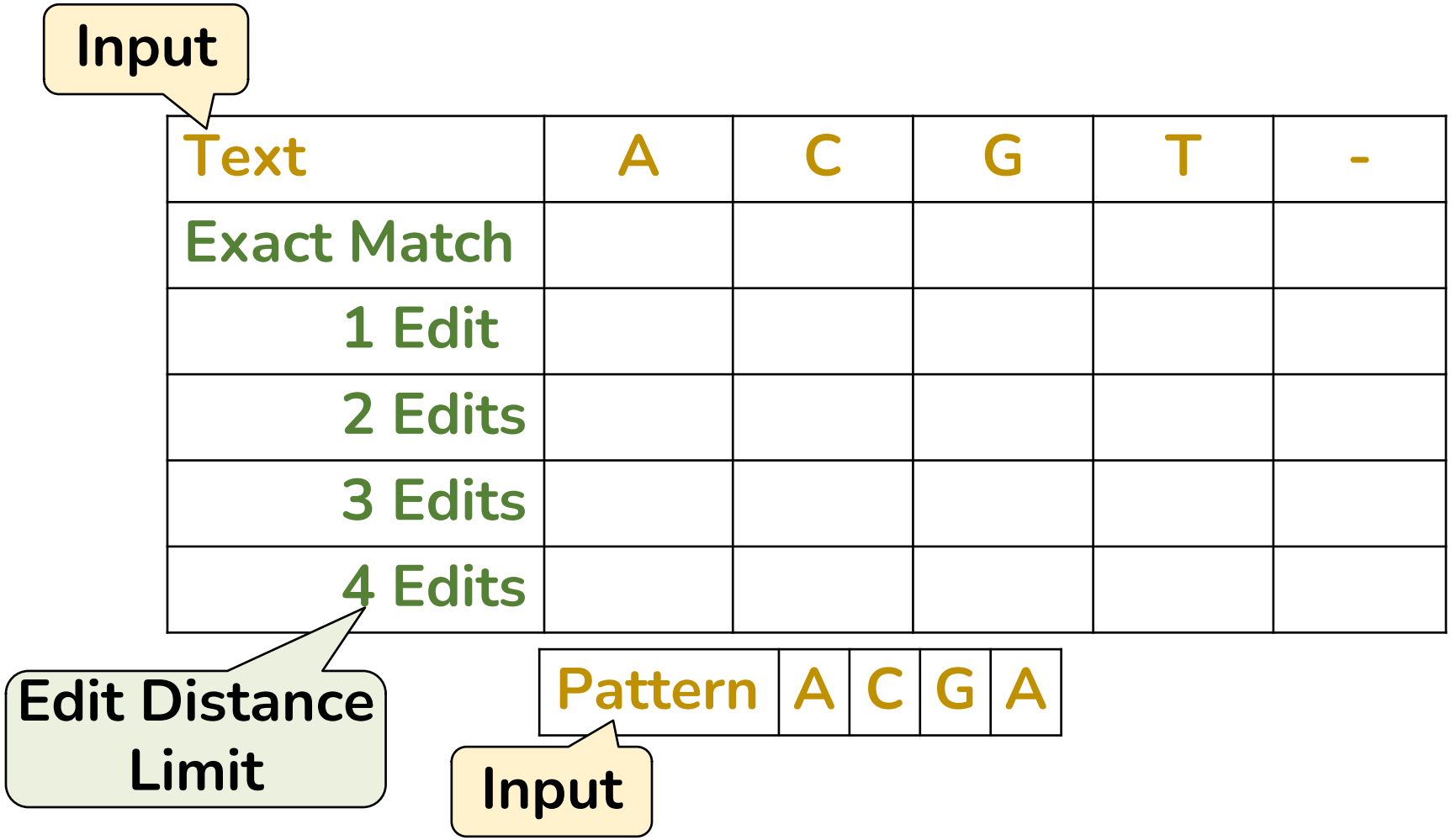
1. Distance Calculation



2. Traceback

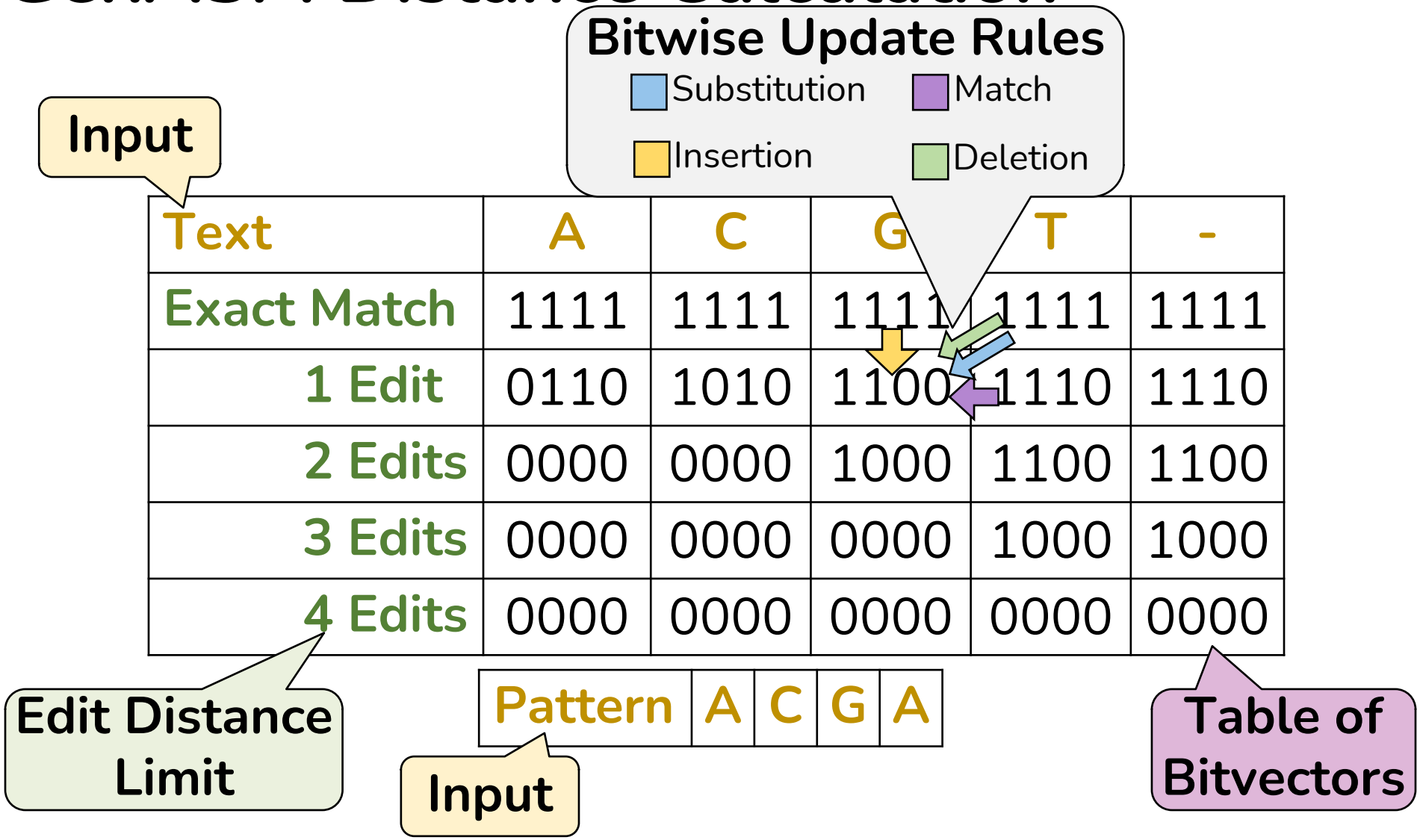


GenASM Distance Calculation



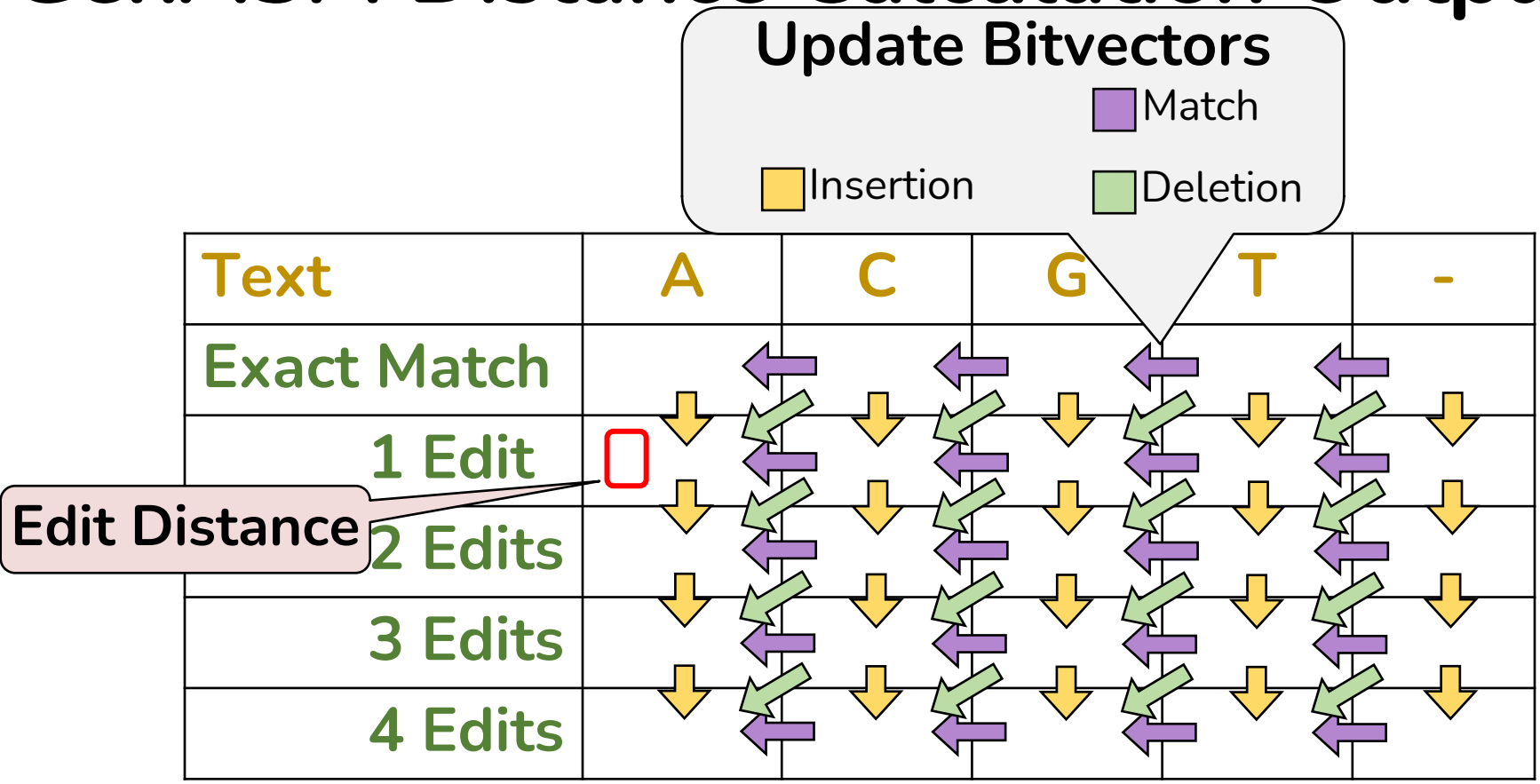
[1] Senol Cali+, "GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework for Genome Sequence Analysis", MICRO 2020

GenASM Distance Calculation



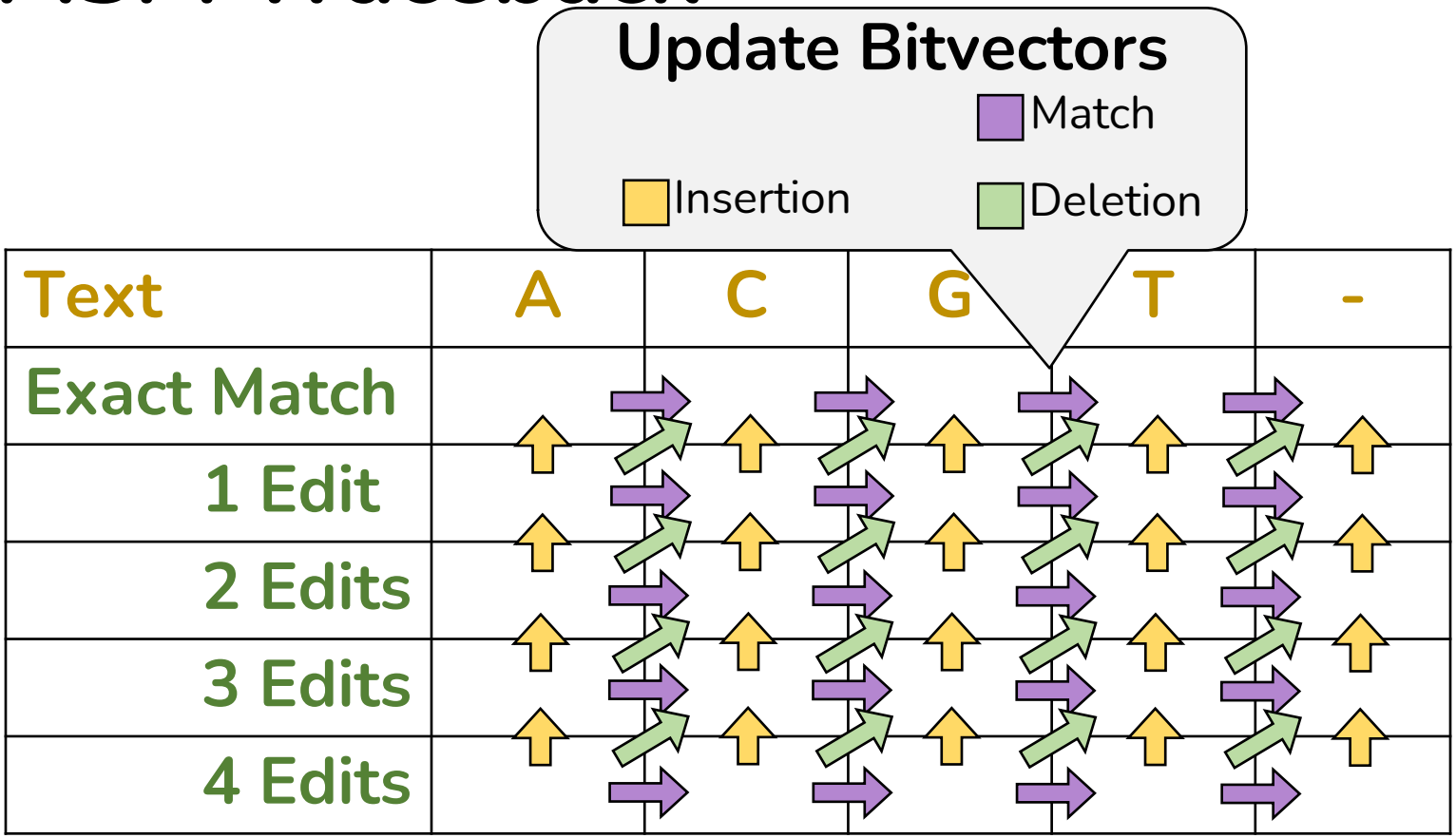
[1] Senol Cali+, "GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework for Genome Sequence Analysis", MICRO 2020

GenASM Distance Calculation Output



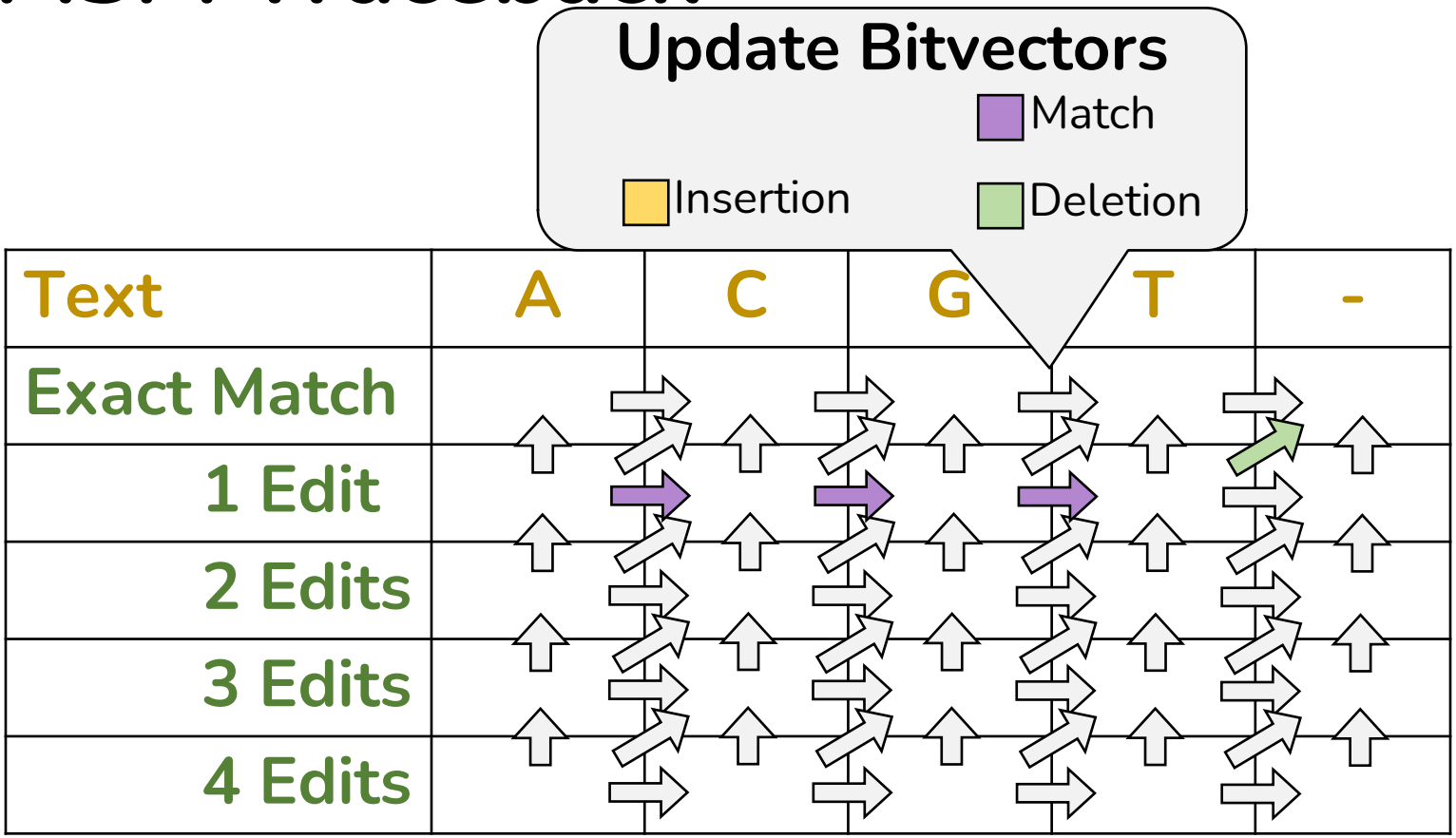
[1] Senol Cali+, "GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework for Genome Sequence Analysis", MICRO 2020

GenASM Traceback



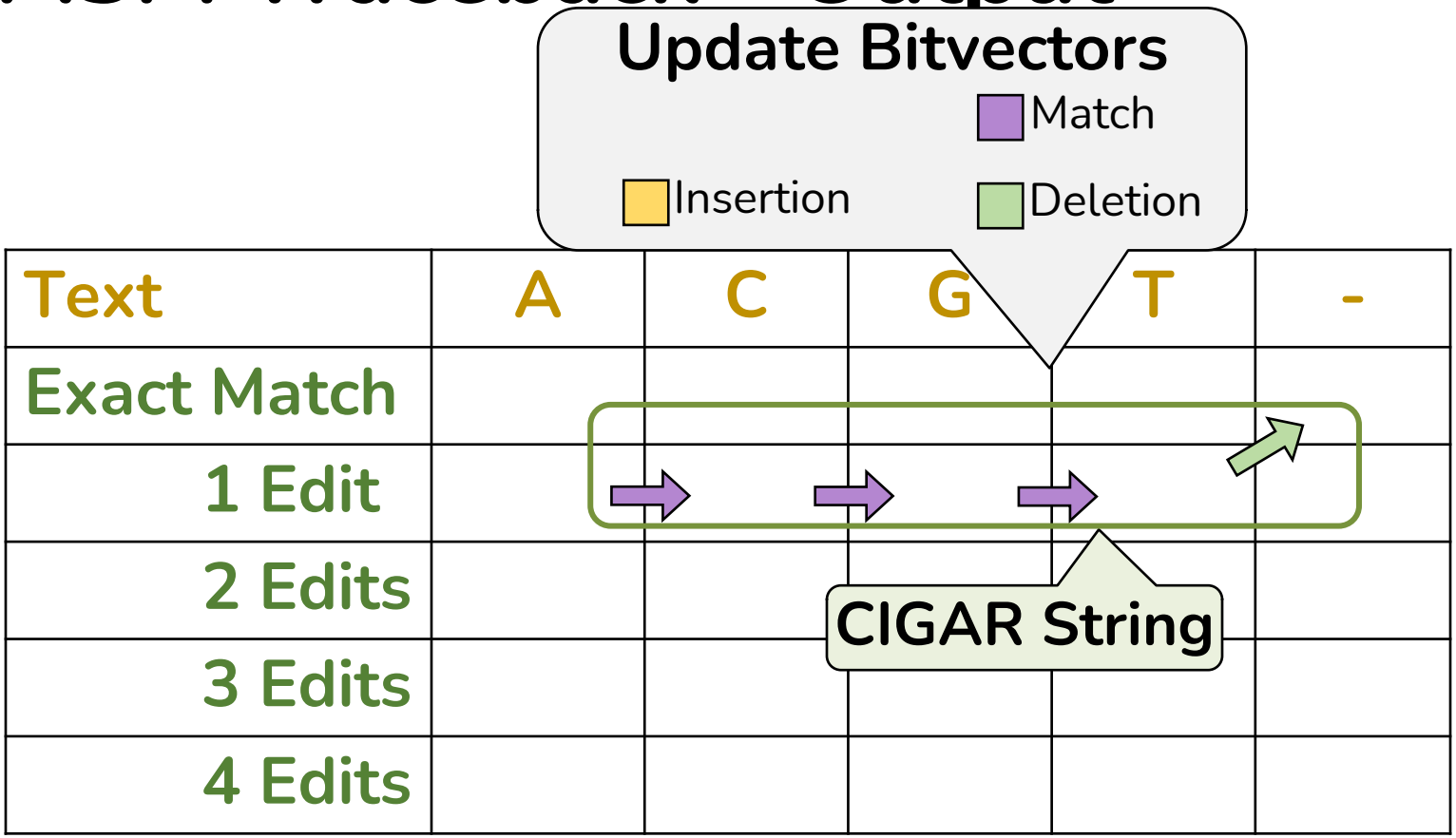
[1] Senol Cali+, "GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework for Genome Sequence Analysis", MICRO 2020

GenASM Traceback



[1] Senol Cali+, "GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework for Genome Sequence Analysis", MICRO 2020

GenASM Traceback - Output



[1] Senol Cali+, "GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework for Genome Sequence Analysis", MICRO 2020