ETH zürich

Lossless Indexing with Counting de Bruijn Graphs

Mikhail Karasikov, Harun Mustafa, Gunnar Rätsch, André Kahles




RECOMB 2022

24 May 2022

ETH zürich

Lossless Indexing with Counting de Bruijn Graphs

Mikhail Karasikov, Harun Mustafa, Gunnar Rätsch, André Kahles




RECOMB 2022

24 May 2022

ETH zürich

Lossless Indexing with Counting de Bruijn Graphs

Mikhail Karasikov, Harun Mustafa, Gunnar Rätsch, André Kahles




RECOMB 2022

24 May 2022

ETH zürich

Lossless Indexing with Counting de Bruijn Graphs

Mikhail Karasikov, Harun Mustafa, Gunnar Rätsch, André Kahles




RECOMB 2022

24 May 2022

Background

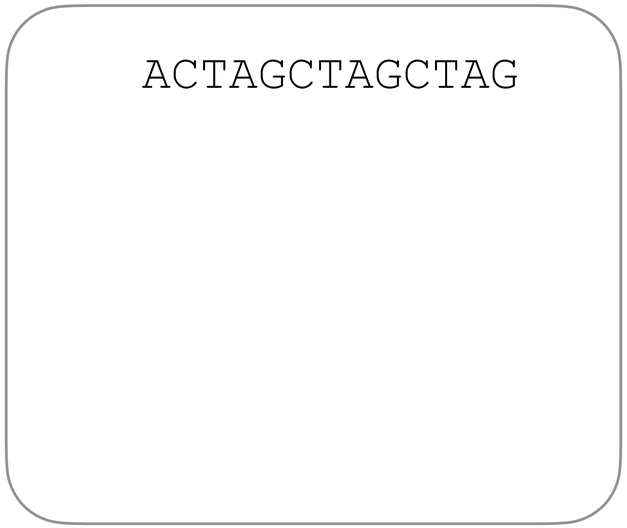
De Bruijn Graphs

… widely used in Bioinformatics since 1989

Background

De Bruijn Graphs

… widely used in Bioinformatics since 1989

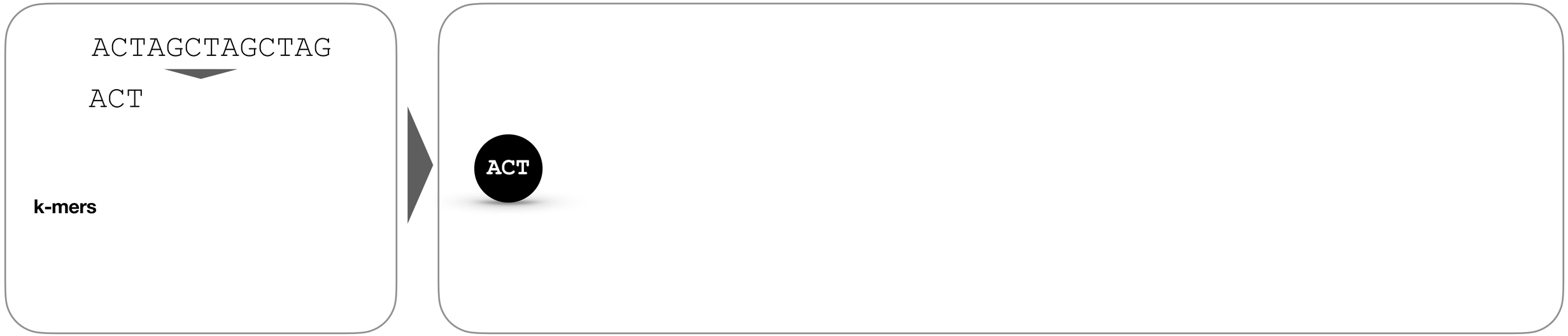


- Originally employed for ***de novo* assembly**

Background

De Bruijn Graphs

… widely used in Bioinformatics since 1989

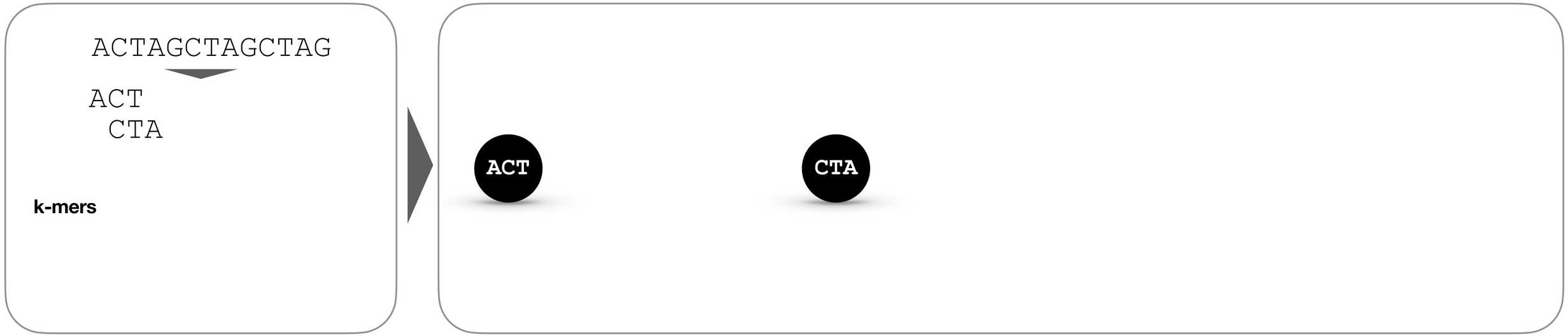


- Originally employed for ***de novo* assembly**

Background

De Bruijn Graphs

… widely used in Bioinformatics since 1989

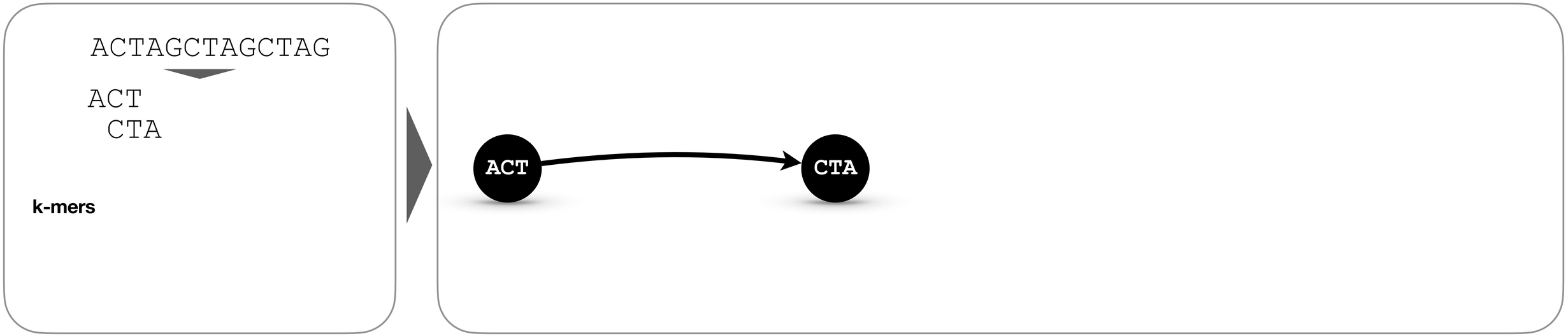


- Originally employed for ***de novo* assembly**

Background

De Bruijn Graphs

… widely used in Bioinformatics since 1989

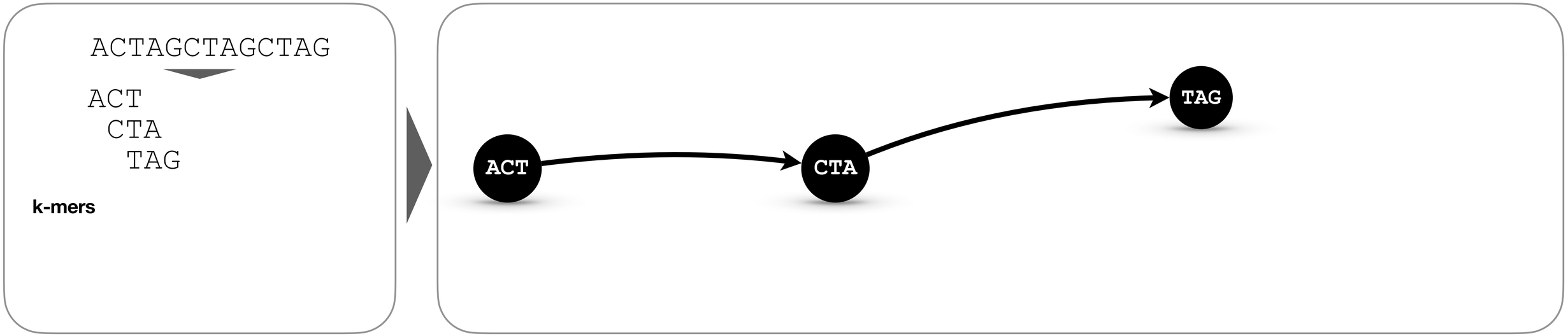


- Originally employed for ***de novo* assembly**

Background

De Bruijn Graphs

… widely used in Bioinformatics since 1989

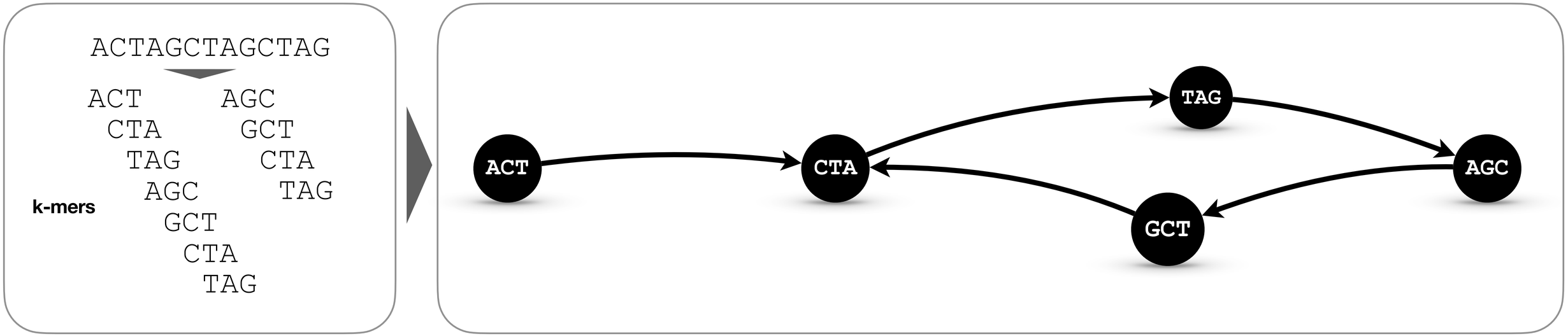


- Originally employed for ***de novo* assembly**

Background

De Bruijn Graphs

… widely used in Bioinformatics since 1989

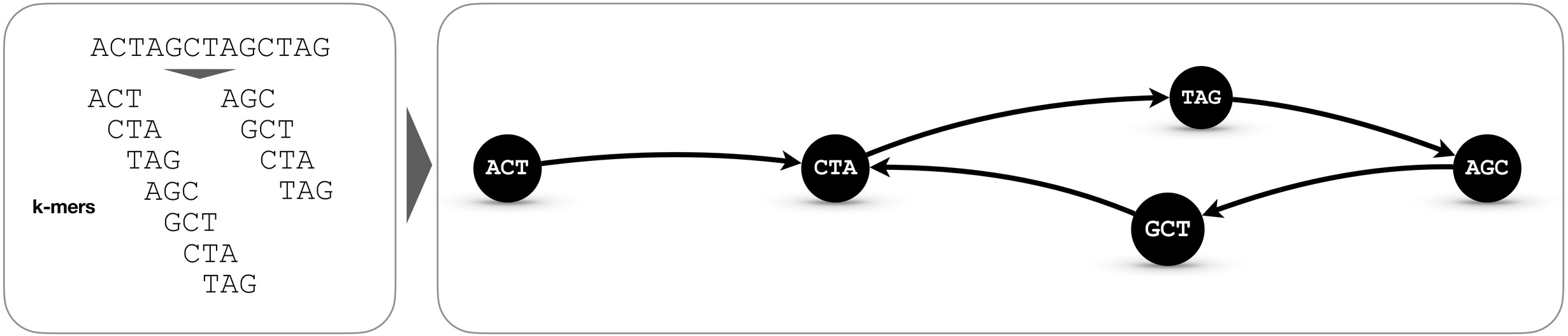


- Originally employed for ***de novo* assembly**

Background

De Bruijn Graphs

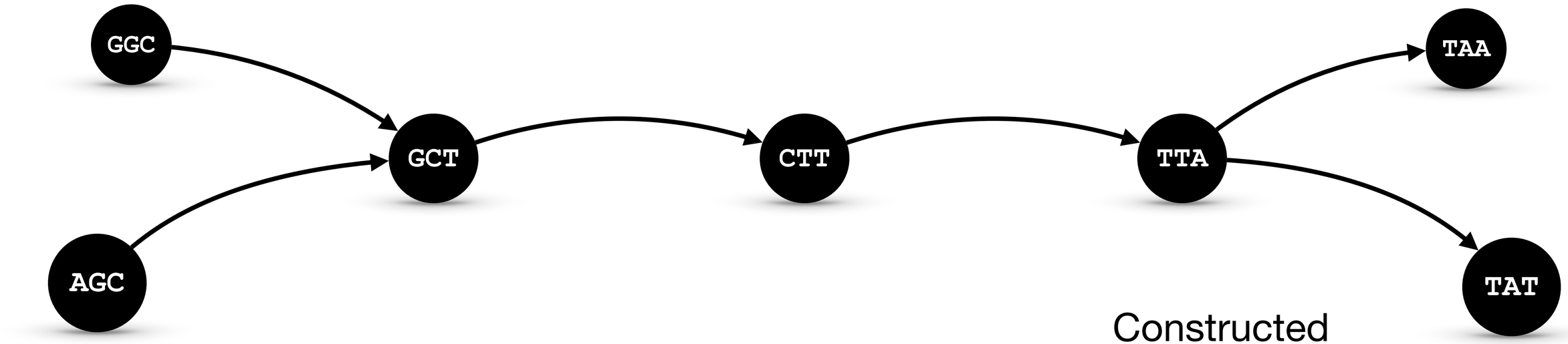
… widely used in Bioinformatics since 1989



- Originally employed for ***de novo* assembly**
- Now, also widely used for **indexing raw sequencing data**

Background

Annotated de Bruijn Graphs



Constructed
from sequences

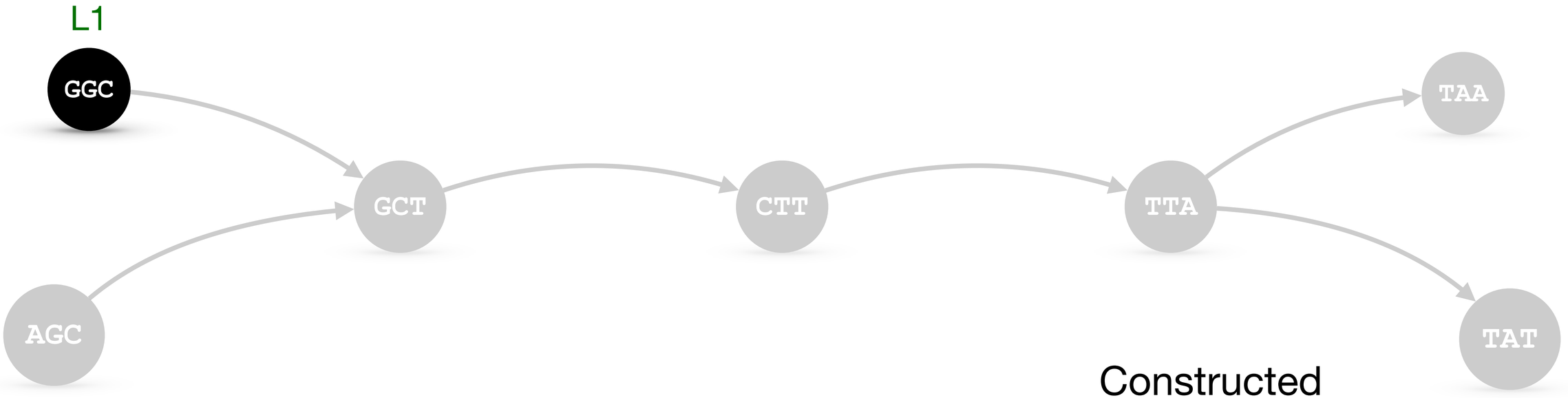
L1: GGCTTAT

L2: AGCTTAA

L3: TTAA

Background

Annotated de Bruijn Graphs



Constructed
from sequences

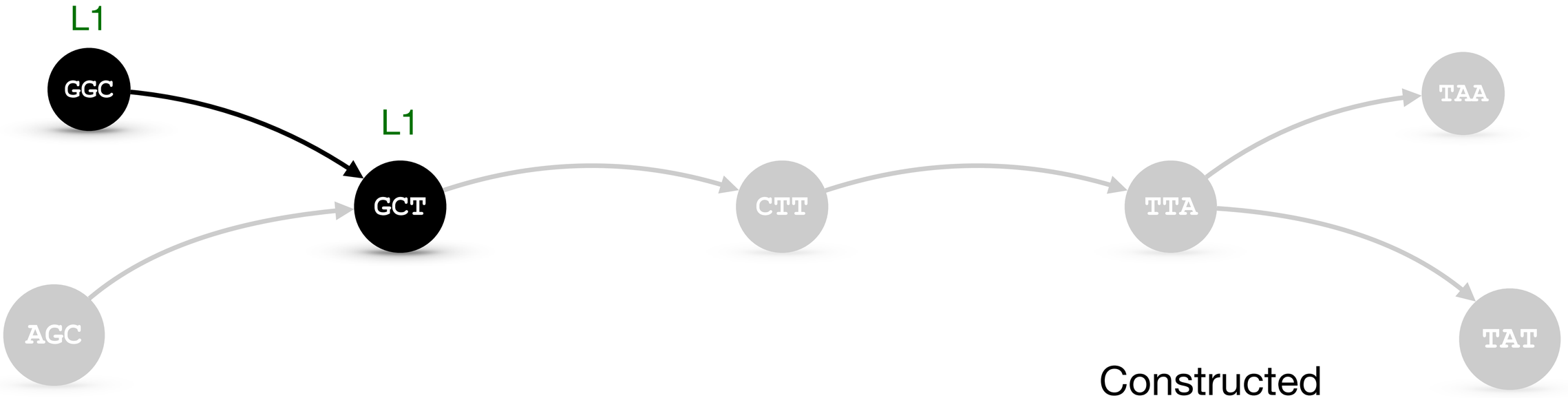
L1: GGCTTAT

L2: AGCTTAA

L3: TTAA

Background

Annotated de Bruijn Graphs



Constructed
from sequences

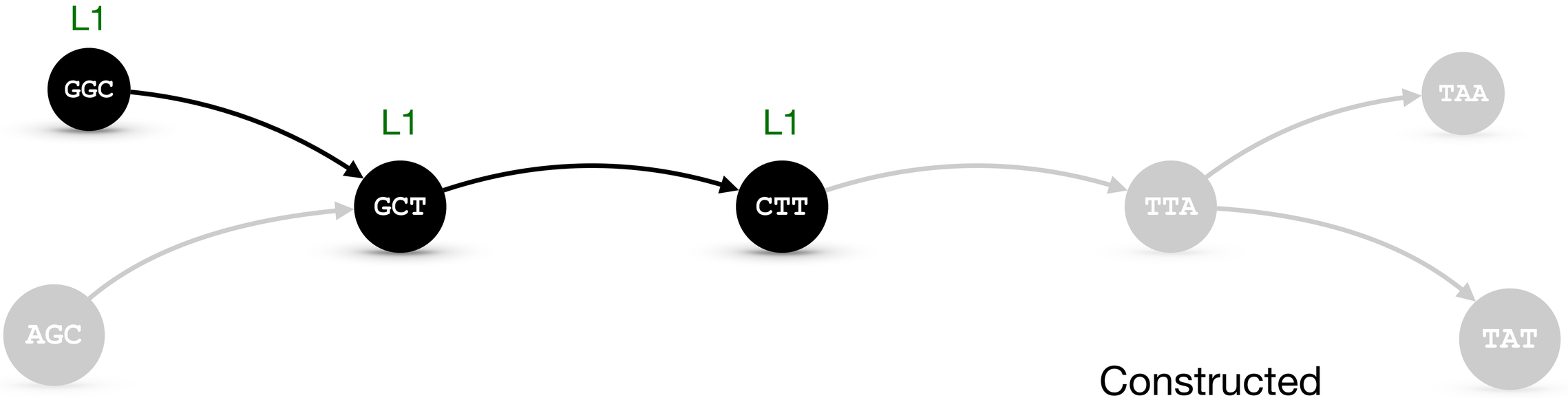
L1: GGCTTAT

L2: AGCTTAA

L3: TTAA

Background

Annotated de Bruijn Graphs



Constructed
from sequences

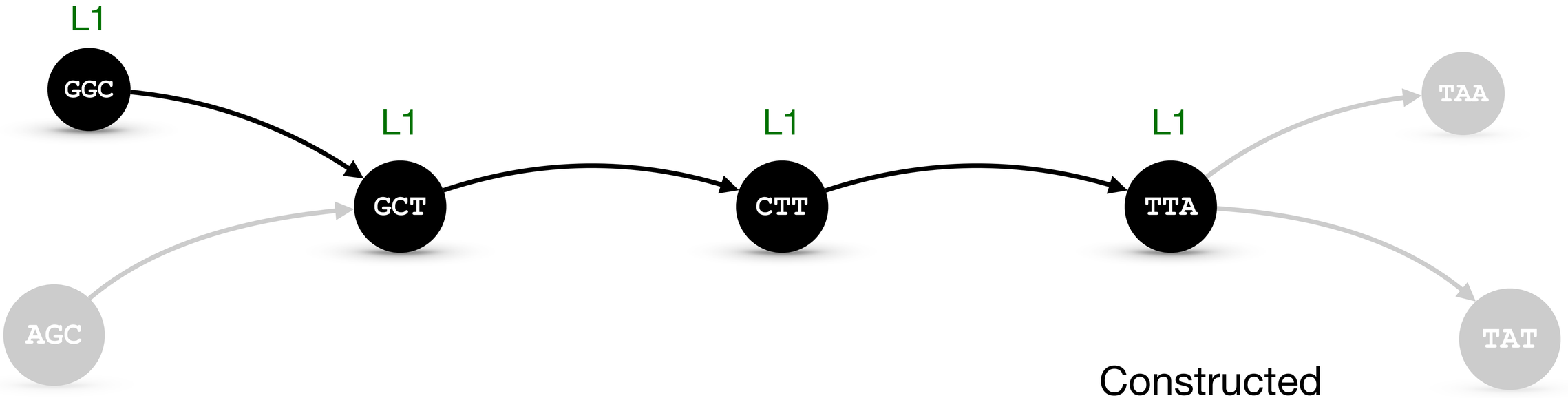
L1: GGCTTAT

L2: AGCTTAA

L3: TTAA

Background

Annotated de Bruijn Graphs



Constructed
from sequences

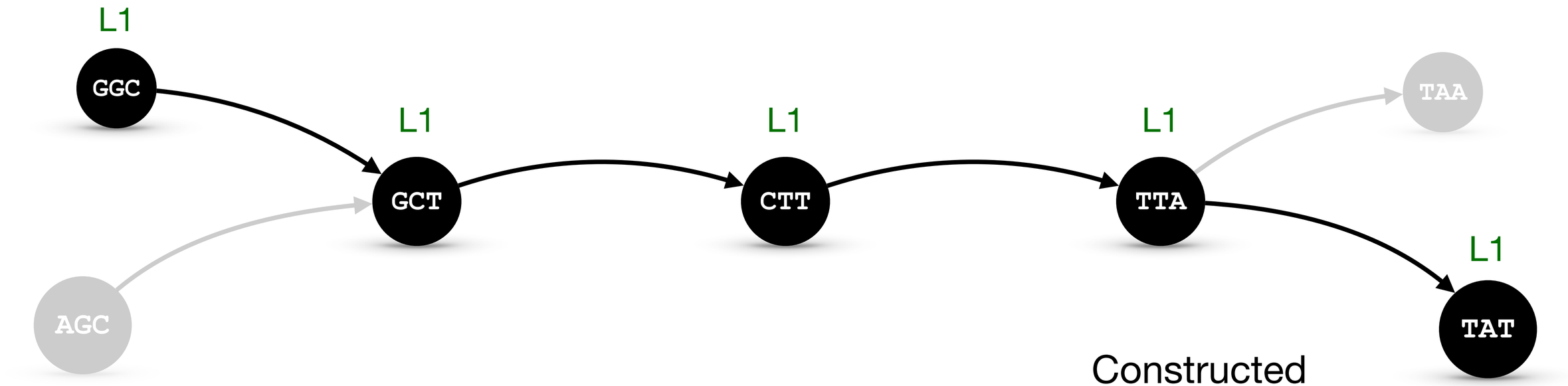
L1: GGCTTATT

L2: AGCTTAA

L3: TTAA

Background

Annotated de Bruijn Graphs



Constructed
from sequences

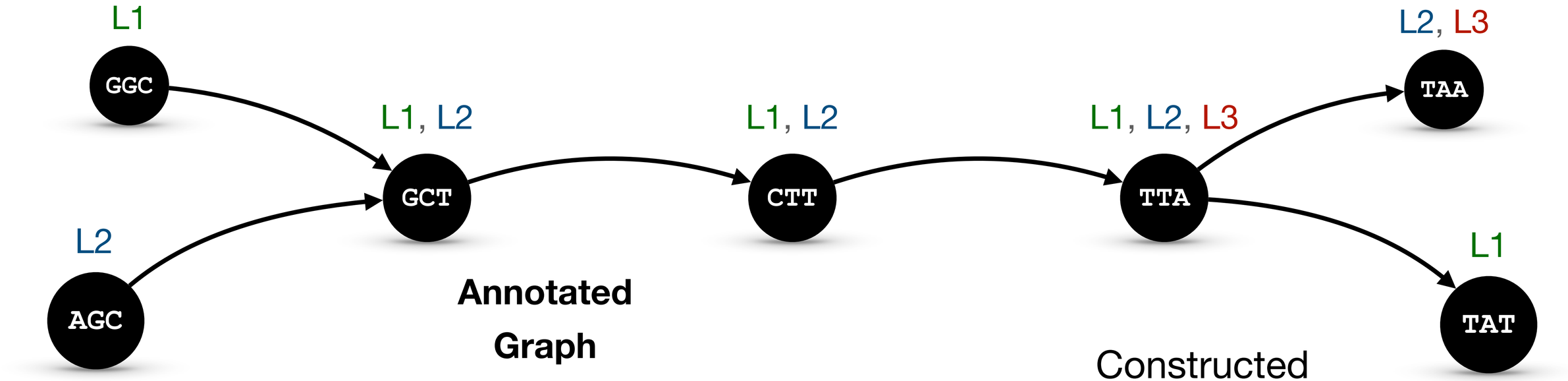
L1: GGCTTAT

L2: AGCTTAA

L3: TTAA

Background

Annotated de Bruijn Graphs



Annotated Graph

Constructed from sequences

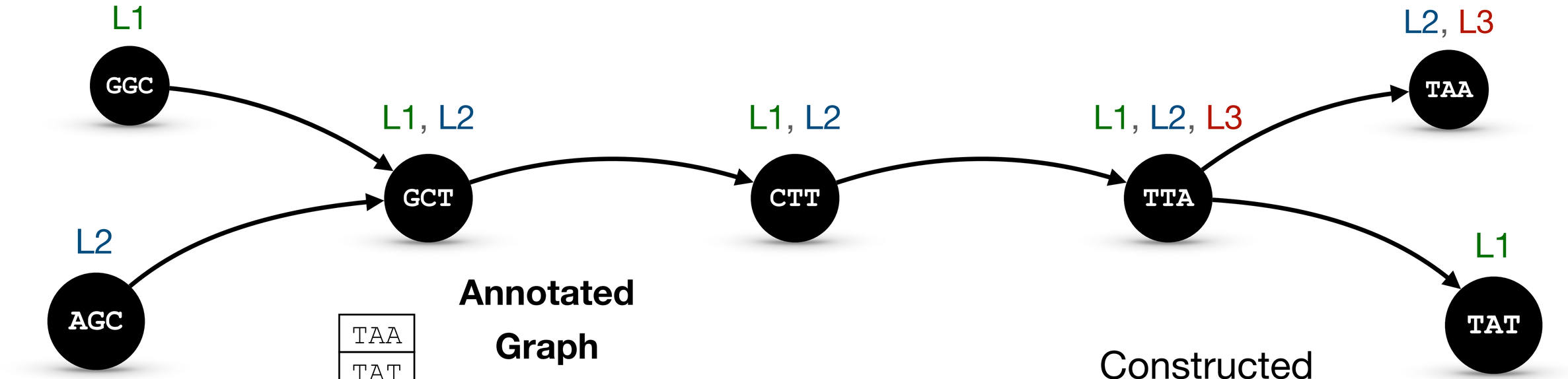
L1: GGCTTAT

L2: AGCTTAA

L3: TTAA

Background

Annotated de Bruijn Graphs



Annotated Graph

TAA
TAT
GCT
AGC
GGC
CTT
TTA

k-mer dictionary
(de Bruijn Graph)

Constructed from sequences

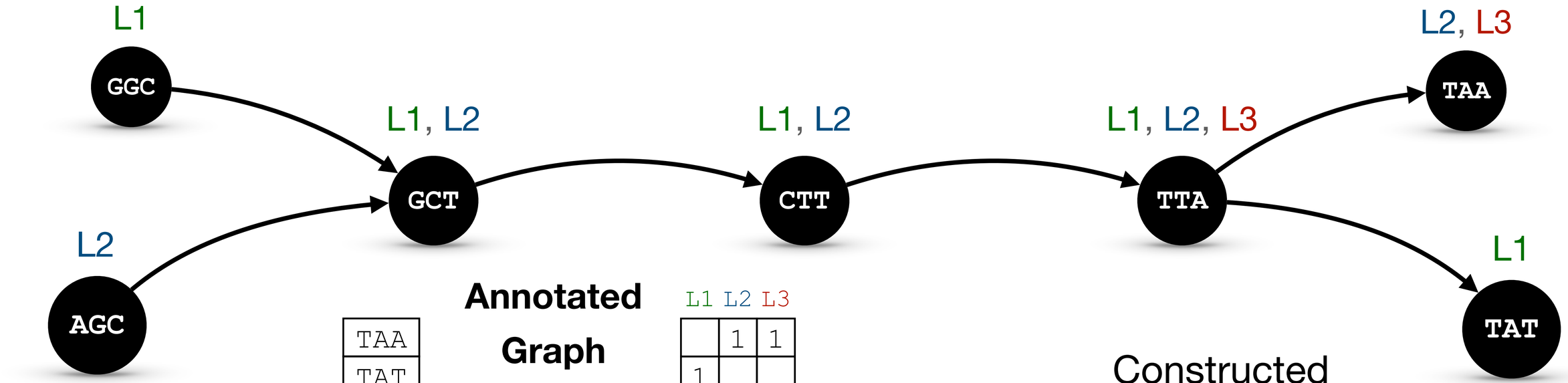
L1: GGCTTAT

L2: AGCTTAA

L3: TTAA

Background

Annotated de Bruijn Graphs



Annotated Graph

TAA
TAT
GCT
AGC
GGC
CTT
TTA

k-mer dictionary
(de Bruijn Graph)

L1	L2	L3
	1	1
1		
1	1	
	1	
1		
1	1	
1	1	1

Graph annotation
(node labeling)

Constructed from sequences

L1: GGCTTAT

L2: AGCTTAA

L3: TTAA

Background

There exist different techniques for representing Annotated DBG

- VARI **[Muggli et al., 2017]**
- Rainbowfish **[Almodaresi et al., 2017]**
- Mantis-MST **[Almodaresi et al., 2019]**
- Multi-BRWT **[Karasikov et al., 2019]**
- Bloom filter-based **[Bradley et al., 2019]**
- RowDiff **[Danciu et al., 2021]**
- MetaGraph **[Karasikov et al., 2020]**
- … and more

Background

There exist different techniques for representing Annotated DBG

- VARI **[Muggli et al., 2017]**
- Rainbowfish **[Almodaresi et al., 2017]**
- Mantis-MST **[Almodaresi et al., 2019]**
- Multi-BRWT **[Karasikov et al., 2019]**
- Bloom filter-based **[Bradley et al., 2019]**
- RowDiff **[Danciu et al., 2021]**
- MetaGraph **[Karasikov et al., 2020]**
- … and more

Our goals will be to **efficiently** represent:

1. **k-mer abundances**

Background

There exist different techniques for representing Annotated DBG

- VARI **[Muggli et al., 2017]**
- Rainbowfish **[Almodaresi et al., 2017]**
- Mantis-MST **[Almodaresi et al., 2019]**
- Multi-BRWT **[Karasikov et al., 2019]**
- Bloom filter-based **[Bradley et al., 2019]**
- RowDiff **[Danciu et al., 2021]**
- MetaGraph **[Karasikov et al., 2020]**
- … and more

Our goals will be to **efficiently** represent:

1. **k-mer abundances**
2. **k-mer coordinates** (to encode sequences **losslessly)**

Background

There exist different techniques for representing Annotated DBG

- VARI **[Muggli et al., 2017]**
- Rainbowfish **[Almodaresi et al., 2017]**
- Mantis-MST **[Almodaresi et al., 2019]**
- Multi-BRWT **[Karasikov et al., 2019]**
- Bloom filter-based **[Bradley et al., 2019]**
- RowDiff **[Danciu et al., 2021]**
- MetaGraph **[Karasikov et al., 2020]**
- … and more

Our goals will be to **efficiently** represent:

1. **k-mer abundances**

sslessly)

Bioinformatics, 36, 2020, i177–i185
doi: 10.1093/bioinformatics/btaa487
ISMB 2020

OXFORD

REINDEER: efficient indexing of *k*-mer presence and abundance in sequencing datasets

Camille Marchet[1,*], Zamin Iqbal[2], Daniel Gautheret[3], Mikaël Salson[1] and Rayan Chikhi[4]

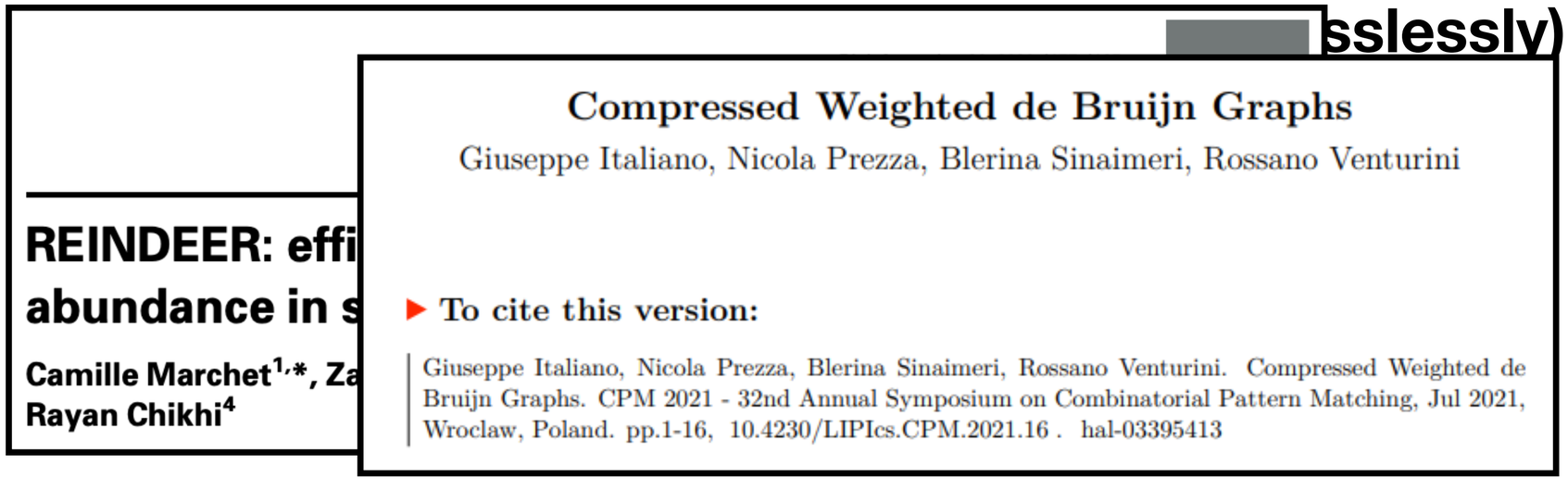
Background

There exist different techniques for representing Annotated DBG

- VARI **[Muggli et al., 2017]**
- Rainbowfish **[Almodaresi et al., 2017]**
- Mantis-MST **[Almodaresi et al., 2019]**
- Multi-BRWT **[Karasikov et al., 2019]**
- Bloom filter-based **[Bradley et al., 2019]**
- RowDiff **[Danciu et al., 2021]**
- MetaGraph **[Karasikov et al., 2020]**
- … and more

Our goals will be to **efficiently** represent:

1. **k-mer abundances**



Background

There exist different techniques for representing Annotated DBG

- VARI **[Muggli et al., 2017]**
- Rainbowfish **[Almodaresi et al., 2017]**
- Mantis-MST **[Almodaresi et al., 2019]**
- Multi-BRWT **[Karasikov et al., 2019]**
- Bloom filter-based **[Bradley et al., 2019]**
- RowDiff **[Danciu et al., 2021]**
- MetaGraph **[Karasikov et al., 2020]**
- … and more

Our goals will be to **efficiently** represent:

1. **k-mer abundances**
2. **k-mer coordinates** (to encode sequences **losslessly)**

Background

There exist different techniques for representing Annotated DBG

- VARI **[Muggli et al., 2017]**
- Rainbowfish **[Almodaresi et al., 2017]**
- Mantis-MST **[Almodaresi et al., 2019]**
- Multi-BRWT **[Karasikov et al., 2019]**
- Bloom filter-based **[Bradley et al., 2019]**
- RowDiff **[Danciu et al., 2021]**
- MetaGraph **[Karasikov et al., 2020]**
- … and more

Our goals will be to **efficiently** represent:

1. **k-mer abundances**
2. **k-mer coordinates** (to encode sequences **losslessly)**

(Pufferfish)

Bioinformatics, 34, 2018, i169–i177
doi: 10.1093/bioinformatics/bty292
ISMB 2018

OXFORD

A space and time-efficient index for the compacted colored de Bruijn graph

Fatemeh Almodaresi†, Hirak Sarkar†, Avi Srivastava and Rob Patro*

Background

There exist different techniques for representing Annotated DBG

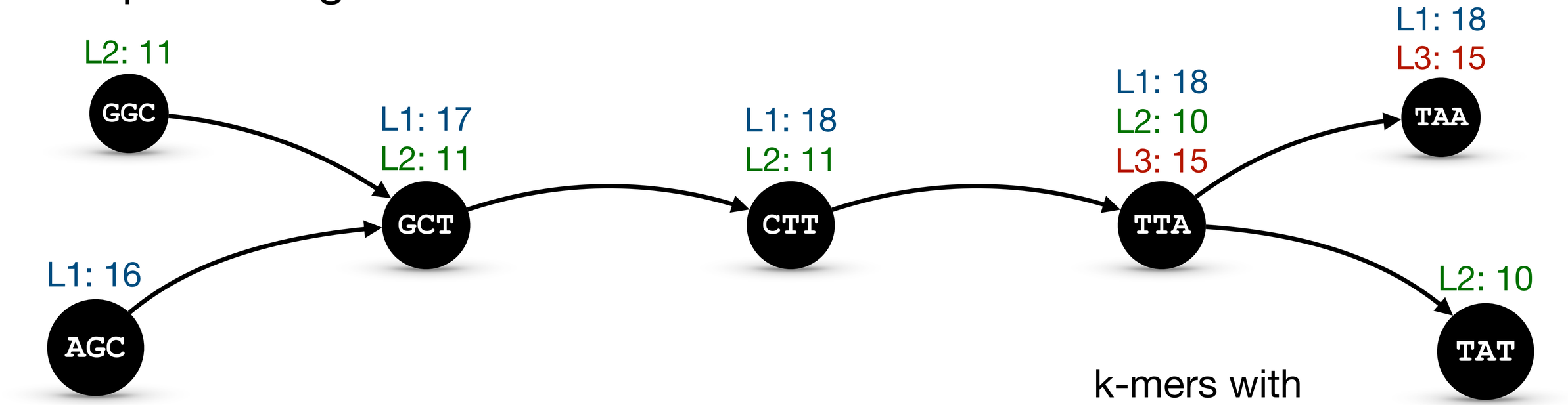
- VARI **[Muggli et al., 2017]**
- Rainbowfish **[Almodaresi et al., 2017]**
- Mantis-MST **[Almodaresi et al., 2019]**
- Multi-BRWT **[Karasikov et al., 2019]**
- Bloom filter-based **[Bradley et al., 2019]**
- RowDiff **[Danciu et al., 2021]**
- MetaGraph **[Karasikov et al., 2020]**
- … and more

Our goals will be to **efficiently** represent:

1. **k-mer abundances**
2. **k-mer coordinates** (to encode sequences **losslessly)**

Motivation

1. Representing **k-mer abundances**



k-mers with multiplicities

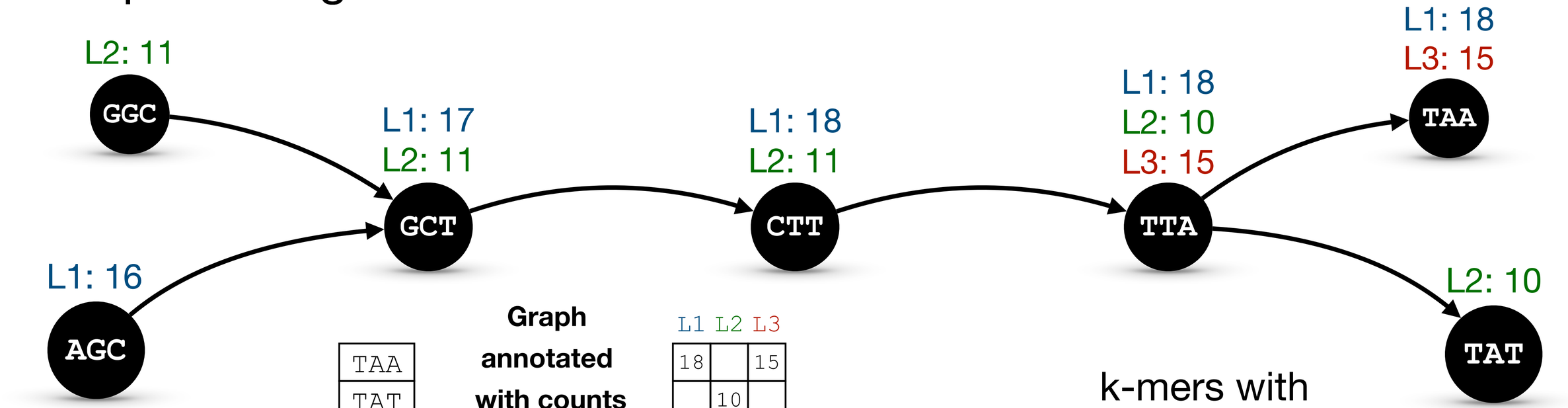
L1: `AGC: 16, …`

L2: `GGC: 11, …`

L3: `TTA: 15, …`

Motivation

1. Representing **k-mer abundances**



TAA
TAT
GCT
AGC
GGC
CTT
TTA

k-mer dictionary
(de Bruijn Graph)

Graph annotated with counts

L1	L2	L3
18		15
	10	
17	11	
16		
	11	
18	11	
18	10	15

Graph annotation
(labeling)

k-mers with multiplicities

L1: AGC: 16, …

L2: GGC: 11, …

L3: TTA: 15, …

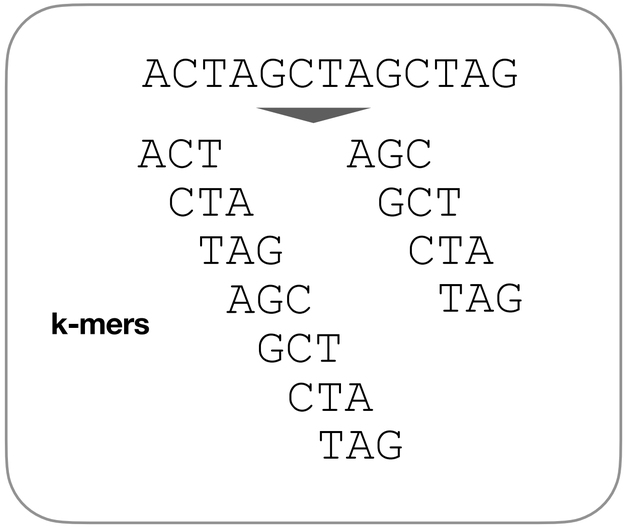
Motivation

2. Representing **k-mer coordinates**

ACTAGCTAGCTAG

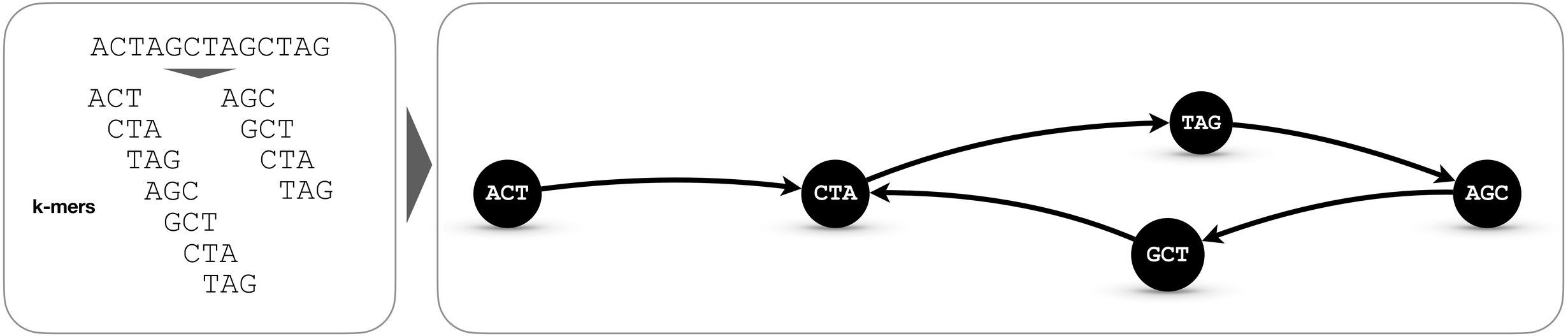
Motivation

2. Representing **k-mer coordinates**



Motivation

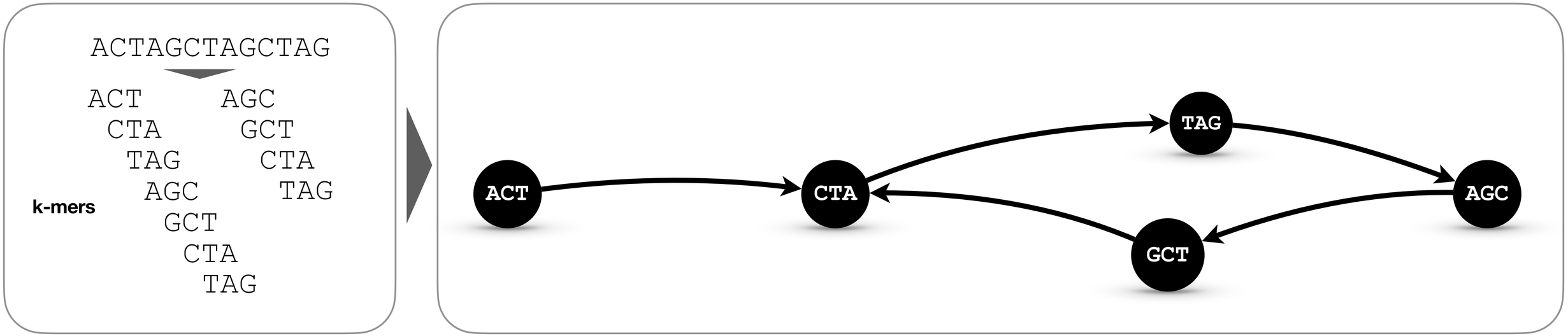
2. Representing **k-mer coordinates**



de Bruijn graph

Motivation

2. Representing **k-mer coordinates**

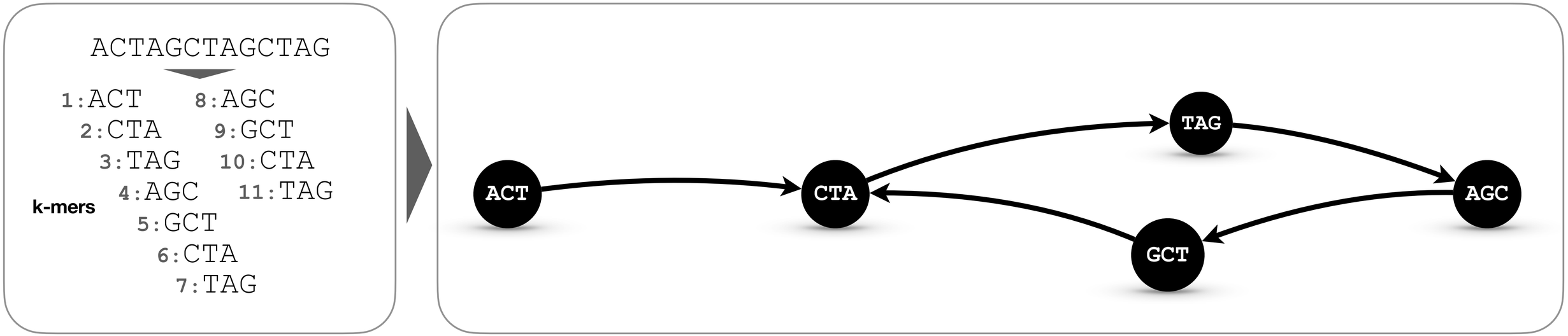


Not invertible representation

de Bruijn graph

Motivation

2. Representing **k-mer coordinates**

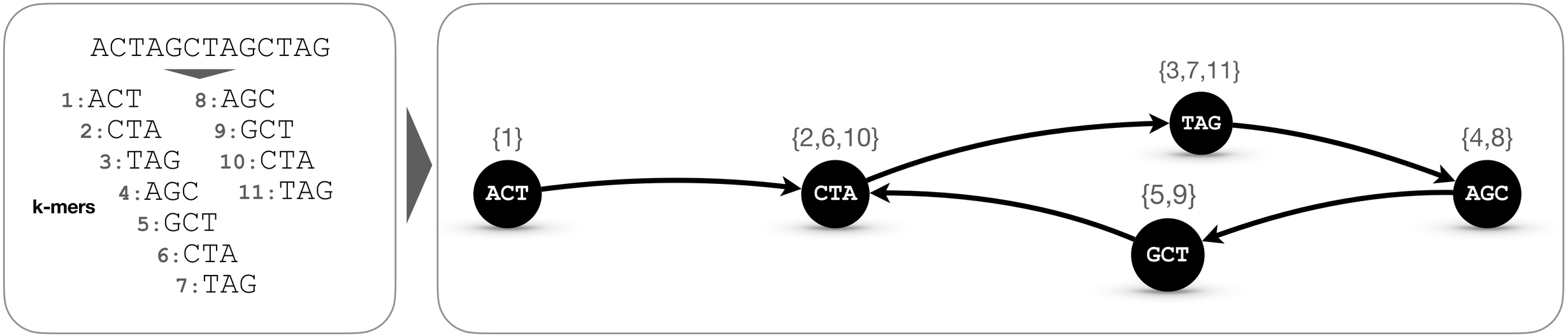


Not invertible representation

de Bruijn graph

Motivation

2. Representing **k-mer coordinates**



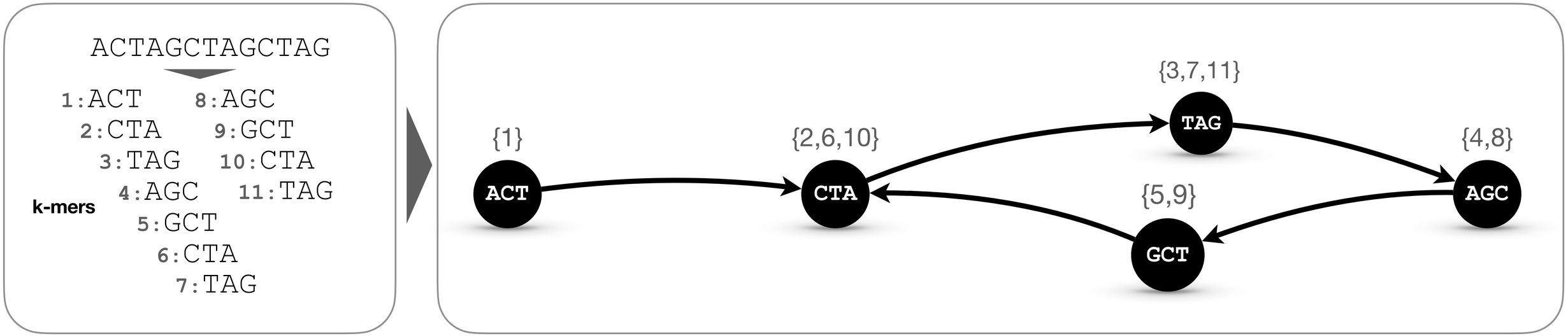
Invertible

de Bruijn graph

with k-mer coordinates

Motivation

2. Representing **k-mer coordinates**



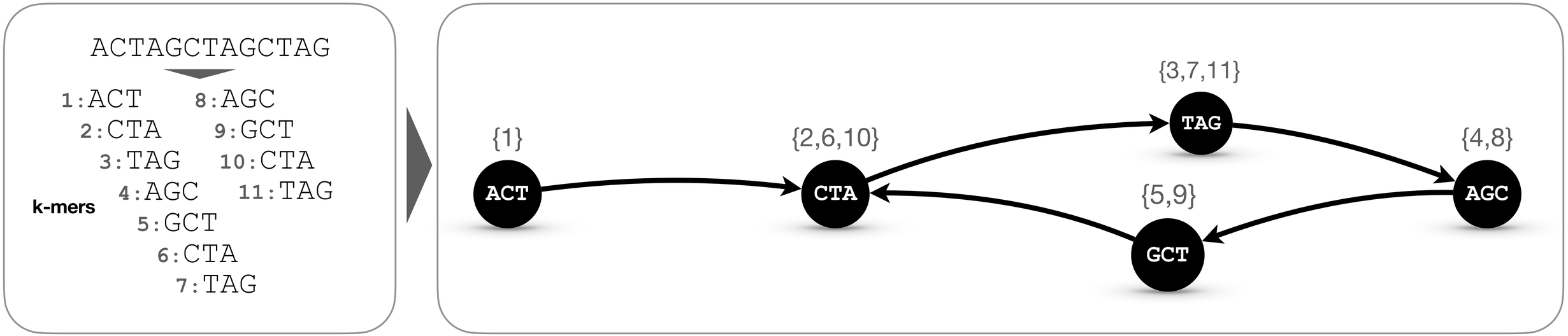
Encoding ***sequence traces*** allows:

de Bruijn graph

with k-mer coordinates

Motivation

2. Representing **k-mer coordinates**



Encoding ***sequence traces*** allows:

- reconstructing indexed sequences

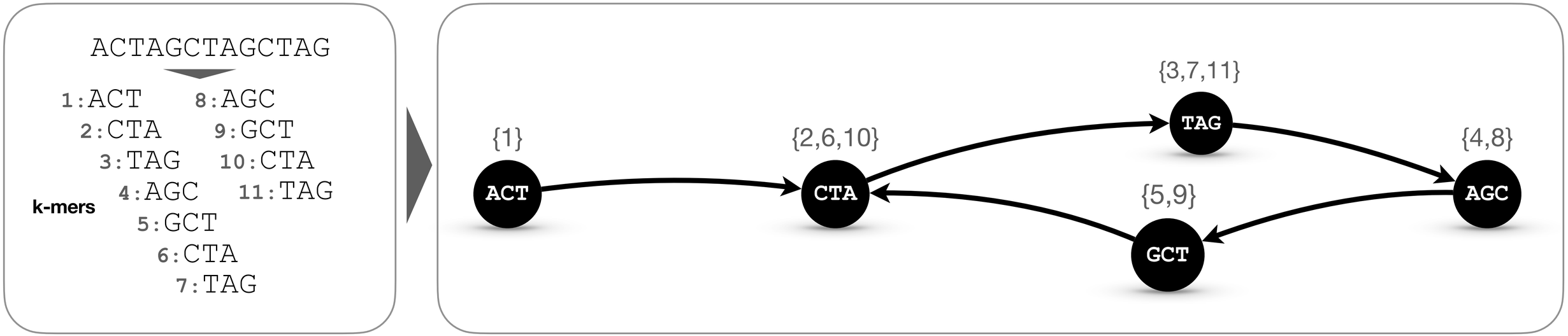
  hence, **lossless sequence representation**

de Bruijn graph

with k-mer coordinates

Motivation

2. Representing **k-mer coordinates**



Encoding ***sequence traces*** allows:

- reconstructing indexed sequences

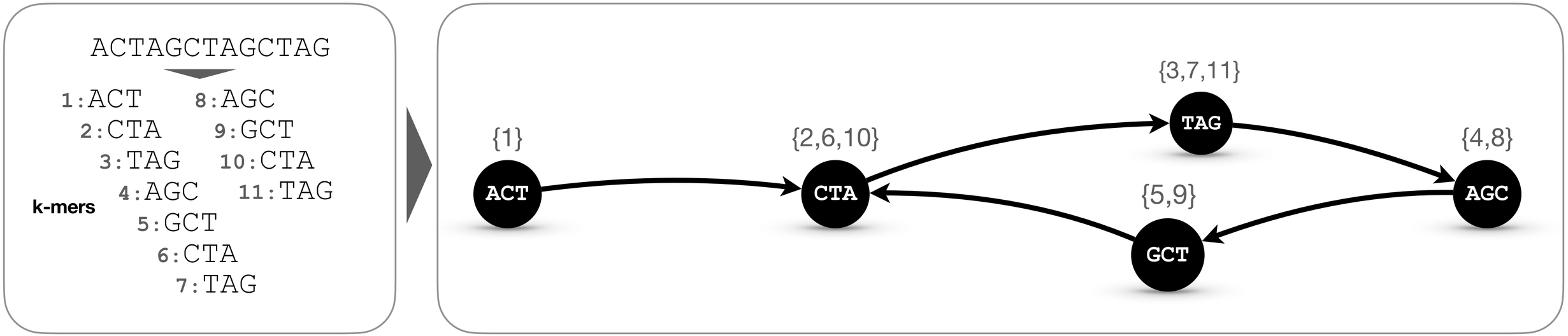
  hence, **lossless sequence representation**
- performing exact sequence alignment

de Bruijn graph

with k-mer coordinates

Motivation

2. Representing **k-mer coordinates**



Encoding ***sequence traces*** allows:

- reconstructing indexed sequences

  hence, **lossless sequence representation**
- performing exact sequence alignment
- getting all **traces** crossing a node

de Bruijn graph

with k-mer coordinates

Challenges

Challenges

Questions to address:

1. How to efficiently represent a huge sparse non-binary matrix?

Challenges

Questions to address:

1. How to efficiently represent a huge sparse non-binary matrix?
2. Can we employ existing repr. schemes for binary matrices?

Challenges

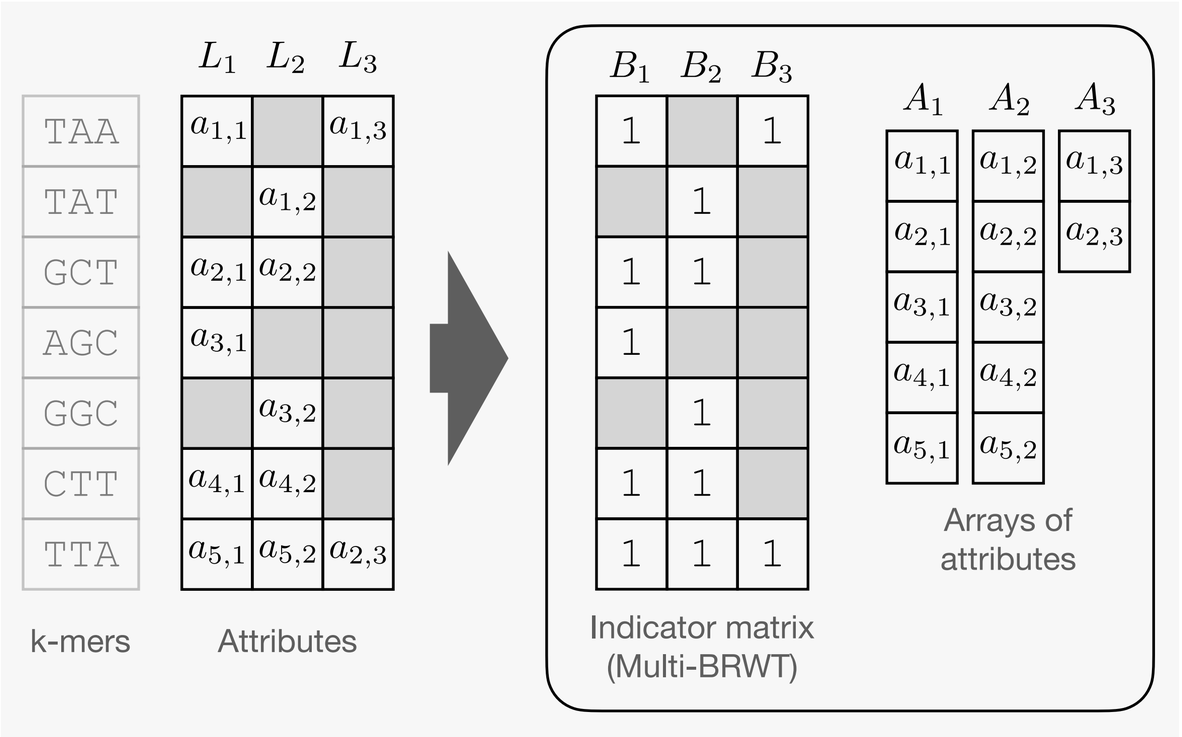
Questions to address:

1. How to efficiently represent a huge sparse non-binary matrix?
2. Can we employ existing repr. schemes for binary matrices?
3. How to exploit regularities in graph annotations?

Method

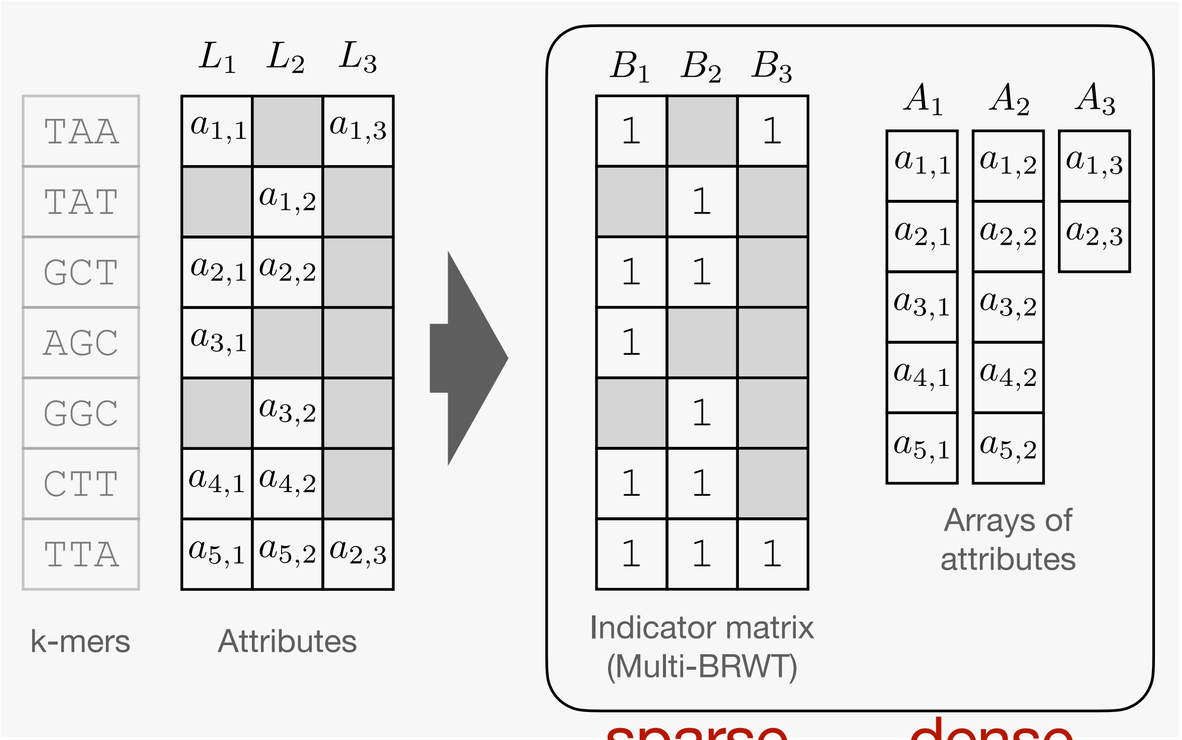
Method

General scheme for sparse matrices



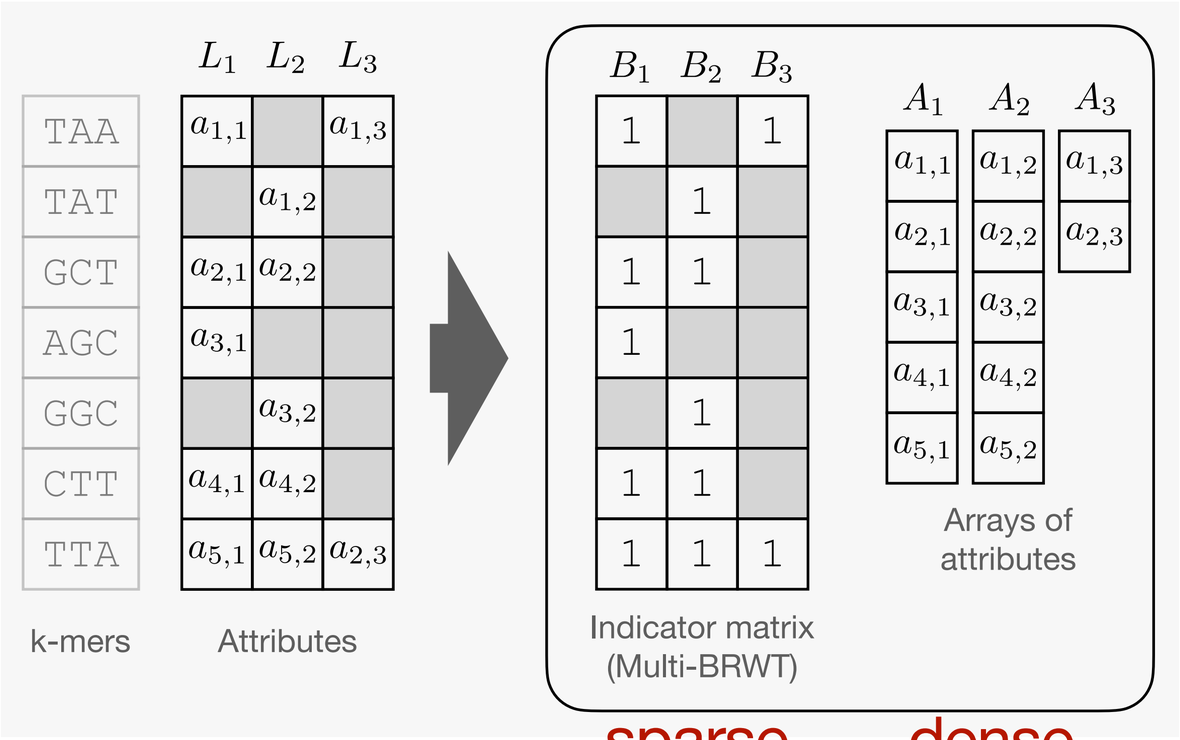
Method

General scheme for sparse matrices



Method

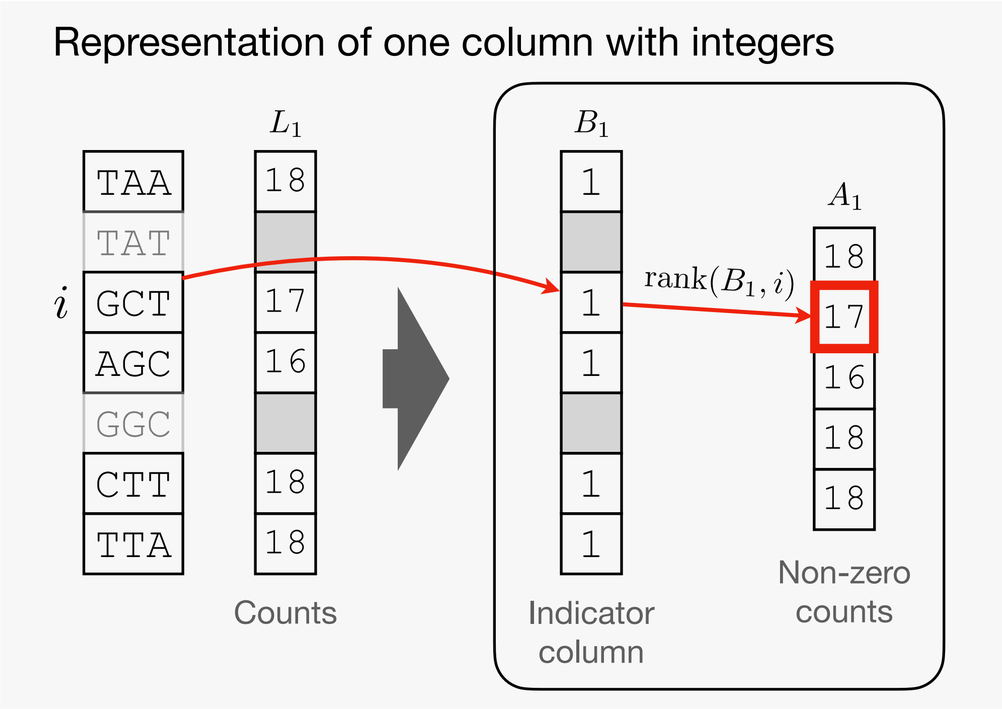
General scheme for sparse matrices

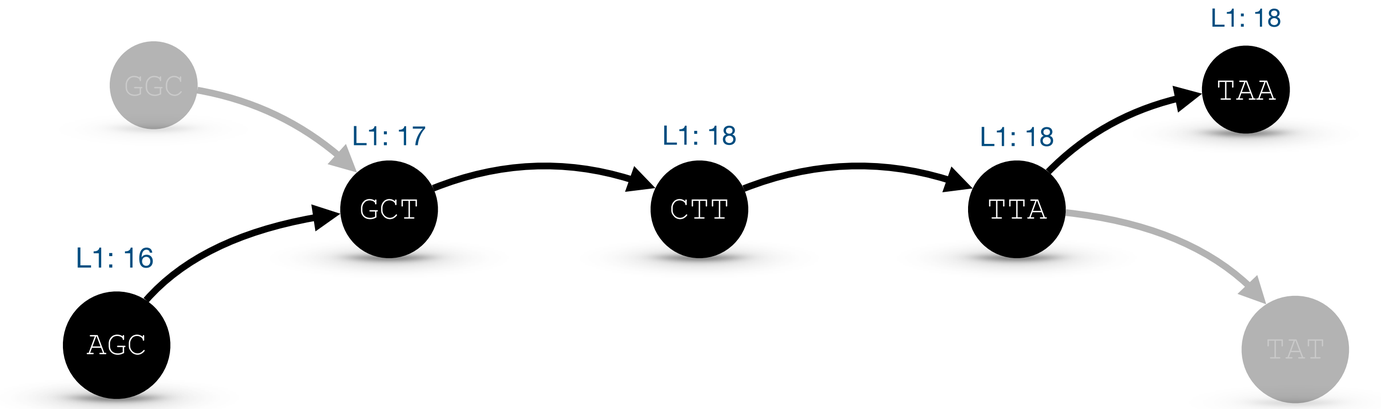


Theorem 1. *If both the indicator matrix and the arrays of attributes are represented succinctly, the proposed scheme also is a succinct representation of the matrix. That is, there is no other data structure that could represent any such matrix with asymptotically fewer bits.*

Method

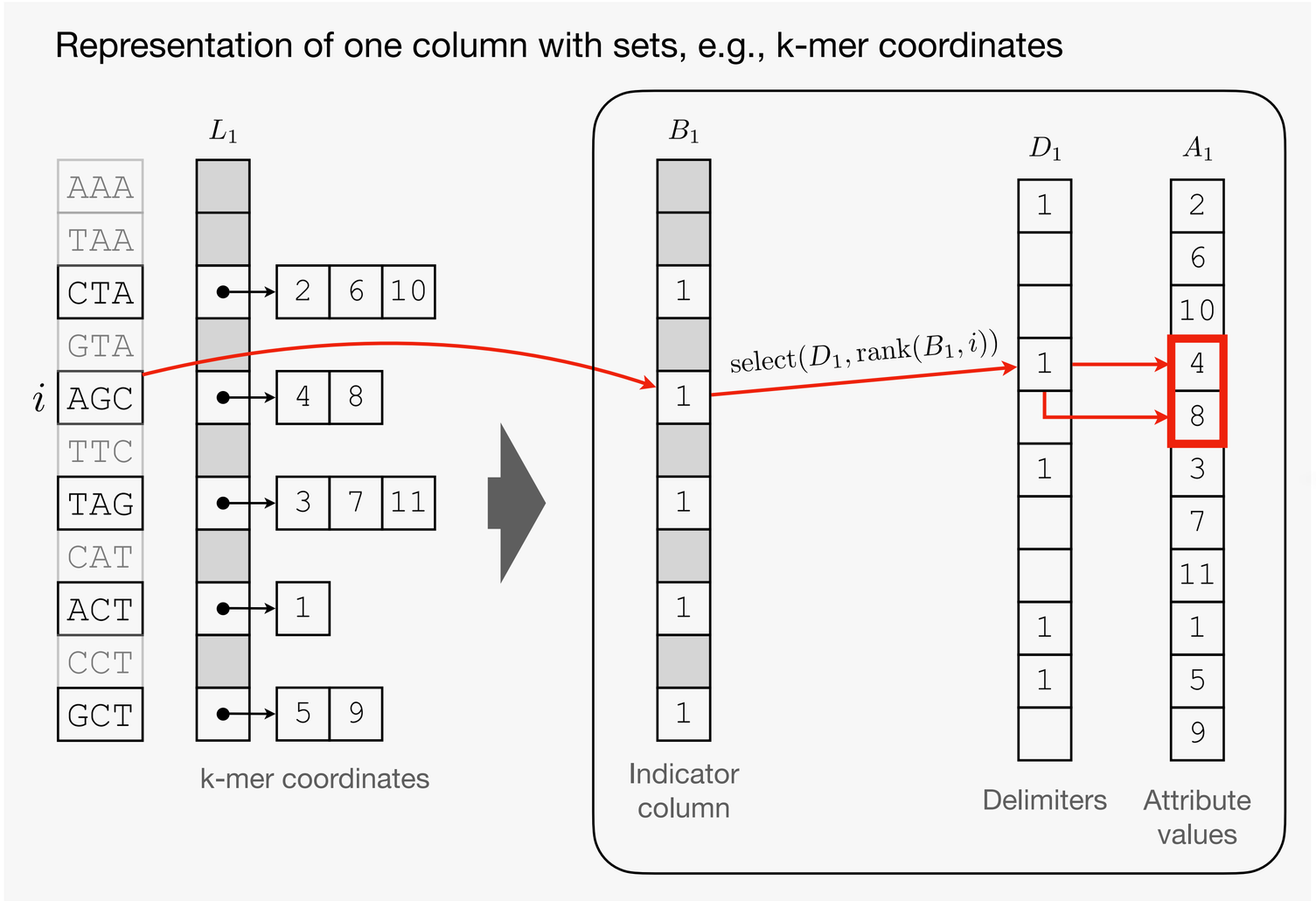
1. One column with **integers** (k-mer abundances)

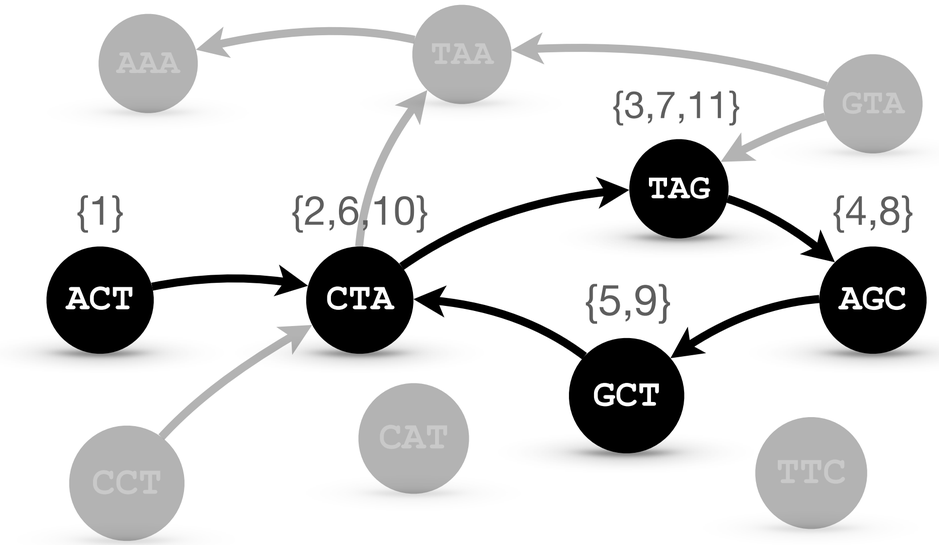




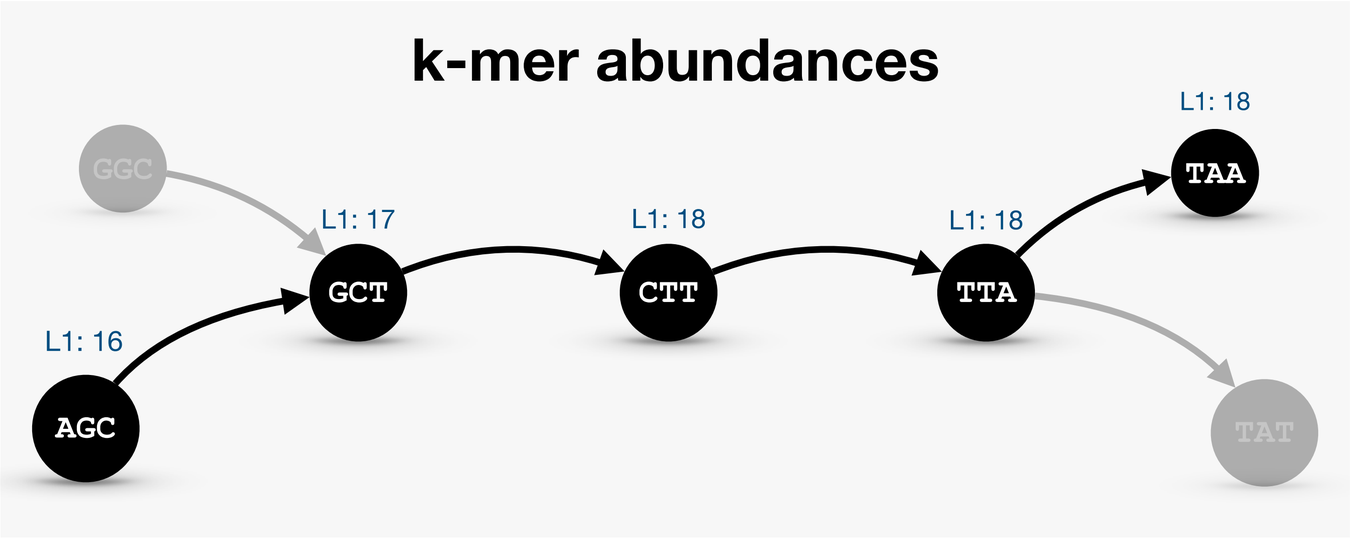
Method

2. One column with integer **tuples** (k-mer coordinates)





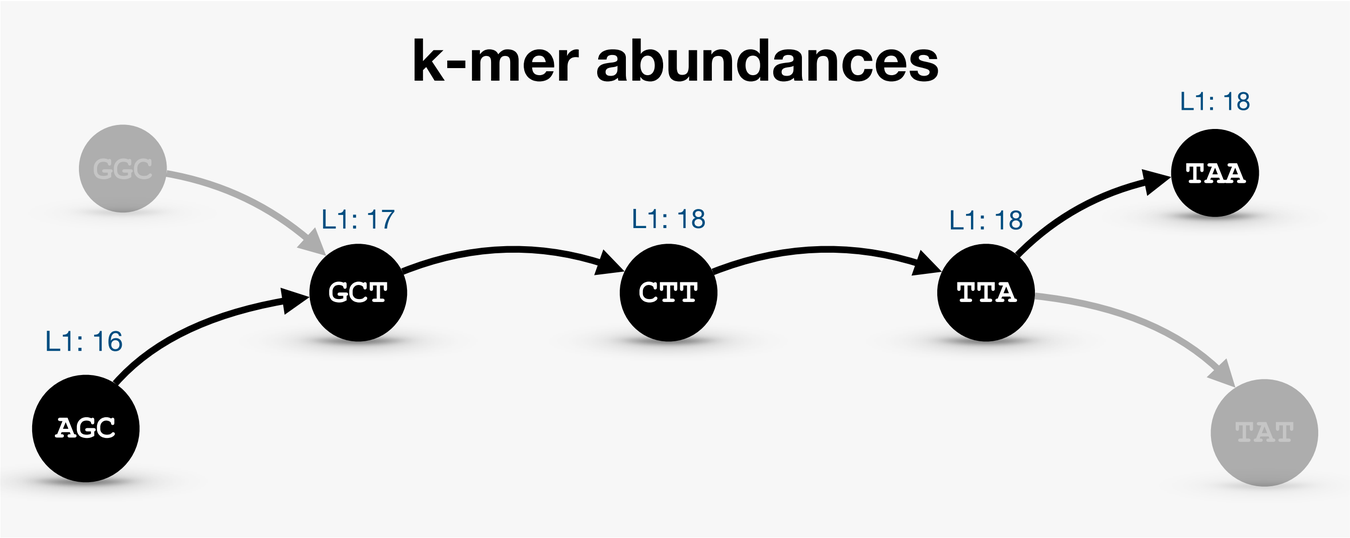
Exploiting regularities

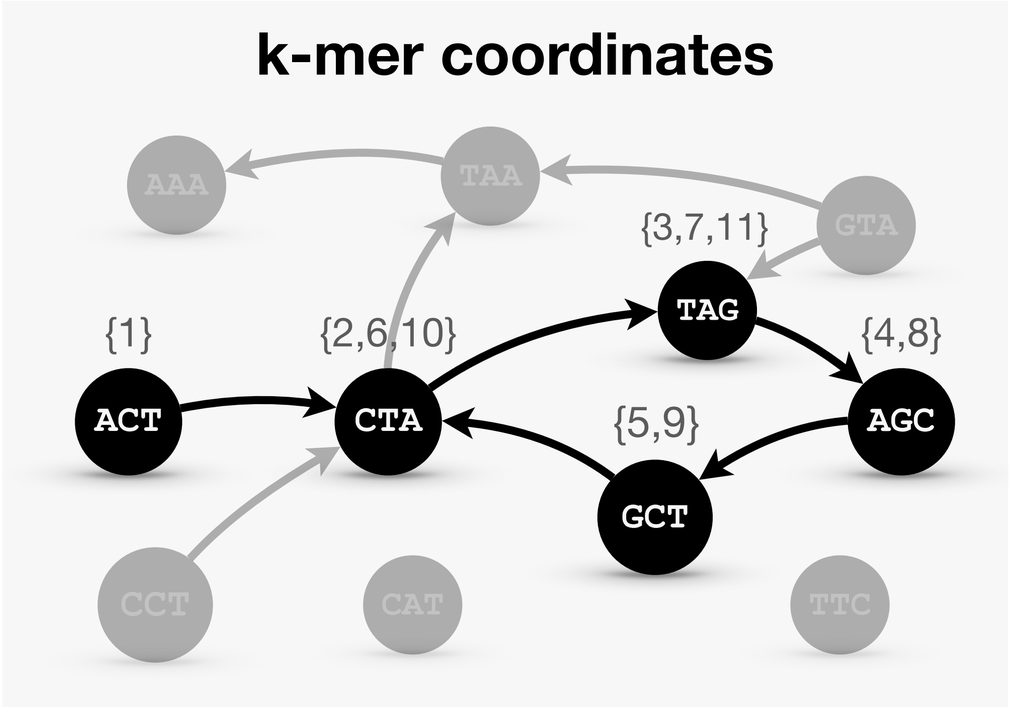


Observe regularities:

- Abundances of neighboring k-mers are often similar

Exploiting regularities

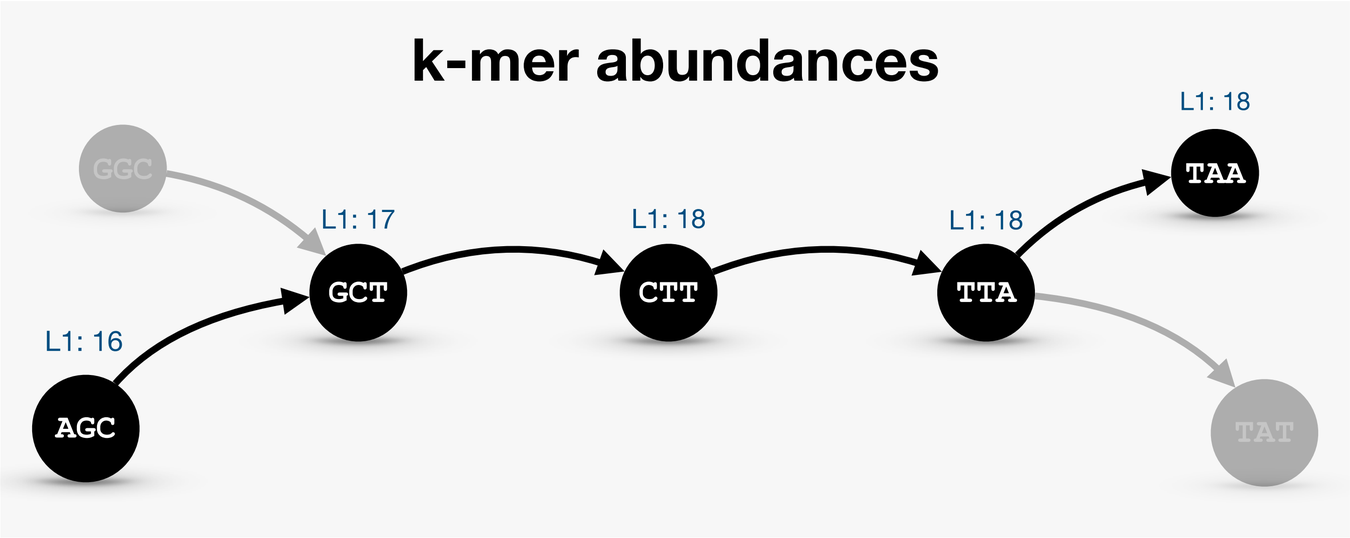


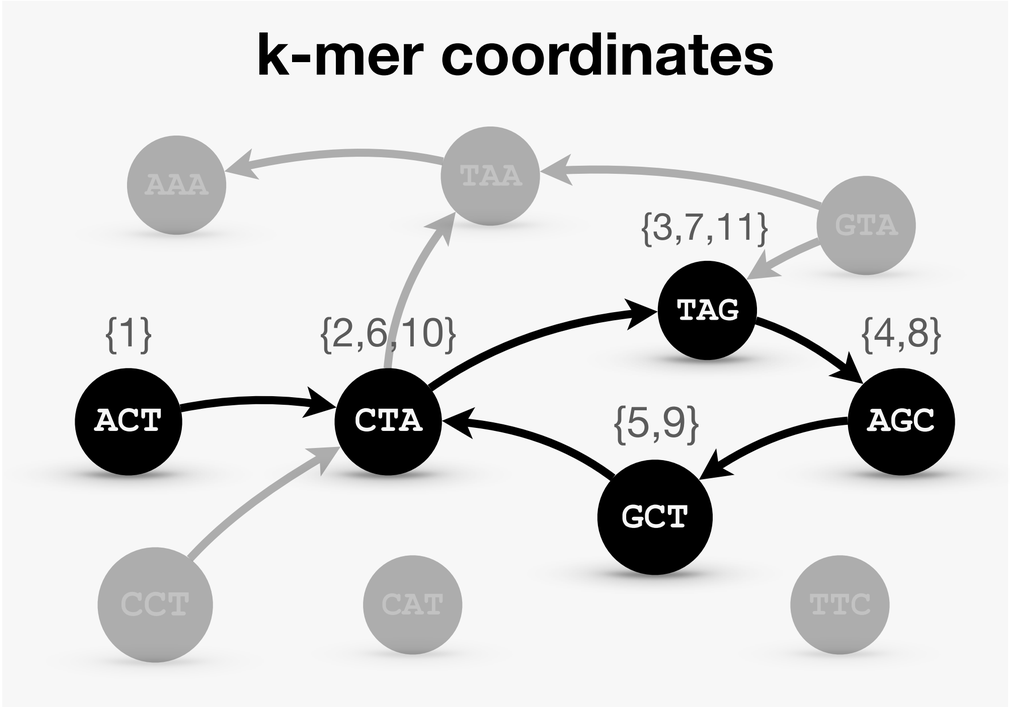


Observe regularities:

- Abundances of neighboring k-mers are often similar
- Coordinates for adjacent k-mers always differ by +1

Exploiting regularities





Observe regularities:

- Abundances of neighboring k-mers are often similar
- Coordinates for adjacent k-mers always differ by +1

Idea

Generalize the RowDiff scheme


Bioinformatics, 37, 2021, i169–i176
doi: 10.1093/bioinformatics/btab330
ISMB/ECCB 2021
OXFORD

Topology-based sparsification of graph annotations

Daniel Danciu[1,2,†], Mikhail Karasikov[1,2,3,†], Harun Mustafa[1,2,3], André Kahles[1,2,3,*] and Gunnar Rätsch[1,2,3,4,*]

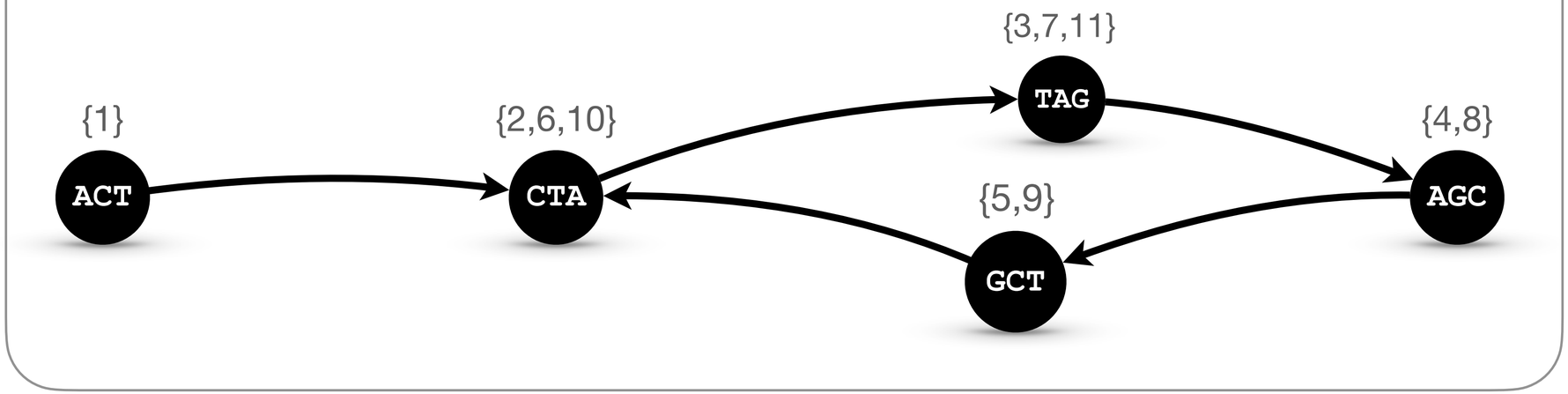
Delta coding for k-mer coordinates

A. Enumeration of k-mers

```
ACTAGCTAGCTAG

1:ACT    8:AGC
 2:CTA    9:GCT
  3:TAG   10:CTA
   4:AGC  11:TAG
    5:GCT
     6:CTA
      7:TAG
```

B. Delta-coding of coordinate annotations (sets)



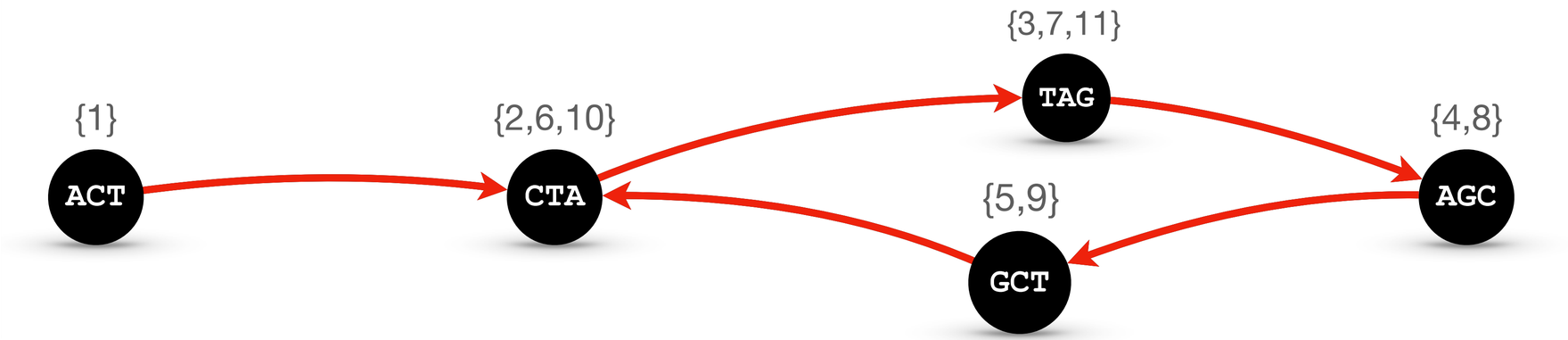
Delta coding for k-mer coordinates

A. Enumeration of k-mers

ACTAGCTAGCTAG

1:ACT
2:CTA
3:TAG
4:AGC
5:GCT
6:CTA
7:TAG
8:AGC
9:GCT
10:CTA
11:TAG

B. Delta-coding of coordinate annotations (sets)



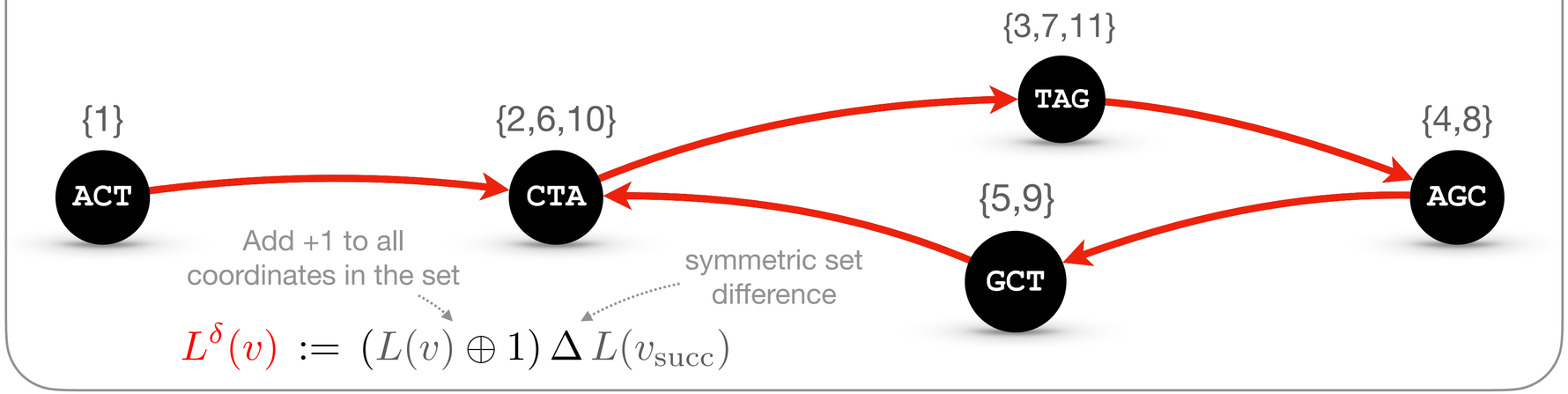
Delta coding for k-mer coordinates

A. Enumeration of k-mers

ACTAGCTAGCTAG

1:ACT
2:CTA
3:TAG
4:AGC
5:GCT
6:CTA
7:TAG
8:AGC
9:GCT
10:CTA
11:TAG

B. Delta-coding of coordinate annotations (sets)



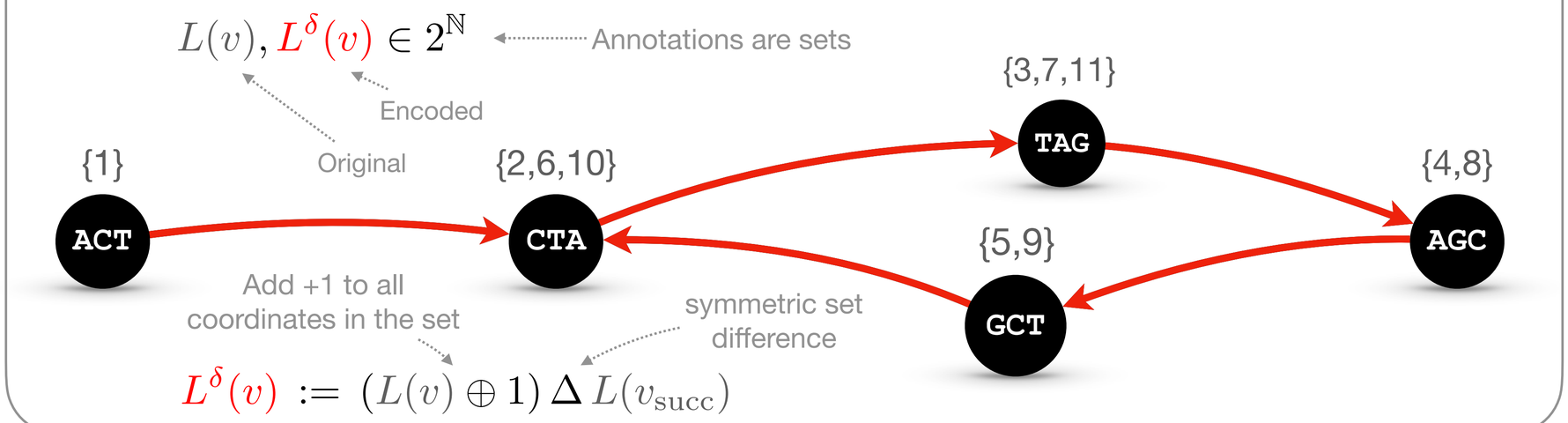
Delta coding for k-mer coordinates

A. Enumeration of k-mers

ACTAGCTAGCTAG

1:ACT
2:CTA
3:TAG
4:AGC
5:GCT
6:CTA
7:TAG
8:AGC
9:GCT
10:CTA
11:TAG

B. Delta-coding of coordinate annotations (sets)



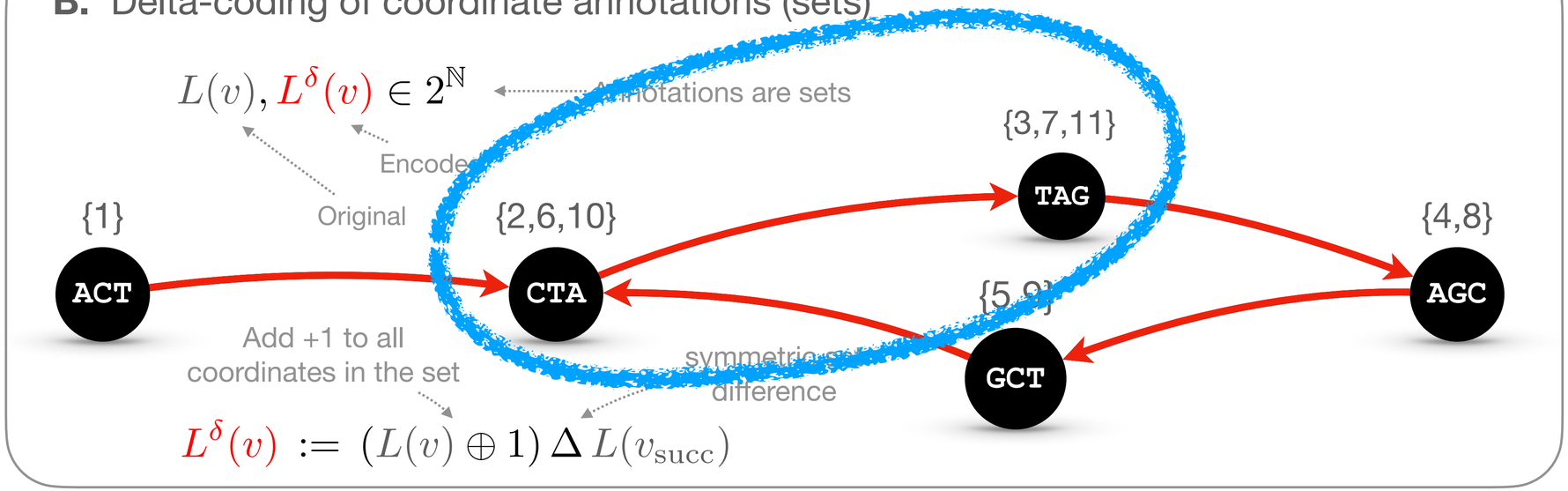
Delta coding for k-mer coordinates

A. Enumeration of k-mers

ACTAGCTAGCTAG

1:ACT
2:CTA
3:TAG
4:AGC
5:GCT
6:CTA
7:TAG
8:AGC
9:GCT
10:CTA
11:TAG

B. Delta-coding of coordinate annotations (sets)



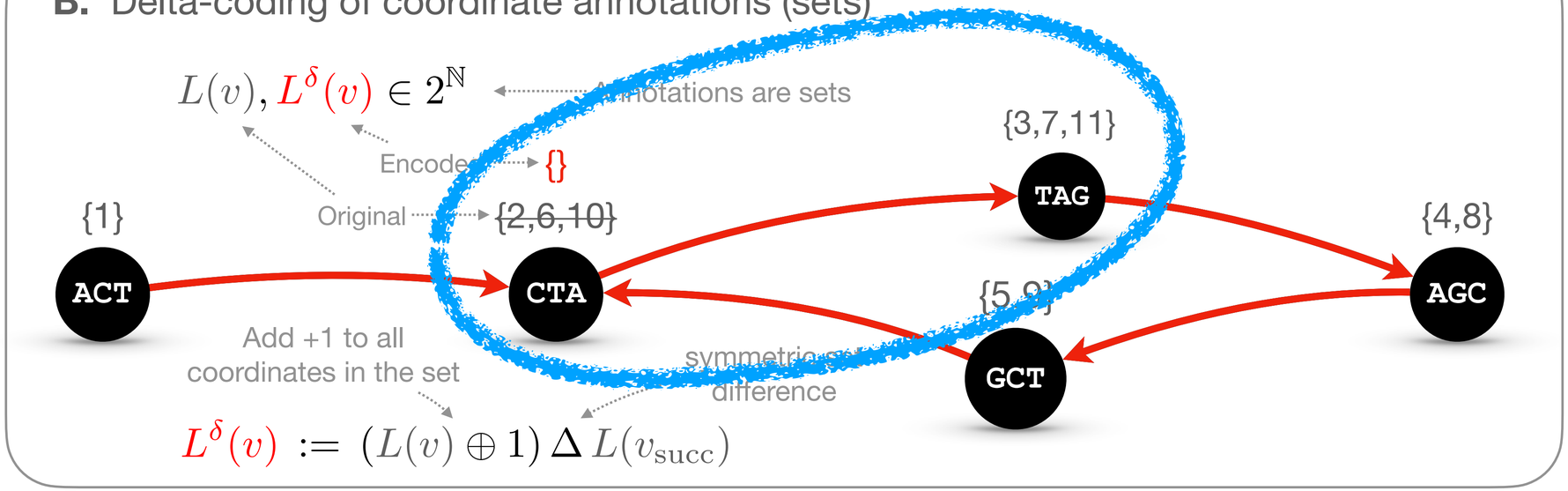
Delta coding for k-mer coordinates

A. Enumeration of k-mers

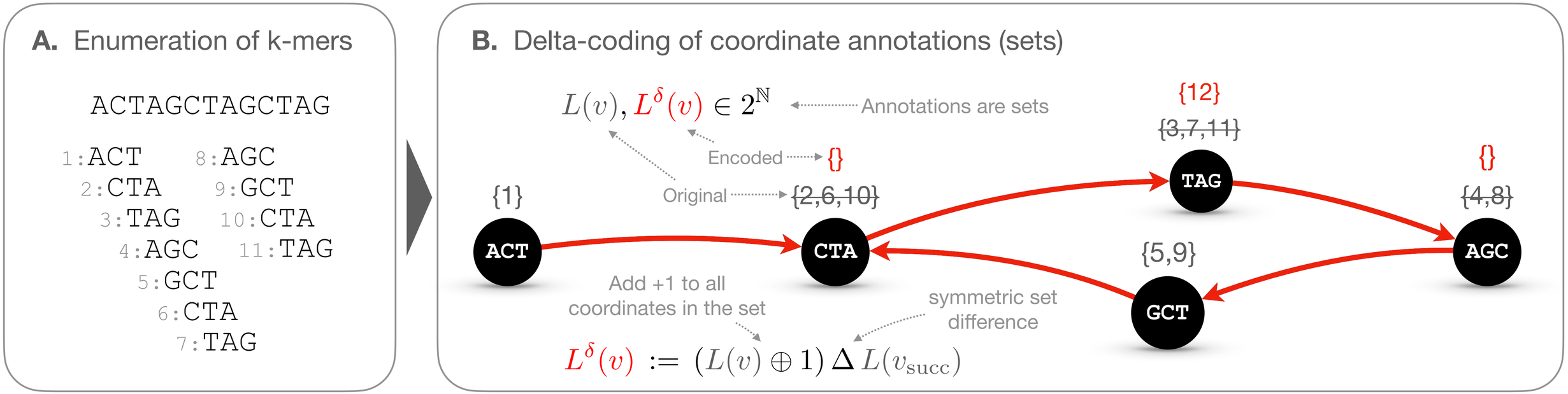
ACTAGCTAGCTAG

1:ACT
2:CTA
3:TAG
4:AGC
5:GCT
6:CTA
7:TAG
8:AGC
9:GCT
10:CTA
11:TAG

B. Delta-coding of coordinate annotations (sets)



Delta coding for k-mer coordinates



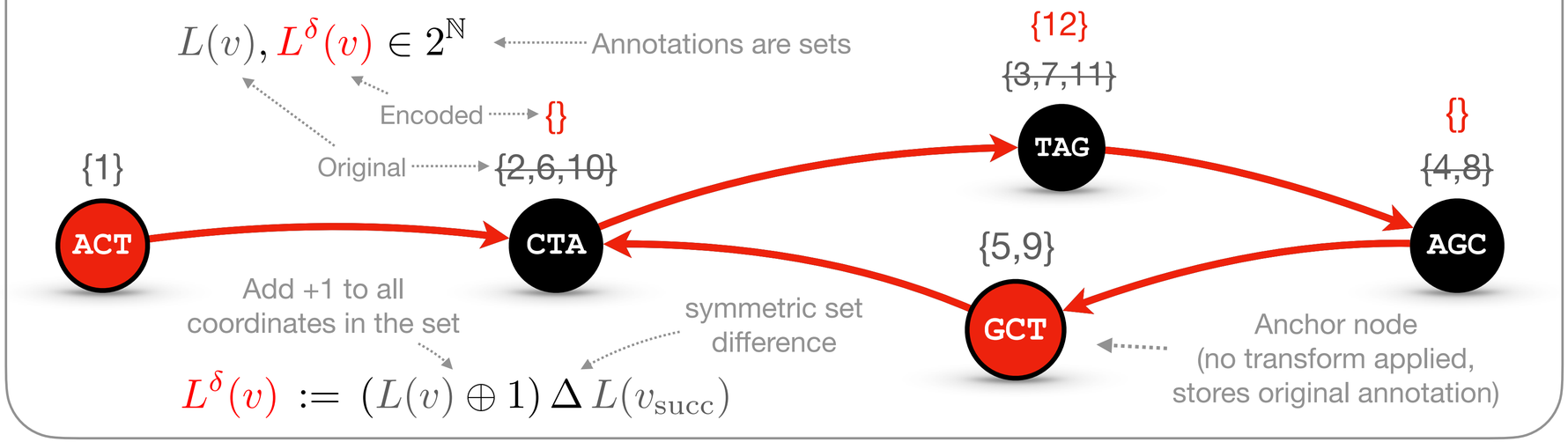
Delta coding for k-mer coordinates

A. Enumeration of k-mers

ACTAGCTAGCTAG

1:ACT
2:CTA
3:TAG
4:AGC
5:GCT
6:CTA
7:TAG
8:AGC
9:GCT
10:CTA
11:TAG

B. Delta-coding of coordinate annotations (sets)



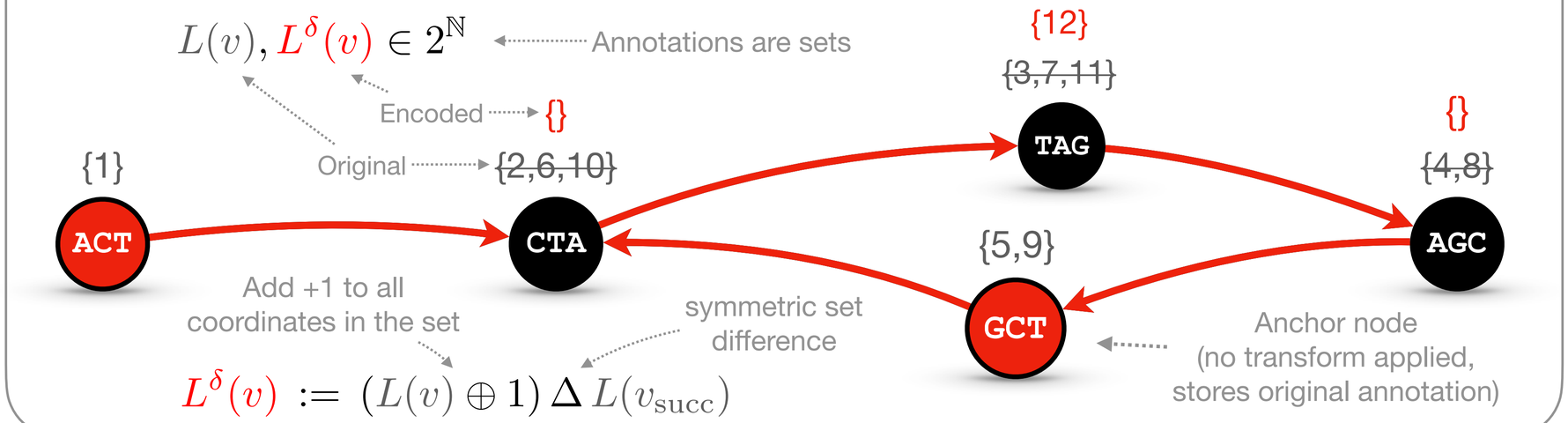
Delta coding for k-mer coordinates

A. Enumeration of k-mers

ACTAGCTAGCTAG

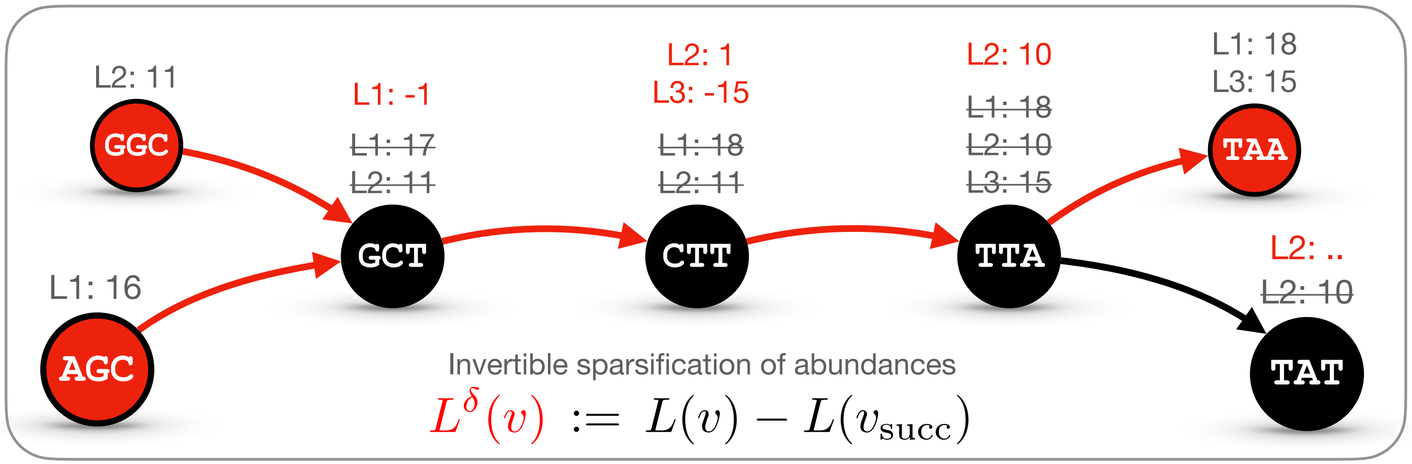
1:ACT 8:AGC
2:CTA 9:GCT
3:TAG 10:CTA
4:AGC 11:TAG
5:GCT
6:CTA
7:TAG

B. Delta-coding of coordinate annotations (sets)

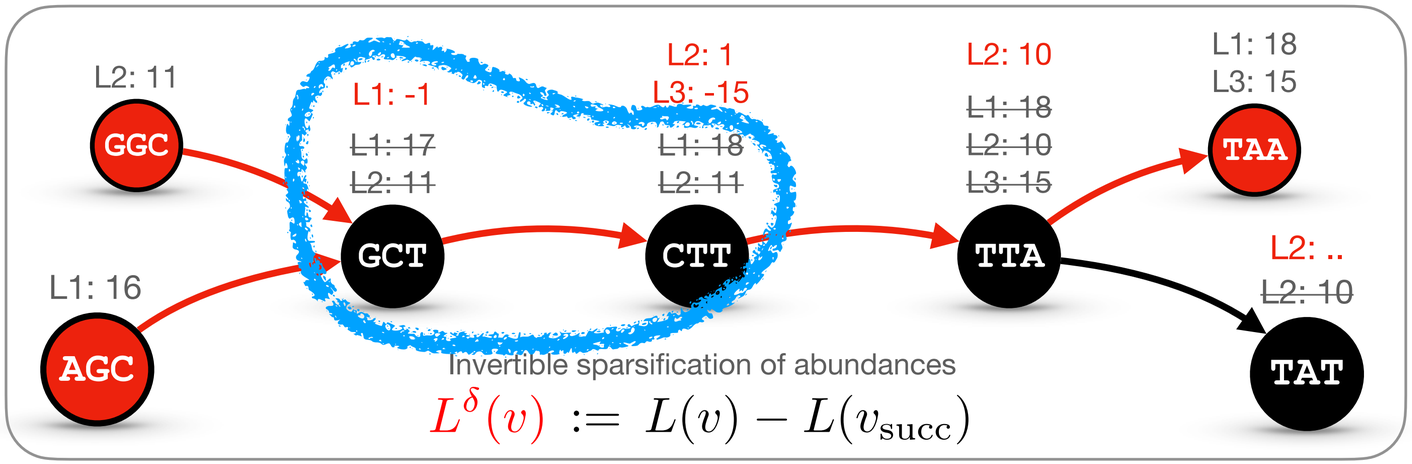


Decoding: $L(v) = (L^{\delta}(v)\, \Delta\, L(v_{\text{succ}})) \ominus 1$

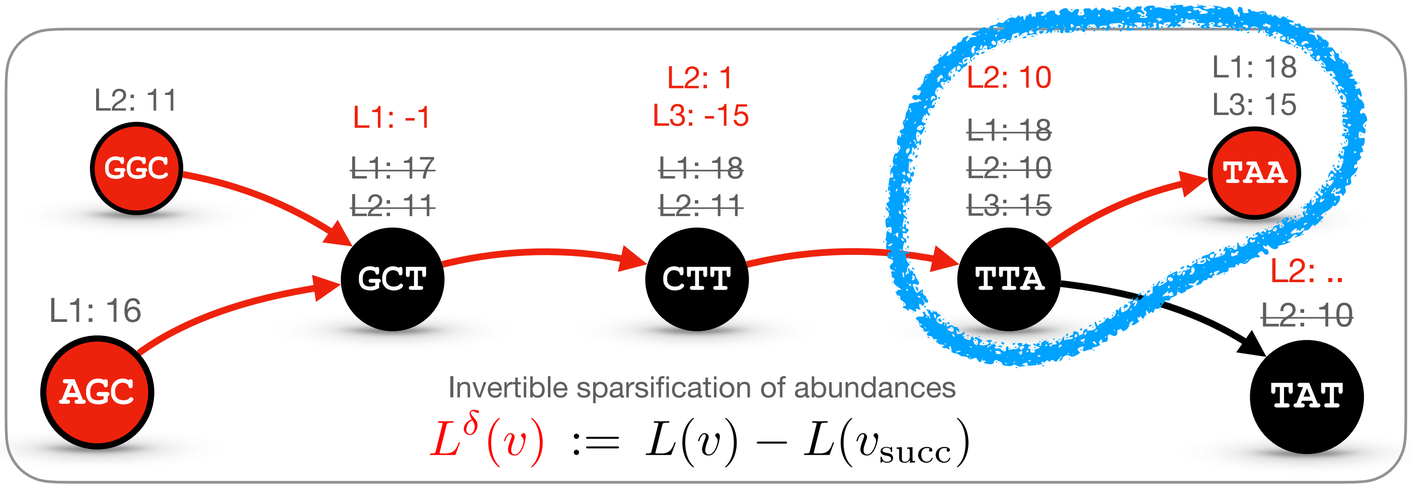
Delta coding for k-mer abundances



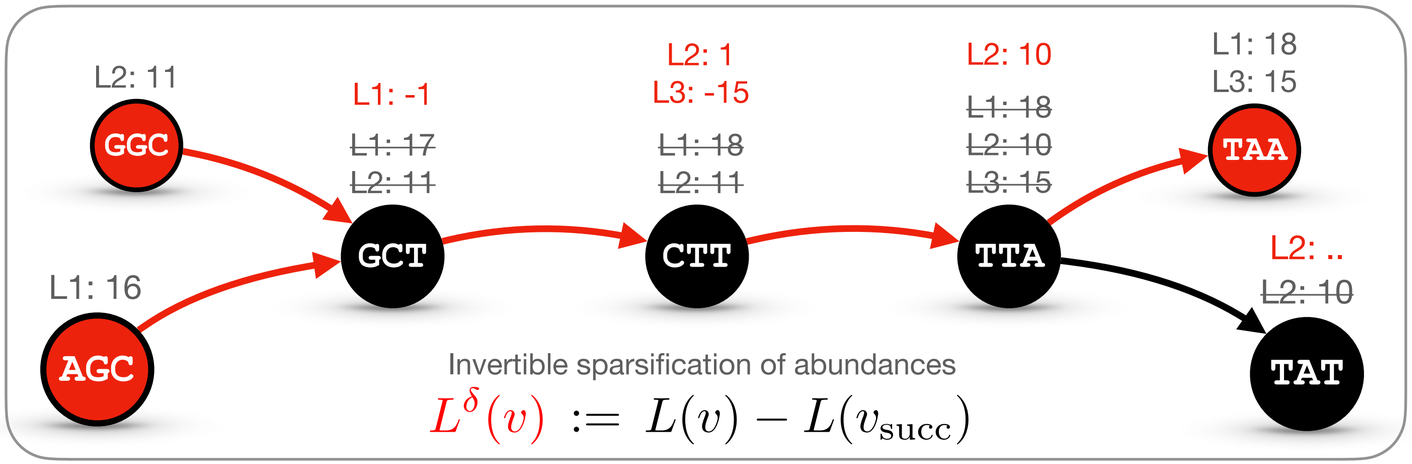
Delta coding for k-mer abundances



Delta coding for k-mer abundances

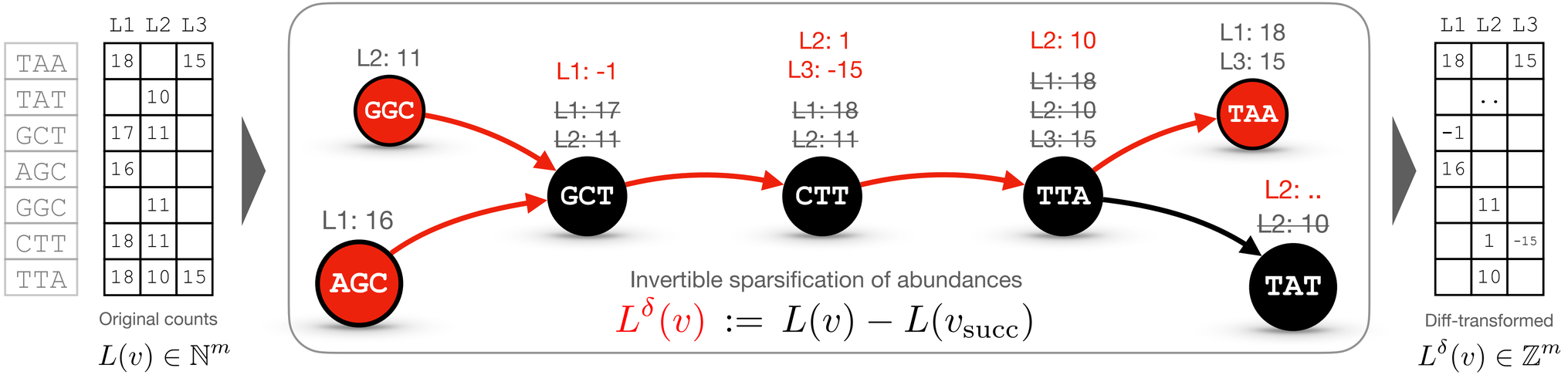


Delta coding for k-mer abundances



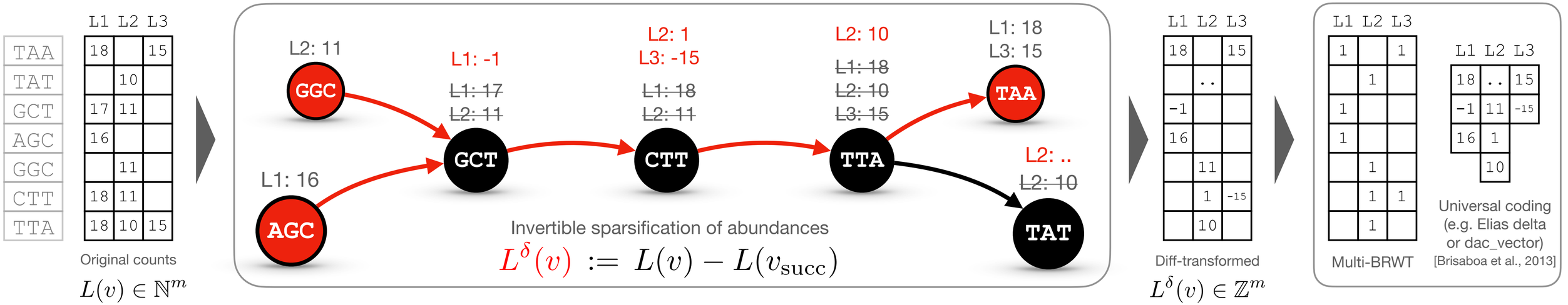
Decoding: $L(v) = L(v_{\text{succ}}) + L^{\delta}(v)$

Delta coding for k-mer abundances



Decoding: $L(v) = L(v_{\text{succ}}) + L^\delta(v)$

Delta coding for k-mer abundances



Decoding: $L(v) = L(v_{\text{succ}}) + L^\delta(v)$

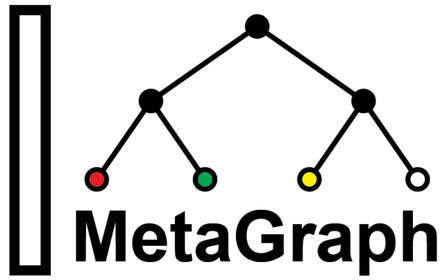
Method

Counting DBG: Implementation

Repository: github.com/ratschlab/counting_dbg

Implemented within the MetaGraph framework

- Succinct graph representations (based on the BOSS table)
- Graph annotation representations (e.g., `Multi-BRWT`)
- Hybrid bit vector representations
- Procedures for query and alignment



Base succinct data structures from sdsl-lite (Succinct Data Structure Library)

- Compressed and packed bitmaps
- Compressed integer arrays (`sdsl::dac_vector_dp<>`)

Experiments

1. Indexing k-mer abundances
2. Indexing k-mer coordinates
    - **Encoding traces** for lossless indexing of sequences
    - Trace Consistent Graph Alignment **(TCG-Aligner)**

Indexing k-mer abundances

Data: 2,586 Illumina RNA-Seq read sets*

On querying 100 random human transcripts (≈ 90 kbp in total)

(excluding index loading)

	Index size			Peak RAM during query			Query time		
Method	**binary**	**smooth counts**	**raw counts**	**binary**	**smooth counts**	**raw counts**	**binary**	**smooth counts**	**raw counts**
Mantis-MST	24.9 GB	-	-	25.1 GB	-	-	**0.6 s**	-	-
RowDiff	7.7 GB	-	-	8.0 GB	-	-	8.6 s	-	-
REINDEER	30.3 GB	59 GB	-	58.9 GB	91 GB	-	53.1 s	56.5 s	-
This work	**6.6 GB**	**11 GB**	**21 GB**	**6.9 GB**	**11 GB**	**21 GB**	6.4 s	**17.6 s**	**21.2 s**

* read sets from [Solomon and Kingsford, 2018]

Indexing k-mer abundances

Data: 2,586 Illumina RNA-Seq read sets*

On querying 100 random human transcripts (≈ 90 kbp in total)

(excluding index loading)

	Index size			Peak RAM during query			Query time		
Method	binary	smooth counts	raw counts	binary	smooth counts	raw counts	binary	smooth counts	raw counts
Mantis-MST	24.9 GB	-	-	25.1 GB	-	-	**0.6 s**	-	-
RowDiff	7.7 GB	-	-	8.0 GB	-	-	8.6 s	-	-
REINDEER	30.3 GB	59 GB	-	58.9 GB	91 GB	-	53.1 s	56.5 s	-
This work	**6.6 GB**	**11 GB**	**21 GB**	**6.9 GB**	**11 GB**	**21 GB**	6.4 s	**17.6 s**	**21.2 s**

- Generates **5x smaller** representations and uses **8x less RAM**

* read sets from [Solomon and Kingsford, 2018]

Indexing k-mer abundances

Data: 2,586 Illumina RNA-Seq read sets*

On querying 100 random human transcripts (≈ 90 kbp in total)

(excluding index loading)

	Index size			Peak RAM during query			Query time		
Method	binary	smooth counts	raw counts	binary	smooth counts	raw counts	binary	smooth counts	raw counts
Mantis-MST	24.9 GB	-	-	25.1 GB	-	-	**0.6 s**	-	-
RowDiff	7.7 GB	-	-	8.0 GB	-	-	8.6 s	-	-
REINDEER	30.3 GB	59 GB	-	58.9 GB	91 GB	-	53.1 s	56.5 s	-
This work	**6.6 GB**	**11 GB**	**21 GB**	**6.9 GB**	**11 GB**	**21 GB**	6.4 s	**17.6 s**	**21.2 s**

- Generates **5x smaller** representations and uses **8x less RAM**
- **3-5x faster** to query

* read sets from [Solomon and Kingsford, 2018]

Experiments

1. Indexing k-mer abundances
2. Indexing k-mer coordinates
    - **Encoding traces** for lossless indexing of sequences
    - Trace Consistent Graph Alignment **(TCG-Aligner)**

Trace Consistent Graph (TCG-) Aligner

Sequence alignment on top of Counting DBG with k-mer coordinates

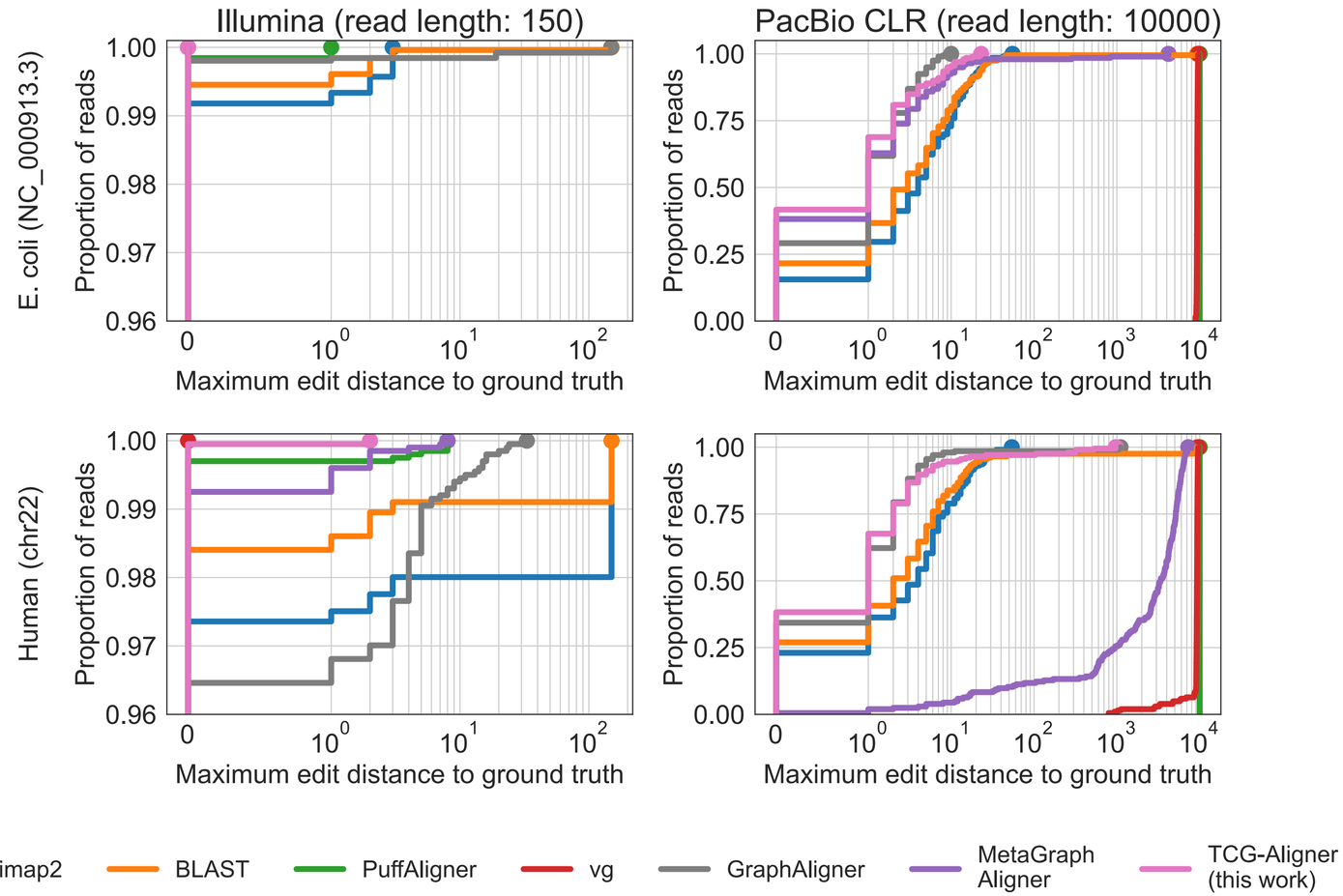
Uses **seed-chain-extend** approach:

1. Find seeds of size k or less
2. Chaining (inspired by Minimap2 [Li, 2018])
   - DP table for pairs (seed, coordinate) sorted by coordinates
   - Dynamic Programming + Backtracking
3. Extension algorithm
   - generalization of Needleman-Wunsch algorithm
   - similar to the MetaGraph aligner [Karasikov et al., 2020]

Alignment accuracy

Evaluation approach:

1. Index references
2. Simulate reads from the same references
3. Align reads and measure distance between the alignments and their generating references



Alignment accuracy for **Counting DBG** and **state-of-the-art aligners** on simulated Illumina- and PacBio-type reads (E. coli NC 000913.3 and human chr22). The edit distance is measured between the alignment (the returned path in the graph) and the ground truth sequence. In the top left subplot, the curves of vg- and TCG-Aligner are superimposed.

Lossless indexing of RefSeq

RefSeq (33M accessions, 1.7 Tbp, 483 GB)

	BLAST	MegaBLAST	This work
Index size	**437 GB (2.05 bits/bp)**	2,358 GB (11.07 bits/bp)	509 GB (2.39 bits/bp)
Align 1 read (*)	353 sec, 417 GB	12.5 sec, **0.090 GB**	**0.66 sec**, 500 GB
Align 1,000 reads	1,857 sec, 428 GB	1,542 sec, **22.0 GB**	**575 sec**, 513 GB

The alignment speed was measured on reads taken from a metagenomic sequencing sample SRR10002688_1.

(*) For aligning single reads, the experiment is independently performed for the 100 first reads and the average time and RAM usage are presented.

Lossless indexing of RefSeq

RefSeq (33M accessions, 1.7 Tbp, 483 GB)

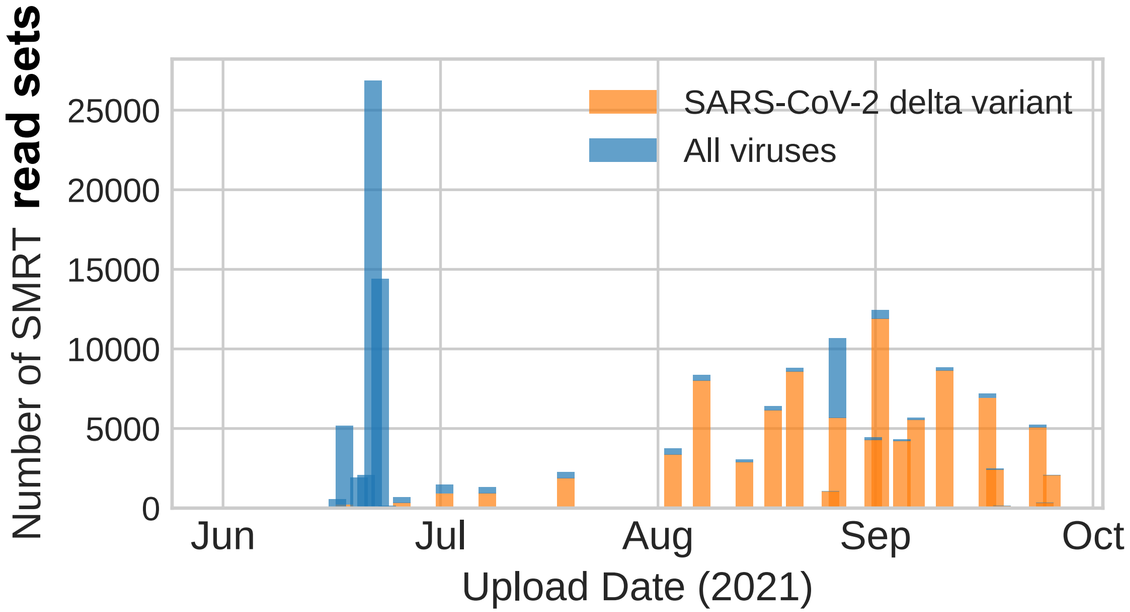
	BLAST	MegaBLAST	This work
Index size	**437 GB (2.05 bits/bp)**	2,358 GB (11.07 bits/bp)	509 GB (2.39 bits/bp)
Align 1 read (*)	353 sec, 417 GB	12.5 sec, **0.090 GB**	**0.66 sec**, 500 GB
Align 1,000 reads	1,857 sec, 428 GB	1,542 sec, **22.0 GB**	**575 sec**, 513 GB

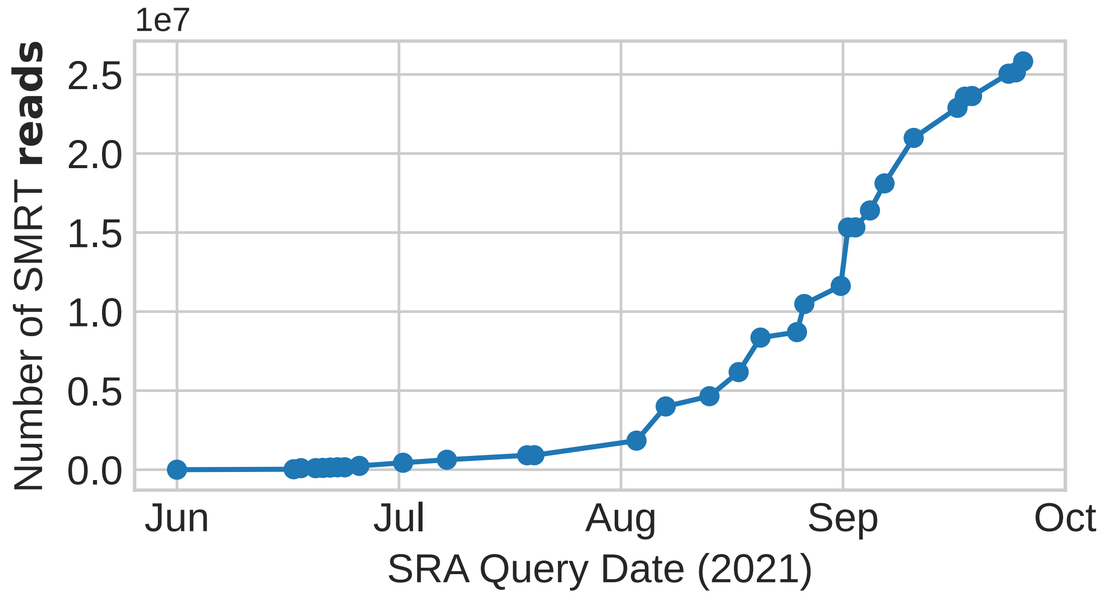
The alignment speed was measured on reads taken from a metagenomic sequencing sample SRR10002688_1.

(*) For aligning single reads, the experiment is independently performed for the 100 first reads and the average time and RAM usage are presented.

Query Delta variant of SARS-CoV-2

1. Constructed a joint index of all 152,884 viral PacBio SMRT read sets from SRA
2. Assembled a list of 9 defining mutations of the SARS-CoV-2 21A (Delta) variant spike protein
3. Retrieve all the occurrences of these specific mutations within the reads (took under 4 min)





Coordinates allow retrieving not only the number of read sets **(left)** but also single reads **(right)**

Conclusion

Big Data National Research Programme

Counting de Bruijn Graphs naturally generalize *Annotated de Bruijn graphs*

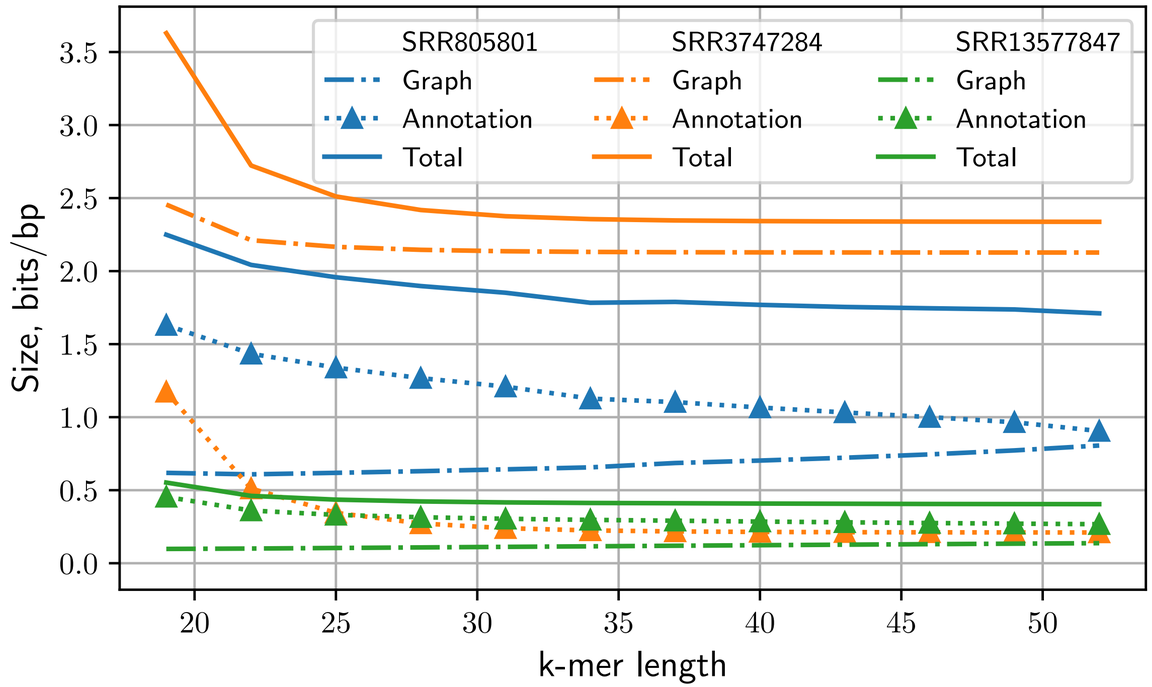
1. Allow **efficiently encoding non-binary annotations**, e.g.:
    - k-mer abundances
    - k-mer coordinates
2. **Scales to very large graphs**
    - constructs in linear time and constant memory
3. Future directions, possible applications:
    - indexing and **querying k-mer abundances** (gene expression levels)
    - can be used as a **backend for aligners** (such as BLAST)
    - alignment with different error models (using k-mer abundances)

The work will soon to be published in journal *Genome Research*

Backup slides

Lossless indexing of k-mer coordinates

Dependence on k-mer length



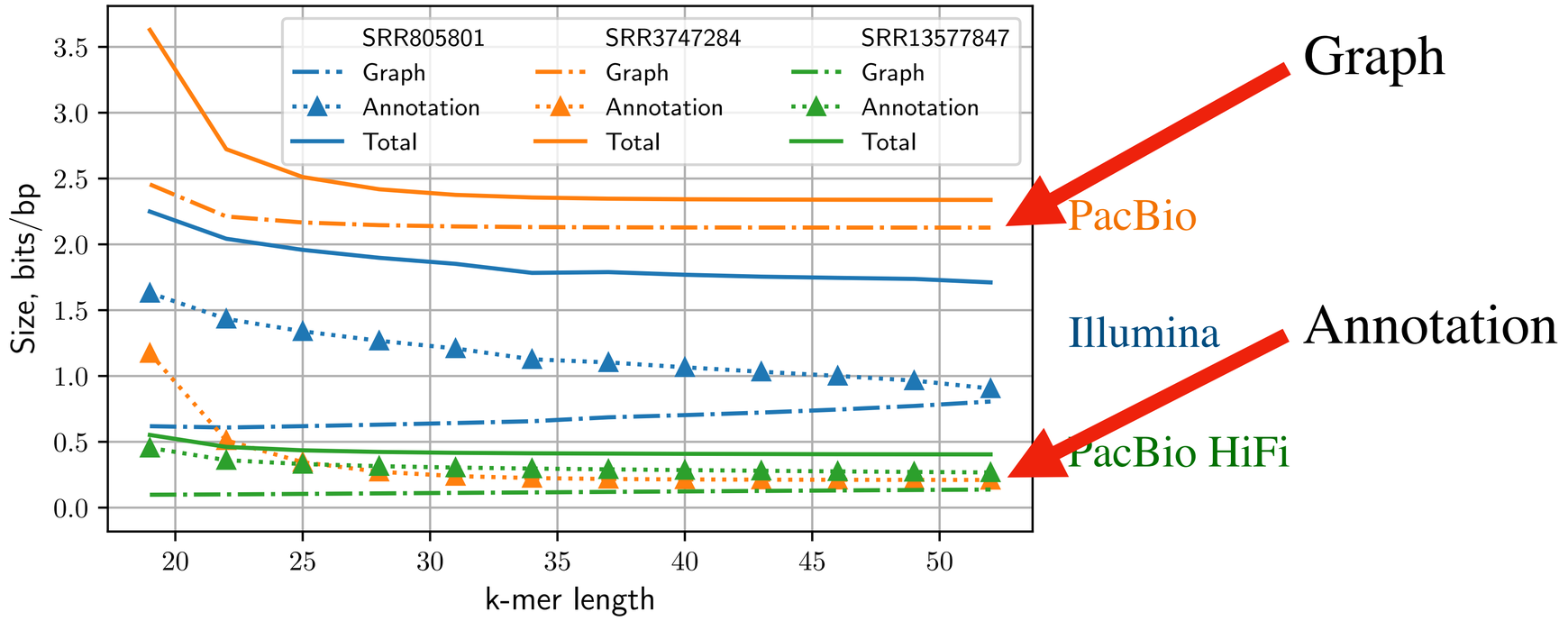
PacBio

Illumina

PacBio HiFi

Lossless indexing of k-mer coordinates

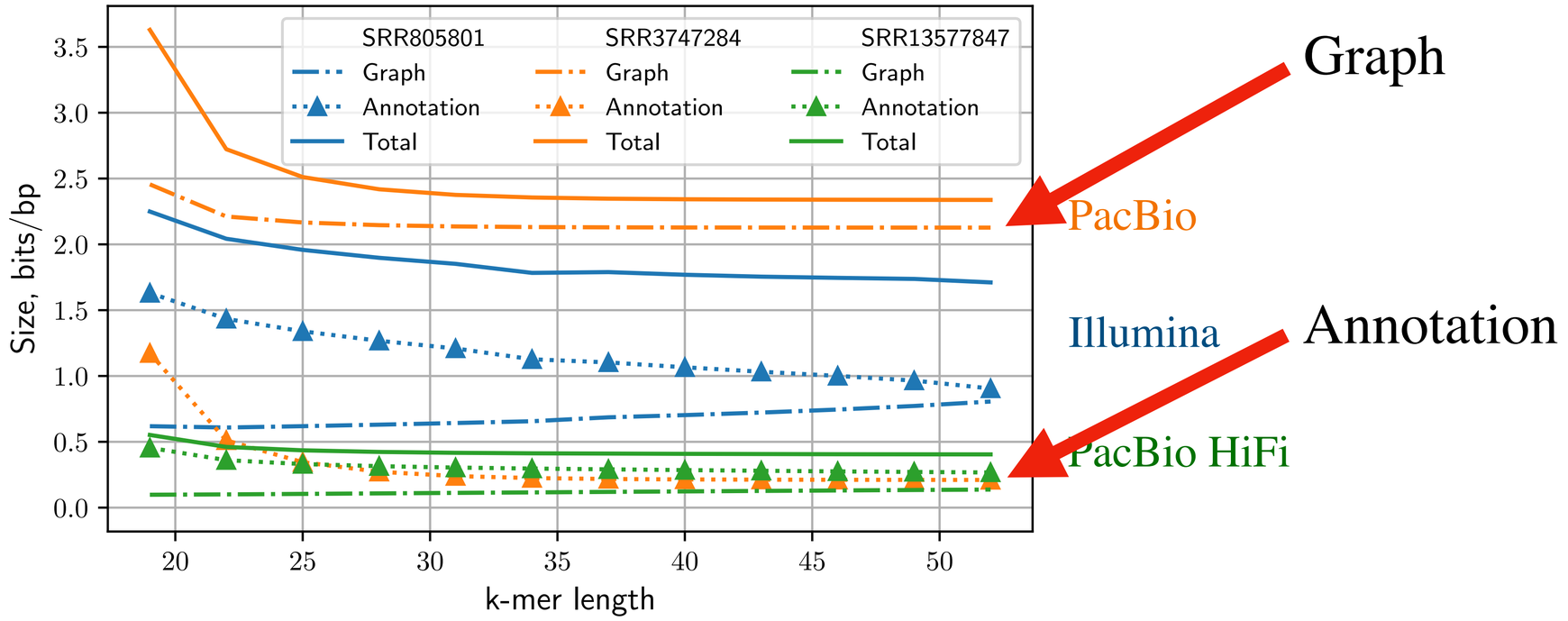
Dependence on k-mer length



1. For low-error reads (HiFi), annotation is tiny compared to graph

Lossless indexing of k-mer coordinates

Dependence on k-mer length



1. For low-error reads (HiFi), annotation is tiny compared to graph
2. Index size does not increase for larger values of k even for short Illumina reads

Construction time

	Construction time			Max. RAM		
Method	**Binary**	**Smooth counts**	**Raw counts**	**Binary**	**Smooth counts**	**Raw counts**
k-mer spectrum estimation with ntCard†	2.0 h	-	-	19 GB	-	-
k-mer counting with Squeakr†	35.4 h	-	-	333 GB	-	-
k-mer counting with Squeakr (single thread)†*	19.3 h	-	-	483 GB	-	-
Graph and annotation construction with Mantis*	9.6 h	-	-	42.7 GB	-	-
Annotation transform to MST	0.4 h	-	-	26.6 GB	-	-
Mantis-MST (all steps)	**29.3 h**	-	-	**483 GB**	-	-
k-mer counting with KMC† (with flag -m2)	3.2 h	-	-	41 GB	-	-
Assembling contigs from k-mers with MetaGraph†	0.7 h	-	-	51 GB	-	-
Graph construction and annotation with MetaGraph	1.7 h	-	-	52 GB	-	-
Annotation transform to RowDiff-MultiBRWT	1.3 h	-	-	58 GB	-	-
RowDiff (all steps)	**6.9 h**	-	-	**58 GB**	-	-
Assembling unitigs with BCALM† (with -max-memory 1000)	40.6 h	40.6 h	-	152 GB	152 GB	-
REINDEER (indexing)*	8.7 h	10.6 h	-	68.8 GB	59.4 GB	-
REINDEER (all steps)	**49.3 h**	**51.2 h**	-	**152 GB**	**152 GB**	-
k-mer counting with KMC† (with flag -m2)	3.2 h	3.2 h	3.2 h	41 GB	41 GB	41 GB
Assembling contigs from k-mers with MetaGraph†	0.7 h	0.9 h	0.8 h	51 GB	70 GB	70 GB
Graph construction and annotation with MetaGraph	1.6 h	2.2 h	2.3 h	52 GB	52 GB	52 GB
Annotation transformation (this work)	1.2 h	1.2 h	1.4 h	59 GB	88 GB	88 GB
This work (all steps)	**6.7 h**	**7.5 h**	**7.7 h**	**59 GB**	**88 GB**	**88 GB**

Alignment performance

Supplemental Table 2. Percentage of simulated reads mapping exactly to their respective ground-truth sequences (top) and total query time in seconds (bottom). For each reference, 2000 Illumina reads were simulated using ART and 200 PacBio CLR reads were simulated using pbsim, respectively. PuffAligner was unable to align the PacBio-type reads.

Reference	Minimap2	BLAST	PuffAligner	vg	GraphAligner	MetaGraph-Aligner	This work
Illumina: `E. coli`	99.18%	99.45%	99.84%	**100.00%**	99.8%	**100.00%**	**100.00%**
Illumina: `chr22`	97.36%	98.4%	99.70%	**100.00%**	96.46%	99.25%	99.95%
PacBio: `E. coli`	15.58%	21.61%	N/A	0.00%	29.15%	38.69%	**41.71%**
PacBio: `chr22`	23.04%	26.96%	N/A	0.00%	34.31%	0.49%	**40.20%**
Illumina: `E. coli`	0.13 s	0.39 s	**0.03 s**	1.16 s	3.22 s	1.13 s	1.47 s
Illumina: `chr22`	**0.47 s**	26.69 s	0.96 s	5.67 s	42.93 s	6.67 s	3.04 s
PacBio: `E. coli`	**1.11 s**	1.51 s	4.31 s	443.23 s	7.39 s	40.20 s	22.19 s
PacBio: `chr22`	**2.43 s**	34.80 s	4.91 s	1059.05 s	48.99 s	828.91 s	37.76 s

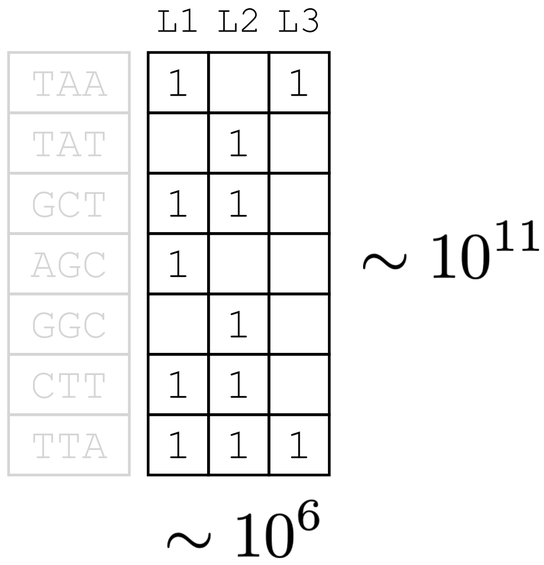
Data

Nine defining Delta gene variants of the SARS-CoV-2 spike protein

```
>OK091006.1:21536-25357:27-85
ACTAGTCTCTAGTCAGTGTGTTAATCTTAGAACCAGAACTCAATTACCCCCTGCATACA
>OK091006.1:21536-25357:444-502
CAACAAAAGTTGGATGGAAAGTGGAGTTTATTCTAGTGCGAATAATTGCACTT
>OK091006.1:21536-25357:1326-1384
TTCTAAGGTTGGTGGTAATTATAATTACCGGTATAGATTGTTTAGGAAGTCTAATCTCA
>OK091006.1:21536-25357:1404-1462
TTCAACTGAAATCTATCAGGCCGGTAGCAAACCTTGTAATGGTGTTGAAGGTTTTAATT
>OK091006.1:21536-25357:1812-1870
TTCTAACCAGGTTGCTGTTCTTTATCAGGGTGTTAACTGCACAGAAGTCCCTGTTGCTA
>OK091006.1:21536-25357:2013-2071
CGCTAGTTATCAGACTCAGACTAATTCTCGTCGGCGGGCACGTAGTGTAGCTAGTCAAT
>OK091006.1:21536-25357:2820-2878
CACAGCAAGTGCACTTGGAAAACTTCAAAATGTGGTCAACCAAAATGCACAAGCTTTAA
```

Background

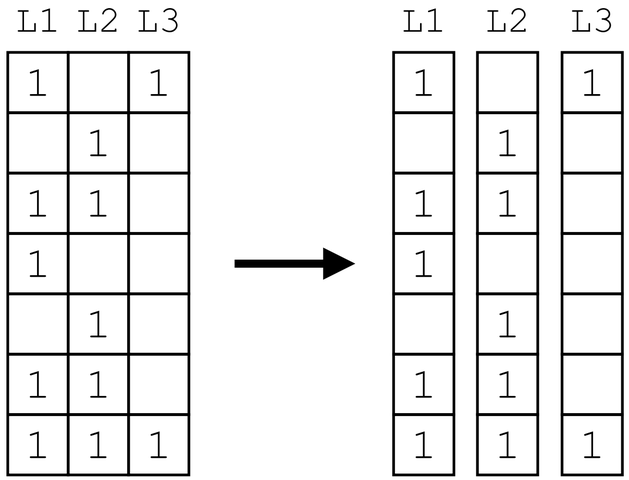
Graph annotation representations

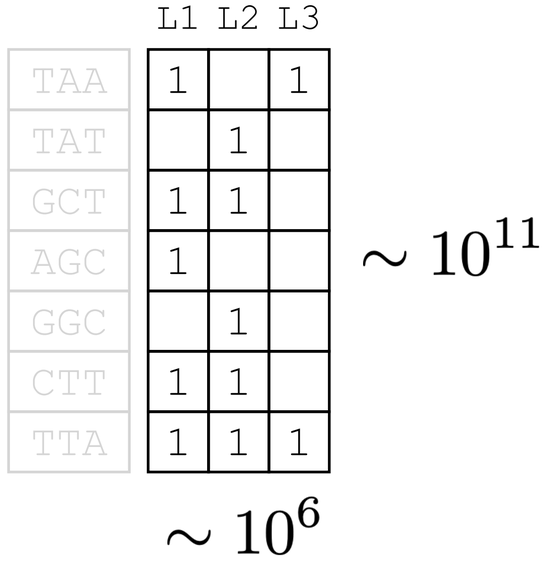


Background

Graph annotation representations

1. Column-major sparse representation

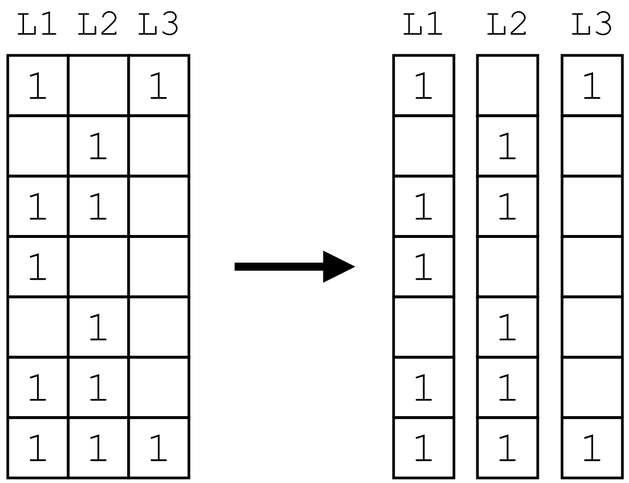


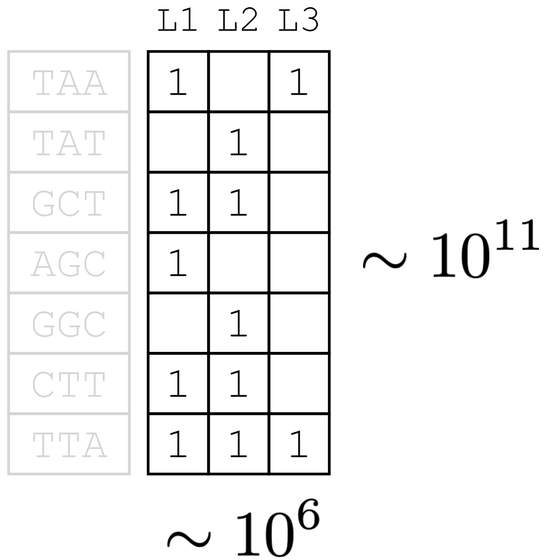


Background

Graph annotation representations

1. Column-major sparse representation



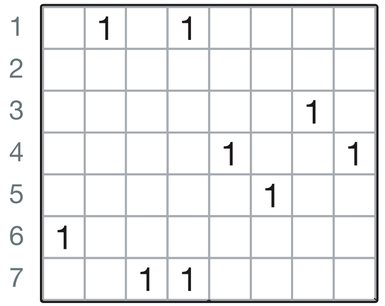


Columns are stored independently as compressed bitmaps
(e.g. sd_vector [Okanohara et al., 2007])

Background

Graph annotation representations

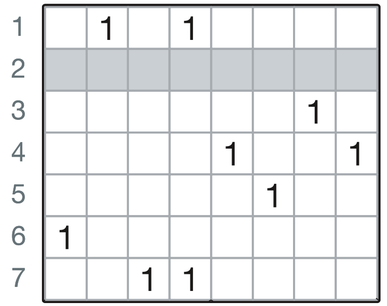
1. Column-major sparse representation
2. Multi-BRWT **[Karasikov et al., 2019]**



Background

Graph annotation representations

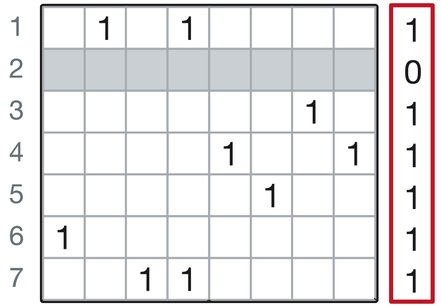
1. Column-major sparse representation
2. Multi-BRWT **[Karasikov et al., 2019]**



Background

Graph annotation representations

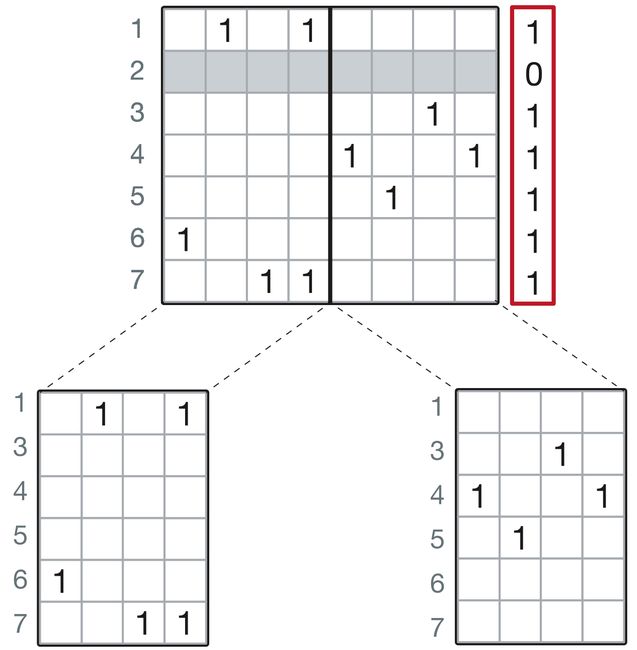
1. Column-major sparse representation
2. Multi-BRWT **[Karasikov et al., 2019]**



Background

Graph annotation representations

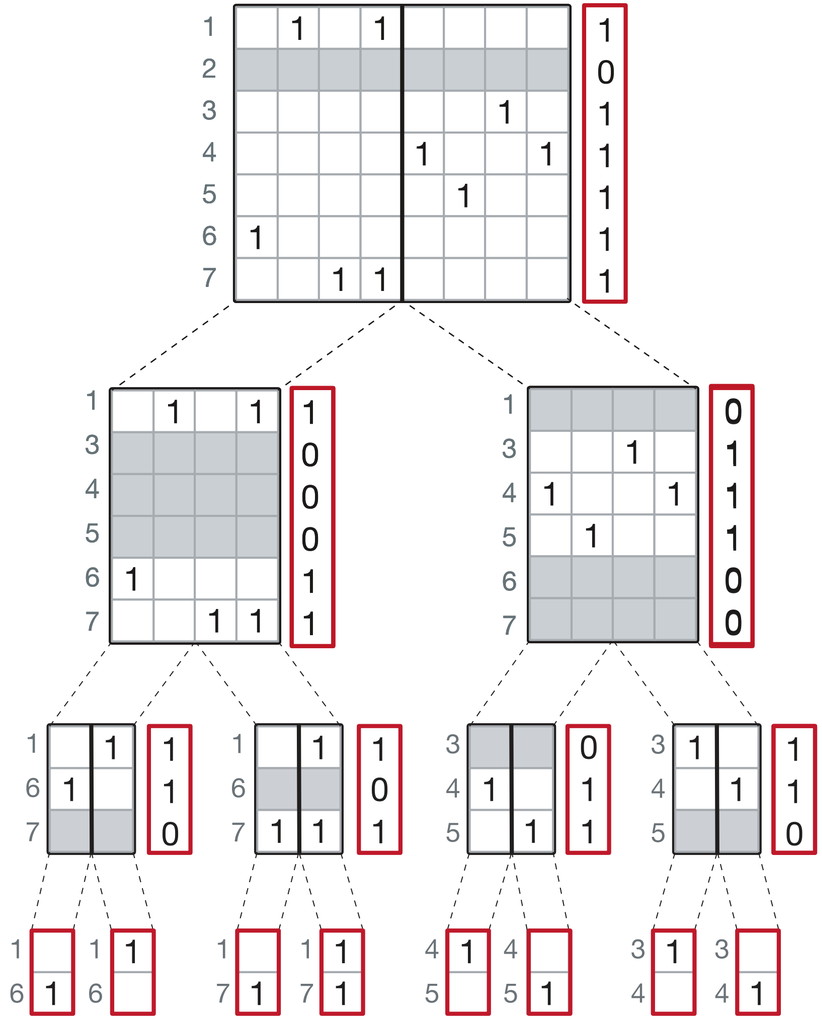
1. Column-major sparse representation
2. Multi-BRWT **[Karasikov et al., 2019]**



Background

Graph annotation representations

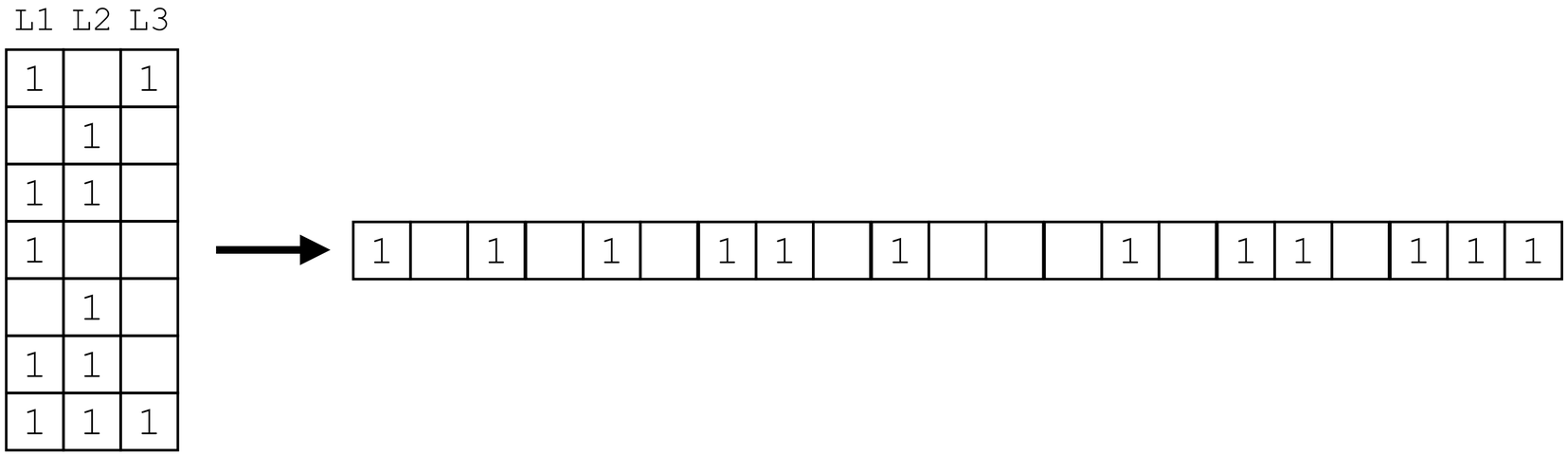
1. Column-major sparse representation
2. Multi-BRWT **[Karasikov et al., 2019]**



Background

Graph annotation representations

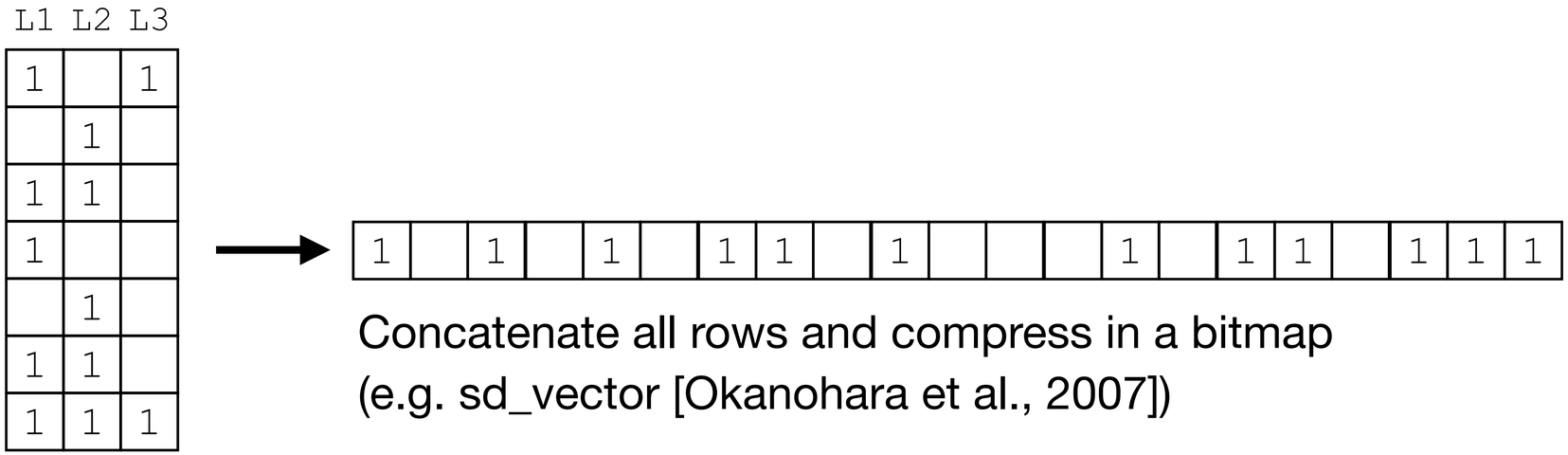
1. Column-major sparse representation
2. Multi-BRWT **[Karasikov et al., 2019]**
3. RowFlat (employed in VARI **[Muggli et al., 2017]**)



Background

Graph annotation representations

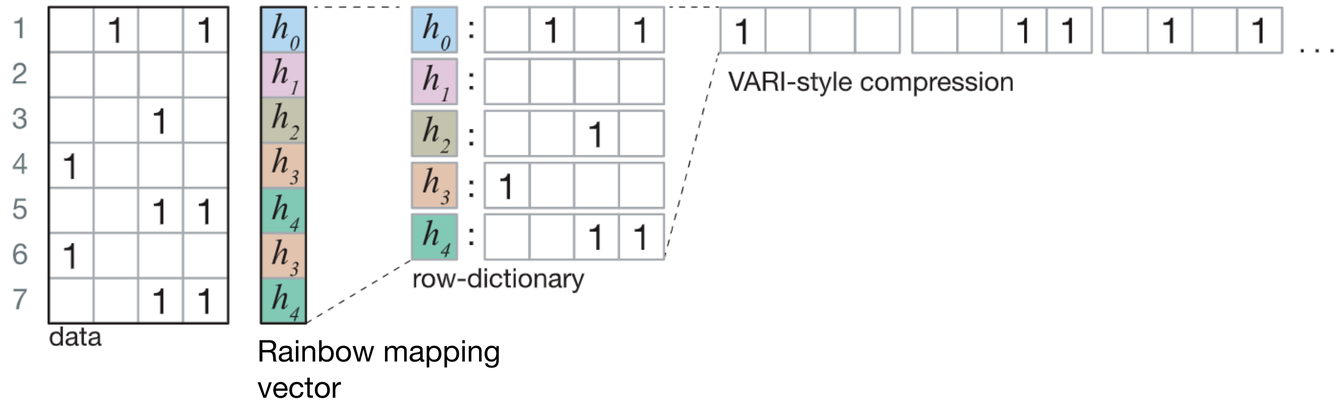
1. Column-major sparse representation
2. Multi-BRWT **[Karasikov et al., 2019]**
3. RowFlat (employed in VARI **[Muggli et al., 2017]**)



Background

Graph annotation representations

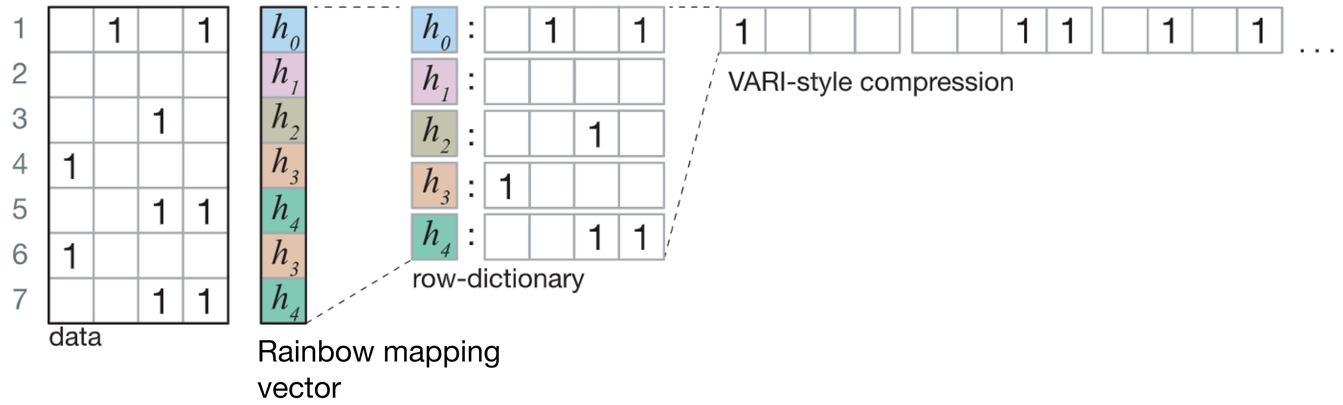
1. Column-major sparse representation
2. Multi-BRWT **[Karasikov et al., 2019]**
3. RowFlat (employed in VARI **[Muggli et al., 2017]**)
4. Rainbowfish **[Almodaresi et al., 2017]**



Background

Graph annotation representations

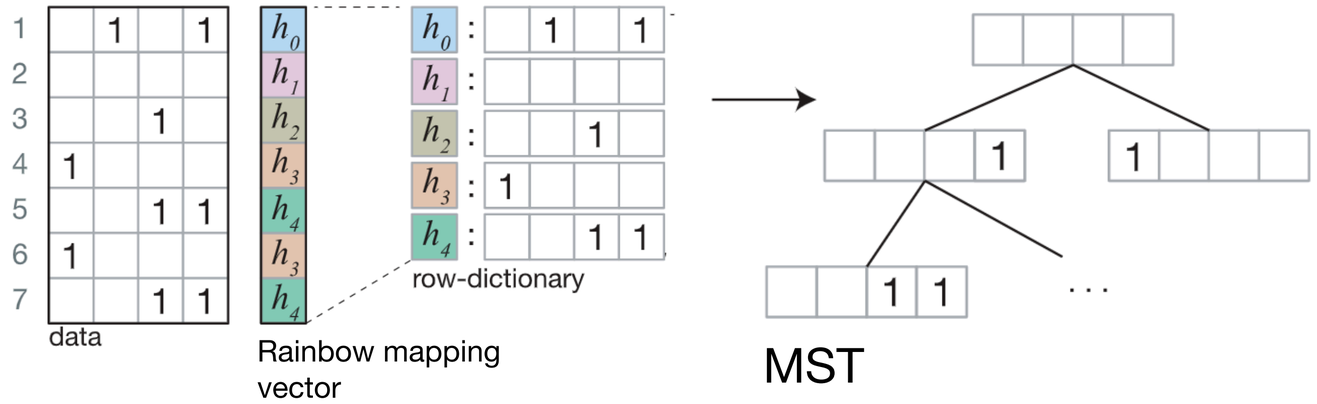
1. Column-major sparse representation
2. Multi-BRWT **[Karasikov et al., 2019]**
3. RowFlat (employed in VARI **[Muggli et al., 2017]**)
4. Rainbowfish **[Almodaresi et al., 2017]**
5. Mantis-MST **[Almodaresi et al., 2019]**



Background

Graph annotation representations

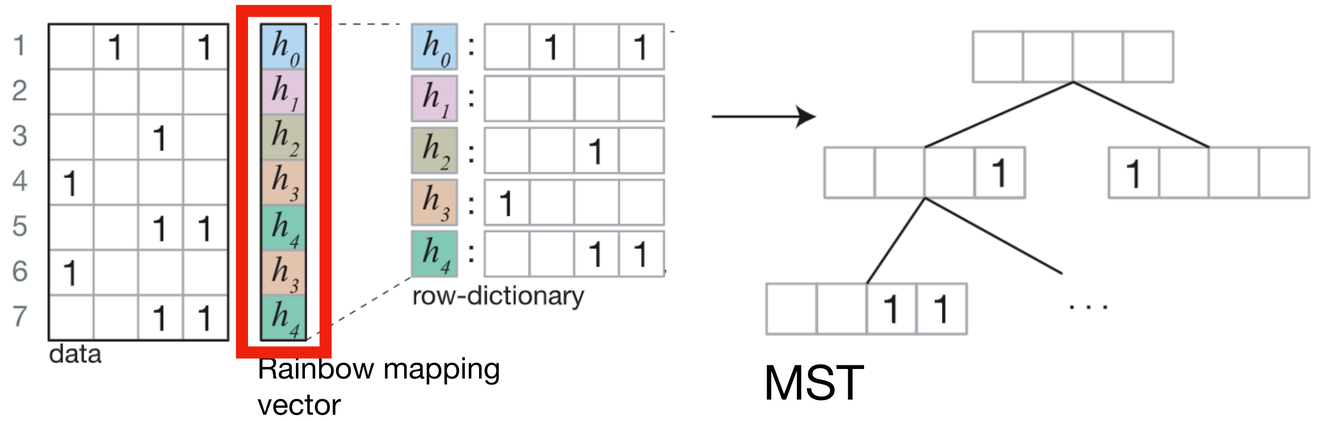
1. Column-major sparse representation
2. Multi-BRWT **[Karasikov et al., 2019]**
3. RowFlat (employed in VARI **[Muggli et al., 2017]**)
4. Rainbowfish **[Almodaresi et al., 2017]**
5. Mantis-MST **[Almodaresi et al., 2019]**



Background

Graph annotation representations

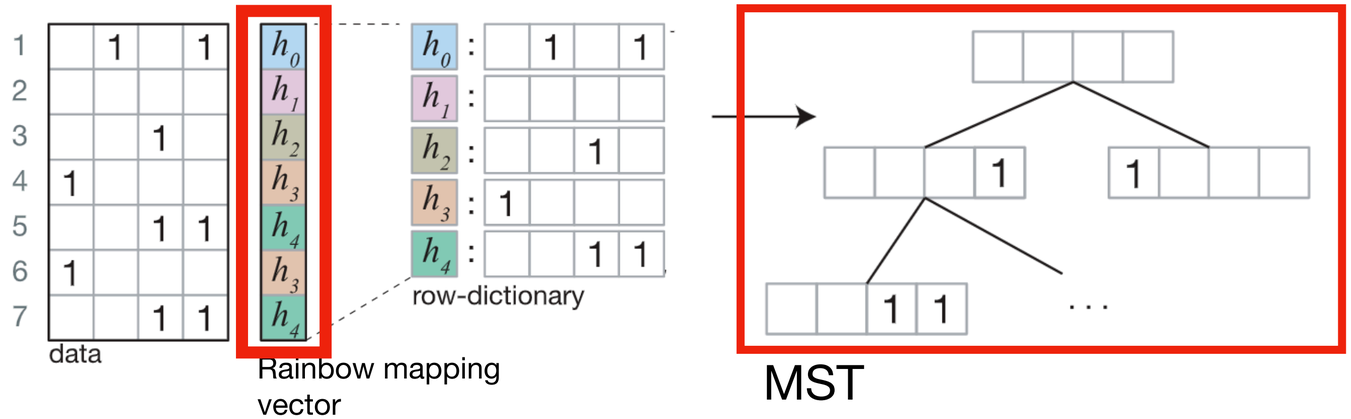
1. Column-major sparse representation
2. Multi-BRWT **[Karasikov et al., 2019]**
3. RowFlat (employed in VARI **[Muggli et al., 2017]**)
4. Rainbowfish **[Almodaresi et al., 2017]**
5. Mantis-MST **[Almodaresi et al., 2019]**



Background

Graph annotation representations

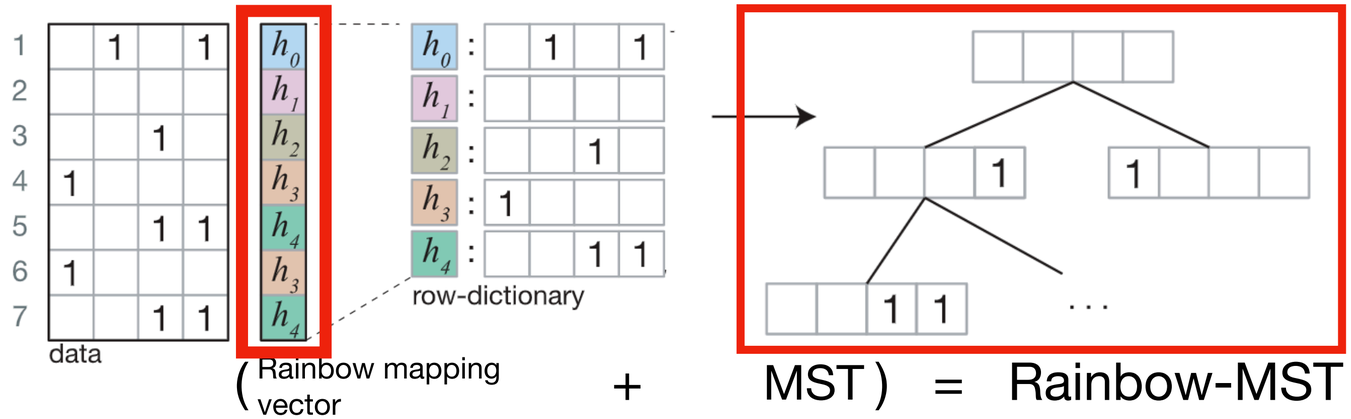
1. Column-major sparse representation
2. Multi-BRWT **[Karasikov et al., 2019]**
3. RowFlat (employed in VARI **[Muggli et al., 2017]**)
4. Rainbowfish **[Almodaresi et al., 2017]**
5. Mantis-MST **[Almodaresi et al., 2019]**



Background

Graph annotation representations

1. Column-major sparse representation
2. Multi-BRWT **[Karasikov et al., 2019]**
3. RowFlat (employed in VARI **[Muggli et al., 2017]**)
4. Rainbowfish **[Almodaresi et al., 2017]**
5. Mantis-MST **[Almodaresi et al., 2019]**



Background

Graph annotation representations

1. Column-major sparse representation
2. Multi-BRWT **[Karasikov et al., 2019]**
3. RowFlat (employed in VARI **[Muggli et al., 2017]**)
4. Rainbowfish **[Almodaresi et al., 2017]**
5. Mantis-MST **[Almodaresi et al., 2019]**

Background

Graph annotation representations

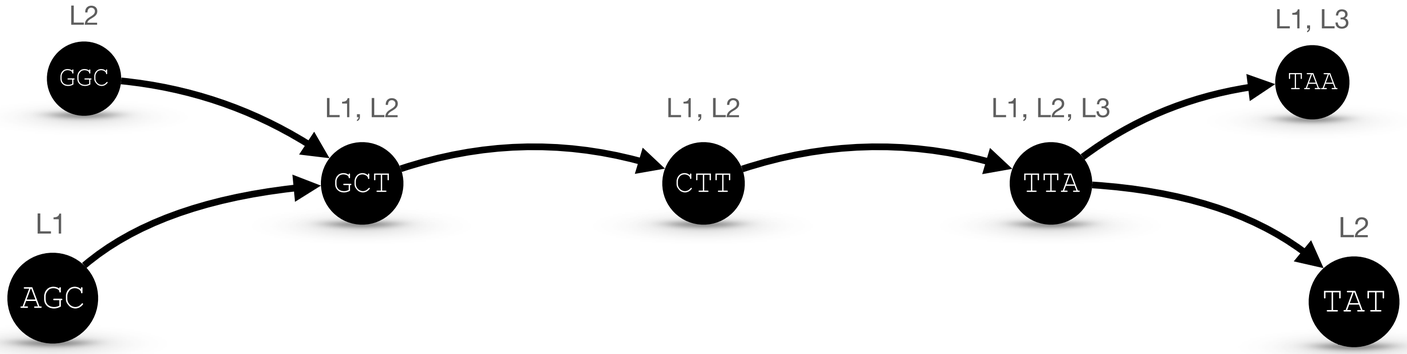
1. Column-major sparse representation
2. Multi-BRWT **[Karasikov et al., 2019]**
3. RowFlat (employed in VARI **[Muggli et al., 2017]**)
4. Rainbowfish **[Almodaresi et al., 2017]**
5. Mantis-MST **[Almodaresi et al., 2019]**
6. RowDiff **[Danciu et al., 2021]**

Background

Graph annotation representations

1. Column-major sparse representation
2. Multi-BRWT **[Karasikov et al., 2019]**
3. RowFlat (employed in VARI **[Muggli et al., 2017]**)
4. Rainbowfish **[Almodaresi et al., 2017]**
5. Mantis-MST **[Almodaresi et al., 2019]**
6. RowDiff **[Danciu et al., 2021]**

	L1	L2	L3
TAA	1		1
TAT		1	
GCT	1	1	
AGC	1		
GGC		1	
CTT	1	1	
TTA	1	1	1



Background

Graph annotation representations

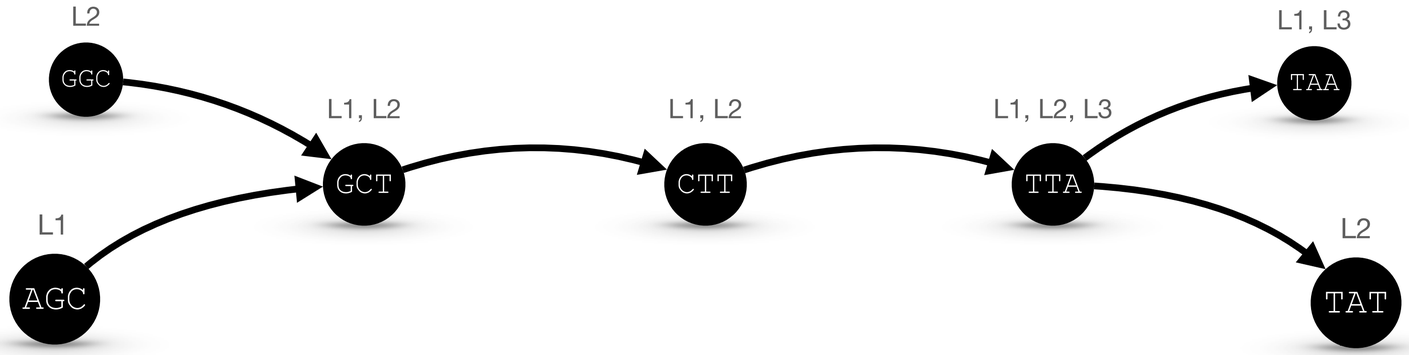
1. Column-major sparse representation
2. Multi-BRWT **[Karasikov et al., 2019]**
3. RowFlat (employed in VARI **[Muggli et al., 2017]**)
4. Rainbowfish **[Almodaresi et al., 2017]**
5. Mantis-MST **[Almodaresi et al., 2019]**
6. RowDiff **[Danciu et al., 2021]**

Store only **diffs**:

$$L^{\delta}(v) := L(v) \oplus L(v_{\text{succ}})$$

($\oplus$ is XOR)

	L1	L2	L3
TAA	1		1
TAT		1	
GCT	1	1	
AGC	1		
GGC		1	
CTT	1	1	
TTA	1	1	1



Background

Graph annotation representations

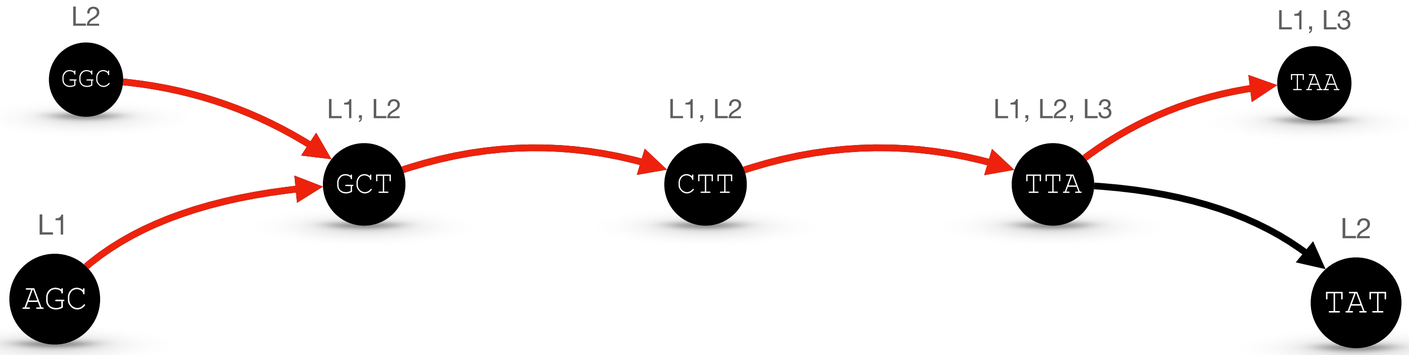
1. Column-major sparse representation
2. Multi-BRWT **[Karasikov et al., 2019]**
3. RowFlat (employed in VARI **[Muggli et al., 2017]**)
4. Rainbowfish **[Almodaresi et al., 2017]**
5. Mantis-MST **[Almodaresi et al., 2019]**
6. RowDiff **[Danciu et al., 2021]**

Store only **diffs**:

$$L^{\delta}(v) := L(v) \oplus L(v_{\text{succ}})$$

($\oplus$ is XOR)

	L1	L2	L3
TAA	1		1
TAT		1	
GCT	1	1	
AGC	1		
GGC		1	
CTT	1	1	
TTA	1	1	1



Background

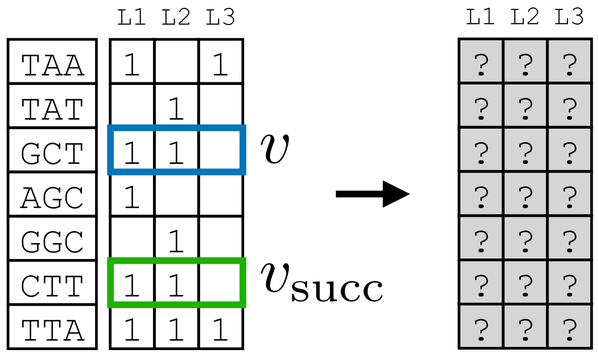
Graph annotation representations

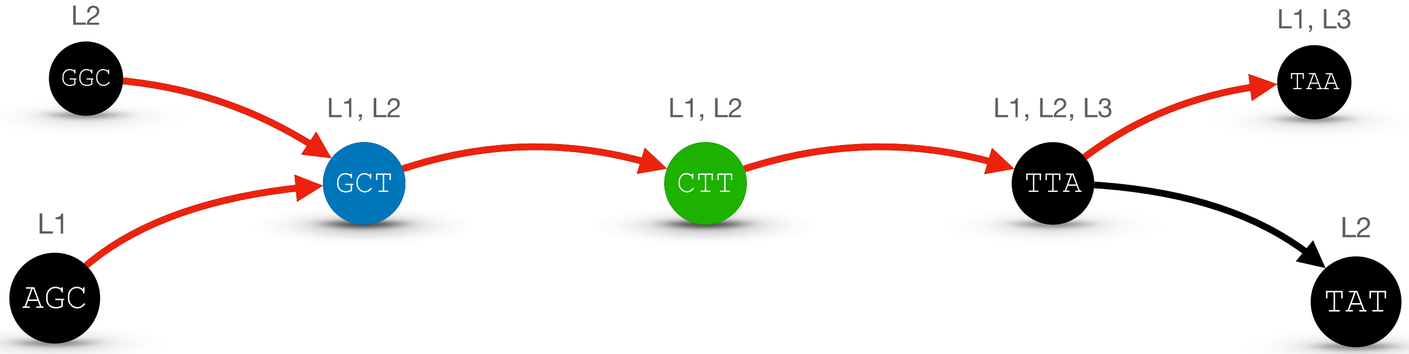
1. Column-major sparse representation
2. Multi-BRWT **[Karasikov et al., 2019]**
3. RowFlat (employed in VARI **[Muggli et al., 2017]**)
4. Rainbowfish **[Almodaresi et al., 2017]**
5. Mantis-MST **[Almodaresi et al., 2019]**
6. RowDiff **[Danciu et al., 2021]**

Store only **diffs**:

$$L^{\delta}(v) := L(v) \oplus L(v_{\text{succ}})$$

($\oplus$ is XOR)





Background

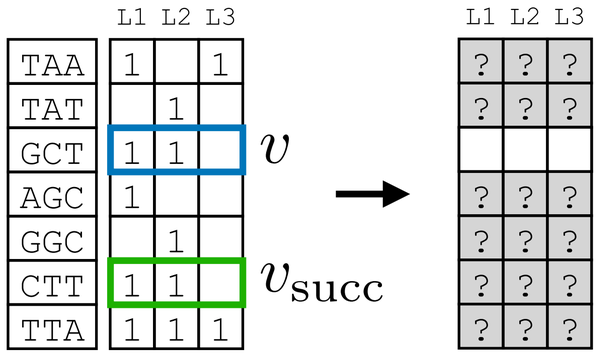
Graph annotation representations

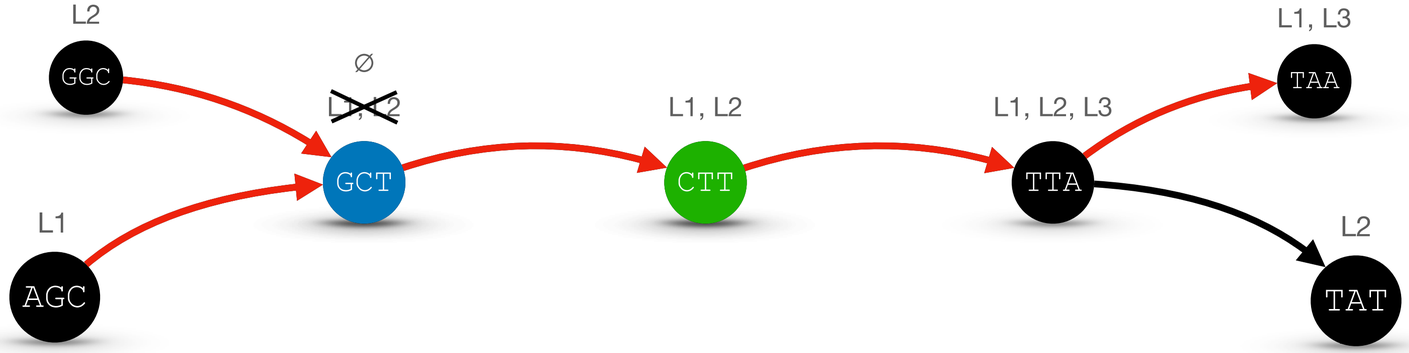
1. Column-major sparse representation
2. Multi-BRWT **[Karasikov et al., 2019]**
3. RowFlat (employed in VARI **[Muggli et al., 2017]**)
4. Rainbowfish **[Almodaresi et al., 2017]**
5. Mantis-MST **[Almodaresi et al., 2019]**
6. RowDiff **[Danciu et al., 2021]**

Store only **diffs**:

$$L^{\delta}(v) := L(v) \oplus L(v_{\text{succ}})$$

($\oplus$ is XOR)





Background

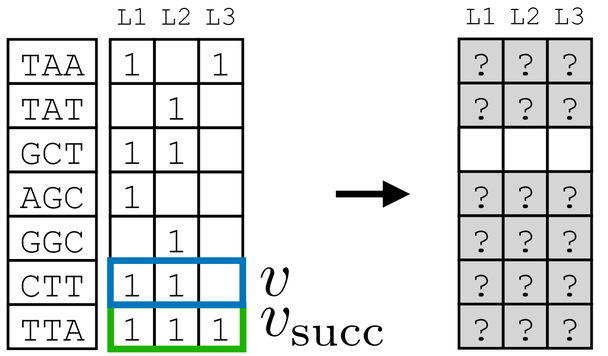
Graph annotation representations

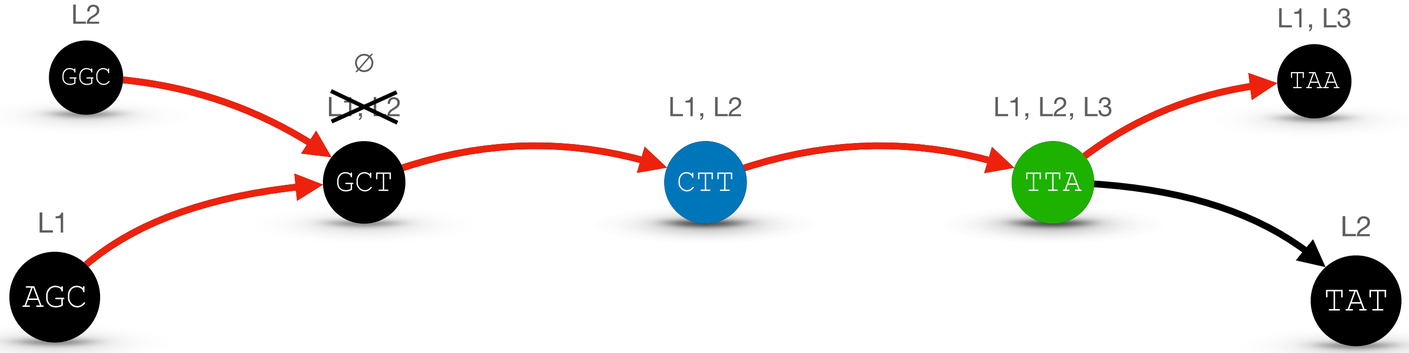
1. Column-major sparse representation
2. Multi-BRWT **[Karasikov et al., 2019]**
3. RowFlat (employed in VARI **[Muggli et al., 2017]**)
4. Rainbowfish **[Almodaresi et al., 2017]**
5. Mantis-MST **[Almodaresi et al., 2019]**
6. RowDiff **[Danciu et al., 2021]**

Store only **diffs**:

$$L^{\delta}(v) := L(v) \oplus L(v_{\text{succ}})$$

($\oplus$ is XOR)





Background

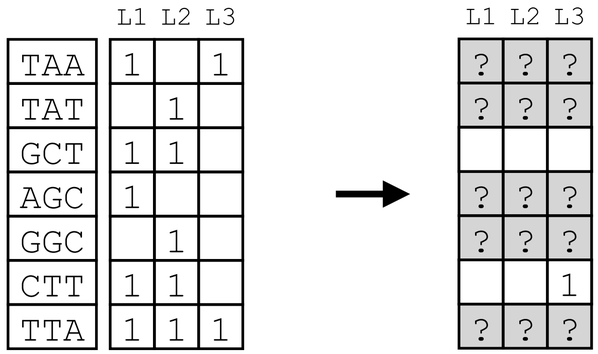
Graph annotation representations

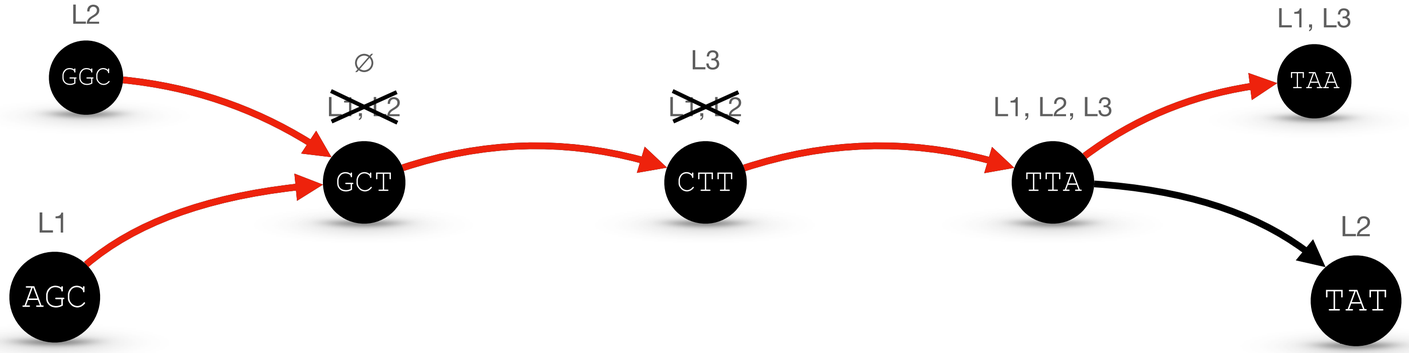
1. Column-major sparse representation
2. Multi-BRWT **[Karasikov et al., 2019]**
3. RowFlat (employed in VARI **[Muggli et al., 2017]**)
4. Rainbowfish **[Almodaresi et al., 2017]**
5. Mantis-MST **[Almodaresi et al., 2019]**
6. RowDiff **[Danciu et al., 2021]**

Store only **diffs**:

$$L^{\delta}(v) := L(v) \oplus L(v_{\text{succ}})$$

($\oplus$ is XOR)





Background

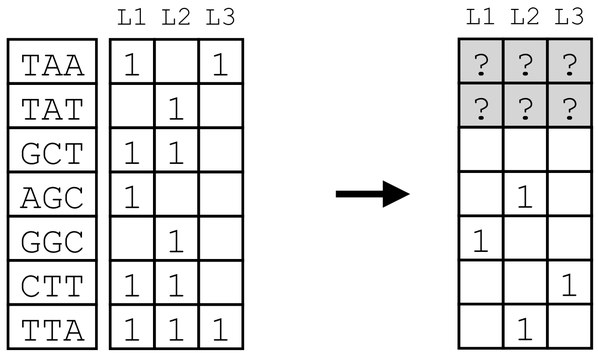
Graph annotation representations

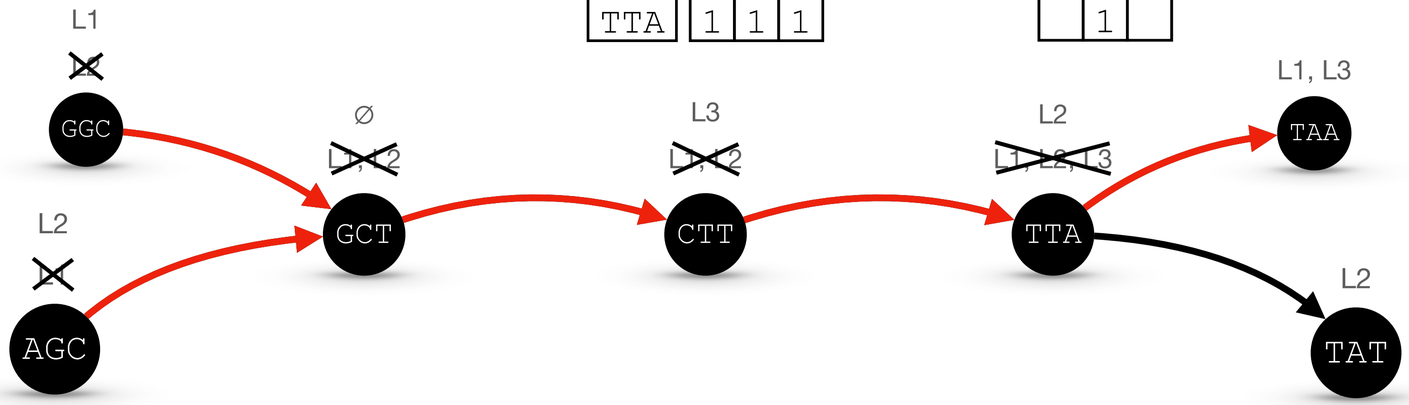
1. Column-major sparse representation
2. Multi-BRWT **[Karasikov et al., 2019]**
3. RowFlat (employed in VARI **[Muggli et al., 2017]**)
4. Rainbowfish **[Almodaresi et al., 2017]**
5. Mantis-MST **[Almodaresi et al., 2019]**
6. RowDiff **[Danciu et al., 2021]**

Store only **diffs**:

$$L^{\delta}(v) := L(v) \oplus L(v_{\text{succ}})$$

($\oplus$ is XOR)





Background

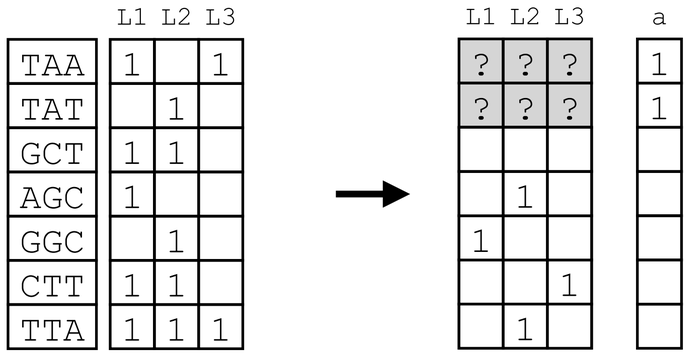
Graph annotation representations

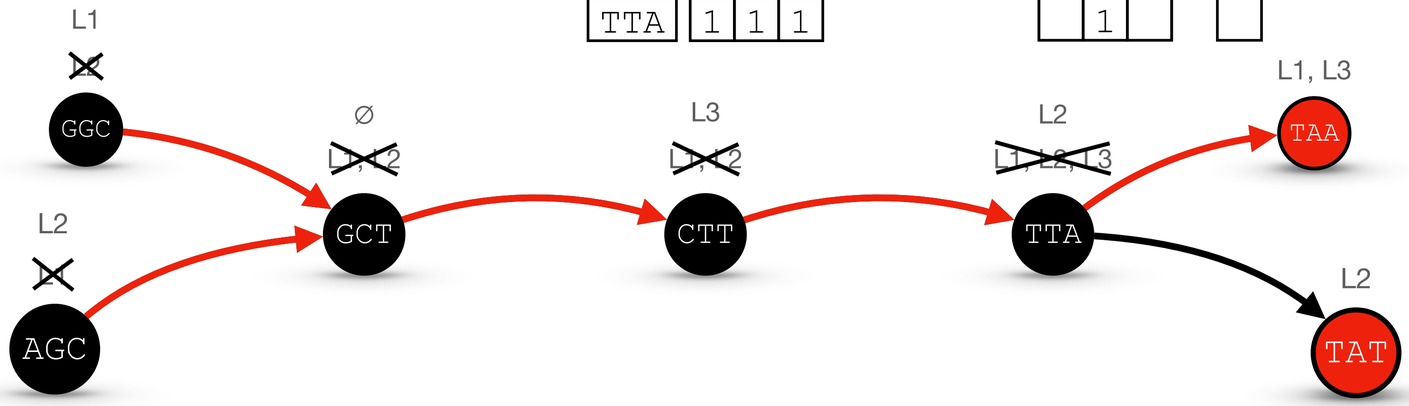
1. Column-major sparse representation
2. Multi-BRWT **[Karasikov et al., 2019]**
3. RowFlat (employed in VARI **[Muggli et al., 2017]**)
4. Rainbowfish **[Almodaresi et al., 2017]**
5. Mantis-MST **[Almodaresi et al., 2019]**
6. RowDiff **[Danciu et al., 2021]**

Store only **diffs**:

$$L^{\delta}(v) := L(v) \oplus L(v_{\text{succ}})$$

($\oplus$ is XOR)





Background

Graph annotation representations

1. Column-major sparse representation
2. Multi-BRWT **[Karasikov et al., 2019]**
3. RowFlat (employed in VARI **[Muggli et al., 2017]**)
4. Rainbowfish **[Almodaresi et al., 2017]**
5. Mantis-MST **[Almodaresi et al., 2019]**
6. RowDiff **[Danciu et al., 2021]**

Store only **diffs**:

$$L^{\delta}(v) := L(v) \oplus L(v_{\text{succ}})$$

($\oplus$ is XOR)