

Scalable Annotated Genome Graphs for Representing Sequence Data

Mikhail Karasikov

Doctoral examination

10 July 2023 (Zurich, Switzerland)

Committee:

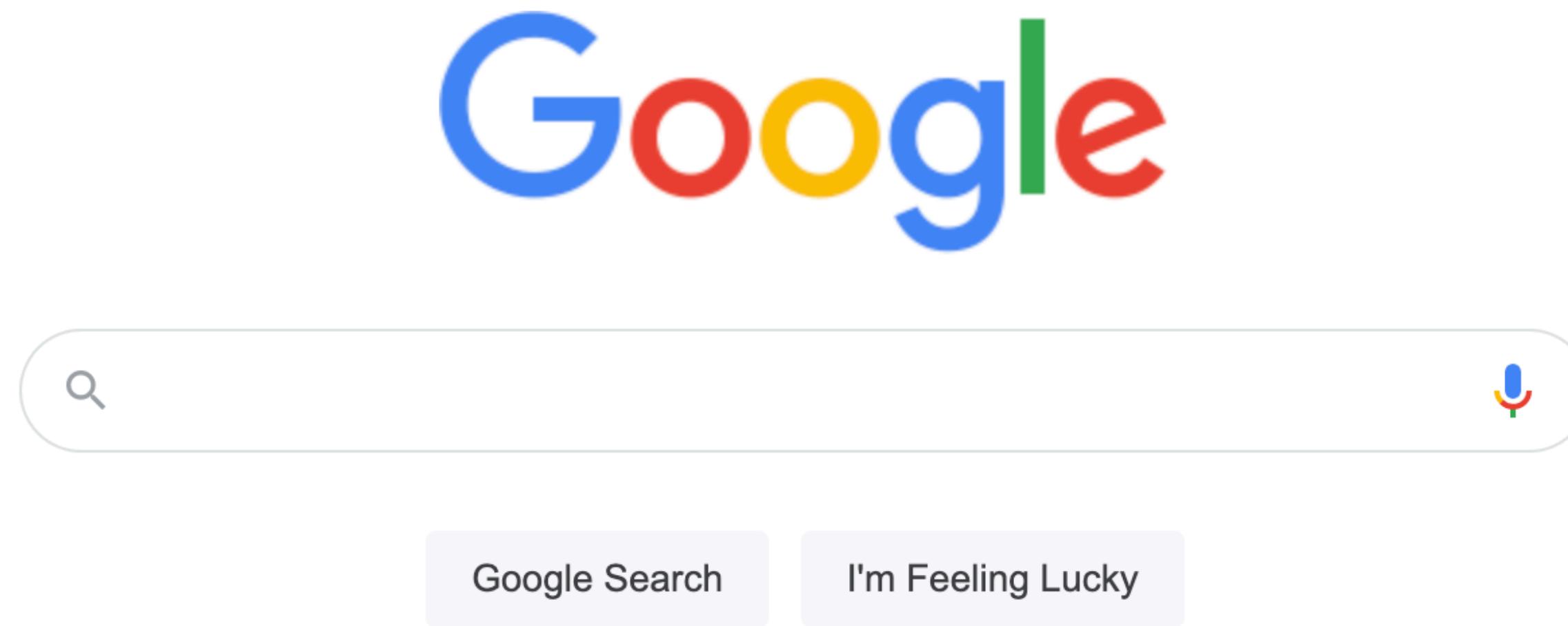
Prof. Dr. Gunnar Rätsch

Prof. Dr. Jean-Philippe Vert

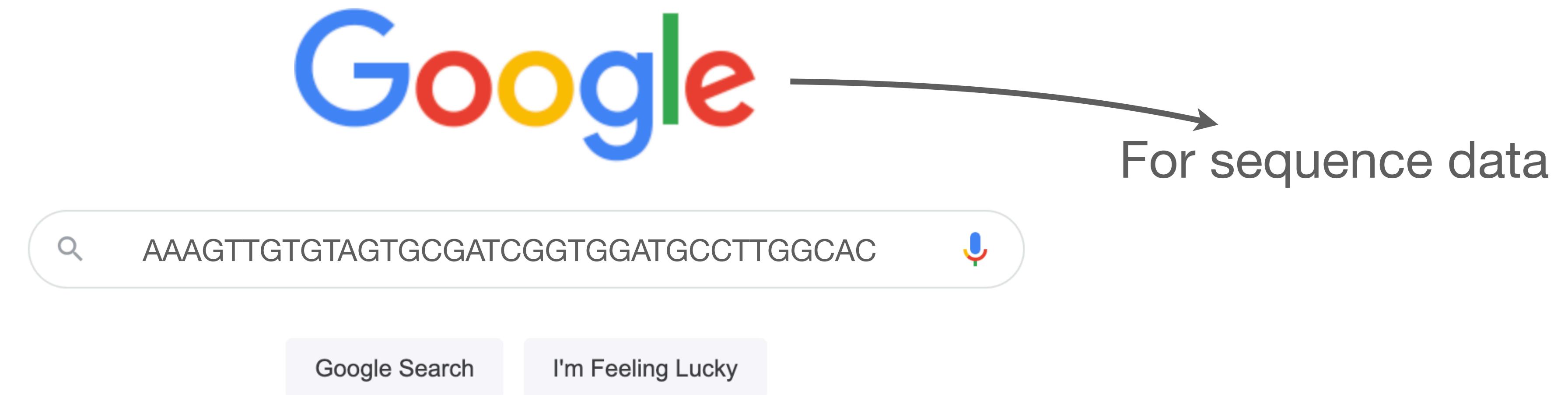
Prof. Dr. Ewan Birney

Dr. Rayan Chikhi

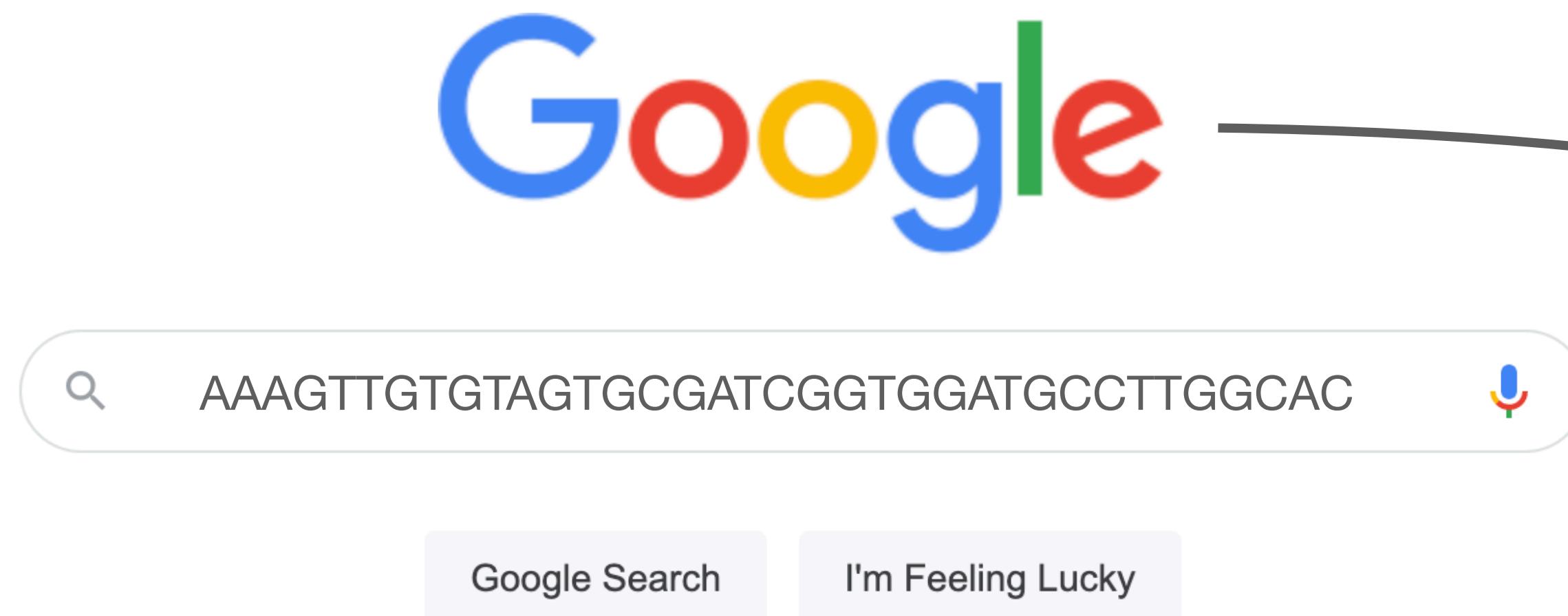
What we want



What we want



What we want



→

For sequence data



European Nucleotide Archive



UK
10K

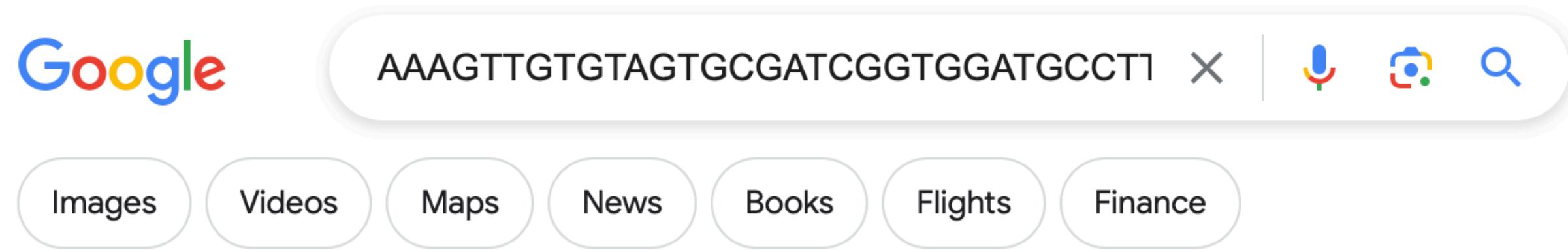


european
genna-pheno-
mome archive



...

Existing search engines do not work (yet)



Existing search engines do not work (yet)

The screenshot shows a Google search interface. The search bar contains the query "AAAGTTGTGTAGTGCGATCGGTGGATGCCTT". Below the search bar are several category buttons: Images, Videos, Maps, News, Books, Flights, and Finance. A message indicates "About 1 results (0.20 seconds)". The main result is a snippet from a page by Mikhail Karasikov, titled "Mikhail Karasikov, Harun Mustafa, Gunnar Rätsch, André ...". The snippet includes the URL <https://karasikov.com>, a PDF link, and a note about sequence data.

Google AAAGTTGTGTAGTGCGATCGGTGGATGCCTT X | ⚡ 📸 🔎

Images Videos Maps News Books Flights Finance

About 1 results (0.20 seconds)

 It looks like there aren't many great matches for your search

Try using words that might appear on the page you're looking for. For example, "cake recipes" instead of "how to make a cake."

Need help? Take a look at [other tips](#) for searching on Google.

Mikhail Karasikov
<https://karasikov.com> › IGGSy_mtg_counting_dbg PDF · · ·

[Mikhail Karasikov, Harun Mustafa, Gunnar Rätsch, André ...](#)

4 Jul 2022 — For sequence data. AAAGTTGTGTAGTGCGATCGGTGGATGCCTTGGCAC.

Page 7. What we want. 3. For sequence data

89 pages

Existing search engines do not work (yet)

Google AAAGTTGTAGTGCATCGGTGGATGCCT1 X |

Images Videos Maps News Books Flights Finance

About 1 results (0.20 seconds)

It looks like there aren't many great matches for your search

Try using words that might appear on the page you're looking for. For example, "cake recipes" instead of "how to make a cake."

Need help? Take a look at [other tips](#) for searching on Google.

Mikhail Karasikov
 <https://karasikov.com> › IGGSy_mtg_counting_dbg [PDF](#) ::

[Mikhail Karasikov, Harun Mustafa, Gunnar Rätsch, André ...](#)

4 Jul 2022 — For sequence data. AAAGTTGTAGTGCATCGGTGGATGCCTGGCAC.
Page 7. What we want. 3. For sequence data
89 pages



BLAST finds reference sequences

BLAST® » blastn suite » results for RID-9UNFW67F016

Home Recent Results Saved Strategies Help

[Edit Search](#) Save Search Search Summary ▾

How to read this report? [BLAST Help Videos](#) [Back to Traditional Results Page](#)

Job Title Nucleotide Sequence

RID [9UNFW67F016](#) Search expires on 06-30 17:59 pm [Download All](#) ▾

Program BLASTN [?](#) [Citation](#) ▾

Database nt [See details](#) ▾

Query ID lcl|Query_78553

Description None

Molecule type dna

Query Length 36

Other reports [Distance tree of results](#) [MSA viewer](#) [?](#)

Filter Results

Organism only top 20 will appear exclude
Type common name, binomial, taxid or group name
[+ Add organism](#)

Percent Identity to **E value** to **Query Coverage** to

Filter **Reset**

Descriptions Graphic Summary Alignments Taxonomy

Sequences producing significant alignments [Download](#) ▾ [Select columns](#) ▾ Show 100 ▾ [?](#)

select all 0 sequences selected

GenBank Graphics Distance tree of results MSA Viewer

	Description	Scientific Name	Max Score	Total Score	Query Cover	E value	Per. Ident	Acc. Len	Accession
<input type="checkbox"/>	Cutibacterium acnes SZ2 DNA, complete genome	Cutibacterium a...	67.6	202	100%	3e-08	100.00%	2504552	AP022845.1
<input type="checkbox"/>	Cutibacterium acnes SZ1 DNA, complete genome	Cutibacterium a...	67.6	202	100%	3e-08	100.00%	2494525	AP022844.1
<input type="checkbox"/>	Cutibacterium acnes KPA171202 chromosome, complete genome	Cutibacterium a...	67.6	202	100%	3e-08	100.00%	2560634	CP025935.1
<input type="checkbox"/>	Cutibacterium acnes DSM 1897 chromosome, complete genome	Cutibacterium a...	67.6	202	100%	3e-08	100.00%	2495002	CP025934.1

BLAST finds reference sequences

BLAST® » blastn suite » results for RID-9UNFW67F016

Home Recent Results Saved Strategies Help

◀ Edit Search Save Search Search Summary ▾

How to read this report? BLAST Help Videos Back to Traditional Results Page

Job Title Nucleotide Sequence

RID [9UNFW67F016](#) Search expires on 06-30 17:59 pm [Download All](#) ▾

Program BLASTN ? Citation ▾

Database nt [See details](#) ▾

Query ID lcl|Query_78553

Description None

Molecule type dna

Query Length 36

Other reports [Distance tree of results](#) [MSA viewer](#) ?

Filter Results

Organism only top 20 will appear exclude

Type common name, binomial, taxid or group name

+ Add organism

Percent Identity E value Query Coverage

[] to [] [] to [] [] to []

Filter Reset

Descriptions Graphic Summary Alignments Taxonomy

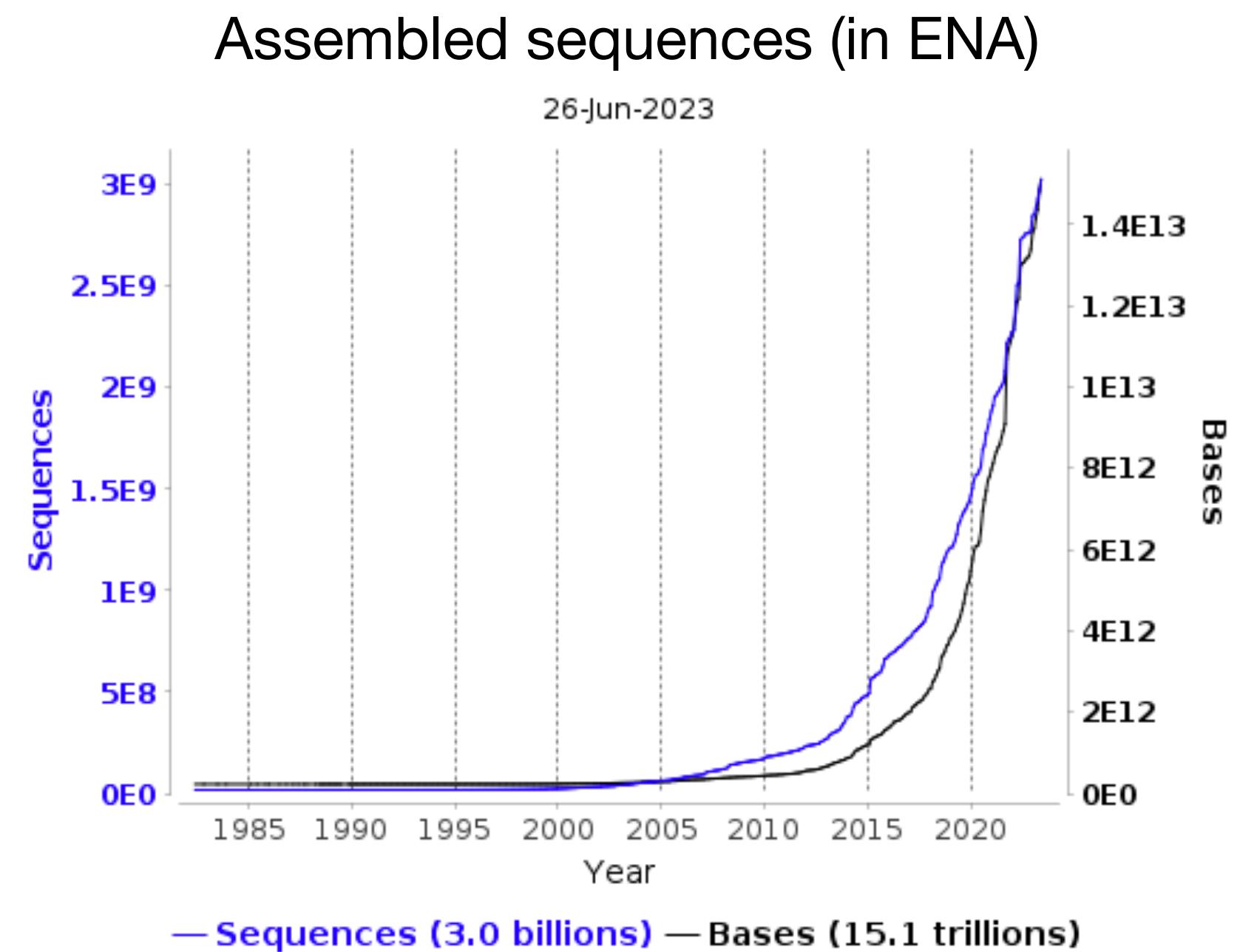
Sequences producing significant alignments Download Select columns Show 100 ?

select all 0 sequences selected GenBank Graphics Distance tree of results MSA Viewer

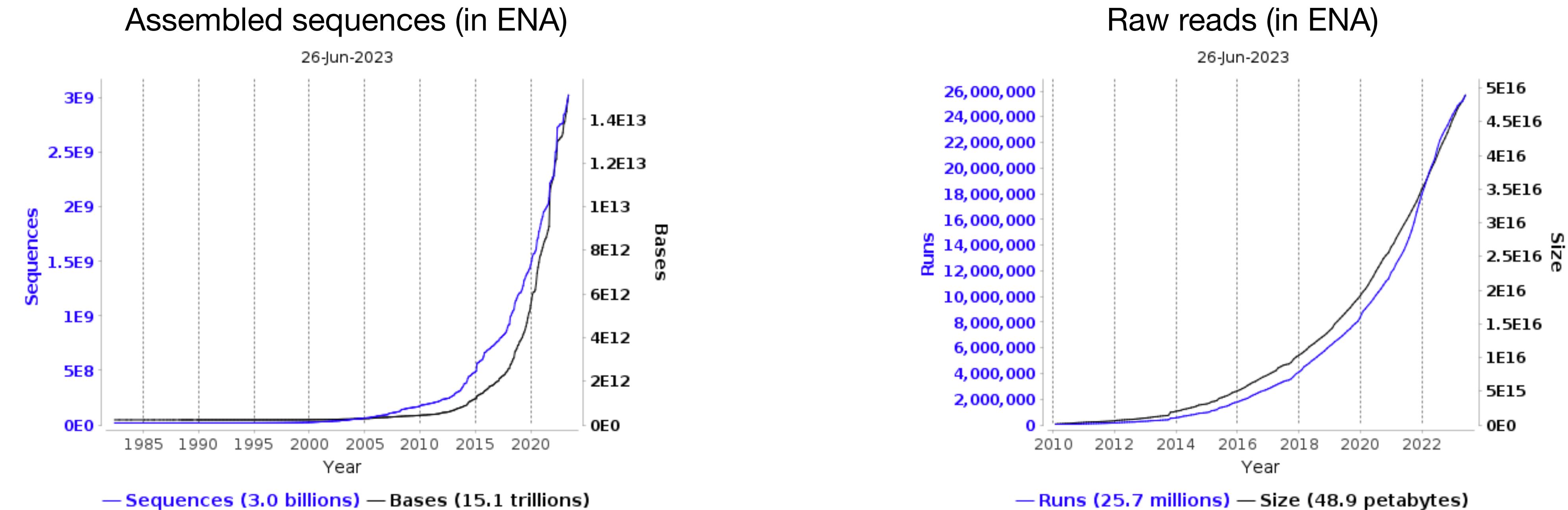
	Description	Scientific Name	Max Score	Total Score	Query Cover	E value	Per. Ident	Acc. Len	Accession
<input type="checkbox"/>	Cutibacterium acnes SZ2 DNA, complete genome	Cutibacterium a...	67.6	202	100%	3e-08	100.00%	2504552	AP022845.1
<input type="checkbox"/>	Cutibacterium acnes SZ1 DNA, complete genome	Cutibacterium a...	67.6	202	100%	3e-08	100.00%	2494525	AP022844.1
<input type="checkbox"/>	Cutibacterium acnes KPA171202 chromosome, complete genome	Cutibacterium a...	67.6	202	100%	3e-08	100.00%	2560634	CP025935.1
<input type="checkbox"/>	Cutibacterium acnes DSM 1897 chromosome, complete genome	Cutibacterium a...	67.6	202	100%	3e-08	100.00%	2495002	CP025934.1

assembled sequences

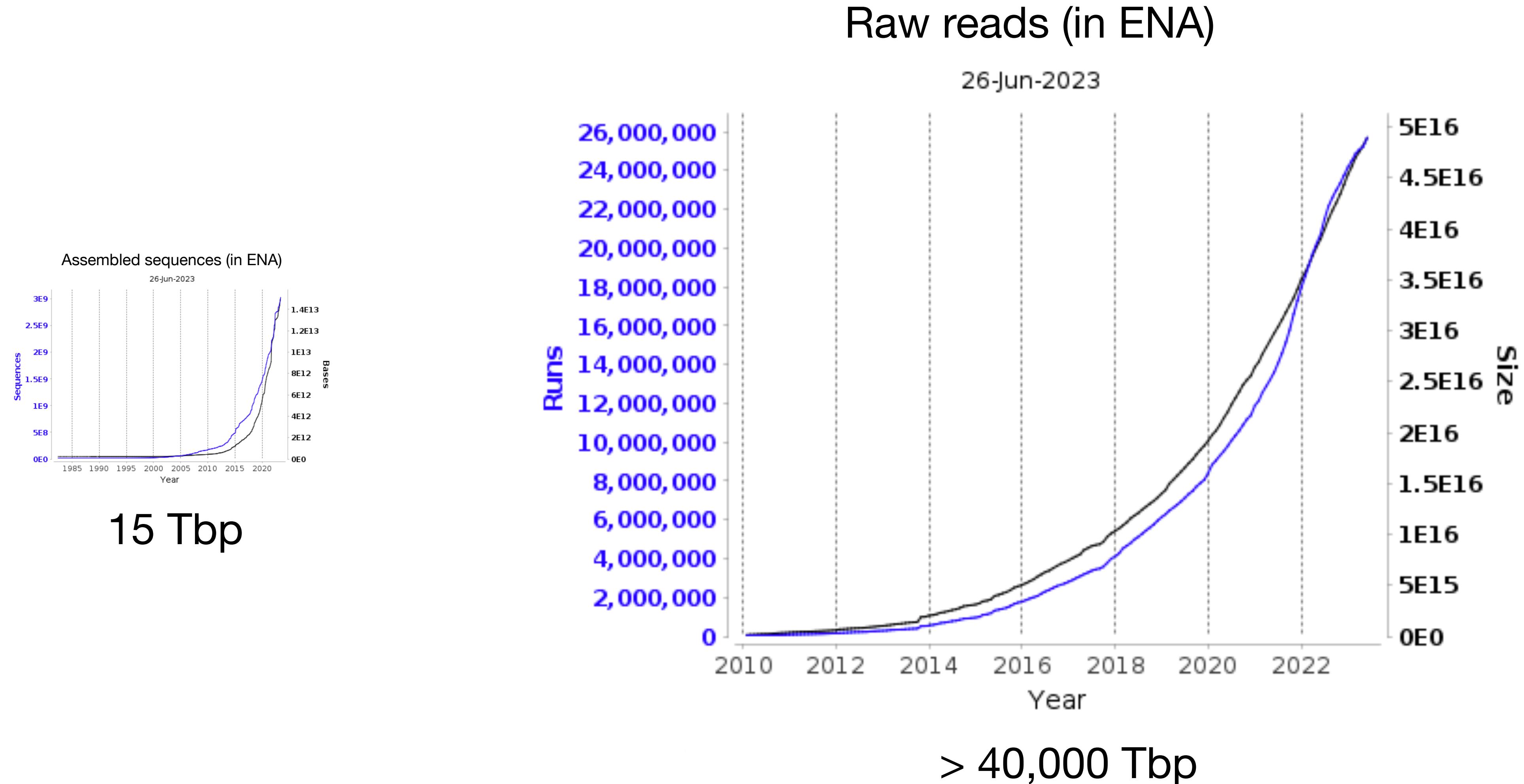
Growth of sequence archives



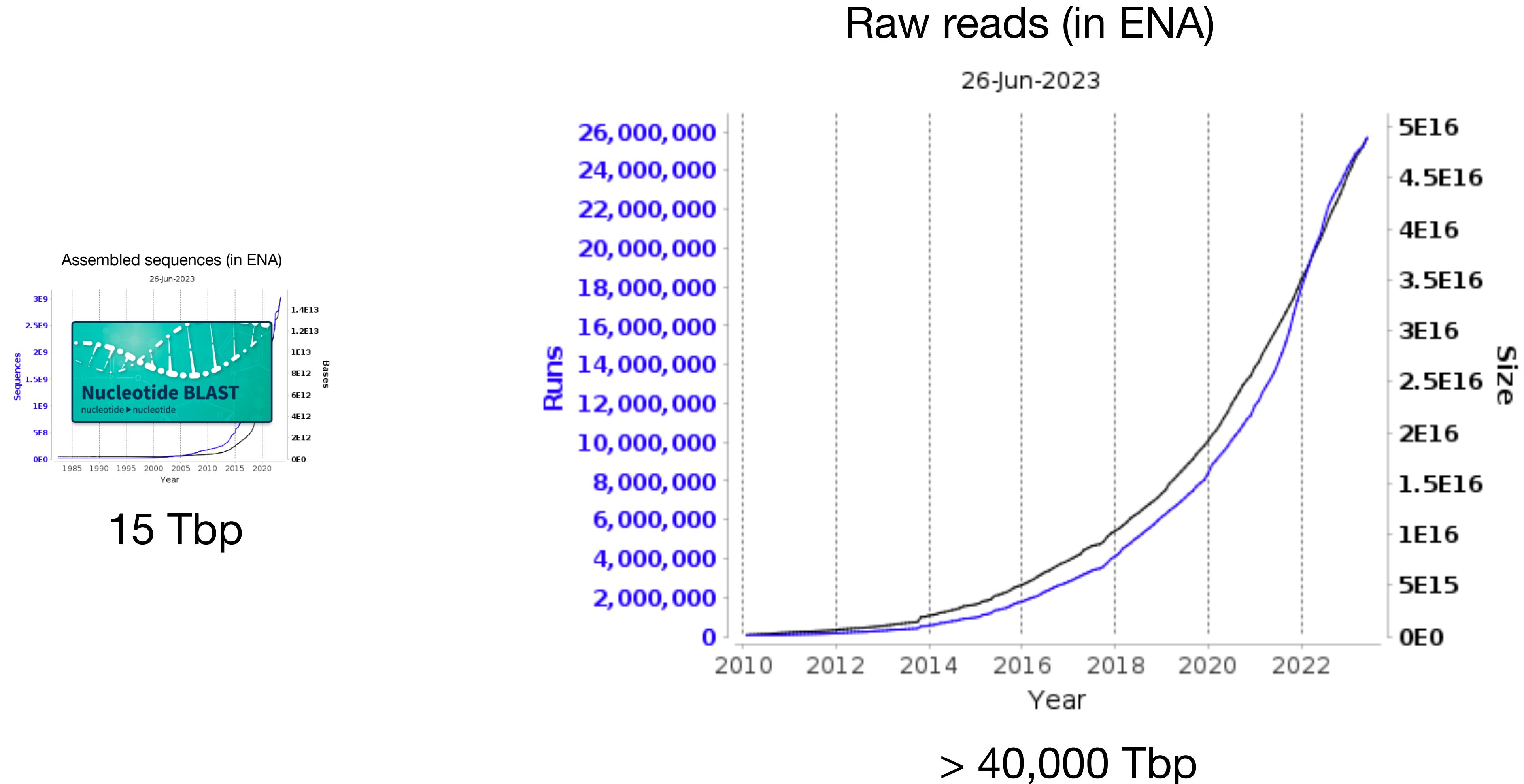
Growth of sequence archives



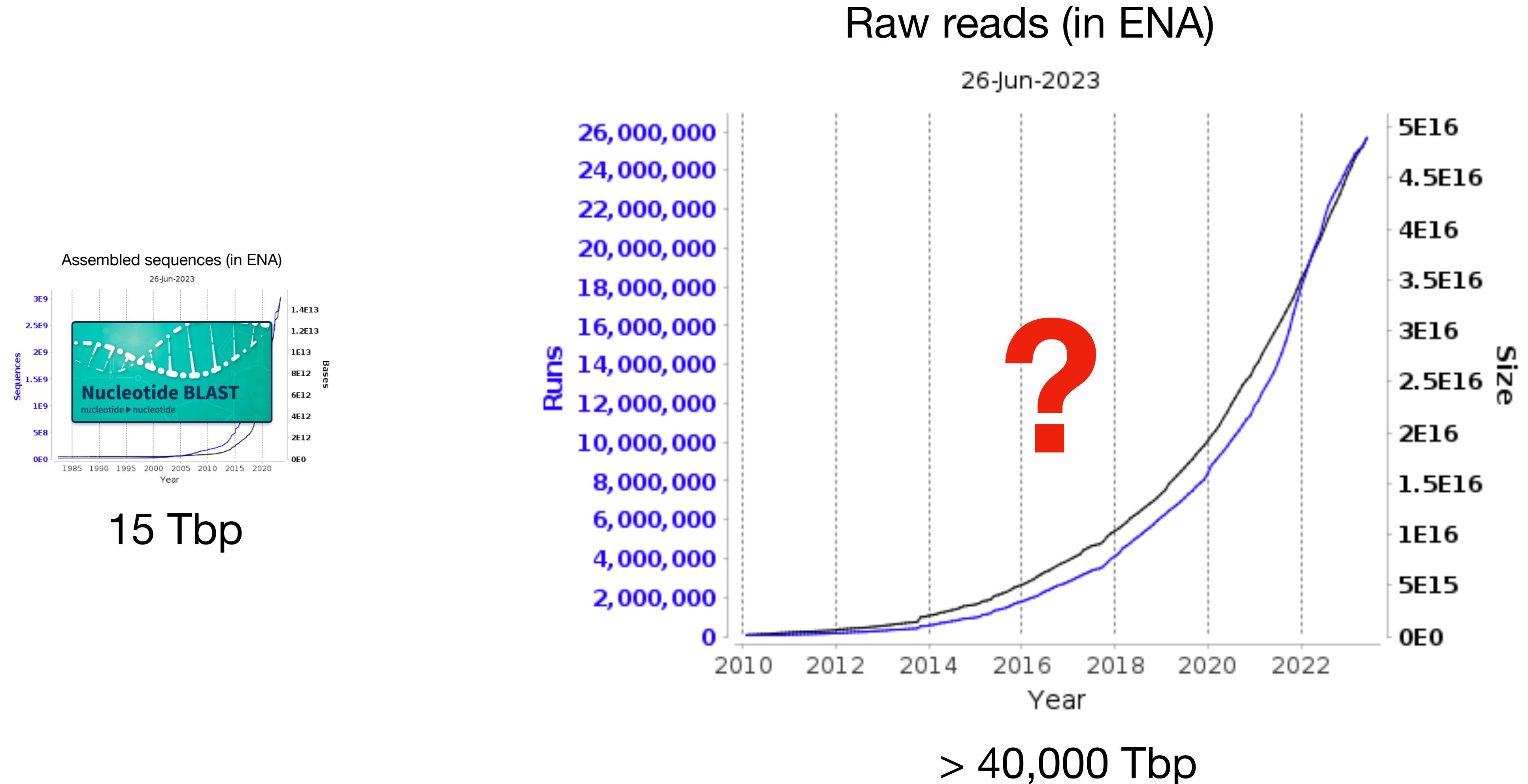
Growth of sequence archives



Growth of sequence archives



Growth of sequence archives



Indexing sequences

- ▶ Why not use methods from typical search engines?

Indexing sequences

- ▶ Why not use methods from typical search engines?
 - sequences lack structure of natural language ("AAAGTTGTGT" vs. "cake recipes")
 - enormous amount of data (> 50 PB)
 - special metrics for inexact search (sequence alignment)

Indexing sequences

- ▶ Why not use methods from typical search engines?
 - sequences lack structure of natural language ("AAAGTTGTGT" vs. "cake recipes")
 - enormous amount of data (> 50 PB)
 - special metrics for inexact search (sequence alignment)
- ▶ Burrows-Wheeler transform and FM-index do not scale

Indexing sequences

- ▶ Why not use methods from typical search engines?
 - sequences lack structure of natural language ("AAAGTTGTGT" vs. "cake recipes")
 - enormous amount of data (> 50 PB)
 - special metrics for inexact search (sequence alignment)
- ▶ Burrows-Wheeler transform and FM-index do not scale
- ▶ Need something simpler

Indexing sequences

- ▶ Why not use methods from typical search engines?
 - sequences lack structure of natural language ("AAAGTTGTGT" vs. "cake recipes")
 - enormous amount of data (> 50 PB)
 - special metrics for inexact search (sequence alignment)
- ▶ Burrows-Wheeler transform and FM-index do not scale
- ▶ Need something simpler
De Bruijn graph

Background

De Bruijn graphs have been introduced to the field by Pevzner in 1989

Background

De Bruijn graphs have been introduced to the field by Pevzner in 1989

ACTAGCTAGCTAG

Background

De Bruijn graphs have been introduced to the field by Pevzner in 1989

ACTAGCTAGCTAG

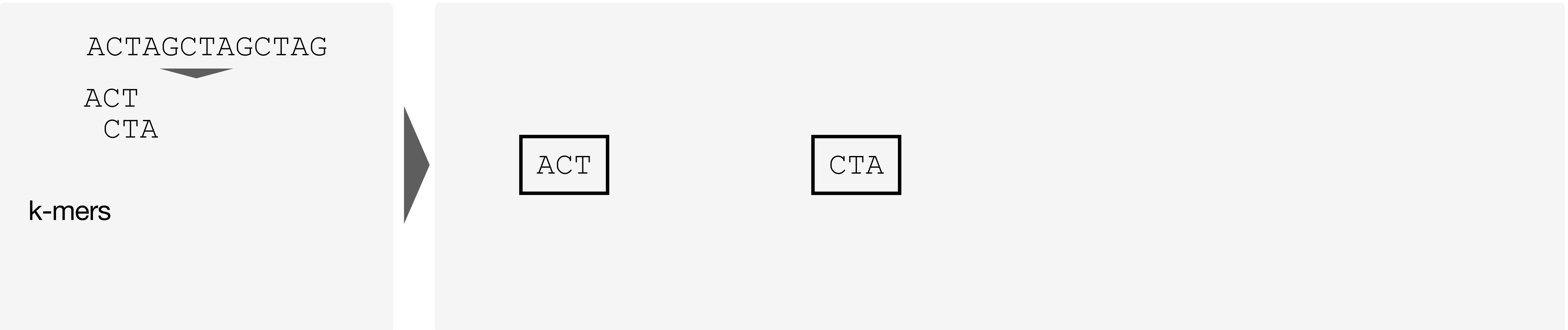
ACT

CTA

k-mers

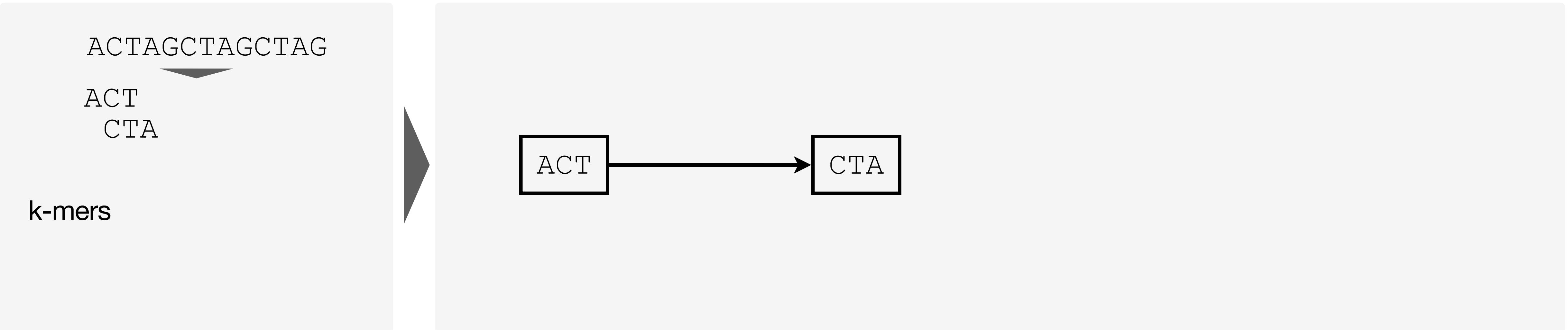
Background

De Bruijn graphs have been introduced to the field by Pevzner in 1989



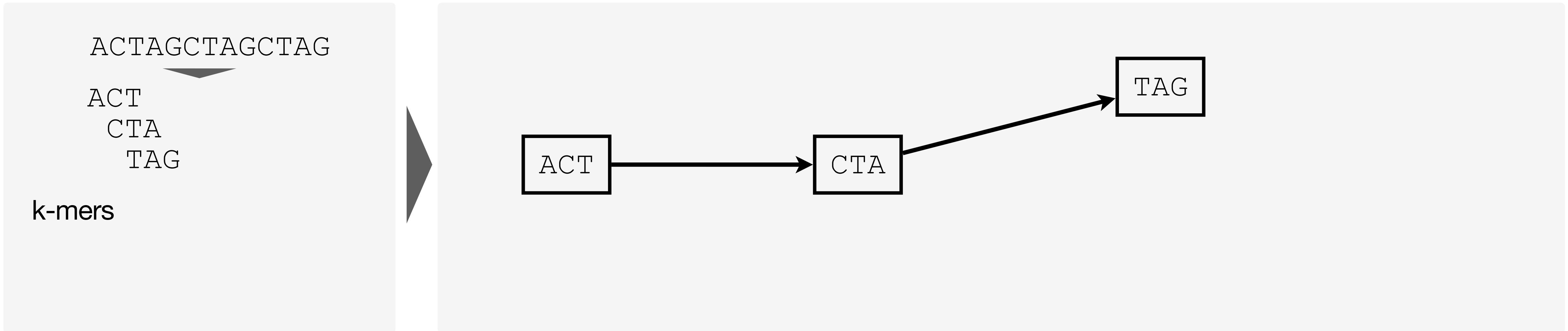
Background

De Bruijn graphs have been introduced to the field by Pevzner in 1989



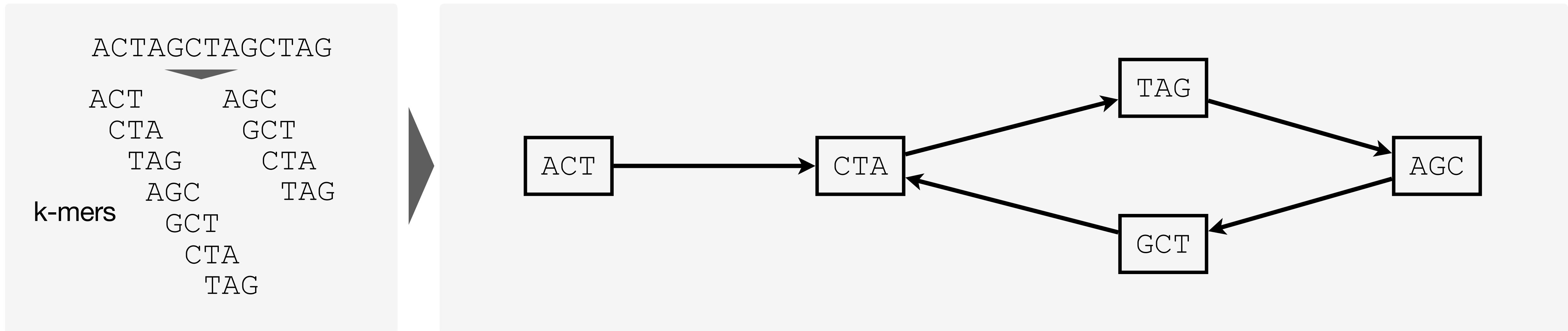
Background

De Bruijn graphs have been introduced to the field by Pevzner in 1989



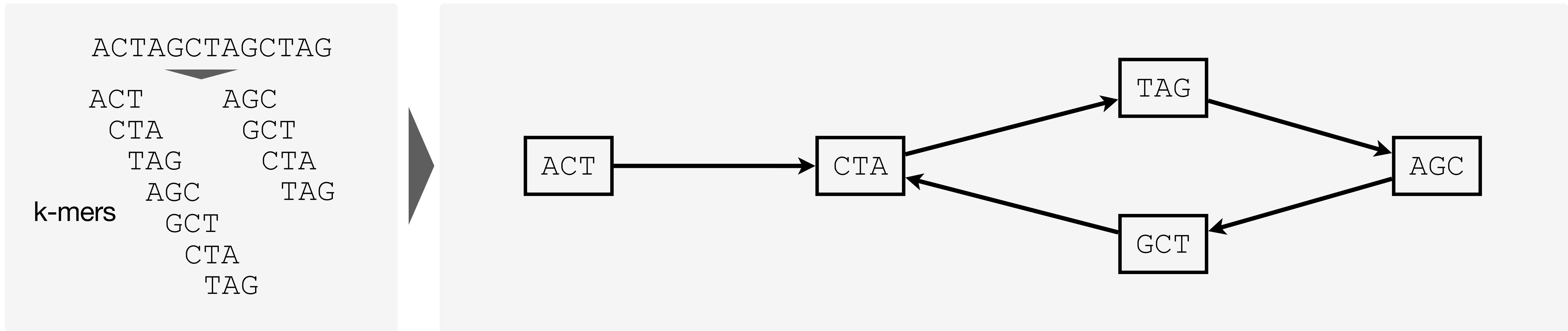
Background

De Bruijn graphs have been introduced to the field by Pevzner in 1989



Background

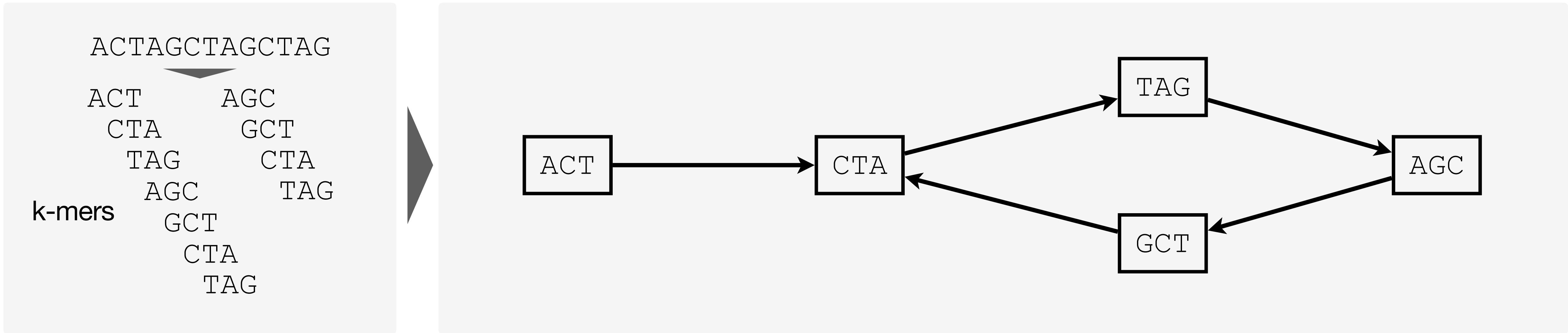
De Bruijn graphs have been introduced to the field by Pevzner in 1989



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

Background

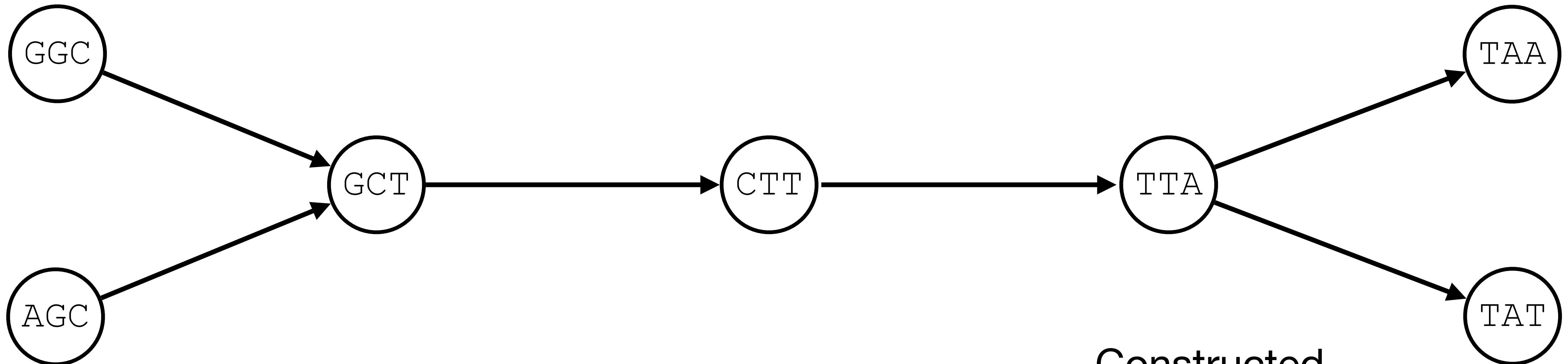
De Bruijn graphs have been introduced to the field by Pevzner in 1989



- ▶ Originally employed for ***de novo* assembly**
 - [Pevzner, 1989]
 - [Idury, Waterman, 1995]
 - [Pevzner, Tang, Waterman, 2001]
 - [Zerbino, Birney, 2008]

Background

Annotated De Bruijn graphs

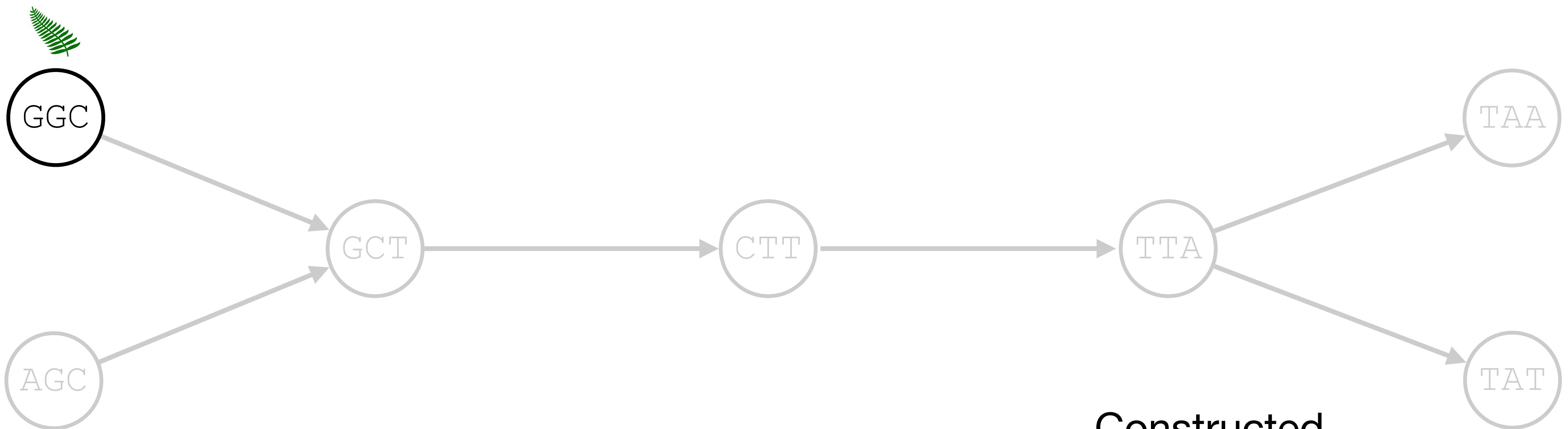


Constructed
from sequences

- L1: GGCTTAT 
- L2: AGCTTAA 
- L3: TTAA 

Background

Annotated De Bruijn graphs

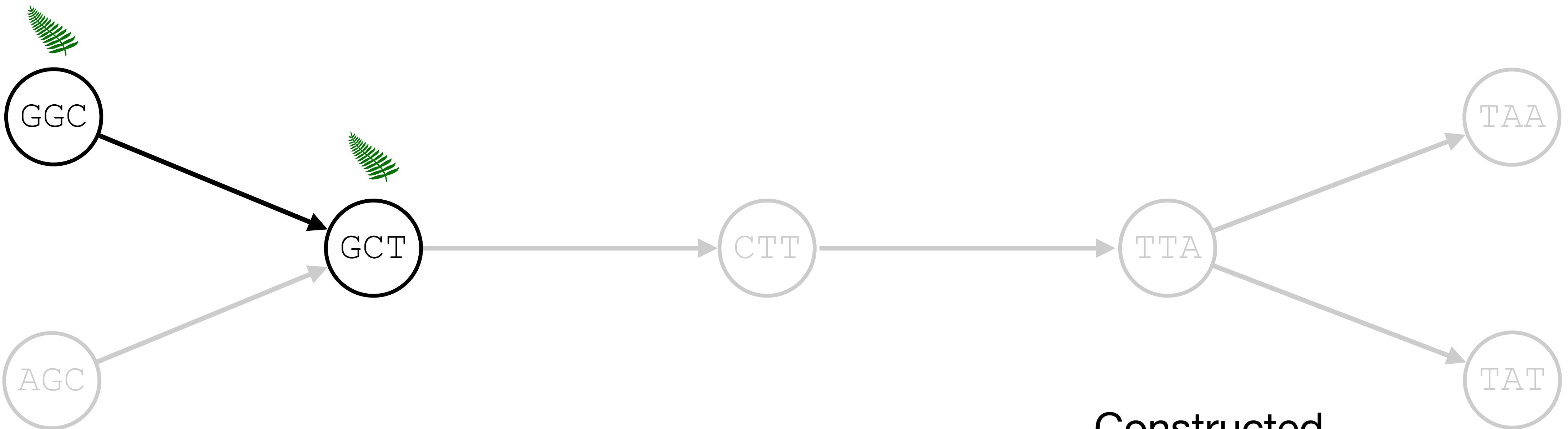


Constructed
from sequences

L1: **GGC**TTAT 
L2: AGCTTAA 
L3: TTAA 

Background

Annotated De Bruijn graphs

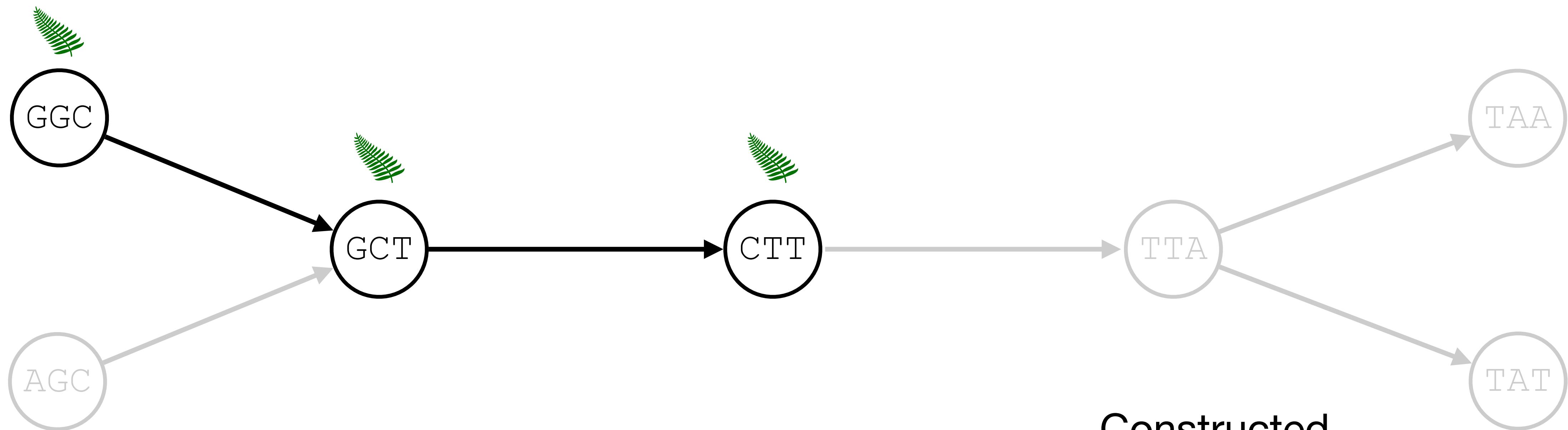


Constructed
from sequences

L1: GG**G**CTTAT 
L2: AGCTTAA 
L3: TTAA 

Background

Annotated De Bruijn graphs

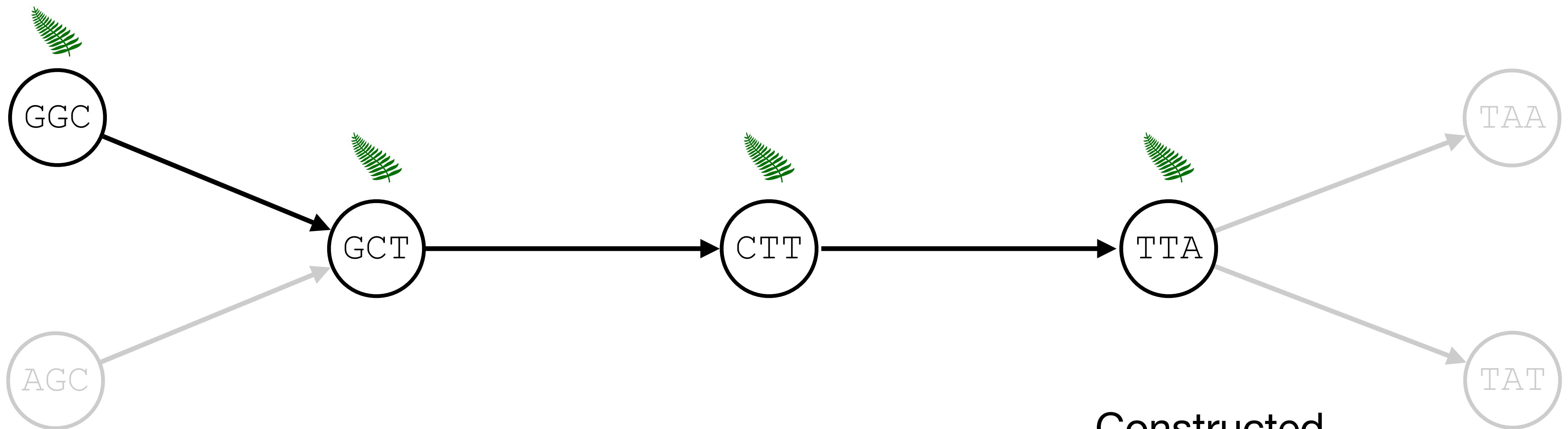


Constructed
from sequences

L1: GGCTTAT fern icon
L2: AGCTTAA person icon
L3: TTAA virus icon

Background

Annotated De Bruijn graphs

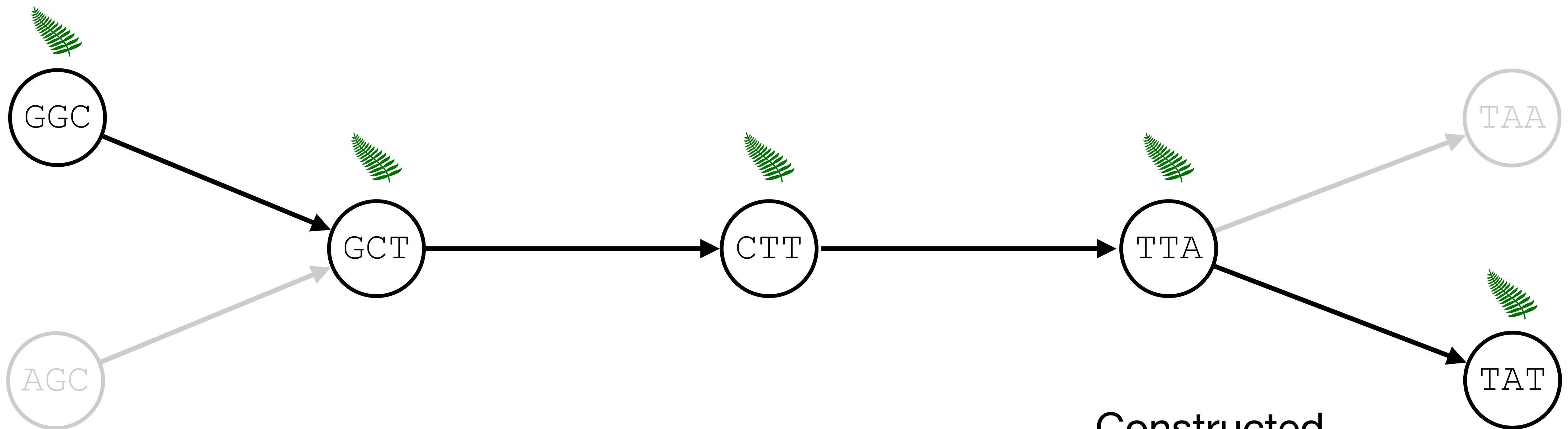


Constructed
from sequences

- L1: GGCTTAT 
- L2: AGCTTAA 
- L3: TTAA 

Background

Annotated De Bruijn graphs

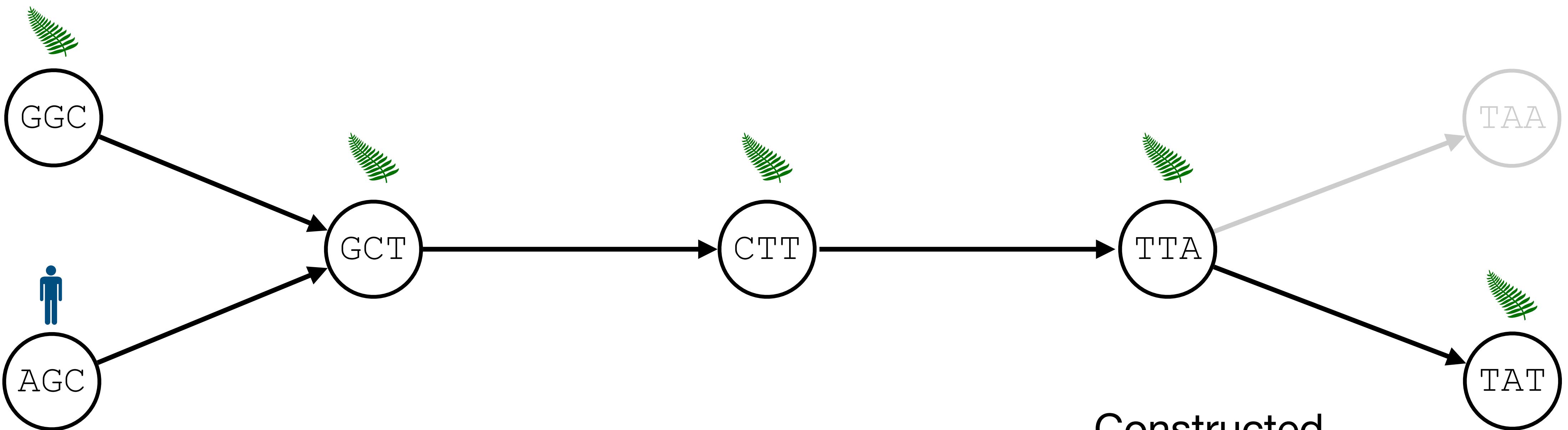


Constructed
from sequences

- L1: GGCTTAT 
- L2: AGCTTAA 
- L3: TTAA 

Background

Annotated De Bruijn graphs



Constructed
from sequences

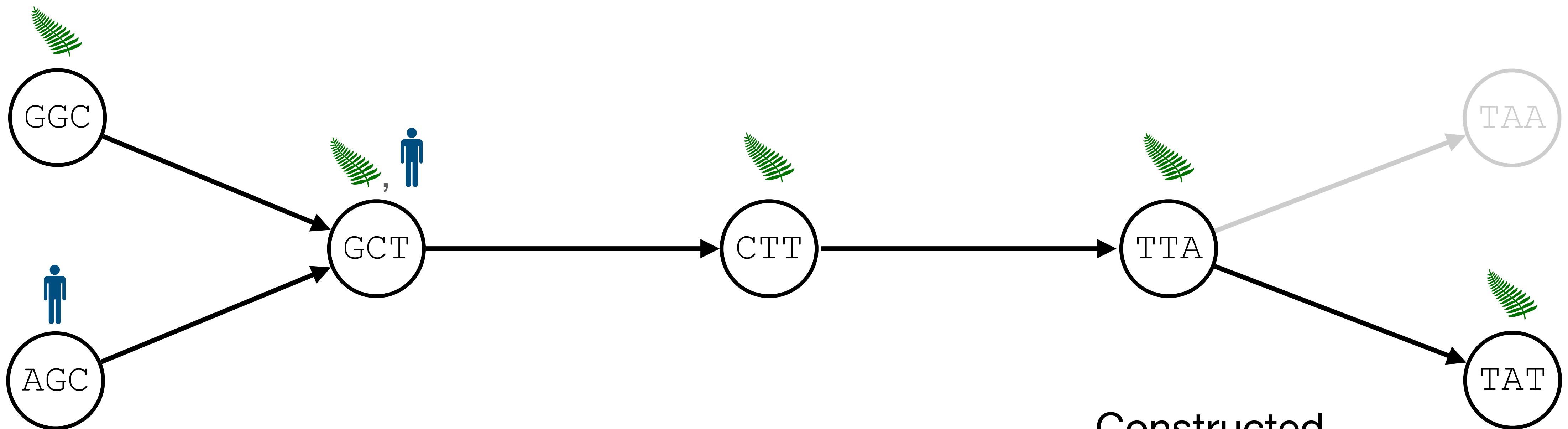
L1: GGCTTAT

L2: AGCTTAA

L3: TTAA

Background

Annotated De Bruijn graphs



Constructed
from sequences

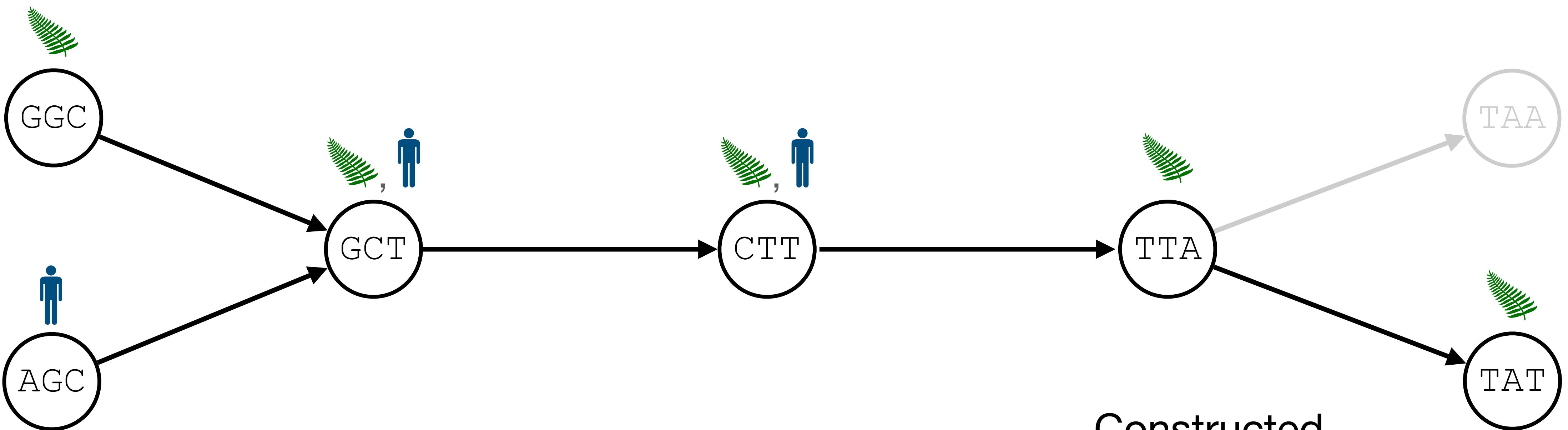
L1: GGCTTAT

L2: AGCTTAA

L3: TTAA

Background

Annotated De Bruijn graphs



Constructed
from sequences

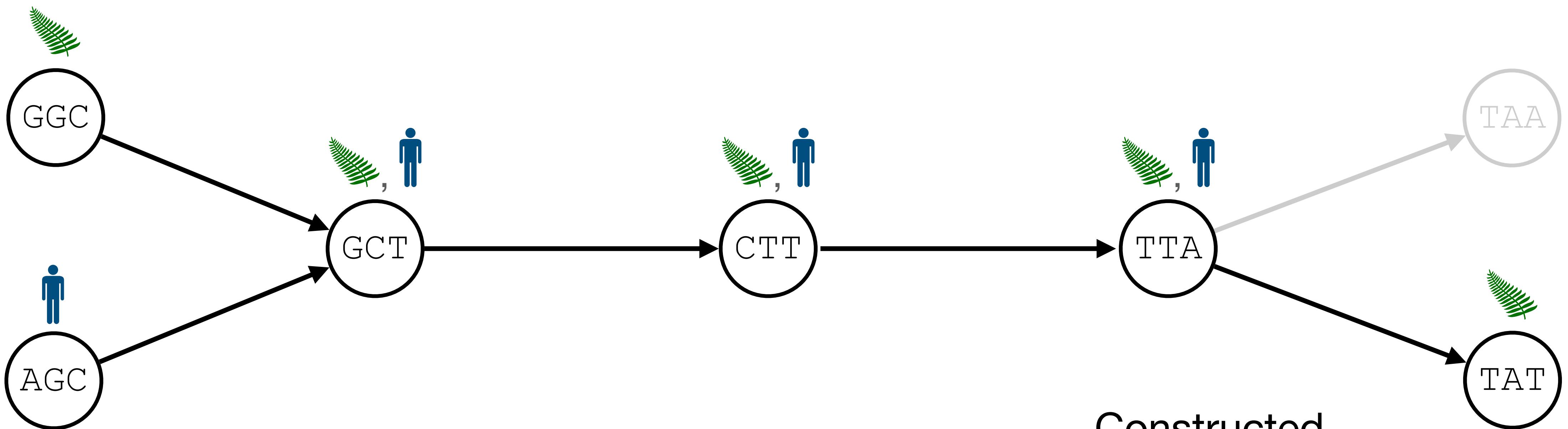
L1: GGCTTAT

L2: AGGCTTAA

L3: TTAA

Background

Annotated De Bruijn graphs



Constructed
from sequences

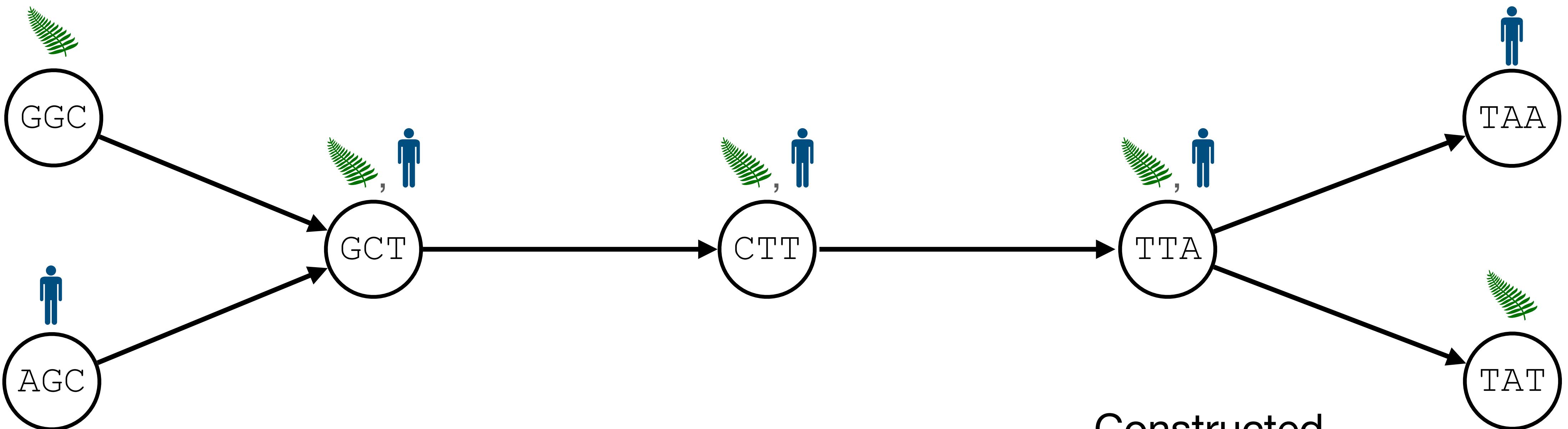
L1: GGCTTAT 

L2: AGC  TTAA

L3: TTAA 

Background

Annotated De Bruijn graphs



Constructed
from sequences

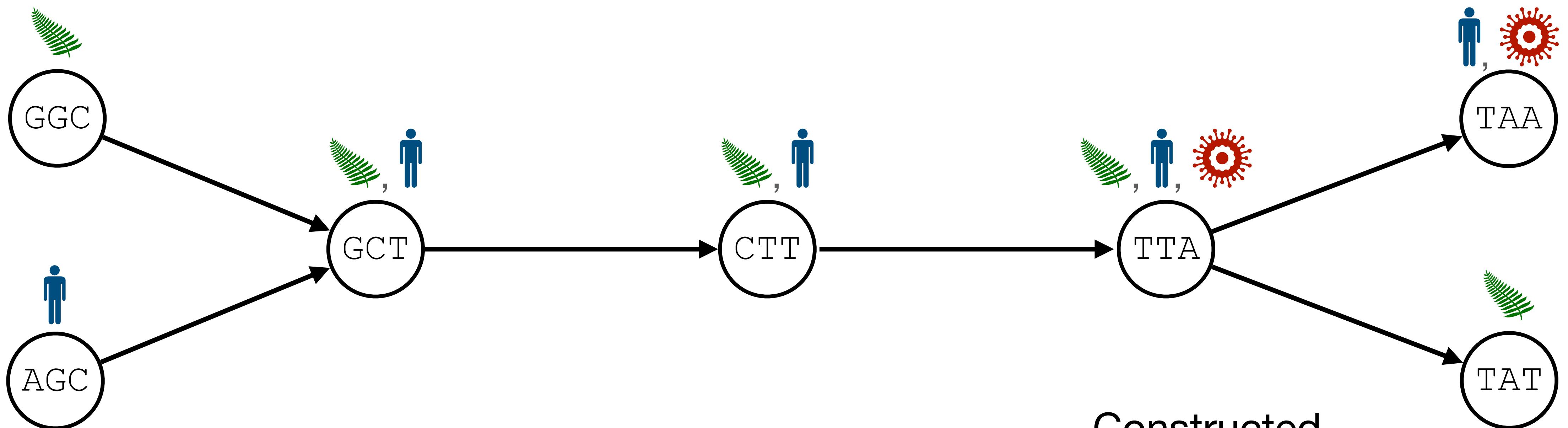
L1: GGCTTAT

L2: AGCTTAA

L3: TTAA

Background

Annotated De Bruijn graphs

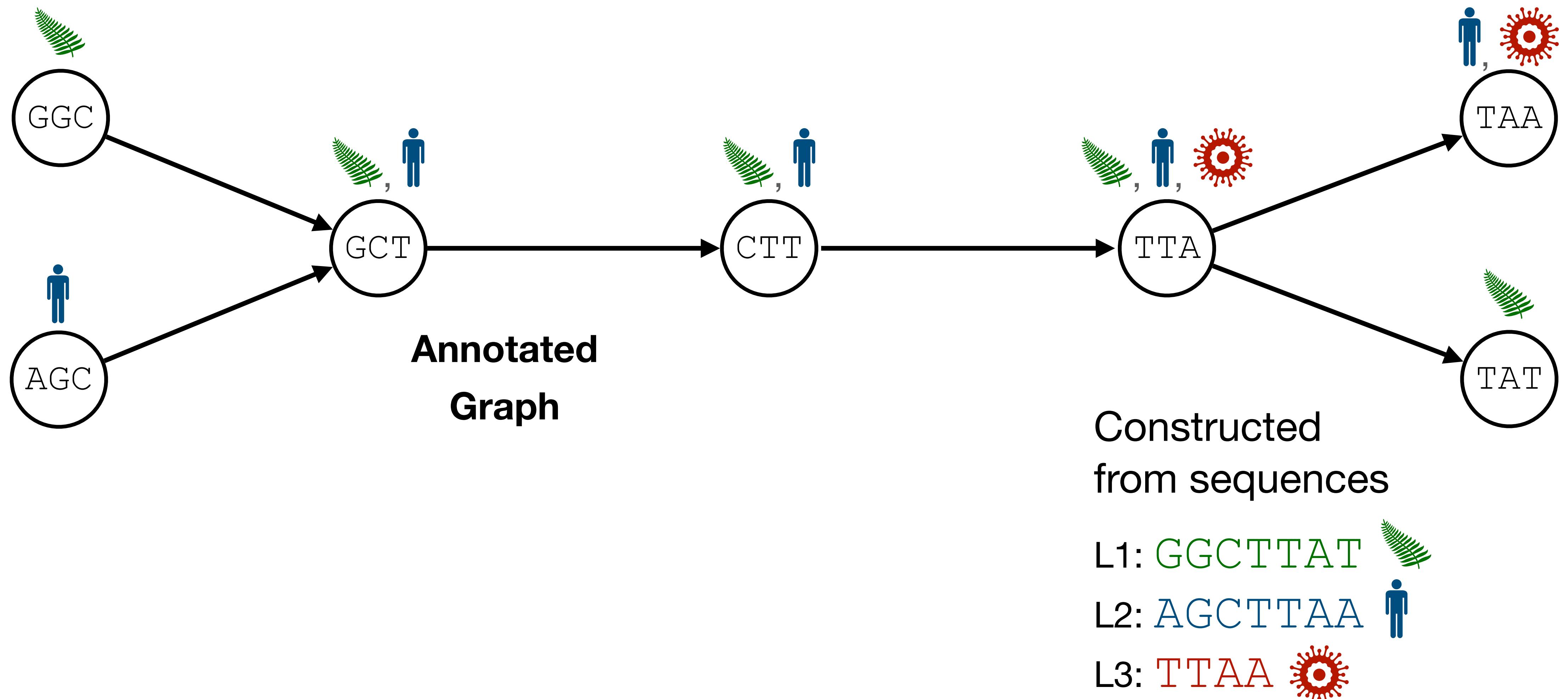


Constructed
from sequences

- L1: GGCTTAT 
- L2: AGCTTAA 
- L3: TTAA 

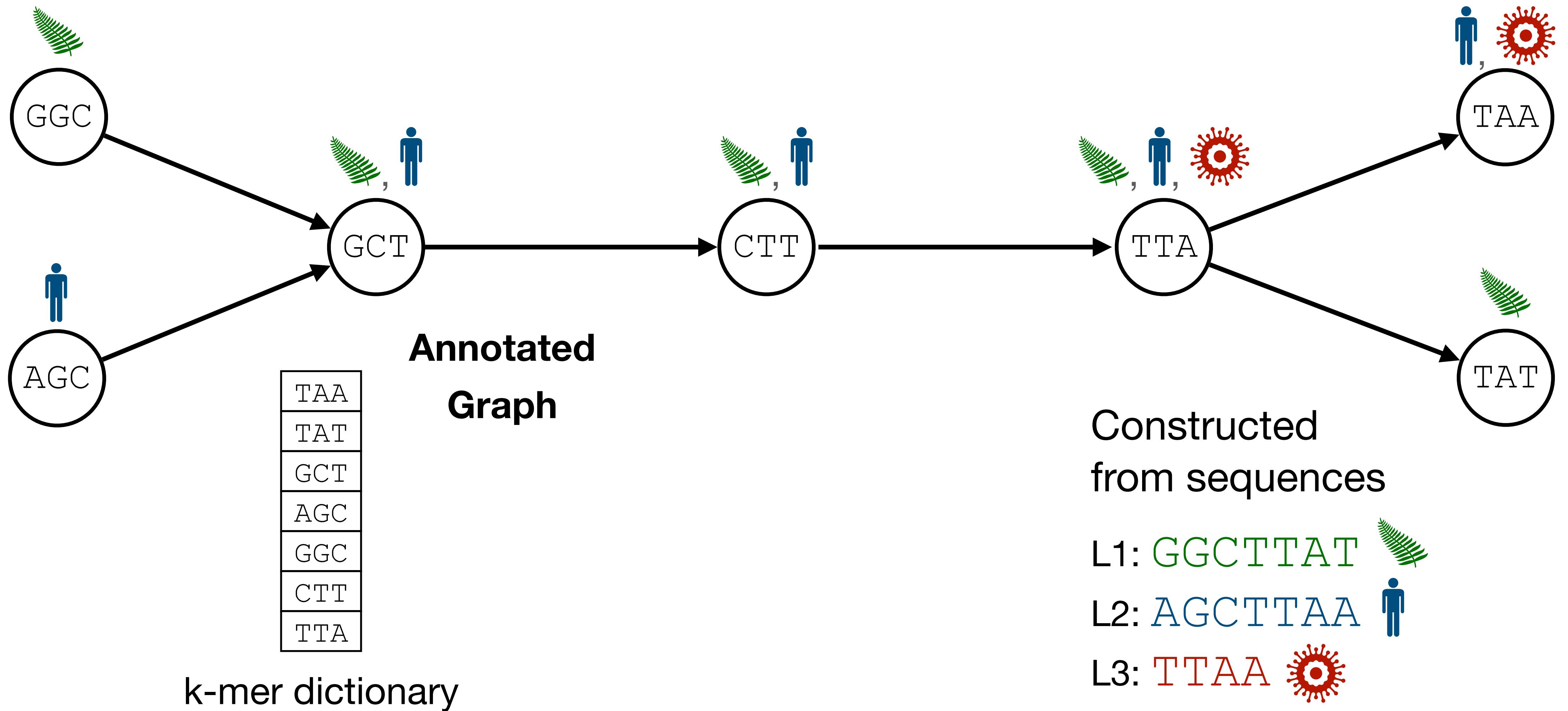
Background

Annotated De Bruijn graphs



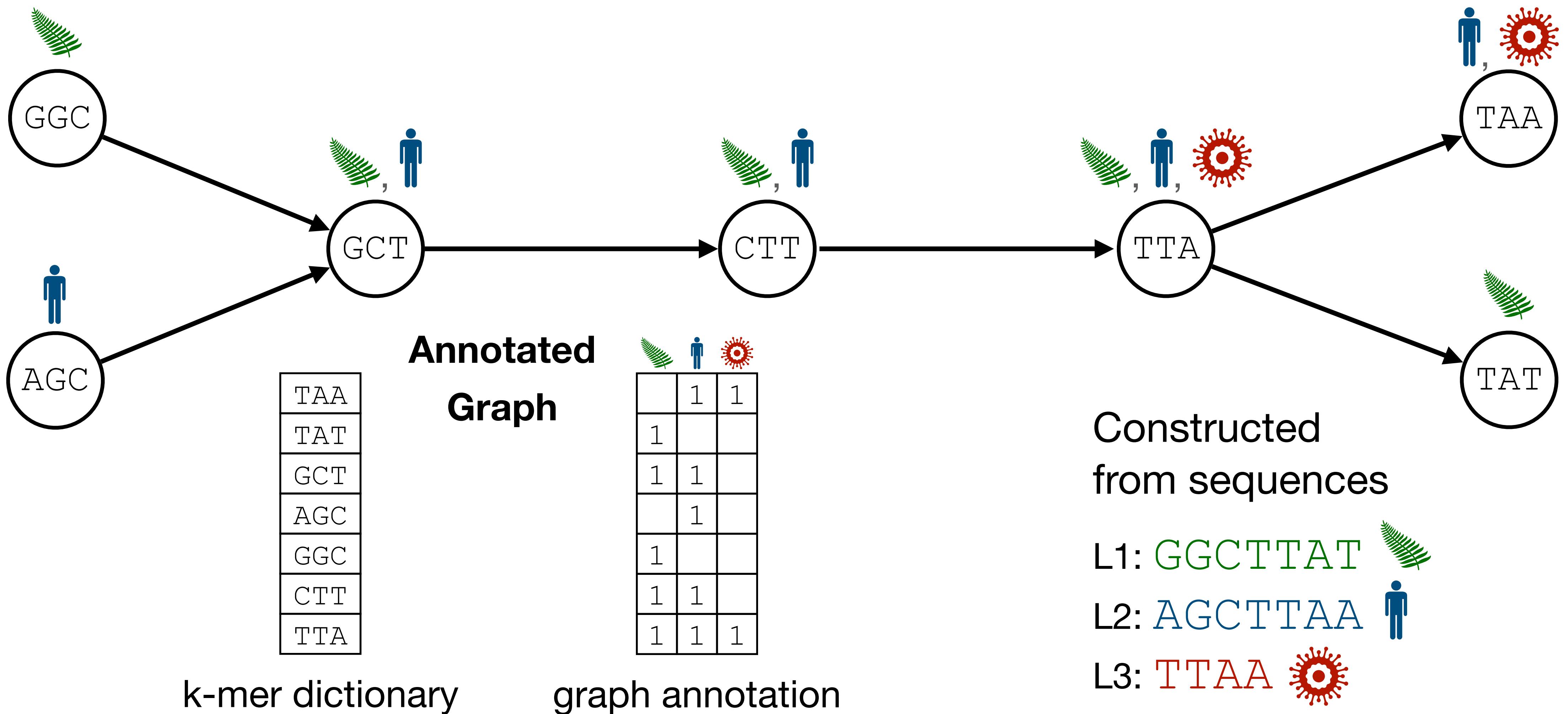
Background

Annotated De Bruijn graphs



Background

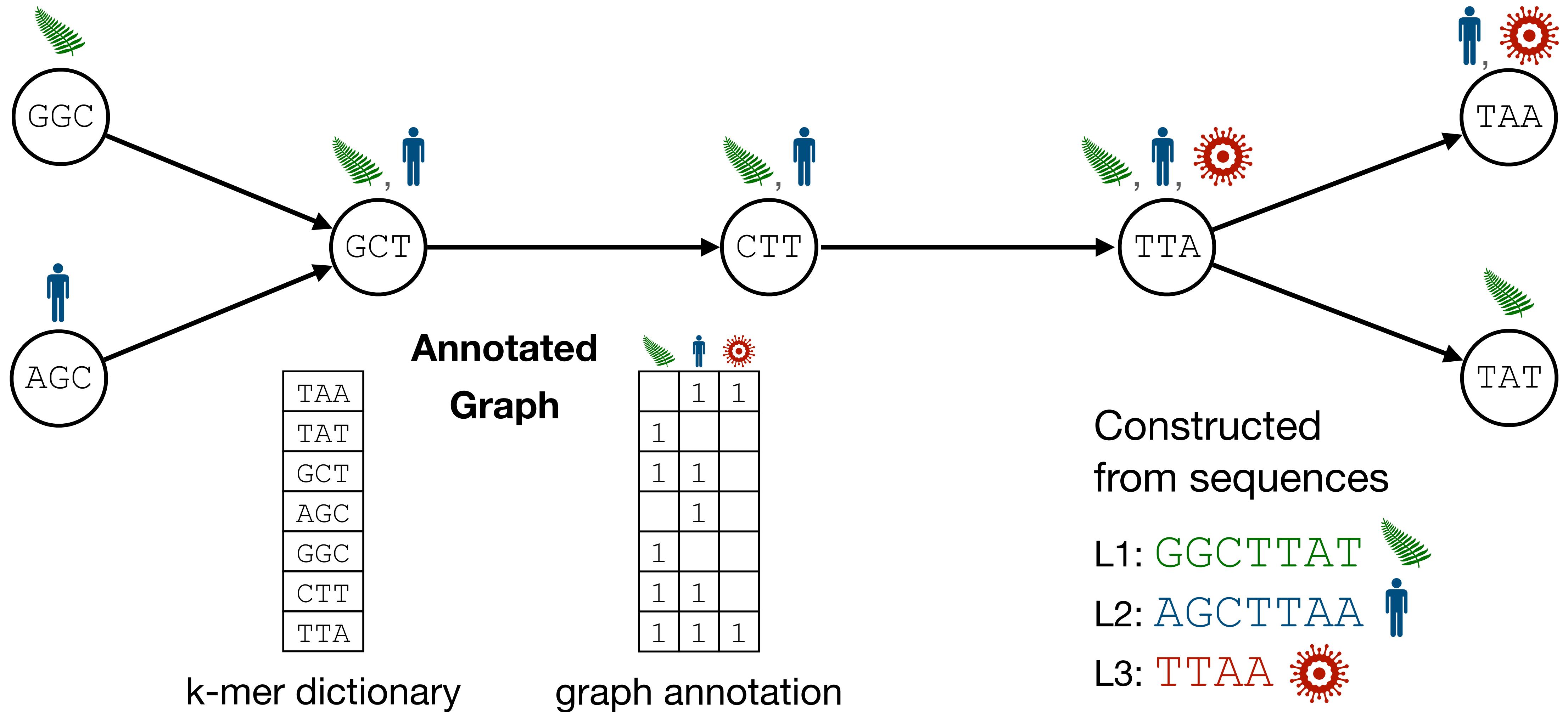
Annotated De Bruijn graphs



Background

Annotated De Bruijn graphs

- [Iqbal, Caccamo, Turner, Flicek, McVean, 2012]
- [Muggli *et al.*, 2017]



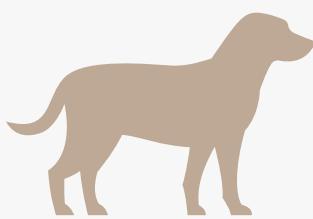
Indexing workflow



>smp_1
ACGTAC
ACGC
CGTAC



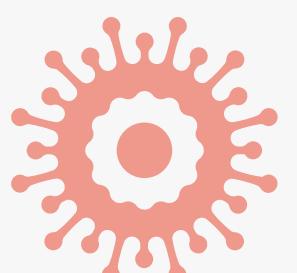
>smp_2
ACGAA
ACGTAC
ACG



>smp_3
ACGTA
GTACT
ACGAT



...

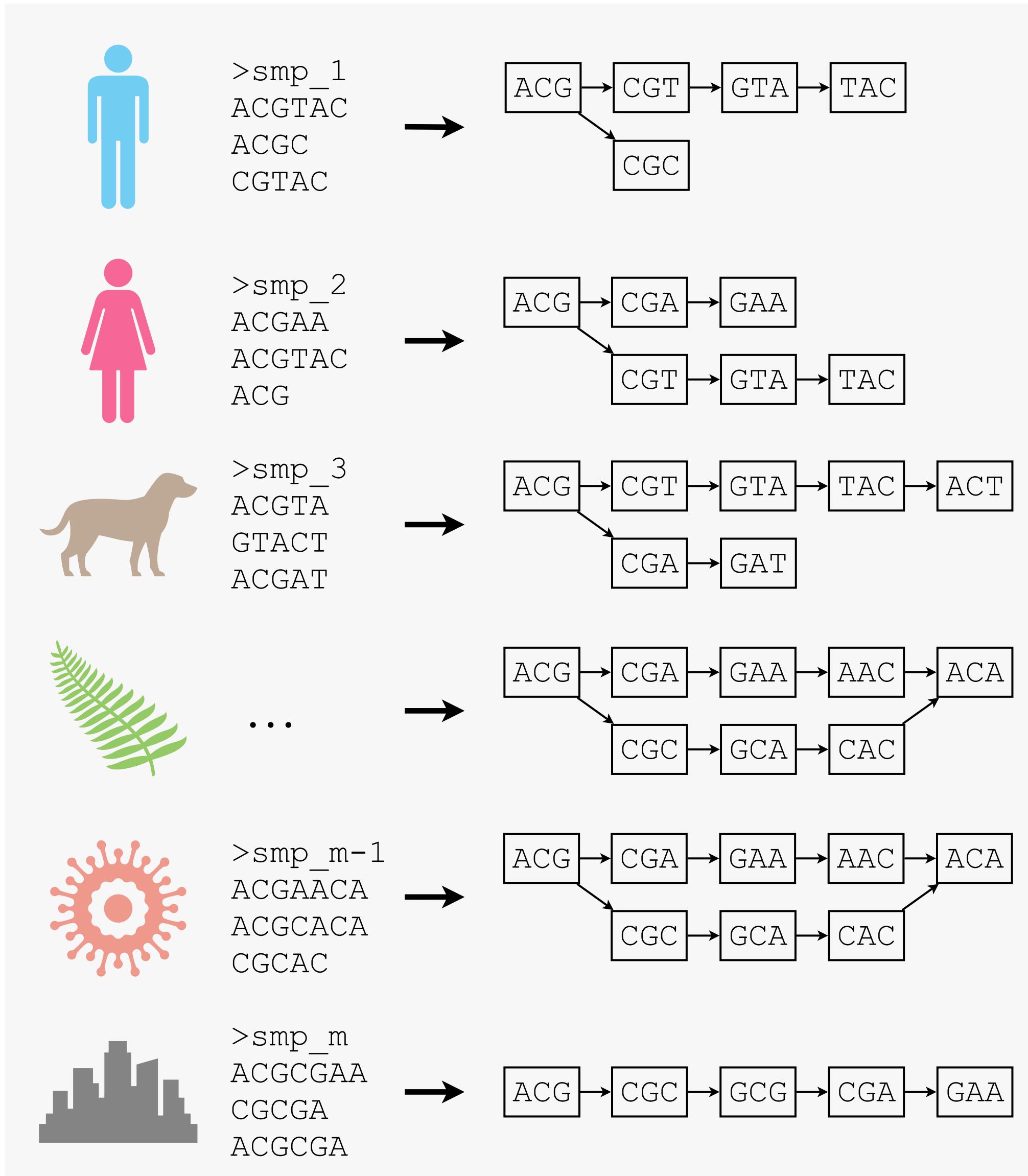


>smp_m-1
ACGAACA
ACGCACA
CGCAC

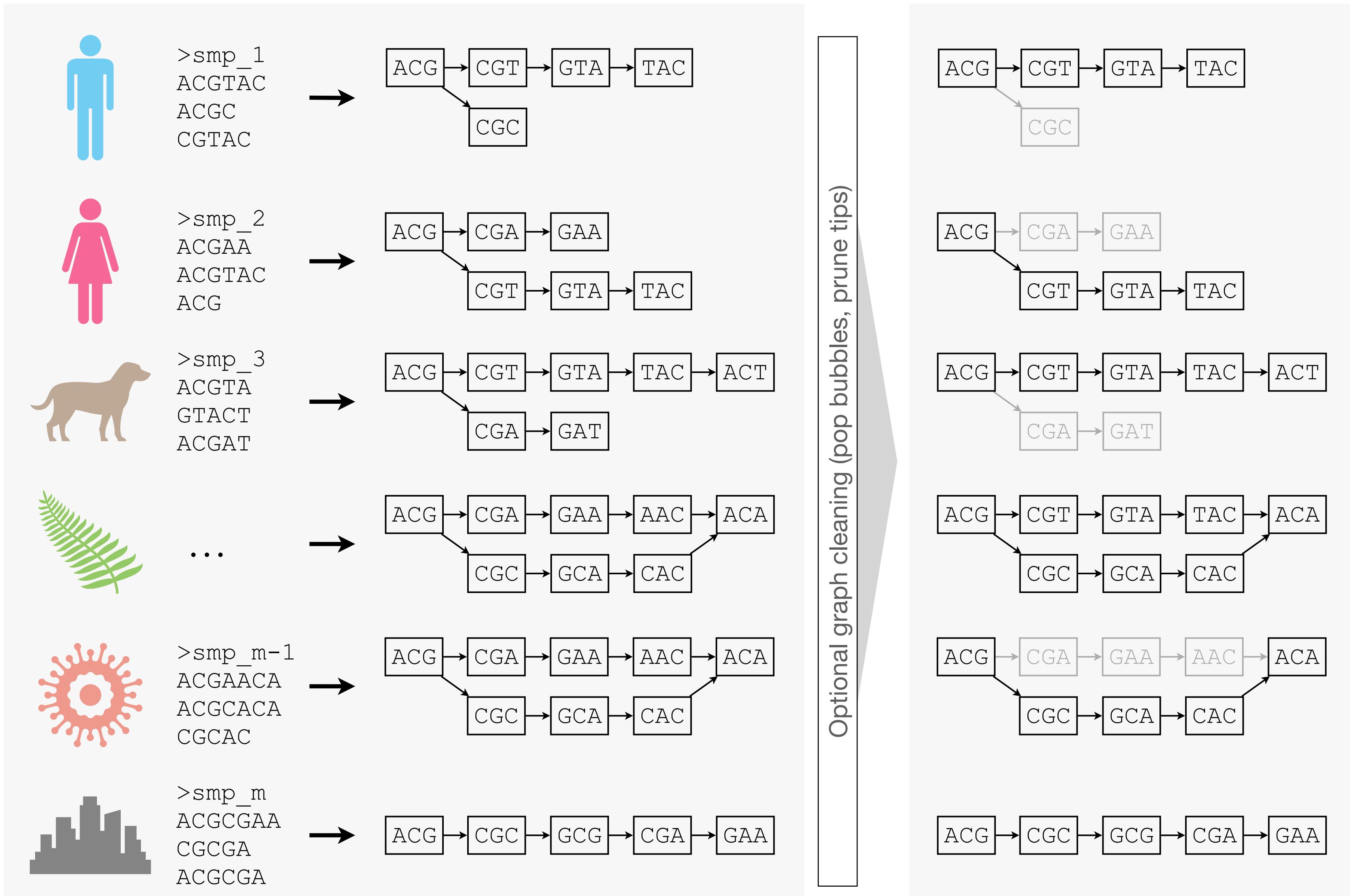


>smp_m
ACGCGAA
CGCGA
ACGCGA

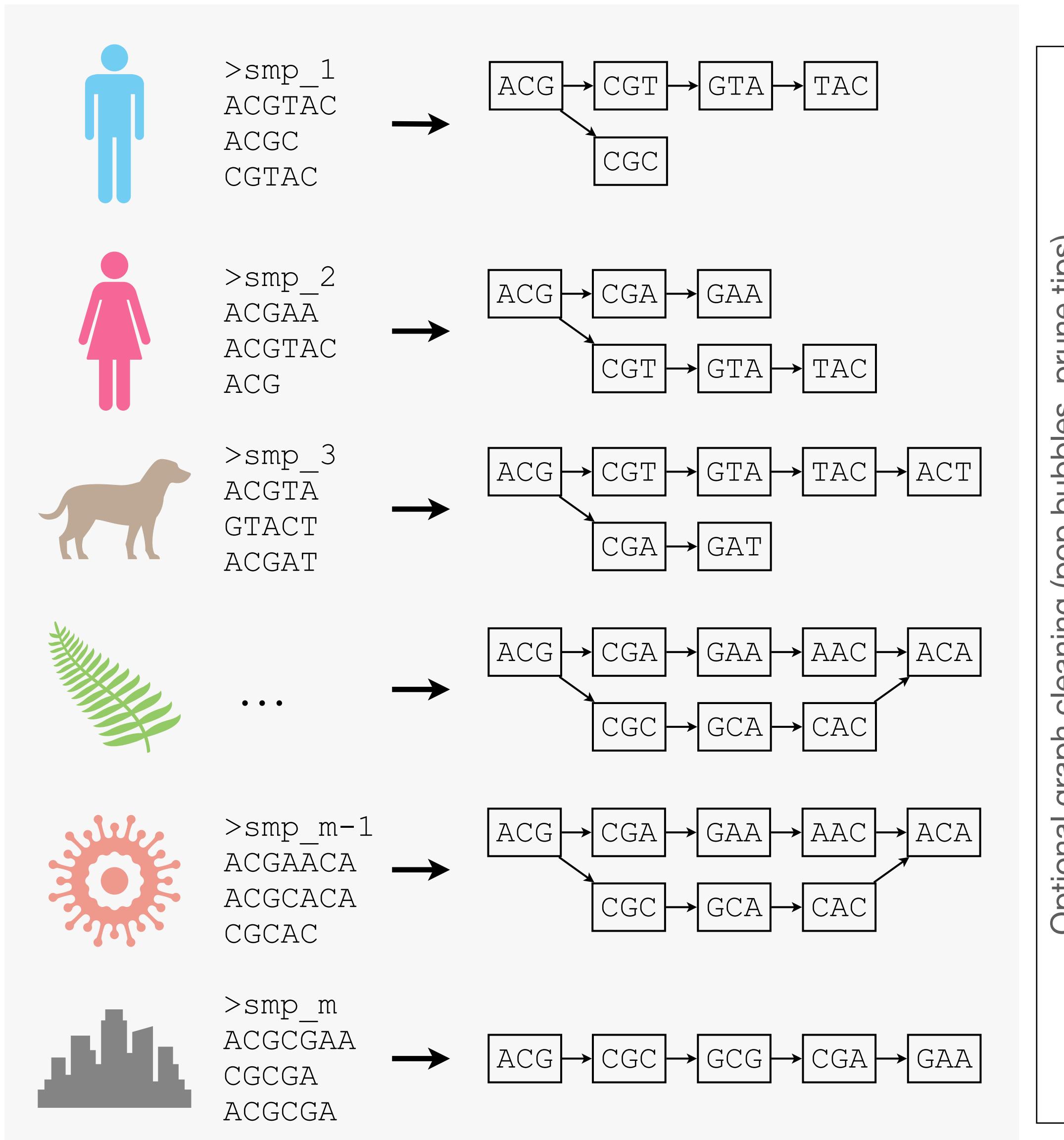
Indexing workflow



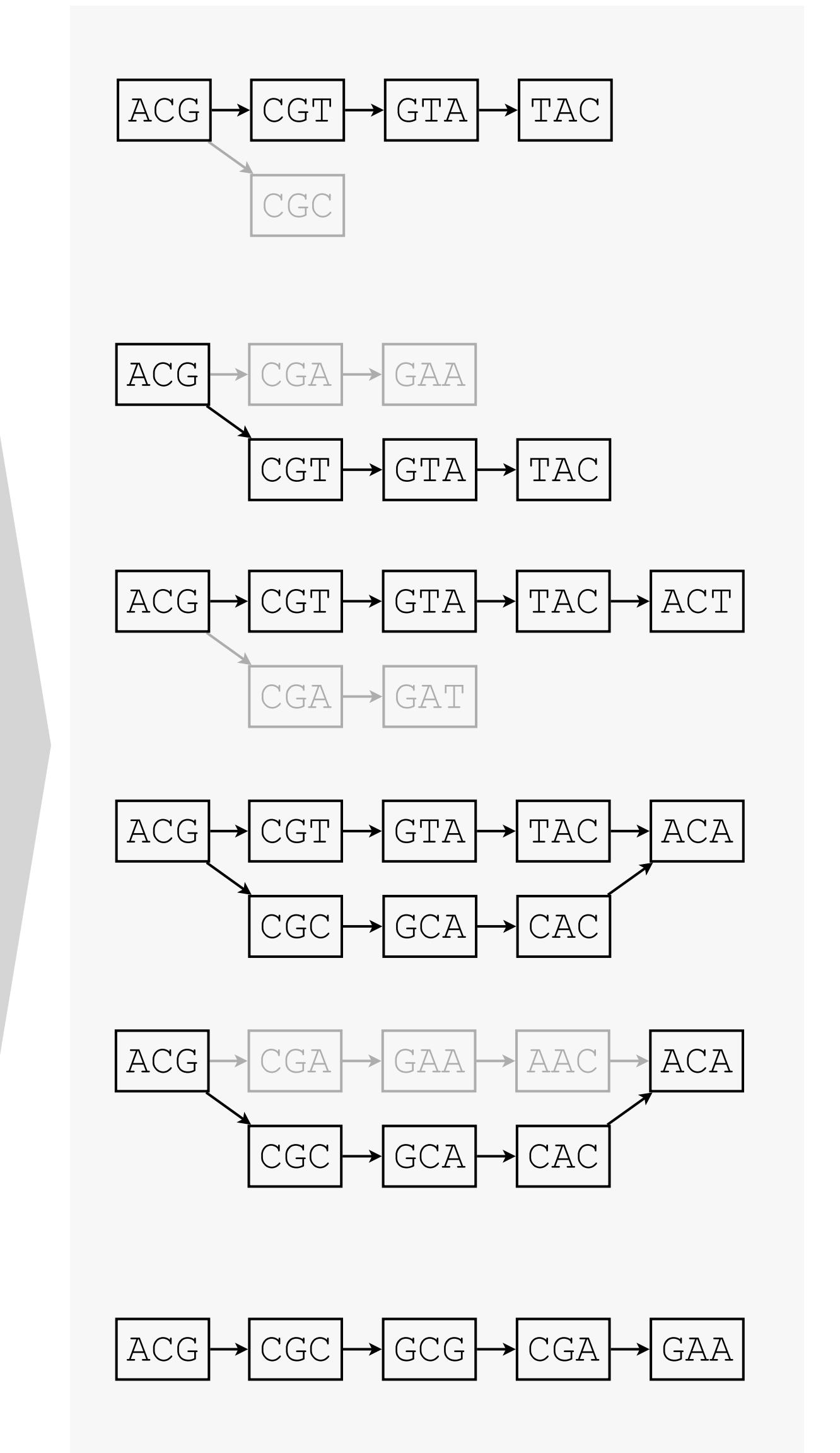
Indexing workflow



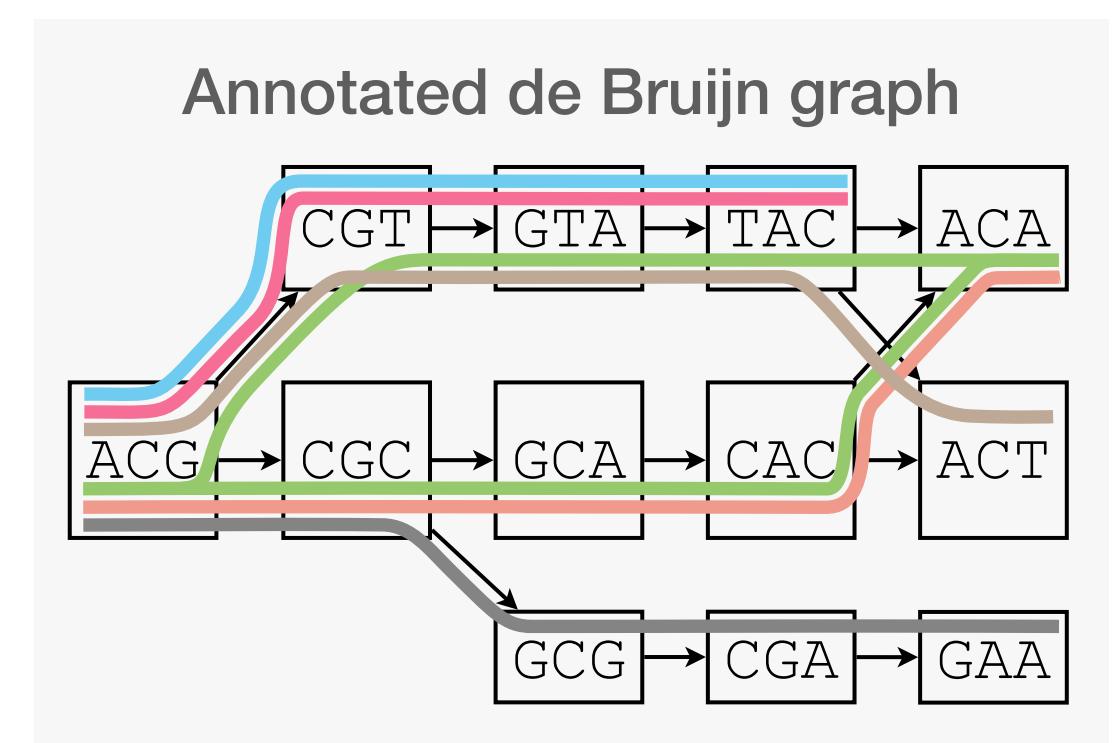
Indexing workflow



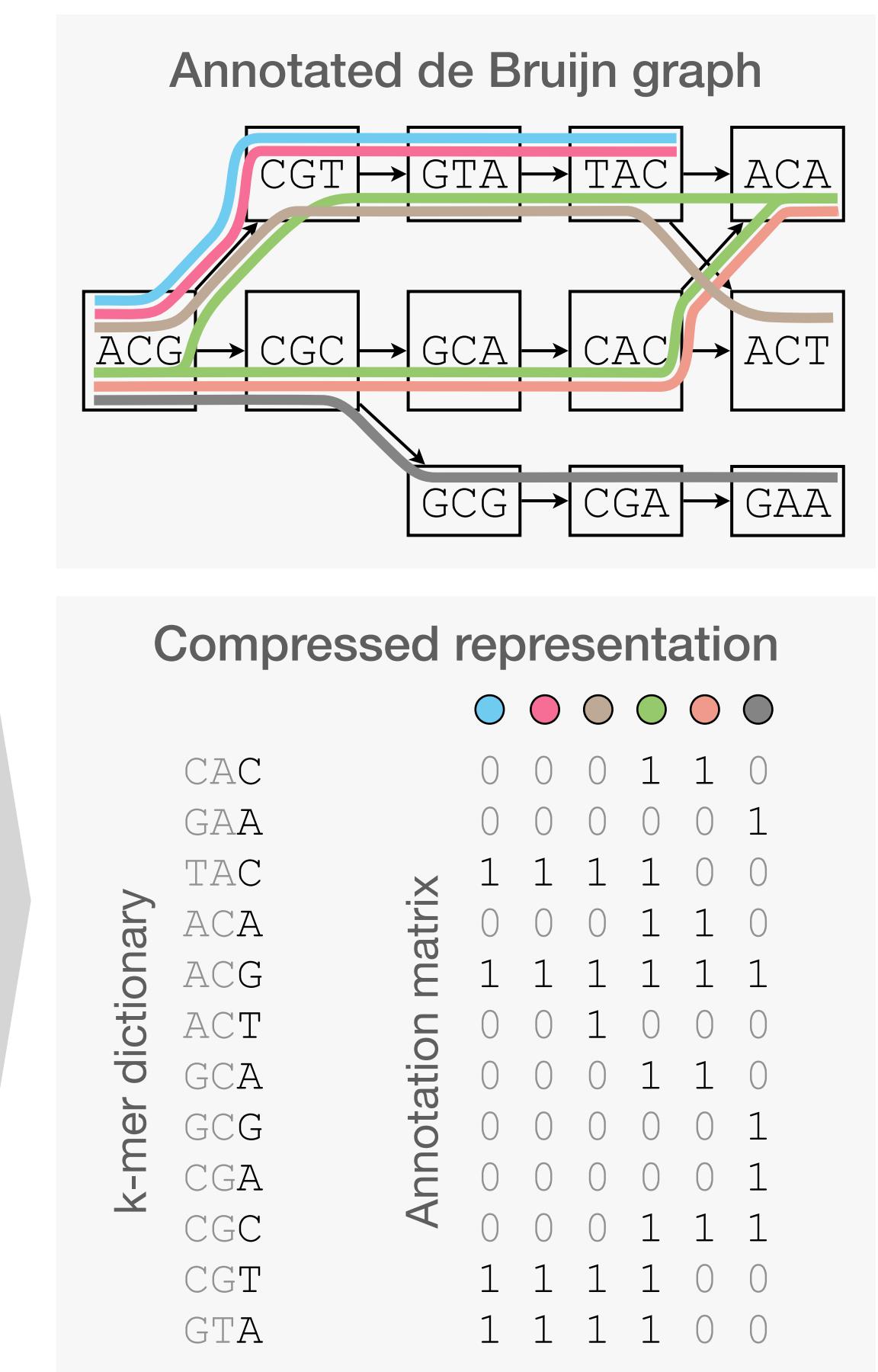
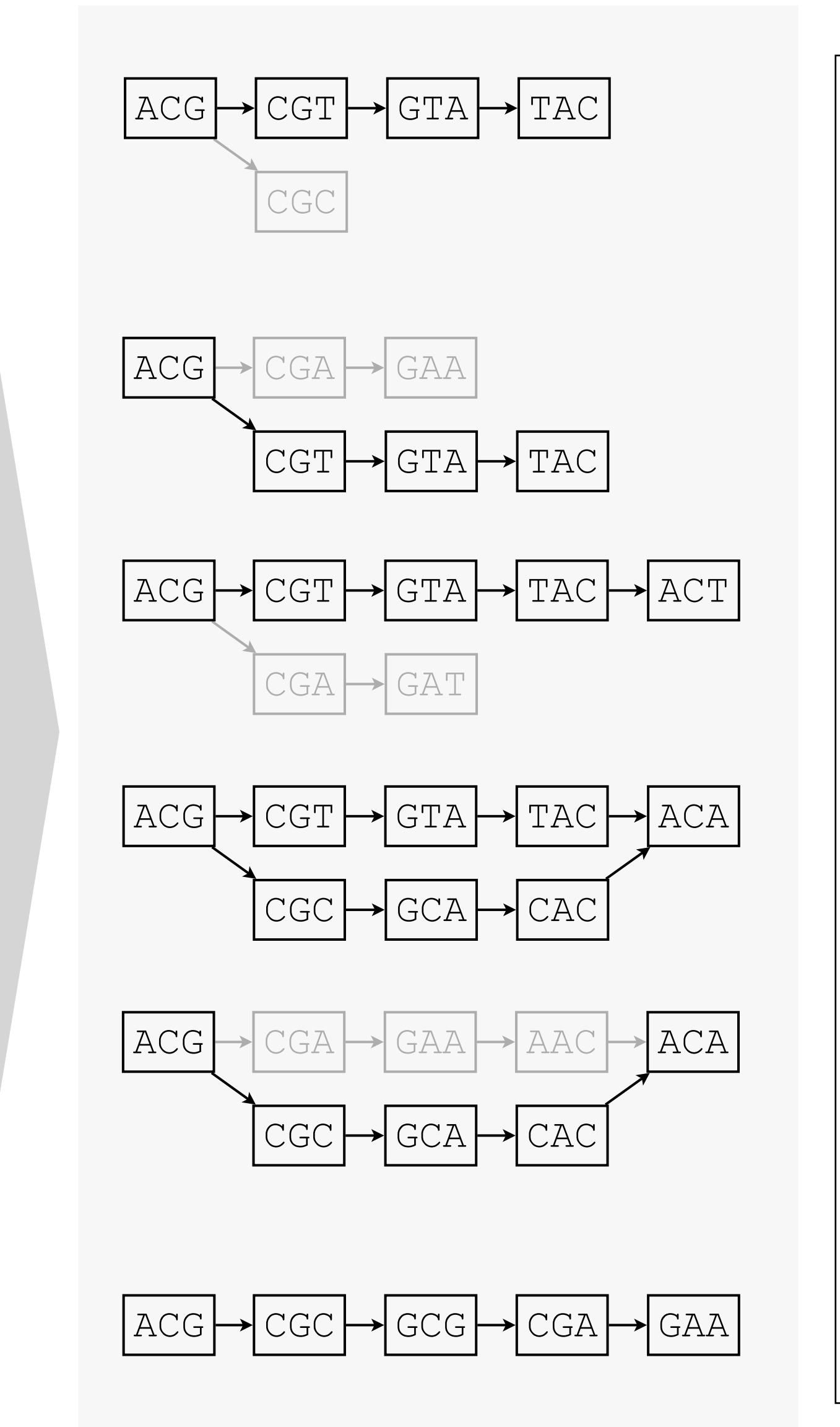
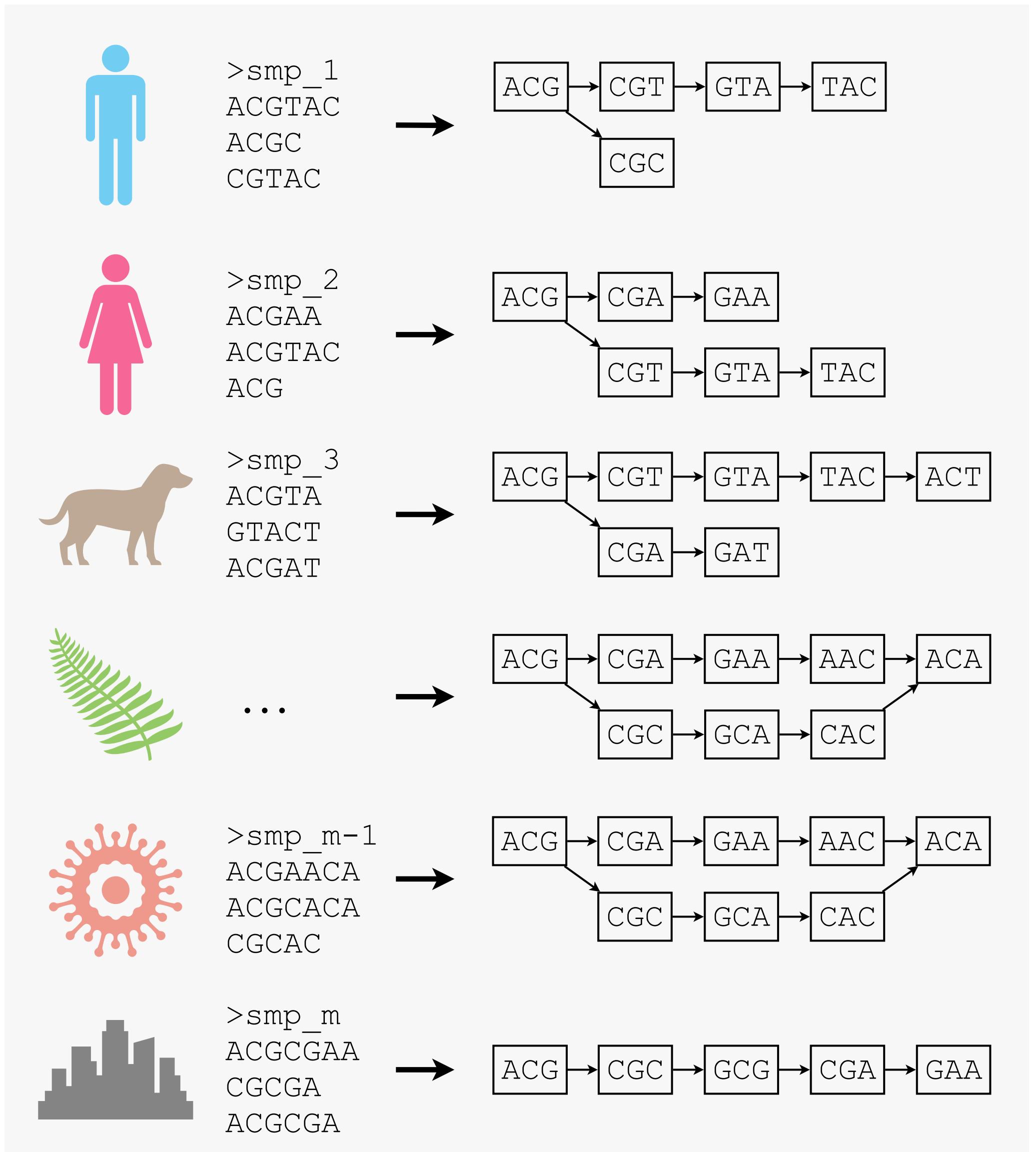
Optional graph cleaning (pop bubbles, prune tips)



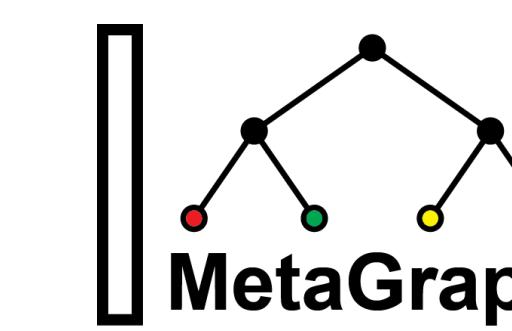
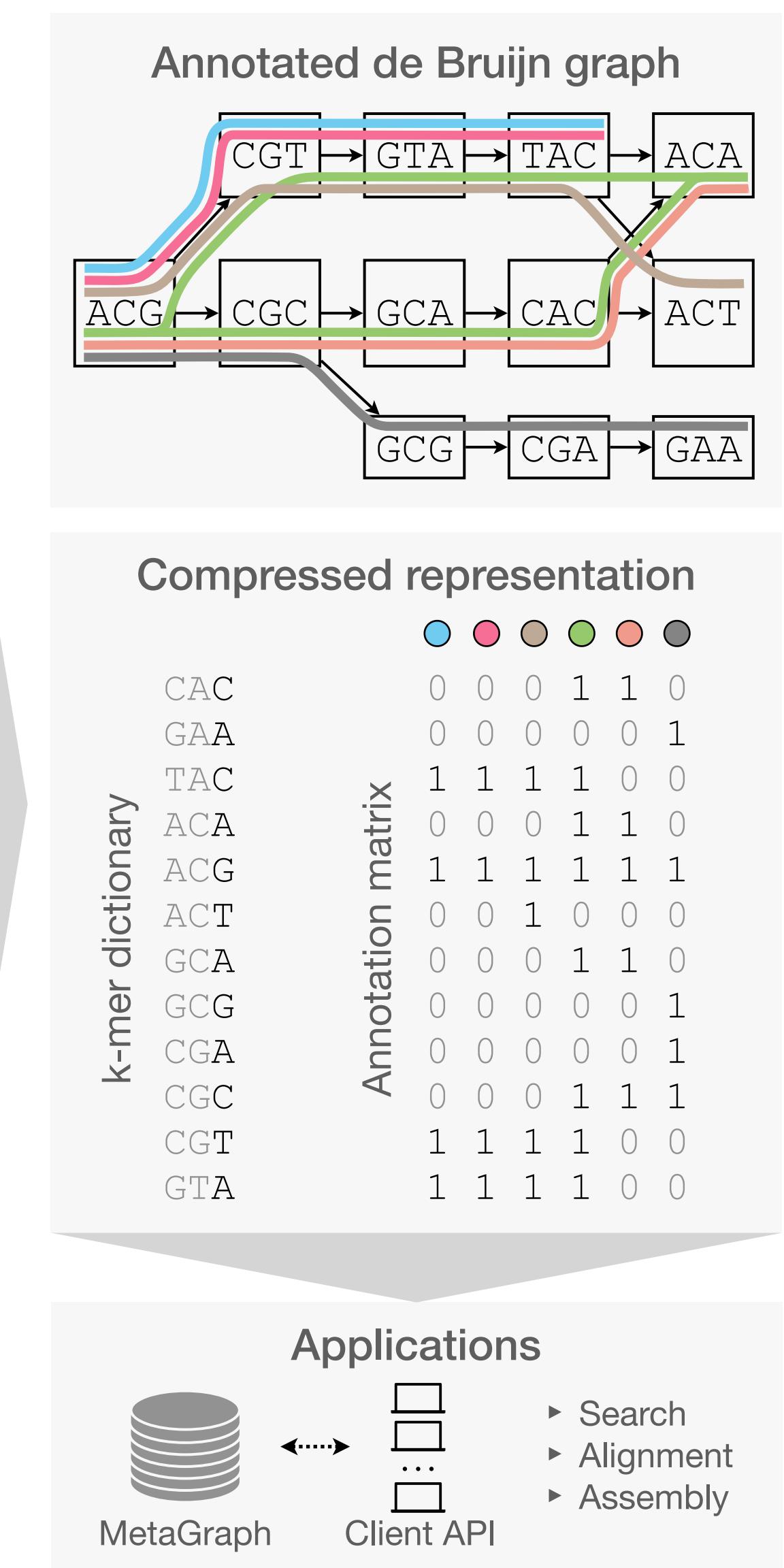
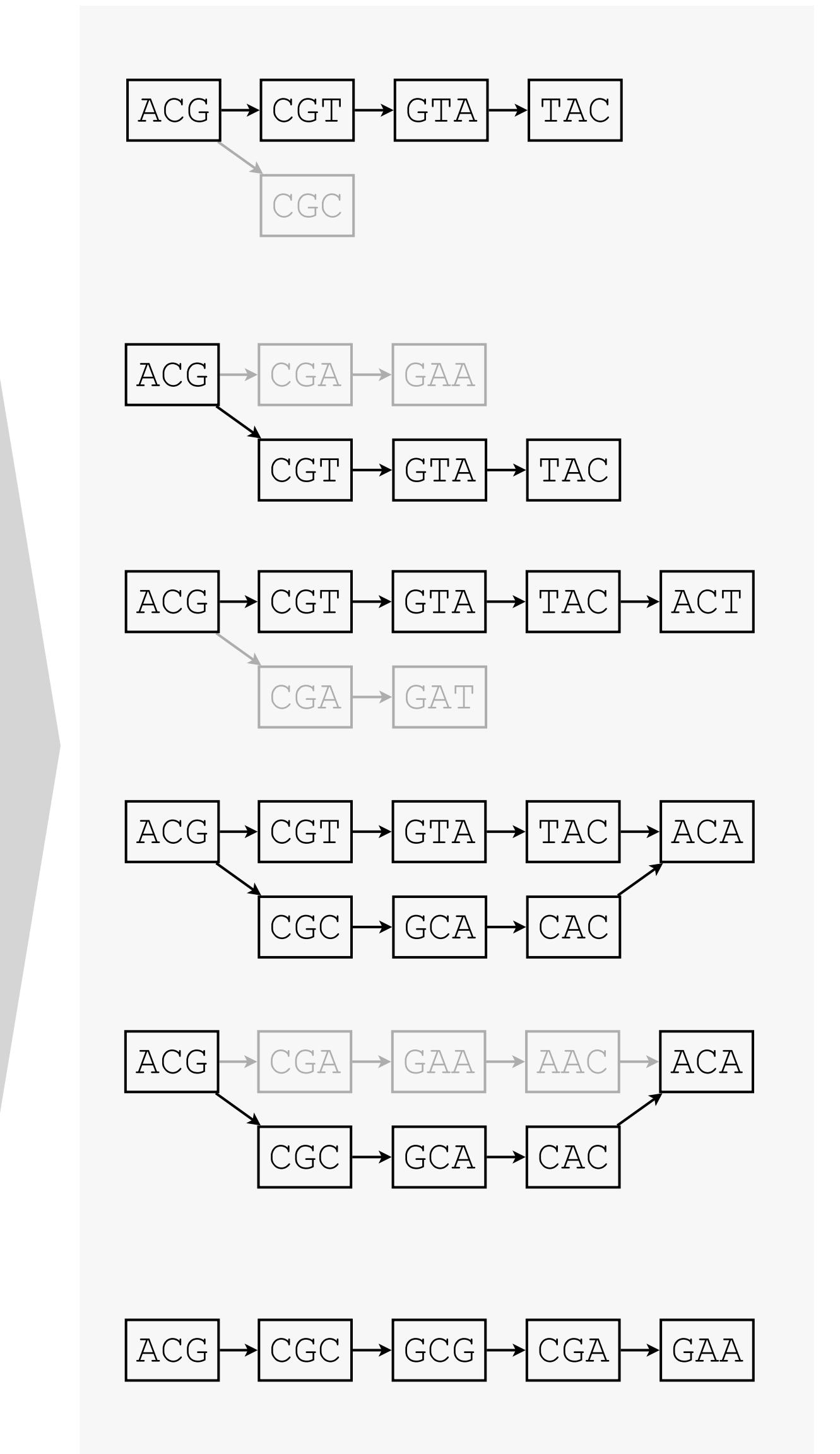
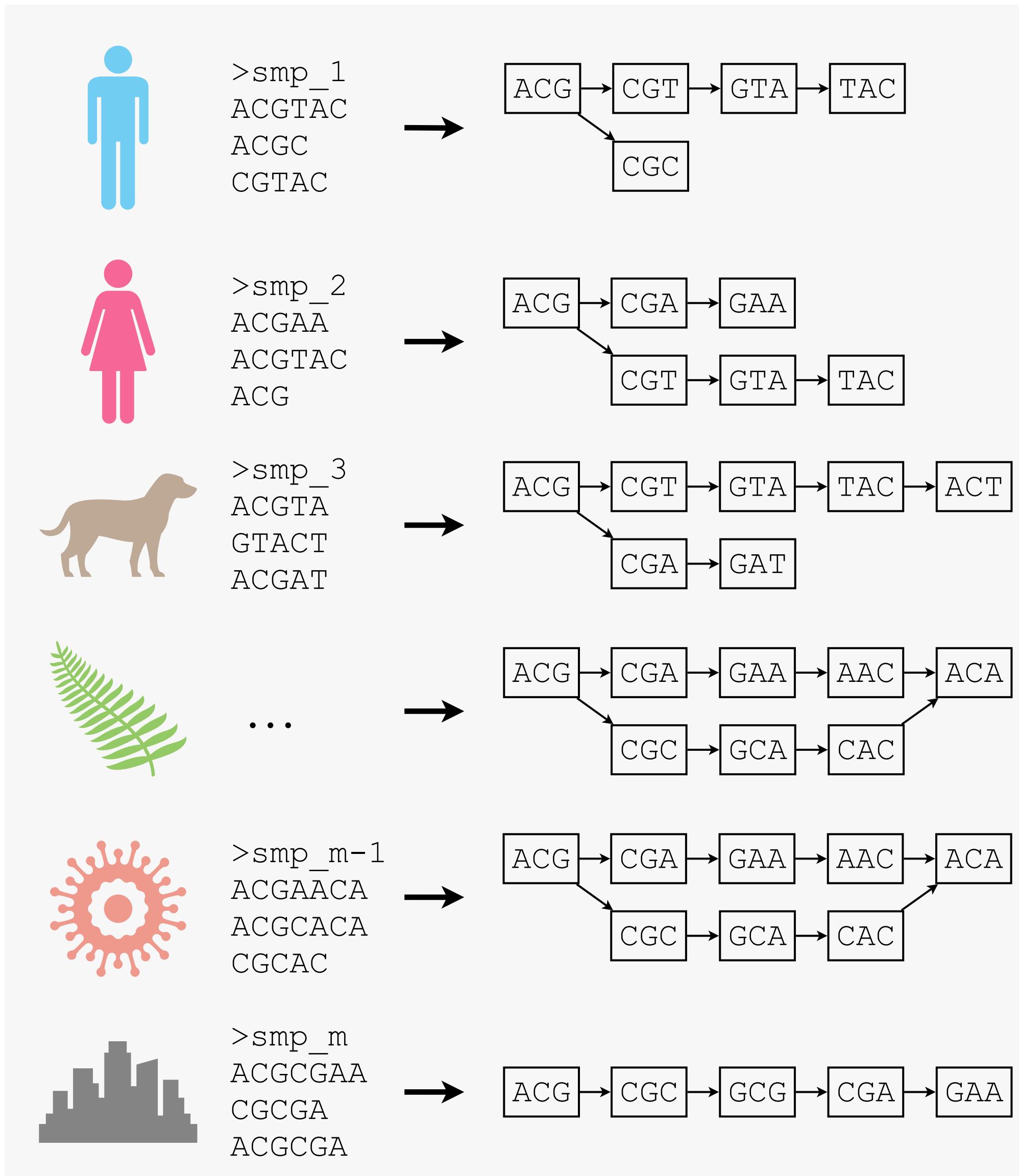
Merge into a joint graph



Indexing workflow



Indexing workflow

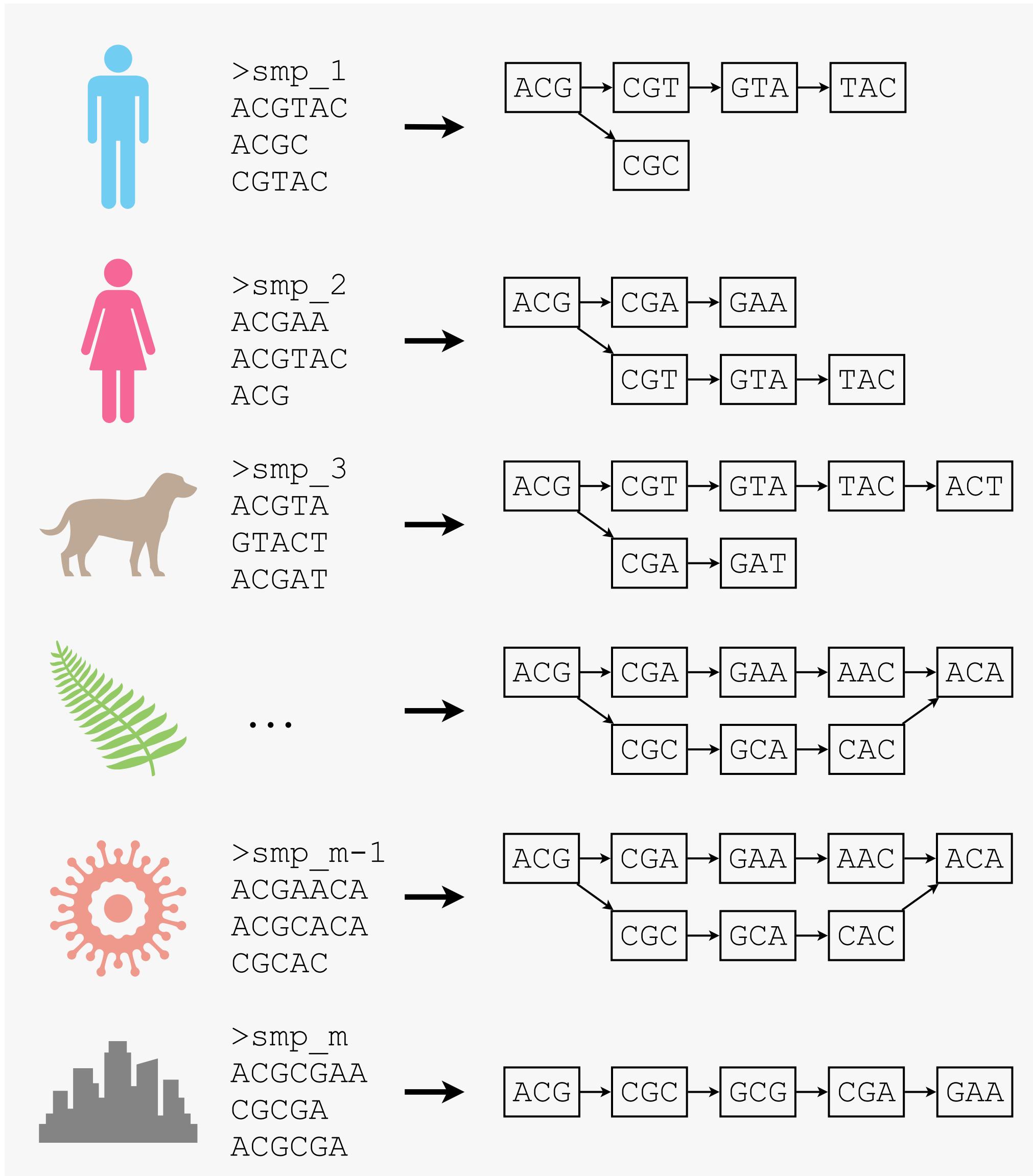


MetaGraph: Indexing and Analysing Nucleotide Archives at Petabase-scale

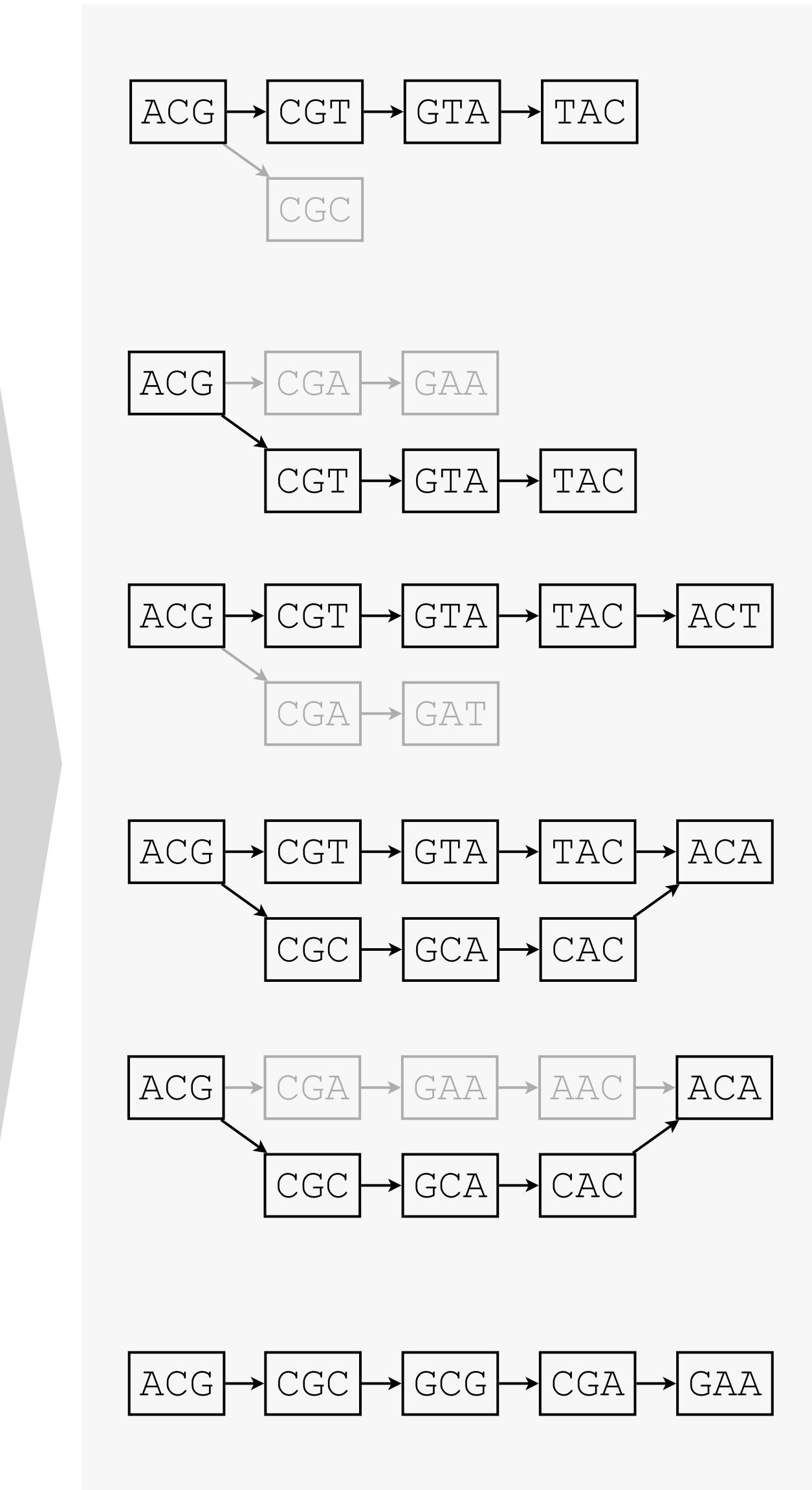
Mikhail Karasikov, Harun Mustafa, Daniel Danciu, Marc Zimmermann, Christopher Barber, Gunnar Rätsch, André Kahles

doi: <https://doi.org/10.1101/2020.10.01.322164>

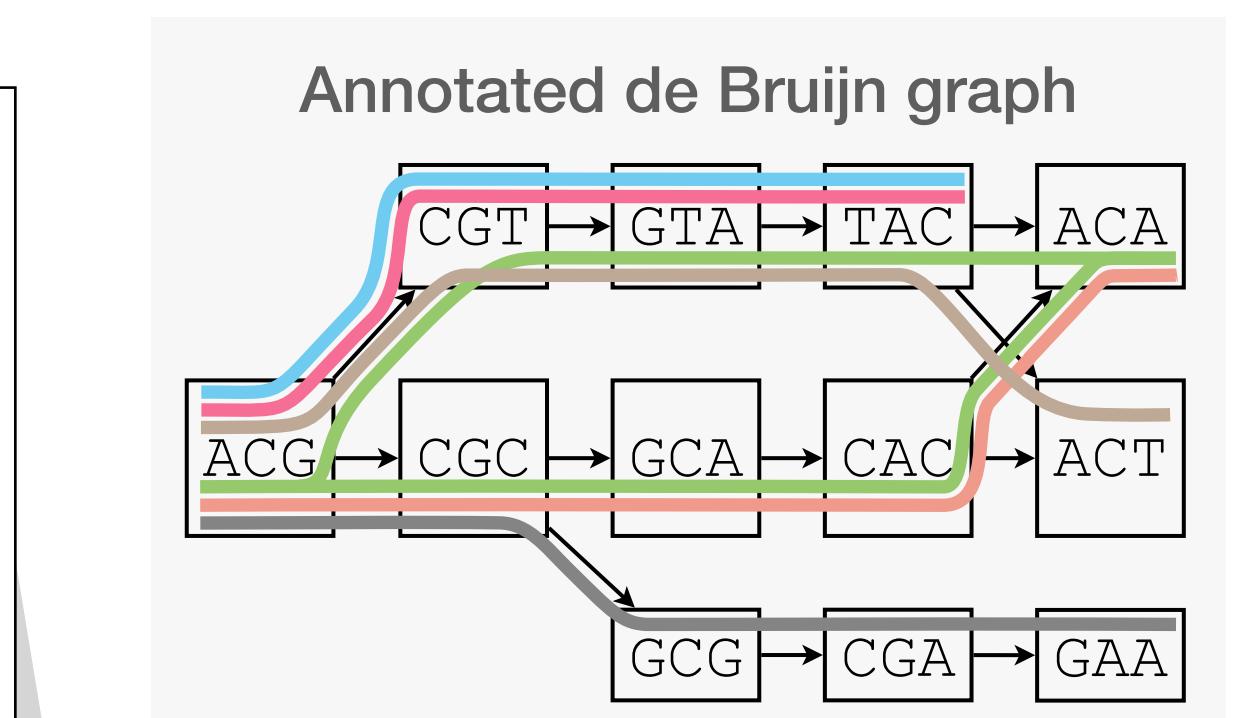
Indexing workflow



Optional graph cleaning (pop bubbles, prune tips)



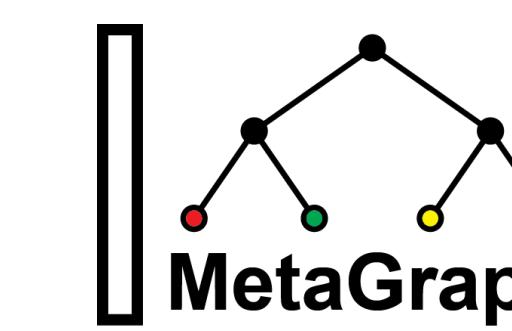
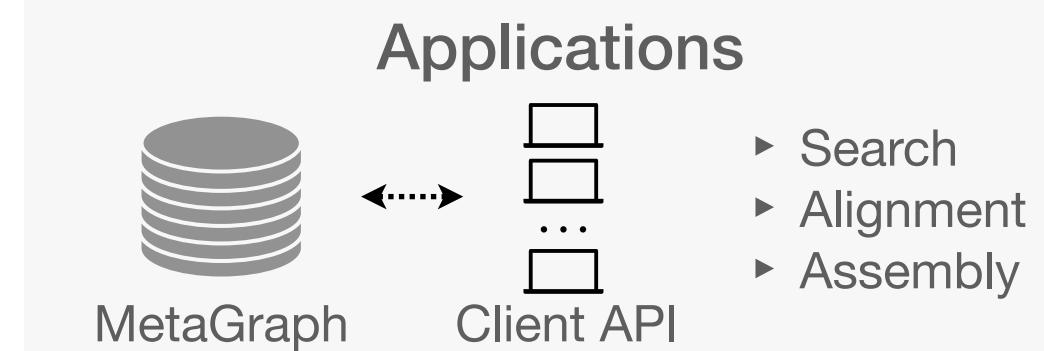
Merge into a joint graph



Compressed representation

	CAC	GAA	TAC	ACA	ACG	ACT	GCA	GCG	CGA	CGC	CGT	GTA
CAC	0	0	0	1	1	0	0	0	0	1	1	0
GAA	0	0	0	0	0	1	0	0	0	0	0	1
TAC	1	1	1	1	0	0	1	0	0	0	0	0
ACA	0	0	0	1	1	0	0	0	1	1	0	0
ACG	1	1	1	1	1	1	0	0	1	1	1	1
ACT	0	0	1	0	0	0	1	0	0	0	0	0
GCA	0	0	0	1	1	0	0	1	1	0	0	0
GCG	0	0	0	0	0	1	0	0	0	0	1	0
CGA	0	0	0	0	0	0	1	0	0	0	1	0
CGC	0	0	0	1	1	0	0	0	0	1	1	1
CGT	1	1	1	1	0	0	0	1	1	1	1	0
GTA	1	1	1	1	0	0	0	0	1	1	1	0

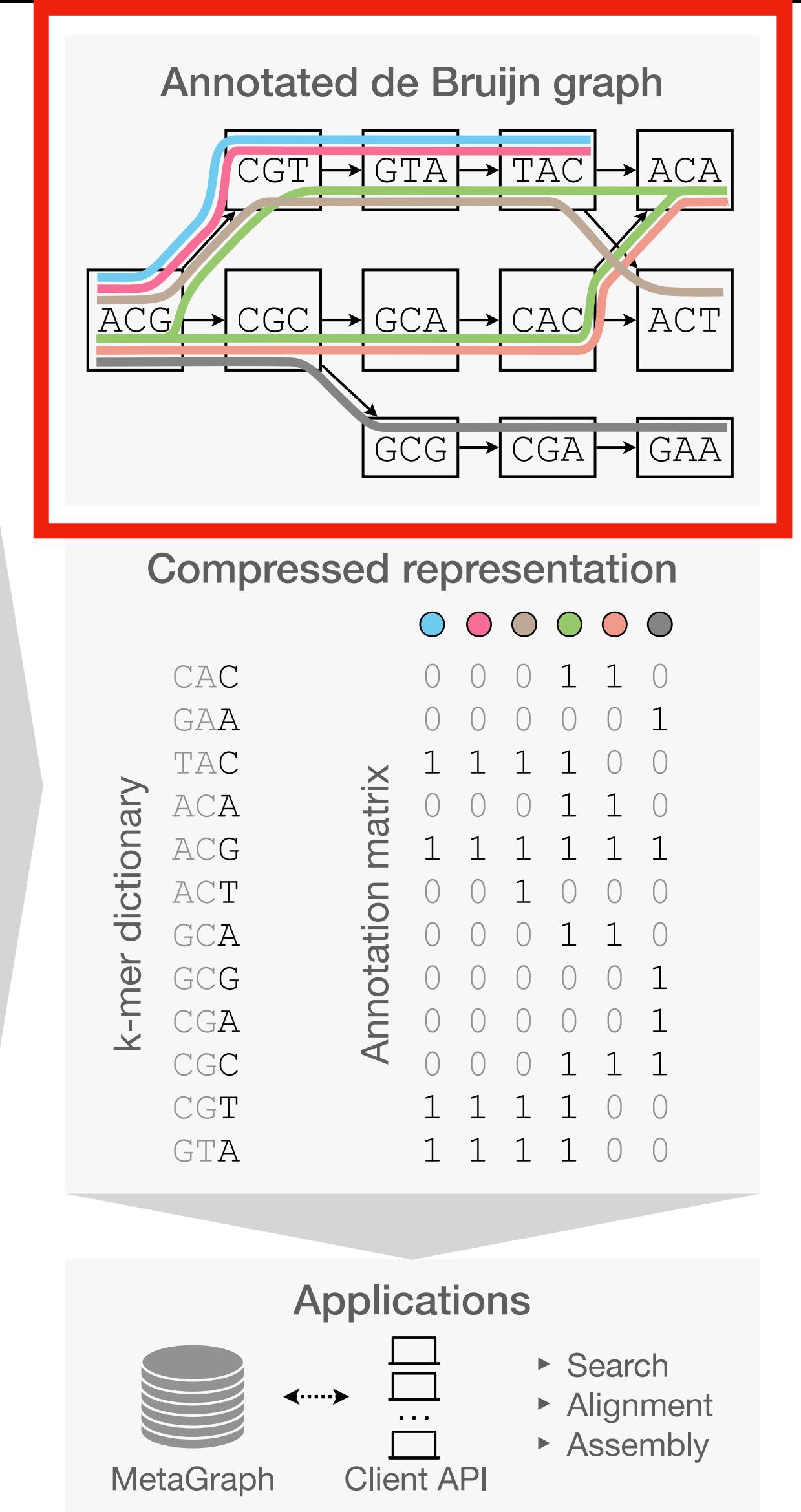
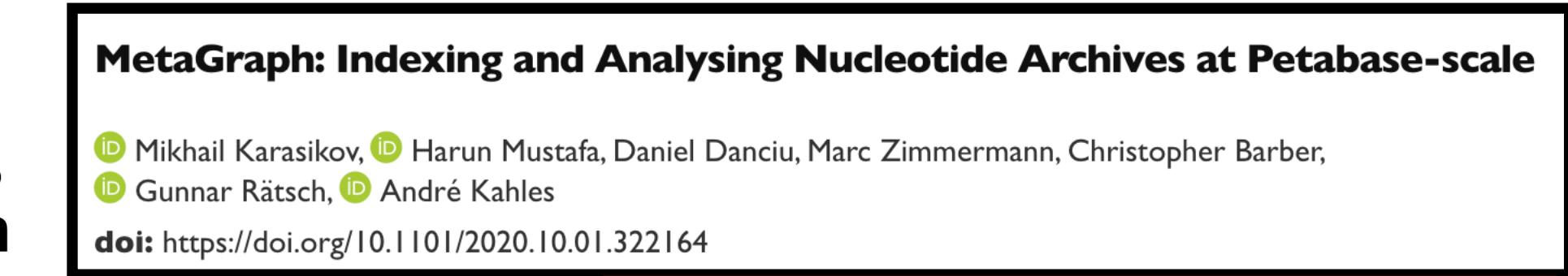
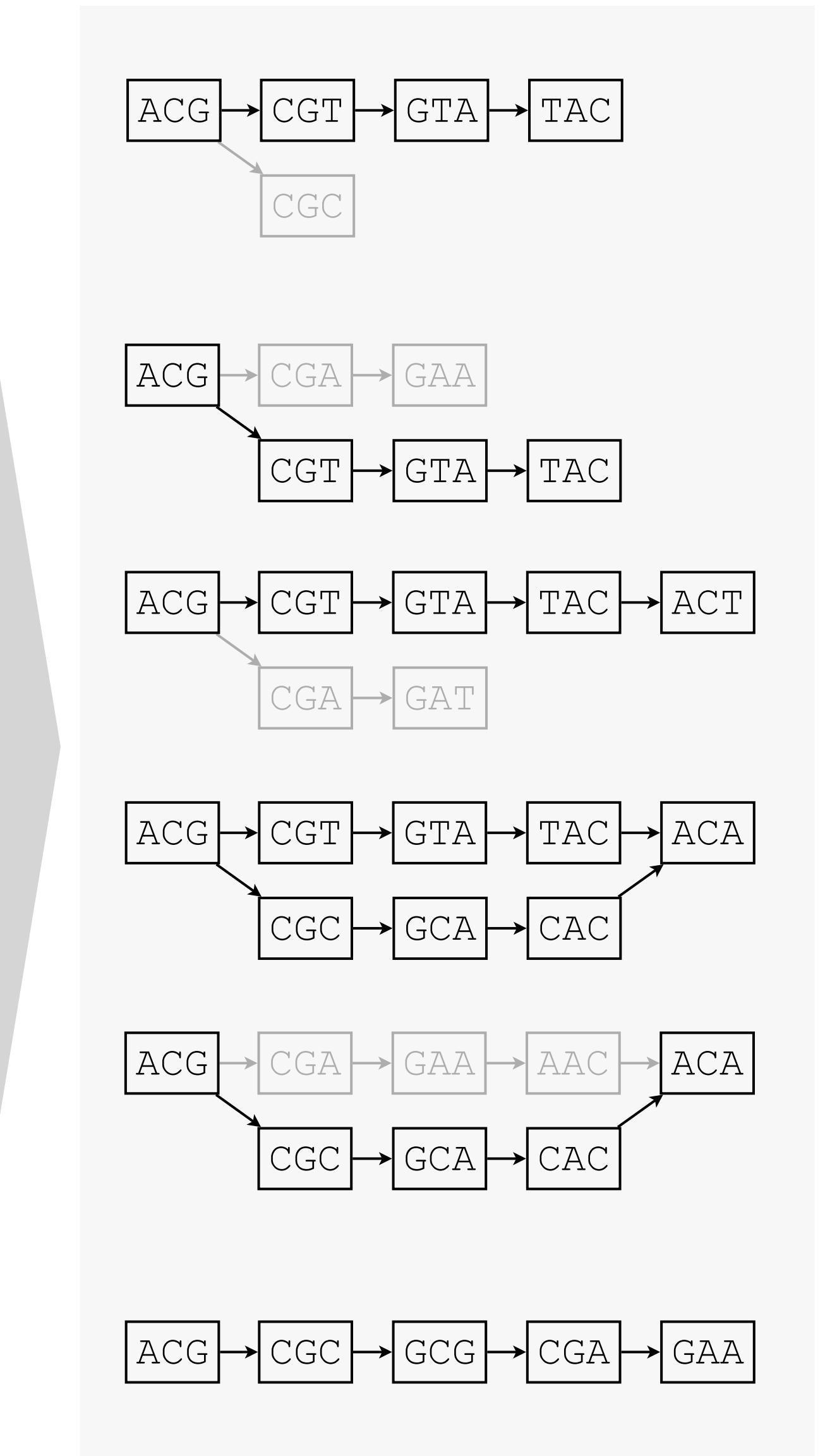
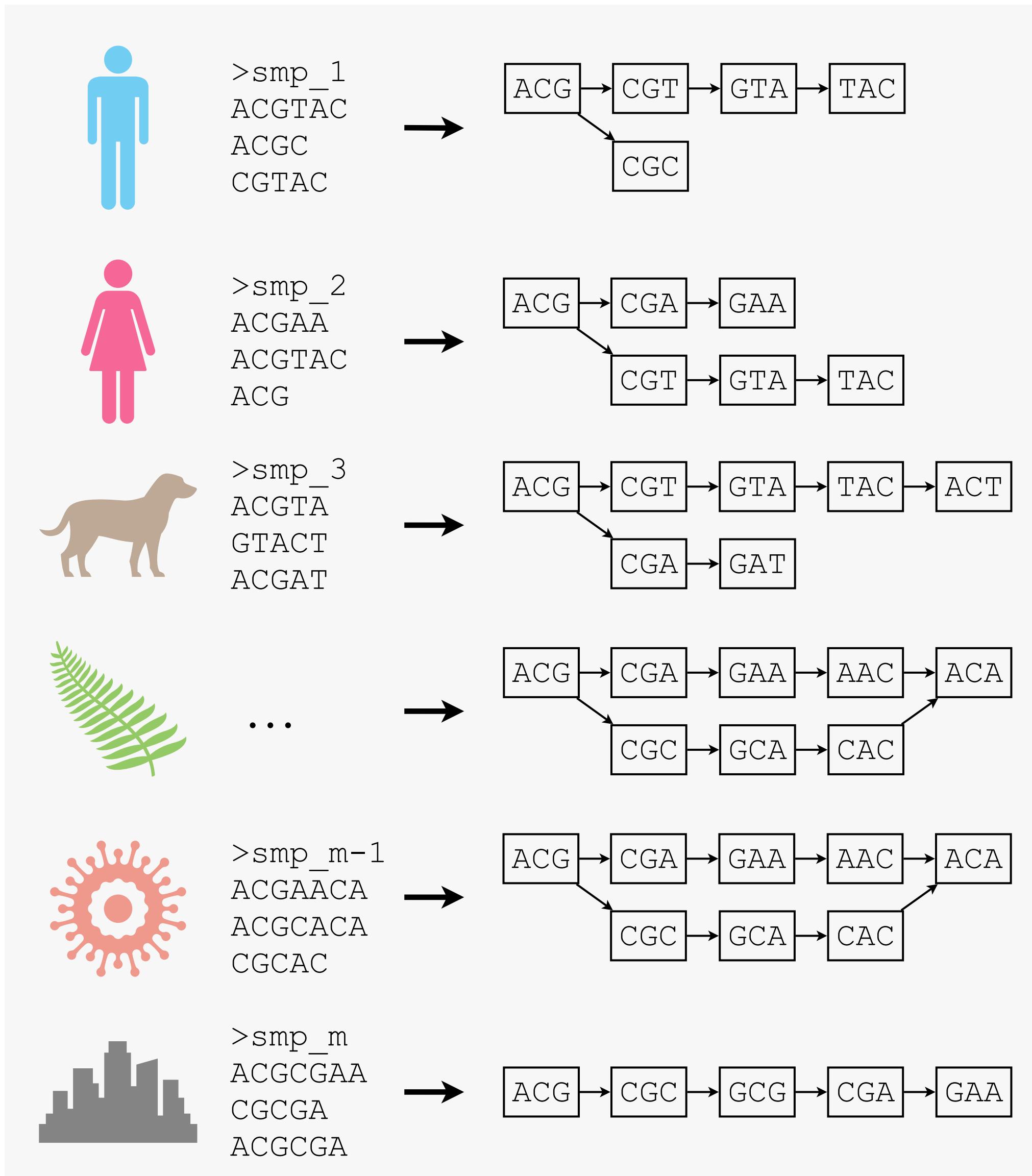
Annotation matrix



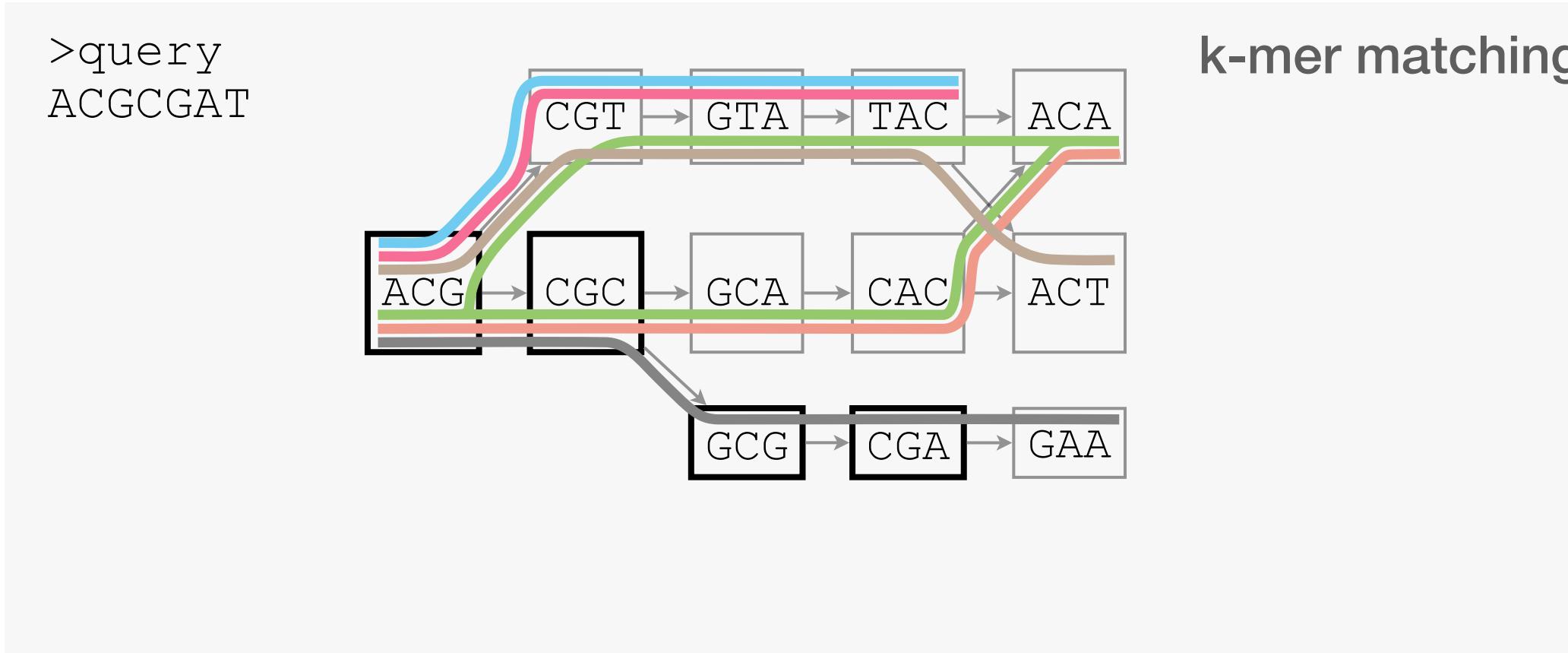
MetaGraph: Indexing and Analysing Nucleotide Archives at Petabase-scale

Mikhail Karasikov, Harun Mustafa, Daniel Danciu, Marc Zimmermann, Christopher Barber, Gunnar Rätsch, André Kahles
doi: <https://doi.org/10.1101/2020.10.01.322164>

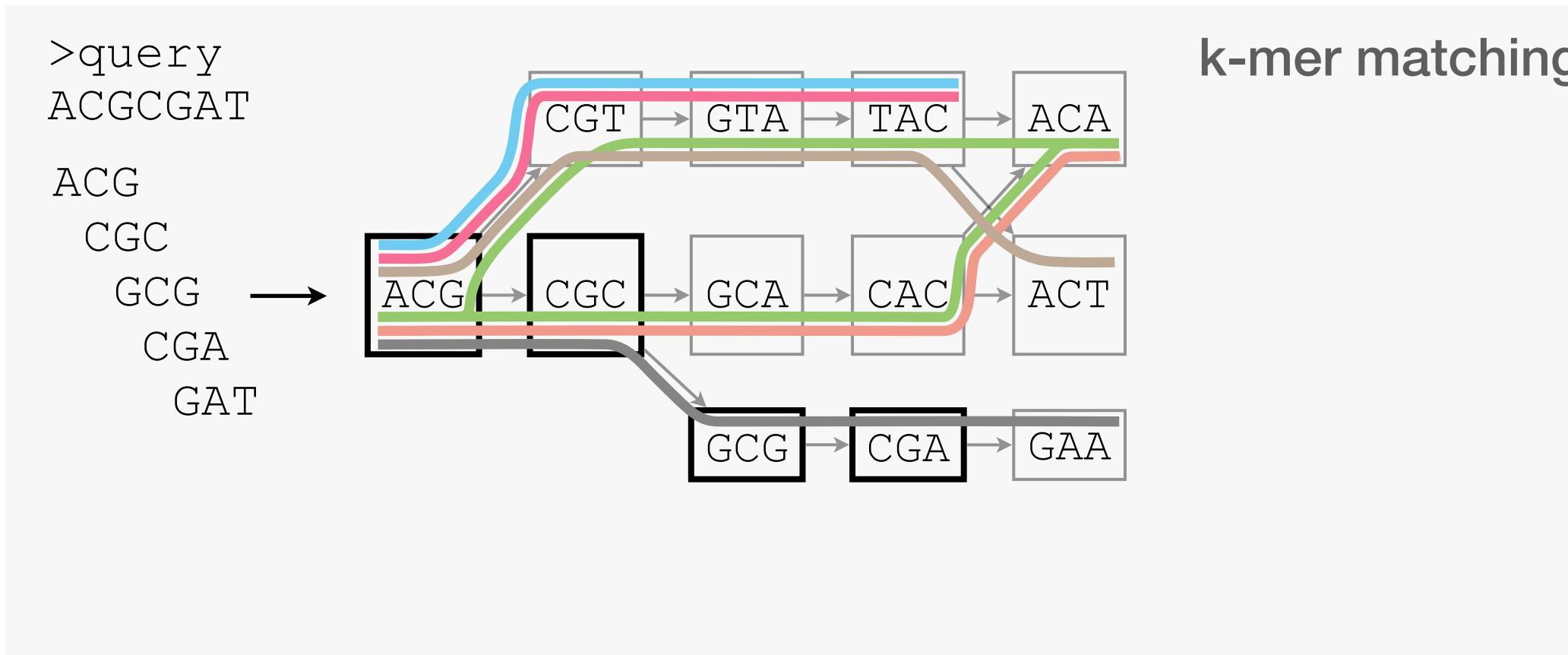
Indexing workflow



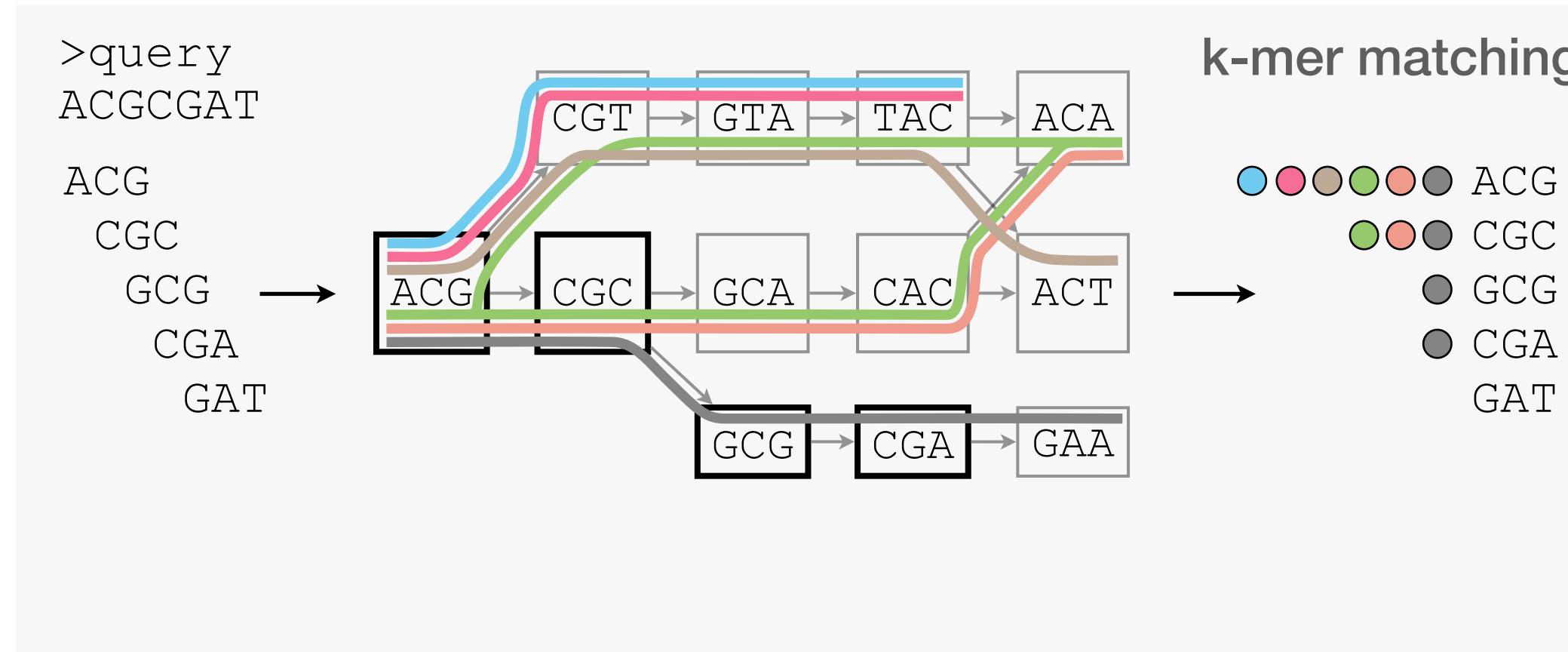
Sequence search



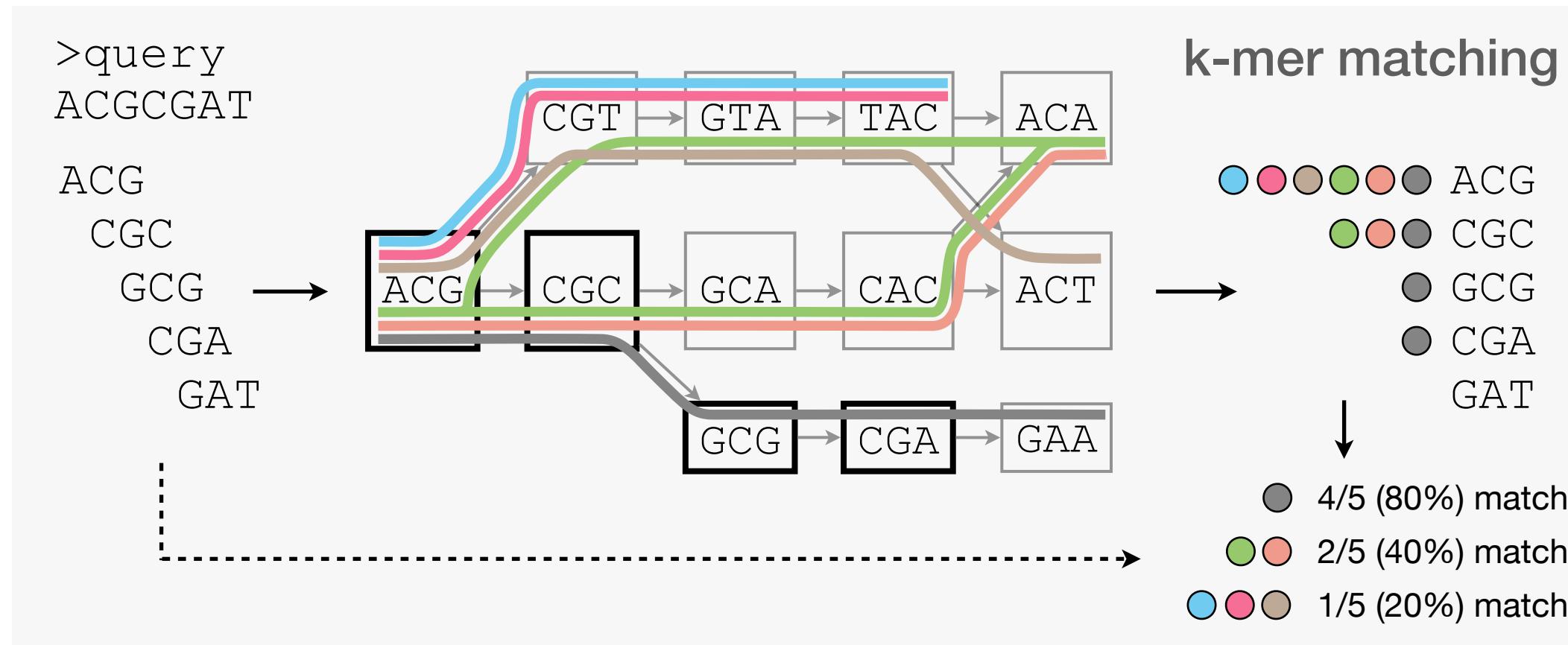
Sequence search



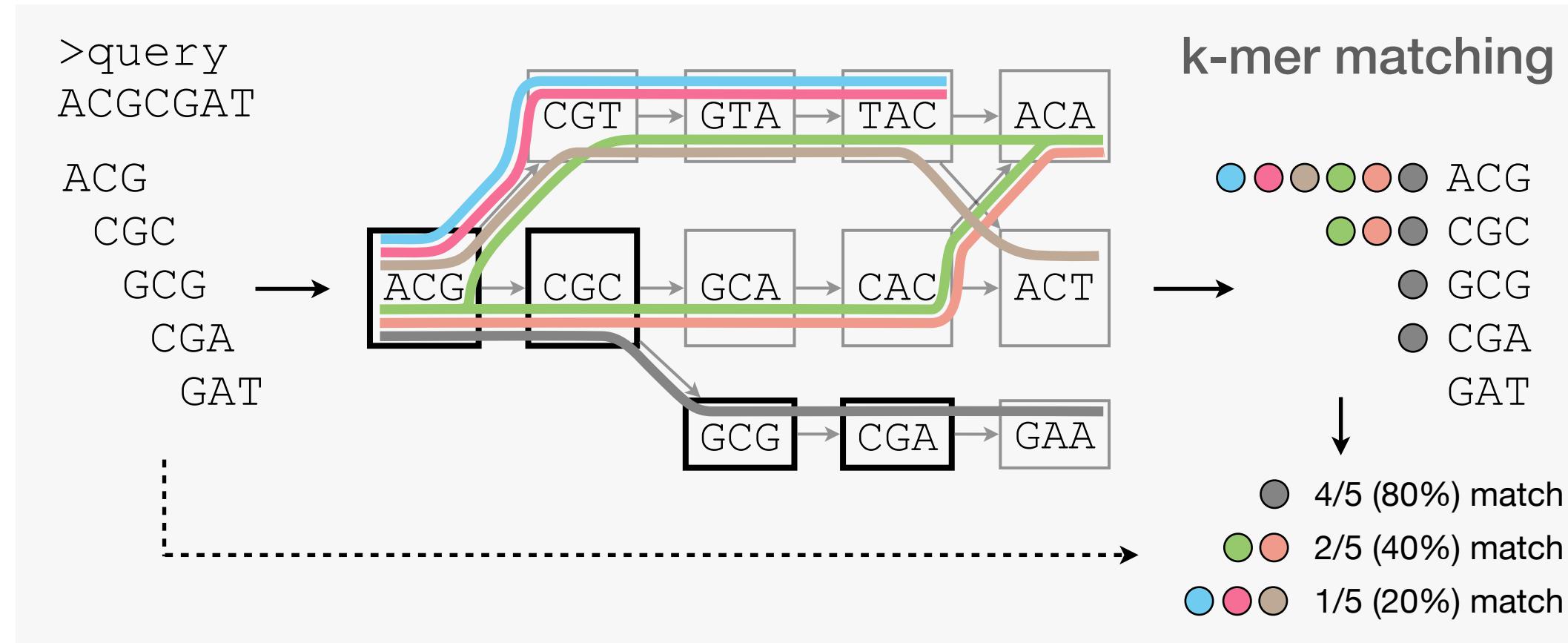
Sequence search



Sequence search

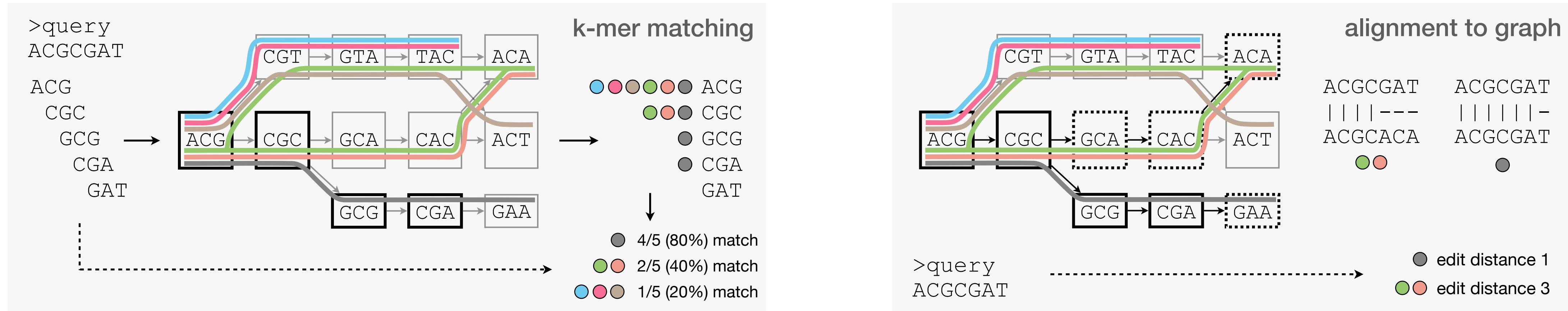


Sequence search



- SBT [Solomon, Kingsford, 2016]
- VARI [Muggli *et al.*, 2017]
- Mantis [Pandey *et al.*, 2018]

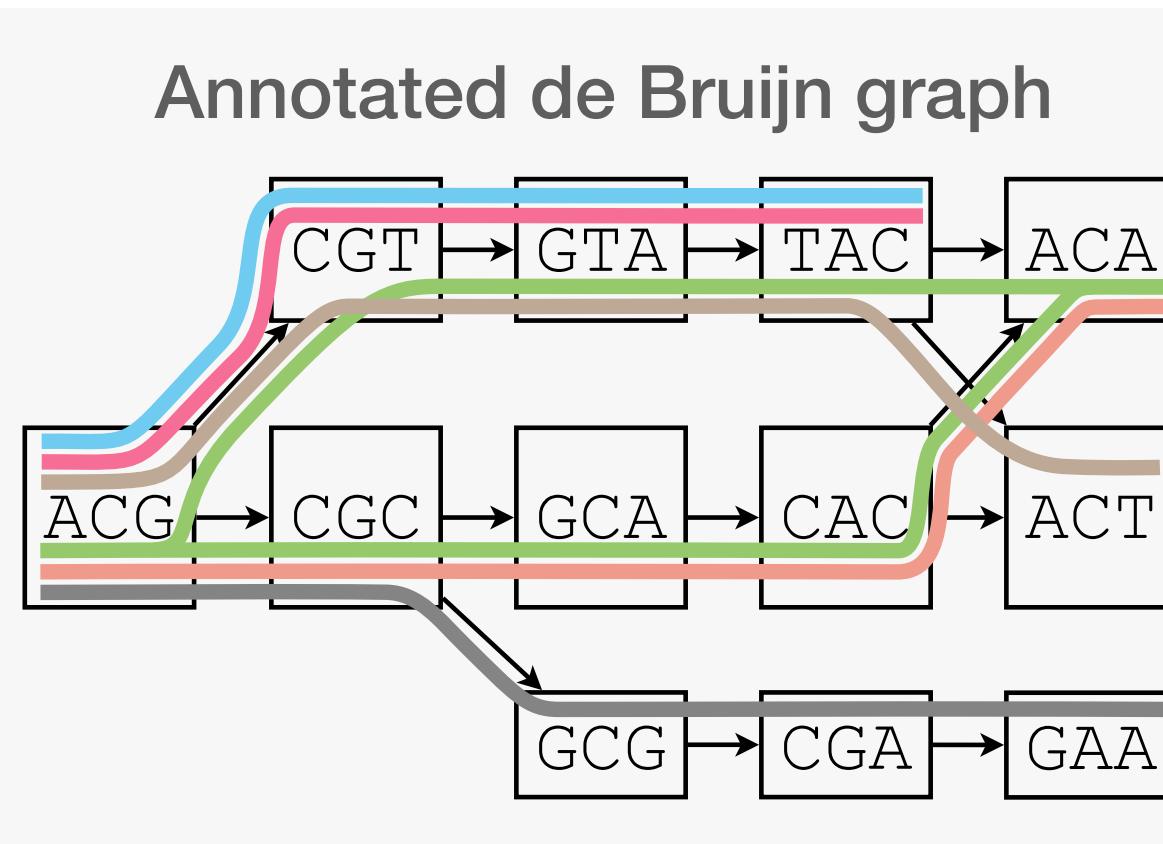
Sequence search



- SBT [Solomon, Kingsford, 2016]
- VARI [Muggli et al., 2017]
- Mantis [Pandey et al., 2018]

- [Lee, Grasso, Sharlow, 2002]
- deBGA [Liu, Guo, Brudno, Wang, 2016]
- SPAAligner [Dvorkina et al., 2020]
- AStarix [Ivanov et al., 2020]
- [Schulz et al., 2021]
- MetaGraph-MLA [Mustafa et al., 2022]

Representing annotated De Bruijn graphs



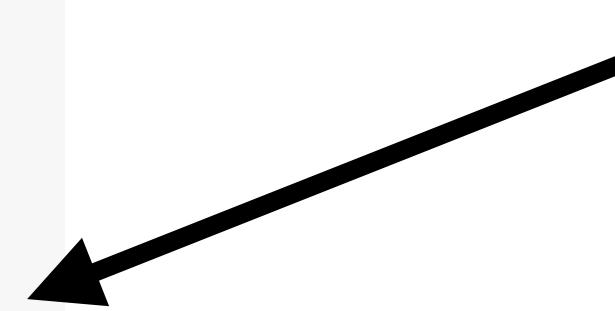
Compressed representation

K-mer dictionary

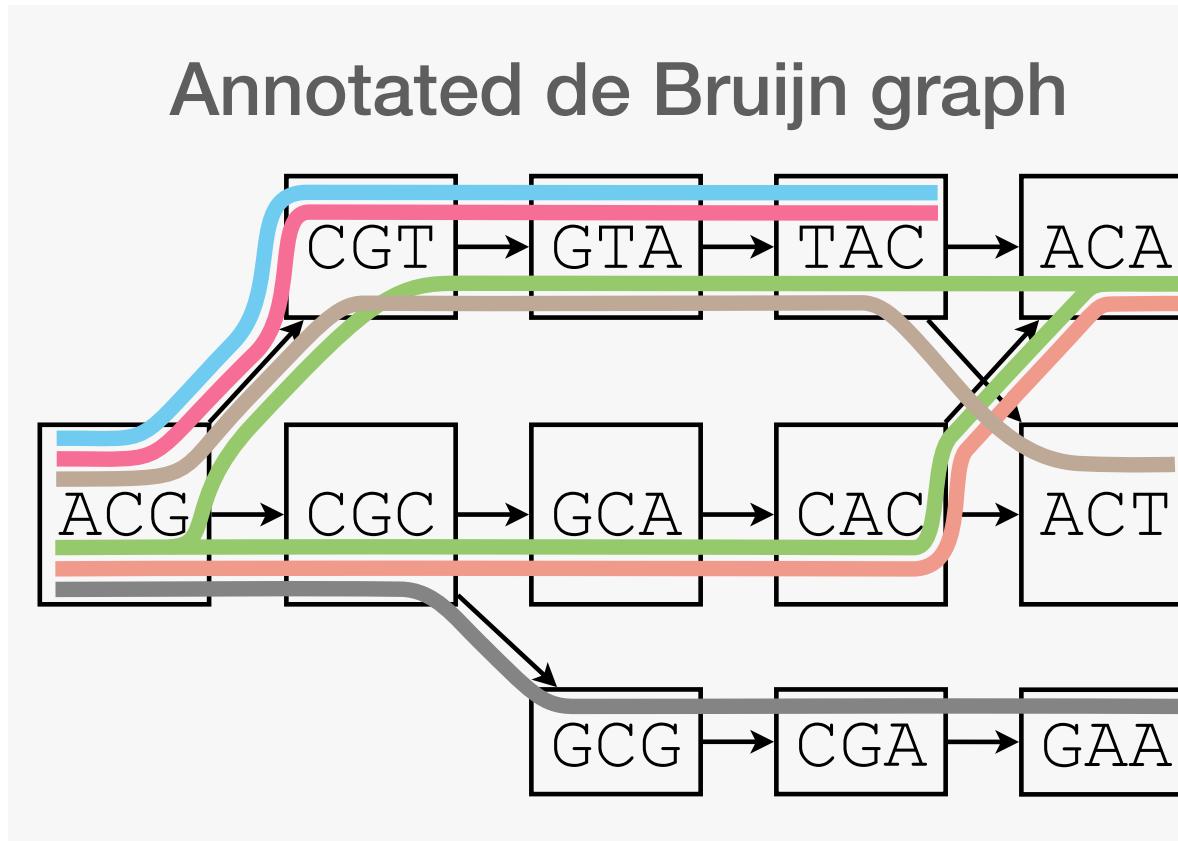
Annotation matrix

	CAC	GAA	TAC	ACA	ACG	ACT	GCA	GCG	CGA	CGC	CGT	GTA
CAC	0	0	0	1	1	0	0	0	0	0	0	0
GAA	0	0	0	0	0	0	1	0	0	0	0	0
TAC	1	1	1	1	0	0	0	0	0	0	0	0
ACA	0	0	0	1	1	0	0	0	0	0	0	0
ACG	1	1	1	1	1	1	0	0	0	0	0	0
ACT	0	0	1	0	0	0	0	0	0	0	0	0
GCA	0	0	0	1	1	0	0	0	0	0	0	0
GCG	0	0	0	0	0	1	0	0	0	0	0	0
CGA	0	0	0	0	0	0	1	0	0	0	0	0
CGC	0	0	0	1	1	1	0	0	0	0	0	0
CGT	1	1	1	1	0	0	0	0	0	0	0	0
GTA	1	1	1	1	0	0	0	0	0	0	0	0

Must be efficiently **queryable**



Representing annotated De Bruijn graphs



Compressed representation

k-mer dictionary

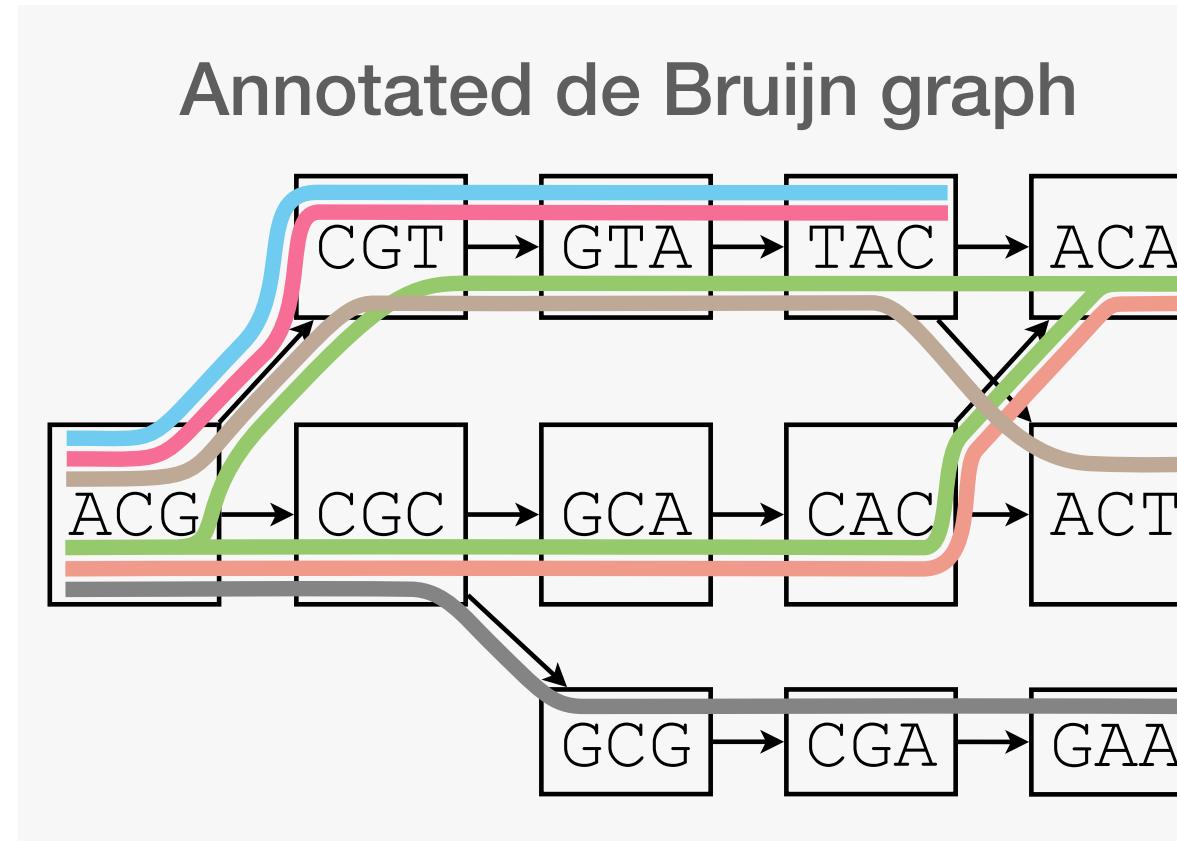
	●	●	●	●	●	●
CAC	0	0	0	1	1	0
GAA	0	0	0	0	0	1
TAC	1	1	1	1	0	0
ACA	0	0	0	1	1	0
ACG	1	1	1	1	1	1
ACT	0	0	1	0	0	0
GCA	0	0	0	1	1	0
GCG	0	0	0	0	0	1
CGA	0	0	0	0	0	1
CGC	0	0	0	1	1	1
CGT	1	1	1	1	0	0
GTA	1	1	1	1	0	0

Annotation matrix

Representing graph

- ▶ Hash table
- ▶ Indicator set bit vector [Conway et al., 2011]
- ▶ BOSS table [Bowe et al., 2012]

Representing annotated De Bruijn graphs



Compressed representation

K-mer dictionary

CAC
GAA
TAC
ACA
ACG
ACT
GCA
GCG
CGA
CGC
CGT
GTA

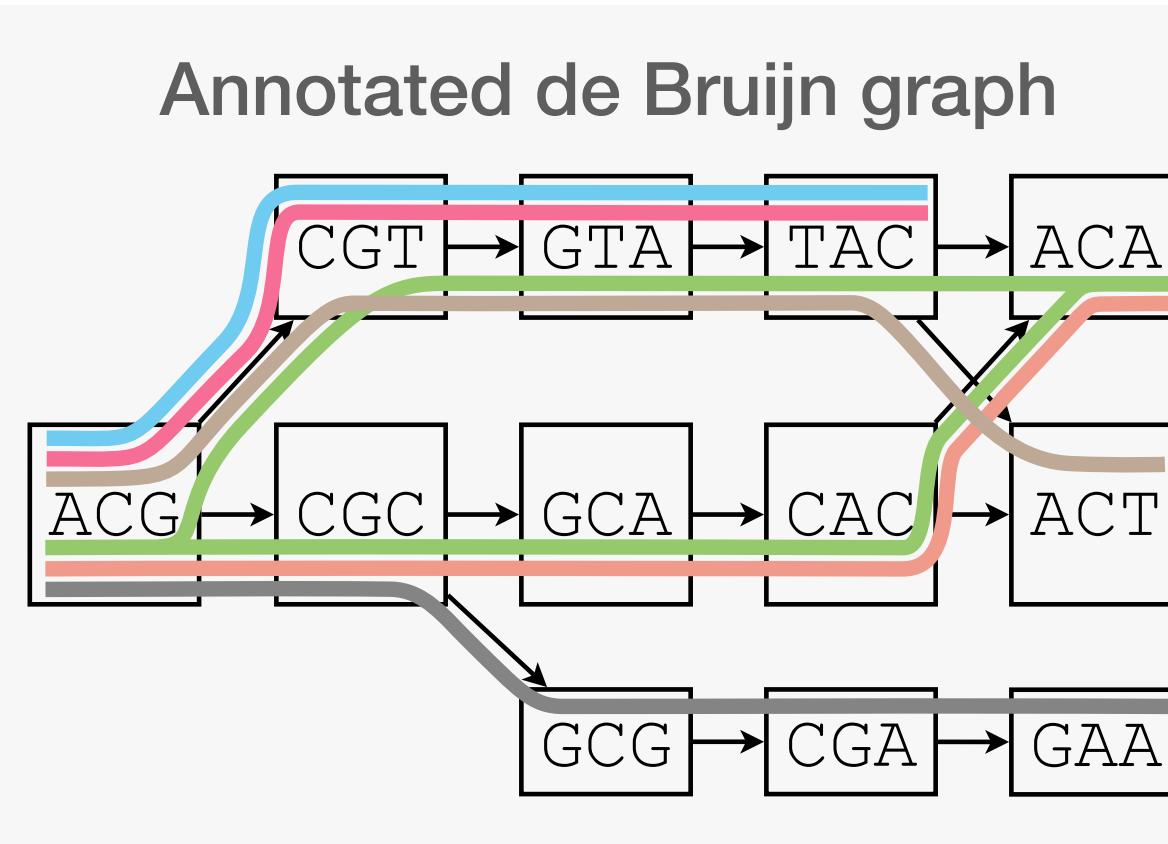
Annotation matrix

	0	0	0	1	1	0
CAC	0	0	0	0	0	1
GAA	0	0	0	0	0	0
TAC	1	1	1	1	0	0
ACA	0	0	0	1	1	0
ACG	1	1	1	1	1	1
ACT	0	0	1	0	0	0
GCA	0	0	0	1	1	0
GCG	0	0	0	0	0	1
CGA	0	0	0	0	0	1
CGC	0	0	0	1	1	1
CGT	1	1	1	1	0	0
GTA	1	1	1	1	0	0

Representing graph

- ▶ Hash table
- ▶ Indicator set bit vector [Conway *et al.*, 2011]
- ▶ BOSS table [Bowe *et al.*, 2012]
 - ✓ ~2–3 bits per k-mer for DNA
 - ✓ supports search of sub-k-mers

Representing annotated De Bruijn graphs



Compressed representation

k-mer dictionary

	CAC	GAA	TAC	ACA	ACG	ACT	GCA	GCG	CGA	CGC	CGT	GTA
Annotation matrix	0	0	0	1	1	0	0	0	0	0	1	1
CAC	0	0	0	0	0	1	0	0	0	0	0	0
GAA	0	0	0	0	0	0	1	0	0	0	0	0
TAC	1	1	1	1	0	0	0	0	0	0	0	0
ACA	0	0	0	1	1	0	0	0	0	0	0	0
ACG	1	1	1	1	1	1	0	0	0	0	0	0
ACT	0	0	1	0	0	0	0	0	0	0	0	0
GCA	0	0	0	1	1	0	0	0	0	0	0	0
GCG	0	0	0	0	0	1	0	0	0	0	0	0
CGA	0	0	0	0	0	0	1	0	0	0	0	0
CGC	0	0	0	1	1	1	0	0	0	0	0	0
CGT	1	1	1	1	0	0	0	0	0	0	0	0
GTA	1	1	1	1	0	0	0	0	0	0	0	0

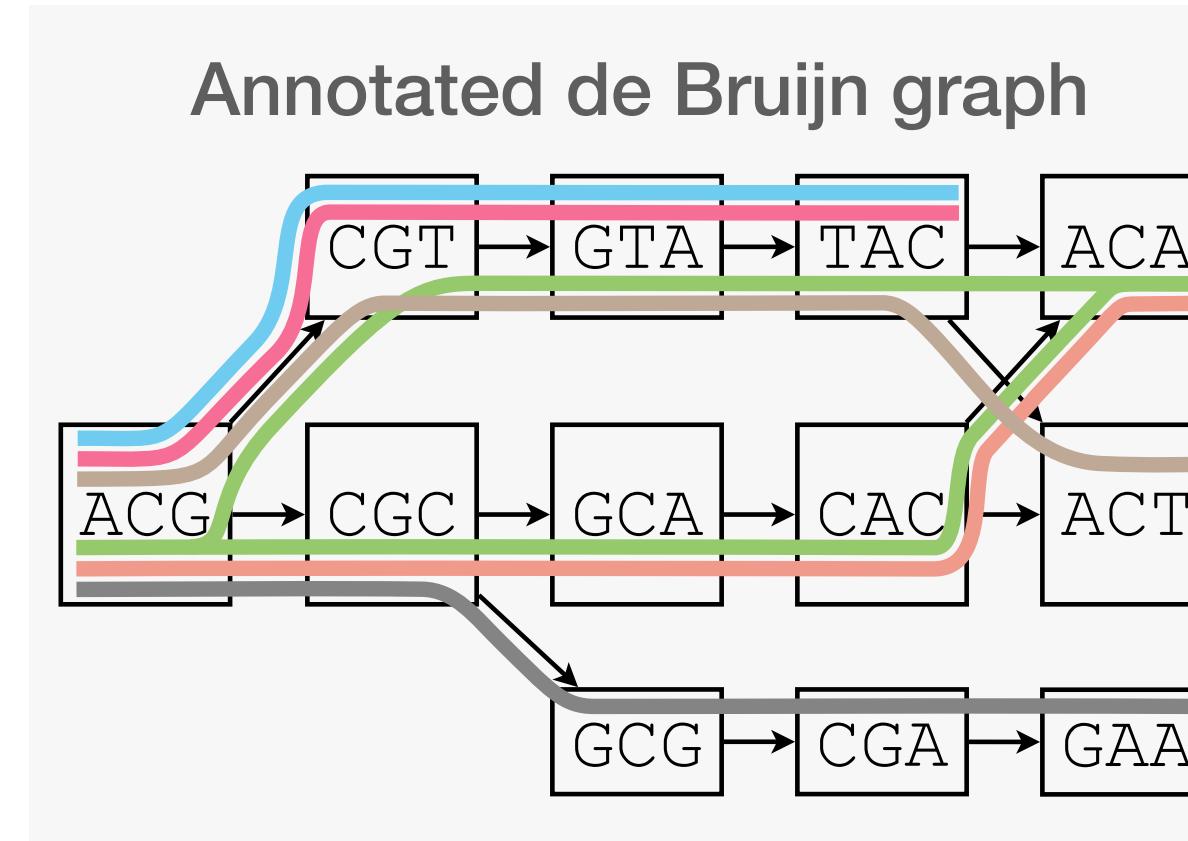
Representing graph

- ▶ Hash table
- ▶ Indicator set bit vector [Conway *et al.*, 2011]
- ▶ BOSS table [Bowe *et al.*, 2012]
 - ✓ ~2–3 bits per k-mer for DNA
 - ✓ supports search of sub-k-mers

Representing annotation

- ▶ Column-major sparse representation
- ▶ RowFlat (employed in VARI [Muggli *et al.*, 2017])
- ▶ Rainbowfish [Almodaresi *et al.*, 2017]
- ▶ Rainbow-MST [Almodaresi *et al.*, 2019]

Representing annotated De Bruijn graphs



Compressed representation

The table shows a compressed representation of the annotated de Bruijn graph. The left column lists the k-mers from the graph: CAC, GAA, TAC, ACA, ACG, ACT, GCA, GCG, CGA, CGC, CGT, and GTA. The top row shows the first few columns of the annotation matrix, with colored circles above each column corresponding to the colors in the original graph: blue, pink, brown, green, red, and grey. The matrix itself is a 12x6 grid of binary values (0 or 1). A red box highlights a 3x3 submatrix in the middle of the matrix. Below the matrix, the value $\sim 10^{11}$ is given, indicating the size of the matrix. At the bottom, the value $\sim 10^6$ is given, likely referring to the size of the k-mer dictionary.

	0	1	2	3	4	5
CAC	0	0	0	1	1	0
GAA	0	0	0	0	0	1
TAC	1	1	1	1	0	0
ACA	0	0	0	1	1	0
ACG	1	1	1	1	1	1
ACT	0	0	1	0	0	0
GCA	0	0	0	1	1	0
GCG	0	0	0	0	0	1
CGA	0	0	0	0	0	1
CGC	0	0	0	1	1	1
CGT	1	1	1	1	0	0
GTA	1	1	1	1	0	0

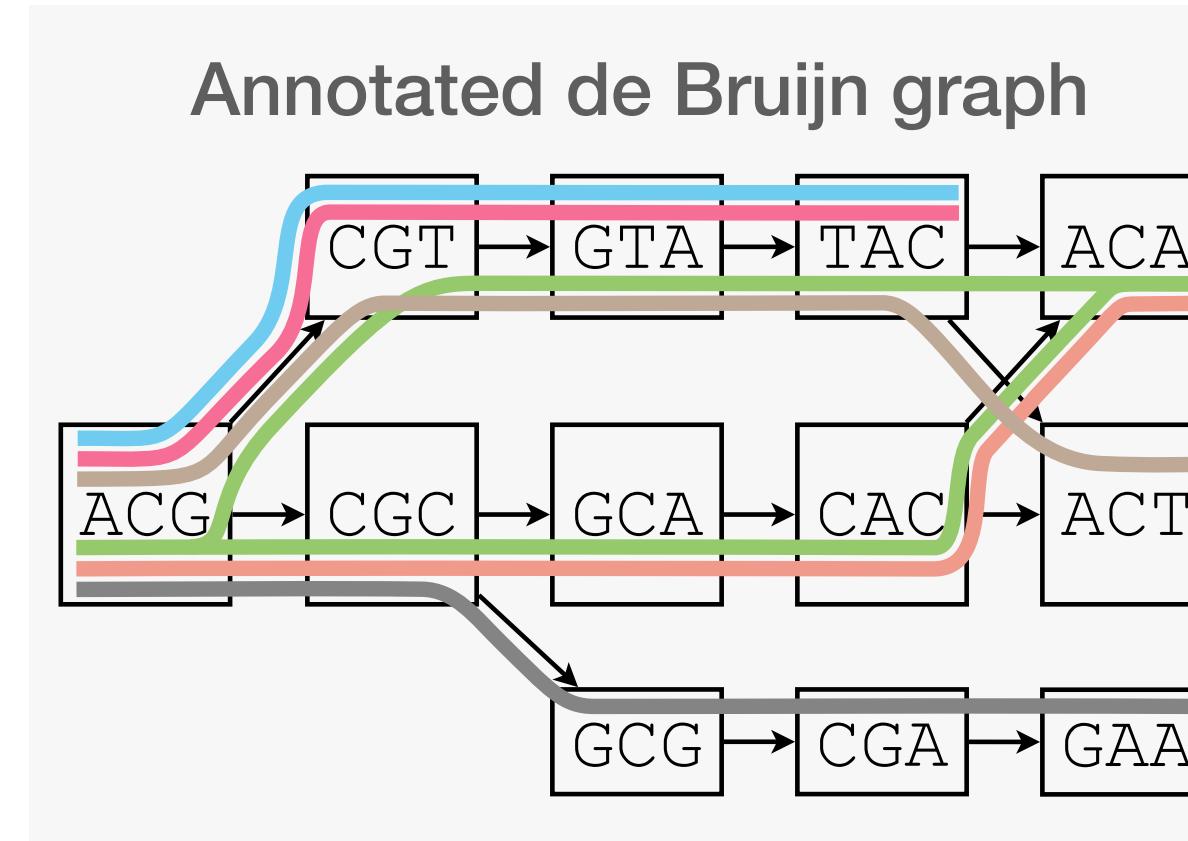
Representing graph

- ▶ Hash table
- ▶ Indicator set bit vector [Conway *et al.*, 2011]
- ▶ BOSS table [Bowe *et al.*, 2012]
 - ✓ ~2–3 bits per k-mer for DNA
 - ✓ supports search of sub-k-mers

Representing annotation

- ▶ Column-major sparse representation
- ▶ RowFlat (employed in VARI [Muggli *et al.*, 2017])
- ▶ Rainbowfish [Almodaresi *et al.*, 2017]
- ▶ Rainbow-MST [Almodaresi *et al.*, 2019]

Representing annotated De Bruijn graphs



Compressed representation

The table shows a compressed representation of the annotated de Bruijn graph. The left column lists the k-mers in the dictionary: CAC, GAA, TAC, ACA, ACG, ACT, GCA, GCG, CGA, CGC, CGT, and GTA. The right side shows the "Annotation matrix" with columns corresponding to the same k-mers. A red box highlights the first four columns of the matrix. Below the table, two values are shown: $\sim 10^{11}$ for the total number of annotations and $\sim 10^6$ for the size of the k-mer dictionary.

	CAC	GAA	TAC	ACA	ACG	ACT	GCA	GCG	CGA	CGC	CGT	GTA
Annotation matrix	0 0 0 1 1 0	0 0 0 0 0 1	1 1 1 1 0 0	0 0 0 1 1 0	1 1 1 1 1 1	0 0 1 0 0 0	0 0 0 1 1 0	0 0 0 0 0 1	0 0 0 0 0 1	0 0 0 1 1 1	1 1 1 1 0 0	1 1 1 1 0 0

$\sim 10^{11}$

$\sim 10^6$

Representing graph

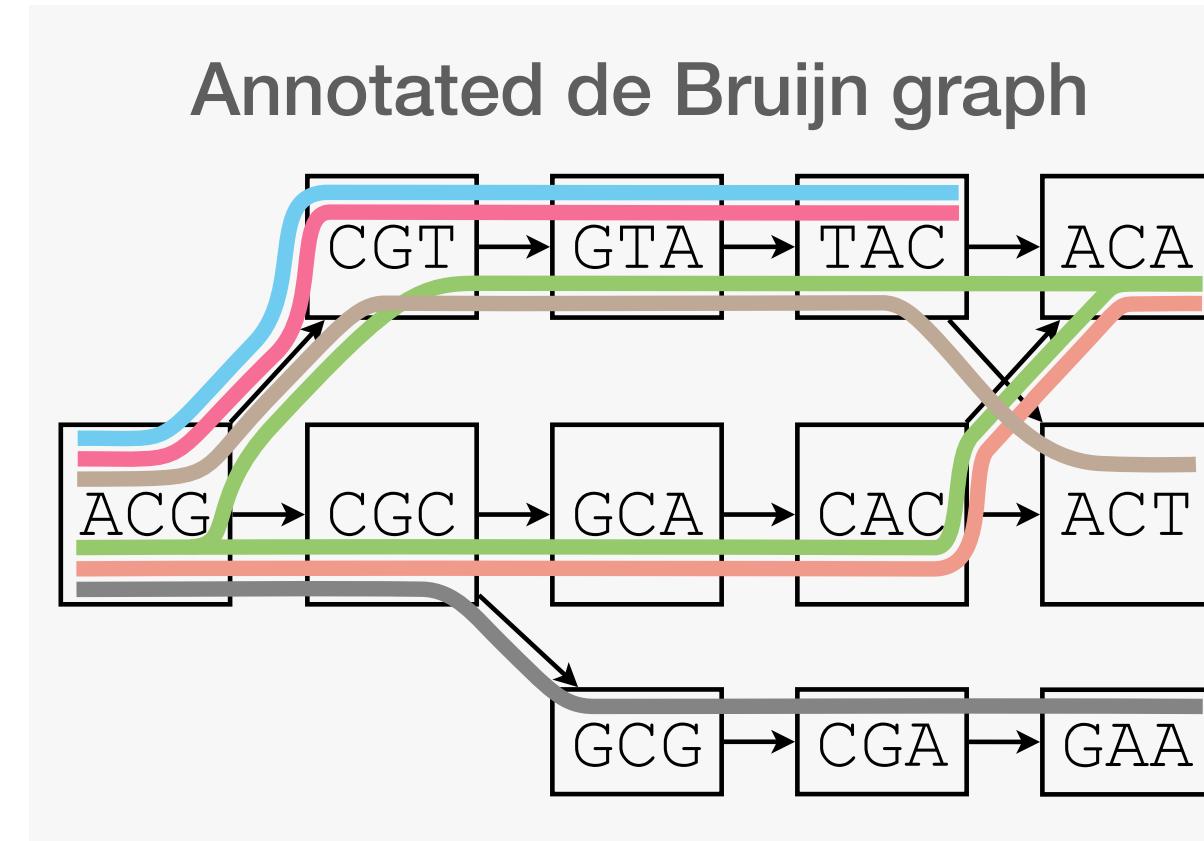
- ▶ Hash table
- ▶ Indicator set bit vector [Conway *et al.*, 2011]
- ▶ BOSS table [Bowe *et al.*, 2012]
 - ✓ ~2–3 bits per k-mer for DNA
 - ✓ supports search of sub-k-mers

Representing annotation

- ▶ Column-major sparse representation
- ▶ RowFlat (employed in VARI [Muggli *et al.*, 2017])
- ▶ Rainbowfish [Almodaresi *et al.*, 2017]
- ▶ Rainbow-MST [Almodaresi *et al.*, 2019]

Challenge:
Represent huge-scale annotations

Representing graph annotations



Compressed representation

K-mer dictionary

	Blue	Pink	Brown	Green	Red	Grey
CAC	0	0	0	1	1	0
GAA	0	0	0	0	0	1
TAC	1	1	1	1	0	0
ACA	0	0	0	1	1	0
ACG	1	1	1	1	1	1
ACT	0	0	1	0	0	0
GCA	0	0	0	1	1	0
GCG	0	0	0	0	0	1
CGA	0	0	0	0	0	1
CGC	0	0	0	1	1	1
CGT	1	1	1	1	0	0
GTA	1	1	1	1	0	0

$\sim 10^6$

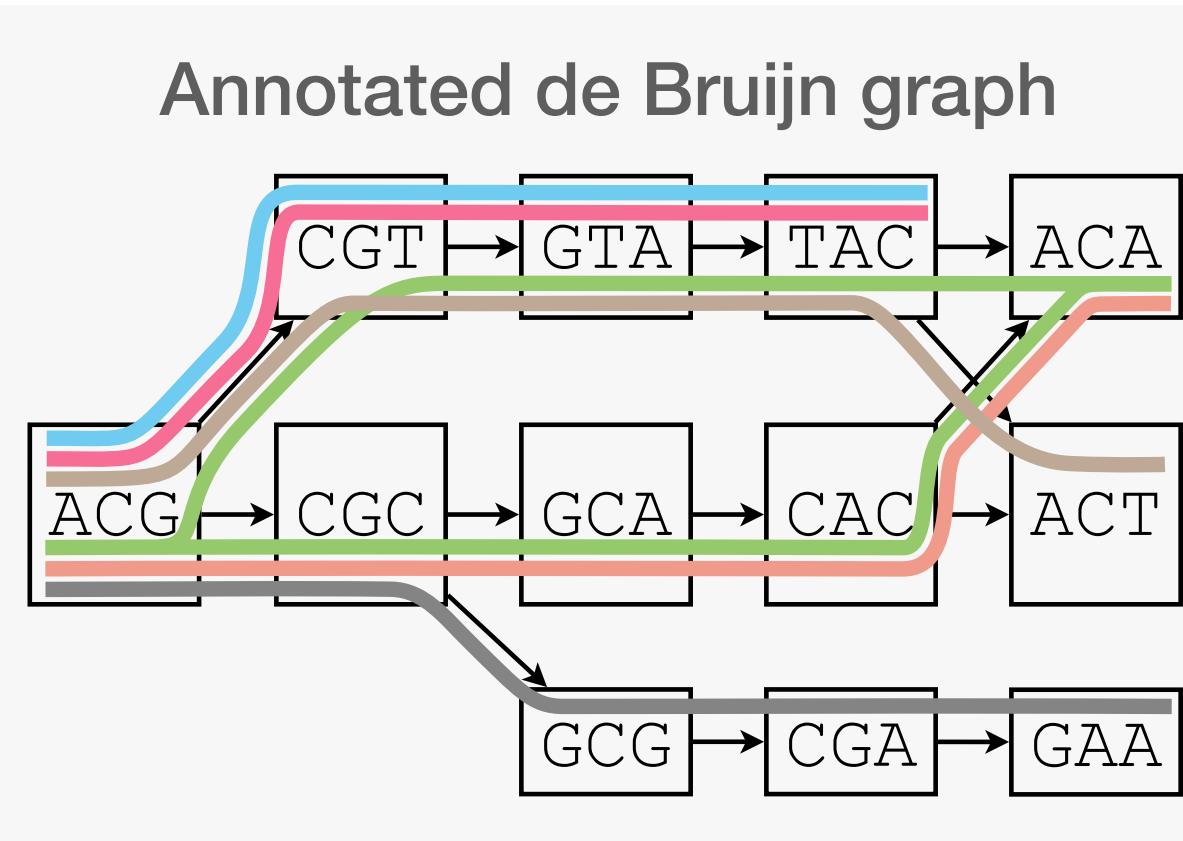
$\sim 10^{11}$

Challenge:
Represent huge-scale annotations

Typical properties

1. high sparsity

Representing graph annotations



Compressed representation

k-mer dictionary

	Annotation matrix					
CAC	0 0 0	1 1 0				
GAA	0 0 0	0 0 1				
TAC	1 1 1	1 0 0				
ACA	0 0 0	1 1 0				
ACG	1 1 1	1 1 1				
ACT	0 0 1	0 0 0				
GCA	0 0 0	1 1 0				
GCG	0 0 0	0 0 1				
CGA	0 0 0	0 0 1				
CGC	0 0 0	1 1 1				
CGT	1 1 1	1 0 0				
GTA	1 1 1	1 0 0				

$$\sim 10^{11}$$

$\sim 10^6$

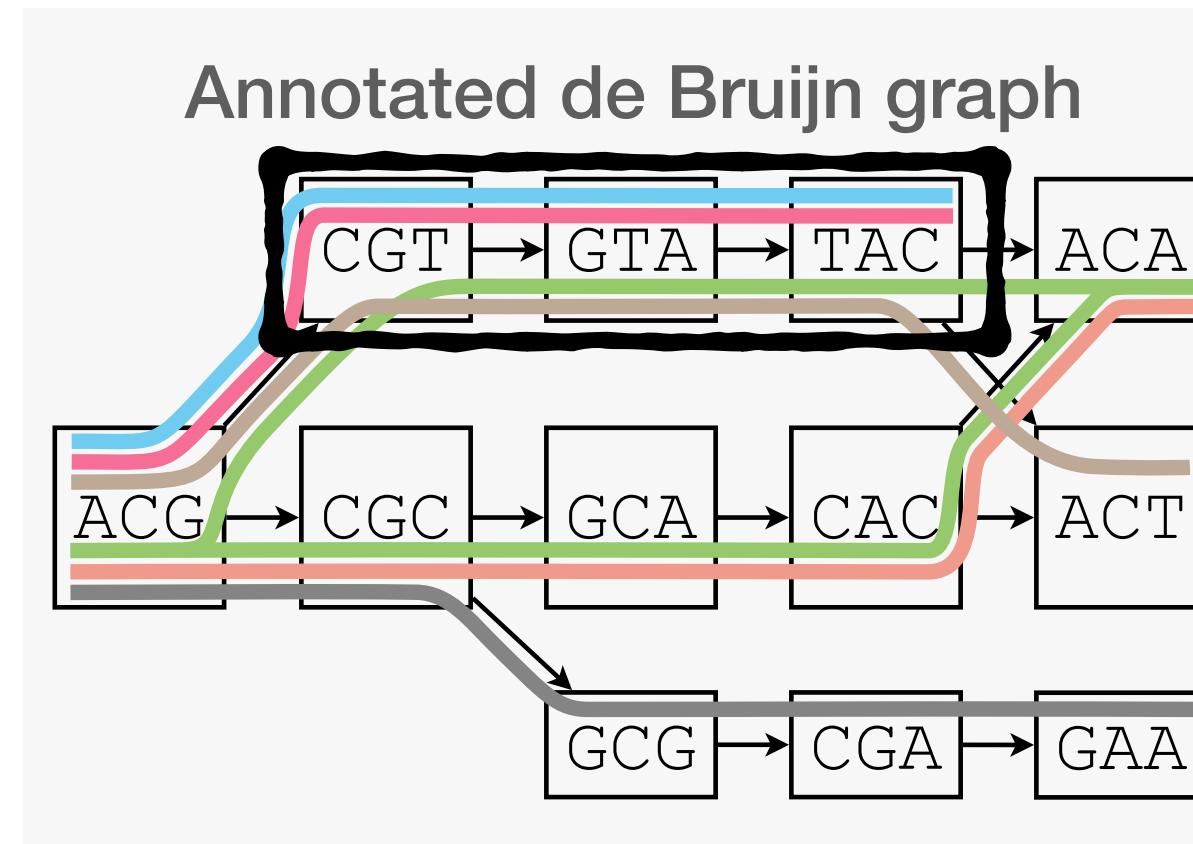
Challenge:

Represent huge-scale annotations

Typical properties

1. high sparsity
 2. similarity of columns

Representing graph annotations



The figure illustrates a k -mer dictionary and its compressed representation as an annotation matrix.

k -mer dictionary:

- CAC
- GAA
- TAC
- ACA
- ACG
- ACT
- GCA
- GCG
- CGA
- CGC
- CGT
- GTA

Annotation matrix:

	Blue	Pink	Brown	Green	Red	Grey
CAC	0	0	0	1	1	0
GAA	0	0	0	0	0	1
TAC	1	1	1	1	0	0
ACA	0	0	0	1	1	0
ACG	1	1	1	1	1	1
ACT	0	0	1	0	0	0
GCA	0	0	0	1	1	0
GCG	0	0	0	0	0	1
CGA	0	0	0	0	0	1
CGC	0	0	0	1	1	1
CGT	1	1	1	1	0	0
GTA	1	1	1	1	0	0

Annotations are represented by colored circles above the matrix:

- Blue circle: Column 1 (CAC, ACA, CGC)
- Pink circle: Column 2 (GAA, GCA, CGT)
- Brown circle: Column 3 (TAC, ACT, GCG)
- Green circle: Column 4 (ACA, ACG, CGA)
- Red circle: Column 5 (ACG, ACT, CGC)
- Grey circle: Column 6 (CGC, CGT, GTA)

Specific rows (TAC, ACG, ACT) are highlighted with a red border around their corresponding columns in the matrix.

$\sim 10^{11}$

$\sim 10^6$

Challenge:

Represent huge-scale annotations

Typical properties

1. high sparsity
 2. similarity of **columns**
 3. similarity of **rows**

Annotation representations

1. Column-major sparse representation

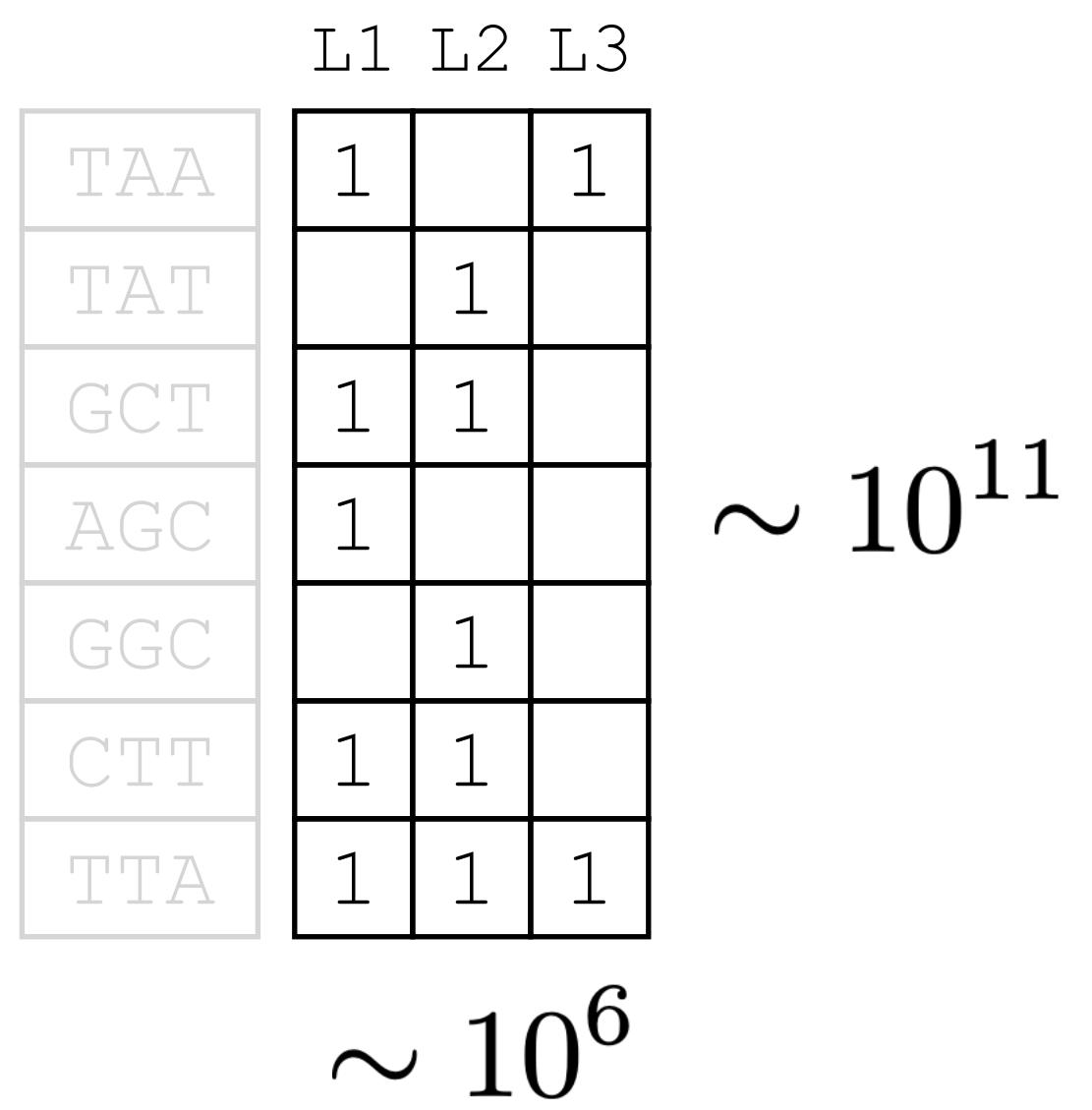
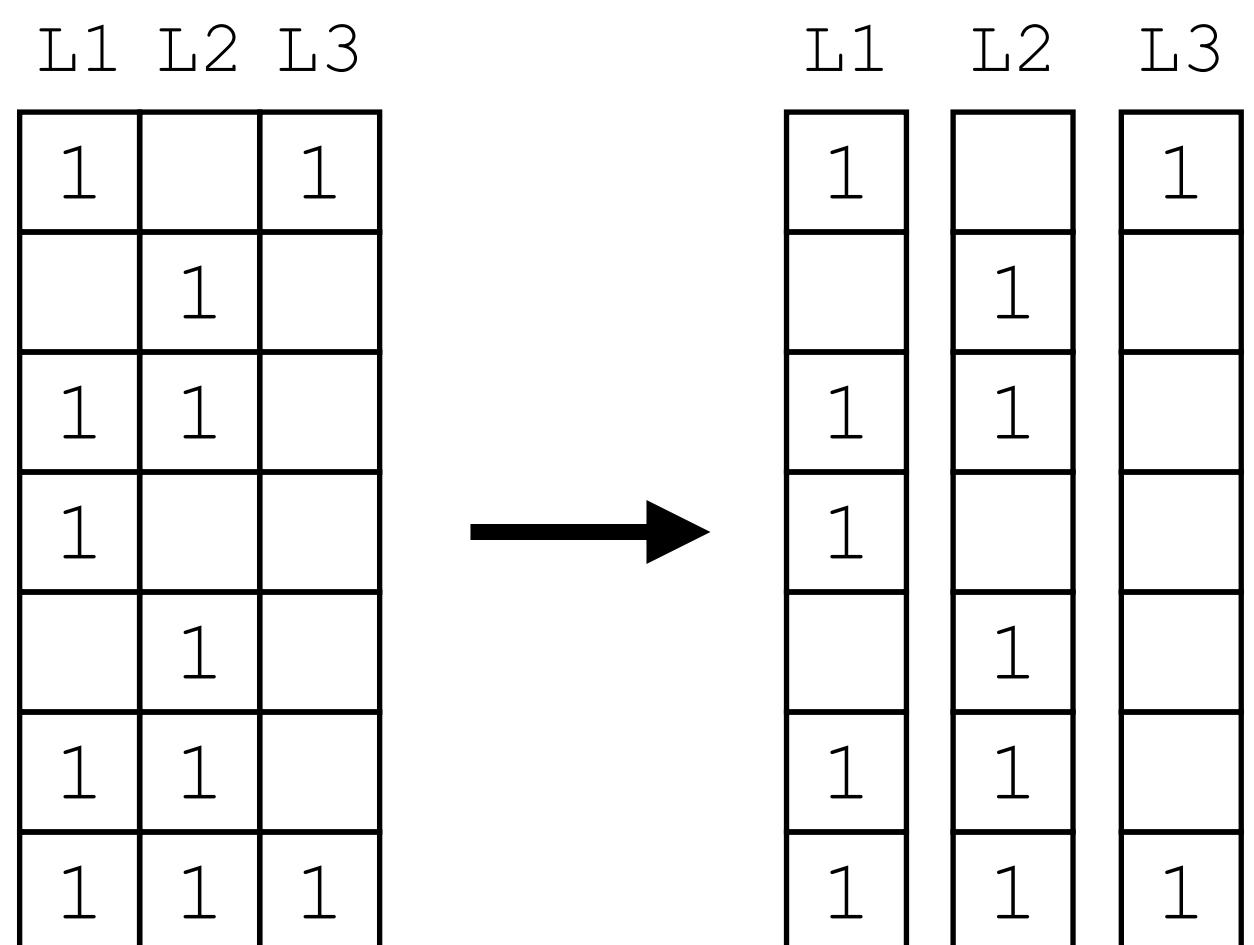
	L1	L2	L3
TAA	1		1
TAT		1	
GCT	1	1	
AGC	1		
GGC		1	
CTT	1	1	
TTA	1	1	1

$\sim 10^{11}$

$\sim 10^6$

Annotation representations

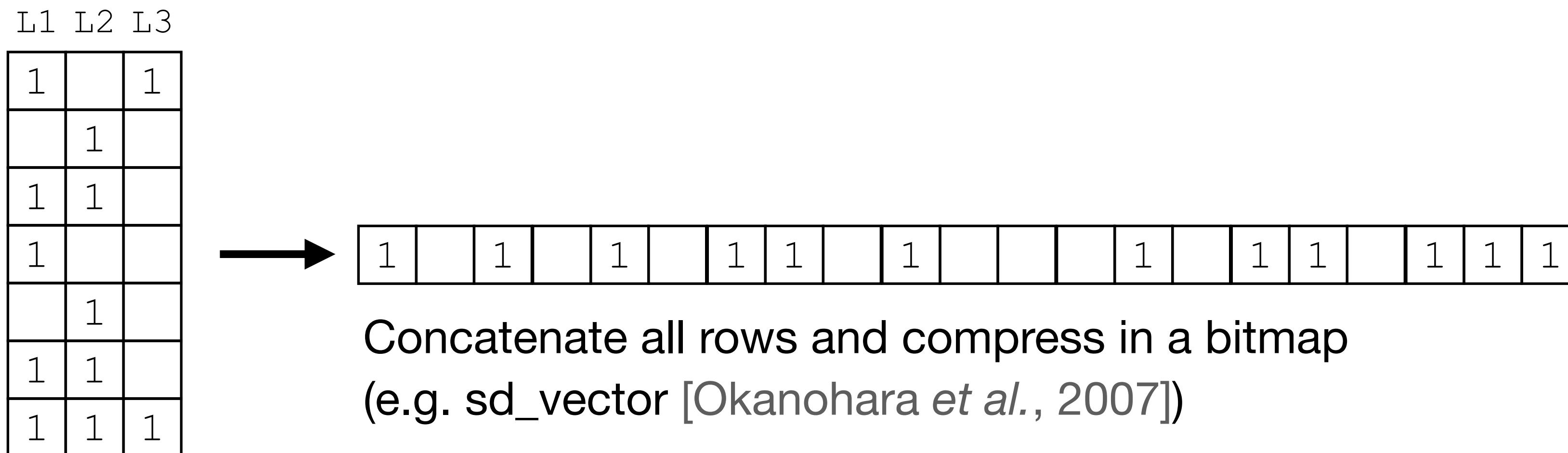
1. Column-major sparse representation



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

Annotation representations

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



Annotation representations

1. Column-major sparse representation
2. RowFlat (employed in VARI [Muggli *et al.*, 2017])
3. BRWT [Barbay *et al.*, 2013], [Karasikov *et al.*, 2019]

1	1	1				
2						
3					1	
4				1		1
5					1	
6	1					
7		1	1			

Annotation representations

1. Column-major sparse representation
2. RowFlat (employed in VARI [Muggli *et al.*, 2017])
3. BRWT [Barbay *et al.*, 2013], [Karasikov *et al.*, 2019]

1	1	1				
2						
3					1	
4				1		1
5					1	
6	1					
7		1	1			

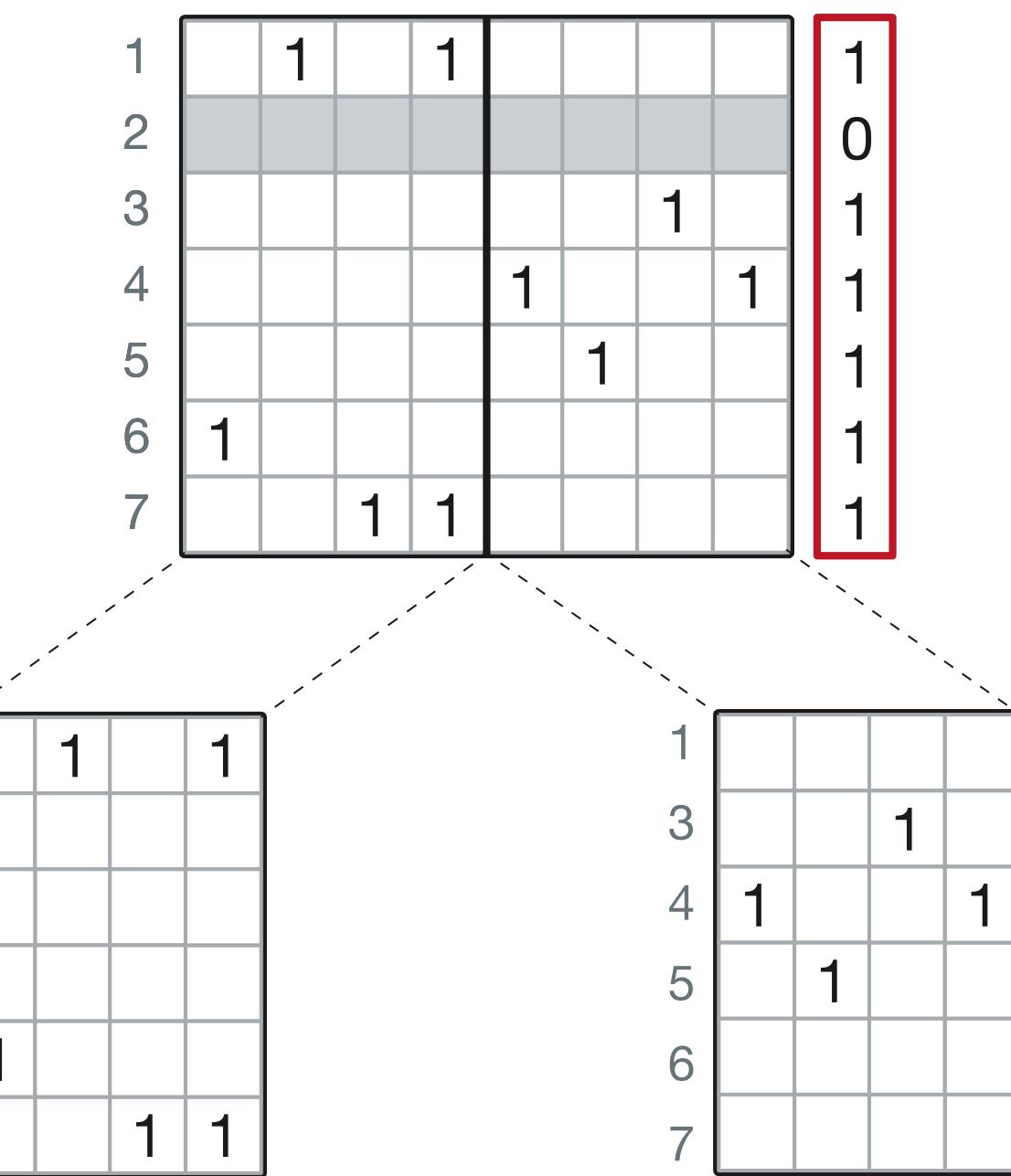
Annotation representations

1. Column-major sparse representation
2. RowFlat (employed in VARI [Muggli *et al.*, 2017])
3. BRWT [Barbay *et al.*, 2013], [Karasikov *et al.*, 2019]

1	1	1					1
2							0
3						1	1
4				1		1	1
5					1		1
6	1						1
7		1	1				1

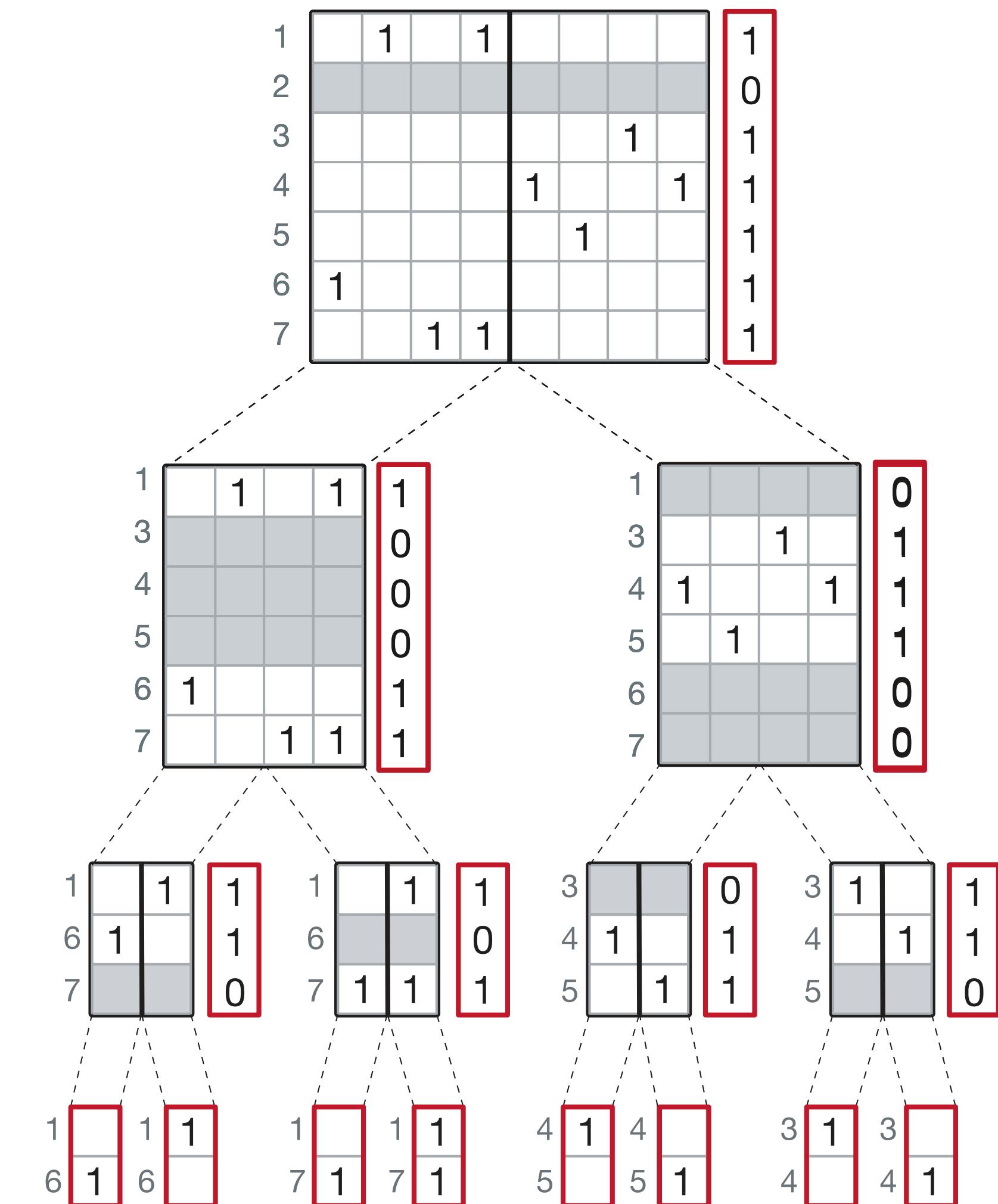
Annotation representations

1. Column-major sparse representation
2. RowFlat (employed in VARI [Muggli *et al.*, 2017])
3. BRWT [Barbay *et al.*, 2013], [Karasikov *et al.*, 2019]



Annotation representations

1. Column-major sparse representation
2. RowFlat (employed in VARI [Muggli *et al.*, 2017])
3. BRWT [Barbay *et al.*, 2013], [Karasikov *et al.*, 2019]



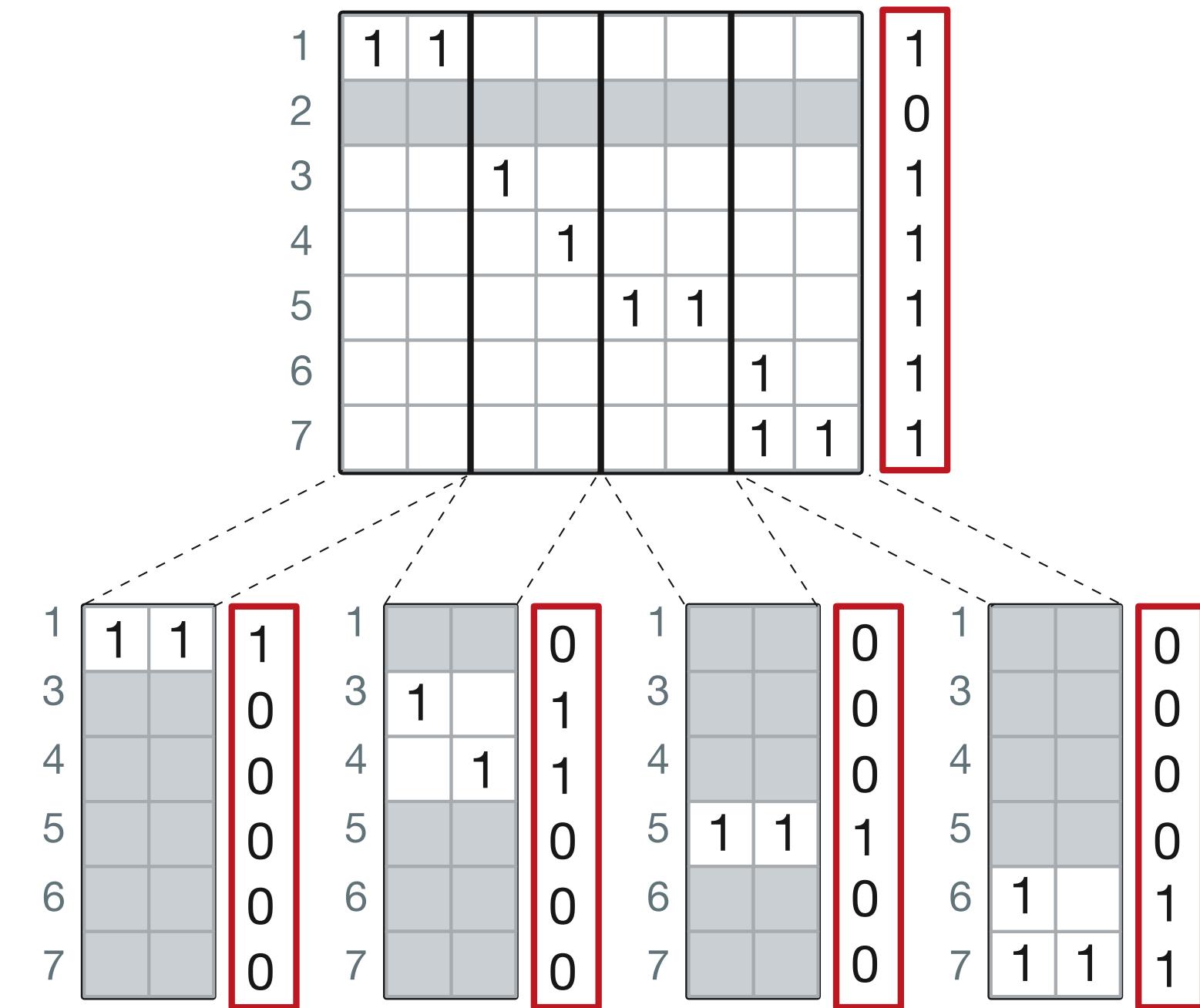
Annotation representations

Sparse Binary Relation Representations for Genome Graph Annotation

Mikhail Karasikov, Harun Mustafa, Amir Joudaki, Sara Javadzadeh-No, Gunnar Rätsch & André Kahles

Conference paper | First Online: 02 April 2019

1. Column-major sparse representation
2. RowFlat (employed in VARI [Muggli *et al.*, 2017])
3. BRWT [Barbay *et al.*, 2013], [Karasikov *et al.*, 2019]
4. Multi-BRWT [Karasikov *et al.*, 2019]
 - ▶ Optimize column arrangement
 - ▶ Use multi-ary trees



Annotation representations

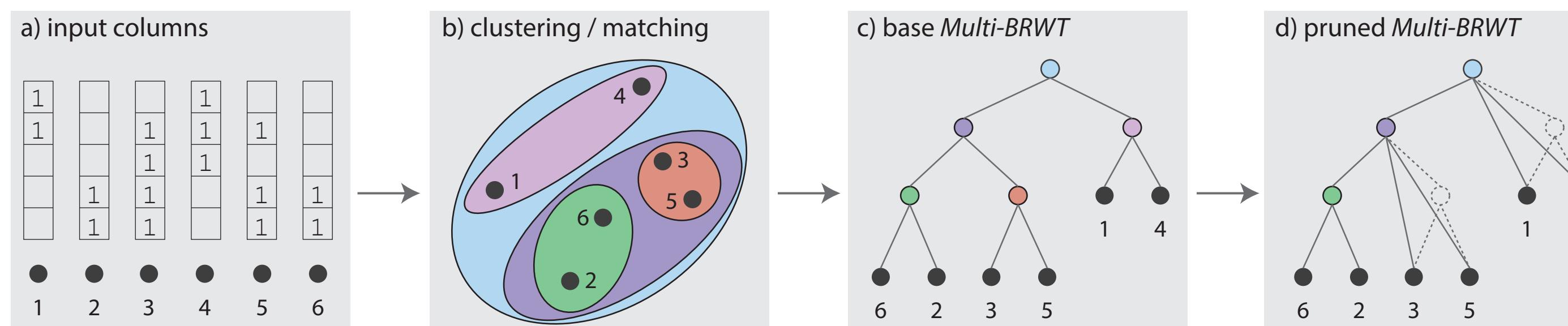
Sparse Binary Relation Representations for Genome Graph Annotation

Mikhail Karasikov, Harun Mustafa, Amir Joudaki, Sara Javadzadeh-No, Gunnar Rätsch & André Kahles

Conference paper | First Online: 02 April 2019

1. Column-major sparse representation
2. RowFlat (employed in VARI [Muggli *et al.*, 2017])
3. BRWT [Barbay *et al.*, 2013], [Karasikov *et al.*, 2019]
4. Multi-BRWT [Karasikov *et al.*, 2019]

- ▶ Optimize column arrangement
- ▶ Use multi-ary trees



A sparse binary relation representation for genome graph annotation is shown as a 7x7 matrix. The matrix has 1s at positions (1,1), (1,2), (2,3), (3,4), (4,5), (5,6), (6,7), (7,1), (7,2), (7,3), (7,4), (7,5), (7,6), and (7,7). A red border highlights the last column. This matrix is decomposed into four smaller 7x7 matrices, each with a red border, representing a column-major sparse representation.

1	1						1
2		1					0
3			1				1
4				1			1
5					1	1	1
6						1	1
7						1	1

1	1						1
3		1					0
4			1				1
5				1			1
6					0		0
7						0	0

1	1						1
3		1					0
4			1				1
5				1			1
6					0		0
7						1	0

1	1						0
3		1					0
4			1				0
5				1			0
6					1		1
7						1	1

Annotation representations

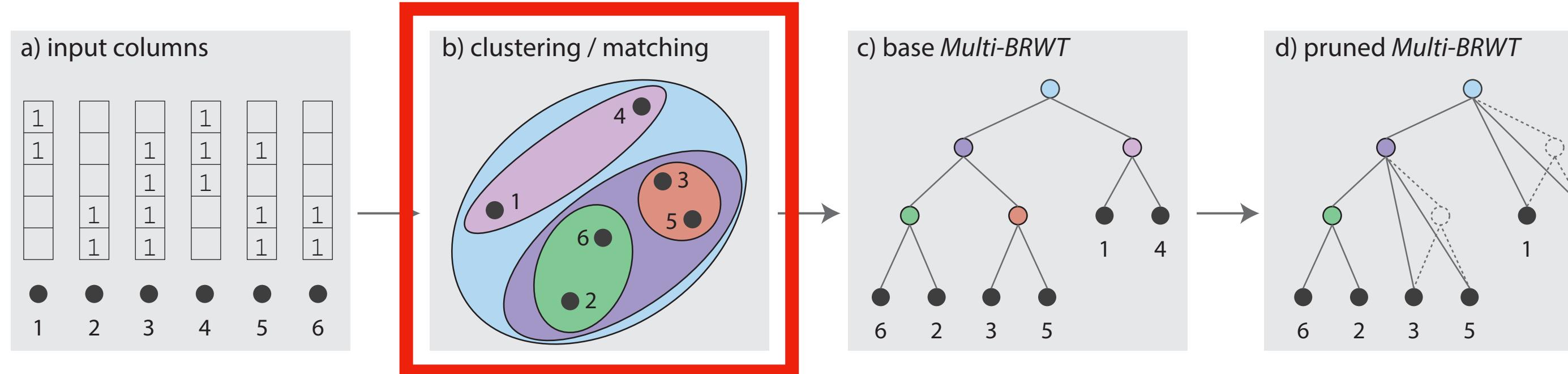
Sparse Binary Relation Representations for Genome Graph Annotation

Mikhail Karasikov, Harun Mustafa, Amir Joudaki, Sara Javadzadeh-No, Gunnar Rätsch & André Kahles

Conference paper | First Online: 02 April 2019

1. Column-major sparse representation
2. RowFlat (employed in VARI [Muggli *et al.*, 2017])
3. BRWT [Barbay *et al.*, 2013], [Karasikov *et al.*, 2019]
4. Multi-BRWT [Karasikov *et al.*, 2019]

- ▶ Optimize column arrangement
- ▶ Use multi-ary trees



The figure shows a sparse binary relation matrix for genome graph annotation. The matrix has 7 rows (labeled 1-7) and 13 columns (labeled 1-13). Red boxes highlight specific columns: column 13 contains all 1s; columns 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, and 12 contain sparse binary values (0 or 1).

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	1												1
2		1											0
3			1										1
4				1									1
5					1								1
6						1							1
7							1						1

Annotation representations

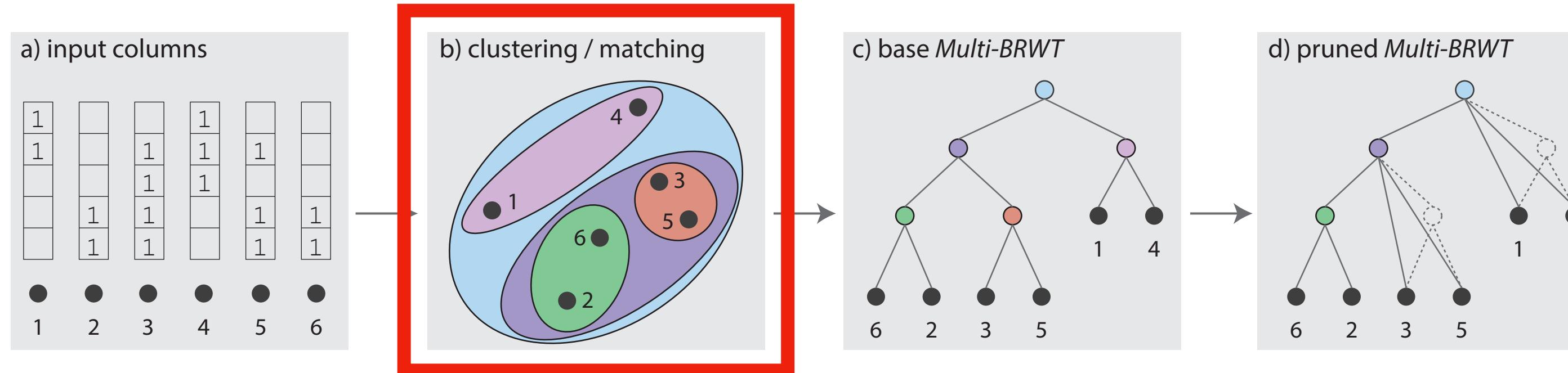
Sparse Binary Relation Representations for Genome Graph Annotation

Mikhail Karasikov, Harun Mustafa, Amir Joudaki, Sara Javadzadeh-No, Gunnar Rätsch & André Kahles

Conference paper | First Online: 02 April 2019

1. Column-major sparse representation
2. RowFlat (employed in VARI [Muggli *et al.*, 2017])
3. BRWT [Barbay *et al.*, 2013], [Karasikov *et al.*, 2019]
4. Multi-BRWT [Karasikov *et al.*, 2019]

- ▶ Optimize column arrangement
- ▶ Use multi-ary trees



Non-trivial for >500,000 columns

The figure shows a sparse binary relation matrix for genome graph annotation. The matrix is 7x7, with rows and columns labeled 1 through 7. Non-zero entries are highlighted in red boxes. The matrix is shown in two configurations: its original state and after pruning.

	1	2	3	4	5	6	7
1	1	1					1
2			1				0
3				1			1
4					1		1
5					1	1	1
6						1	1
7						1	1

Annotation representations

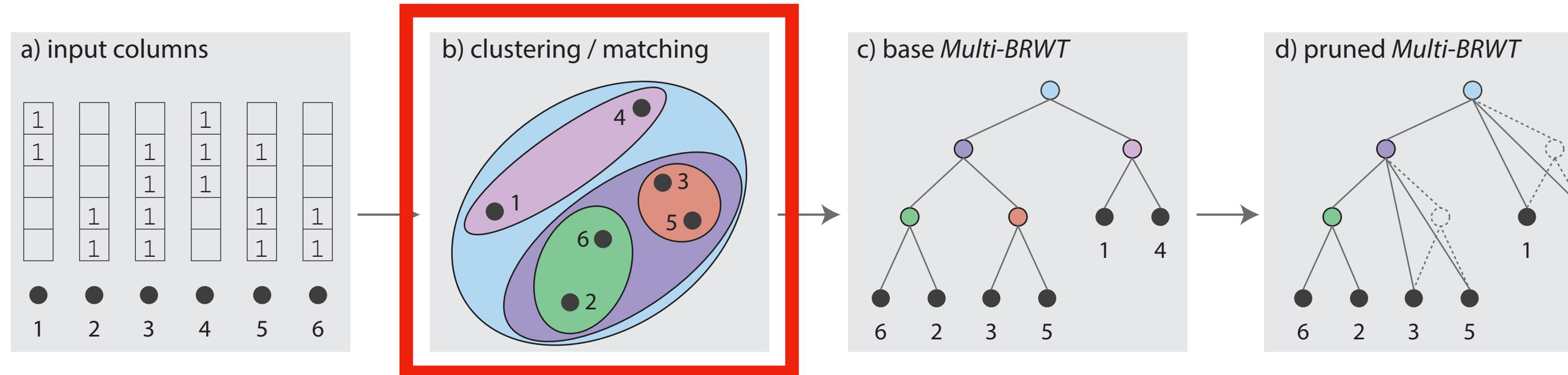
Sparse Binary Relation Representations for Genome Graph Annotation

Mikhail Karasikov, Harun Mustafa, Amir Joudaki, Sara Javadzadeh-No, Gunnar Rätsch & André Kahles

Conference paper | First Online: 02 April 2019

1. Column-major sparse representation
2. RowFlat (employed in VARI [Muggli *et al.*, 2017])
3. BRWT [Barbay *et al.*, 2013], [Karasikov *et al.*, 2019]
4. Multi-BRWT [Karasikov *et al.*, 2019]

- ▶ Optimize column arrangement
- ▶ Use multi-ary trees



Non-trivial for >500,000 columns
Use taxonomy

The figure shows a sparse binary relation representation for genome graph annotation. It consists of a main matrix and several smaller matrices connected by dashed arrows.

Main Matrix:

1	1						1
2		1					0
3			1				1
4				1			1
5					1	1	1
6						1	1
7						1	1

Small Matrices:

- Matrix 1: Rows 1-3, Columns 1-2.
- Matrix 2: Rows 3-5, Columns 3-4.
- Matrix 3: Rows 5-7, Columns 5-6.
- Matrix 4: Rows 6-7, Columns 6-7.
- Matrix 5: Rows 6-7, Columns 7-8.

Red boxes highlight specific columns in the main matrix and the first small matrix, indicating the focus on specific columns during the representation process.

Annotation representations

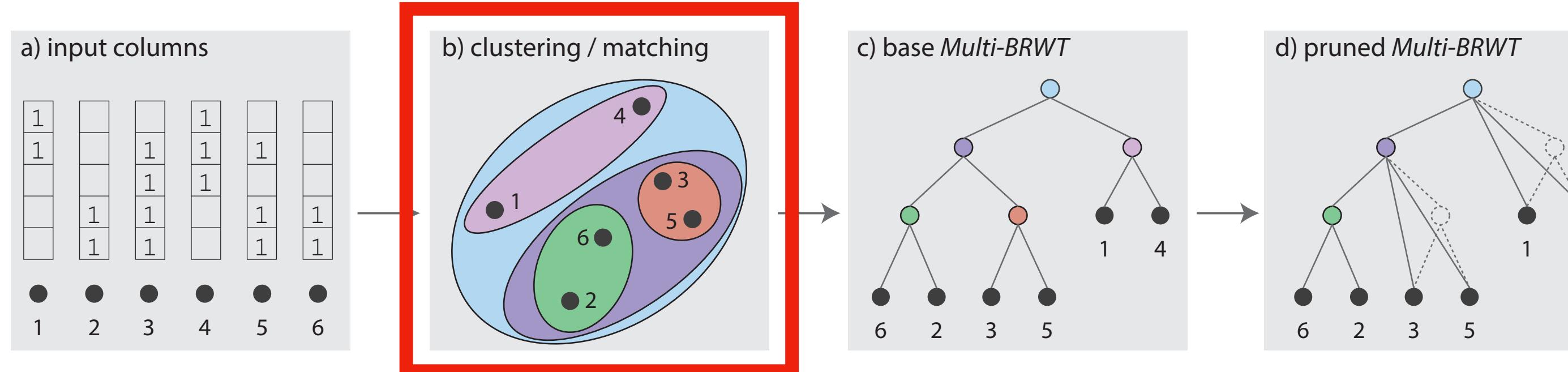
Sparse Binary Relation Representations for Genome Graph Annotation

Mikhail Karasikov, Harun Mustafa, Amir Joudaki, Sara Javadzadeh-No, Gunnar Rätsch & André Kahles

Conference paper | First Online: 02 April 2019

1. Column-major sparse representation
2. RowFlat (employed in VARI [Muggli *et al.*, 2017])
3. BRWT [Barbay *et al.*, 2013], [Karasikov *et al.*, 2019]
4. Multi-BRWT [Karasikov *et al.*, 2019]

- ▶ Optimize column arrangement
- ▶ Use multi-ary trees



Non-trivial for >500,000 columns
Use taxonomy

A hierarchical sparse binary matrix representation. The top matrix is a 7x7 grid of binary values (0s and 1s). It is partitioned into 4x4 submatrices, which are further partitioned into 3x3 submatrices. Some of these 3x3 submatrices have red borders, indicating they are non-zero or part of a specific structure. Dashed lines connect the top matrix to the bottom matrices, showing the hierarchical decomposition.

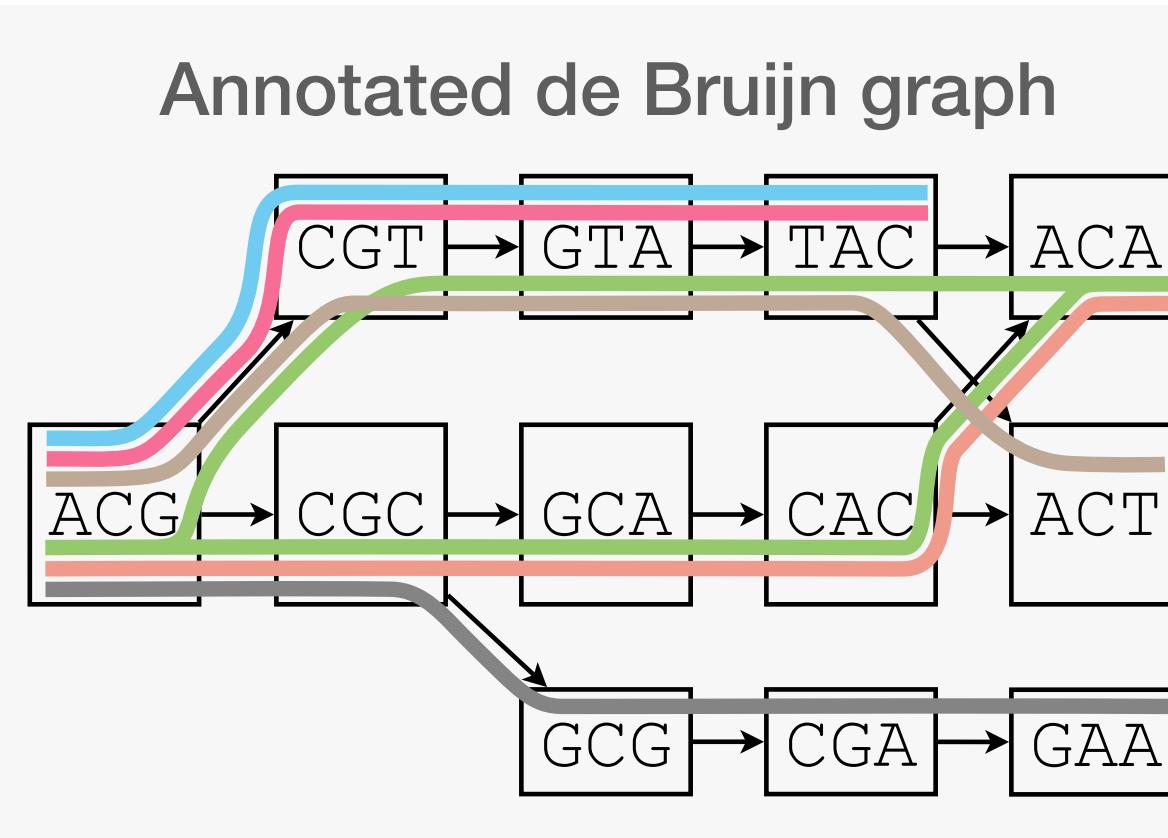
Efficient and Robust Search of Microbial Genomes via Phylogenetic Compression

Karel Brinda, Leandro Lima, Simone Pignotti, Natalia Quinones-Olvera, Kamil Salikhov, Rayan Chikhi, Gregory Kucherov, Zamin Iqbal, Michael Baym

doi: <https://doi.org/10.1101/2023.04.15.536996>

Posted April 18, 2023.

Representing graph annotations



Compressed representation

k-mer dictionary

	CAC	GAA	TAC	ACA	ACG	ACT	GCA	GCG	CGA	CGC	CGT	GTA
Annotation matrix	0	0	0	1	1	0	0	0	0	0	0	0
CAC	0	0	0	0	0	0	1	1	0	0	0	0
GAA	0	0	0	0	0	0	0	0	0	0	0	1
TAC	1	1	1	1	1	0	0	0	0	0	0	0
ACA	0	0	0	1	1	0	0	0	0	0	0	0
ACG	1	1	1	1	1	1	1	1	1	1	1	1
ACT	0	0	1	0	0	0	0	0	0	0	0	0
GCA	0	0	0	1	1	0	0	0	0	0	0	0
GCG	0	0	0	0	0	0	1	1	1	1	1	1
CGA	0	0	0	0	0	0	0	0	1	1	1	1
CGC	0	0	0	1	1	1	1	1	1	1	1	1
CGT	1	1	1	1	1	0	0	0	0	0	0	0
GTA	1	1	1	1	1	0	0	0	0	0	0	0

$\sim 10^{11}$

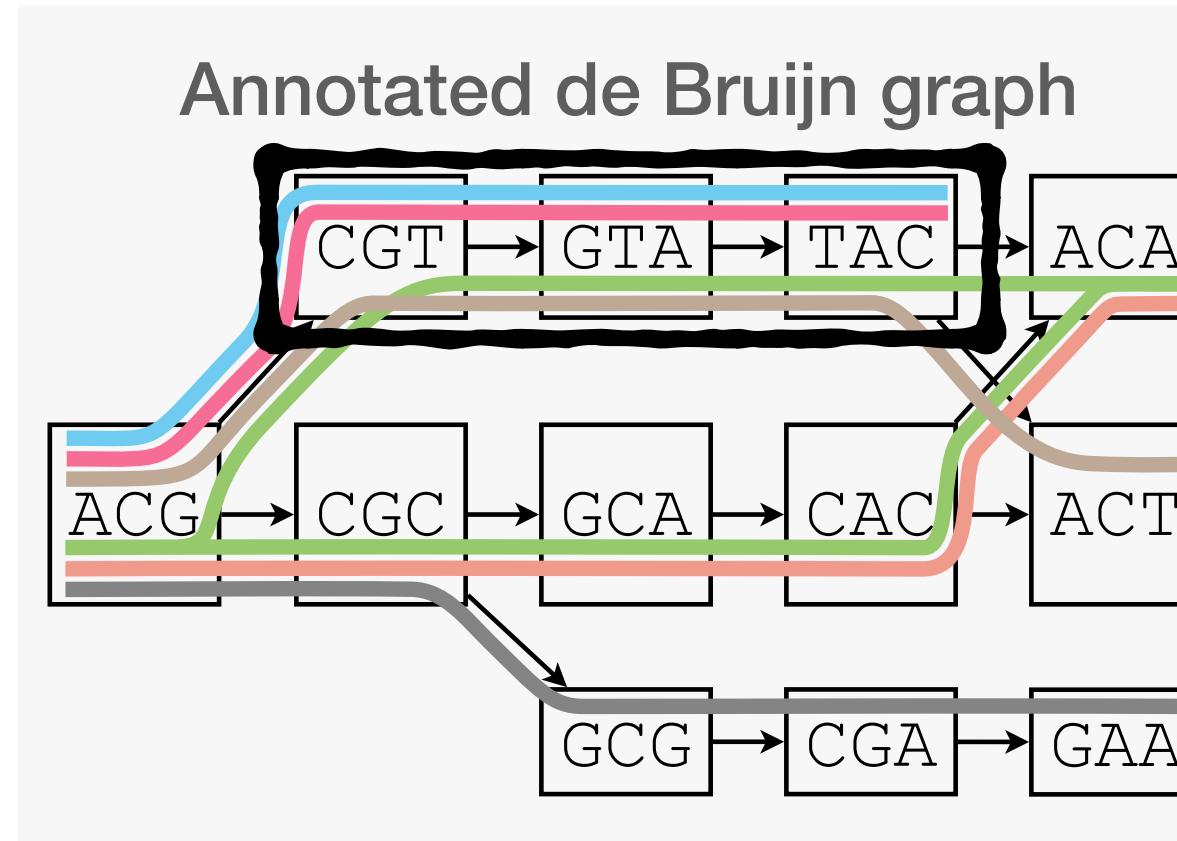
$\sim 10^6$

Challenge:
Represent huge-scale annotations

Typical properties

1. high sparsity
2. similarity of columns \Rightarrow **Multi-BRWT**

Representing graph annotations



Compressed representation

The table illustrates the compressed representation of the de Bruijn graph. It includes a k-mer dictionary mapping node names to their corresponding k-mers and an annotation matrix where columns represent nodes and rows represent k-mers. The matrix uses binary values (0 or 1) to indicate the presence of an annotation. A legend at the top shows colored circles corresponding to the annotations in the graph.

	CAC	GAA	TAC	ACA	ACG	ACT	GCA	GCG	CGA	CGC	CGT	GTA	
k-mer dictionary	0 0 0	1 1 0	0 0 0 0 0 1	1 1 1 1 0 0	0 0 0 1 1 0	1 1 1 1 1 1	0 0 1 0 0 0	0 0 0 1 1 0	0 0 0 0 0 1	0 0 0 0 0 1	0 0 0 1 1 1	1 1 1 1 0 0	1 1 1 1 0 0
Annotation matrix	0	0	0	1	1	0	0	0	0	0	1	0	0

$\sim 10^{11}$

$\sim 10^6$

Challenge:
Represent huge-scale annotations

Typical properties

1. high sparsity
2. similarity of **columns** \Rightarrow **Multi-BRWT**
3. similarity of **rows**
 - **adjacent nodes have similar annotations**

RowDiff

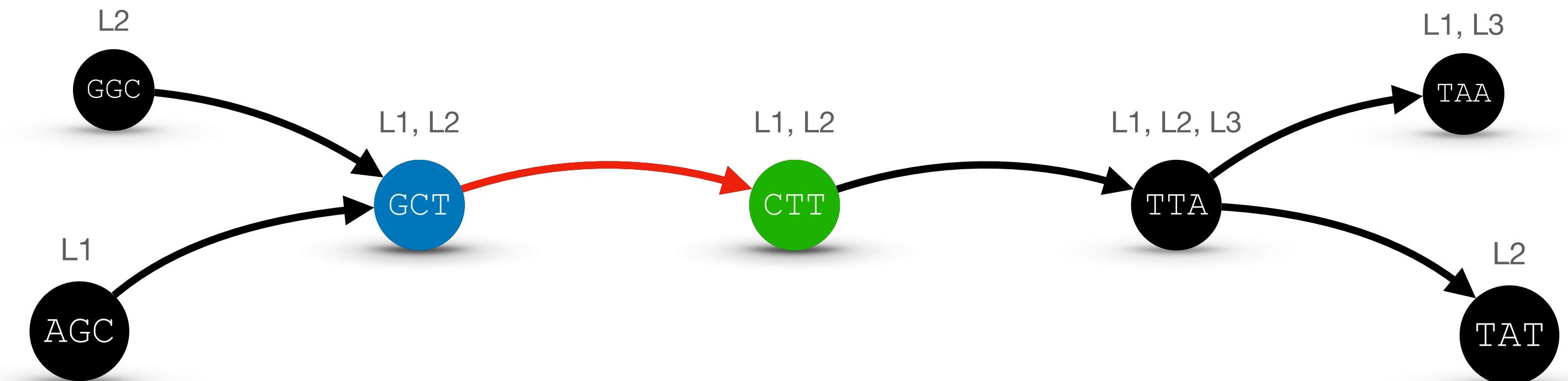
Delta compression

Idea:

- Store only **diffs**

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

↑
XOR



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,*}

RowDiff

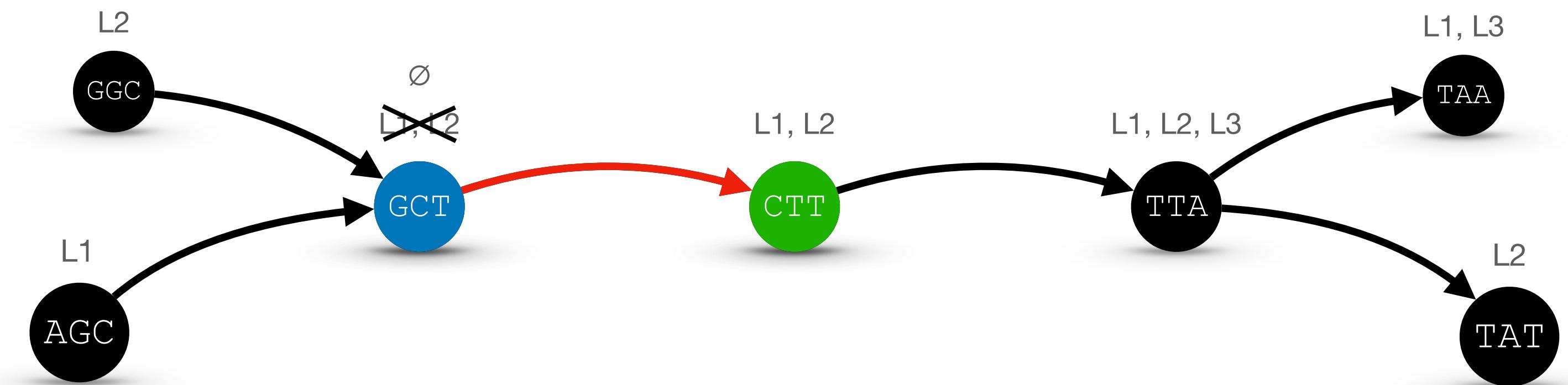
Delta compression

Idea:

- Store only **diffs**

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

↑
XOR



RowDiff

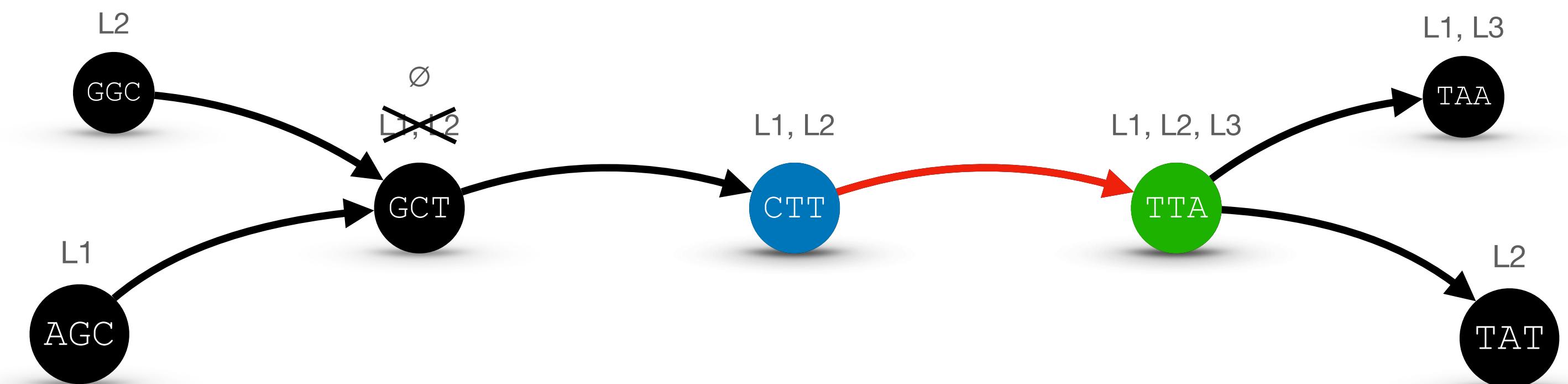
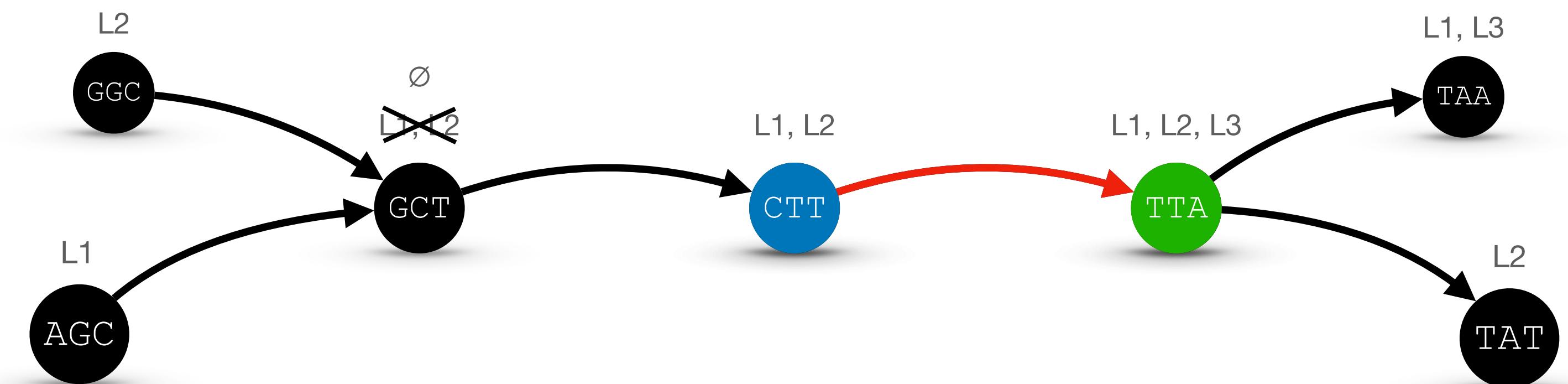
Delta compression

Idea:

- Store only **diffs**

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

↑
XOR



RowDiff

Delta compression

Topology-based sparsification of graph annotations

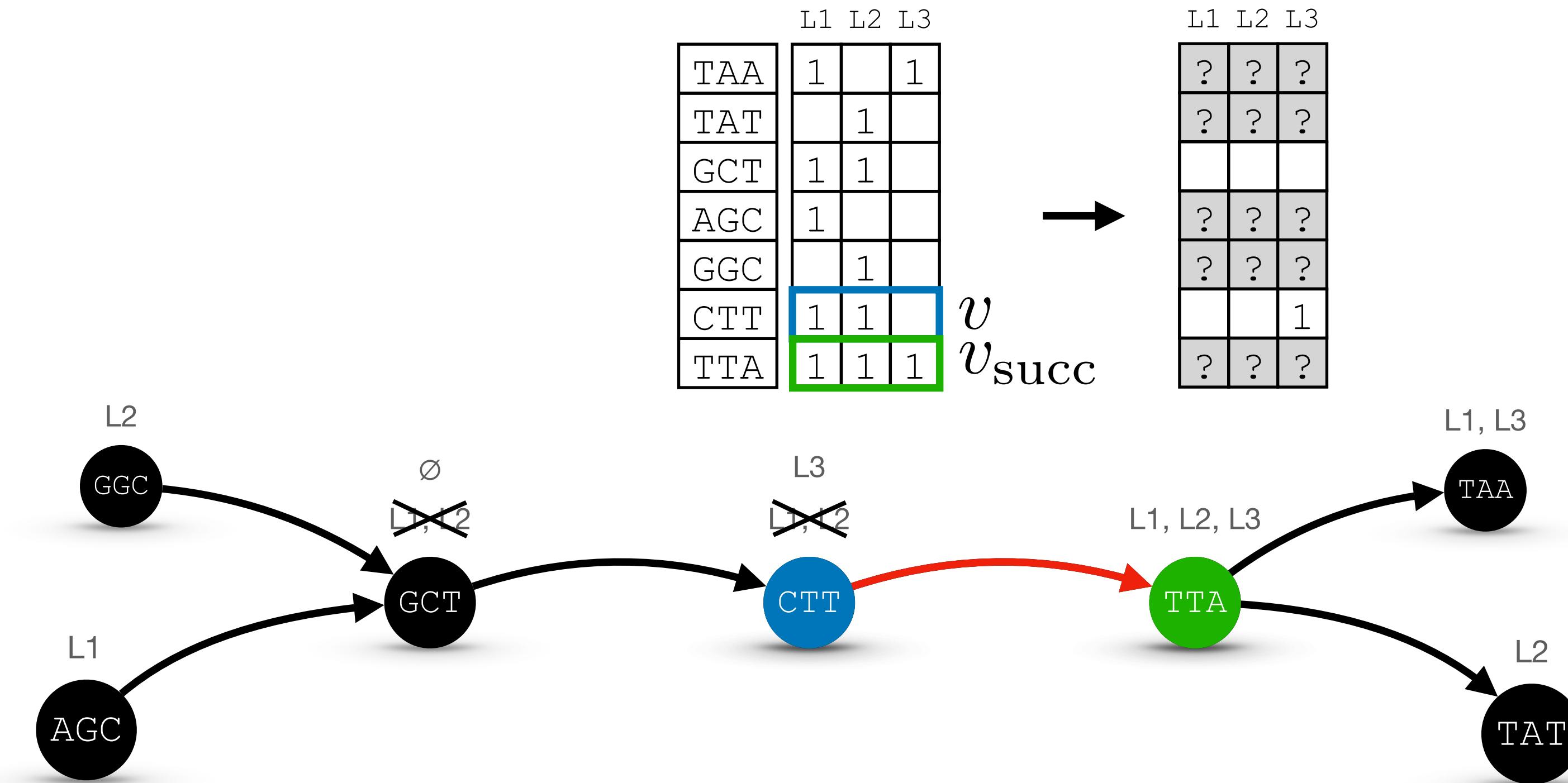
Daniel Danciu^{1,2,†}, Mikhail Karasikov^{1,2,3,†}, Harun Mustafa^{1,2,3}, André Kahles^{1,2,3,*} and Gunnar Rätsch^{1,2,3,4,*}

Idea:

- Store only **diffs**

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

↑
XOR



RowDiff

Delta compression

Topology-based sparsification of graph annotations

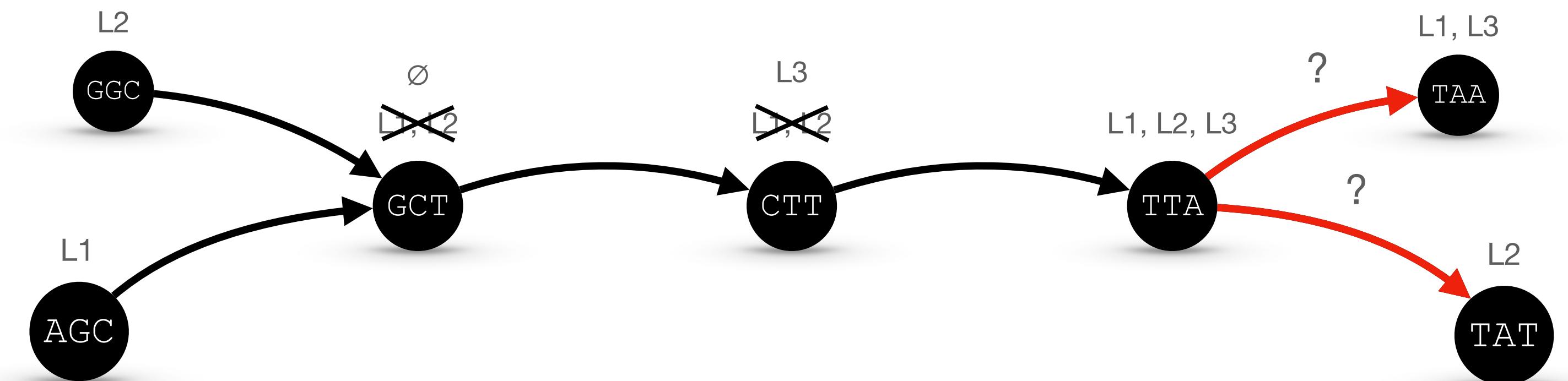
Daniel Danciu^{1,2,†}, Mikhail Karasikov^{1,2,3,†}, Harun Mustafa^{1,2,3}, André Kahles^{1,2,3,*} and Gunnar Rätsch^{1,2,3,4,*}

Idea:

- Store only **diffs**

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

XOR



RowDiff

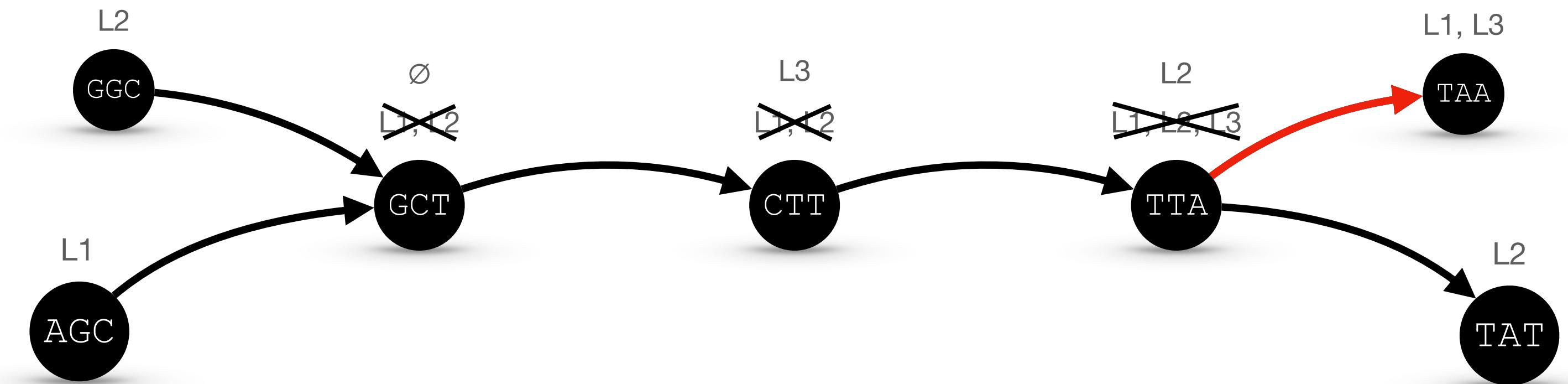
Delta compression

Idea:

- Store only **diffs**

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

XOR



RowDiff

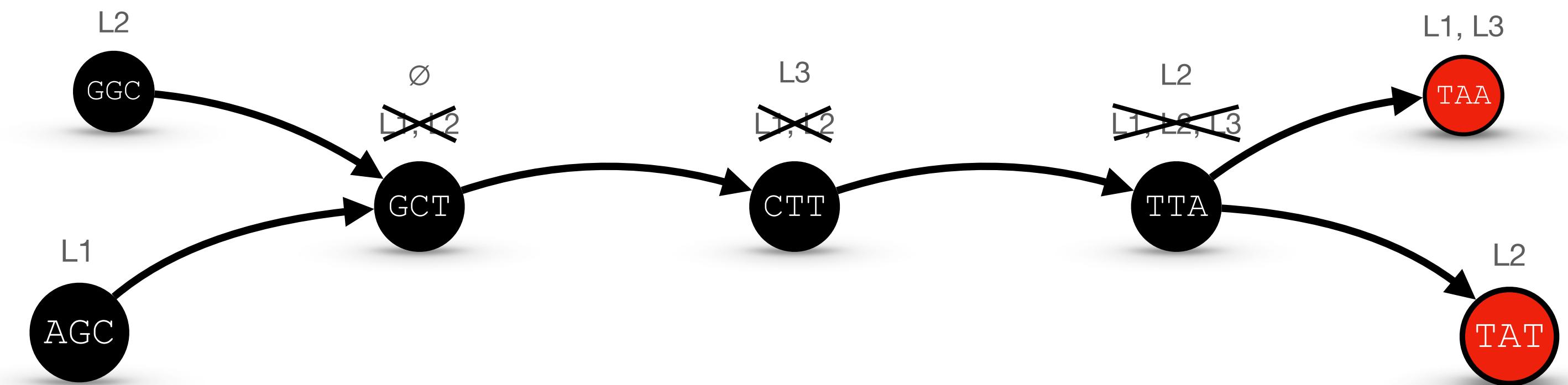
Delta compression

Idea:

- Store only **diffs**

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

↑
XOR



RowDiff

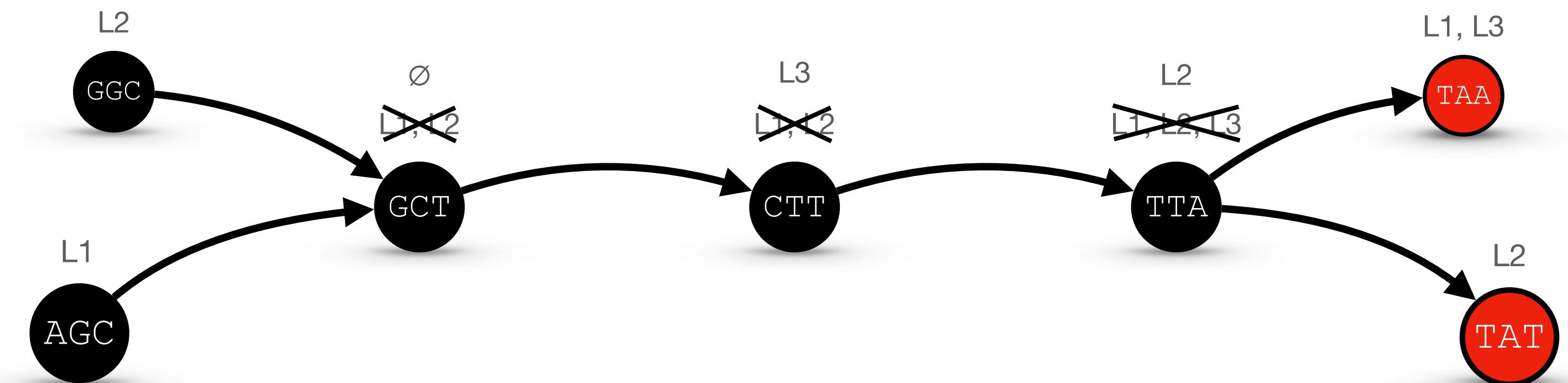
Delta compression

Idea:

- Store only **diffs**

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

↑
XOR



RowDiff

Delta compression

Topology-based sparsification of graph annotations

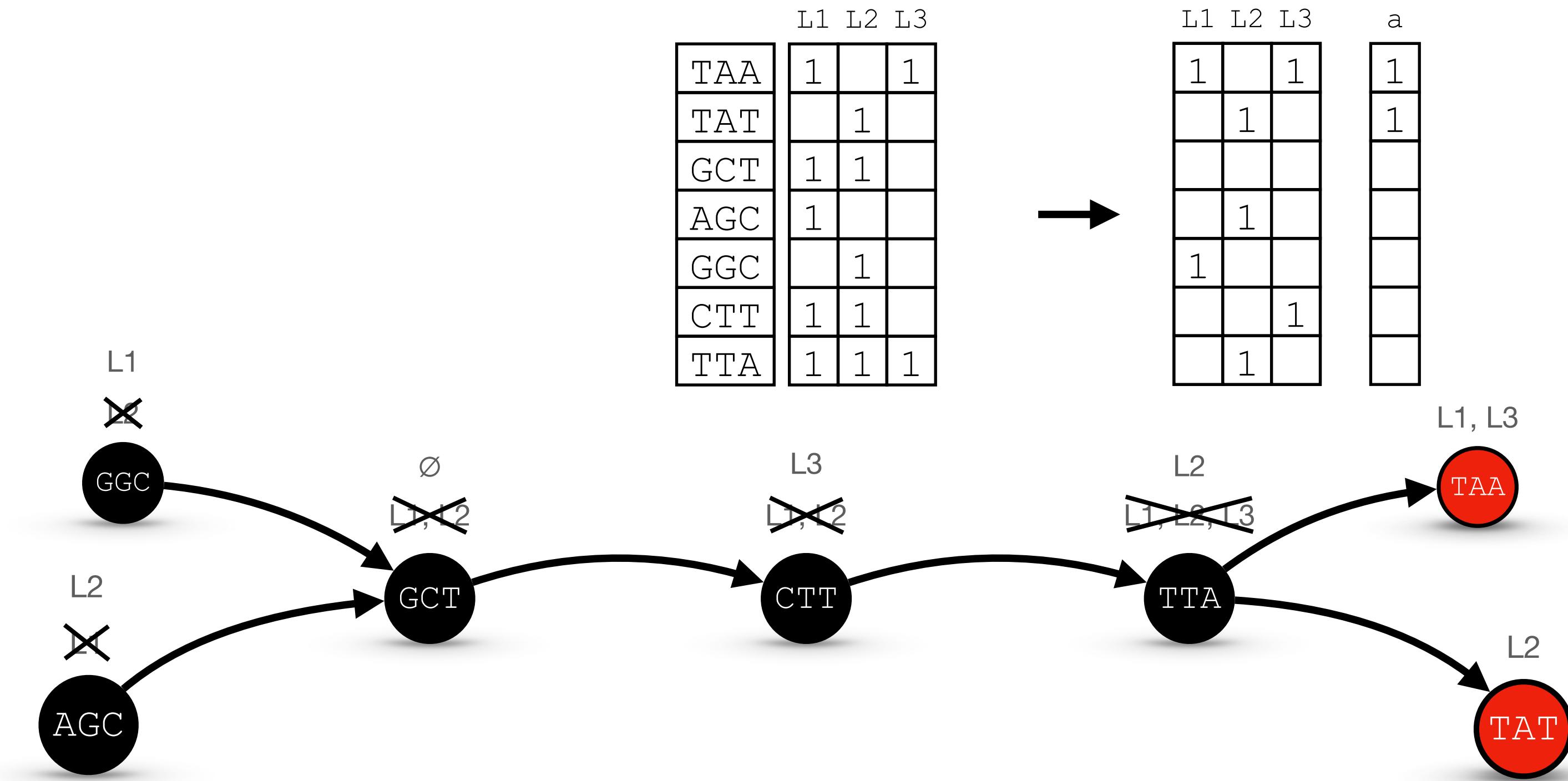
Daniel Danciu^{1,2,†}, Mikhail Karasikov^{1,2,3,†}, Harun Mustafa^{1,2,3}, André Kahles^{1,2,3,*} and Gunnar Rätsch^{1,2,3,4,*}

Idea:

- Store only **diffs**

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

XOR



RowDiff effectively **transforms** the matrix:

- **makes it sparser**, and thus, more compressible
- **any representation scheme** can then be applied on top
- the overhead is very small (<1 bit per node)

RowDiff

Delta compression

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,*}

Idea:

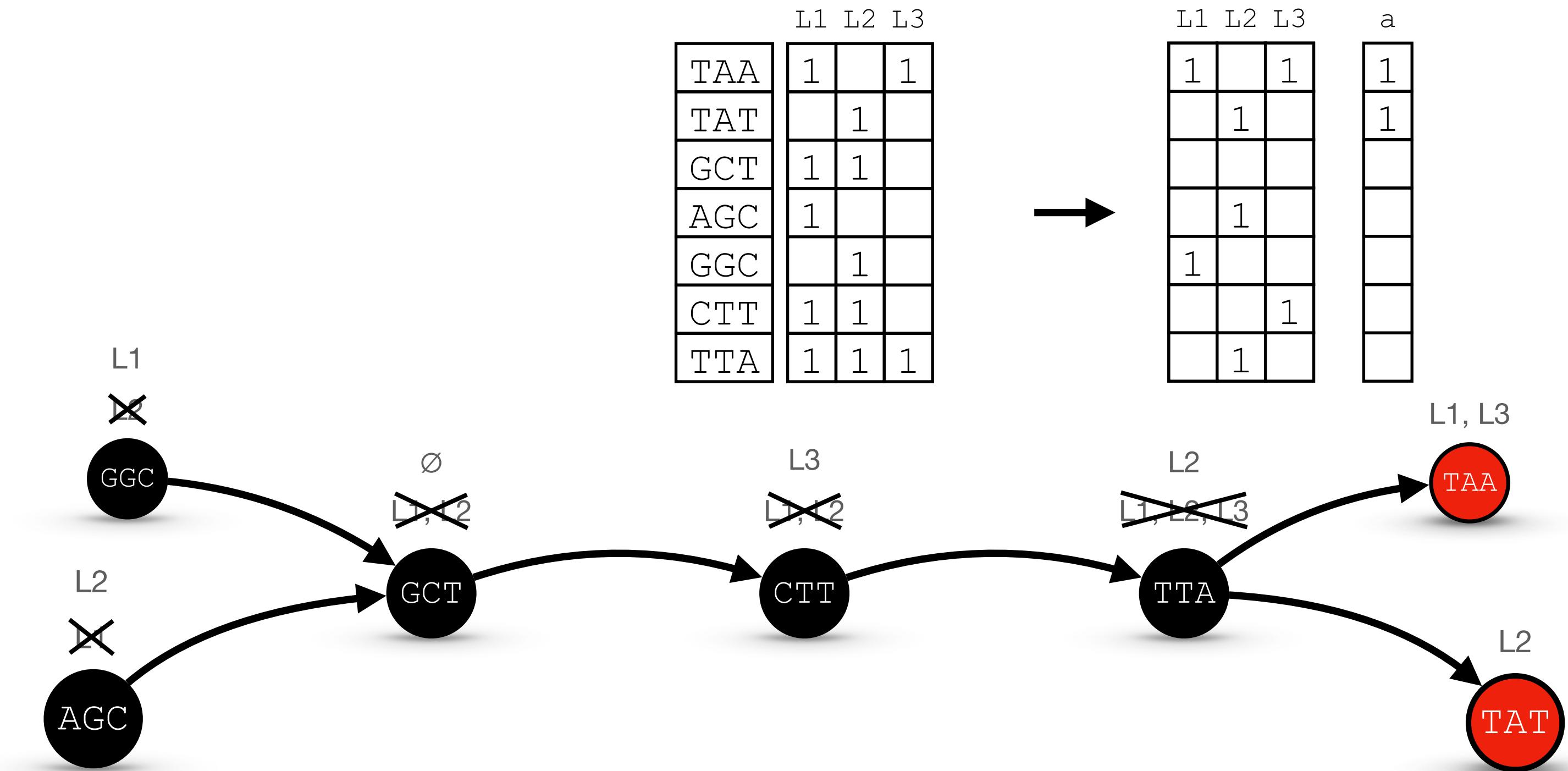
- Store only **diffs**

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

- Reconstruct

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

\uparrow
reconstruct recursively



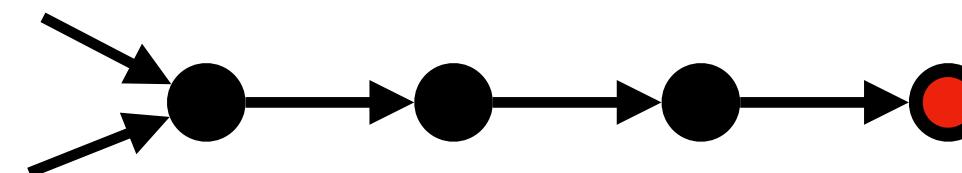
RowDiff effectively **transforms** the matrix:

- **makes it sparser**, and thus, more compressible
- **any representation scheme** can then be applied on top
- the overhead is very small (<1 bit per node)

RowDiff

Anchor nodes

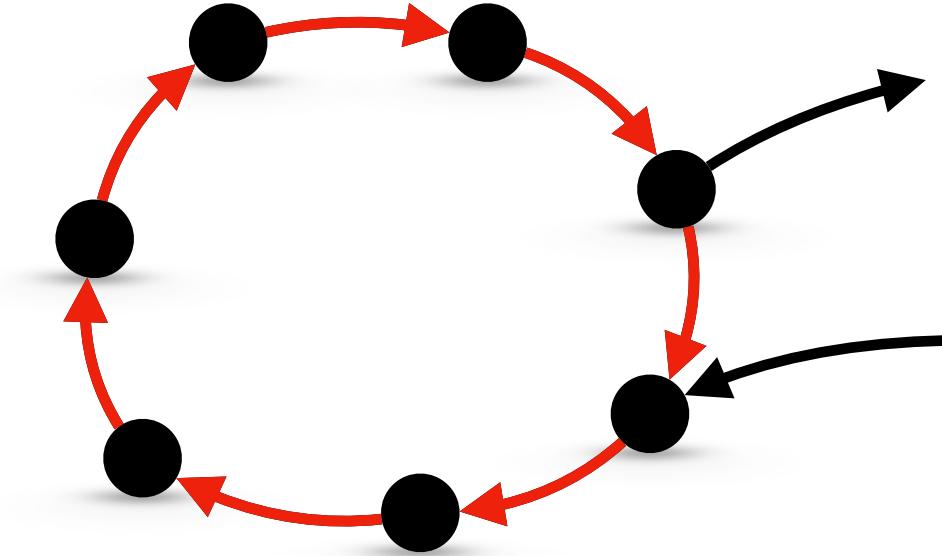
1. **Each *sink node* (with no outgoing edges) must be anchored**



RowDiff

Anchor nodes

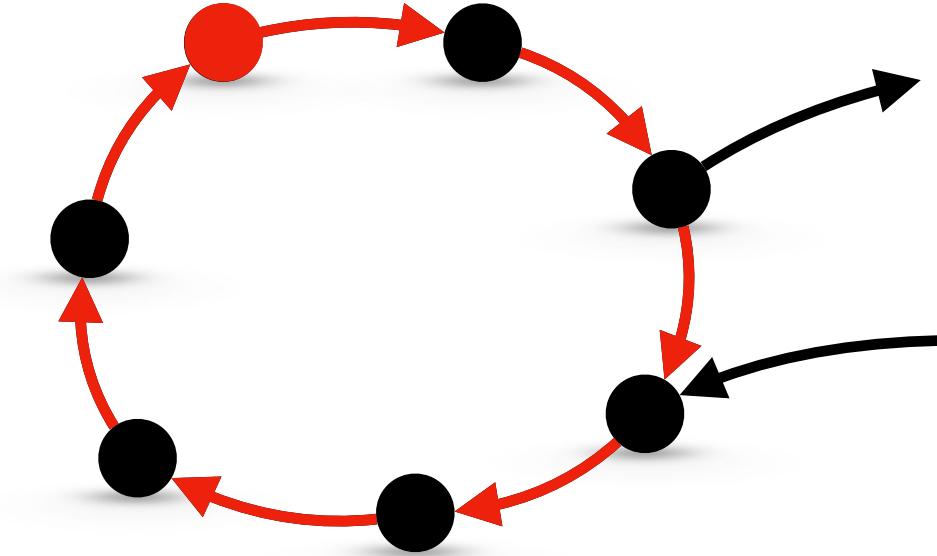
1. **Each *sink node* (with no outgoing edges) must be anchored**
2. **Each *row-diff cycle* must have at least one anchor node in it**



RowDiff

Anchor nodes

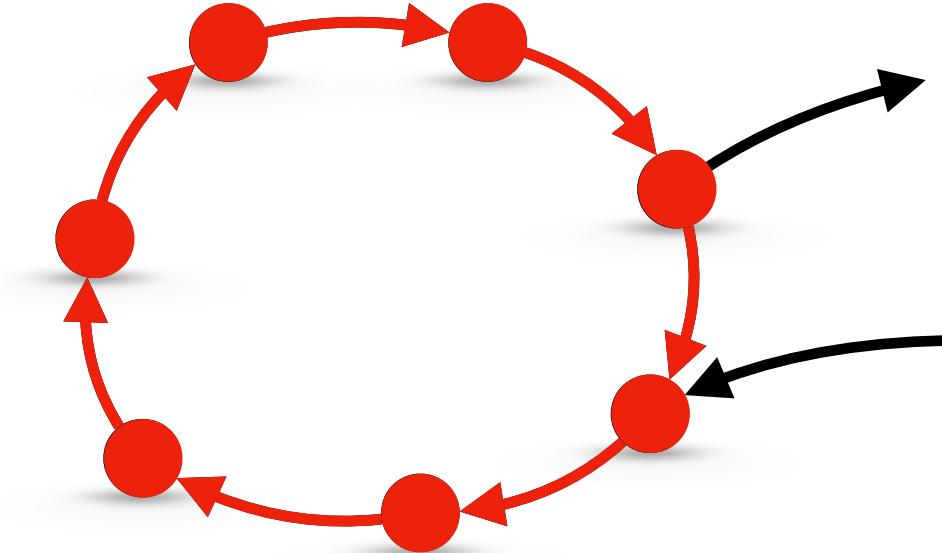
1. **Each *sink node* (with no outgoing edges) must be anchored**
2. **Each *row-diff cycle* must have at least one anchor node in it**



RowDiff

Anchor nodes

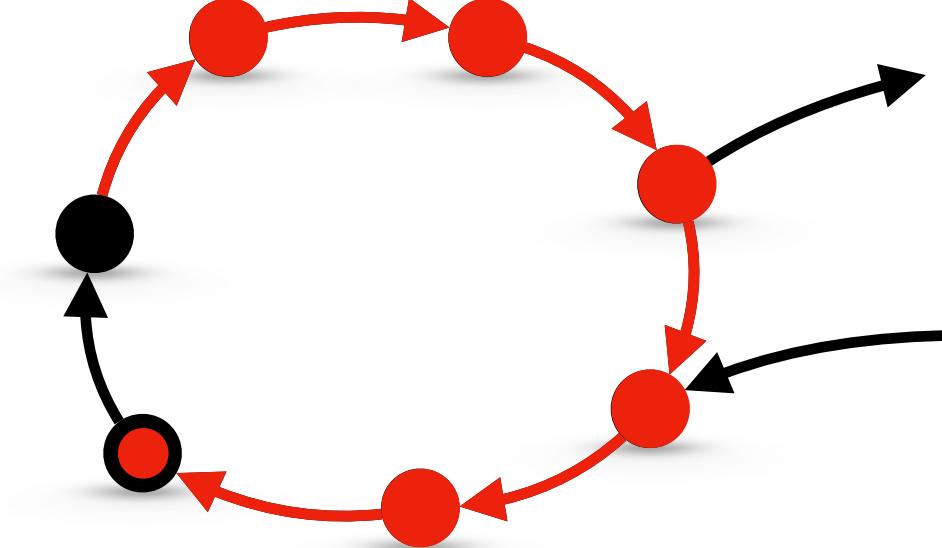
1. **Each *sink node* (with no outgoing edges) must be anchored**
2. **Each *row-diff cycle* must have at least one anchor node in it**



RowDiff

Anchor nodes

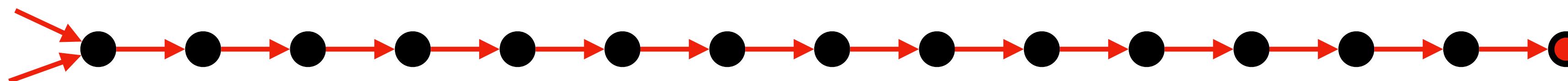
1. **Each *sink node* (with no outgoing edges) must be anchored**
2. **Each *row-diff cycle* must have at least one anchor node in it**



RowDiff

Anchor nodes

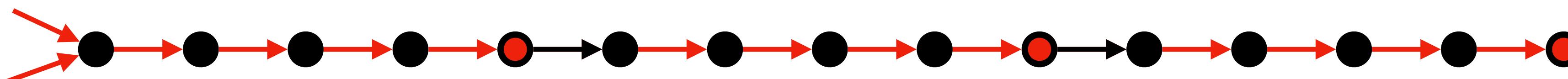
1. **Each *sink node* (with no outgoing edges) must be anchored**
2. **Each *row-diff cycle* must have at least one **anchor node** in it**
3. **Each *row-diff path* is bounded** (to keep a constant query time)



RowDiff

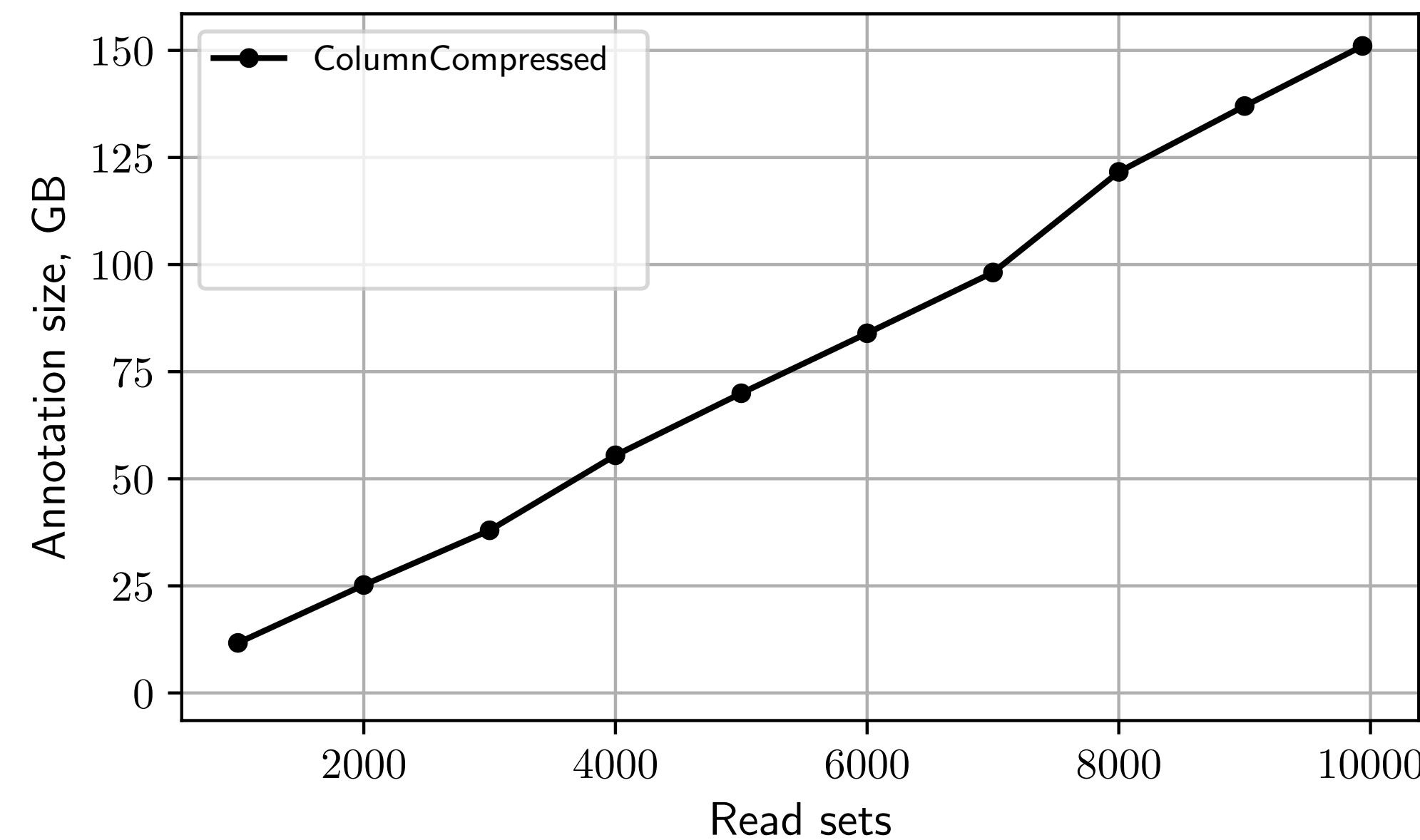
Anchor nodes

1. **Each *sink node* (with no outgoing edges) must be anchored**
2. **Each *row-diff cycle* must have at least one **anchor node** in it**
3. **Each *row-diff path* is bounded** (to keep a constant query time)



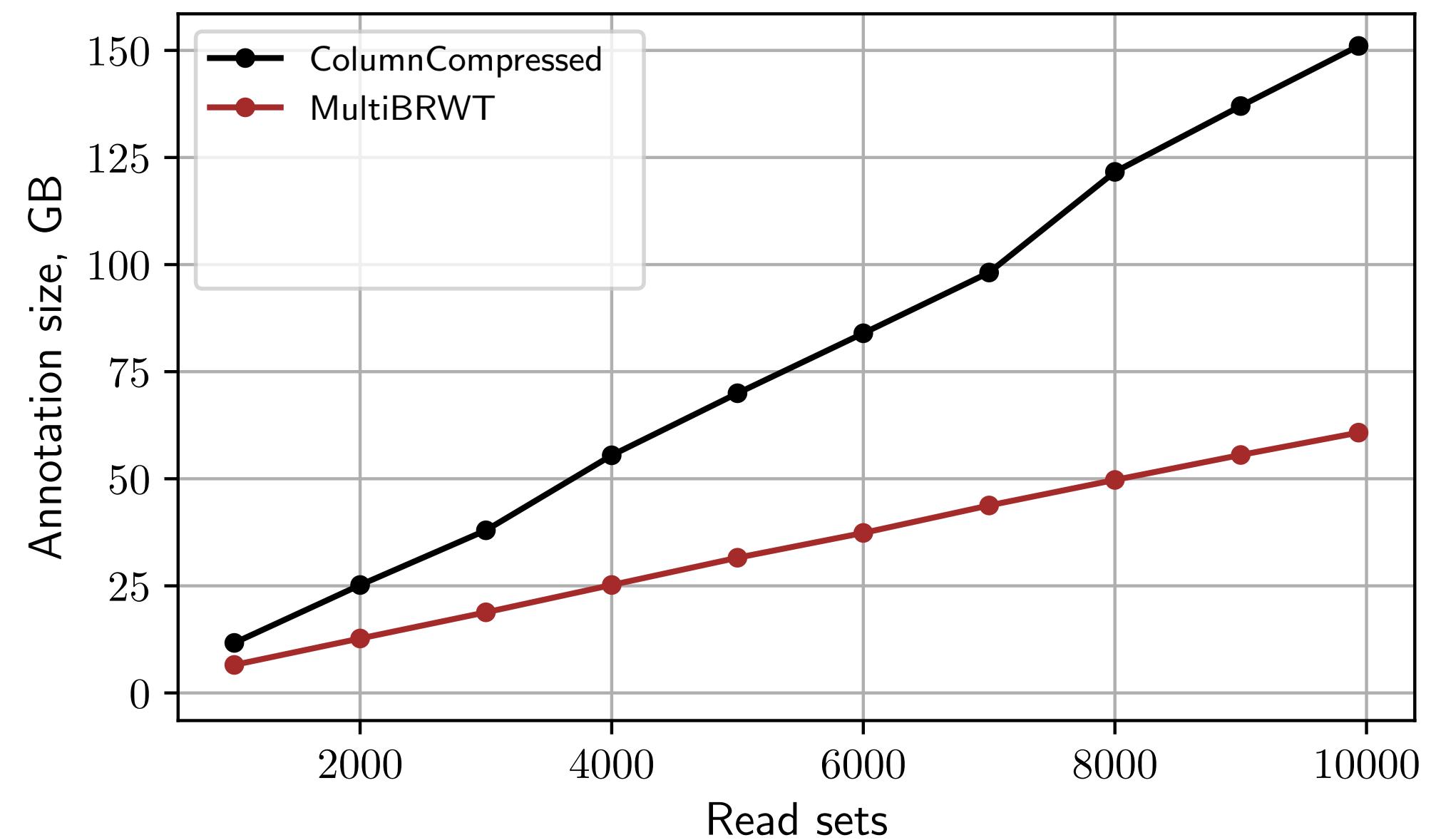
RowDiff: results

Indexing 10,000 (25 TB) human RNA-seq experiments from ENA [Almodaresi et al., 2019]



RowDiff: results

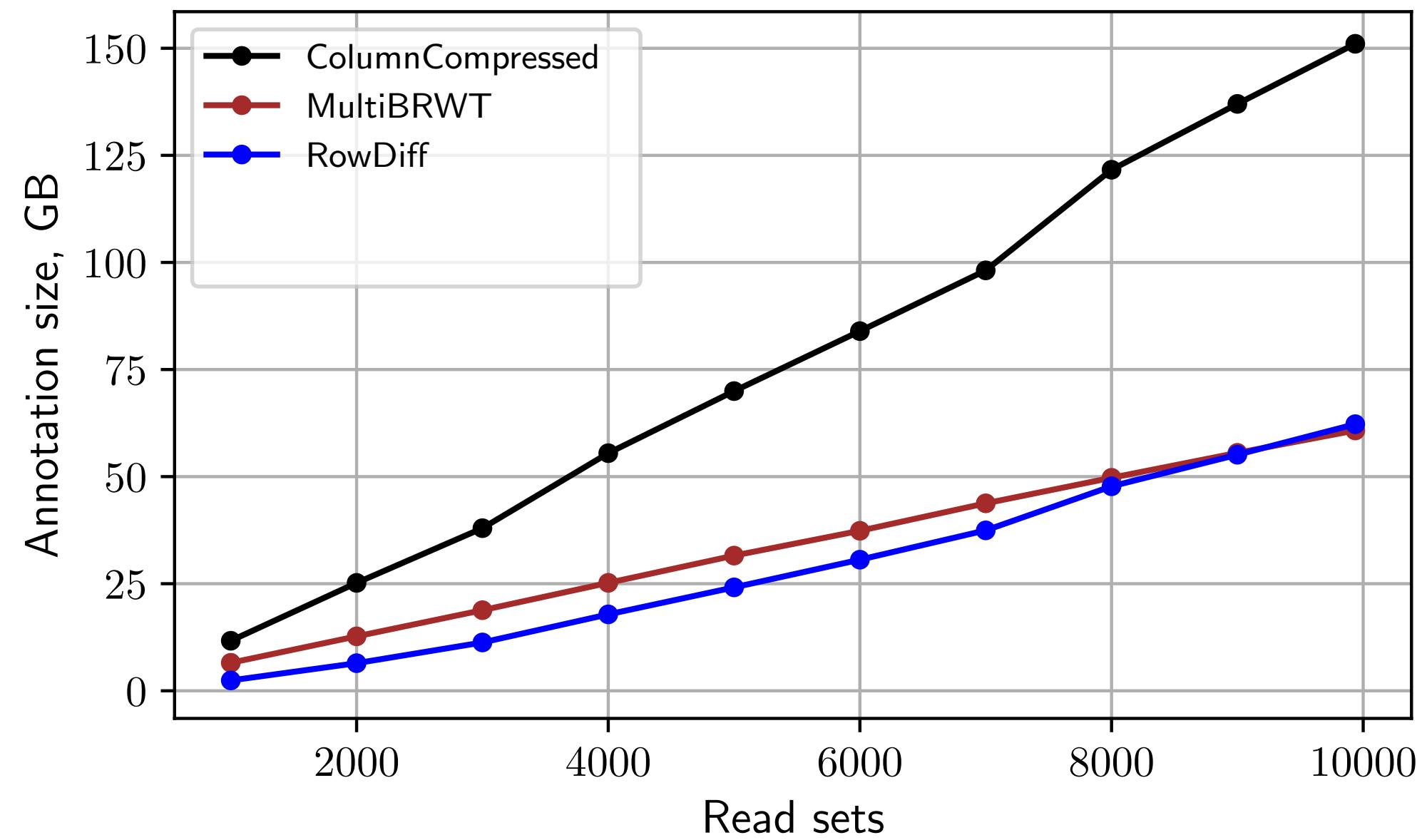
Indexing 10,000 (25 TB) human RNA-seq experiments from ENA [Almodaresi et al., 2019]



- ▶ **Multi-BRWT** exploits similarity of **columns**

RowDiff: results

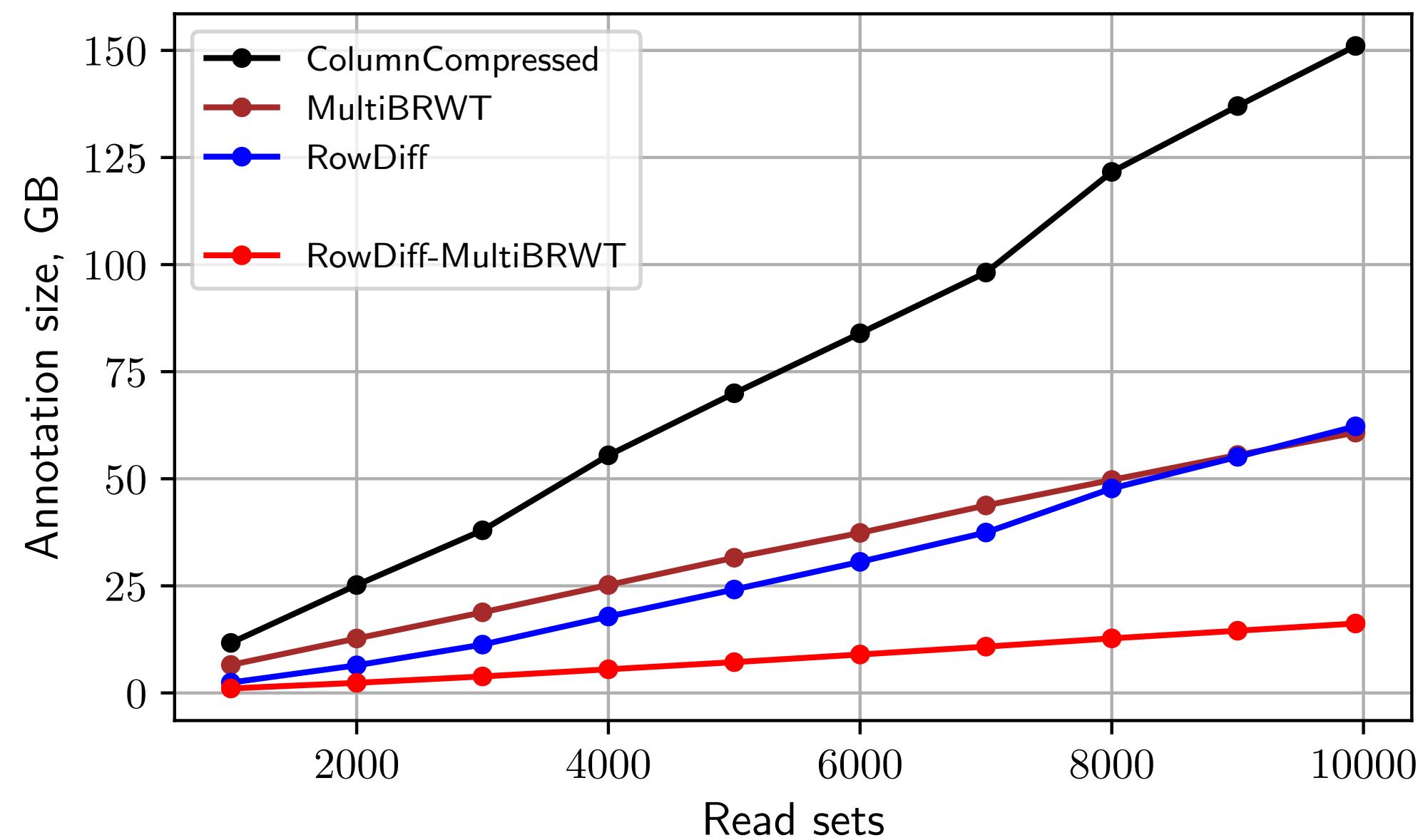
Indexing 10,000 (25 TB) human RNA-seq experiments from ENA [Almodaresi et al., 2019]



- ▶ **Multi-BRWT** exploits similarity of **columns**
- ▶ **RowDiff** exploits similarity of **rows**
 - effectively transforms the matrix
 - can be combined with any representation scheme

RowDiff: results

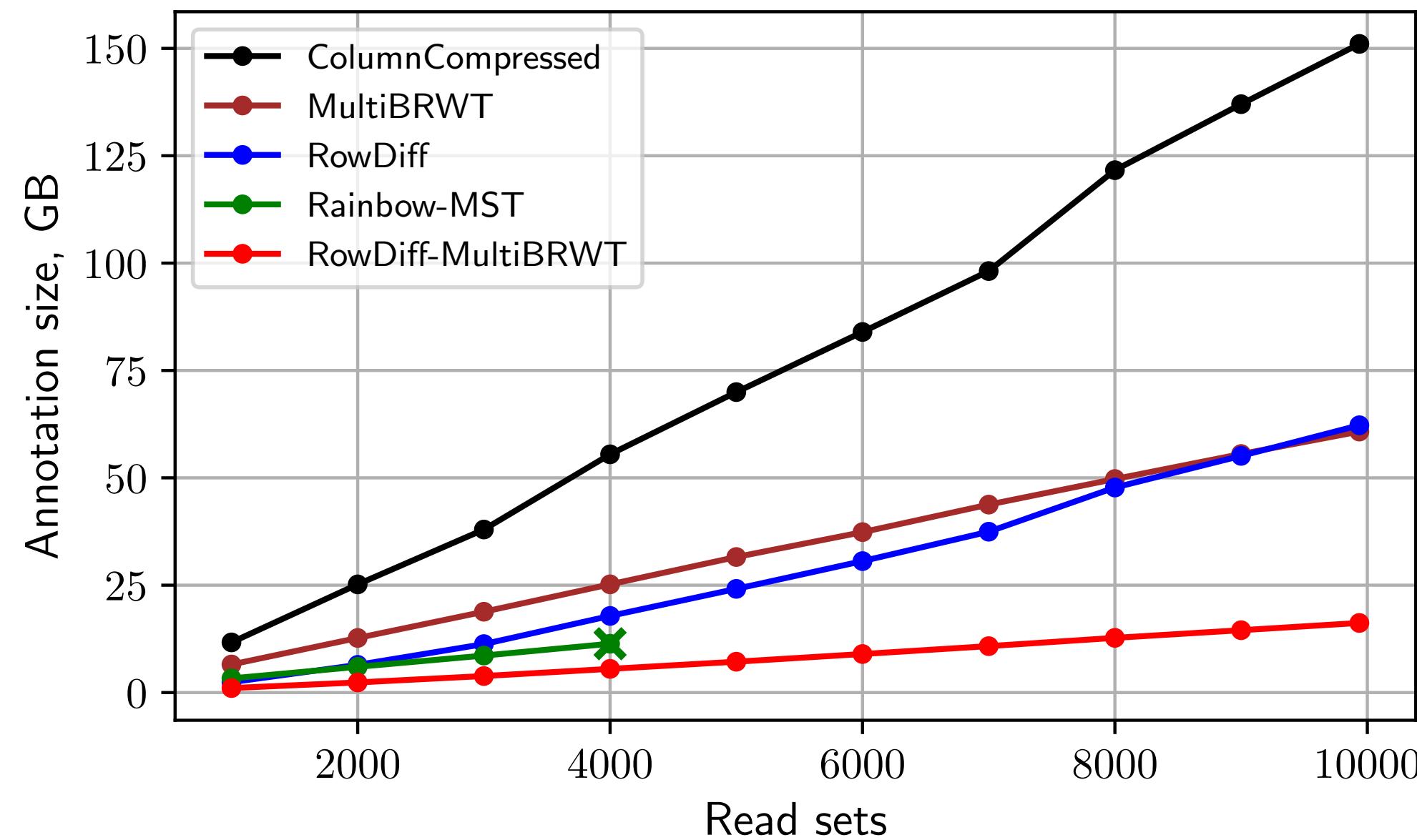
Indexing 10,000 (25 TB) human RNA-seq experiments from ENA [Almodaresi et al., 2019]



- ▶ **Multi-BRWT** exploits similarity of **columns**
- ▶ **RowDiff** exploits similarity of **rows**
 - effectively transforms the matrix
 - can be combined with any representation scheme

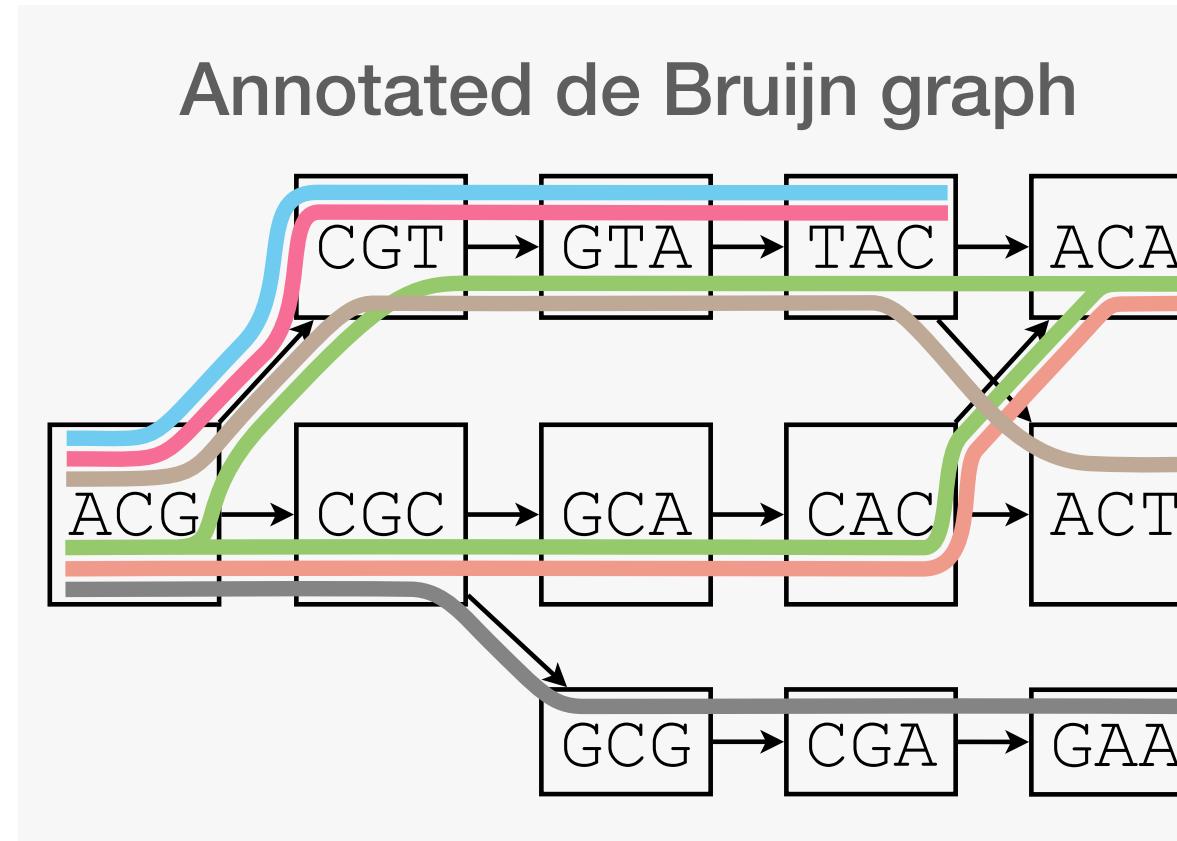
RowDiff: results

Indexing 10,000 (25 TB) human RNA-seq experiments from ENA [Almodaresi et al., 2019]



- ▶ **Multi-BRWT** exploits similarity of **columns**
- ▶ **RowDiff** exploits similarity of **rows**
 - effectively transforms the matrix
 - can be combined with any representation scheme
- ▶ When combined together, **outperform** the **state-of-the-art**

Representing graph annotations



Compressed representation

k-mer dictionary

	Annotation matrix
CAC	0 0 0 1 1 0
GAA	0 0 0 0 0 1
TAC	1 1 1 1 0 0
ACA	0 0 0 1 1 0
ACG	1 1 1 1 1 1
ACT	0 0 1 0 0 0
GCA	0 0 0 1 1 0
GCG	0 0 0 0 0 1
CGA	0 0 0 0 0 1
CGC	0 0 0 1 1 1
CGT	1 1 1 1 0 0
GTA	1 1 1 1 0 0

$\sim 10^{11}$

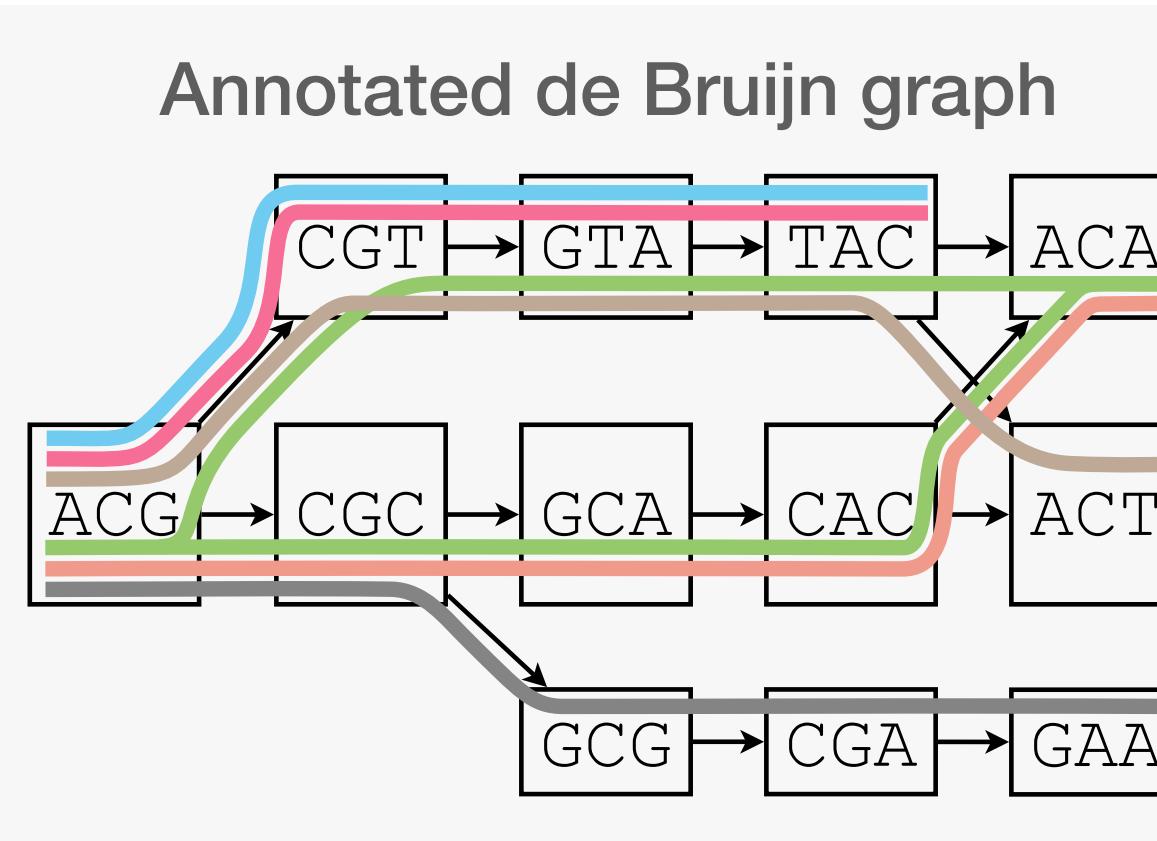
$\sim 10^6$

Challenge:
Represent huge-scale annotations

Typical properties

1. high sparsity
2. similarity of **columns** \Rightarrow **Multi-BRWT**
3. similarity of **rows** \Rightarrow **RowDiff**

Representing graph annotations



Compressed representation

k-mer dictionary

Annotation matrix

	0	1	2	3	4	5
CAC	0	0	0	1	1	0
GAA	0	0	0	0	0	1
TAC	1	1	1	1	0	0
ACA	0	0	0	1	1	0
ACG	1	1	1	1	1	1
ACT	0	0	1	0	0	0
GCA	0	0	0	1	1	0
GCG	0	0	0	0	0	1
CGA	0	0	0	0	0	1
CGC	0	0	0	1	1	1
CGT	1	1	1	1	0	0
GTA	1	1	1	1	0	0

$\sim 10^{11}$

$\sim 10^6$

Challenge:
Represent huge-scale annotations

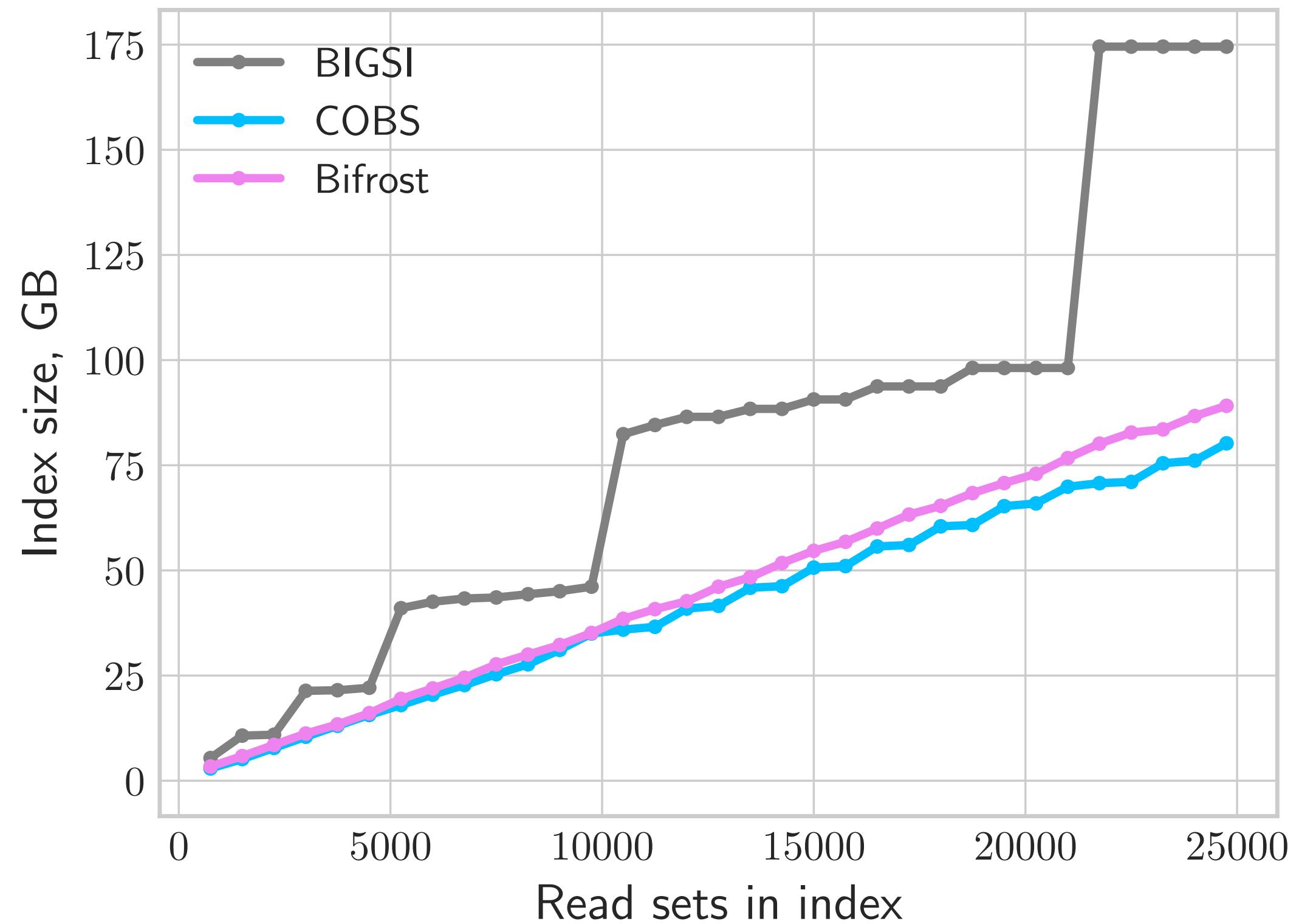


Typical properties

1. high sparsity
2. similarity of **columns** \Rightarrow **Multi-BRWT**
3. similarity of **rows** \Rightarrow **RowDiff**

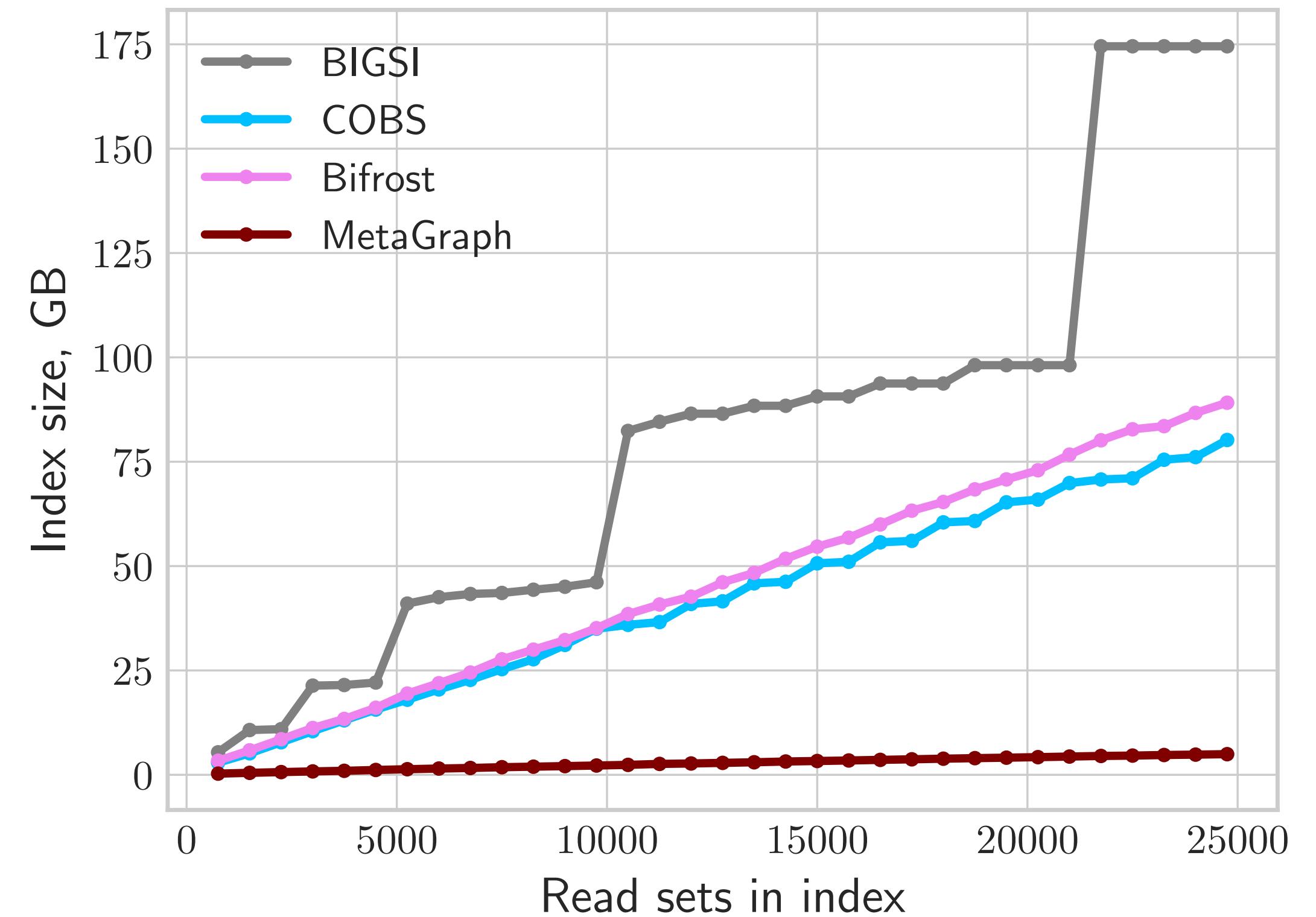
Scalability of MetaGraph

Indexing microbial samples (BIGSI [Bradley *et al.*, 2019] subsets)



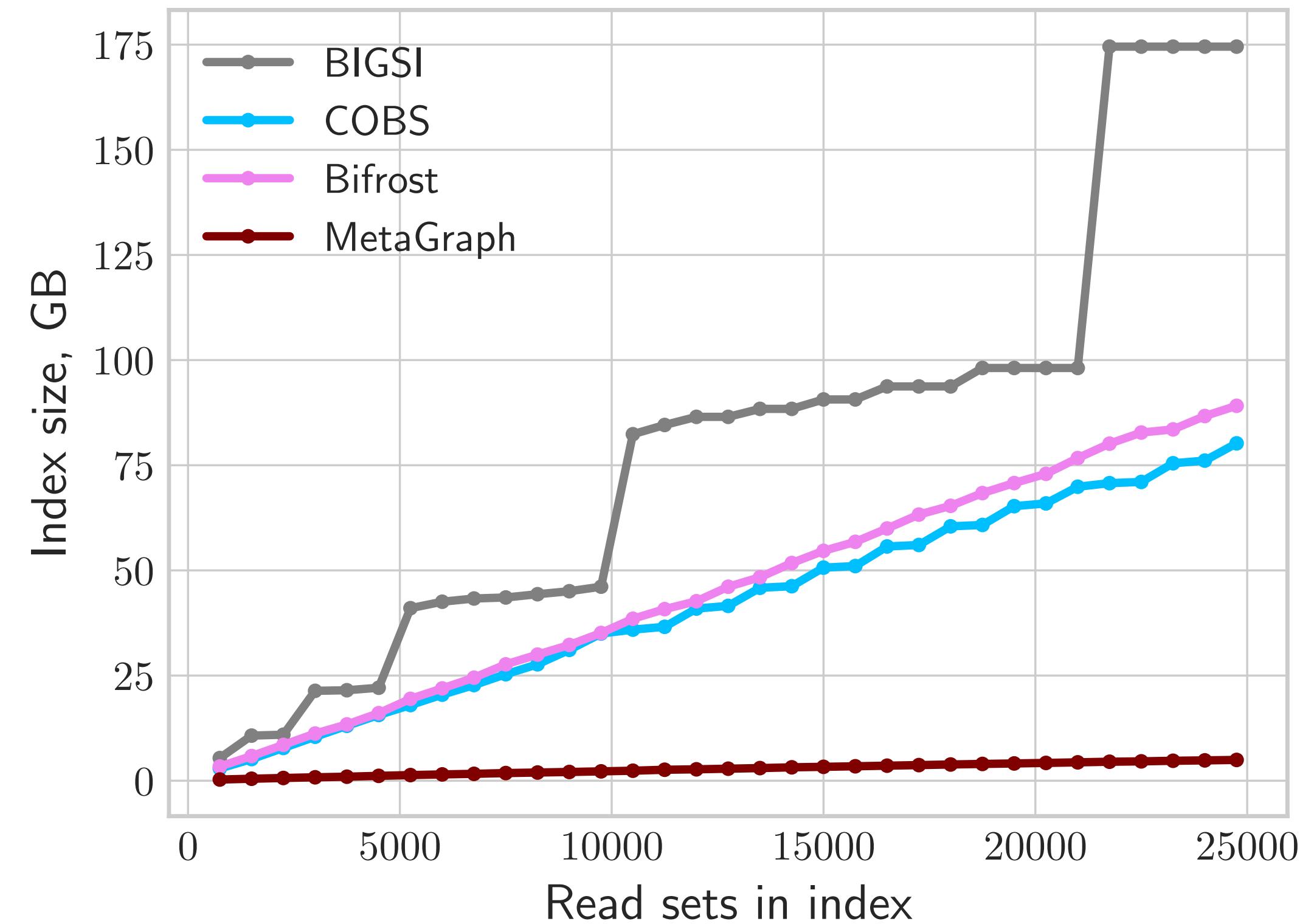
Scalability of MetaGraph

Indexing microbial samples (BIGSI [Bradley *et al.*, 2019] subsets)

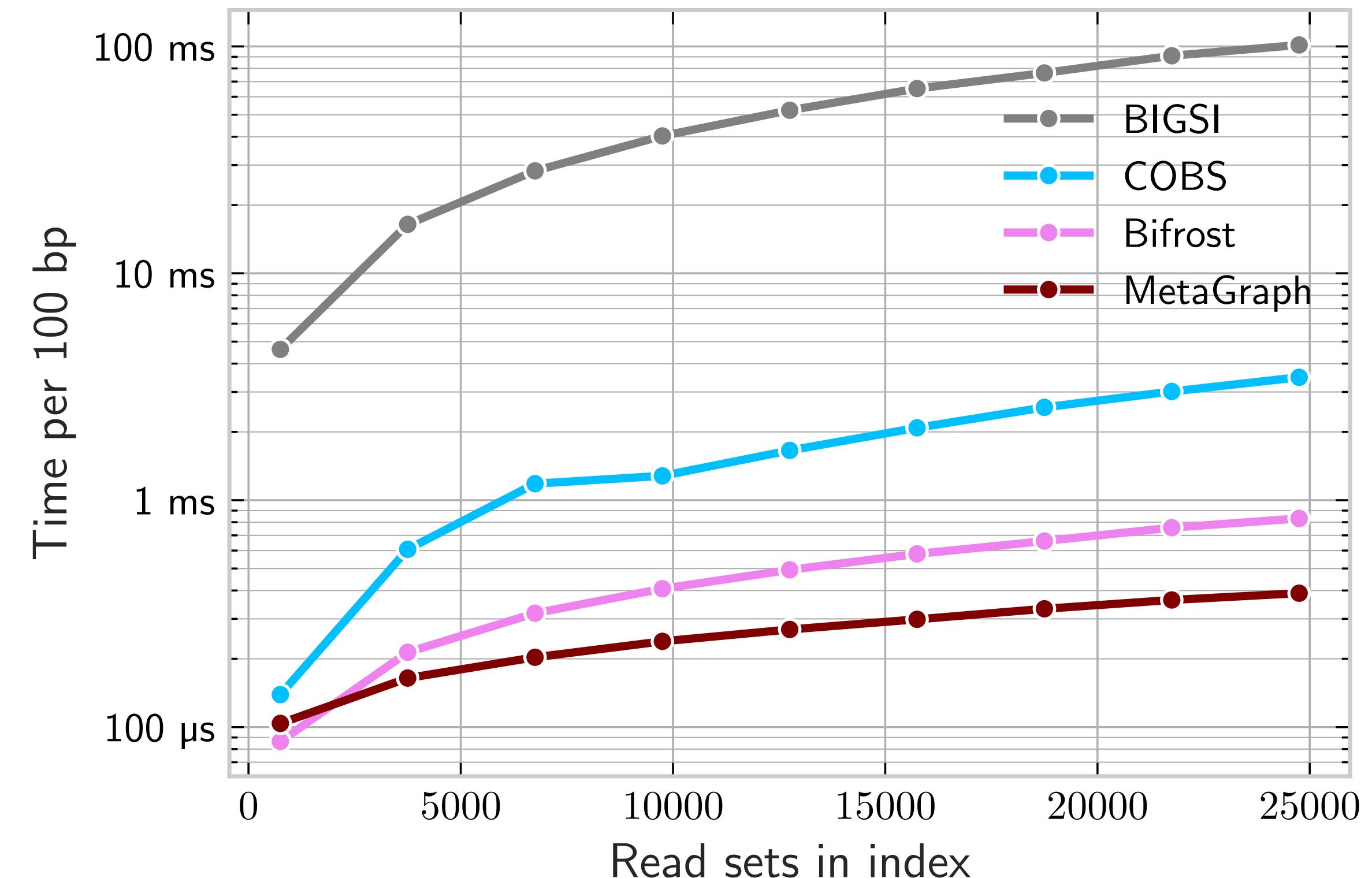


Scalability of MetaGraph

Indexing microbial samples (BIGSI [Bradley et al., 2019] subsets)

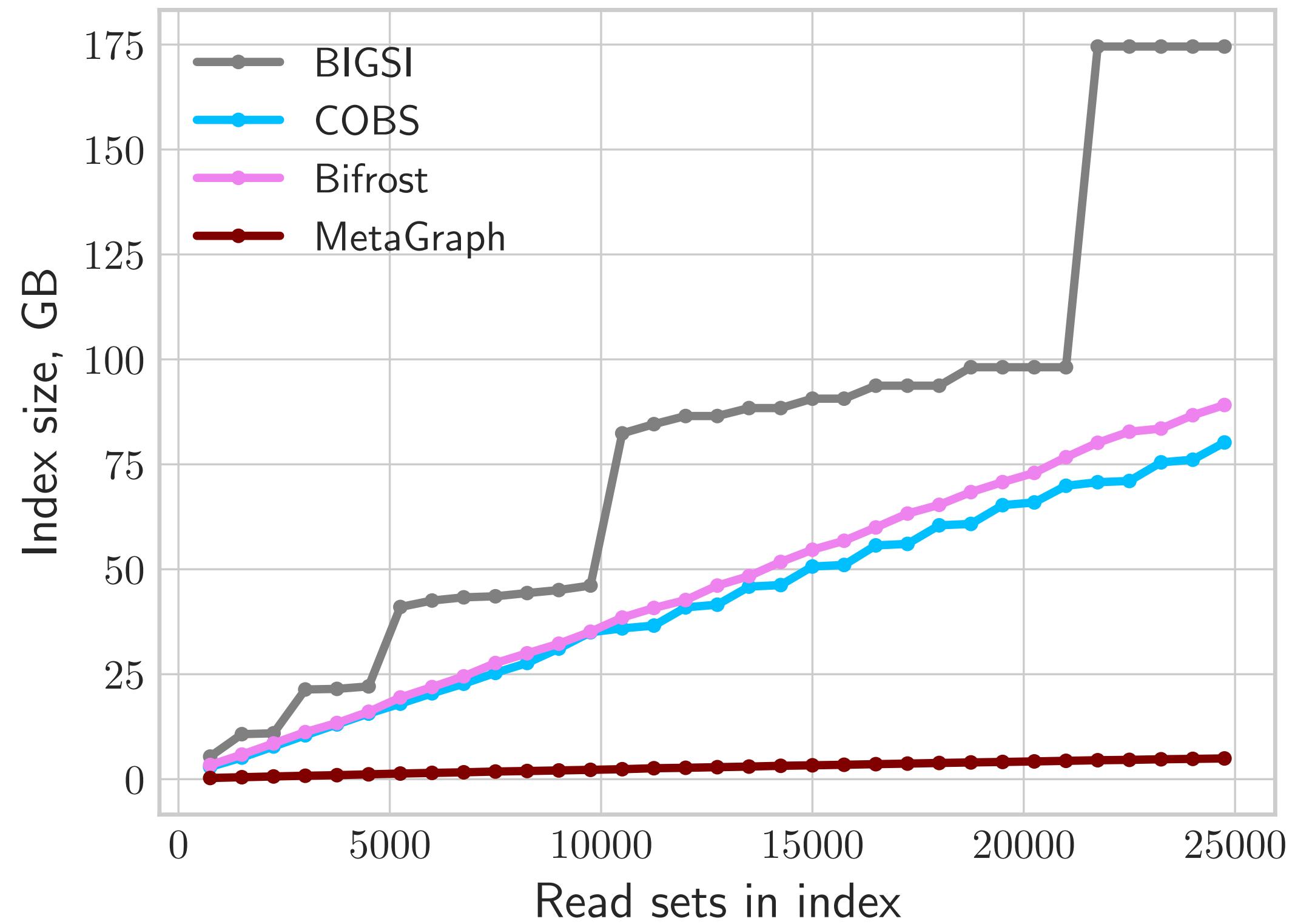


Querying DRR067889

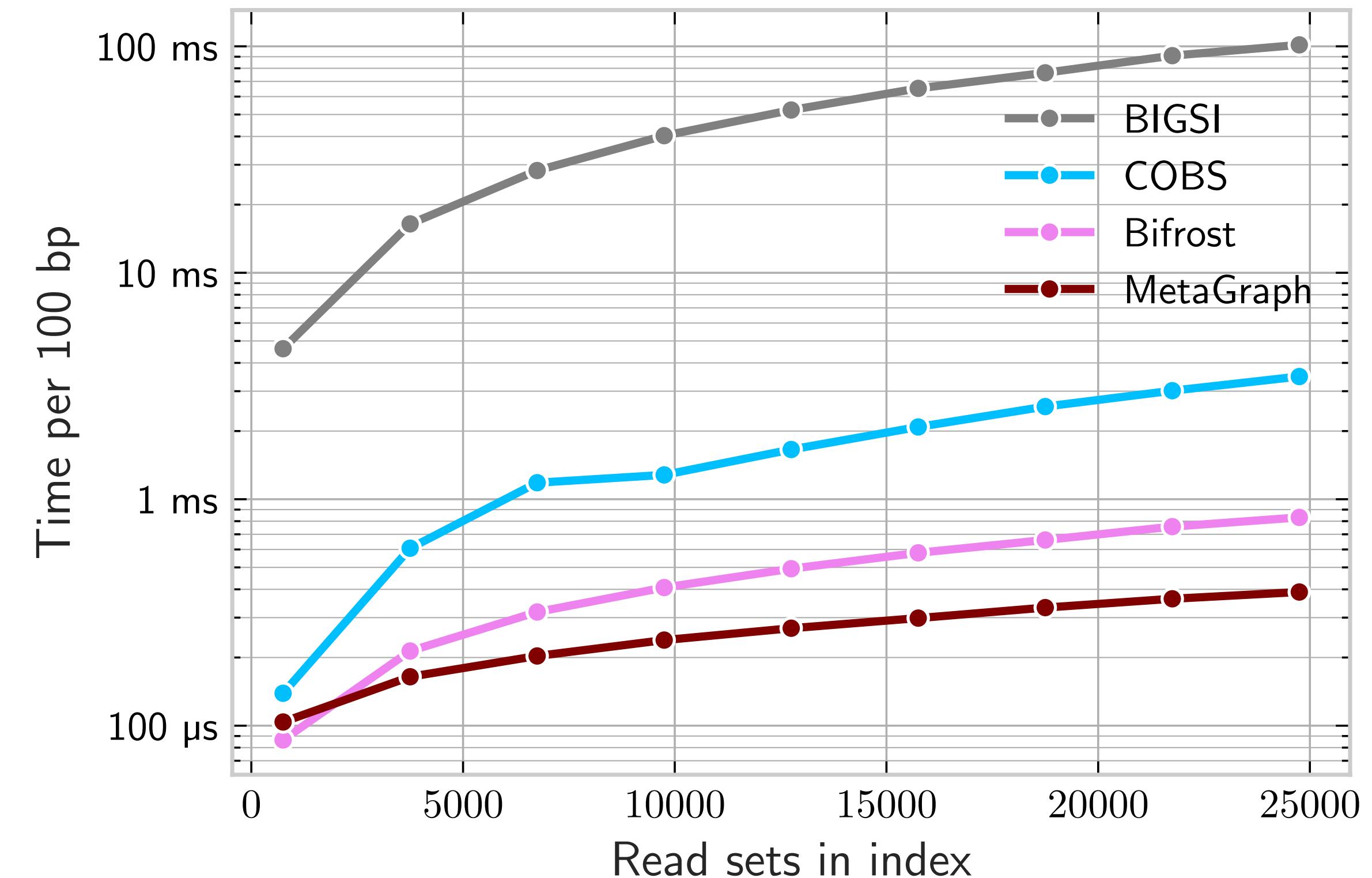


Scalability of MetaGraph

Indexing microbial samples (BIGSI [Bradley et al., 2019] subsets)



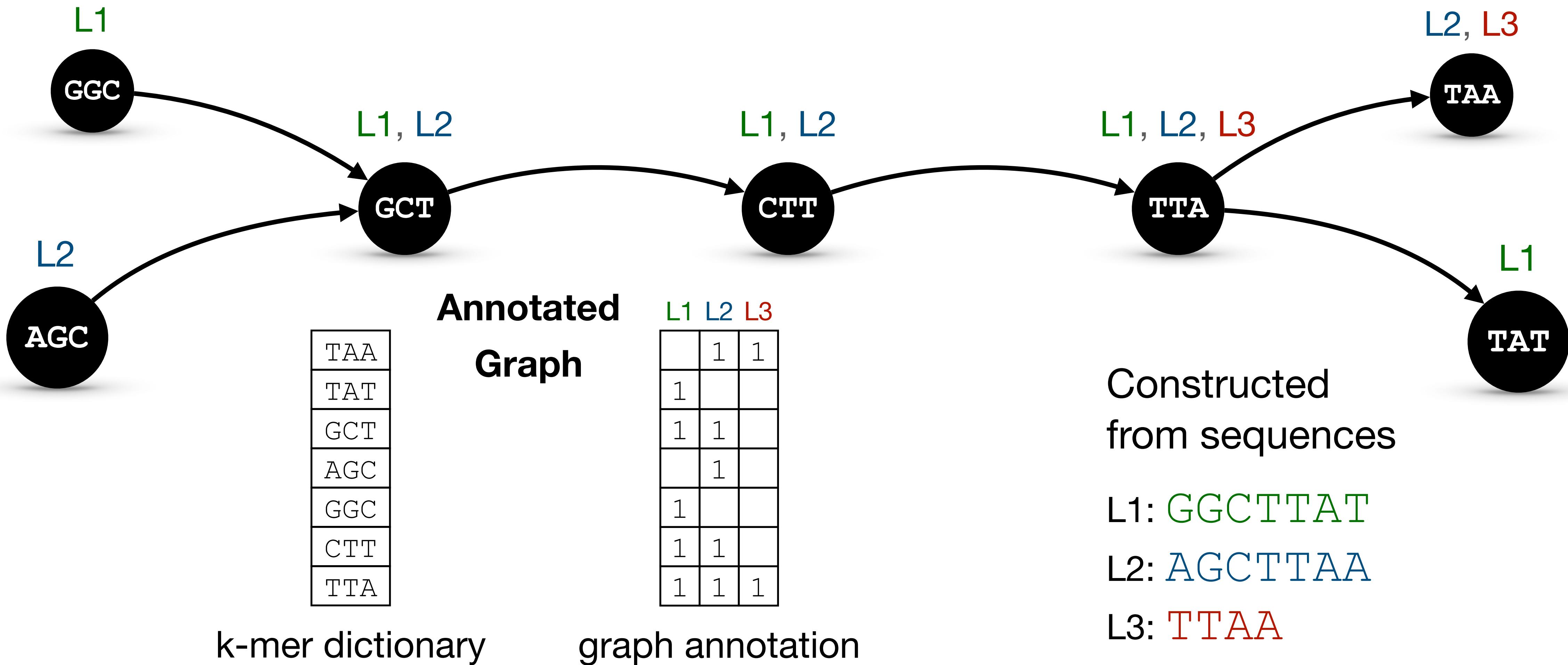
Querying DRR067889



- Use of **succinct data structures** and efficient representation schemes
- Choice of **efficient algorithms** (e.g., batch operations)

From qualitative to quantitative

0. Representing presence/absence



Constructed
from sequences

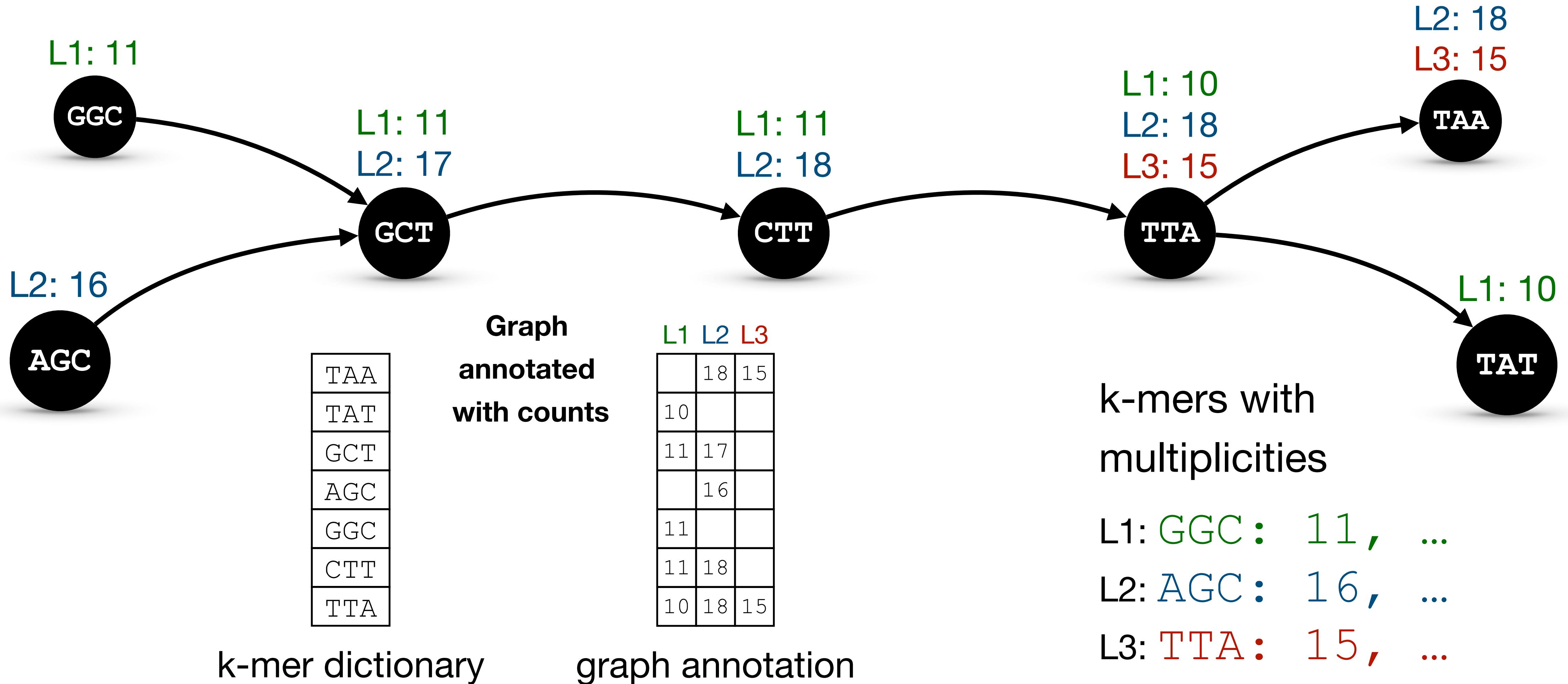
L1: GGCTTAT

L2: AGCTTAA

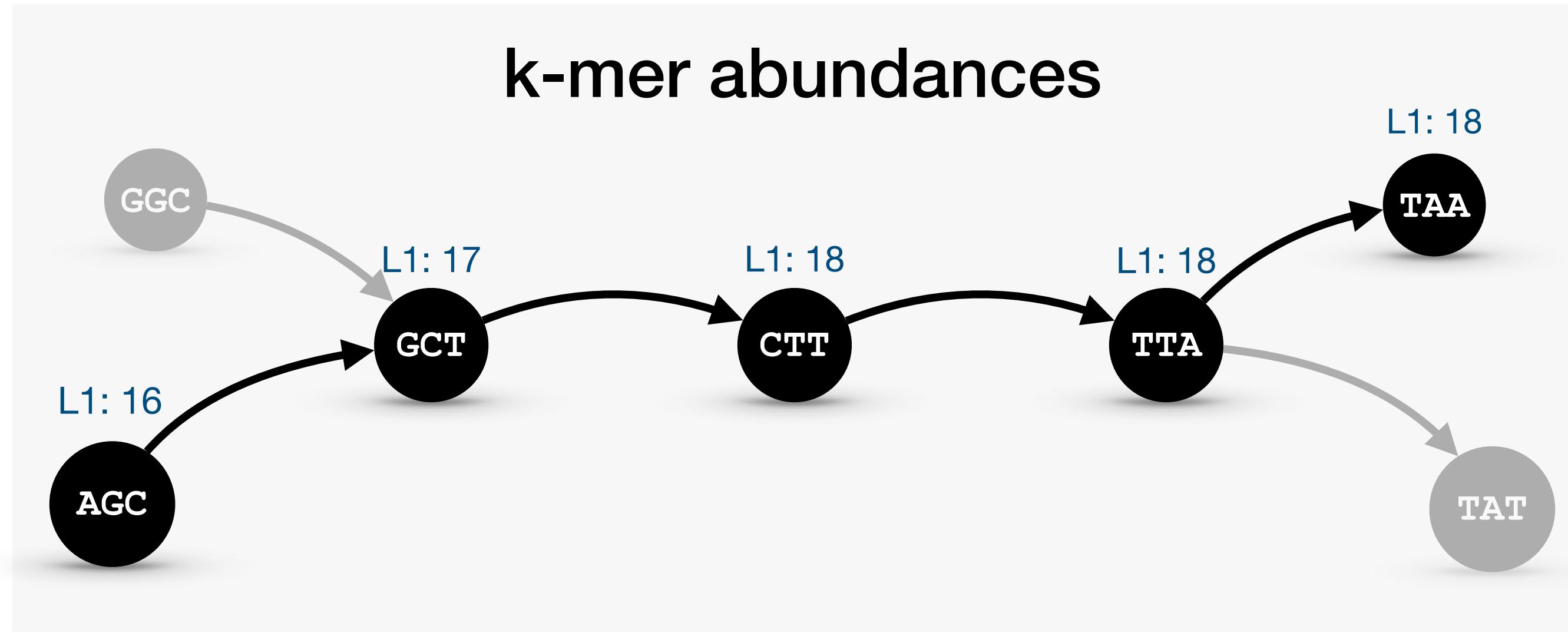
L3: TTAA

From qualitative to quantitative

1. Representing k-mer abundances



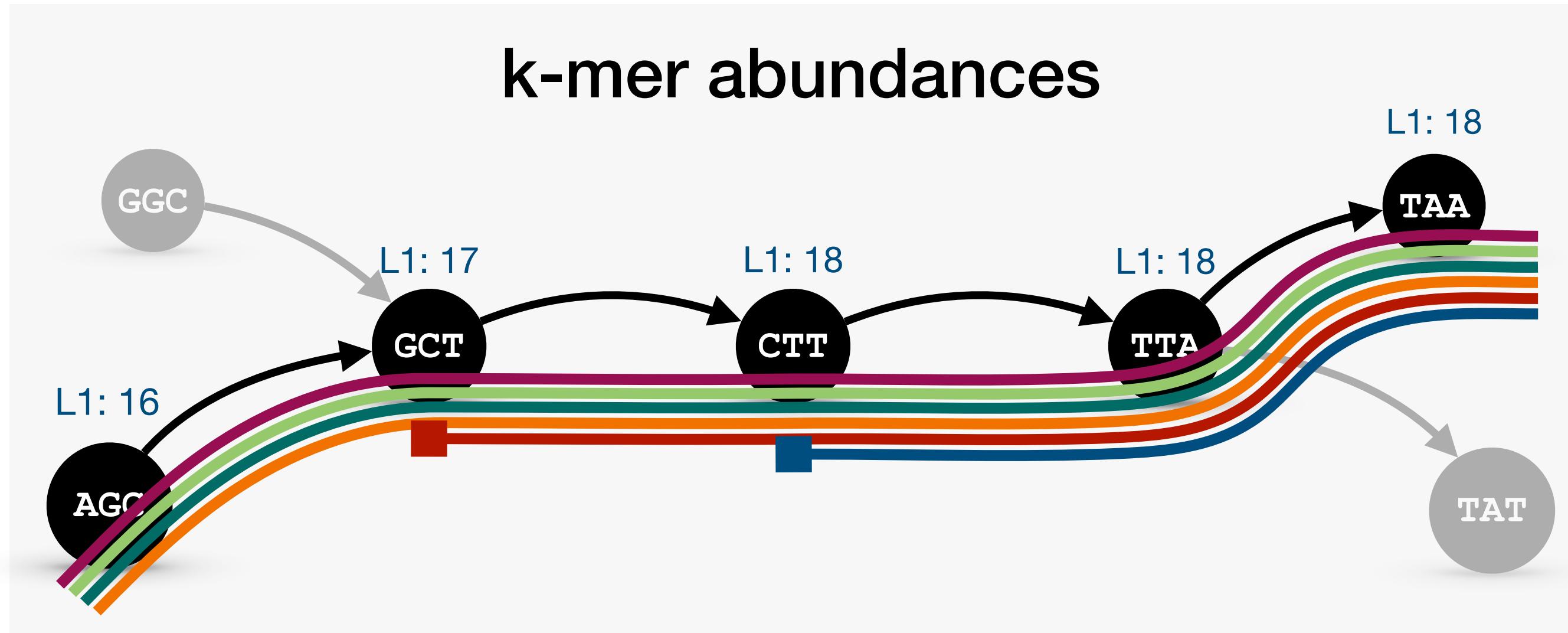
Exploiting regularities



Observe regularities:

Abundances of adjacent k-mers are often similar

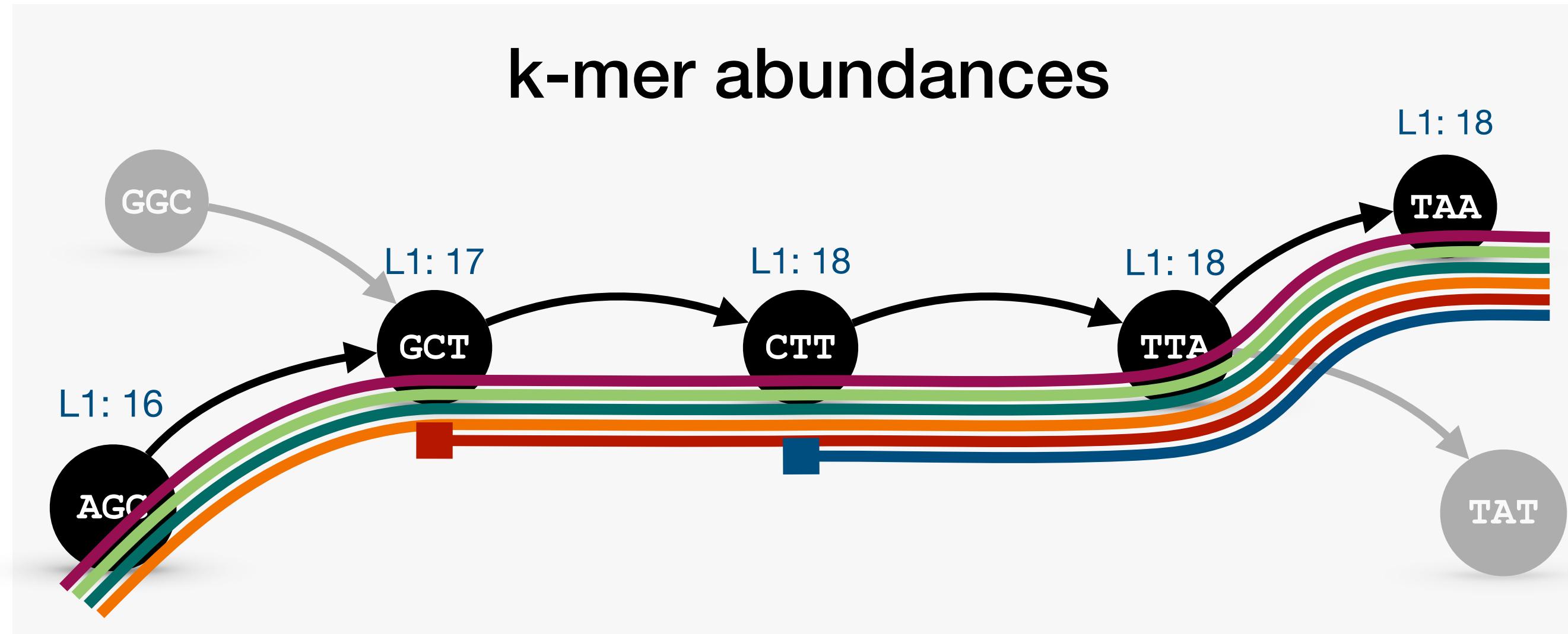
Exploiting regularities



Observe regularities:

Abundances of adjacent k-mers are often similar

Exploiting regularities



Observe regularities:

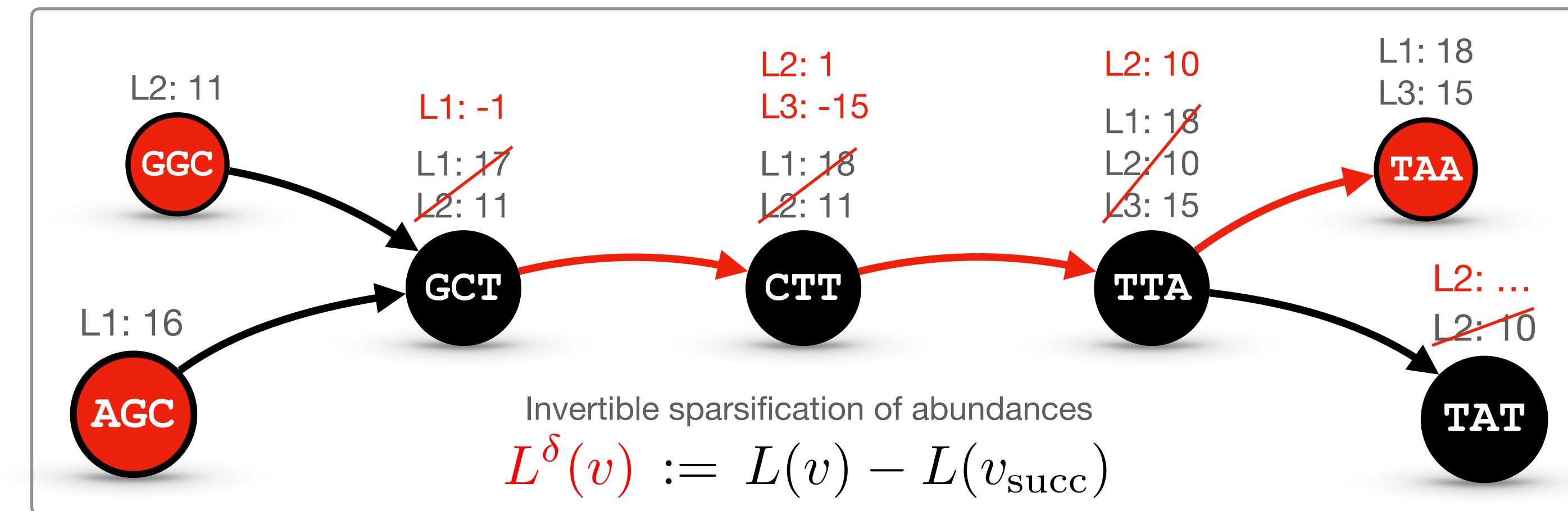
Abundances of adjacent k-mers are often similar

Idea:

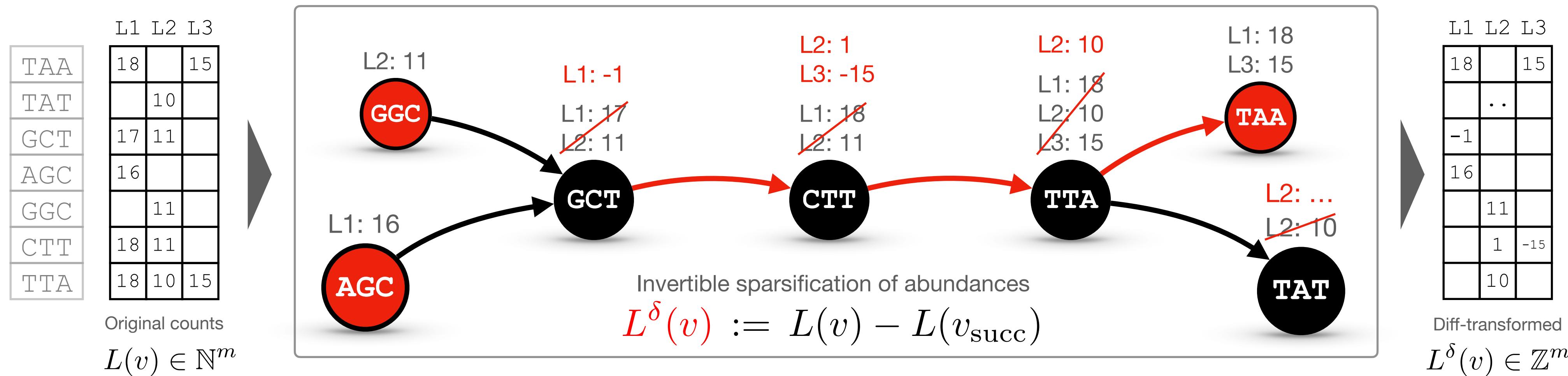
Generalize the RowDiff scheme

Lossless Indexing with Counting de Bruijn Graphs
Mikhail Karasikov, Harun Mustafa, Gunnar Rätsch, André Kahles
Genome Res. 2022; doi: <https://doi.org/10.1101/gr.276607.122>

RowDiff for k-mer abundances



RowDiff for k-mer abundances



Indexing k-mer counts

Lossless Indexing with Counting de Bruijn Graphs

Mikhail Karasikov, Harun Mustafa, Gunnar Rätsch, André Kahles
Genome Res. 2022; doi: <https://doi.org/10.1101/gr.276607.122>

Indexing 2,586 human RNA-Seq read sets [Solomon, Kingsford, 2016].
Querying 100 random human transcripts (\approx 90 kbp).

Method	Index size	Peak RAM	Query time
REINDEER	59 GB	91 GB	56.5 sec
MetaGraph	11 GB	11 GB	17.6 sec

Indexing k-mer counts

Lossless Indexing with Counting de Bruijn Graphs

Mikhail Karasikov, Harun Mustafa, Gunnar Rätsch, André Kahles
Genome Res. 2022; doi: <https://doi.org/10.1101/gr.276607.122>

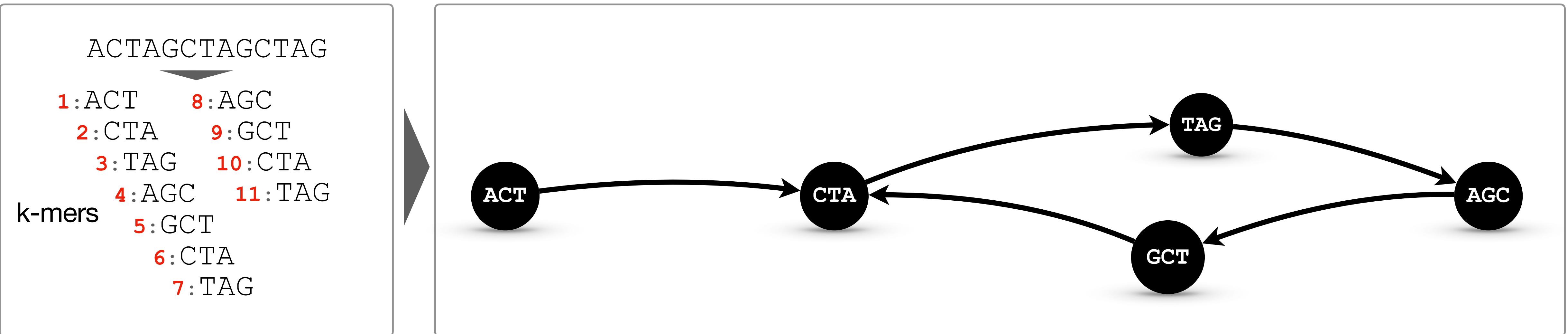
Indexing 2,586 human RNA-Seq read sets [Solomon, Kingsford, 2016].
Querying 100 random human transcripts (\approx 90 kbp).

Method	Index size	Peak RAM	Query time
REINDEER	59 GB	91 GB	56.5 sec
MetaGraph	11 GB	11 GB	17.6 sec

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

From qualitative to quantitative

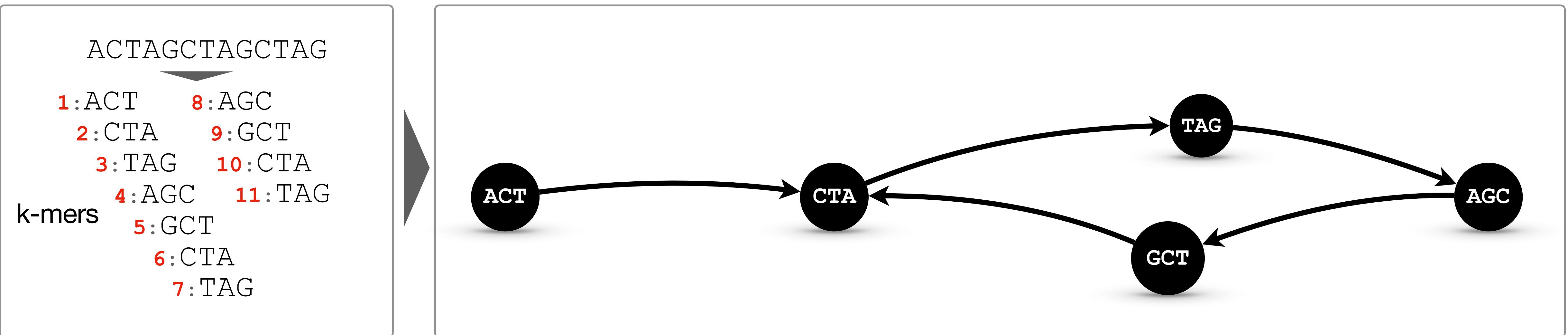
2. Representing k-mer coordinates



De Bruijn graph

From qualitative to quantitative

2. Representing k-mer coordinates

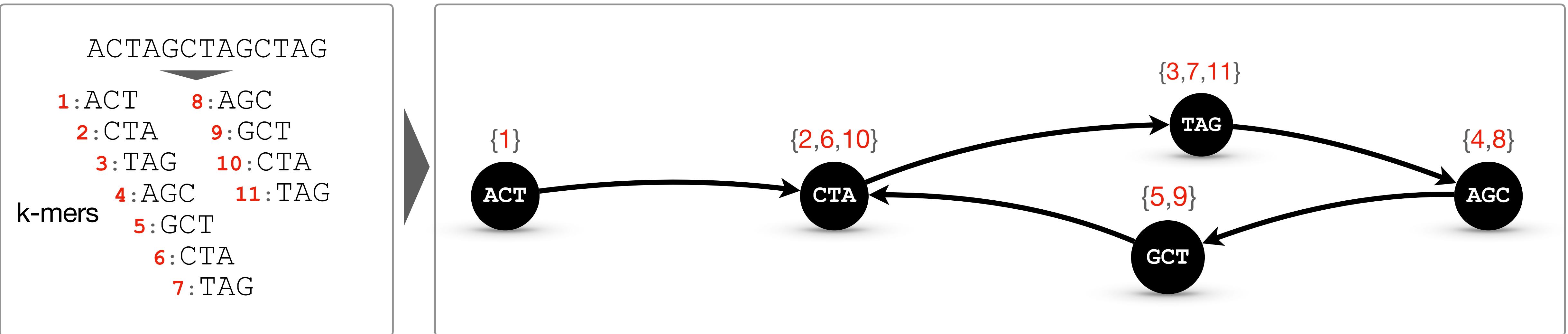


**Not invertible
representation**

De Bruijn graph

From qualitative to quantitative

2. Representing k-mer coordinates

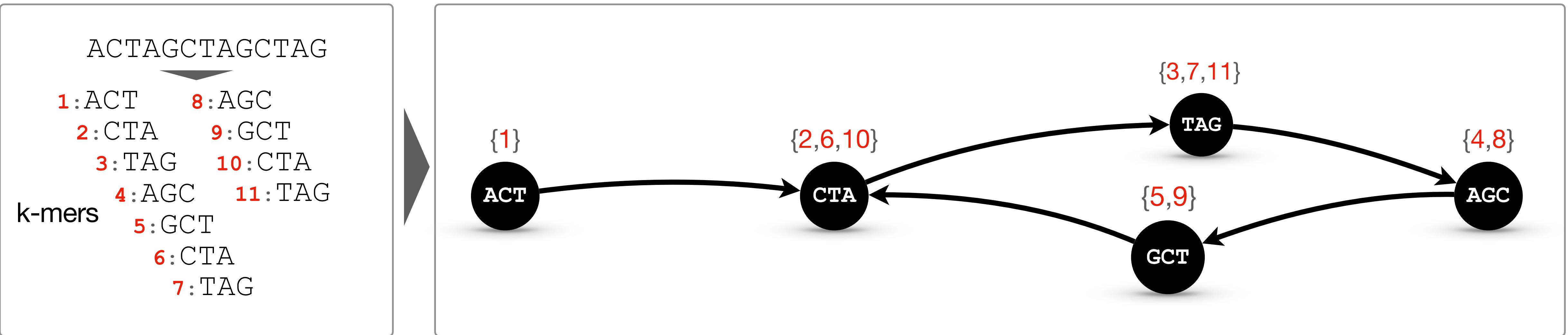


Invertible

De Bruijn graph
with k-mer coordinates

From qualitative to quantitative

2. Representing k-mer coordinates



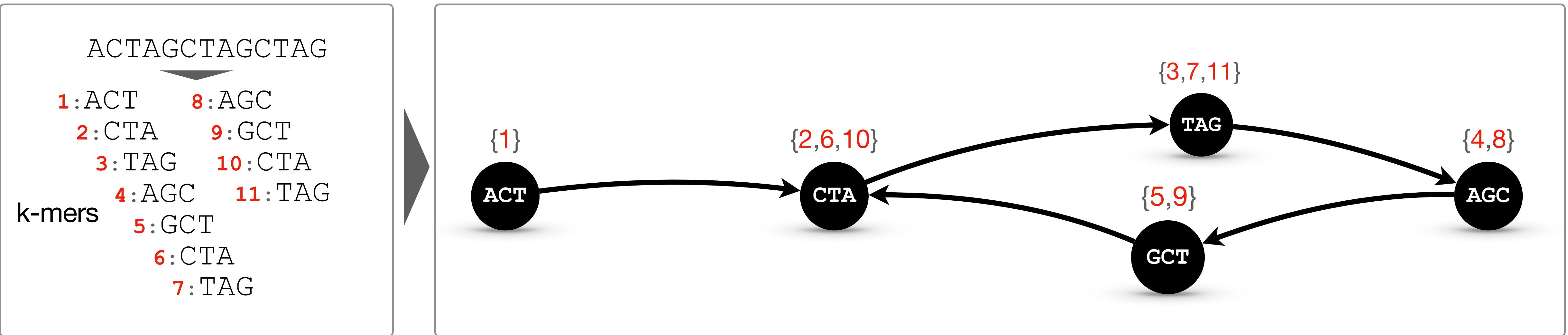
Encoding ***k-mer coordinates*** allows

- reconstructing original sequences
(hence, **lossless sequence representation**)

De Bruijn graph
with **k-mer coordinates**

From qualitative to quantitative

2. Representing k-mer coordinates



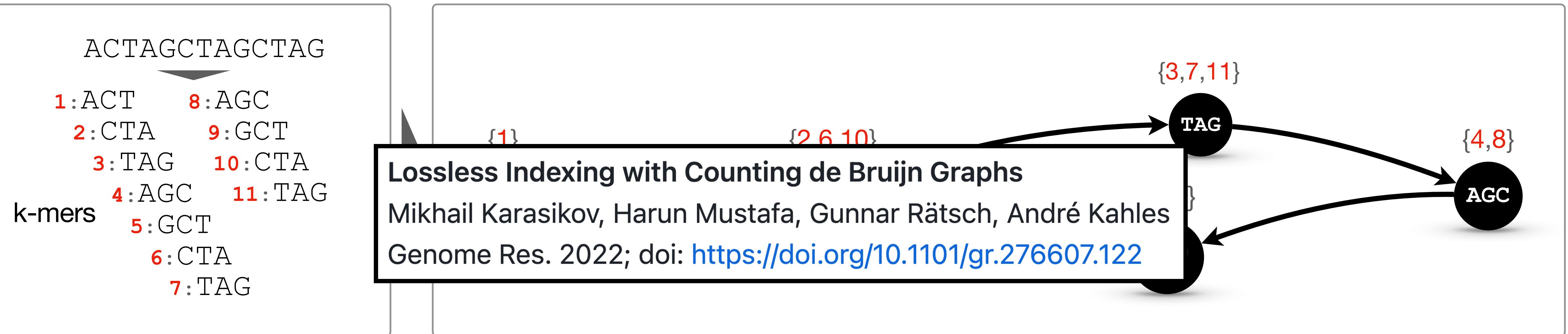
Encoding ***k-mer coordinates*** allows

- reconstructing original sequences
(hence, **lossless sequence representation**)
- performing exact sequence alignment

De Bruijn graph
with k-mer coordinates

From qualitative to quantitative

2. Representing k-mer coordinates



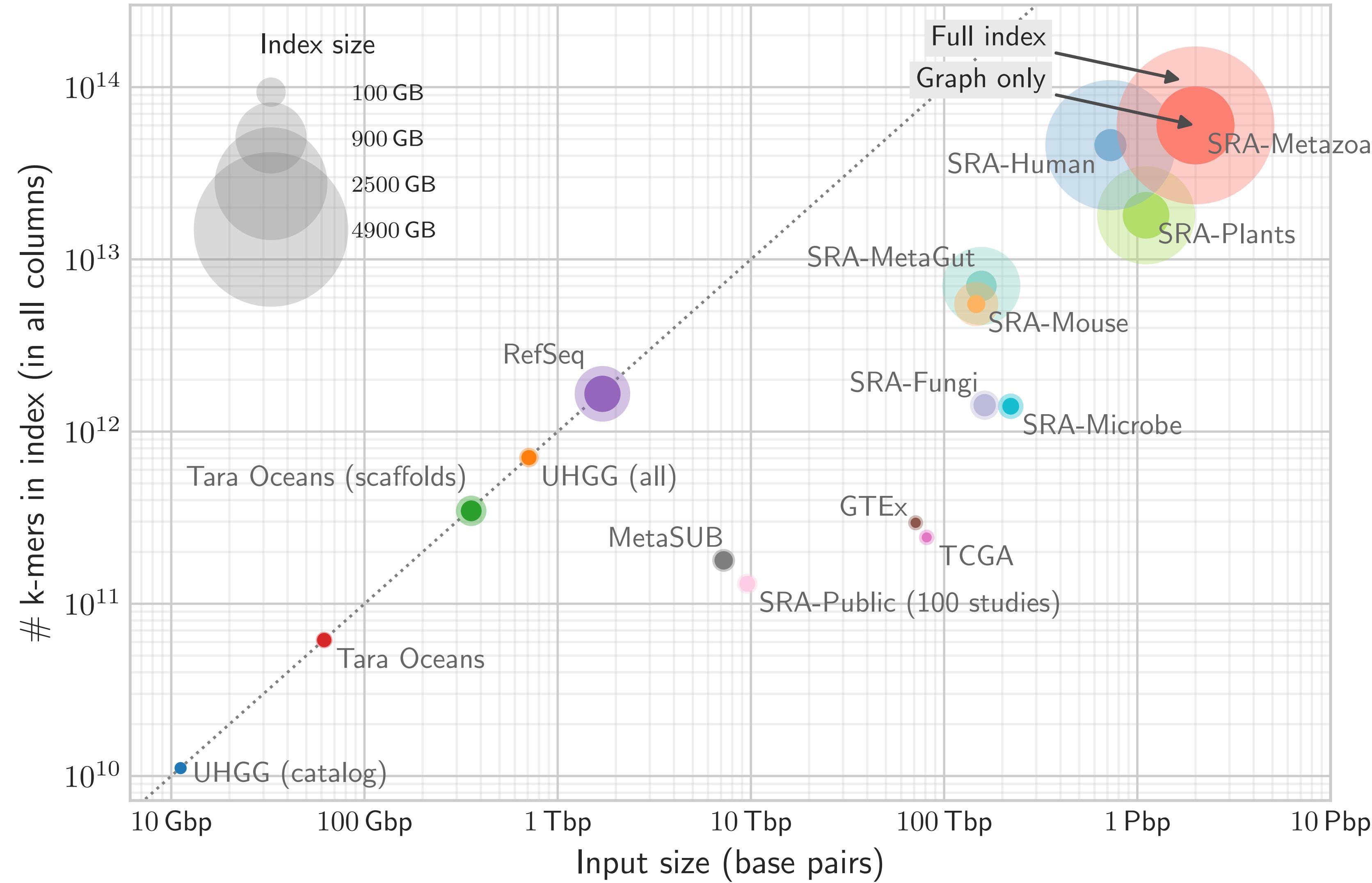
Encoding ***k-mer coordinates*** allows

- reconstructing original sequences
(hence, **lossless sequence representation**)
- performing exact sequence alignment

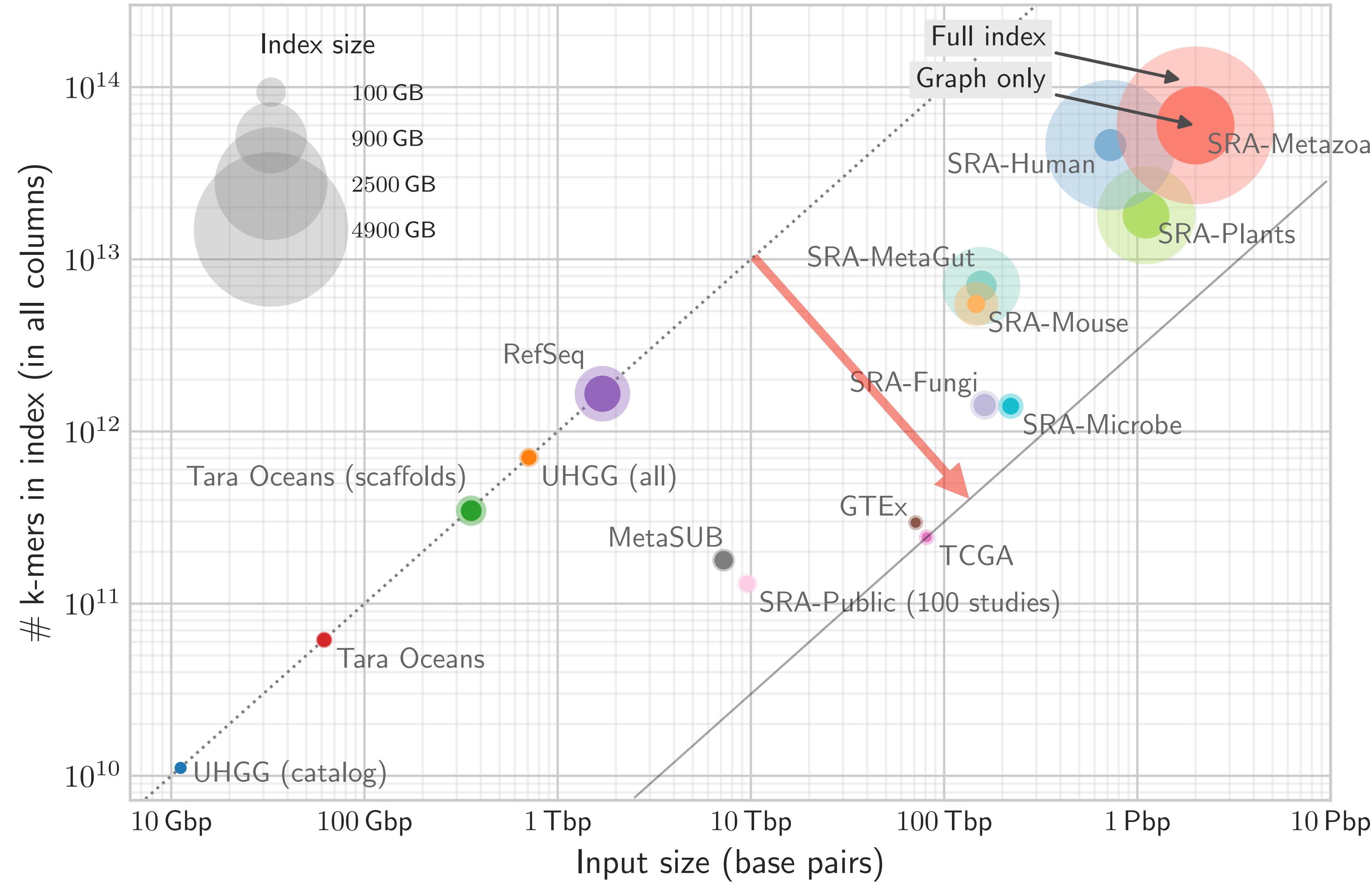
**De Bruijn graph
with k-mer coordinates**

Indexing large sequence archives

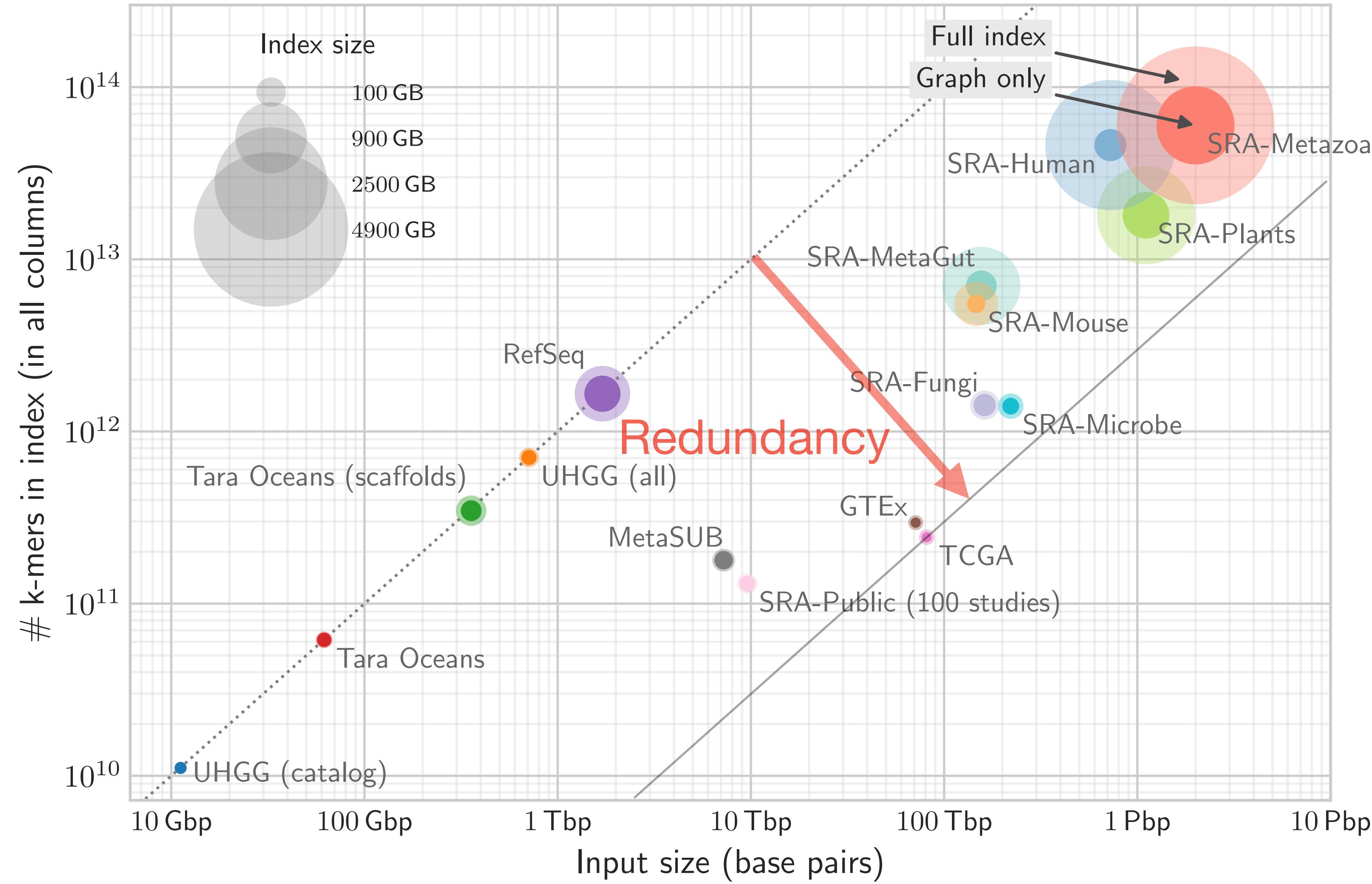
Indexing at Petabase-scale



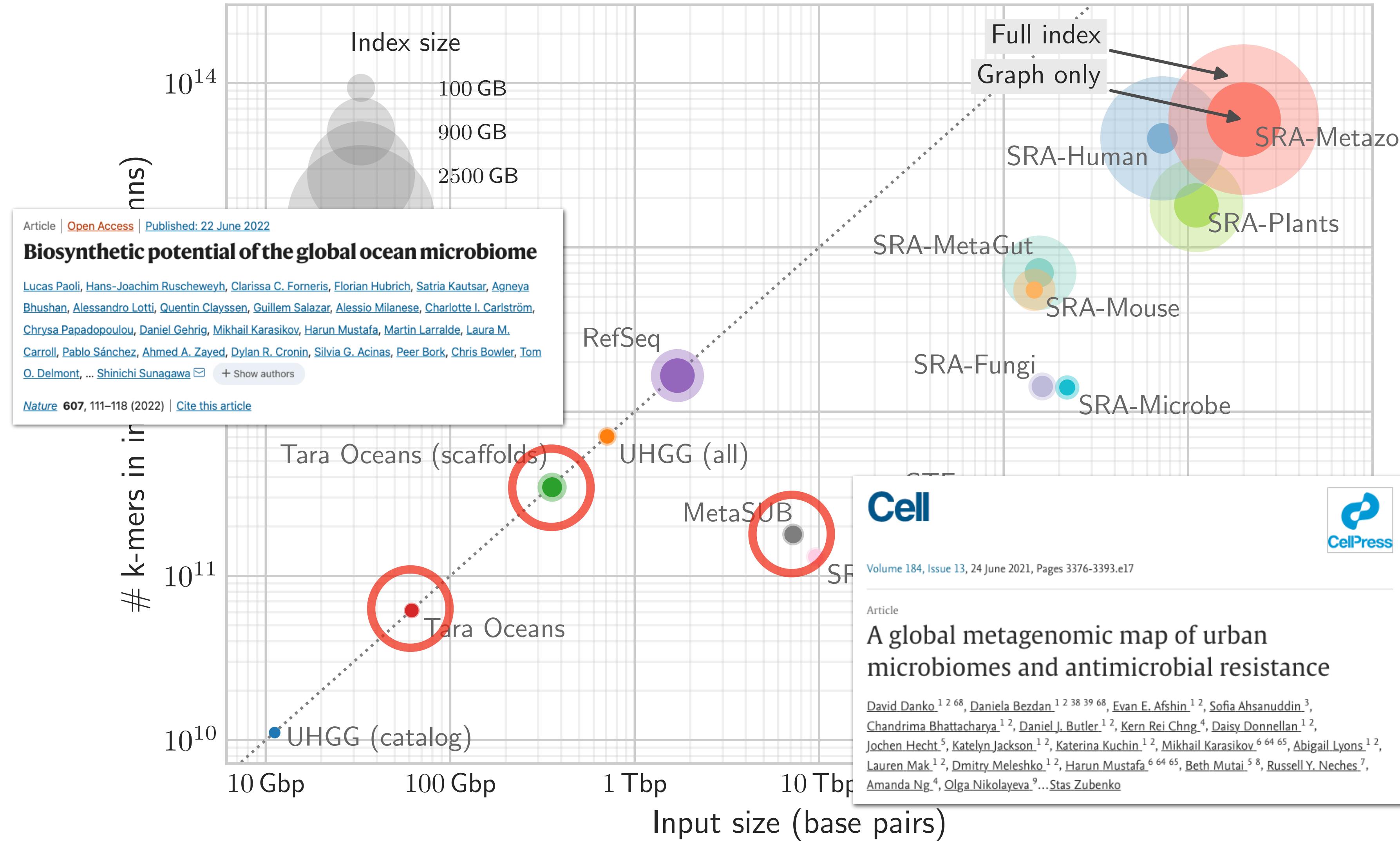
Indexing at Petabase-scale



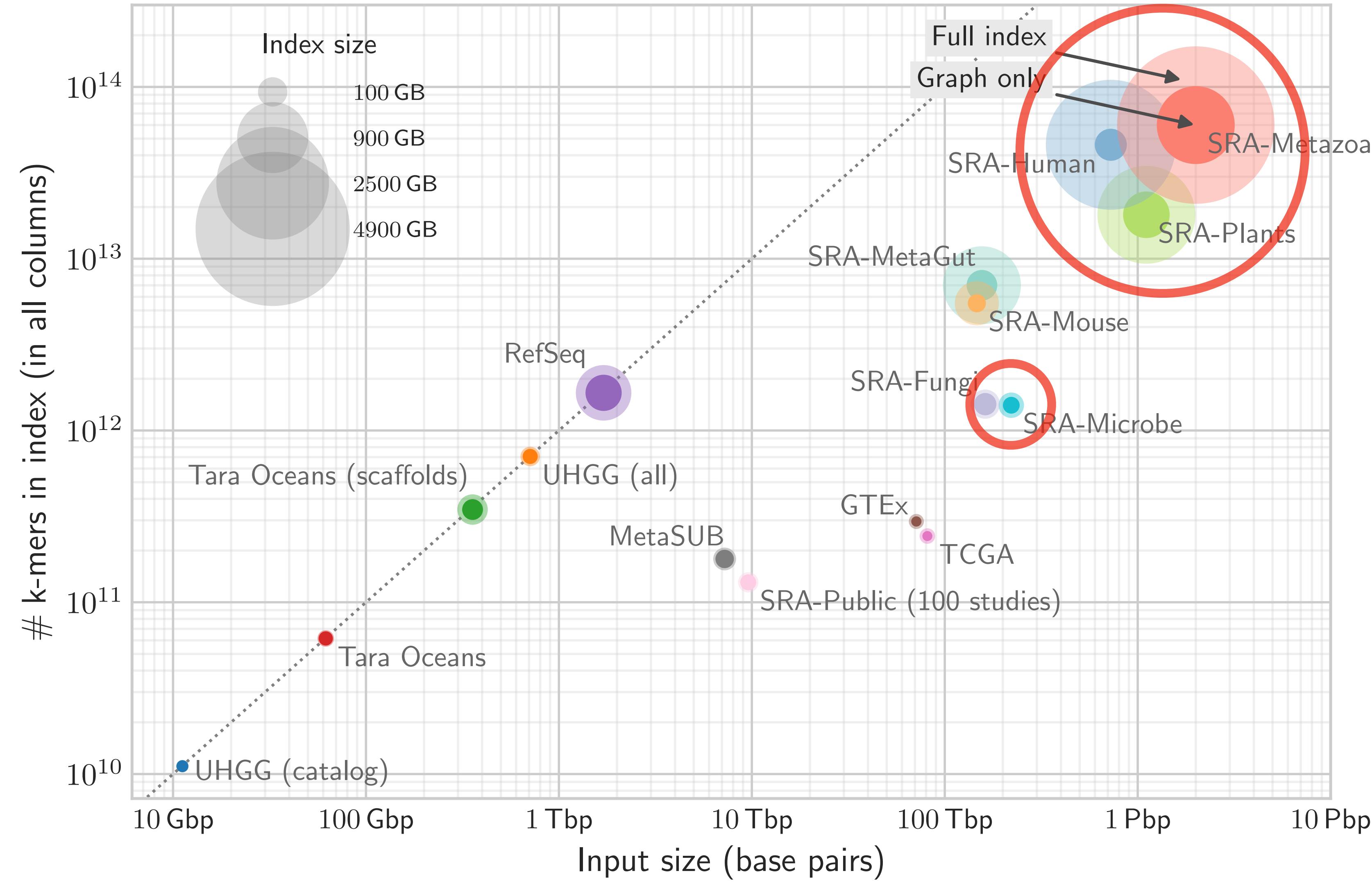
Indexing at Petabase-scale



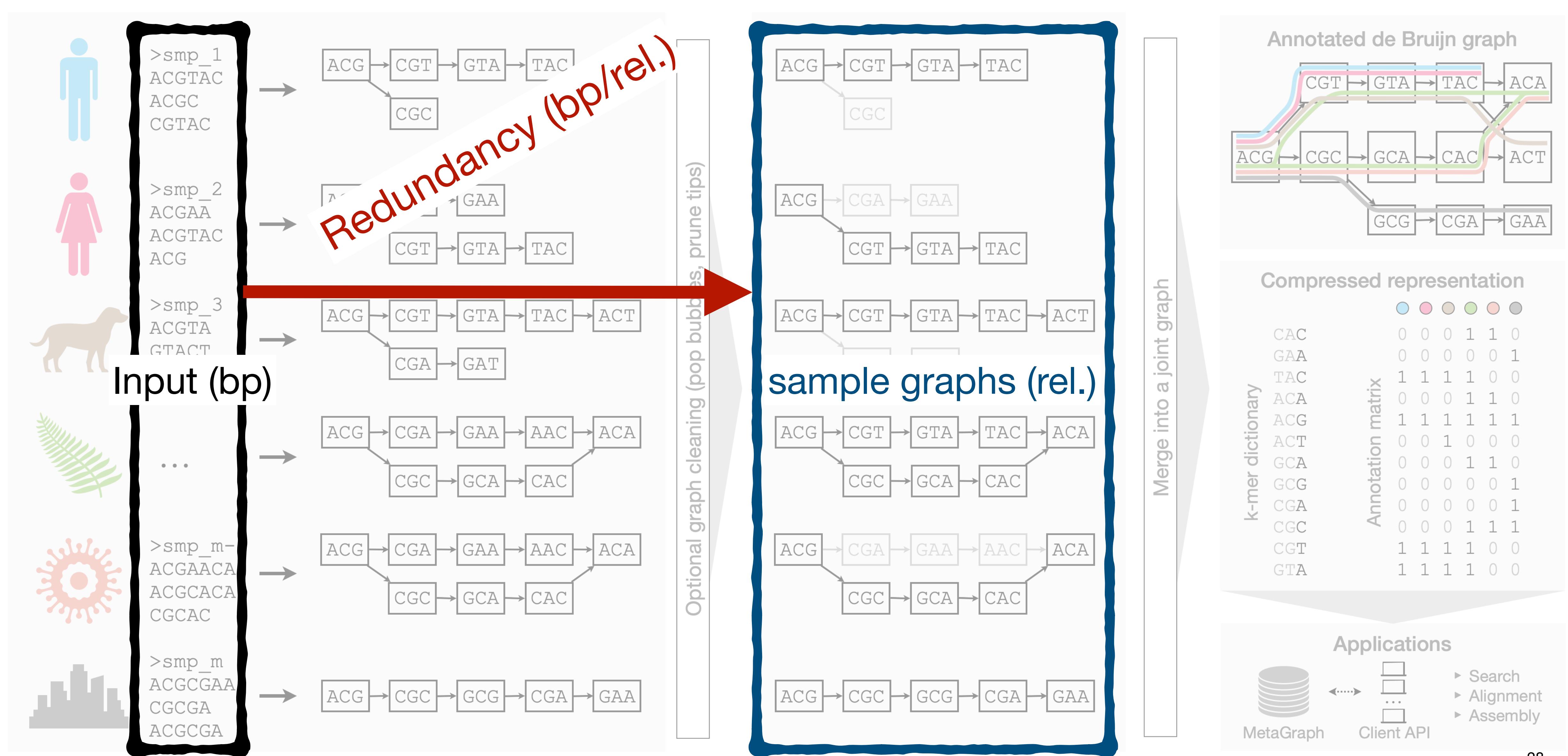
Indexing at Petabase-scale



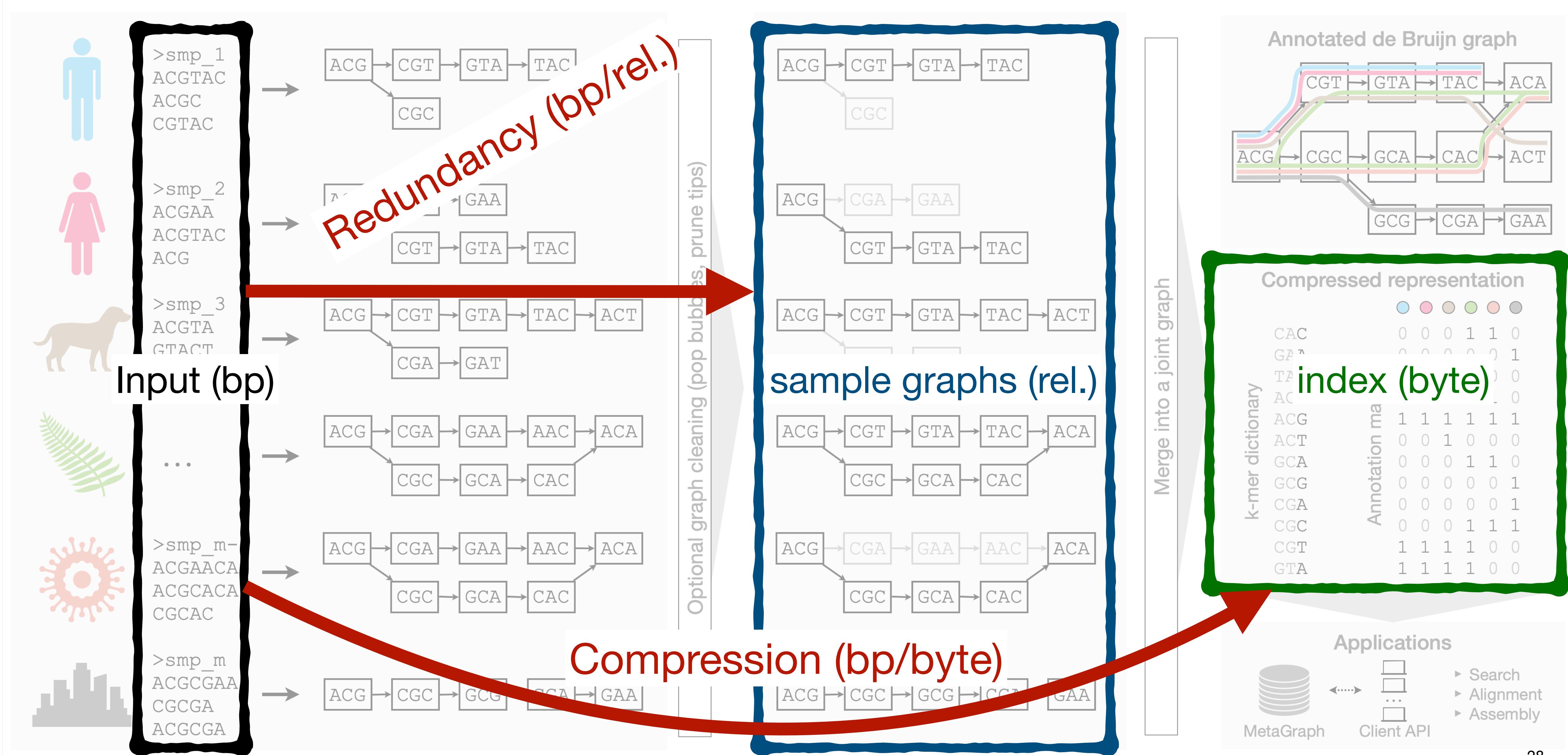
Indexing at Petabase-scale



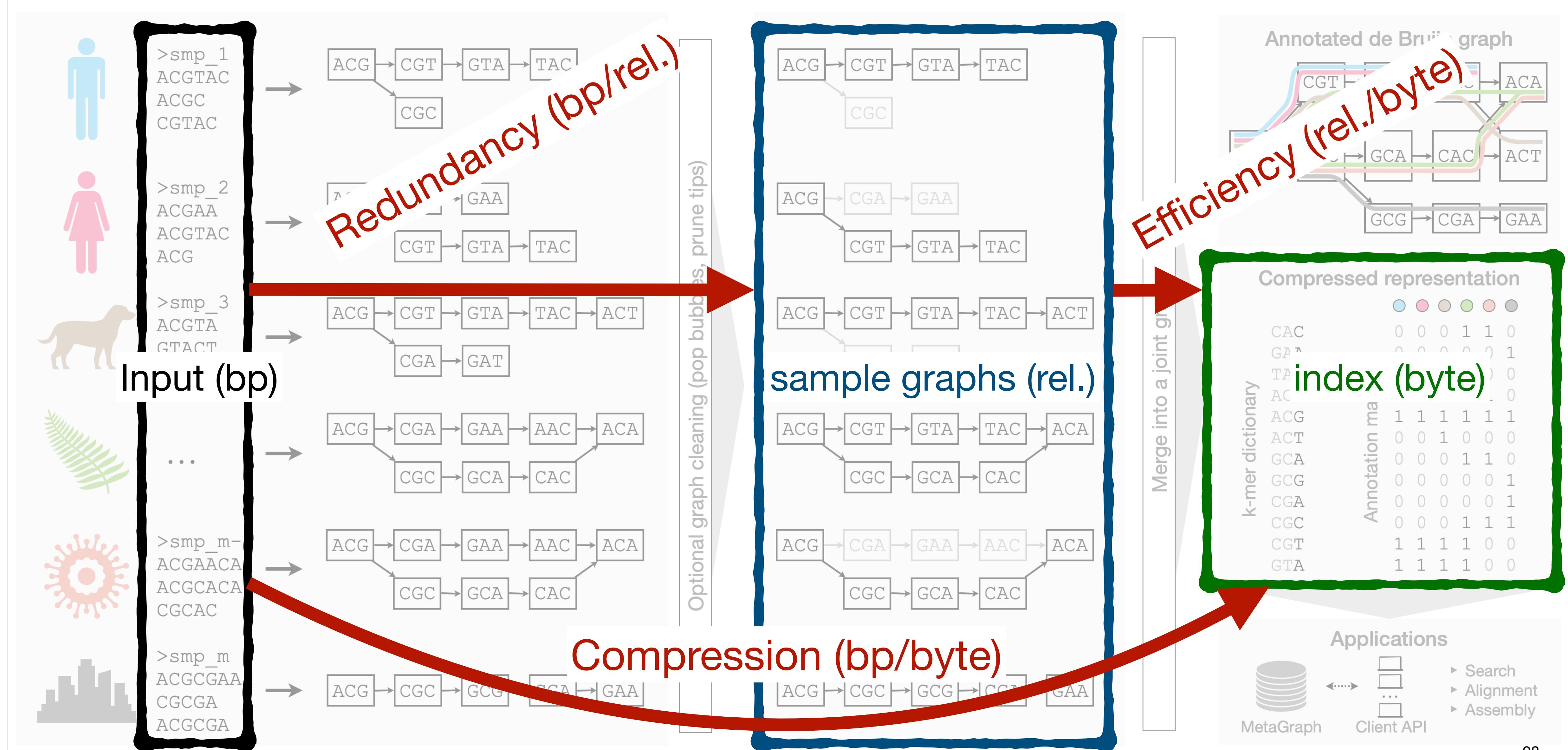
Compression, redundancy, efficiency



Compression, redundancy, efficiency



Compression, redundancy, efficiency



Indexing petabase-scale inputs

Type	Data set	Tbp	# labels	Index size	Compression (bp/byte)	Redund. (bp/rel.)	Efficiency (rel./byte)
Assembled seq.	UHGG (catalog)	0.01	4,644	3.2 GB	3.5	1.0	3.5
	UHGG (all)	0.71	286,997	27.3 GB	26.0	1.0	25.9
	Tara Oceans (scaffolds)	0.36	318,205,057	110.2 GB	3.2	1.0	3.1
	Tara Oceans with coord. [†]	0.06	34,815	14.6 GB	4.2	1.0	4.2
	RefSeq with coordinates [†]	1.70	85,375	508.9 GB	3.3	1.0	3.3

Indexing petabase-scale inputs

Type	Data set	Tbp	# labels	Index size	Compression (bp/byte)	Redund. (bp/rel.)	Efficiency (rel./byte)
Assembled seq.	UHGG (catalog)	0.01	4,644	3.2 GB	3.5	1.0	3.5
	UHGG (all)	0.71	286,997	27.3 GB	26.0	1.0	25.9
	Tara Oceans (scaffolds)	0.36	318,205,057	110.2 GB	3.2	1.0	3.1
	Tara Oceans with coord. [†]	0.06	34,815	14.6 GB	4.2	1.0	4.2
	RefSeq with coordinates [†]	1.70	85,375	508.9 GB	3.3	1.0	3.3
RNA-Seq	GTEX	71.2	9,759	9.6 GB	7,416	241.2	30.7
	GTEX with counts [‡]	71.2	9,759	76.3 GB	934	241.2	3.9
	TCGA	81.2	11,095	11.1 GB	7,288	334.4	21.8
	TCGA with counts [‡]	81.2	11,095	81.2 GB	1,001	320.4	3.1

Indexing petabase-scale inputs

Type	Data set	Tbp	# labels	Index size	Compression (bp/byte)	Redund. (bp/rel.)	Efficiency (rel./byte)
Assembled seq.	UHGG (catalog)	0.01	4,644	3.2 GB	3.5	1.0	3.5
	UHGG (all)	0.71	286,997	27.3 GB	26.0	1.0	25.9
	Tara Oceans (scaffolds)	0.36	318,205,057	110.2 GB	3.2	1.0	3.1
	Tara Oceans with coord. [†]	0.06	34,815	14.6 GB	4.2	1.0	4.2
	RefSeq with coordinates [†]	1.70	85,375	508.9 GB	3.3	1.0	3.3
RNA-Seq	GTEX	71.2	9,759	9.6 GB	7,416	241.2	30.7
	GTEX with counts [‡]	71.2	9,759	76.3 GB	934	241.2	3.9
	TCGA	81.2	11,095	11.1 GB	7,288	334.4	21.8
	TCGA with counts [‡]	81.2	11,095	81.2 GB	1,001	320.4	3.1
MGS	MetaSUB	7.2	4,220	46.7 GB	155	40.5	3.8
	SRA-MetaGut	155.8	241,384	1,111.3 GB	140	22.2	6.3

Indexing petabase-scale inputs

Type	Data set	Tbp	# labels	Index size	Compression (bp/byte)	Redund. (bp/rel.)	Efficiency (rel./byte)
Assembled seq.	UHGG (catalog)	0.01	4,644	3.2 GB	3.5	1.0	3.5
	UHGG (all)	0.71	286,997	27.3 GB	26.0	1.0	25.9
	Tara Oceans (scaffolds)	0.36	318,205,057	110.2 GB	3.2	1.0	3.1
	Tara Oceans with coord. [†]	0.06	34,815	14.6 GB	4.2	1.0	4.2
	RefSeq with coordinates [†]	1.70	85,375	508.9 GB	3.3	1.0	3.3
RNA-Seq	GTEX	71.2	9,759	9.6 GB	7,416	241.2	30.7
	GTEX with counts [‡]	71.2	9,759	76.3 GB	934	241.2	3.9
	TCGA	81.2	11,095	11.1 GB	7,288	334.4	21.8
	TCGA with counts [‡]	81.2	11,095	81.2 GB	1,001	320.4	3.1
MGS	MetaSUB	7.2	4,220	46.7 GB	155	40.5	3.8
	SRA-MetaGut	155.8	241,384	1,111.3 GB	140	22.2	6.3
SRA subsets	SRA-Microbe	221.1	446,506	65.5 GB	3,376	157.6	21.4
	SRA-Fungi	162.1	121,900	98.3 GB	1,649	113.9	14.5
	SRA-Plants	1,109.2	531,714	1,844.1 GB	602	61.5	9.8
	SRA-Human	725.4	436,494	3,402.1 GB	213	15.7	13.5
	SRA-Metazoa (Mouse)	146.6	57,938	291.6 GB	503	26.6	18.9
	SRA-Metazoa (1k studies)	118.8	67,391	302.7 GB	393	33.5	11.7
	SRA-Metazoa	1,999.5	805,239	5.1 TB*	390*	33.3	11.7*

Indexing petabase-scale inputs

Type	Data set	Tbp	# labels	Index size	Compression (bp/byte)	Redund. (bp/rel.)	Efficiency (rel./byte)
Assembled seq.	UHGG (catalog)	0.01	4,644	3.2 GB	3.5	1.0	3.5
	UHGG (all)	0.71	286,997	27.3 GB	26.0	1.0	25.9
	Tara Oceans (scaffolds)	0.36	318,205,057	110.2 GB	3.2	1.0	3.1
	Tara Oceans with coord. [†]	0.06	34,815	14.6 GB	4.2	1.0	4.2
	RefSeq with coordinates [†]	1.70	85,375	508.9 GB	3.3	1.0	3.3
RNA-Seq	GTEX	71.2	9,759	9.6 GB	7,416	241.2	30.7
	GTEX with counts [‡]	71.2	9,759	76.3 GB	934	241.2	3.9
	TCGA	81.2	11,095	11.1 GB	7,288	334.4	21.8
	TCGA with counts [‡]	81.2	11,095	81.2 GB	1,001	320.4	3.1
MGS	MetaSUB	7.2	4,220	46.7 GB	155	40.5	3.8
	SRA-MetaGut	155.8	241,384	1,111.3 GB	140	22.2	6.3
SRA subsets	SRA-Microbe	221.1	446,506	65.5 GB	3,376	157.6	21.4
	SRA-Fungi	162.1	121,900	98.3 GB	1,649	113.9	14.5
	SRA-Plants	1,109.2	531,714	1,844.1 GB	602	61.5	9.8
	SRA-Human	725.4	436,494	3,402.1 GB	213	15.7	13.5
	SRA-Metazoa (Mouse)	146.6	57,938	291.6 GB	503	26.6	18.9
	SRA-Metazoa (1k studies)	118.8	67,391	302.7 GB	393	33.5	11.7
	SRA-Metazoa	1,999.5	805,239	5.1 TB*	390*	33.3	11.7*
SRA	SRA-Public to 01.01.2023	38,949.9	23,010,648				

Indexing petabase-scale inputs

Type	Data set	Tbp	# labels	Index size	Compression (bp/byte)	Redund. (bp/rel.)	Efficiency (rel./byte)
Assembled seq.	UHGG (catalog)	0.01	4,644	3.2 GB	3.5	1.0	3.5
	UHGG (all)	0.71	286,997	27.3 GB	26.0	1.0	25.9
	Tara Oceans (scaffolds)	0.36	318,205,057	110.2 GB	3.2	1.0	3.1
	Tara Oceans with coord. [†]	0.06	34,815	14.6 GB	4.2	1.0	4.2
	RefSeq with coordinates [†]	1.70	85,375	508.9 GB	3.3	1.0	3.3
RNA-Seq	GTEX	71.2	9,759	9.6 GB	7,416	241.2	30.7
	GTEX with counts [‡]	71.2	9,759	76.3 GB	934	241.2	3.9
	TCGA	81.2	11,095	11.1 GB	7,288	334.4	21.8
	TCGA with counts [‡]	81.2	11,095	81.2 GB	1,001	320.4	3.1
MGS	MetaSUB	7.2	4,220	46.7 GB	155	40.5	3.8
	SRA-MetaGut	155.8	241,384	1,111.3 GB	140	22.2	6.3
SRA subsets	SRA-Microbe	221.1	446,506	65.5 GB	3,376	157.6	21.4
	SRA-Fungi	162.1	121,900	98.3 GB	1,649	113.9	14.5
	SRA-Plants	1,109.2	531,714	1,844.1 GB	602	61.5	9.8
	SRA-Human	725.4	436,494	3,402.1 GB	213	15.7	13.5
	SRA-Metazoa (Mouse)	146.6	57,938	291.6 GB	503	26.6	18.9
	SRA-Metazoa (1k studies)	118.8	67,391	302.7 GB	393	33.5	11.7
	SRA-Metazoa	1,999.5	805,239	5.1 TB*	390*	33.3	11.7*
SRA	SRA-Public (100 studies)	9.6	5,184	32.0 GB	300	73.3	4.1
	SRA-Public to 01.01.2023	38,949.9	23,010,648				

Indexing petabase-scale inputs

Type	Data set	Tbp	# labels	Index size	Compression (bp/byte)	Redund. (bp/rel.)	Efficiency (rel./byte)
Assembled seq.	UHGG (catalog)	0.01	4,644	3.2 GB	3.5	1.0	3.5
	UHGG (all)	0.71	286,997	27.3 GB	26.0	1.0	25.9
	Tara Oceans (scaffolds)	0.36	318,205,057	110.2 GB	3.2	1.0	3.1
	Tara Oceans with coord. [†]	0.06	34,815	14.6 GB	4.2	1.0	4.2
	RefSeq with coordinates [†]	1.70	85,375	508.9 GB	3.3	1.0	3.3
RNA-Seq	GTEX	71.2	9,759	9.6 GB	7,416	241.2	30.7
	GTEX with counts [‡]	71.2	9,759	76.3 GB	934	241.2	3.9
	TCGA	81.2	11,095	11.1 GB	7,288	334.4	21.8
	TCGA with counts [‡]	81.2	11,095	81.2 GB	1,001	320.4	3.1
MGS	MetaSUB	7.2	4,220	46.7 GB	155	40.5	3.8
	SRA-MetaGut	155.8	241,384	1,111.3 GB	140	22.2	6.3
SRA subsets	SRA-Microbe	221.1	446,506	65.5 GB	3,376	157.6	21.4
	SRA-Fungi	162.1	121,900	98.3 GB	1,649	113.9	14.5
	SRA-Plants	1,109.2	531,714	1,844.1 GB	602	61.5	9.8
	SRA-Human	725.4	436,494	3,402.1 GB	213	15.7	13.5
	SRA-Metazoa (Mouse)	146.6	57,938	291.6 GB	503	26.6	18.9
	SRA-Metazoa (1k studies)	118.8	67,391	302.7 GB	393	33.5	11.7
	SRA-Metazoa	1,999.5	805,239	5.1 TB*	390*	33.3	11.7*
SRA	SRA-Public (100 studies)	9.6	5,184	32.0 GB	300	73.3	4.1
	SRA-Public to 01.01.2023	38,949.9	23,010,648	130 TB*	300*	73.3*	4.1*

Search in the Sequence Read Archive

- ▶ Search with Serratus [Edgar *et al.*, 2022]
 - (viral pangenome search of 82 Mbp against 4M samples at 0.62 cents per sample)

Search in the Sequence Read Archive

- ▶ Search with Serratus [Edgar *et al.*, 2022]
 - (viral pangenome search of 82 Mbp against 4M samples at 0.62 cents per sample)
 - effective query cost for SRA-Public (23M samples): **\$1,700** per **1 Mbp** of query

Search in the Sequence Read Archive

- ▶ Search with Serratus [Edgar *et al.*, 2022]
 - (viral pangenome search of 82 Mbp against 4M samples at 0.62 cents per sample)
 - effective query cost for SRA-Public (23M samples): **\$1,700 per 1 Mbp** of query
- ▶ Search with MetaGraph
 - (extrapolated from ‘SRA-Public (100 studies)’ with prices for Google Cloud)

Search in the Sequence Read Archive

- ▶ Search with Serratus [Edgar *et al.*, 2022]
 - (viral pangenome search of 82 Mbp against 4M samples at 0.62 cents per sample)
 - effective query cost for SRA-Public (23M samples): **\$1,700 per 1 Mbp** of query
- ▶ Search with MetaGraph
 - (extrapolated from ‘SRA-Public (100 studies)’ with prices for Google Cloud)
 - effective query cost for SRA-Public (23M samples): **\$2.8 per 1 Mbp** of query
 - **600x cheaper** than Serratus
 - indexing cost: only 2.2 cents per sample

MetaGraph Online

Home Search Align Graphs BioRxiv

MetaGraph: Search DNA Sequences

```
TTTCACTCTTGATAGCAGCATGCTTAGTACTAAGCTAAGTCTCCAAGATTGTCGAGTCAGTCGCTTCATTTCTTACCTGATACTAGTATGACTTGATCCCTCCCC  
CTGCACGTAAACACCACAAAAGATACACTTAATTACCACTAGAAATATAACATCAATGCAGTCATAGAACATCGGAGGAACATTTGCCAAGCAGGGTTT
```

Select graph: SRA-Fungi

Minimum k-mer matches: 100%

Search with alignment ?

Search SRA-Fungi

Search results

Show 10 ✓ entries

Download as csv

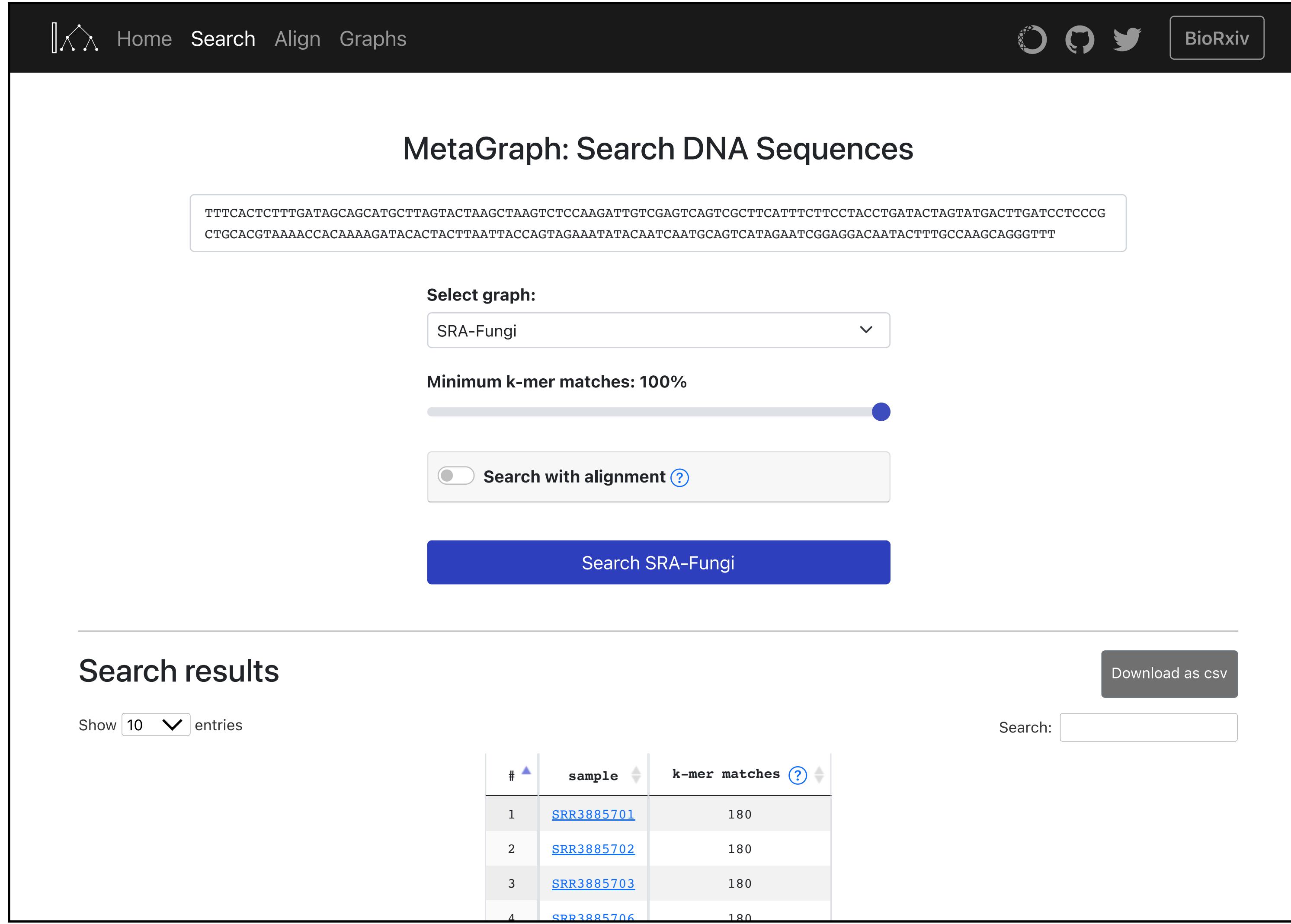
Search:

#	sample	k-mer matches
1	SRR3885701	180
2	SRR3885702	180
3	SRR3885703	180
4	SRR3885706	180



metagraph.ethz.ch/search

MetaGraph Online



The screenshot shows the MetaGraph Online search interface. At the top, there's a navigation bar with icons for Home, Search, Align, and Graphs, along with social media sharing buttons and a BioRxiv link. Below the navigation is a title "MetaGraph: Search DNA Sequences". A text input field contains a DNA sequence: TTTCACTCTTGATAGCAGCATGCTTAGTACTAAGCTAAGTCTCCAAGATTGTCGAGTCAGTCGCTTCATTCTTCTACCTGATACTAGTATGACTTGATCCCTCCCCGCTGCACGTAAACCAACAAAAGATACACTTAATTACCACTAGAAATATAATCAATGCAGTCATAGAACATCGGAGGAATACTTGCCAAGCAGGGTTT. A dropdown menu labeled "Select graph:" shows "SRA-Fungi" selected. A slider for "Minimum k-mer matches" is set to 100%. A toggle switch for "Search with alignment" is turned off. A large blue button labeled "Search SRA-Fungi" is centered. Below this is a section titled "Search results" with a table showing four entries. The table has columns for "#", "sample", "k-mer matches", and a question mark icon. The samples listed are SRR3885701, SRR3885702, SRR3885703, and SRR3885706, each with 180 k-mer matches. There are buttons for "Download as csv" and "Search:" with a search input field.



metagraph.ethz.ch/search

Python Client API

```
In [1]: from metagraph.client import GraphClient  
  
SRV = "metagraph.ethz.ch"  
PORT = 12345  
g1 = GraphClient(SRV, PORT, api_path="/metasub")  
g2 = GraphClient(SRV, PORT, api_path="/refseq")
```



```
In [2]: query = "GGCTAACTACGTGCCAGCAGCCGCGGTAATAC"  
g1.search(query, align=True)
```

```
Out [2]:
```

#	sample	sequence	score
0	SRR2201245	GGCTAACTACGTGCCAGCAGCCGCGGTAATAC	64
1	ERR1732568	GGCTAACTACGTGCCAGCAGCCGCGGTAATAC	64
2	ERR847096	GGCTAACTACGTGCCAGCAGCCGCGGTAATAC	64
...

Conclusion

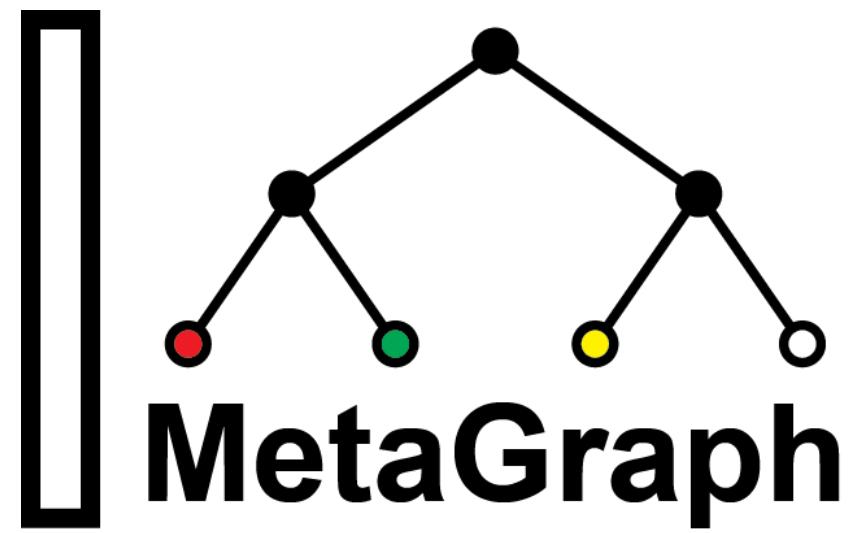
Main results of the thesis

- ▶ Showed the feasibility of indexing **all existing sequence archives**

Main results of the thesis

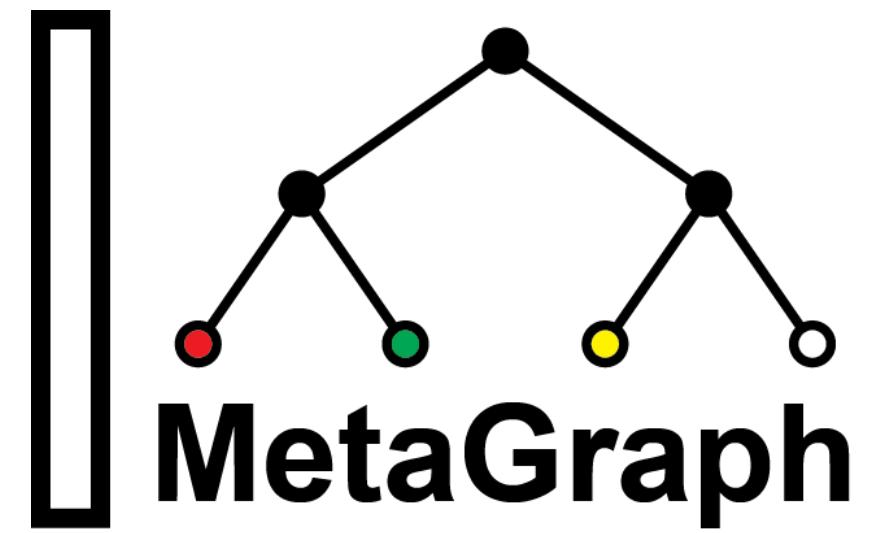
- ▶ Showed the feasibility of indexing **all existing sequence archives**
- ▶ Developed efficient methods for scalable **indexing** and **retrieval**
 - k-mer presence/absence
 - k-mer abundances
 - k-mer positions

Main results of the thesis



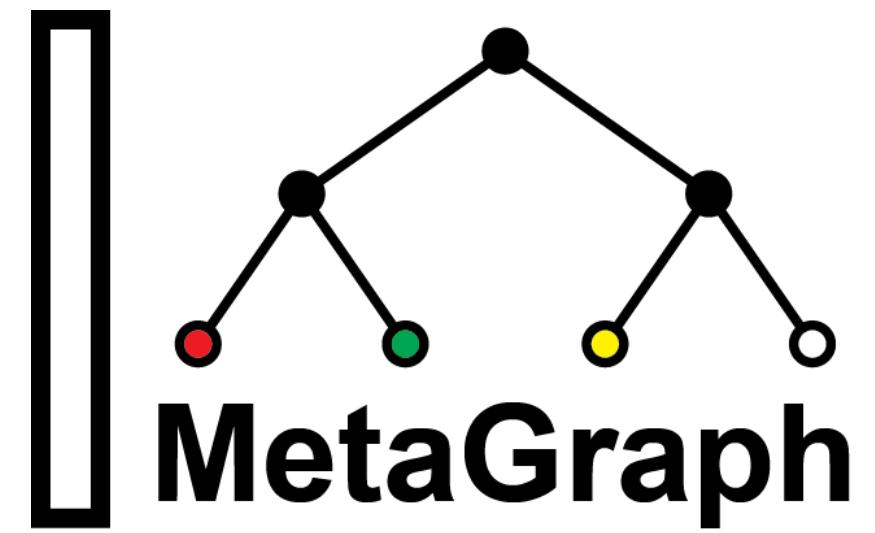
- ▶ Showed the feasibility of indexing **all existing sequence archives**
- ▶ Developed efficient methods for scalable **indexing** and **retrieval**
 - k-mer presence/absence
 - k-mer abundances
 - k-mer positions
- ▶ Efficient and scalable tool **MetaGraph**

Main results of the thesis



- ▶ Showed the feasibility of indexing **all existing sequence archives**
- ▶ Developed efficient methods for scalable **indexing** and **retrieval**
 - k-mer presence/absence
 - k-mer abundances
 - k-mer positions
- ▶ Efficient and scalable tool **MetaGraph**
- ▶ **Community resource indexes** of very large sequence data sets

Main results of the thesis



- ▶ Showed the feasibility of indexing **all existing sequence archives**
- ▶ Developed efficient methods for scalable **indexing** and **retrieval**
 - k-mer presence/absence
 - k-mer abundances
 - k-mer positions
- ▶ Efficient and scalable tool **MetaGraph**
- ▶ **Community resource indexes** of very large sequence data sets
- ▶ **MetaGraph Online** <https://metagraph.ethz.ch>

Publications

Primary works

- Mikhail Karasikov, Harun Mustafa, Daniel Danciu, Marc Zimmermann, Christopher Barber, Gunnar Rätsch, and André Kahles. "MetaGraph: Indexing and Analysing Nucleotide Archives at Petabase-scale". In: *bioRxiv* (2020). doi: [10.1101/2020.10.01.322164](https://doi.org/10.1101/2020.10.01.322164)
- Mikhail Karasikov, Harun Mustafa, Gunnar Rätsch, and André Kahles. "Lossless indexing with counting de Bruijn graphs". In: *Genome Research* 32.9 (2022), pp. 1754–1764. doi: [10.1101/gr.276607.122](https://doi.org/10.1101/gr.276607.122)
- Mikhail Karasikov, Harun Mustafa, Amir Joudaki, Sara Javadzadeh-No, Gunnar Rätsch, and André Kahles. "Sparse binary relation representations for genome graph annotation". In: *Journal of Computational Biology* 27.4 (2020), pp. 626–639. doi: [10.1089/cmb.2019.0324](https://doi.org/10.1089/cmb.2019.0324)
- Daniel Danciu, Mikhail Karasikov, Harun Mustafa, André Kahles, and Gunnar Rätsch. "Topology-based sparsification of graph annotations". In: *Bioinformatics* 37.Supplement_1 (July 2021), pp. i169–i176. doi: [10.1093/bioinformatics/btab330](https://doi.org/10.1093/bioinformatics/btab330)

Significant contributions

- David Danko et al. "A global metagenomic map of urban microbiomes and antimicrobial resistance". In: *Cell* 184.13 (2021), 3376–3393.e17. doi: [10.1016/j.cell.2021.05.002](https://doi.org/10.1016/j.cell.2021.05.002)
- Lucas Paoli, Hans-Joachim Ruscheweyh, Clarissa C. Forneris, Florian Hubrich, Satria Kautsar, Agneya Bhushan, Alessandro Lotti, Quentin Clayssen, Guillem Salazar, Alessio Milanese, Charlotte I. Carlström, Chrysa Papadopoulou, Daniel Gehrig, Mikhail Karasikov, Harun Mustafa, Martin Larralde, Laura M. Carroll, Pablo Sánchez, Ahmed A. Zayed, Dylan R. Cronin, Silvia G. Acinas, Peer Bork, Chris Bowler, Tom O. Delmont, Josep M. Gasol, Alvar D. Gossert, André Kahles, Matthew B. Sullivan, Patrick Wincker, Georg Zeller, Serina L. Robinson, Jörn Piel, and Shinichi Sunagawa. "Biosynthetic potential of the global ocean microbiome". In: *Nature* 607.7917 (2022), pp. 111–118. doi: [10.1038/s41586-022-04862-3](https://doi.org/10.1038/s41586-022-04862-3)
- Harun Mustafa, Mikhail Karasikov, Gunnar Rätsch, and André Kahles. "MetaGraph-MLA: Label-guided alignment to variable-order De Bruijn graphs". In: *bioRxiv* (2022). doi: [10.1101/2022.11.04.514718](https://doi.org/10.1101/2022.11.04.514718)
- Harun Mustafa, Ingo Schilken, Mikhail Karasikov, Carsten Eickhoff, Gunnar Rätsch, André Kahles, and John Hancock. "Dynamic compression schemes for graph coloring". In: *Bioinformatics* (2018). doi: [10.1093/bioinformatics/bty632](https://doi.org/10.1093/bioinformatics/bty632)
- Maciej Besta, Raghavendra Kanakagiri, Harun Mustafa, Mikhail Karasikov, Gunnar Rätsch, Torsten Hoefer, and Edgar Solomonik. "Communication-Efficient Jaccard similarity for High-Performance Distributed Genome Comparisons". In: *2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. Los Alamitos, CA, USA: IEEE Computer Society, 2020, pp. 1122–1132. doi: [10.1109/IPDPS47924.2020.00118](https://doi.org/10.1109/IPDPS47924.2020.00118)

Outreach

Conferences

Talks:

- RECOMB 2019 (Washington, DC, USA) — presented by HM
- ISMB/ECCB 2021, HiTSeq COSI (online)
- RECOMB 2022 (La Jolla, California, USA)
- IGGSy 2022 (Ascona, Switzerland)
- JOBIM 2022 (Rennes, France) — **invited talk**

Poster:

- Biological Data Science - CSHL Meeting 2022 (Cold Spring Harbor, New York, USA)

Other talks

- MLSS 2020 (online)
- Zurich Seminars in Bioinformatics 2022 (Zurich, Switzerland)

Collaborations

- Tara Oceans (Sunagawa lab)
- **MetaSUB consortia** (C. Mason)
- Differential assembly (M. Huber)

Possible future applications

- ▶ Detection/confirmation of
 - novel splice junctions
 - **trans-splice junctions**
 - genomic variants (incl. structural variants)
 - novel sequence elements
 - gene fusions
 - phages in metagenomes
 - novel functional elements in metagenomes
 - pathogen pervasiveness (e.g. SARS CoV-II)
- ▶ Differential assembly
- ▶ Joint variant calling
- ▶ Large association studies on sequence elements

Thanks to the team



Mikhail Karasikov



Marc Zimmermann



Andre Kahles



Harun Mustafa



Daniel Danciu



Gunnar Rätsch

NRP75 Team



Torsten Hoefler



Mario Stanke



Matthis Ebel



Giovanna Migliorelli

Further contributors

- Marek Kokot
- Thomas Zhou
- Chris Barber
- Radu Muntean
- Jan Studený
- Sara Javadzadeh-No
- Predrag Krnetic



Big Data
National Research Programme



ETH zürich