ETH zürich

Searching in nucleotide archives at petabase scale with MetaGraph

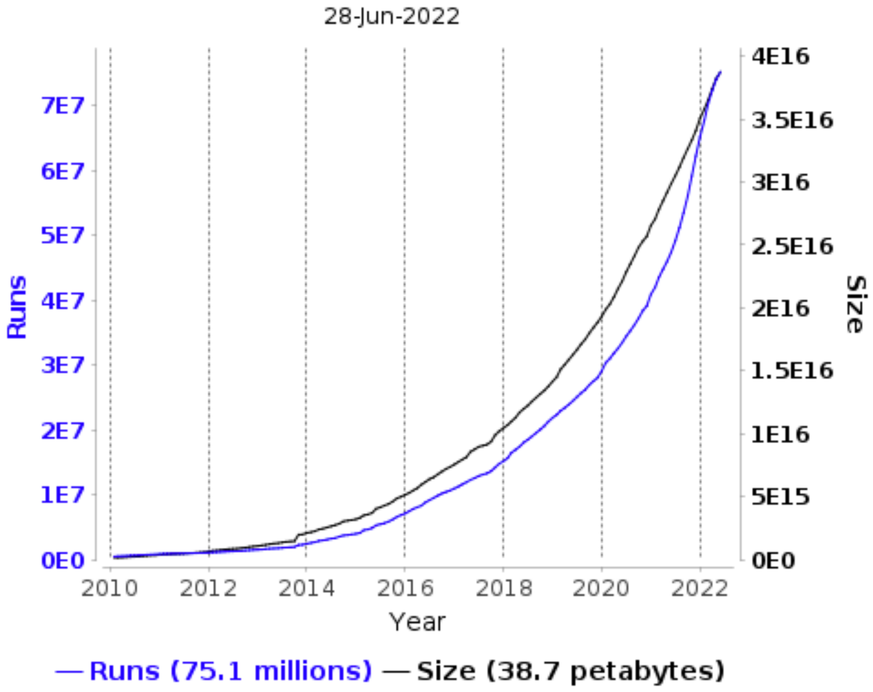
Mikhail Karasikov

Zurich Seminars in Bioinformatics 2022

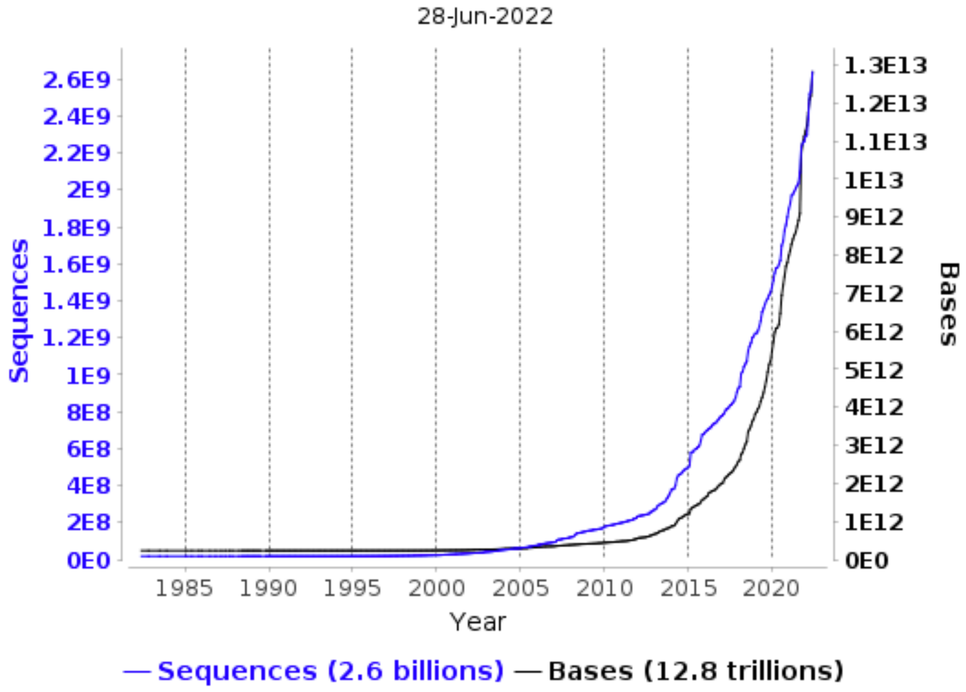
24 November 2022 (Zurich, Switzerland)

Growth of sequence archives

Sequence Read Archive (SRA)



Assembled sequences

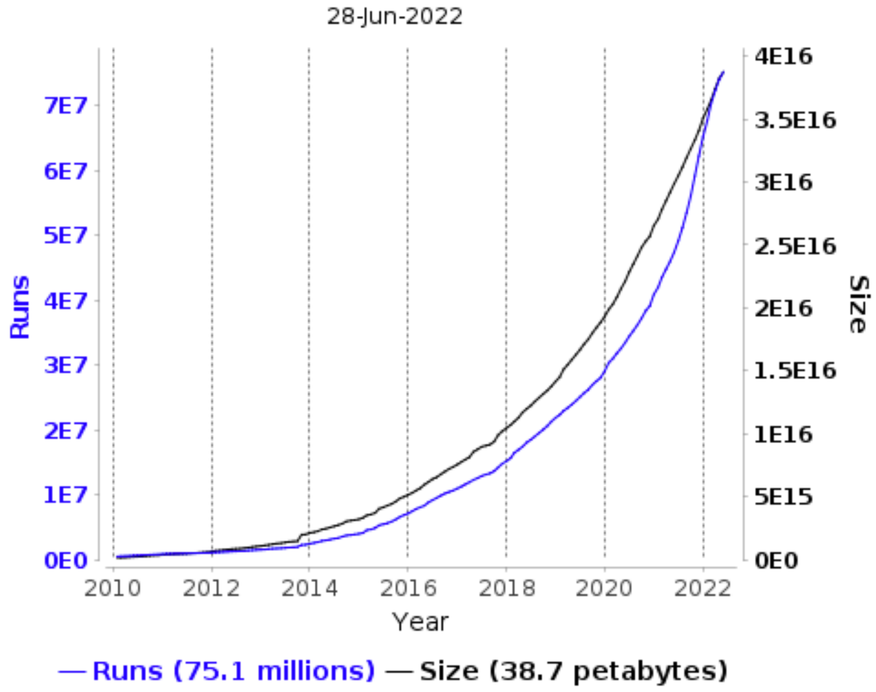


ENA
European Nucleotide Archive

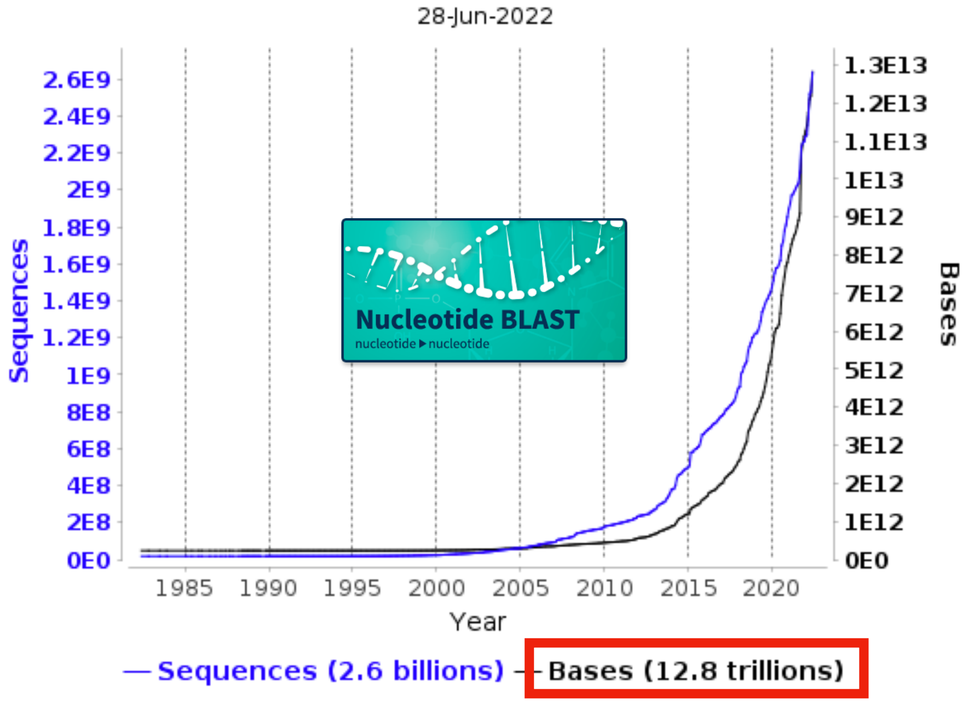
Data source: https://www.ebi.ac.uk/ena/browser/about/statistics

Growth of sequence archives

Sequence Read Archive (SRA)



Assembled sequences

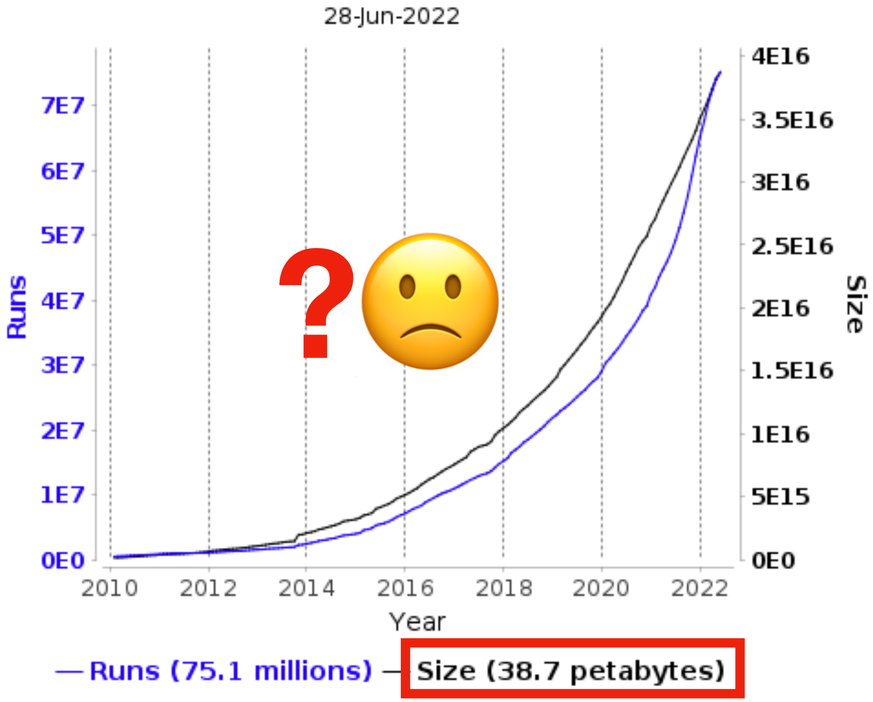


ENA
European Nucleotide Archive

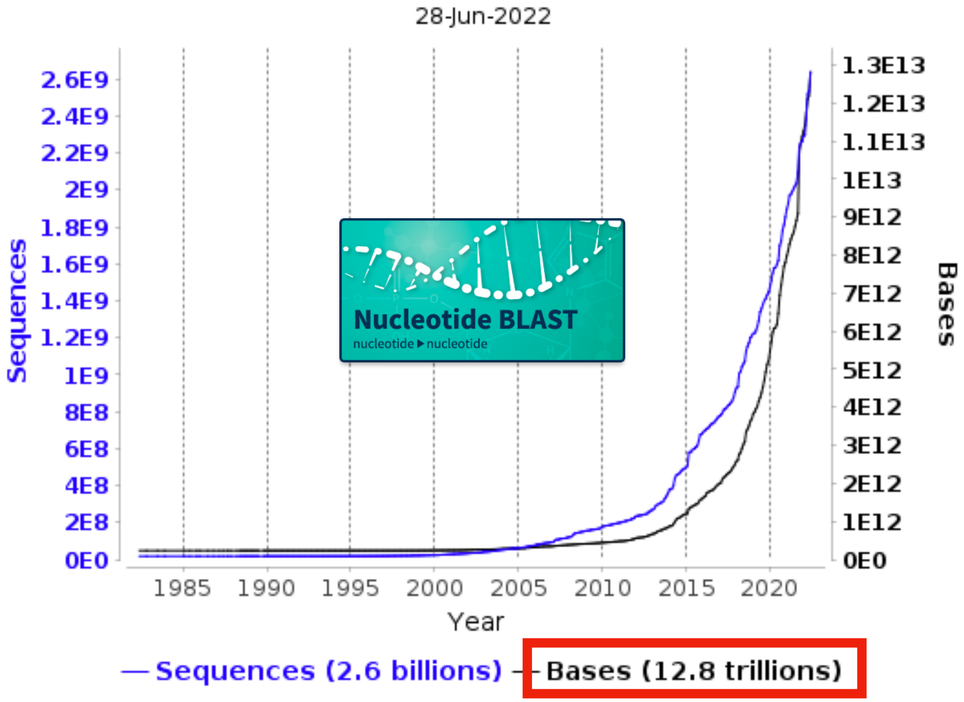
Data source: https://www.ebi.ac.uk/ena/browser/about/statistics

Growth of sequence archives

Sequence Read Archive (SRA)



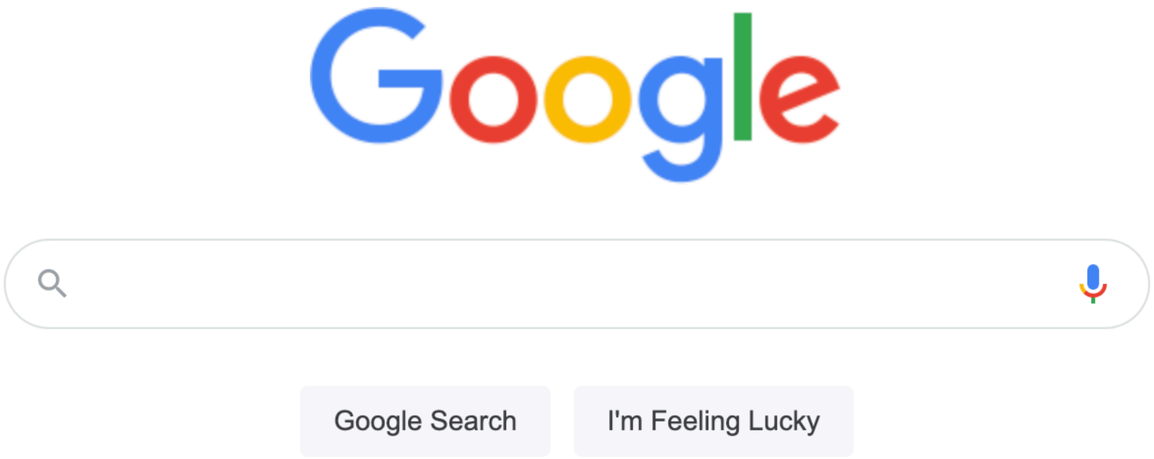
Assembled sequences



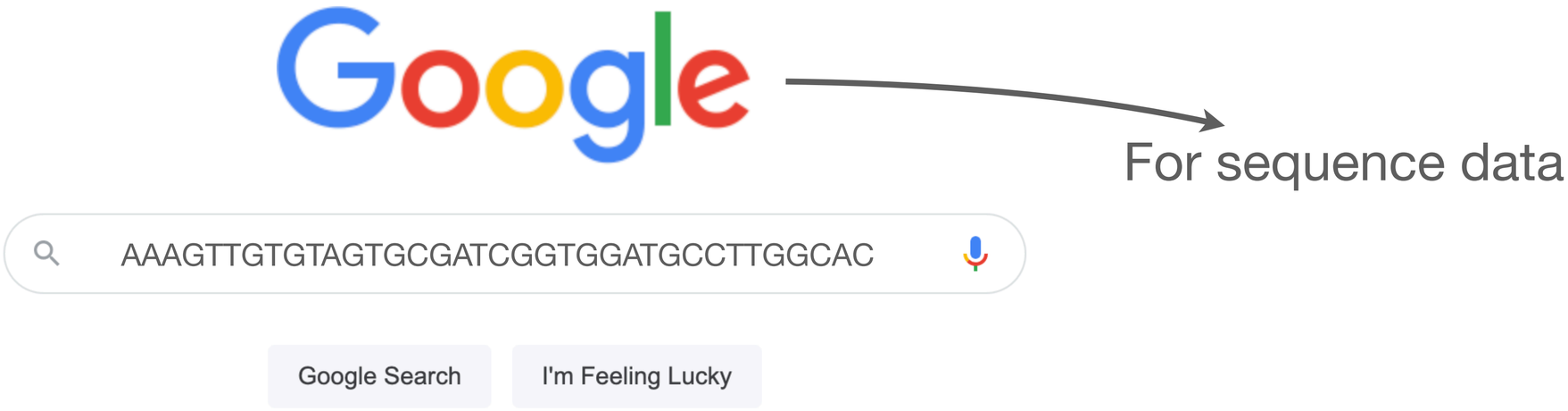
European Nucleotide Archive

Data source: https://www.ebi.ac.uk/ena/browser/about/statistics

What we want

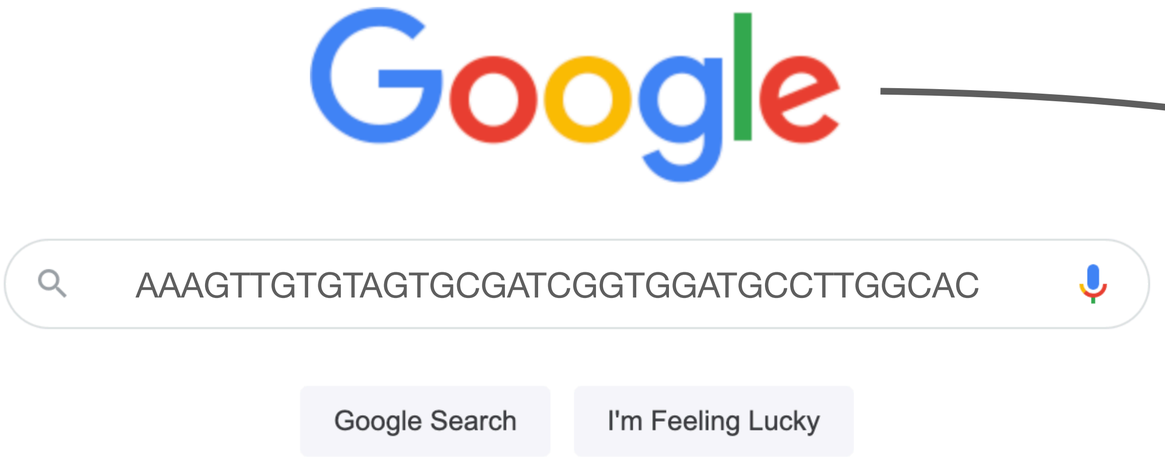


What we want



For sequence data

What we want



For sequence data

ENA European Nucleotide Archive

UK 10K

HMP

GENOME 10K

european genome-phenome archive

CATA AC SRA

MetaSUB

...

Biological sequence archives

Name	Institute(s)	Type	Size (Tc)	Date
Protein Data Bank (PDB) [34]	RCSB	Protein structure	0.000037	06.2022
UniProtKB [76]	EMBL-EBI, SIB	Amino acid	0.00018	06.2022
Non-redundant (nr) [288]	NCBI	Amino acid	0.186	06.2022
Nucleotide (nt) [288]	NCBI	Chromosome + Assembled	0.757	06.2022
RefSeq [243]	NCBI	Chromosome + Assembled Amino acid	2.97 0.089	05.2022
GenBank [70]	NCBI	Chromosome Assembled	1.17 15.9	02.2022
Sequence Read Archive [168]	DBJJ	DNA read	17,470	03.2022
European Nucleotide Archive (ENA) [129]	EMBL-EBI	DNA read	35,000	02.2022
Sequence read archive (SRA) [160]	NCBI	DNA read Open access DNA read	66,577 32,226	06.2022

Table by Harun Mustafa

Biological sequence archives

Name	Institute(s)	Type	Size (Tc)	Date
Protein Data Bank (PDB) [34]	RCSB	Protein structure	0.000037	06.2022
UniProtKB [76]	EMBL-EBI, SIB	Amino acid	0.00018	06.2022
Non-redundant (nr) [288]	NCBI	Amino acid	0.186	06.2022
Nucleotide (nt) [288]	NCBI	Chromosome + Assembled	0.757	06.2022
RefSeq [243]	NCBI	Chromosome + Assembled Amino acid	2.97 0.089	05.2022
GenBank [70]	NCBI	Chromosome Assembled	1.17 15.9	02.2022
Sequence Read Archive [168]	DBJJ	DNA read	17,470	03.2022
European Nucleotide Archive (ENA) [129]	EMBL-EBI	DNA read	35,000	02.2022
Sequence read archive (SRA) [160]	NCBI	DNA read Open access DNA read	66,577 32,226	06.2022

Table by Harun Mustafa

Indexing workflow

```
>smp_1
ACGTAC
ACGC
CGTAC
```

```
>smp_2
ACGAA
ACGTAC
ACG
```

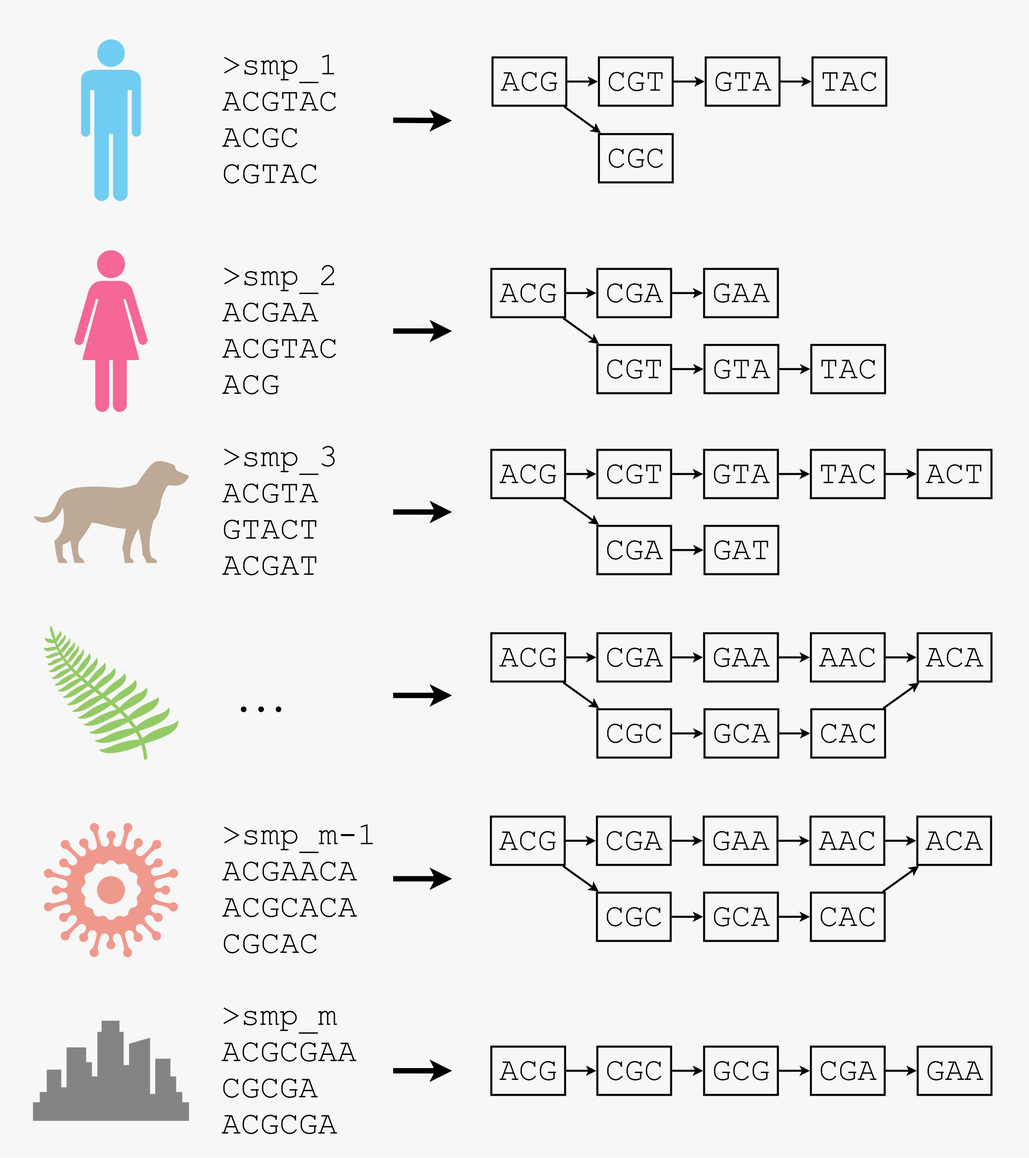
```
>smp_3
ACGTA
GTACT
ACGAT
```

...

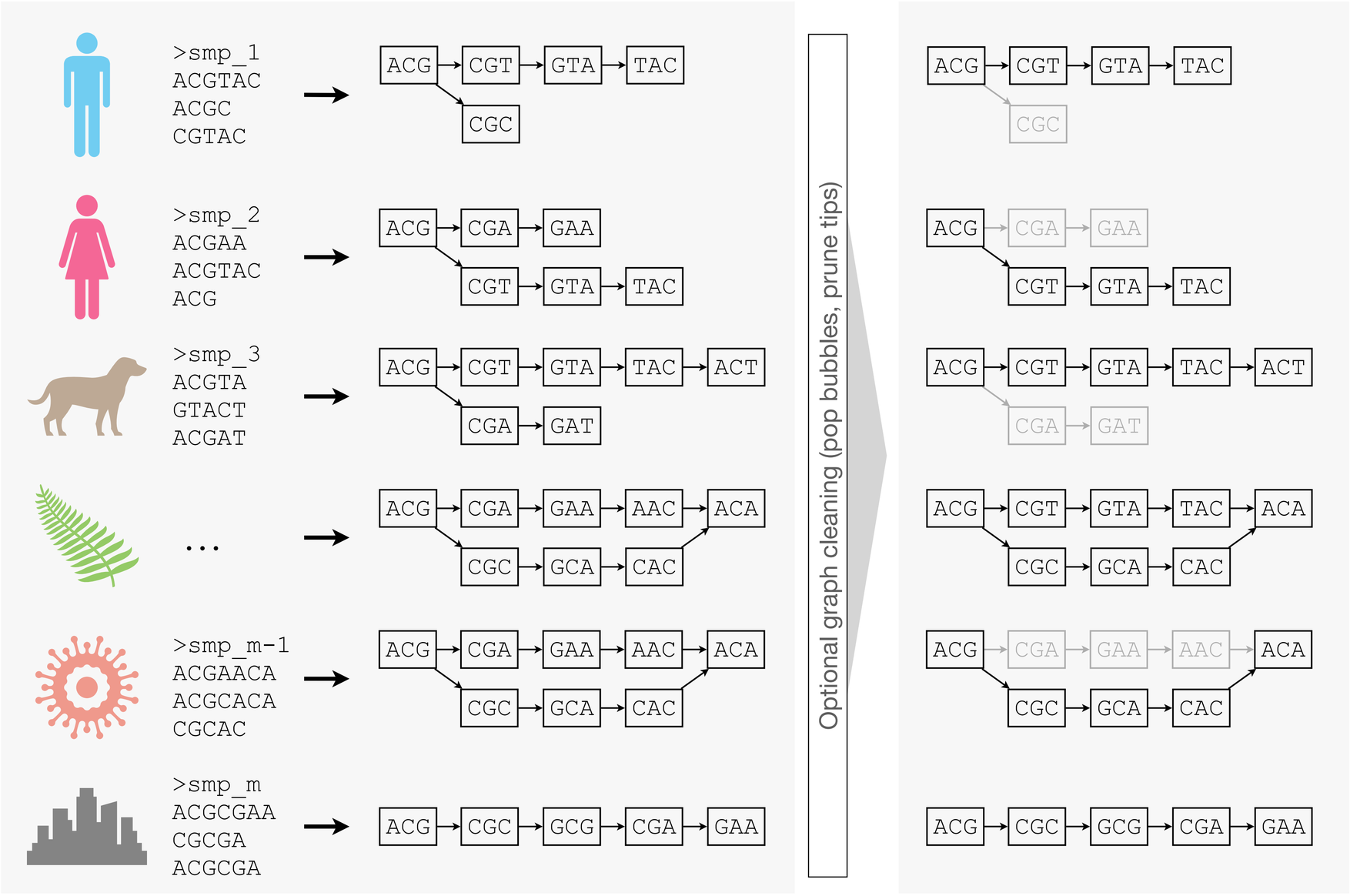
```
>smp_m-1
ACGAACA
ACGCACA
CGCAC
```

```
>smp_m
ACGCGAA
CGCGA
ACGCGA
```

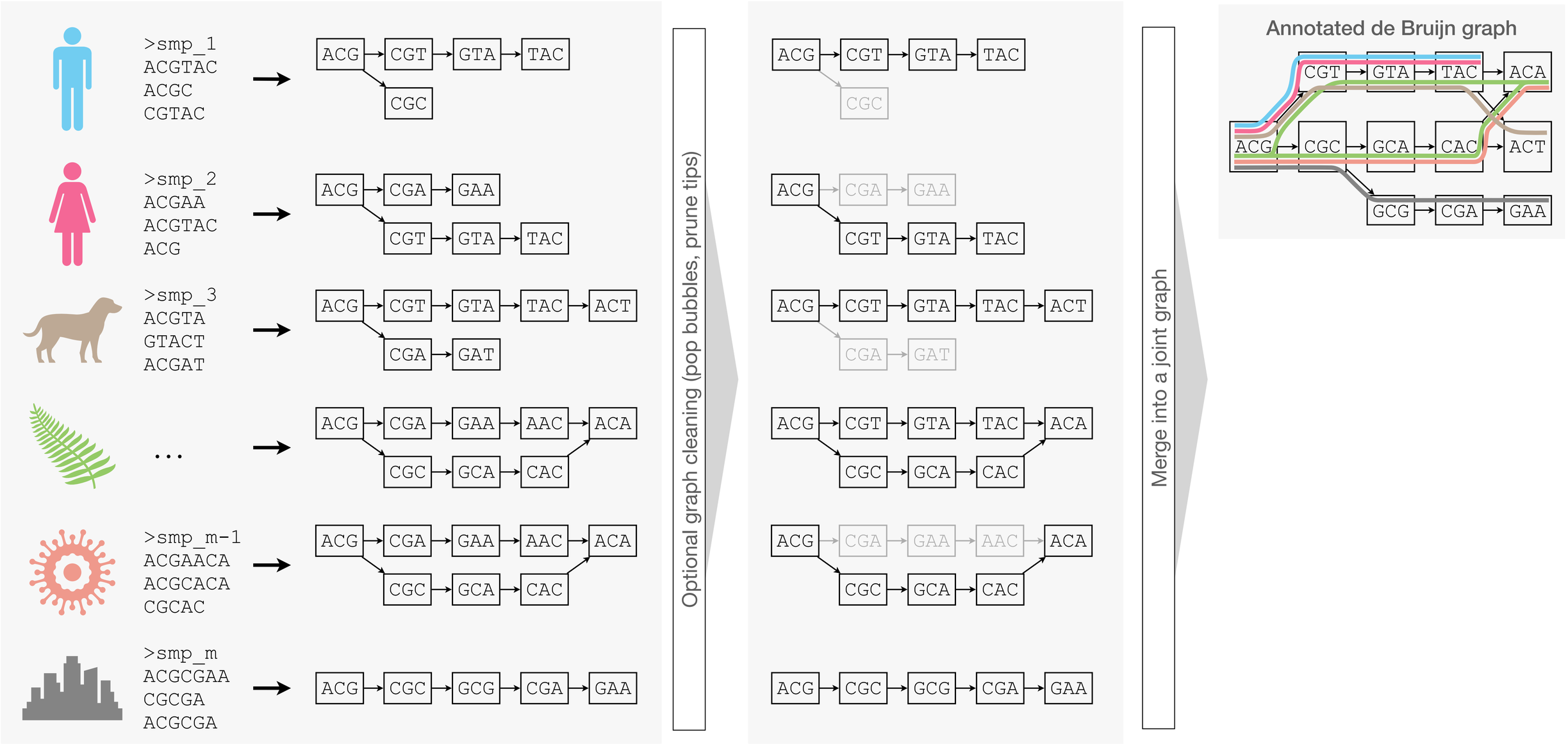
Indexing workflow



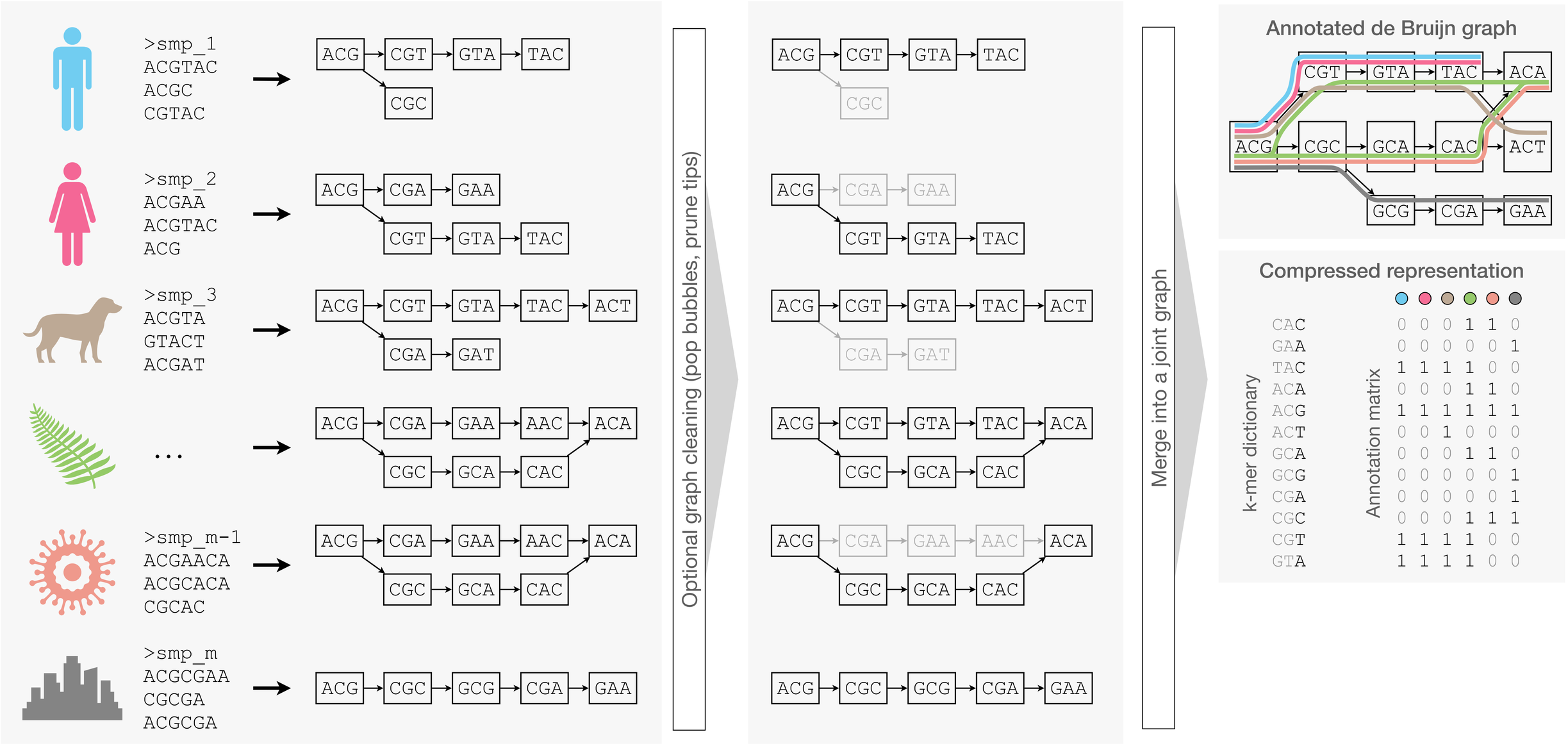
Indexing workflow



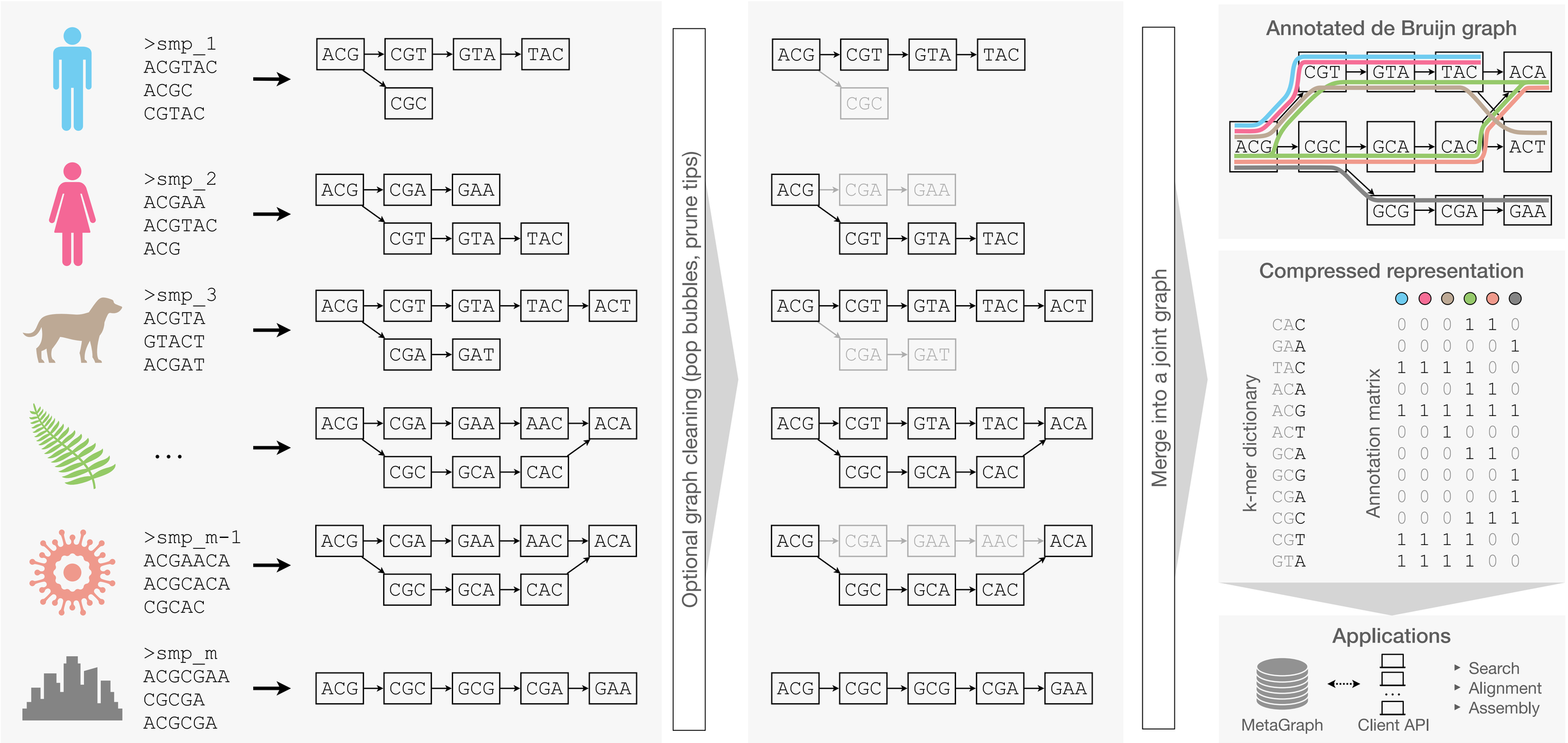
Indexing workflow



Indexing workflow

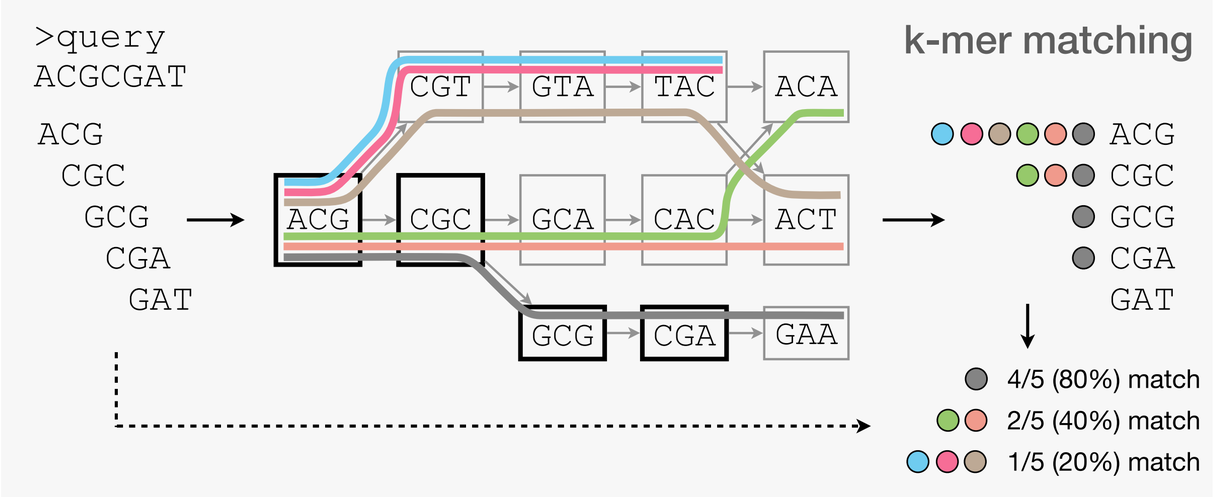


Indexing workflow

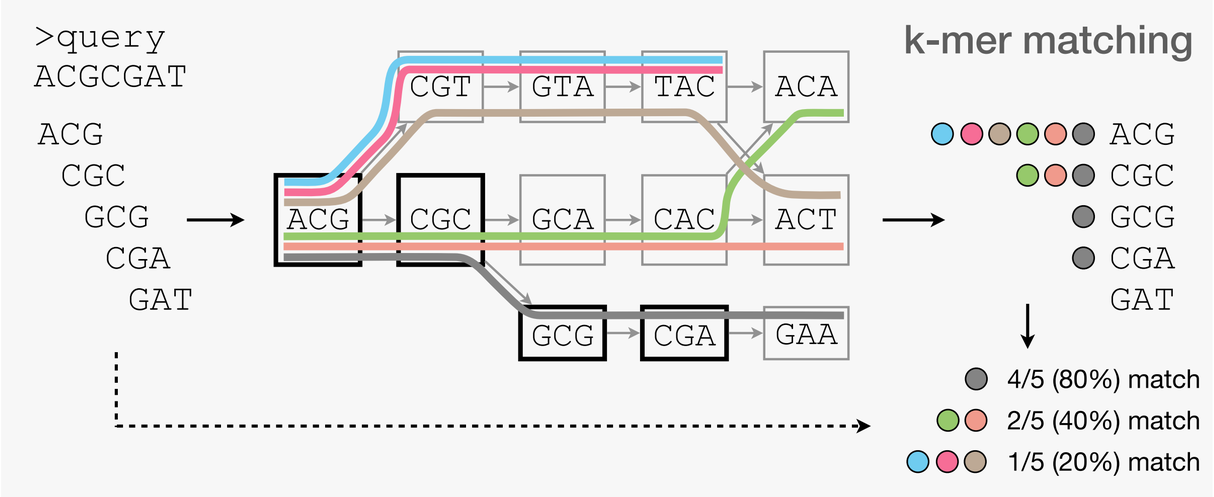


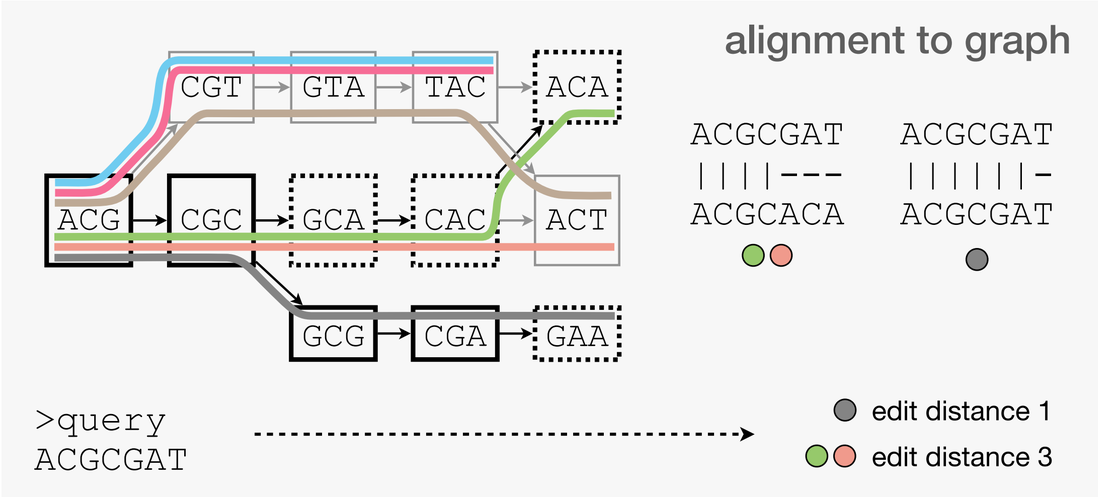
Sequence search

Sequence search

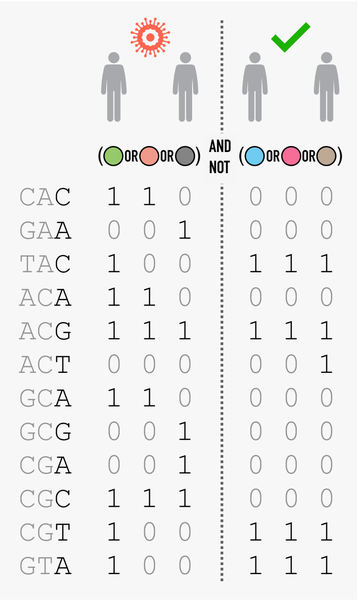


Sequence search

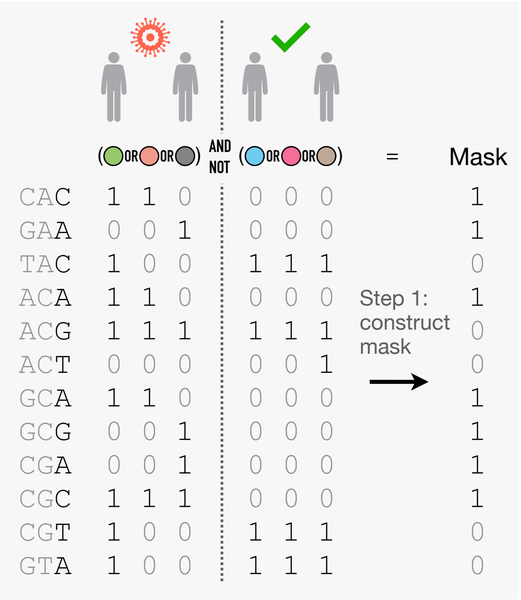




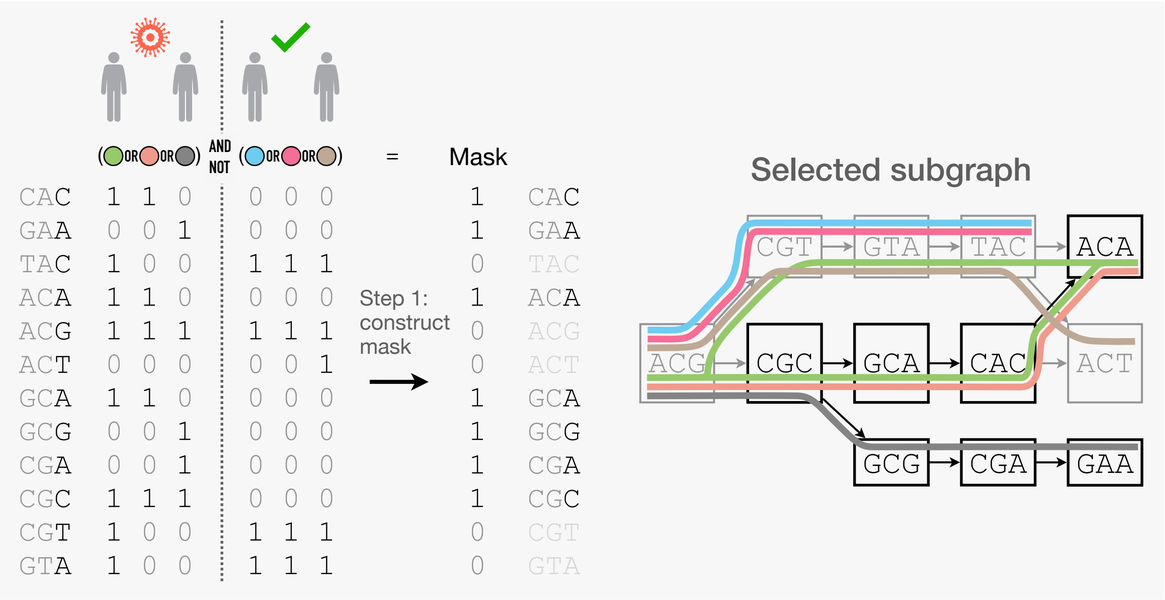
Differential assembly



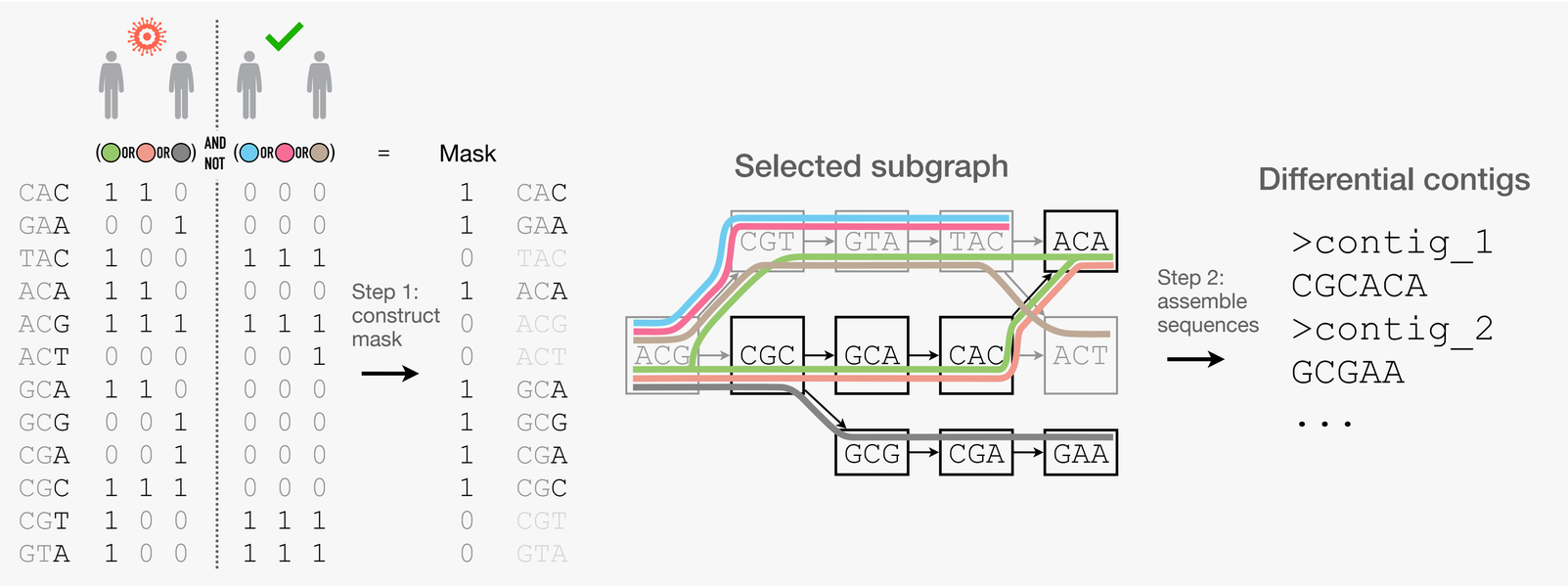
Differential assembly



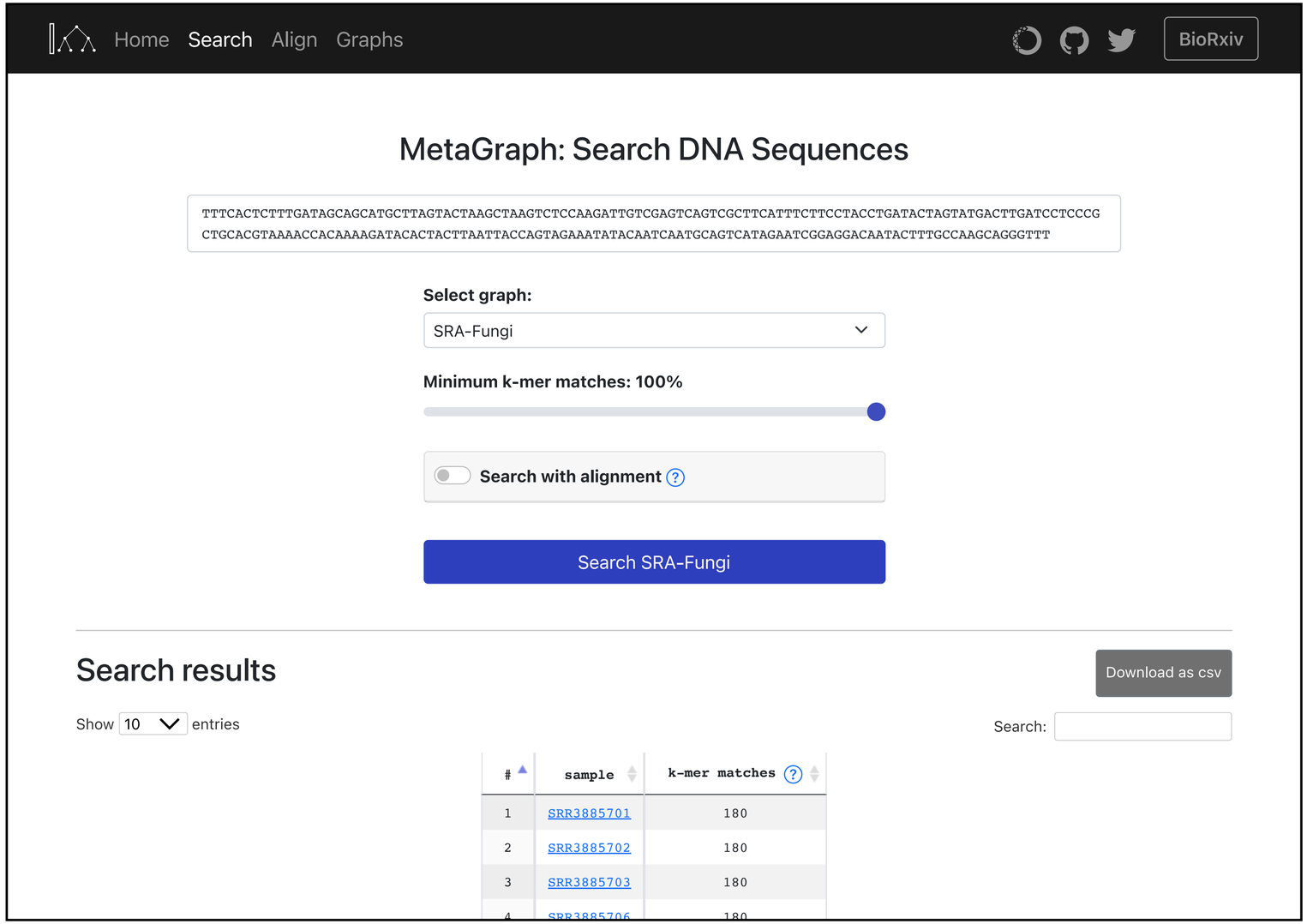
Differential assembly



Differential assembly

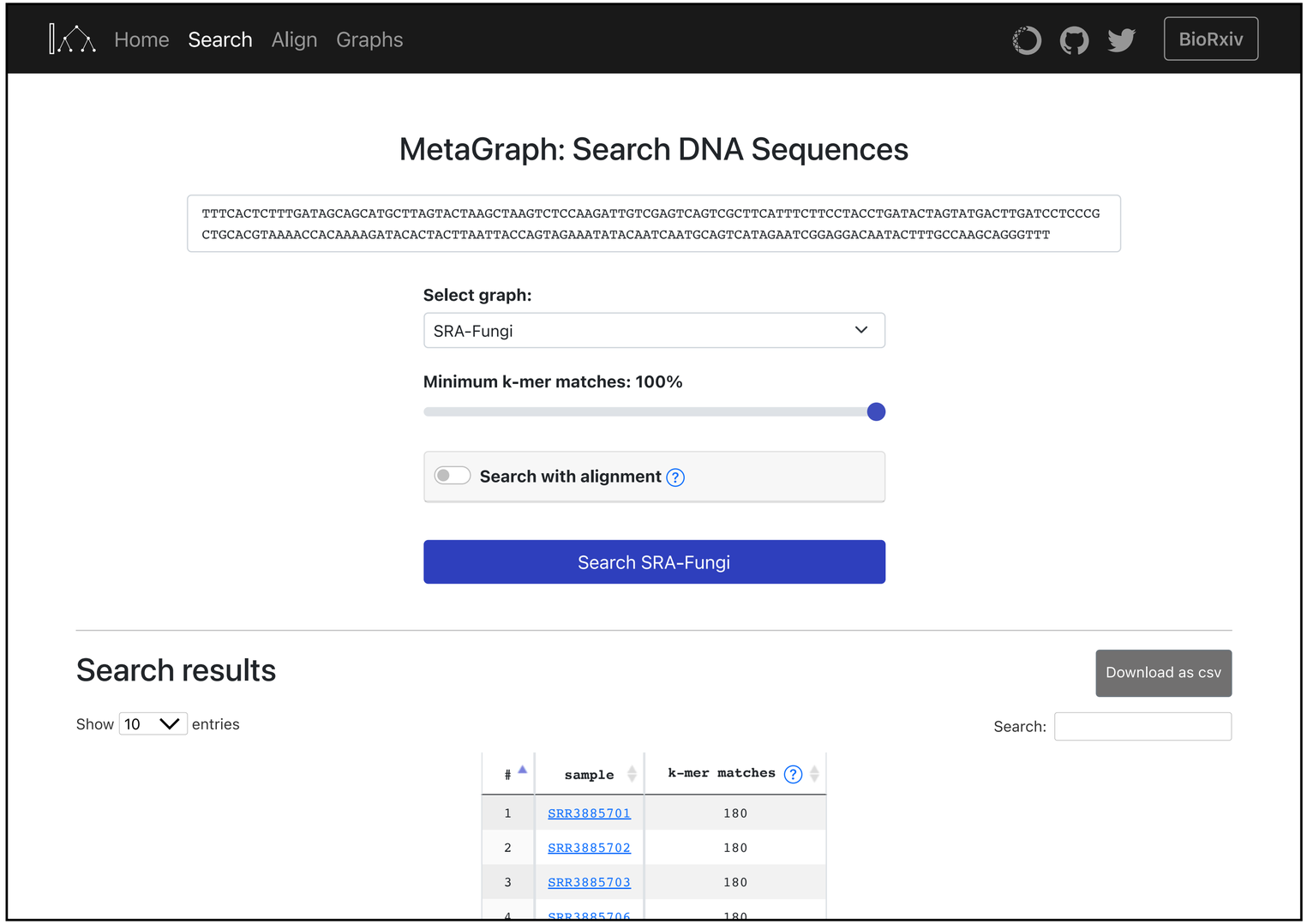


MetaGraph Online



metagraph.ethz.ch/search

MetaGraph Online



MetaGraph

metagraph.ethz.ch/search

Python Client API

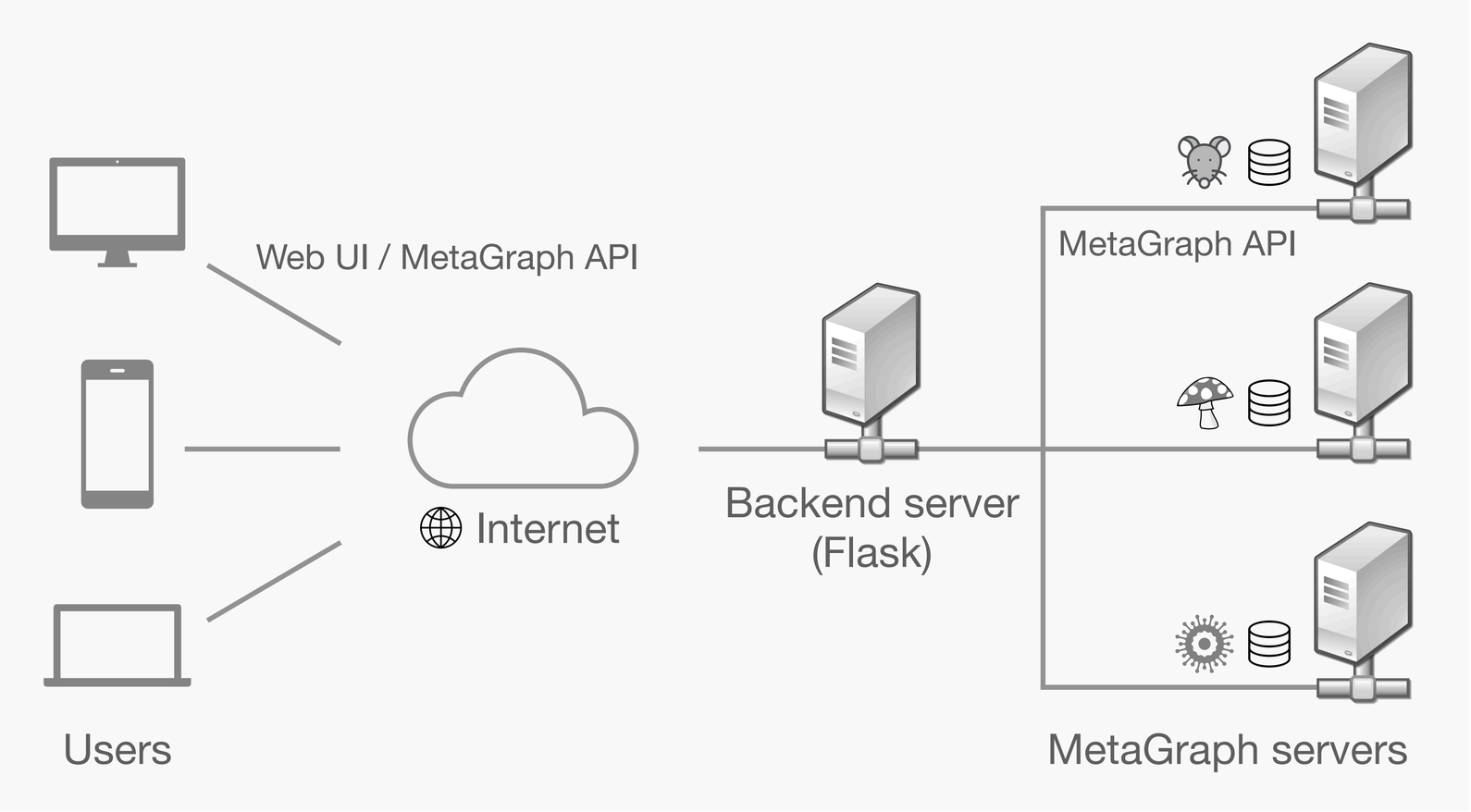
In [1]:
```
from metagraph.client import GraphClient

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
...	...	...	...

MetaGraph Online (architecture)



metagraph.ethz.ch/search

Why MetaGraph

MetaGraph

metagraph.ethz.ch/search

Why MetaGraph

- MetaGraph is an open source **modular framework**
  - various (succinct) graph representations (Bowe *et al.*, 2012; Conway *et al.*, 2011)
  - different schemes for annotation representation (Almodaresi et al., 2017, Muggli et al., 2017, Karasikov et al., 2019, Danciu et al., 2021, Karasikov et al., 2022, …)
  - also available as an Anaconda package or Docker container

MetaGraph

metagraph.ethz.ch/search

Why MetaGraph

- MetaGraph is an open source **modular framework**
  - various (succinct) graph representations (Bowe *et al.*, 2012; Conway *et al.*, 2011)
  - different schemes for annotation representation (Almodaresi et al., 2017, Muggli et al., 2017, Karasikov et al., 2019, Danciu et al., 2021, Karasikov et al., 2022, …)
  - also available as an Anaconda package or Docker container
- MetaGraph is **highly scalable**
  - scalable construction ⇒ works on Petabase size inputs
  - fast query ⇒ can query millions of sequences per hour
  - distributed representation ⇒ flexible API for client / server setup

MetaGraph

metagraph.ethz.ch/search

Why MetaGraph

- MetaGraph is an open source **modular framework**
  - various (succinct) graph representations (Bowe *et al.*, 2012; Conway *et al.*, 2011)
  - different schemes for annotation representation (Almodaresi et al., 2017, Muggli et al., 2017, Karasikov et al., 2019, Danciu et al., 2021, Karasikov et al., 2022, …)
  - also available as an Anaconda package or Docker container
- MetaGraph is **highly scalable**
  - scalable construction ⇒ works on Petabase size inputs
  - fast query ⇒ can query millions of sequences per hour
  - distributed representation ⇒ flexible API for client / server setup
- Compressed and **lossless k-mer set representation**
  - up to 8,000x compression on transcriptome data

MetaGraph

metagraph.ethz.ch/search

Why MetaGraph

- MetaGraph is an open source **modular framework**
  - various (succinct) graph representations (Bowe *et al.*, 2012; Conway *et al.*, 2011)
  - different schemes for annotation representation (Almodaresi et al., 2017, Muggli et al., 2017, Karasikov et al., 2019, Danciu et al., 2021, Karasikov et al., 2022, …)
  - also available as an Anaconda package or Docker container
- MetaGraph is **highly scalable**
  - scalable construction ⇒ works on Petabase size inputs
  - fast query ⇒ can query millions of sequences per hour
  - distributed representation ⇒ flexible API for client / server setup
- Compressed and **lossless k-mer set representation**
  - up to 8,000x compression on transcriptome data
- Sensitive **sequence-to-graph alignment**
  - efficient sub-k seeding and performant chaining

MetaGraph

metagraph.ethz.ch/search

Why MetaGraph

- MetaGraph is an open source **modular framework**
  - various (succinct) graph representations (Bowe *et al.*, 2012; Conway *et al.*, 2011)
  - different schemes for annotation representation (Almodaresi et al., 2017, Muggli et al., 2017, Karasikov et al., 2019, Danciu et al., 2021, Karasikov et al., 2022, …)
  - also available as an Anaconda package or Docker container
- MetaGraph is **highly scalable**
  - scalable construction ⇒ works on Petabase size inputs
  - fast query ⇒ can query millions of sequences per hour
  - distributed representation ⇒ flexible API for client / server setup
- Compressed and **lossless k-mer set representation**
  - up to 8,000x compression on transcriptome data
- Sensitive **sequence-to-graph alignment**
  - efficient sub-k seeding and performant chaining
- Support for **integrative analysis**
  - novel concept of differential assembly

MetaGraph

metagraph.ethz.ch/search

Why MetaGraph

- MetaGraph is an open source **modular framework**
  - various (succinct) graph representations (Bowe *et al.*, 2012; Conway *et al.*, 2011)
  - different schemes for annotation representation (Almodaresi et al., 2017, Muggli et al., 2017, Karasikov et al., 2019, Danciu et al., 2021, Karasikov et al., 2022, …)
  - also available as an Anaconda package or Docker container
- MetaGraph is **highly scalable**
  - scalable construction ⇒ works on Petabase size inputs
  - fast query ⇒ can query millions of sequences per hour
  - distributed representation ⇒ flexible API for client / server setup
- Compressed and **lossless k-mer set representation**
  - up to 8,000x compression on transcriptome data
- Sensitive **sequence-to-graph alignment**
  - efficient sub-k seeding and performant chaining
- Support for **integrative analysis**
  - novel concept of differential assembly

MetaGraph

metagraph.ethz.ch/search

Now also
- Indexing quantitative data
  - **k-mer abundances**
  - **k-mer coordinates** (positions in genomes)
  - fully lossless representation

Why MetaGraph

- MetaGraph is an open source **modular framework**
  - various (succinct) graph representations (Bowe *et al.*, 2012; Conway *et al.*, 2011)
  - different schemes for annotation representation (Almodaresi et al., 2017, Muggli et al., 2017, Karasikov et al., 2019, Danciu et al., 2021, Karasikov et al., 2022, …)
  - also available as an Anaconda package or Docker container
- MetaGraph is **highly scalable**
  - scalable construction ⇒ works on Petabase size inputs
  - fast query ⇒ can query millions of sequences per hour
  - distributed representation ⇒ flexible API for client / server setup
- Compressed and **lossless k-mer set representation**
  - up to 8,000x compression on transcriptome data
- Sensitive **sequence-to-graph alignment**
  - efficient sub-k seeding and performant chaining
- Support for **integrative analysis**
  - novel concept of differential assembly

MetaGraph

metagraph.ethz.ch/search

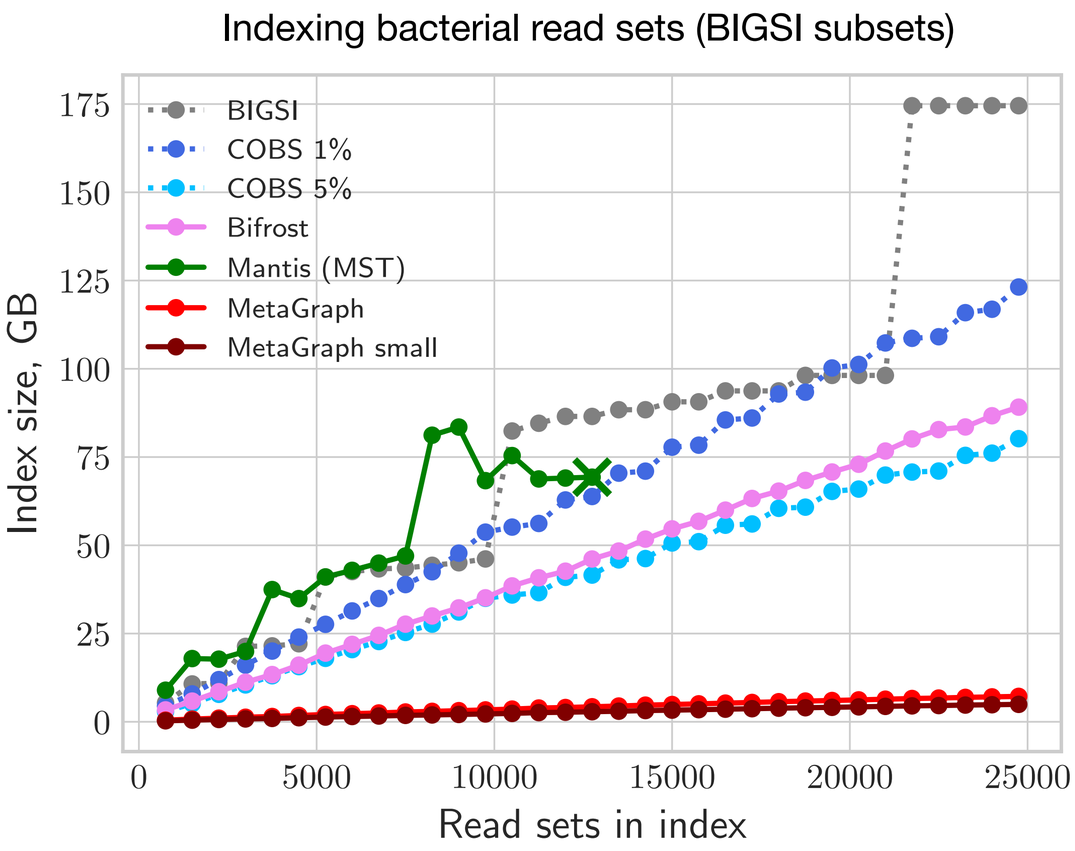
Now also

- Indexing quantitative data
  - **k-mer abundances**
  - **k-mer coordinates** (positions in genomes)
  - fully lossless representation

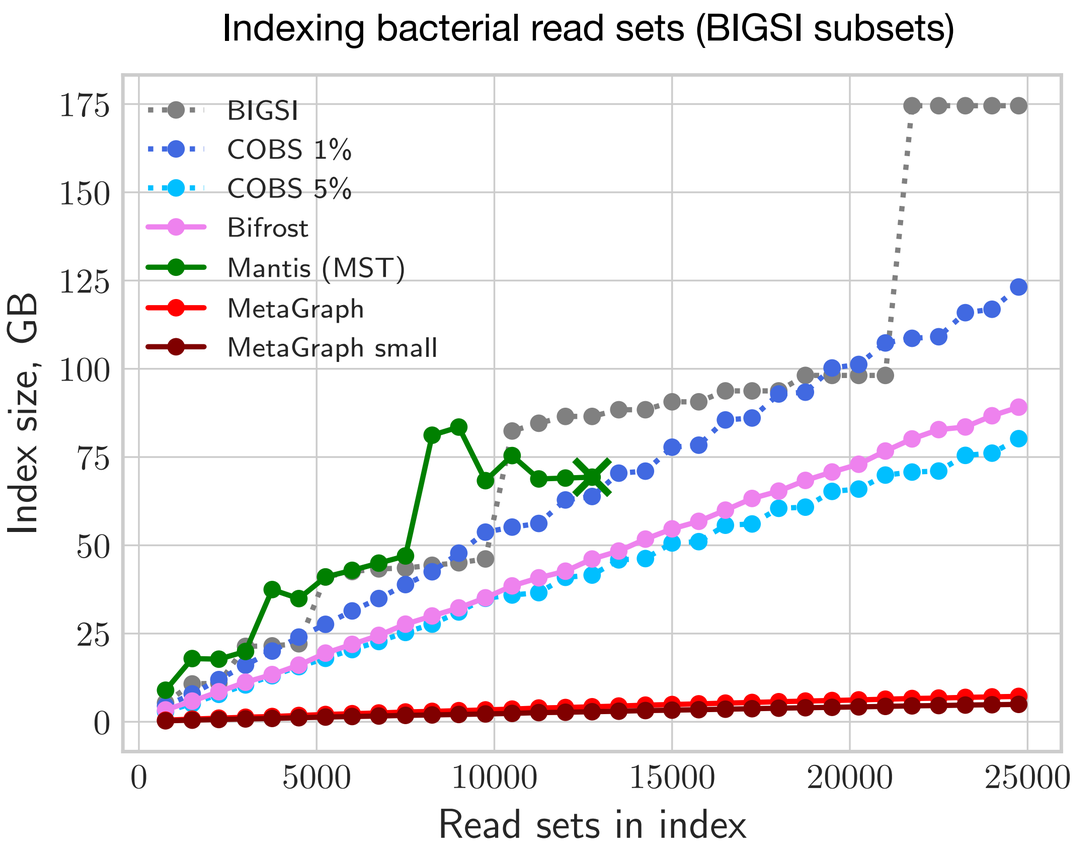
Soon: memory mapping (by Marek Kokot)

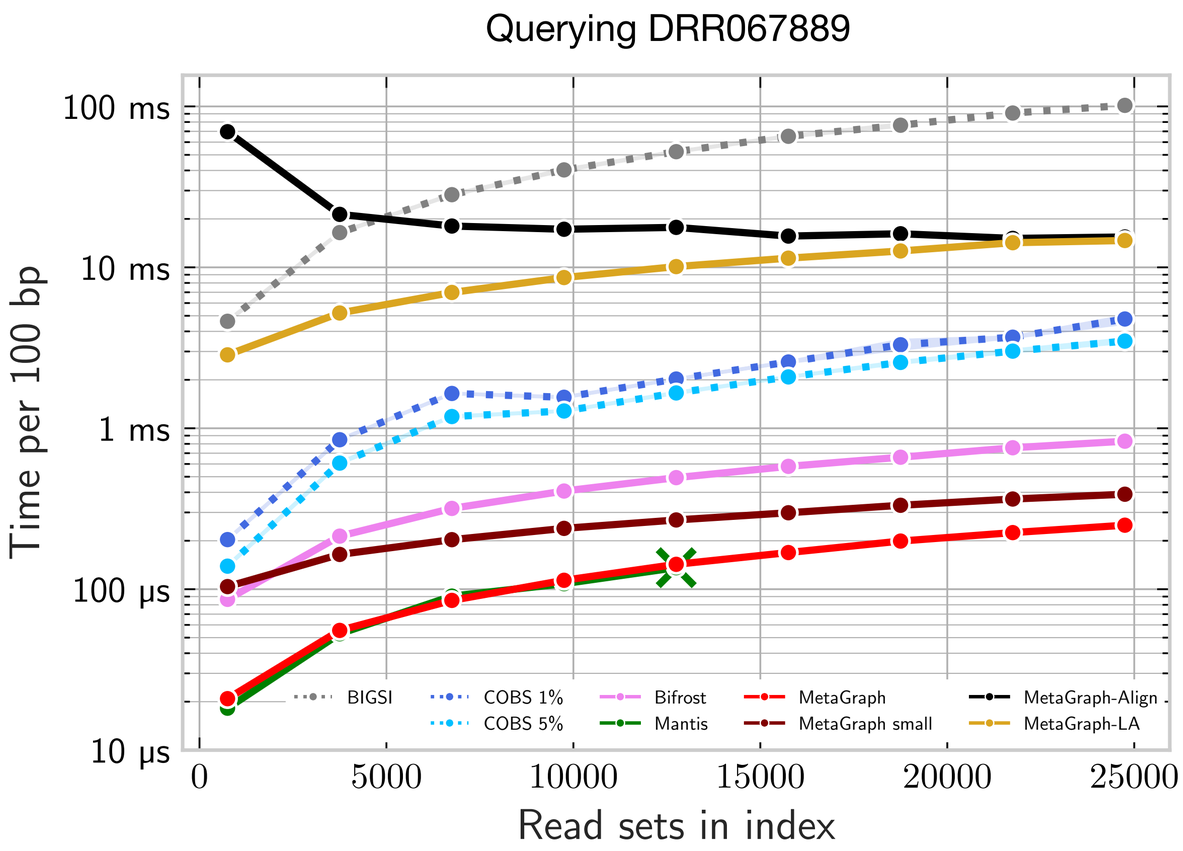
- host terabyte-size indexes with **less RAM**

Scalability of MetaGraph

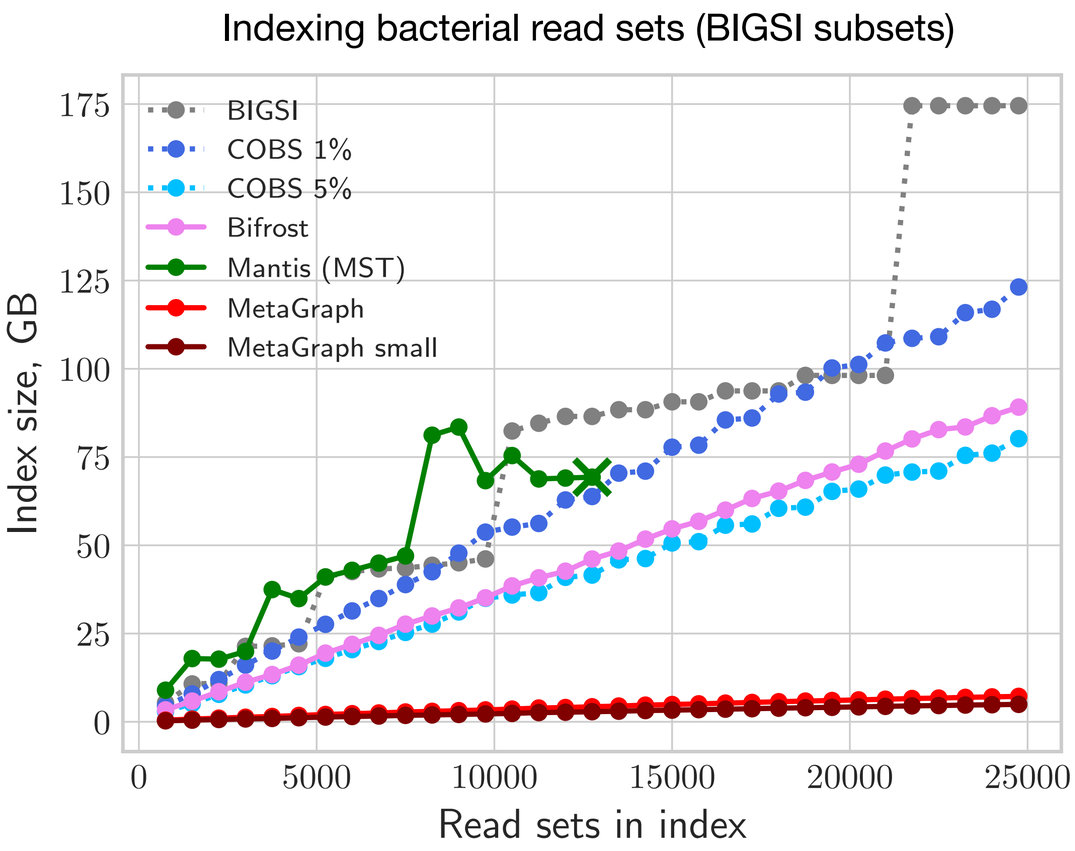


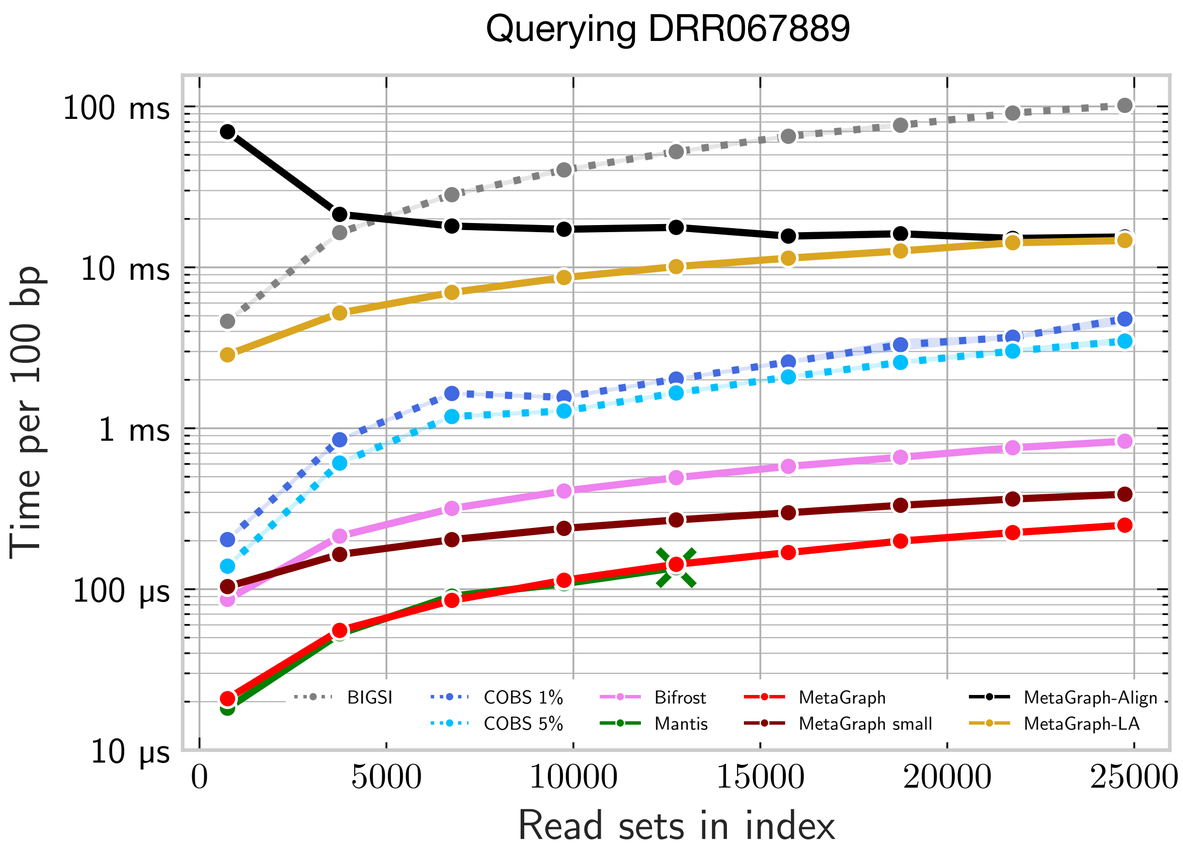
Scalability of MetaGraph





Scalability of MetaGraph

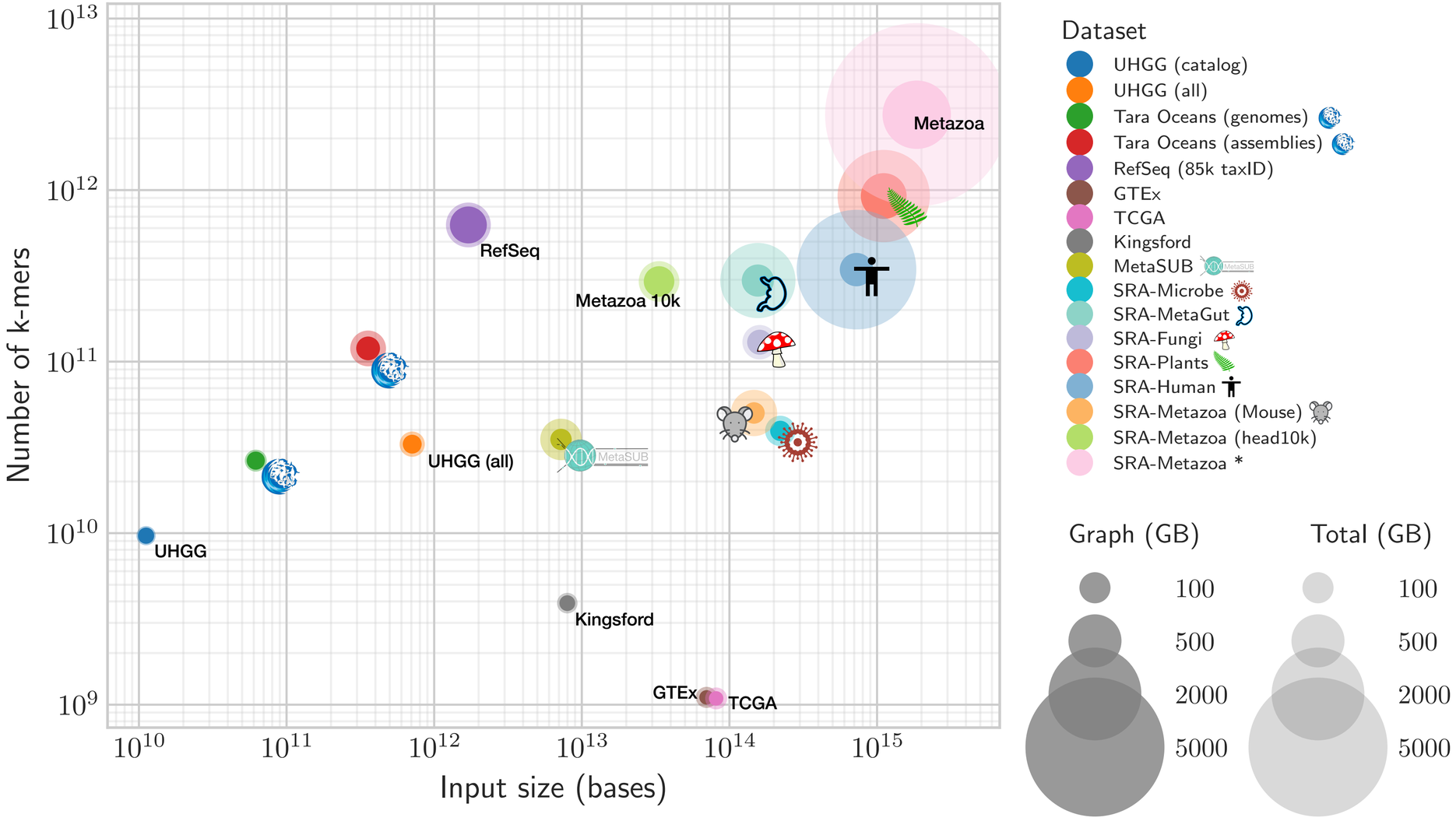




Achieved by:

- Use of **succinct data structures** and **efficient representation** schemes
  - algorithmic choices that work efficiently with succinct data structures (e.g., batch operations)

Indexing petabase-scale inputs



Datasets: UHGG (Almeida *et al.*, 2020), Tara Oceans (Paoli, 2022), GTEx (Lonsdale *et al.*, 2013), TCGA (Kaiser, 2005), MetaSUB (Danko *et al.*, 2019), SRA-Microbe (Bradley *et al.*, 2019)

Indexing petabase-scale inputs

Dataset	Tbp	Input (gz)	k	# k-mers	# labels	Graph	Annot.	Index	Ratio
UHGG (catalog)	0.01	3.3 GB	31	$9.7 \cdot 10^9$	4,644	2.6 GB	0.6 GB	3.2 GB	1.0×
UHGG (all)	0.71	206.0 GB	31	$33.0 \cdot 10^9$	286,997	9.6 GB	17.7 GB	27.3 GB	7.6×
Tara Oceans (genomes)	0.06	17.9 GB	31	$26.5 \cdot 10^9$	6,414,647	7.3 GB	3.8 GB	11.1 GB	1.6×
Tara Oceans (assemblies)	0.36	106.8 GB	31	$119.4 \cdot 10^9$	318,205,057	34.4 GB	89.7 GB	124.1 GB	0.9×
RefSeq (85k taxID)	1.70	502.4 GB	31	$626.2 \cdot 10^9$	85,375	176.0 GB	87.5 GB	263.6 GB	1.9×
RefSeq (33M accessions)	1.70	502.5 GB	31	$626.2 \cdot 10^9$	32,881,348	176.0 GB	287.0 GB	463.1 GB	1.1×
Kingsford	8.0	2.9 TB	21	$3.9 \cdot 10^9$	2,652	1.8 GB	4.8 GB	6.6 GB	437×
GTEx	70.0	40.0 TB	41	$1.1 \cdot 10^9$	9,759	0.4 GB	8.0 GB	8.4 GB	4,742×
TCGA	81.2	65.0 TB	31	$1.1 \cdot 10^9$	11,095	0.4 GB	10.7 GB	11.1 GB	5,831×
MetaSUB	7.2	5.5 TB	19	$35.2 \cdot 10^9$	4,220	20.5 GB	185.9 GB	206.4 GB	27×
SRA-MetaGut	155.8	86.0 TB	31	$296.9 \cdot 10^9$	242,619	112.2 GB	999.0 GB	1,111.3 GB	77×
SRA-Microbe	221.1	170.0 TB	31	$39.5 \cdot 10^9$	446,506	15.4 GB	50.1 GB	65.5 GB	2,595×
SRA-Fungi	160.2	80.0 TB	31	$129.7 \cdot 10^9$	121,900	43.5 GB	64.7 GB	108.1 GB	740×
SRA-Plants	1,109.2	575.9 TB	31	$923.4 \cdot 10^9$	531,736	333.8 GB	1,510.3 GB	1,844.1 GB	312×
SRA-Human	725.4	345.7 TB	31	$343.9 \cdot 10^9$	436,502	127.6 GB	3,274.5 GB	3,402.1 GB	102×
SRA-Metazoa (Mouse)	146.6	61.3 TB	31	$50.2 \cdot 10^9$	57,938	21.1 GB	270.5 GB	291.6 GB	210×
SRA-Metazoa (10k)	33.4	16.5 TB	31	$293.3 \cdot 10^9$	10,000	94.0 GB	98.7 GB	192.7 GB	86×
SRA-Metazoa *	1,856.8	925.3 TB	31	$2749.8 \cdot 10^9$	797,883	960.4 GB	7,898.4 GB	8,858.8 GB	104×

Datasets: UHGG (Almeida *et al.*, 2020), Tara Oceans (Paoli, 2022), GTEx (Lonsdale *et al.*, 2013), TCGA (Kaiser, 2005), MetaSUB (Danko *et al.*, 2019), SRA-Microbe (Bradley *et al.*, 2019)

Indexing petabase-scale inputs

Dataset	Tbp	Input (gz)	k	# k-mers	# labels	Graph	Annot.	Index	Ratio
UHGG (catalog)	0.01	3.3 GB	31	$9.7 \cdot 10^9$	4,644	2.6 GB	0.6 GB	3.2 GB	1.0×
UHGG (all)	0.71	206.0 GB	31	$33.0 \cdot 10^9$	286,997	9.6 GB	17.7 GB	27.3 GB	7.6×
Tara Oceans (genomes)	[illegible]	[illegible]	[illegible]	[illegible]	4,647	7.3 GB	3.8 GB	11.1 GB	1.6×
Tara Oceans (assemblies)	0.36	106.8 GB	31	$119.4 \cdot 10^9$	318,205,057	34.4 GB	89.7 GB	124.1 GB	0.9×
RefSeq (85k taxID)	1.70	502.4 GB	31	$626.2 \cdot 10^9$	85,375	176.0 GB	87.5 GB	263.6 GB	1.9×
RefSeq (33M accessions)	1.70	502.5 GB	31	$626.2 \cdot 10^9$	32,881,348	176.0 GB	287.0 GB	463.1 GB	1.1×
Kingsford	8.0	2.9 TB	21	$3.9 \cdot 10^9$	2,652	1.8 GB	4.8 GB	6.6 GB	437×
GTEx	70.0	40.0 TB	41	$1.1 \cdot 10^9$	9,759	0.4 GB	8.0 GB	8.4 GB	4,742×
TCGA	81.2	65.0 TB	31	$1.1 \cdot 10^9$	11,095	0.4 GB	10.7 GB	11.1 GB	5,831×
MetaSUB	7.2	5.5 TB	19	$35.2 \cdot 10^9$	4,220	20.5 GB	185.9 GB	206.4 GB	27×
SRA-MetaGut	155.8	86.0 TB	31	$296.9 \cdot 10^9$	242,619	112.2 GB	999.0 GB	1,111.3 GB	77×
SRA-Microbe	221.1	170.0 TB	31	$39.5 \cdot 10^9$	446,506	15.4 GB	50.1 GB	65.5 GB	2,595×
SRA-Fungi	160.2	80.0 TB	31	$129.7 \cdot 10^9$	121,900	43.5 GB	64.7 GB	108.1 GB	740×
SRA-Plants	1,109.2	575.9 TB	31	$923.4 \cdot 10^9$	531,736	333.8 GB	1,510.3 GB	1,844.1 GB	312×
SRA-Human	725.4	345.7 TB	31	$343.9 \cdot 10^9$	436,502	127.6 GB	3,274.5 GB	3,402.1 GB	102×
SRA-Metazoa (Mouse)	146.6	61.3 TB	31	$50.2 \cdot 10^9$	57,938	21.1 GB	270.5 GB	291.6 GB	210×
SRA-Metazoa (10k)	33.4	16.5 TB	31	$293.3 \cdot 10^9$	10,000	94.0 GB	98.7 GB	192.7 GB	86×
SRA-Metazoa *	1,856.8	925.3 TB	31	$2749.8 \cdot 10^9$	797,883	960.4 GB	7,898.4 GB	8,858.8 GB	104×

Can we index k-mer positions as well?

Datasets: UHGG (Almeida *et al.*, 2020), Tara Oceans (Paoli, 2022), GTEx (Lonsdale *et al.*, 2013), TCGA (Kaiser, 2005), MetaSUB (Danko *et al.*, 2019), SRA-Microbe (Bradley *et al.*, 2019)

Indexing petabase-scale inputs

Dataset	Tbp	Input (gz)	k	# k-mers	# labels	Graph	Annot.	Index	Ratio
UHGG (catalog)	0.01	3.3 GB	31	$9.7 \cdot 10^9$	4,644	2.6 GB	0.6 GB	3.2 GB	1.0×
UHGG (all)	0.71	206.0 GB	31	$33.0 \cdot 10^9$	286,997	9.6 GB	17.7 GB	27.3 GB	7.6×
Tara Oceans (genomes)	[illegible]	[illegible]	[illegible]	[illegible]	4,647	7.3 GB	3.8 GB	11.1 GB	1.6×
Tara Oceans (assemblies)	0.36	106.8 GB	31	$119.4 \cdot 10^9$	318,205,057	34.4 GB	89.7 GB	124.1 GB	0.9×
RefSeq (85k taxID)	1.70	502.4 GB	31	$626.2 \cdot 10^9$	85,375	176.0 GB	87.5 GB	263.6 GB	1.9×
RefSeq (33M accessions)	1.70	502.5 GB	31	$626.2 \cdot 10^9$	32,881,348	176.0 GB	287.0 GB	463.1 GB	1.1×
Kingsford	8.0	2.9 TB	21	$3.9 \cdot 10^9$	2,652	1.8 GB	4.8 GB	6.6 GB	437×
GTEx	[illegible]	[illegible]	41	$1.1 \cdot 10^9$	9,759	0.4 GB	8.0 GB	8.4 GB	4,742×
TCGA	81.2	65.0 TB	31	$1.1 \cdot 10^9$	11,095	0.4 GB	10.7 GB	11.1 GB	5,831×
MetaSUB	7.2	5.5 TB	19	$35.2 \cdot 10^9$	4,220	20.5 GB	185.9 GB	206.4 GB	27×
SRA-MetaGut	155.8	86.0 TB	31	$296.9 \cdot 10^9$	242,619	112.2 GB	999.0 GB	1,111.3 GB	77×
SRA-Microbe	221.1	170.0 TB	31	$39.5 \cdot 10^9$	446,506	15.4 GB	50.1 GB	65.5 GB	2,595×
SRA-Fungi	160.2	80.0 TB	31	$129.7 \cdot 10^9$	121,900	43.5 GB	64.7 GB	108.1 GB	740×
SRA-Plants	1,109.2	575.9 TB	31	$923.4 \cdot 10^9$	531,736	333.8 GB	1,510.3 GB	1,844.1 GB	312×
SRA-Human	725.4	345.7 TB	31	$343.9 \cdot 10^9$	436,502	127.6 GB	3,274.5 GB	3,402.1 GB	102×
SRA-Metazoa (Mouse)	146.6	61.3 TB	31	$50.2 \cdot 10^9$	57,938	21.1 GB	270.5 GB	291.6 GB	210×
SRA-Metazoa (10k)	33.4	16.5 TB	31	$293.3 \cdot 10^9$	10,000	94.0 GB	98.7 GB	192.7 GB	86×
SRA-Metazoa *	1,856.8	925.3 TB	31	$2749.8 \cdot 10^9$	797,883	960.4 GB	7,898.4 GB	8,858.8 GB	104×

Can we index k-mer positions as well?

What about expression?

Datasets: UHGG (Almeida *et al.*, 2020), Tara Oceans (Paoli, 2022), GTEx (Lonsdale *et al.*, 2013), TCGA (Kaiser, 2005), MetaSUB (Danko *et al.*, 2019), SRA-Microbe (Bradley *et al.*, 2019)

Background

De Bruijn Graphs

… widely used in Bioinformatics since 1989

Background

De Bruijn Graphs

… widely used in Bioinformatics since 1989

```
ACTAGCTAGCTAG
```

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

Background

De Bruijn Graphs

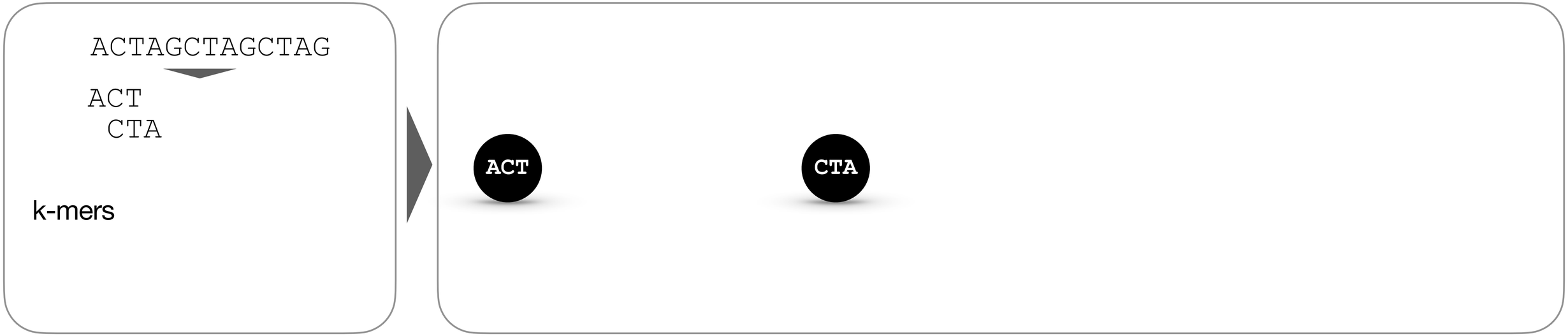
… widely used in Bioinformatics since 1989

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

Background

De Bruijn Graphs

… widely used in Bioinformatics since 1989

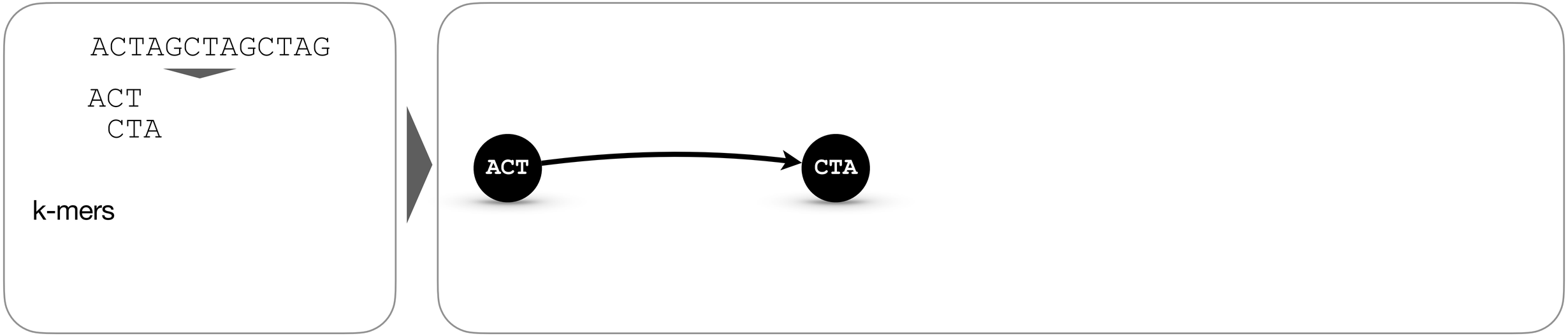


- Originally employed for ***de novo* assembly**

Background

De Bruijn Graphs

… widely used in Bioinformatics since 1989

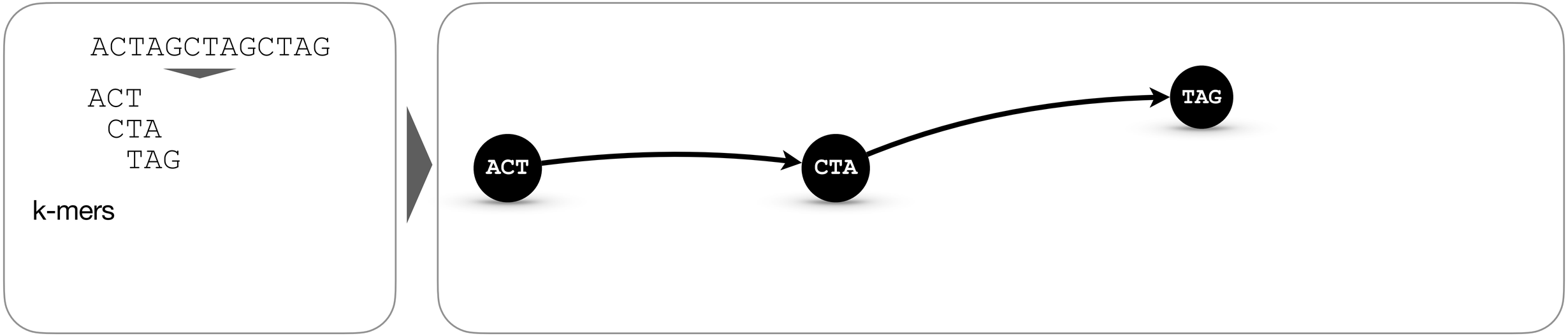


- Originally employed for ***de novo* assembly**

Background

De Bruijn Graphs

… widely used in Bioinformatics since 1989

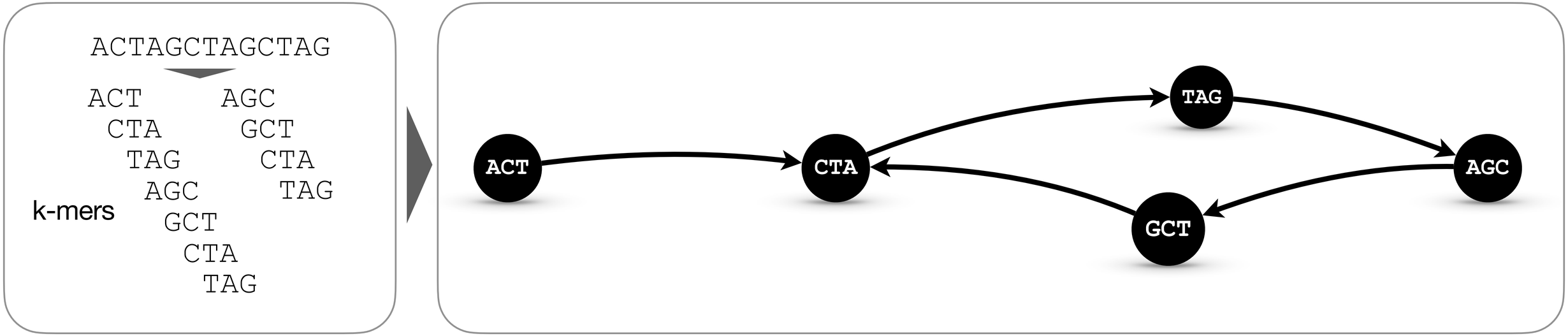


- Originally employed for ***de novo* assembly**

Background

De Bruijn Graphs

… widely used in Bioinformatics since 1989

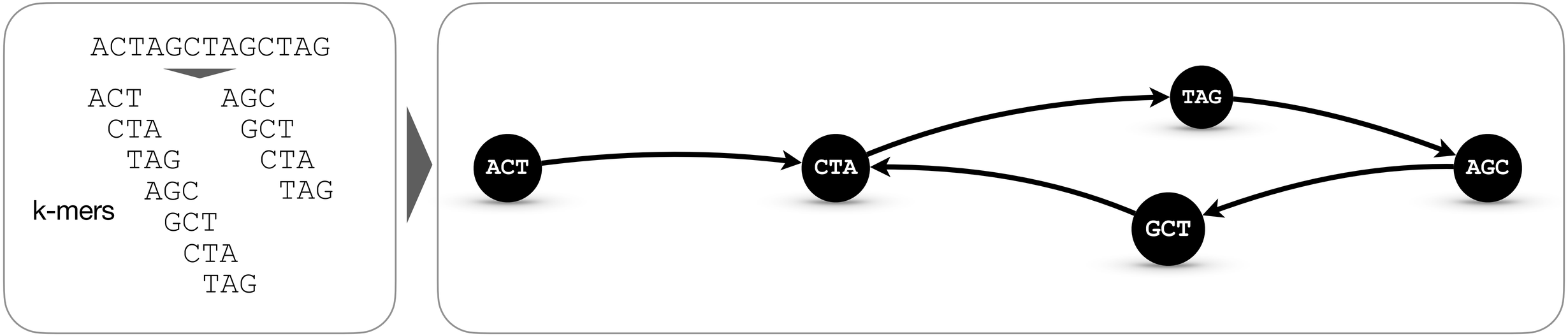


- Originally employed for ***de novo* assembly**

Background

De Bruijn Graphs

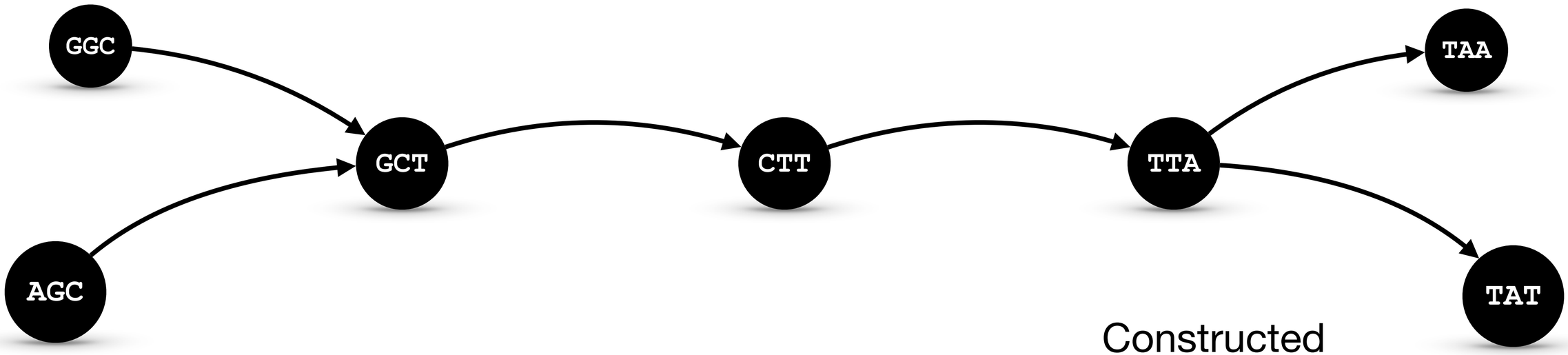
… widely used in Bioinformatics since 1989



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

Background

Annotated de Bruijn Graphs



Constructed
from sequences

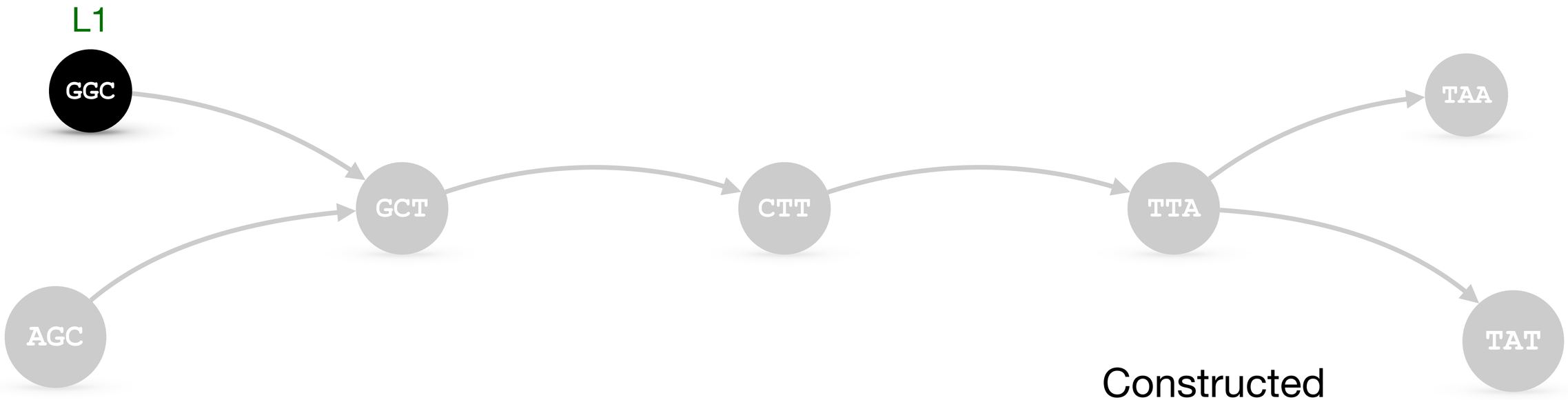
L1: GGCTTAT

L2: AGCTTAA

L3: TTAA

Background

Annotated de Bruijn Graphs



Constructed
from sequences

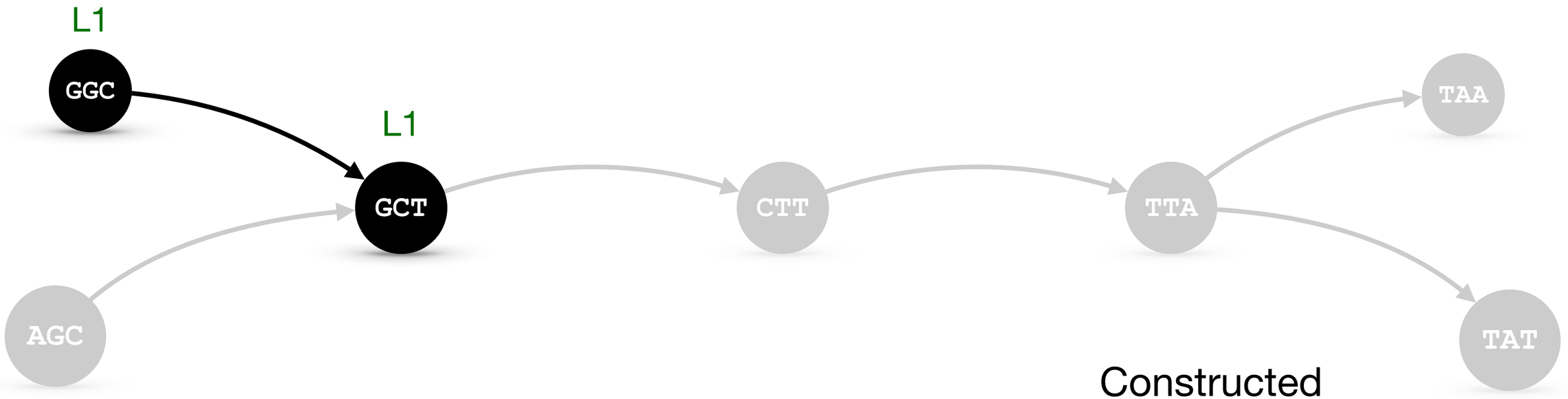
L1: GGCTTAT

L2: AGCTTAA

L3: TTAA

Background

Annotated de Bruijn Graphs



Constructed
from sequences

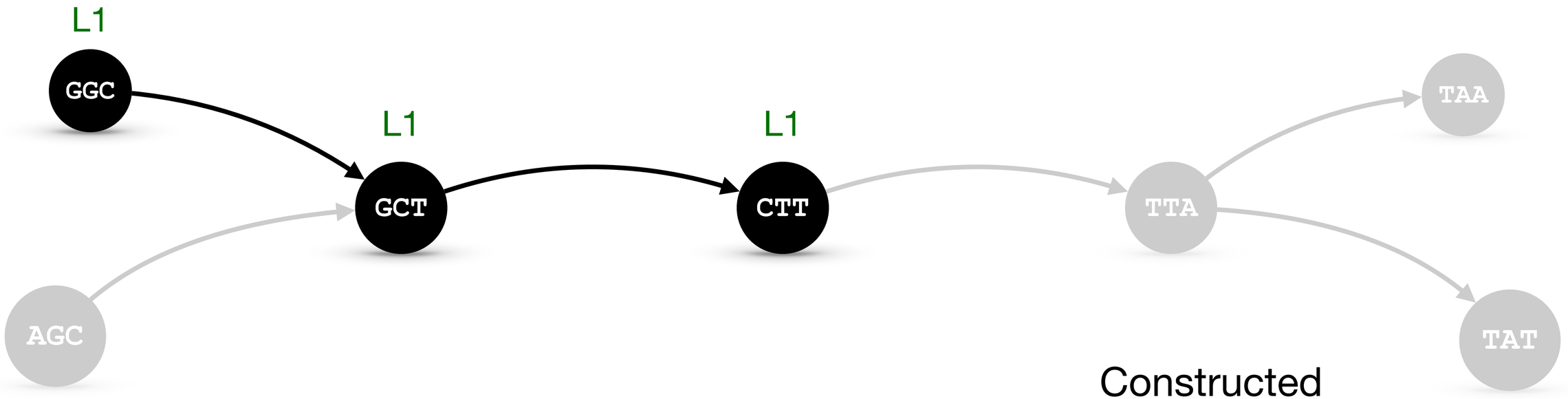
L1: GGCTTAT

L2: AGCTTAA

L3: TTAA

Background

Annotated de Bruijn Graphs



Constructed
from sequences

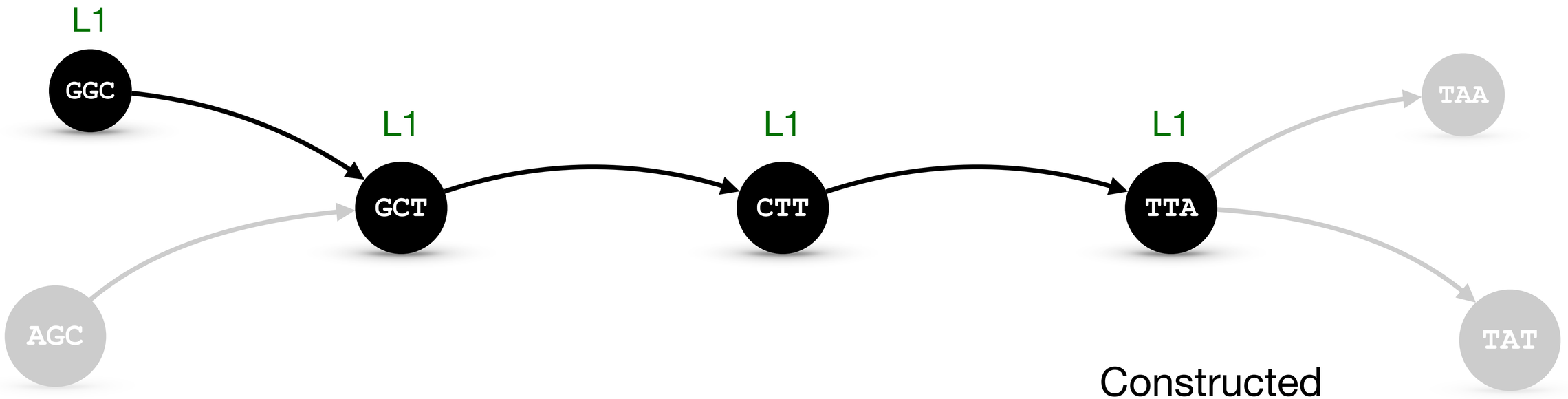
L1: GGCTTAT

L2: AGCTTAA

L3: TTAA

Background

Annotated de Bruijn Graphs



Constructed
from sequences

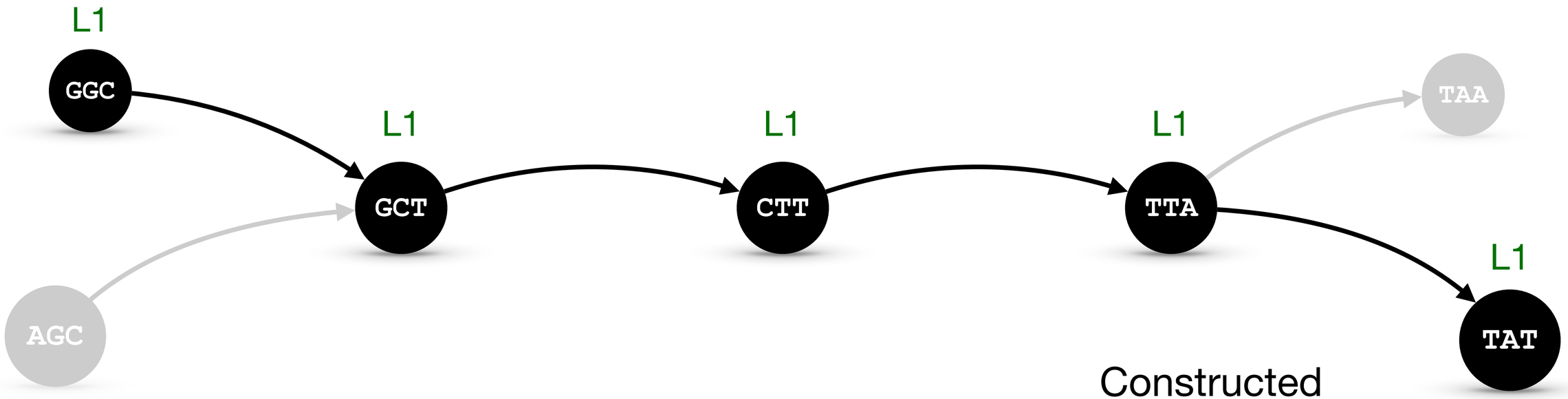
L1: GGCTTAT

L2: AGCTTAA

L3: TTAA

Background

Annotated de Bruijn Graphs



Constructed
from sequences

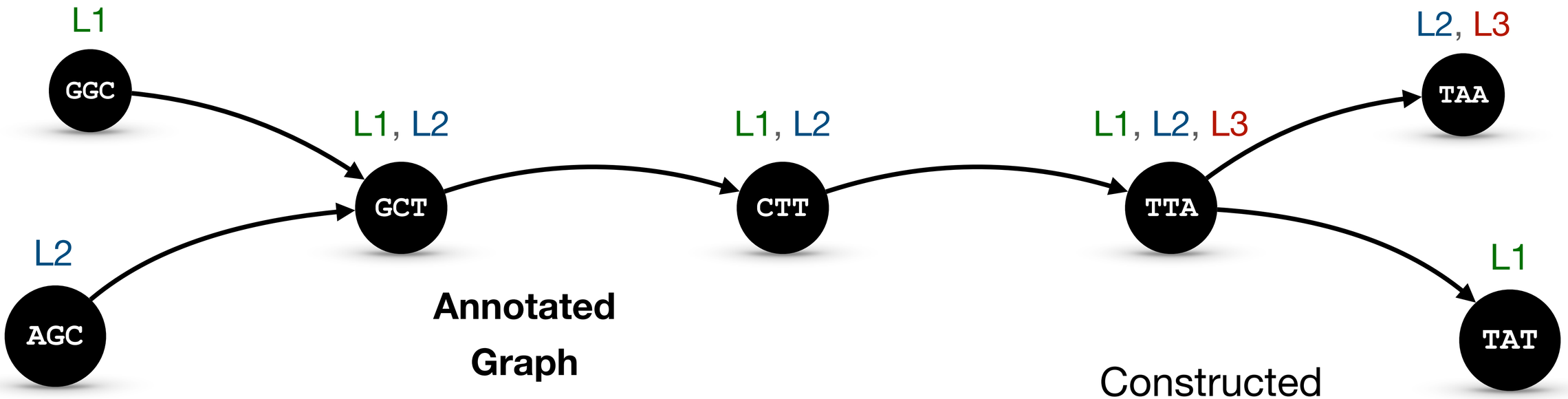
L1: GGCTTAT

L2: AGCTTAA

L3: TTAA

Background

Annotated de Bruijn Graphs



Annotated Graph

Constructed from sequences

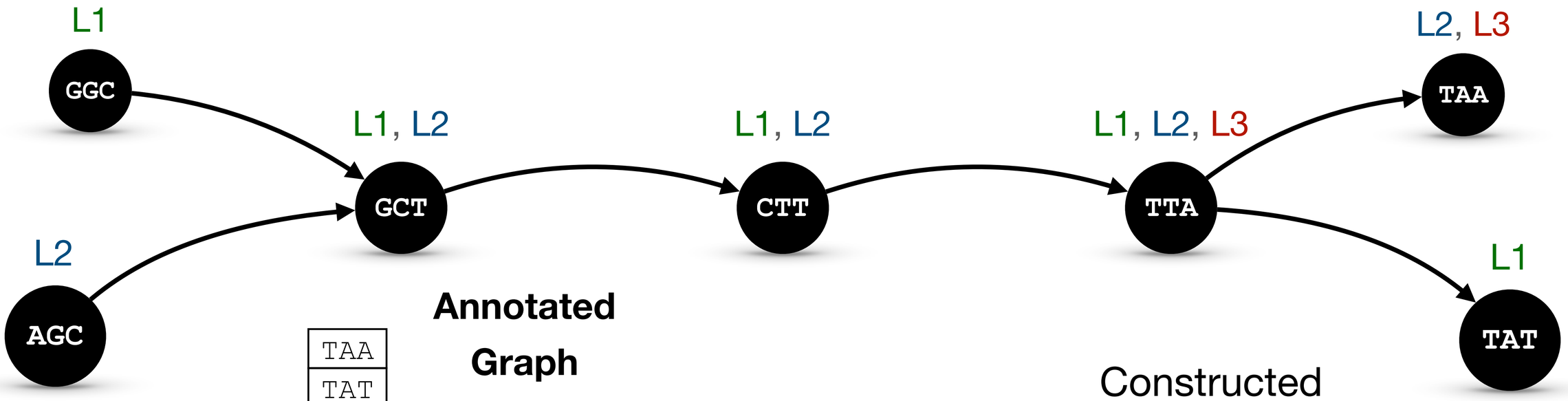
L1: GGCTTAT

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

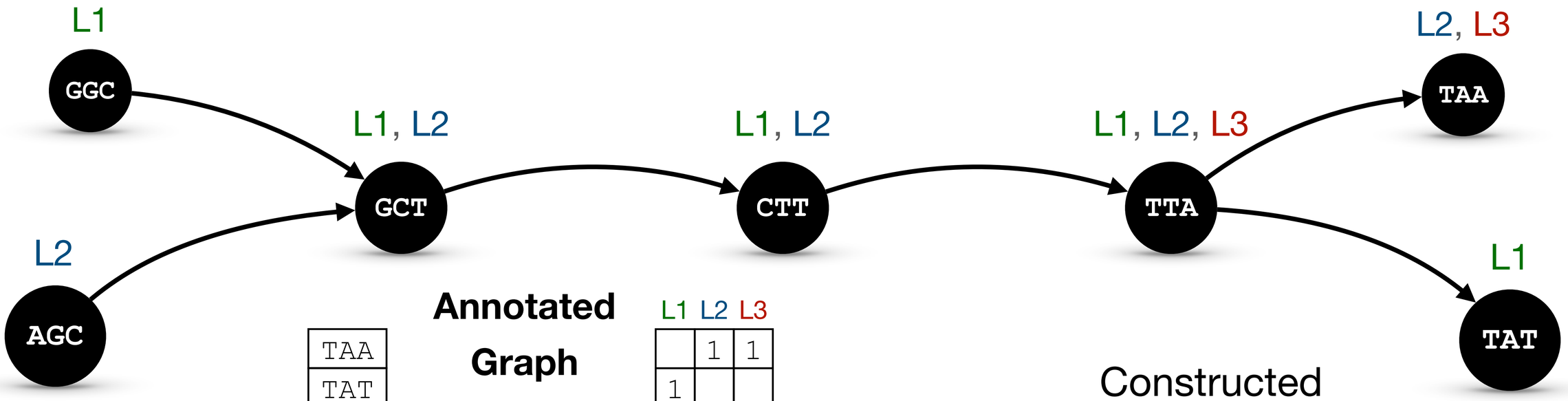
L1: GGCTTAT

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

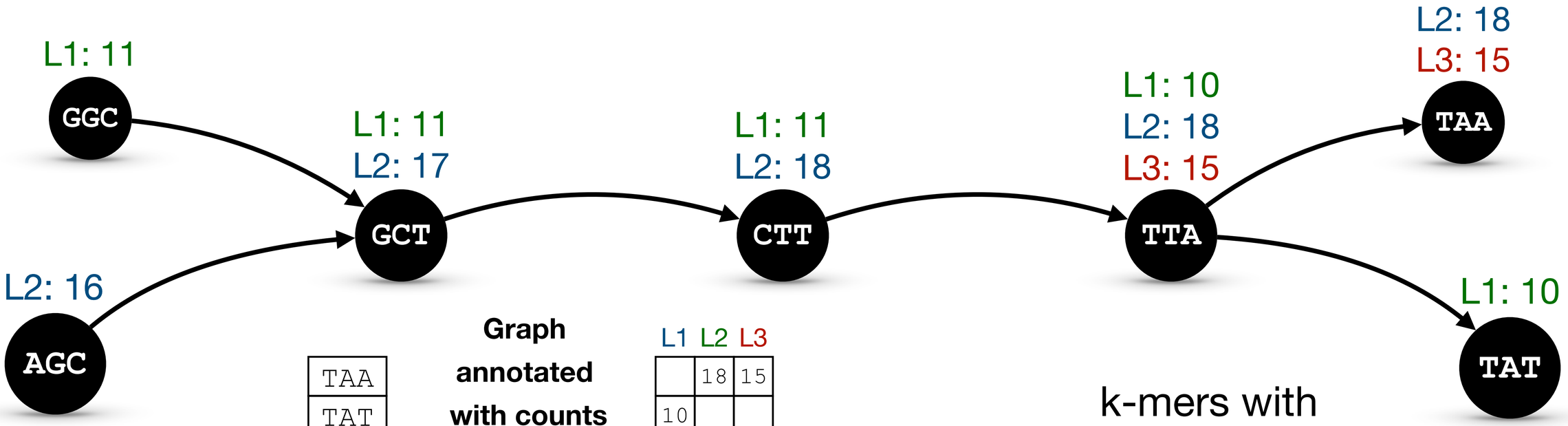
L1: GGCTTAT

L2: AGCTTAA

L3: TTAA

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		
11	17	
	16	
11		
11	18	
10	18	15

Graph annotation
(node labeling)

k-mers with multiplicities

L1: `GGC: 11, …`

L2: `AGC: 16, …`

L3: `TTA: 15, …`

Motivation

2. Representing **k-mer coordinates**

ACTAGCTAGCTAG

Motivation

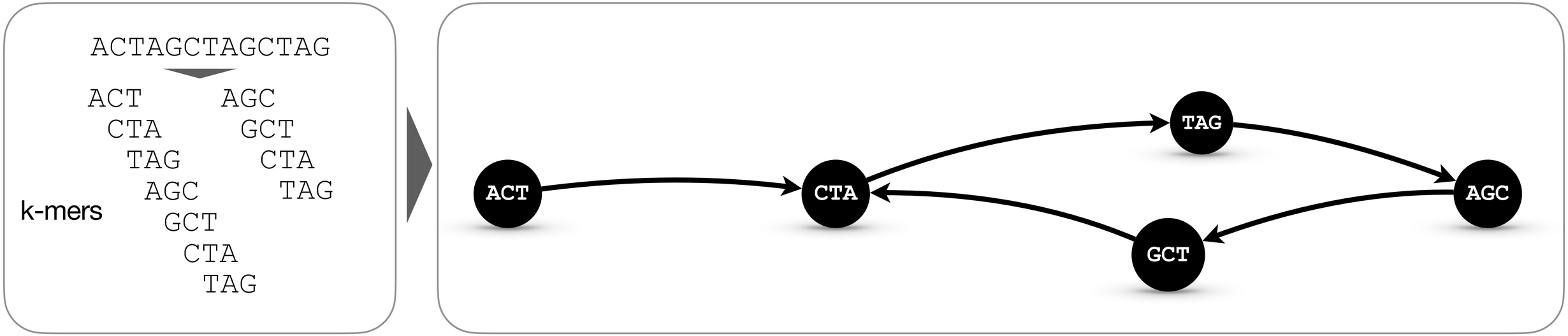
2. Representing **k-mer coordinates**

```
ACTAGCTAGCTAG
ACT     AGC
 CTA     GCT
  TAG     CTA
   AGC     TAG
    GCT
     CTA
      TAG
```

k-mers

Motivation

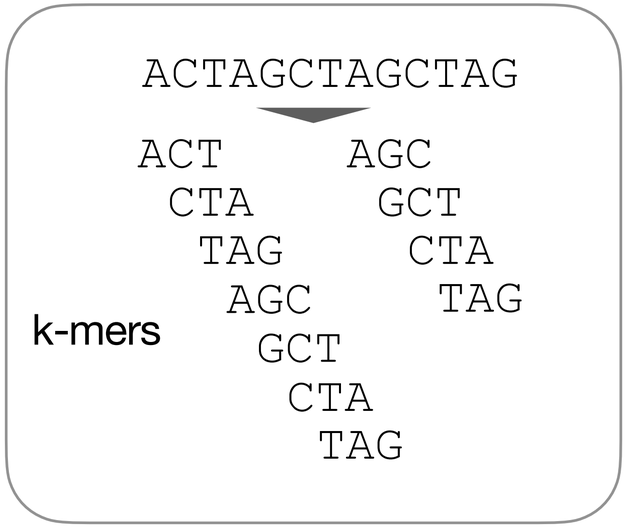
2. Representing **k-mer coordinates**

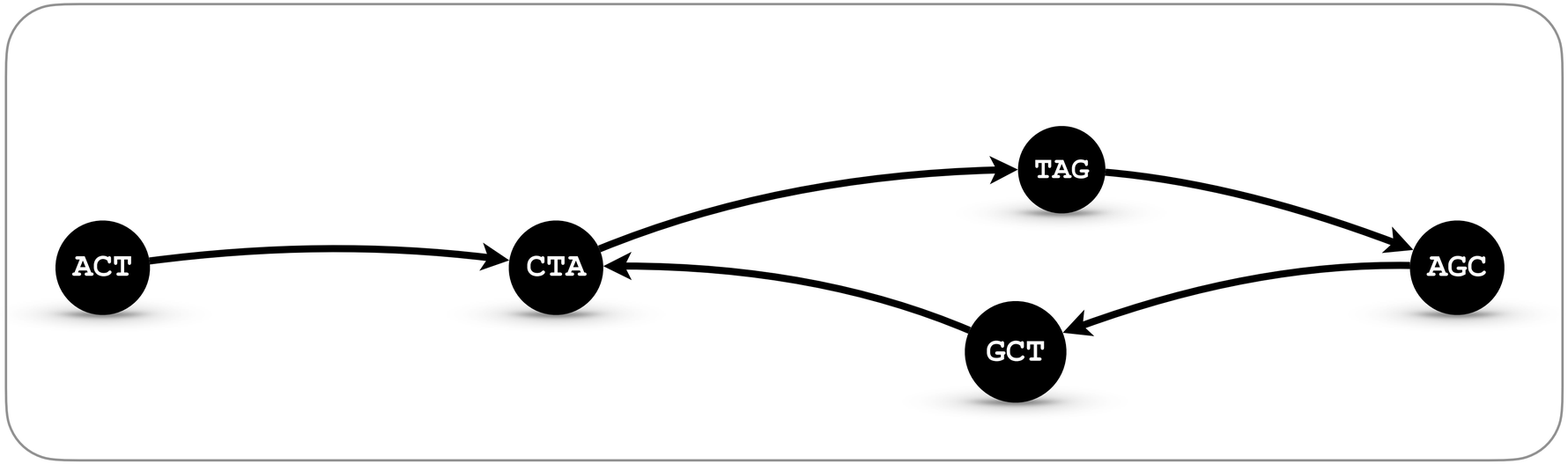


de Bruijn graph

Motivation

2. Representing **k-mer coordinates**



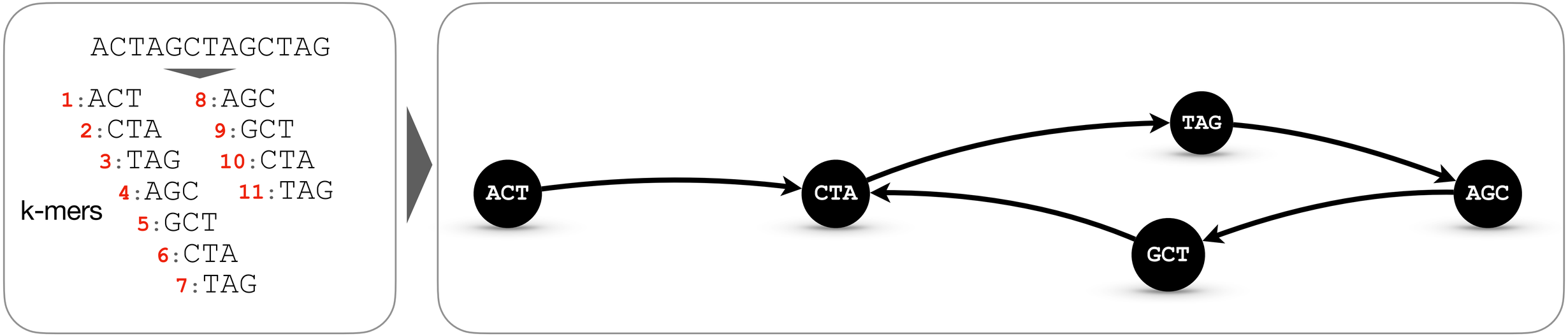


Not invertible representation 😕

de Bruijn graph

Motivation

2. Representing **k-mer coordinates**

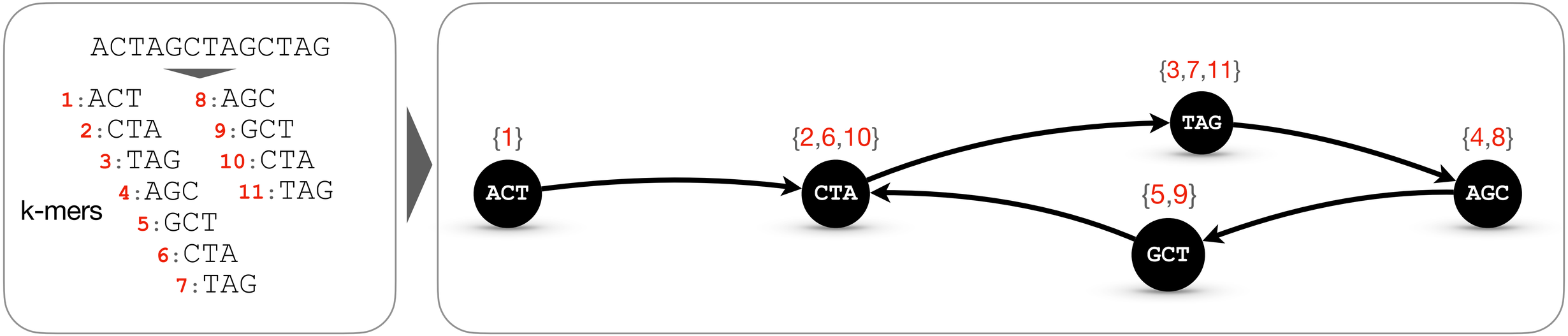


Not invertible representation 🙁

de Bruijn graph

Motivation

2. Representing **k-mer coordinates**



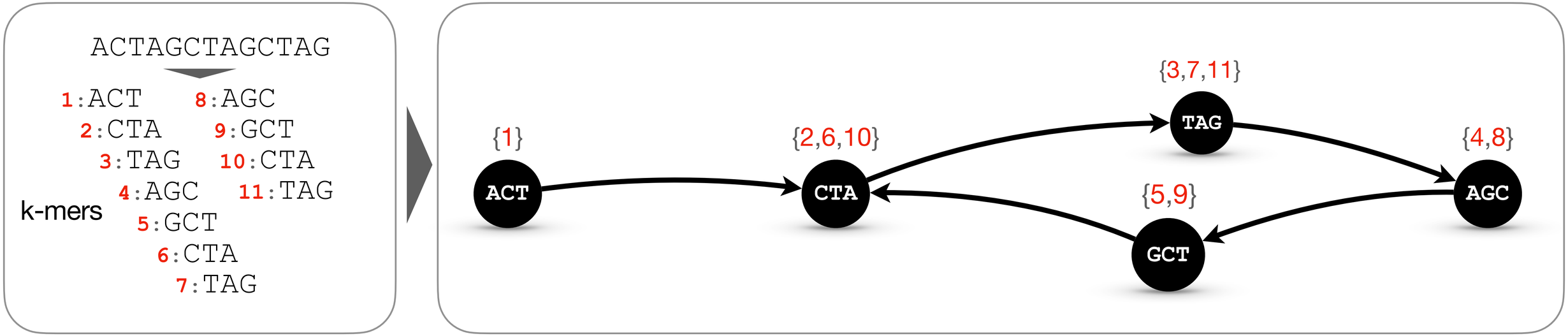
Invertible 🙂

de Bruijn graph

with k-mer coordinates

Motivation

2. Representing **k-mer coordinates**



Encoding ***sequence traces*** allows

- reconstructing indexed sequences

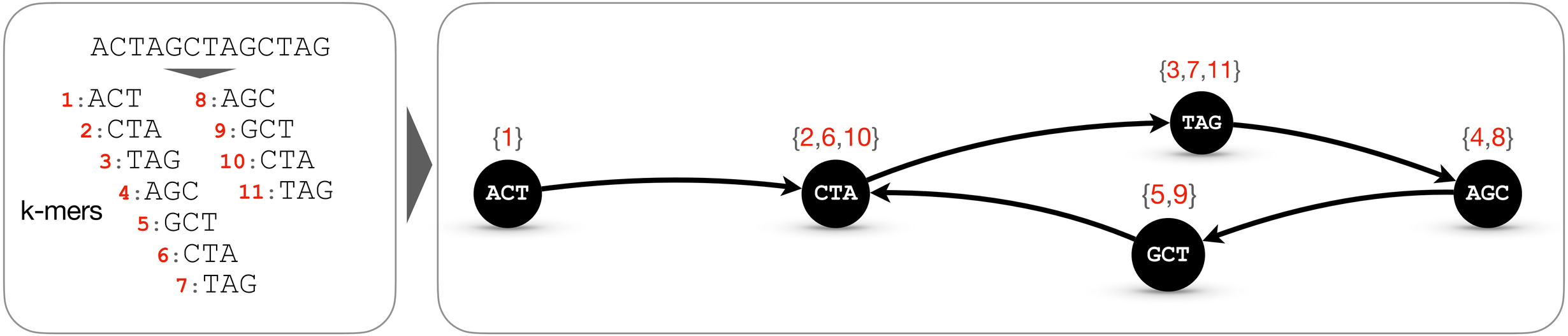
  (hence, **lossless sequence representation)**

de Bruijn graph

with k-mer coordinates

Motivation

2. Representing **k-mer coordinates**



Encoding ***sequence traces*** allows

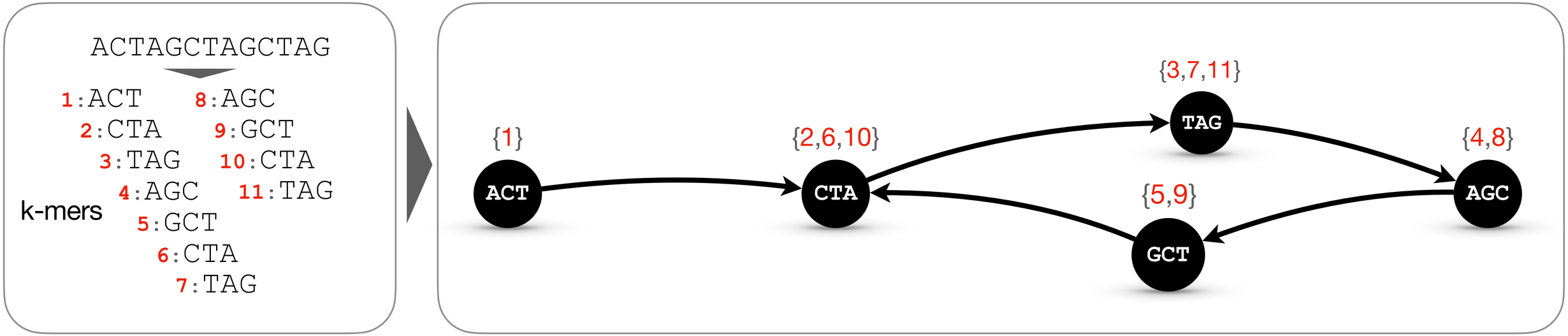
- reconstructing indexed sequences
  (hence, **lossless sequence representation)**
- performing exact sequence alignment

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
- retrieving all **traces** crossing a node

de Bruijn graph

with k-mer coordinates

Trace Consistent Graph (TCG-) Aligner

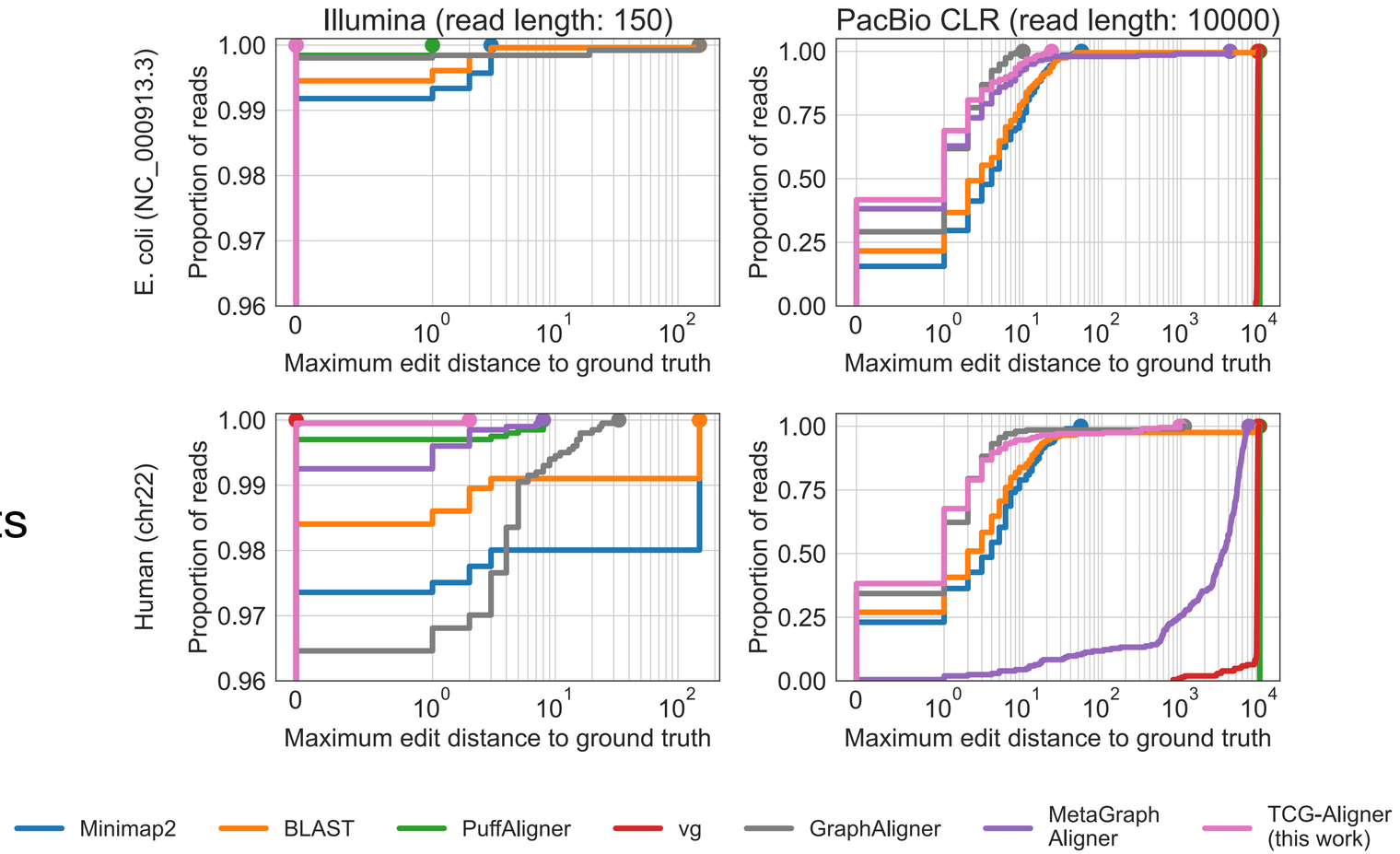
Sequence alignment on top of Counting DBG with k-mer coordinates with the **seed-chain-extend** approach:

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, Mustafa et al., 2020]

Alignment accuracy

Evaluation approach

1. Index a reference genome
2. Simulate reads from the same reference
3. Align reads and measure the 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.

Sequence alignment in MetaGraph

Pangenomes | Low-coverage assemblies | High-coverage assemblies | Reference genomes

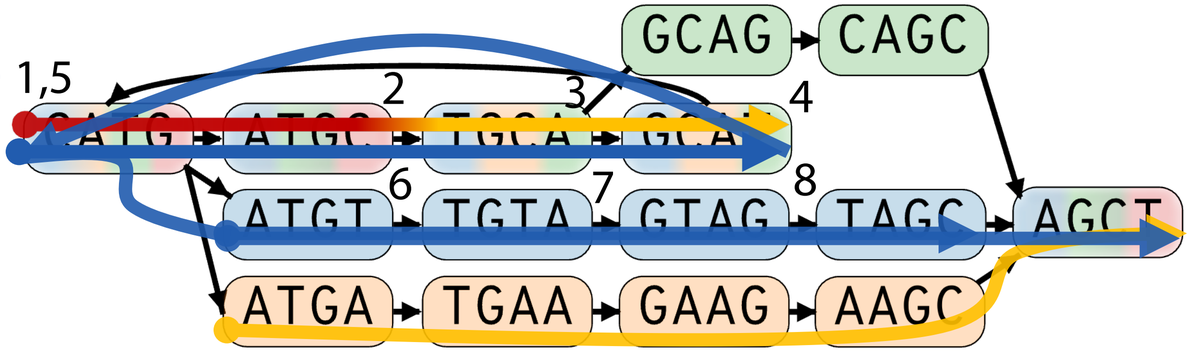
← Least restricted/biggest search space — Most restricted/smallest search space →

Label-free walks
All walks are valid

Label-consistent walks
All nodes must share a common label

Trace-consistent walks (i.e., "BLASTing")
Must be a subsequence of one of the inputs

Multi-label walks
May change the common label mid-walk



Why do we need multi-label alignment for low-coverage joint assembly graphs?

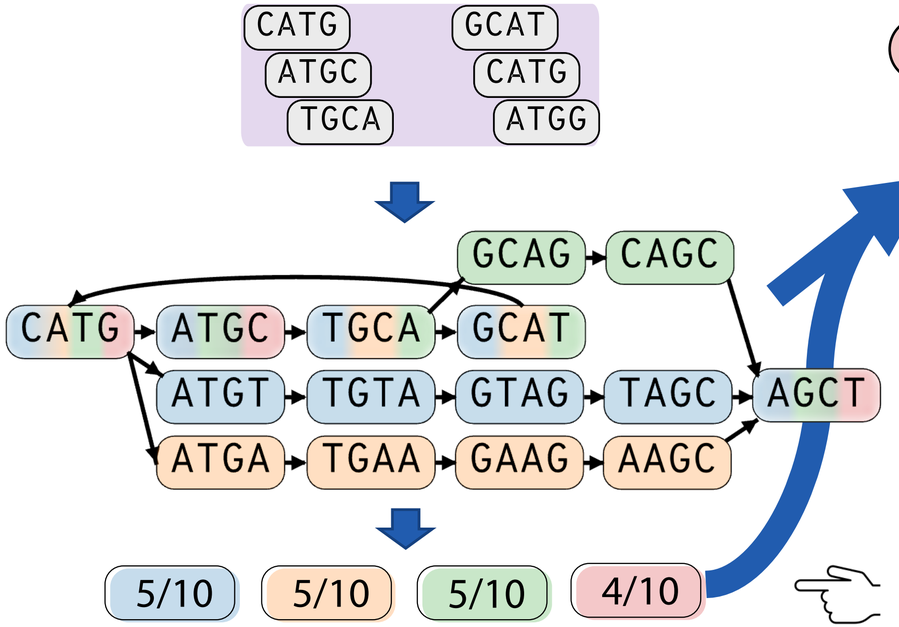
Slide by Harun Mustafa

Mixed label alignment in MetaGraph

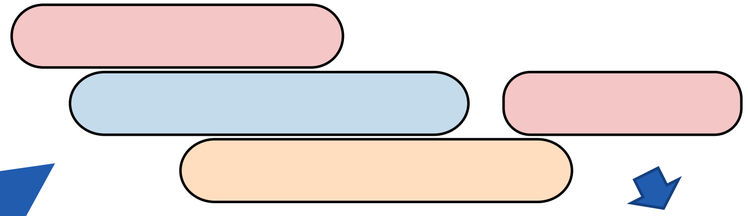
Task: Given a query sequence, find a graph walk that best matches the query

CATGCATGGCGCT

1. Find seeds in the graph



2. Search graph to extend seeds to alignments



"If you only need fast exact *k*-mer matching, stop here"

3. Chain to form a Multi-label alignment

Each alignment step can use one of many **different modules** depending on the use case

Slide by Harun Mustafa

Takeaways

Big Data
National Research Programme

MetaGraph is a tool for **indexing**, **search** and **assembly** of sequences

Takeaways

Big Data National Research Programme

MetaGraph is a tool for **indexing**, **search** and **assembly** of sequences

- Allows **efficiently representing**
  - k-mer presence/absence
  - k-mer abundances
  - k-mer positions

Takeaways

Big Data National Research Programme

MetaGraph is a tool for **indexing**, **search** and **assembly** of sequences

- Allows **efficiently representing**
  - k-mer presence/absence
  - k-mer abundances
  - k-mer positions
- **Extremely scalable**
  - efficient construction algorithms with limited memory

Takeaways

Big Data National Research Programme

MetaGraph is a tool for **indexing**, **search** and **assembly** of sequences

- Allows **efficiently representing**
  - k-mer presence/absence
  - k-mer abundances
  - k-mer positions
- **Extremely scalable**
  - efficient construction algorithms with limited memory
- Supports inexact search with **alignment**
  - supports sub-*k* seeding

Takeaways

Big Data National Research Programme

MetaGraph is a tool for **indexing**, **search** and **assembly** of sequences

- Allows **efficiently representing**
  - k-mer presence/absence
  - k-mer abundances
  - k-mer positions
- **Extremely scalable**
  - efficient construction algorithms with limited memory
- Supports inexact search with **alignment**
  - supports sub-*k* seeding
  - different constraints (sequences / labels / mixed-labels / none)

Takeaways

Big Data National Research Programme

MetaGraph is a tool for **indexing**, **search** and **assembly** of sequences

- Allows **efficiently representing**
  - k-mer presence/absence
  - k-mer abundances
  - k-mer positions
- **Extremely scalable**
  - efficient construction algorithms with limited memory
- Supports inexact search with **alignment**
  - supports sub-*k* seeding
  - different constraints (sequences / labels / mixed-labels / none)
- A number of **pre-constructed indexes** are available at https://metagraph.ethz.ch

Team

@m_karasikov


Mikhail Karasikov


Marc Zimmermann


Andre Kahles


Harun Mustafa


Daniel Danciu


Gunnar Rätsch

NRP75 Team


Torsten Hoefler


Mario Stanke


Matthis Ebel


Giovanna Migliorelli

Further contributors

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

75 NRP Big Data National Research Programme

Swiss National Science Foundation

ETH zürich