Long read splice alignment — theory and practice

Kristoffer Sahlin

Department of Mathematics, Science for Life Laboratory,
Stockholm University

Intended Learning Outcomes

Should

- Learn the basics of an aligner (concepts: seeding, chaining, extension)
- Be able to run minimap2 to obtain alignments
- Think critically about reliability of alignments for downstream analysis
- ~~(Learn some methods for basic sanity checking)~~

Workshop overview

Theory

- Long-read alignment (seeding, chaining, extension) à la minimap2
- Long-read splice alignment à la minimap2

Parameters and heuristics

- Seed and window size (k, w) - uniqueness and speed
- Some minimap2 specific parameters
- Thresholds

Exercise

- Mapping transcripts to references with minimap2

More theory

- Aligner variants (uLTRA, deSALT)

Interpreting the output

- SAM format
- MAPQ score and secondary and supplementary alignments
- CIGAR format

~~**Troubleshooting**~~

- ~~samtools, BLAT, IGV~~

1. Seeding with *k*-mers

- A match between read and reference
- Types of seeds: ***k*-mers** (there are others, e.g., spaced *k*-mers, MEMs etc)
- A *k*-mer is a substring of length k

```
CACGACTCTGGTACCTAGACTCGATCGATCGTACTGT….
CACG
 ACGA
  CGAC
```

k-mers with k=4

1. Seeding with *k*-mers

- A match between read and reference
- Types of seeds: ***k*-mers** (there are others, e.g., spaced *k*-mers, MEMs etc)
- A *k*-mer is a substring of length k

ch1	CACGACTCTGGTACCTAGACTCGATCGATCGTACTGT….
ch2	GCACACTGGTACCTAGACACGATCGCCCGTTGTGTCA….
ch3	GCTGACTGTTACAACTATACTCGATCGCCCGTTGTCT….
…	

Index

k-mer		Positions
CACG	:	[(ch1,0), (ch17,1202), …]
AGAC	:	[(ch1,16), (ch2, 14), …]
TATA	:	[(ch2, 25), (ch13, 205), …]
GACT	:	…
…		

1. Seeding with *k*-mers

- A match between read and reference
- Types of seeds: ***k*-mers** (there are others, e.g., spaced *k*-mers, MEMs etc)
- A *k*-mer is a substring of length k

ch1 CACGACTCTGGTACCTAGACTCGATCGATCGTACTGT….

ch2 GCACACTGGTACCTAGACACGATCGCCCGTTGTGTCA….

ch3 GCTGACTGTTACAACTATACTCGATCGCCCGTTGTCT….

…

?

AGACCCGAT

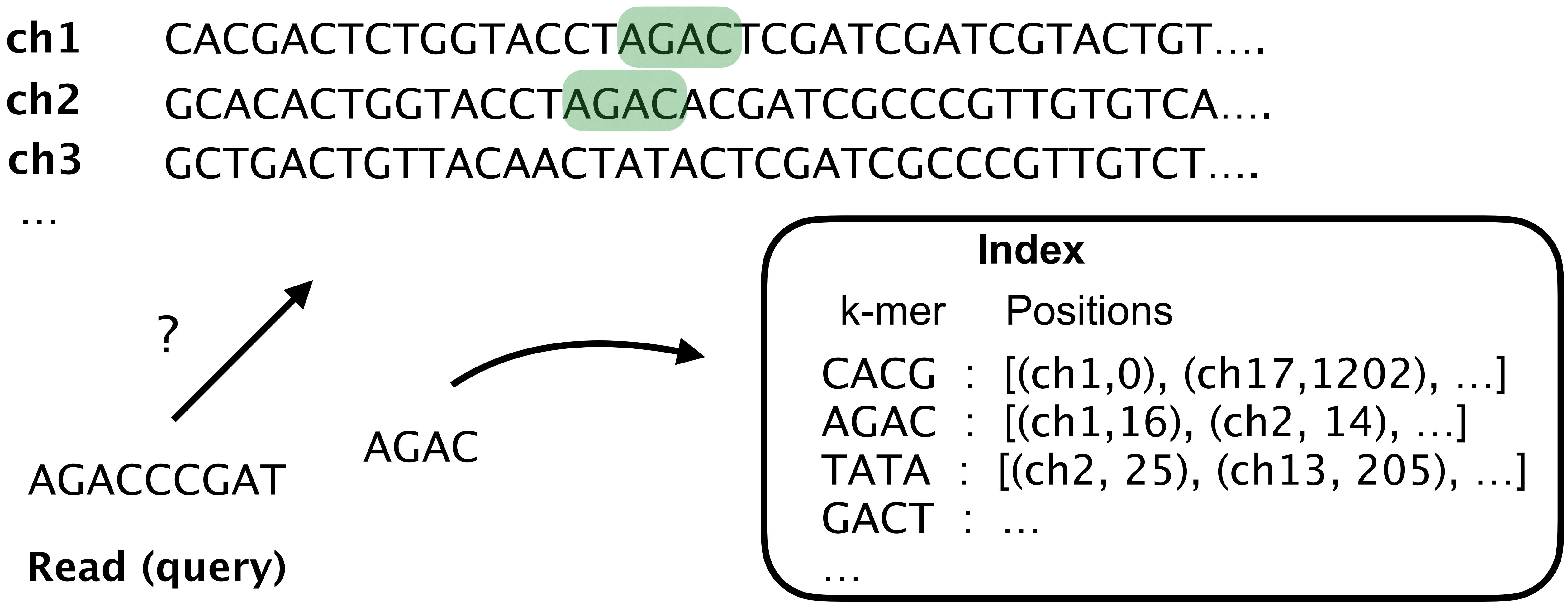
Read (query)

Index

k-mer		Positions
CACG	:	[(ch1,0), (ch17,1202), …]
AGAC	:	[(ch1,16), (ch2, 14), …]
TATA	:	[(ch2, 25), (ch13, 205), …]
GACT	:	…
…		

1. Seeding with *k*-mers

- A match between read and reference
- Types of seeds: ***k*-mers** (there are others, e.g., spaced *k*-mers, MEMs etc)
- A *k*-mer is a substring of length k



ch1 CACGACTCTGGTACCTAGACTCGATCGATCGTACTGT….

ch2 GCACACTGGTACCTAGACACGATCGCCCGTTGTGTCA….

ch3 GCTGACTGTTACAACTATACTCGATCGCCCGTTGTCT….

…

?

AGACCCGAT

Read (query)

AGAC

Index

k-mer		Positions
CACG	:	[(ch1,0), (ch17,1202), …]
AGAC	:	[(ch1,16), (ch2, 14), …]
TATA	:	[(ch2, 25), (ch13, 205), …]
GACT	:	…
…		

1. Seeding with *k*-mers

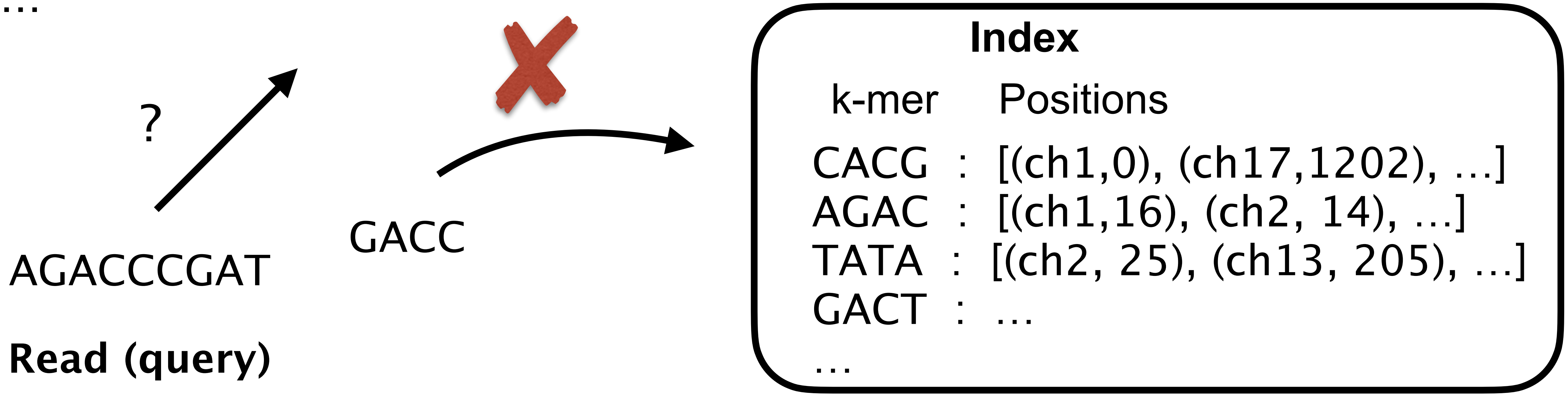
- A match between read and reference
- Types of seeds: ***k*-mers** (there are others, e.g., spaced *k*-mers, MEMs etc)
- A *k*-mer is a substring of length k

ch1 CACGACTCTGGTACCTAGACTCGATCGATCGTACTGT….

ch2 GCACACTGGTACCTAGACACGATCGCCCGTTGTGTCA….

ch3 GCTGACTGTTACAACTATACTCGATCGCCCGTTGTCT….

…

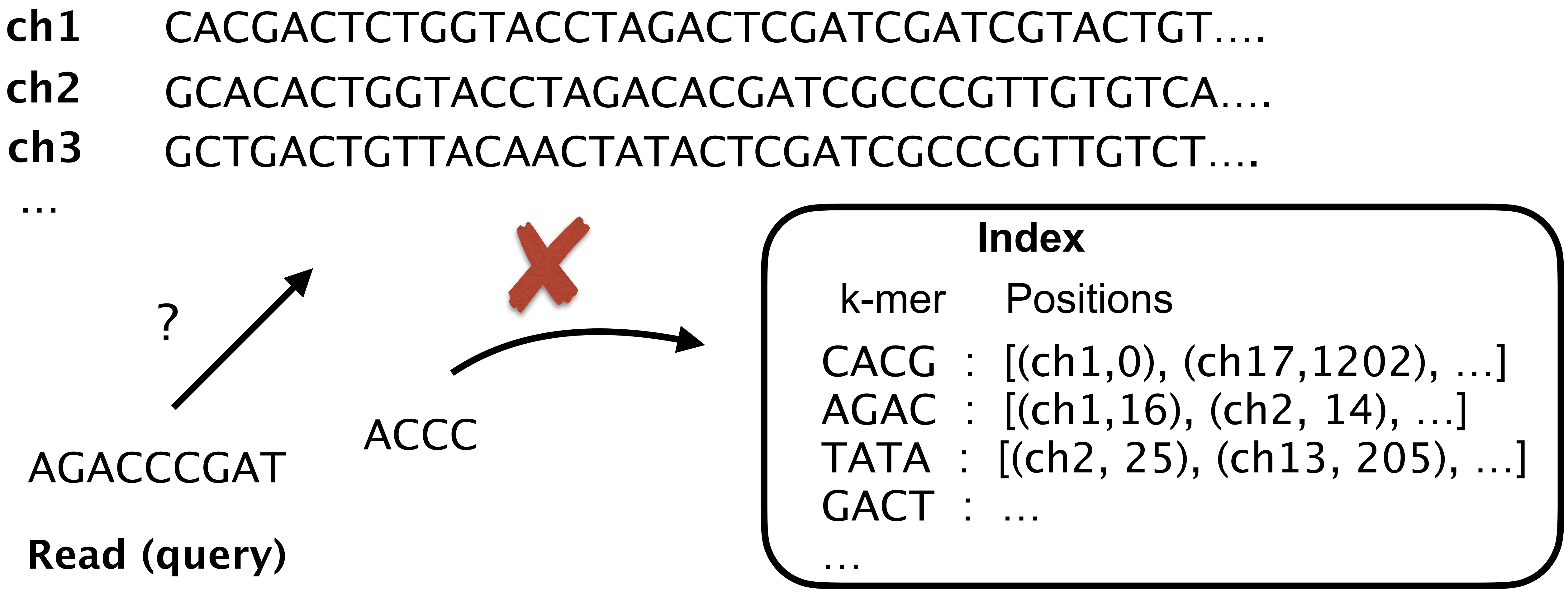


Index

k-mer		Positions
CACG	:	[(ch1,0), (ch17,1202), …]
AGAC	:	[(ch1,16), (ch2, 14), …]
TATA	:	[(ch2, 25), (ch13, 205), …]
GACT	:	…
…		

1. Seeding with *k*-mers

- A match between read and reference
- Types of seeds: ***k*-mers** (there are others, e.g., spaced *k*-mers, MEMs etc)
- A *k*-mer is a substring of length k



ch1 CACGACTCTGGTACCTAGACTCGATCGATCGTACTGT….
ch2 GCACACTGGTACCTAGACACGATCGCCCGTTGTGTCA….
ch3 GCTGACTGTTACAACTATACTCGATCGCCCGTTGTCT….
…

AGACCCGAT

Read (query)

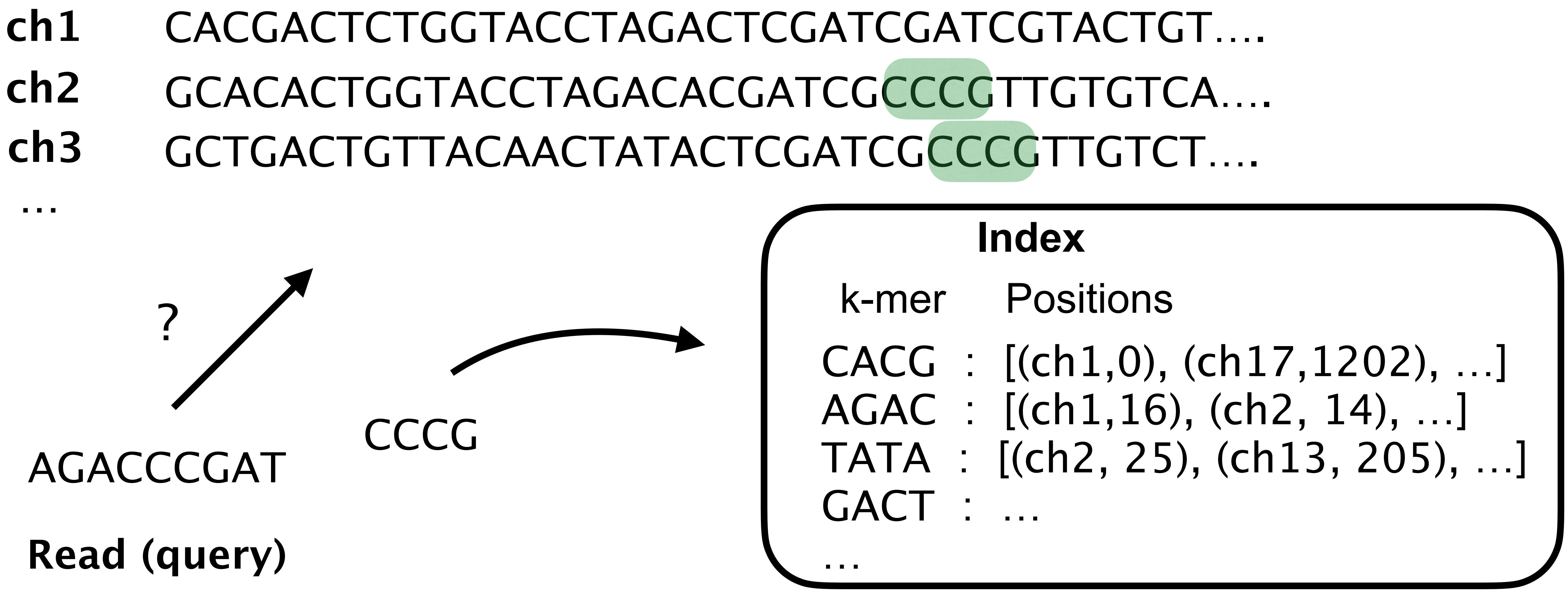
ACCC

Index

k-mer		Positions
CACG	:	[(ch1,0), (ch17,1202), …]
AGAC	:	[(ch1,16), (ch2, 14), …]
TATA	:	[(ch2, 25), (ch13, 205), …]
GACT	:	…
…		

1. Seeding with *k*-mers

- A match between read and reference
- Types of seeds: ***k*-mers** (there are others, e.g., spaced *k*-mers, MEMs etc)
- A *k*-mer is a substring of length k



ch1 CACGACTCTGGTACCTAGACTCGATCGATCGTACTGT….
ch2 GCACACTGGTACCTAGACACGATCGCCCGTTGTGTCA….
ch3 GCTGACTGTTACAACTATACTCGATCGCCCGTTGTCT….
…
?
AGACCCGAT
Read (query)
CCCG
Index
k-mer Positions
CACG : [(ch1,0), (ch17,1202), …]
AGAC : [(ch1,16), (ch2, 14), …]
TATA : [(ch2, 25), (ch13, 205), …]
GACT : …
…

1. Seeding with *k*-mers

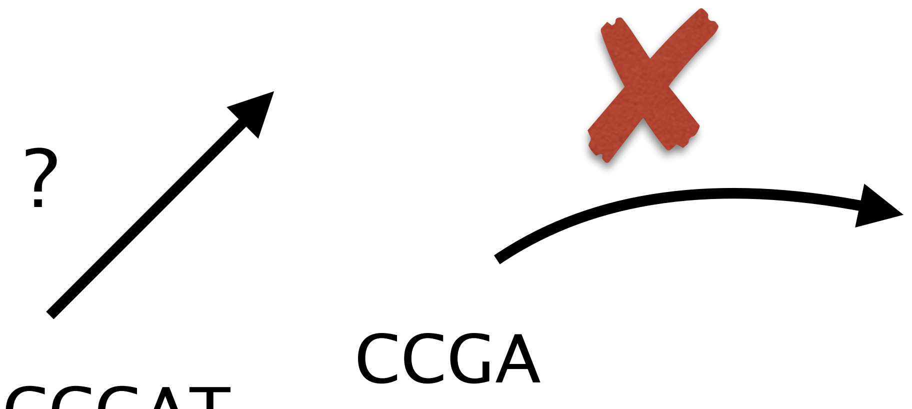
- A match between read and reference
- Types of seeds: ***k*-mers** (there are others, e.g., spaced *k*-mers, MEMs etc)
- A *k*-mer is a substring of length k

ch1 CACGACTCTGGTACCTAGACTCGATCGATCGTACTGT….

ch2 GCACACTGGTACCTAGACACGATCGCCCGTTGTGTCA….

ch3 GCTGACTGTTACAACTATACTCGATCGCCCGTTGTCT….

…



AGACCCGAT

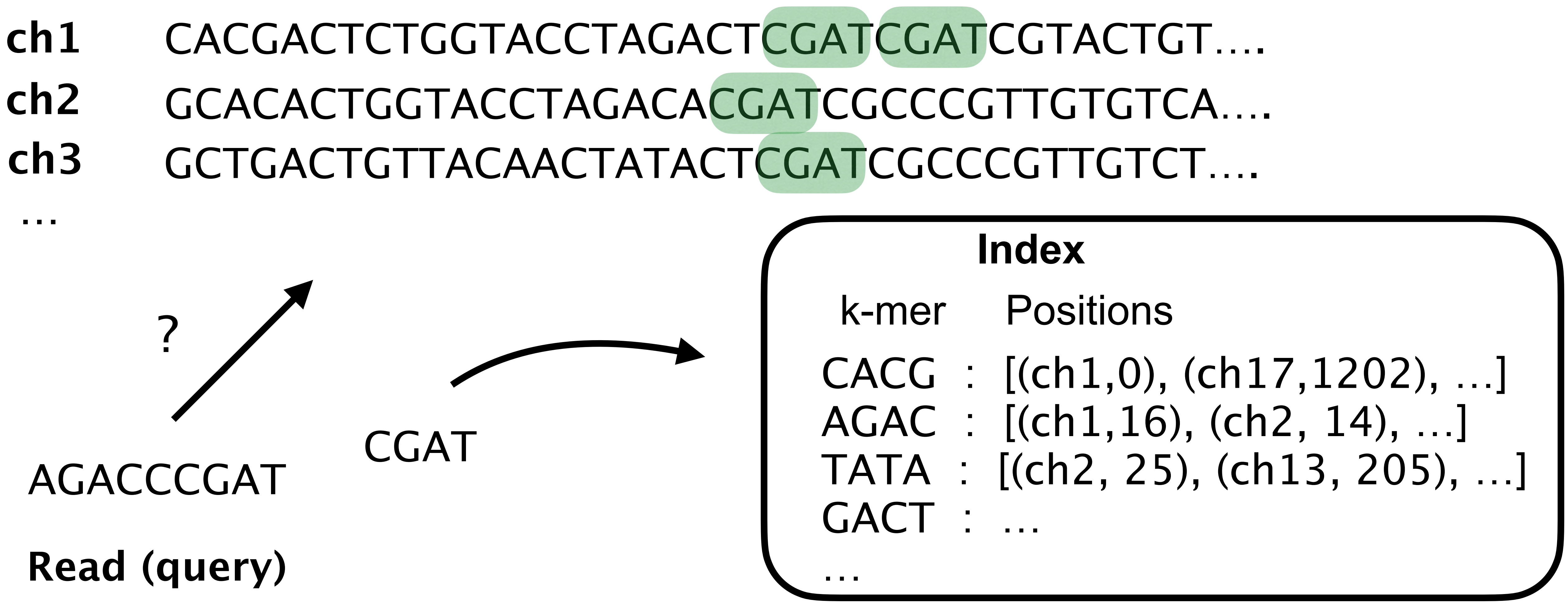
Read (query)

Index

k-mer		Positions
CACG	:	[(ch1,0), (ch17,1202), …]
AGAC	:	[(ch1,16), (ch2, 14), …]
TATA	:	[(ch2, 25), (ch13, 205), …]
GACT	:	…
…		

1. Seeding with *k*-mers

- A match between read and reference
- Types of seeds: ***k*-mers** (there are others, e.g., spaced *k*-mers, MEMs etc)
- A *k*-mer is a substring of length k



ch1 CACGACTCTGGTACCTAGACTCGATCGATCGTACTGT….

ch2 GCACACTGGTACCTAGACACGATCGCCCGTTGTGTCA….

ch3 GCTGACTGTTACAACTATACTCGATCGCCCGTTGTCT….

…

AGACCCGAT

Read (query)

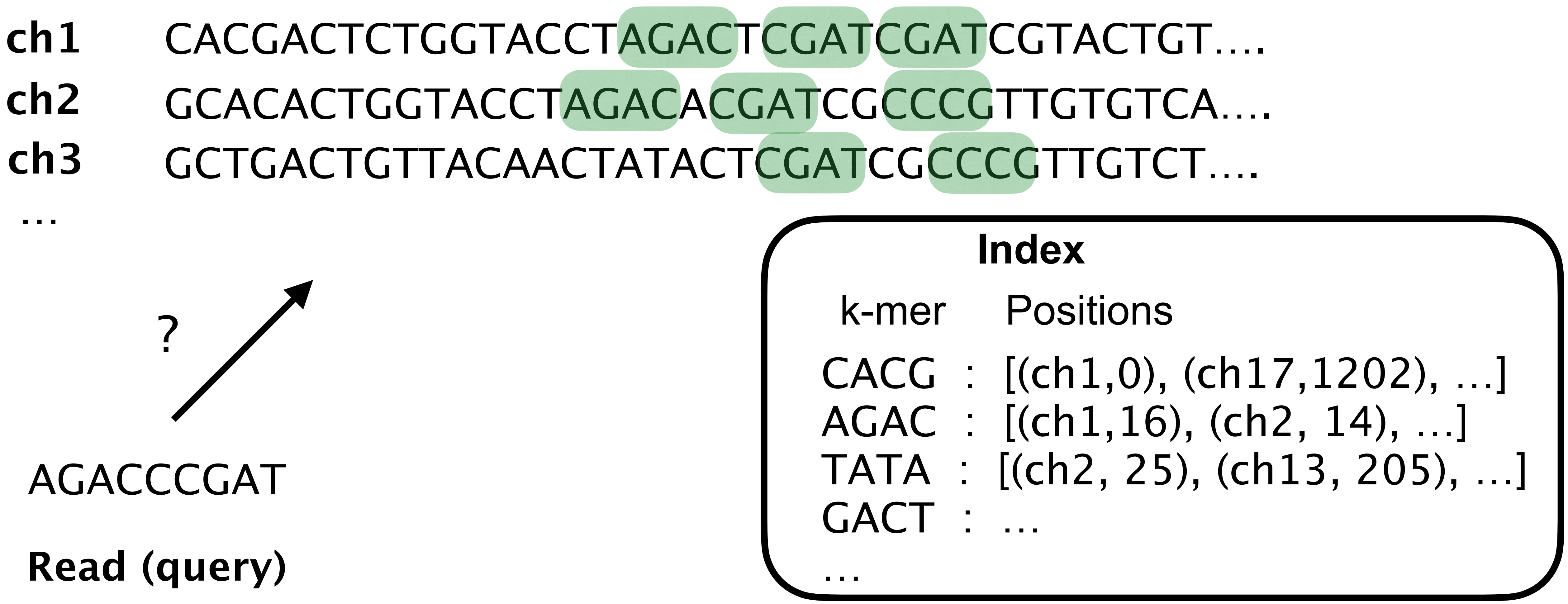
CGAT

Index

k-mer		Positions
CACG	:	[(ch1,0), (ch17,1202), …]
AGAC	:	[(ch1,16), (ch2, 14), …]
TATA	:	[(ch2, 25), (ch13, 205), …]
GACT	:	…
…		

1. Seeding with *k*-mers

- A match between read and reference
- Types of seeds: ***k*-mers** (there are others, e.g., spaced *k*-mers, MEMs etc)
- A *k*-mer is a substring of length k



ch1 CACGACTCTGGTACCTAGACTCGATCGATCGTACTGT….

ch2 GCACACTGGTACCTAGACACGATCGCCCGTTGTGTCA….

ch3 GCTGACTGTTACAACTATACTCGATCGCCCGTTGTCT….

…

?

AGACCCGAT

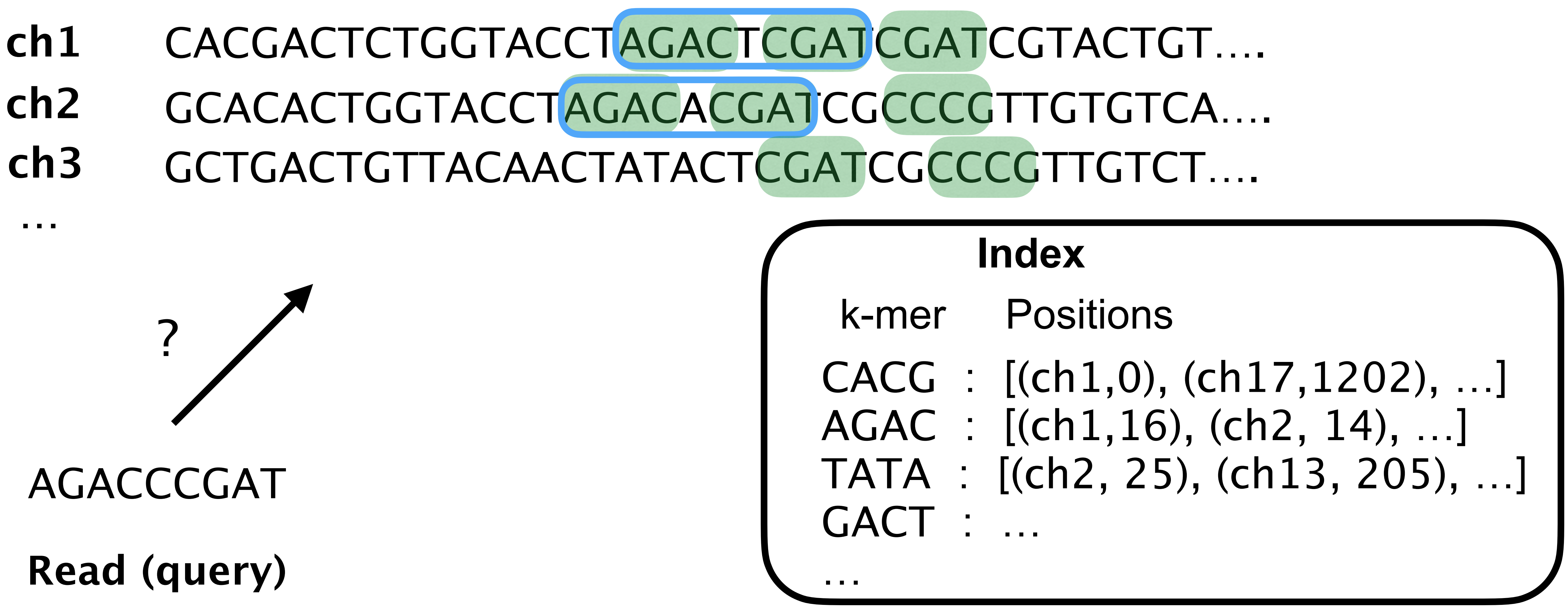
Read (query)

Index

k-mer		Positions
CACG	:	[(ch1,0), (ch17,1202), …]
AGAC	:	[(ch1,16), (ch2, 14), …]
TATA	:	[(ch2, 25), (ch13, 205), …]
GACT	:	…
…		

1. Seeding with *k*-mers

- A match between read and reference
- Types of seeds: ***k*-mers** (there are others, e.g., spaced *k*-mers, MEMs etc)
- A *k*-mer is a substring of length k



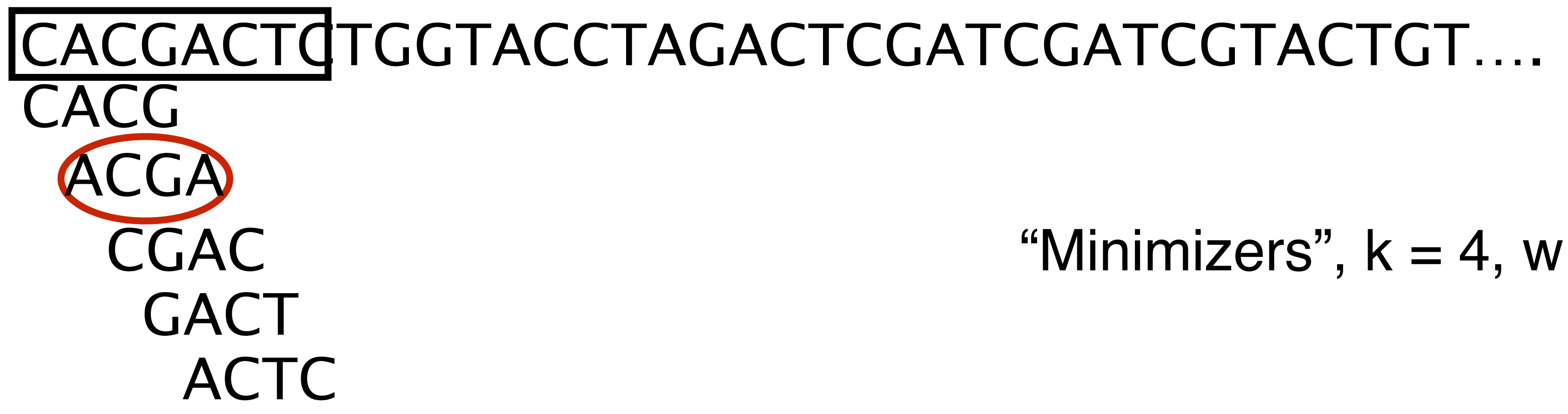
ch1 CACGACTCTGGTACCTAGACTCGATCGATCGTACTGT….
ch2 GCACACTGGTACCTAGACACGATCGCCCGTTGTGTCA….
ch3 GCTGACTGTTACAACTATACTCGATCGCCCGTTGTCT….
…
?
AGACCCGAT
Read (query)

Index

k-mer		Positions
CACG	:	[(ch1,0), (ch17,1202), …]
AGAC	:	[(ch1,16), (ch2, 14), …]
TATA	:	[(ch2, 25), (ch13, 205), …]
GACT	:	…
…		

1. Subsampling seeds

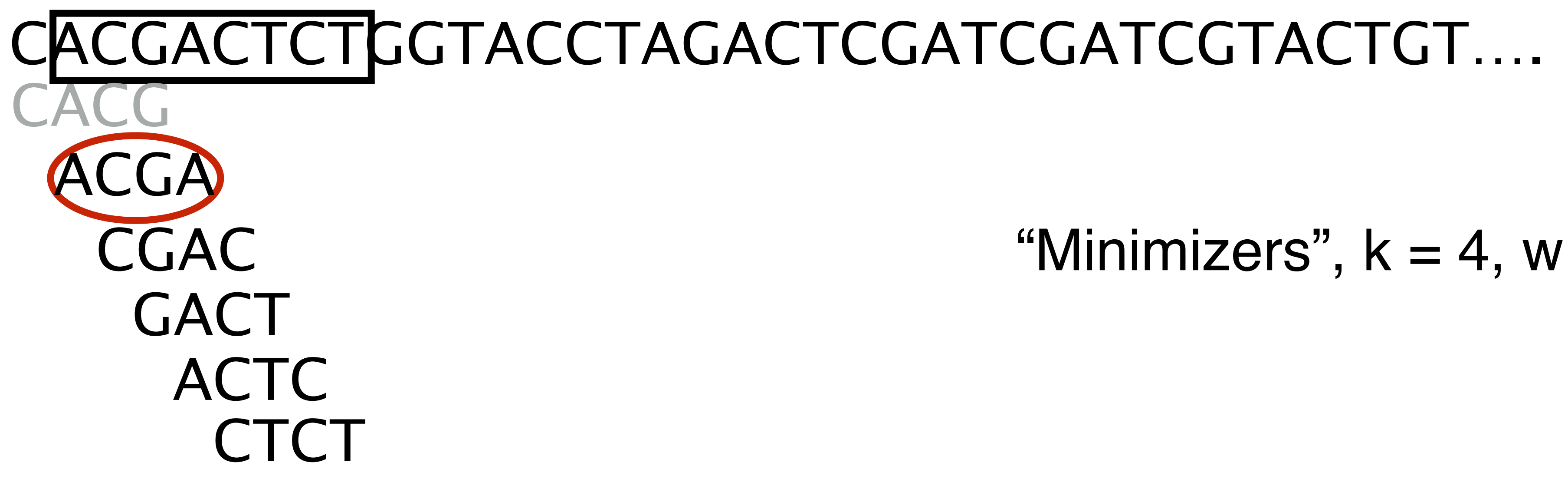
- Store only a subset of k-mers (aka *sketching* or *thinning*)
- At least one seed in every “window” - guarantee
- Parameter w (window size) controls density



“Minimizers”, k = 4, w = 5

1. Subsampling seeds

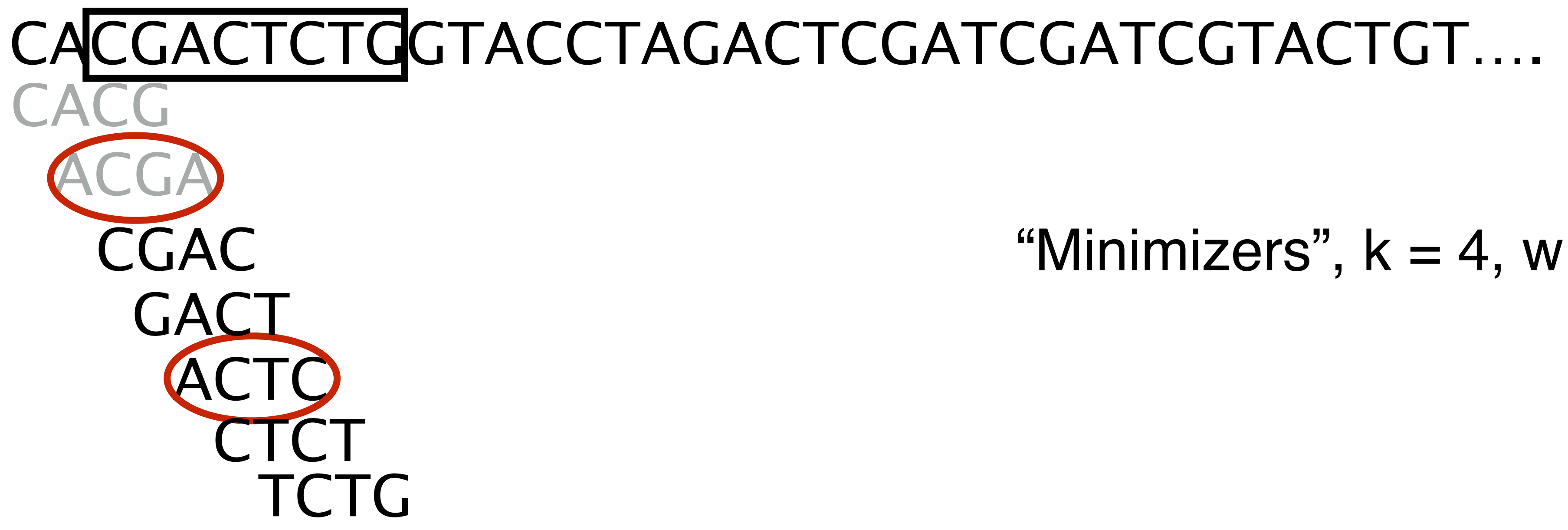
- Store only a subset of k-mers (aka *sketching* or *thinning*)
- At least one seed in every "window" - guarantee
- Parameter w (window size) controls density



"Minimizers", k = 4, w = 5

1. Subsampling seeds

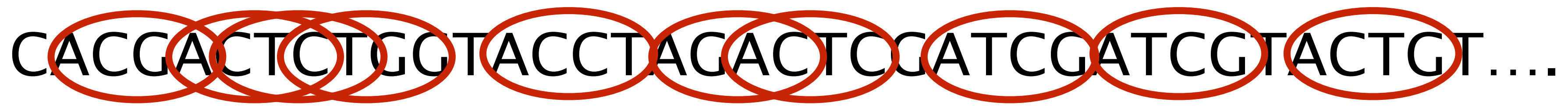
- Store only a subset of k-mers (aka *sketching* or *thinning*)
- At least one seed in every “window” - guarantee
- Parameter w (window size) controls density



“Minimizers”, k = 4, w = 5

1. Subsampling seeds

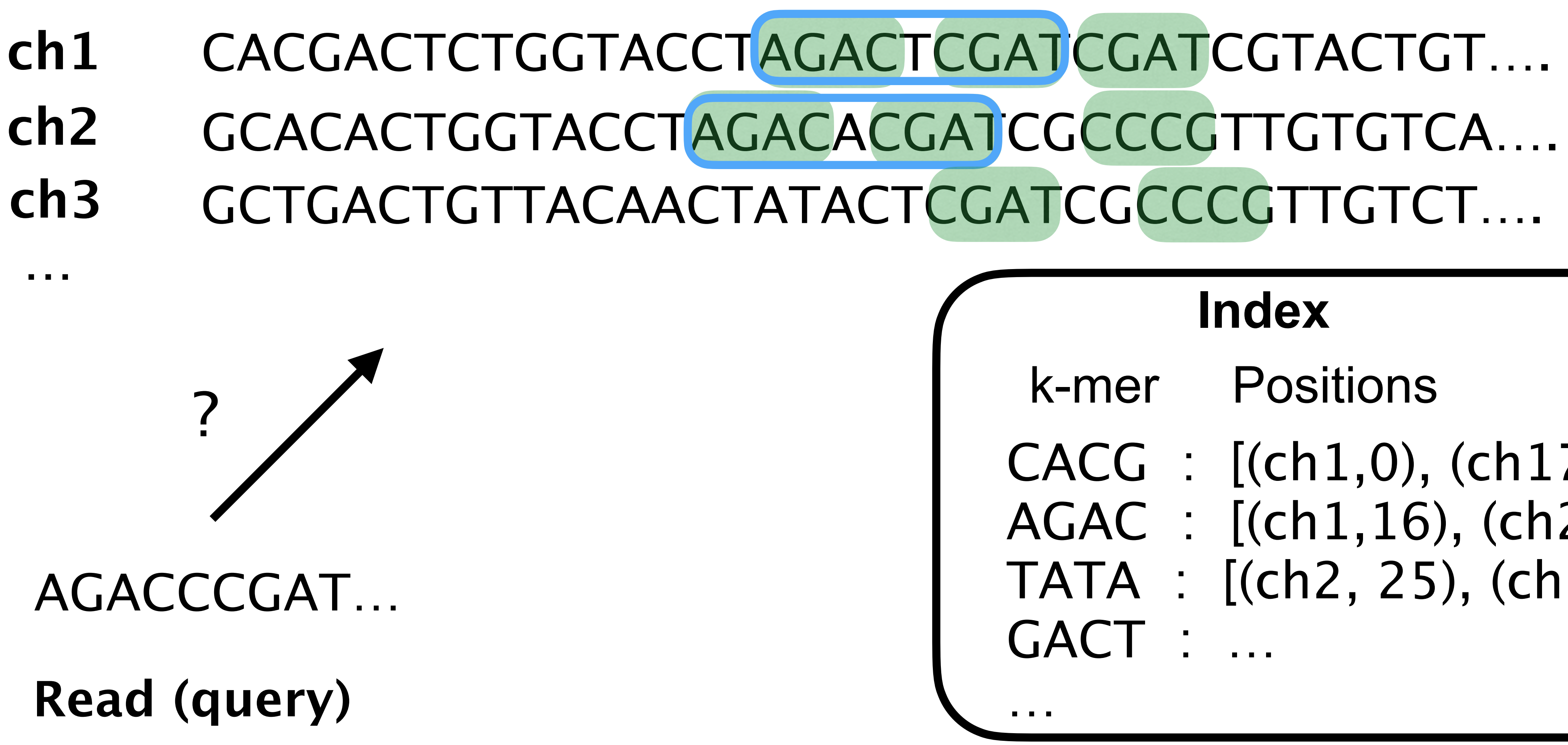
- Store only a subset of k-mers (aka *sketching* or *thinning*)
- At least one seed in every "window" - guarantee
- Parameter w (window size) controls density



"Minimizers", k = 4, w = 5

1. Subsampling seeds

- Store only a subset of k-mers (aka *sketching* or *thinning*)
- At least one seed in every "window" - guarantee
- Parameter w (window size) controls density

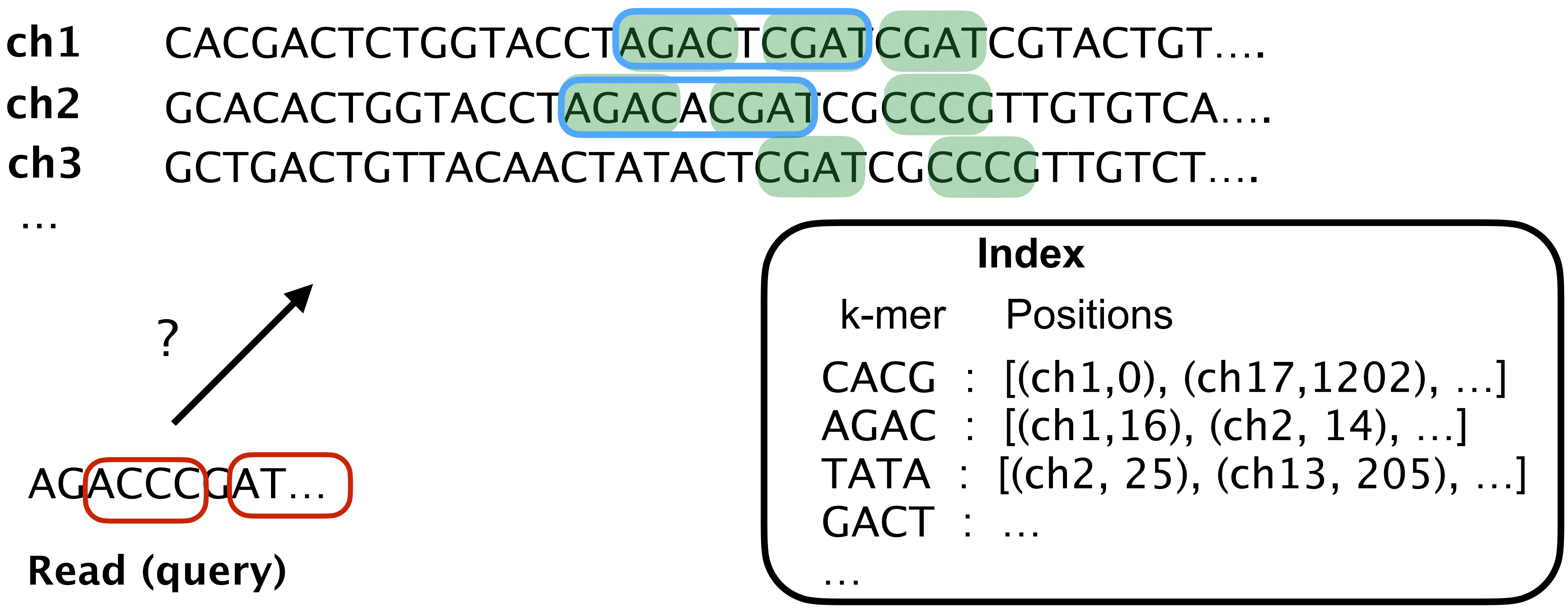

ch1 CACGACTCTGGTACCTAGACTCGATCGATCGTACTGT….
ch2 GCACACTGGTACCTAGACACGATCGCCCGTTGTGTCA….
ch3 GCTGACTGTTACAACTATACTCGATCGCCCGTTGTCT….
…
?
AGACCCGAT…
Read (query)

Index

k-mer		Positions
CACG	:	[(ch1,0), (ch17,1202), …]
AGAC	:	[(ch1,16), (ch2, 14), …]
TATA	:	[(ch2, 25), (ch13, 205), …]
GACT	:	…
…		

1. Subsampling seeds

- Store only a subset of k-mers (aka *sketching* or *thinning*)
- At least one seed in every "window" - guarantee
- Parameter w (window size) controls density

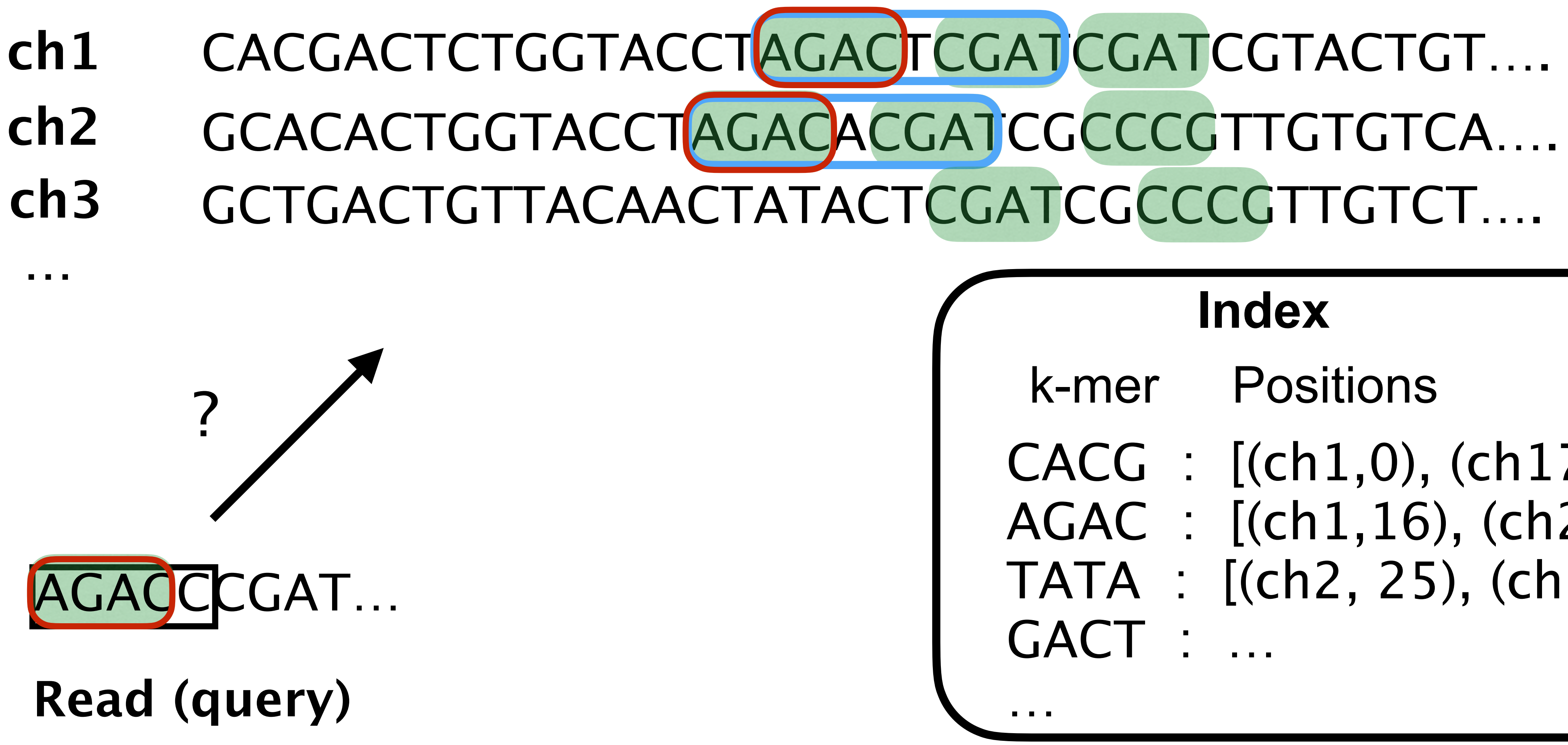


ch1 CACGACTCTGGTACCTAGACTCGATCGATCGTACTGT….
ch2 GCACACTGGTACCTAGACACGATCGCCCGTTGTGTCA….
ch3 GCTGACTGTTACAACTATACTCGATCGCCCGTTGTCT….
…
?
AGACCCGAT…
Read (query)
Index
k-mer
Positions
CACG : [(ch1,0), (ch17,1202), …]
AGAC : [(ch1,16), (ch2, 14), …]
TATA : [(ch2, 25), (ch13, 205), …]
GACT : …
…

Only minimizers in read are queried! -> No hits!

1. Subsampling seeds

- Store only a subset of k-mers (aka *sketching* or *thinning*)
- At least one seed in every "window" - guarantee
- Parameter w (window size) controls density

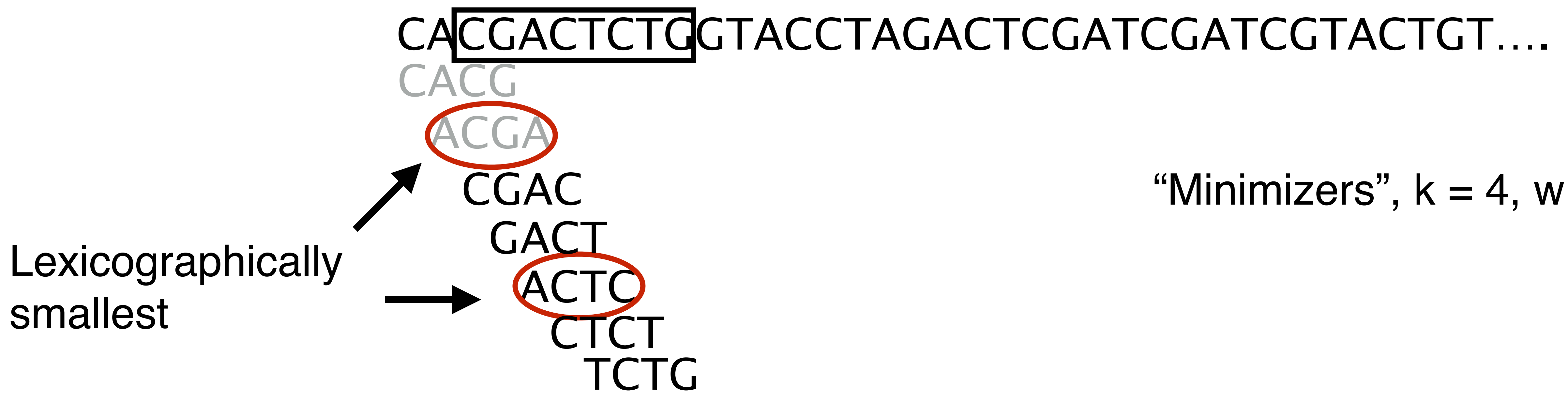

ch1 CACGACTCTGGTACCTAGACTCGATCGATCGTACTGT….
ch2 GCACACTGGTACCTAGACACGATCGCCCGTTGTGTCA….
ch3 GCTGACTGTTACAACTATACTCGATCGCCCGTTGTCT….
…
?
AGACCCGAT…
Read (query)

Index

k-mer		Positions
CACG	:	[(ch1,0), (ch17,1202), …]
AGAC	:	[(ch1,16), (ch2, 14), …]
TATA	:	[(ch2, 25), (ch13, 205), …]
GACT	:	…
…		

Pick a smaller w to get more seeds

From now on: No more visually pleasant lexicographical order on minimizers!


CACGACTCTGGTACCTAGACTCGATCGATCGTACTGT….
CACG
ACGA
CGAC
GACT
ACTC
CTCT
TCTG
Lexicographically smallest

"Minimizers", k = 4, w = 5

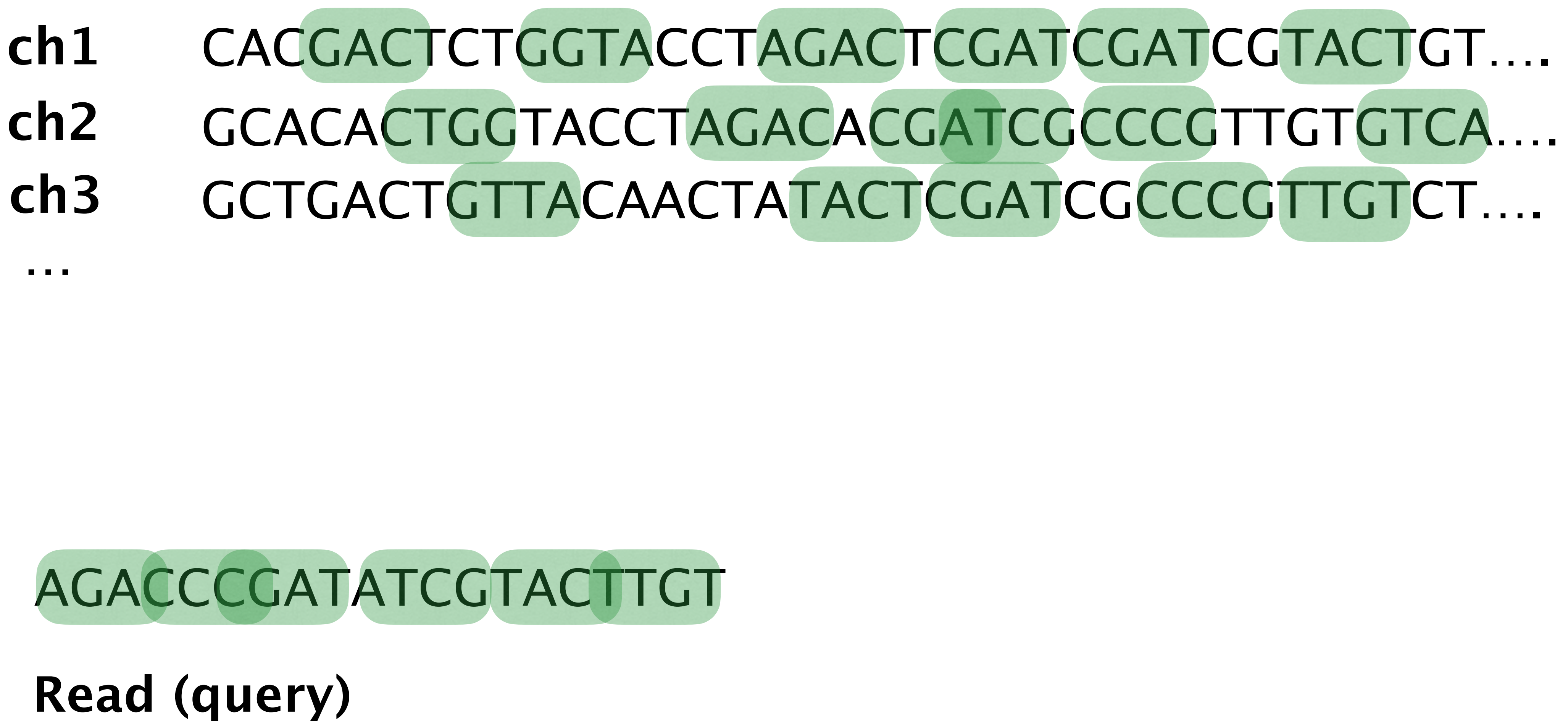
In practice: A *hash function* is used to scramble ordering -> sample k-mers more uniformly across the set of k-mers

2. Chaining (of seeds)

- Find collinear “chains” of seeds
- Collinearity: same order in read and ref
- (Additional: Not too differing in distance)

2. Chaining (of seeds)

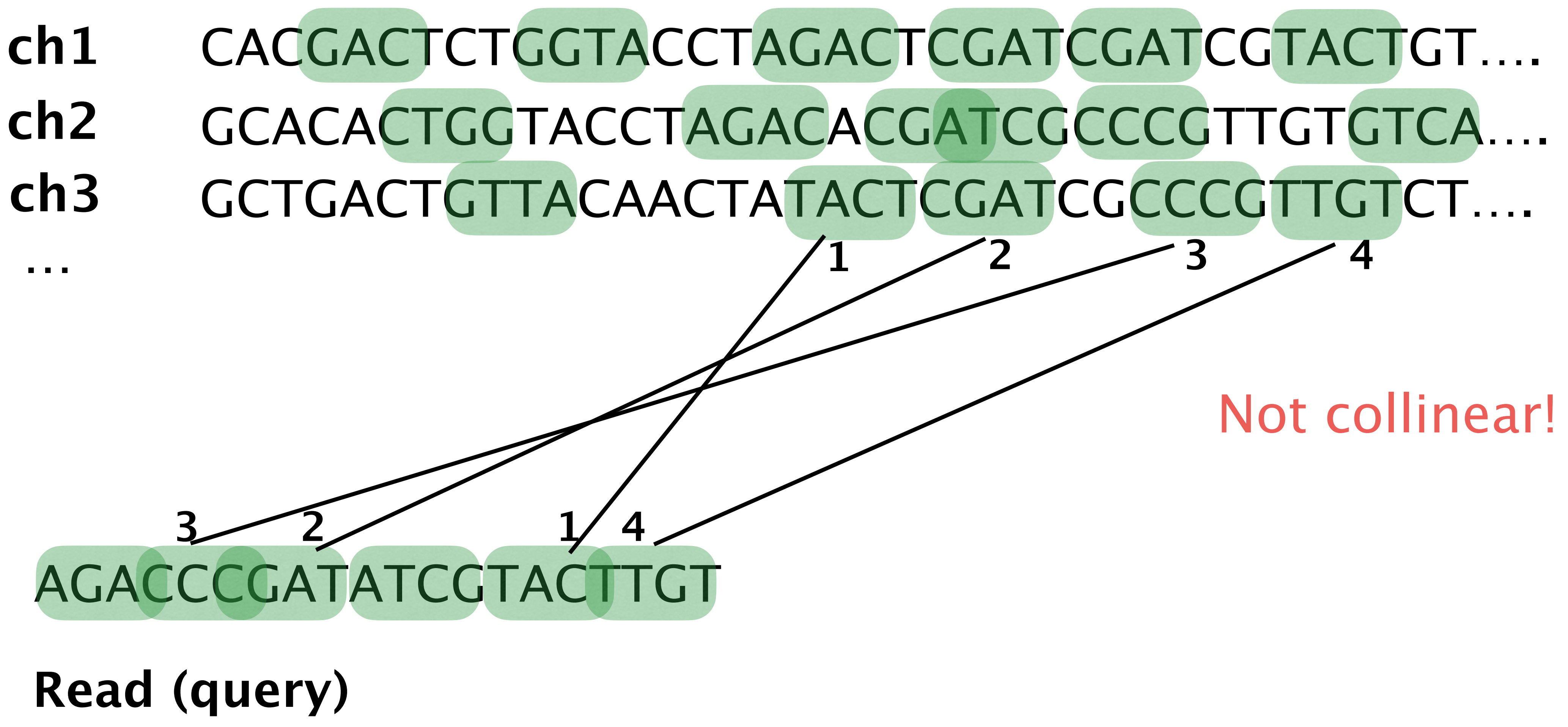
- Find collinear “chains” of seeds
- Collinearity: same order in read and ref
- (Additional: Not too differing in distance)



ch1 CACGACTCTGGTACCTAGACTCGATCGATCGTACTGT....
ch2 GCACACTGGTACCTAGACACGATCGCCCGTTGTGTCA....
ch3 GCTGACTGTTACAACTATACTCGATCGCCCGTTGTCT....
...
AGACCCGATATCGTACTTGT
Read (query)

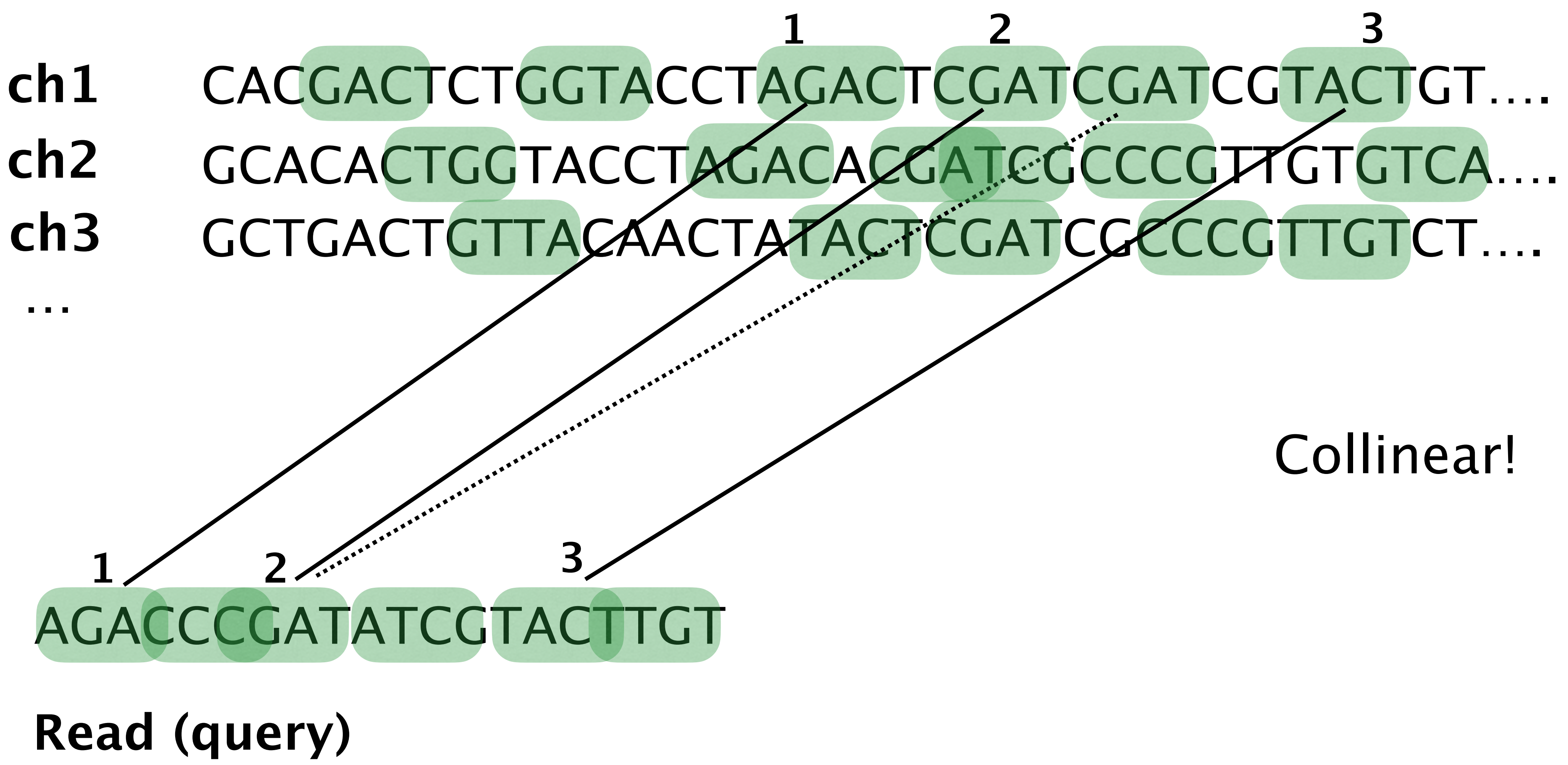
2. Chaining (of seeds)

- Find collinear “chains” of seeds
- Collinearity: same order in read and ref
- (Additional: Not too differing in distance)


ch1 CACGACTCTGGTACCTAGACTCGATCGATCGTACTGT….
ch2 GCACACTGGTACCTAGACACGATCGCCCGTTGTGTCA….
ch3 GCTGACTGTTACAACTATACTCGATCGCCCGTTGTCT….
…
1
2
3
4
Not collinear!
3
2
1
4
AGACCCGATATCGTACTTGT
Read (query)

2. Chaining (of seeds)

- Find collinear “chains” of seeds
- Collinearity: same order in read and ref
- (Additional: Not too differing in distance)

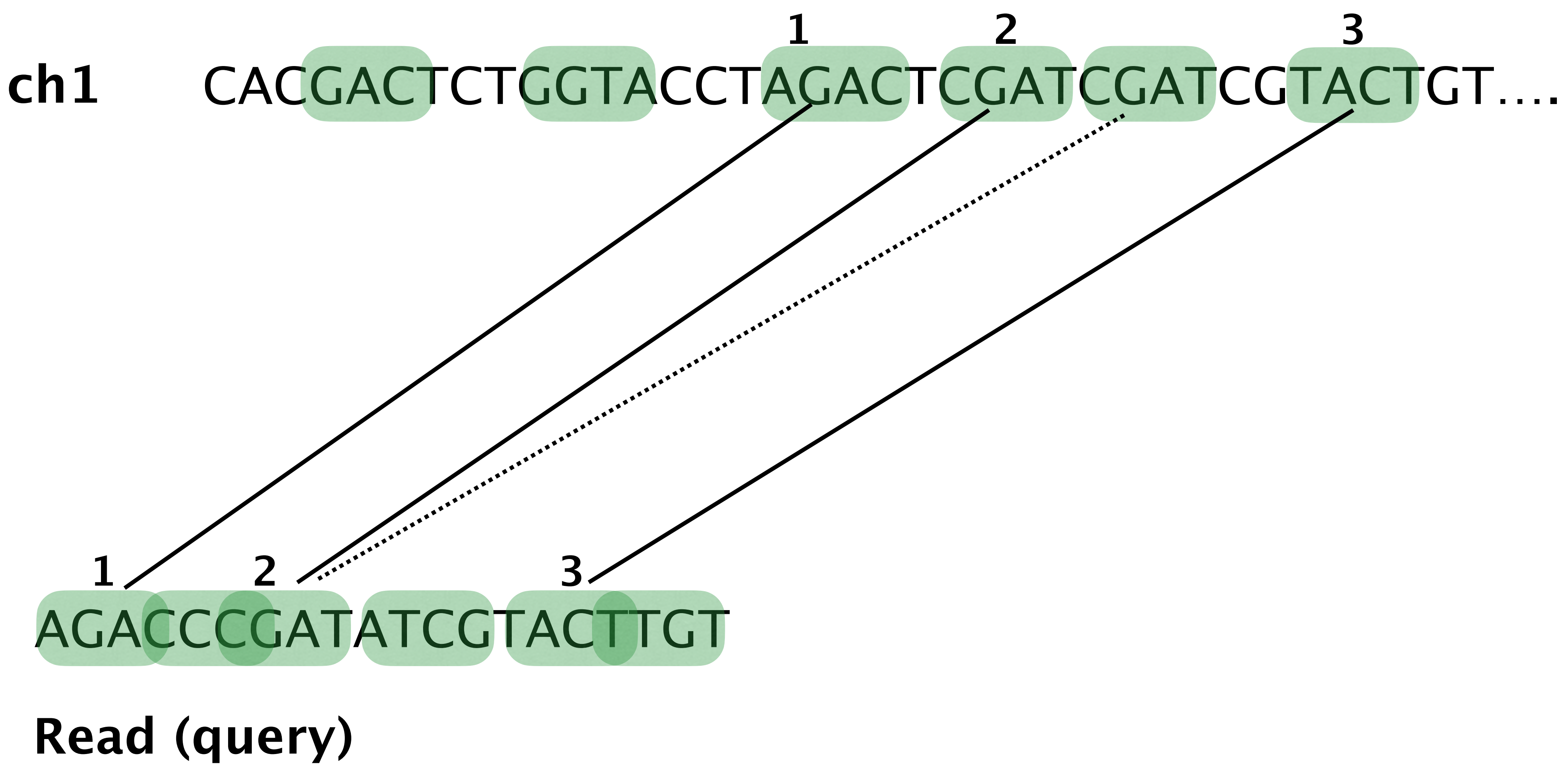

1
2
3
ch1 CACGACTCTGGTACCTAGACTCGATCGATCGTACTGT….
ch2 GCACACTGGTACCTAGACACGATCGCCCGTTGTGTCA….
ch3 GCTGACTGTTACAACTATACTCGATCGCCCGTTGTCT….
…
Collinear!
1
2
3
AGACCCGATATCGTACTTGT
Read (query)

3. Extension

- Grow out base-level pairwise *extension alignment* from seeds
- Get an alignment score

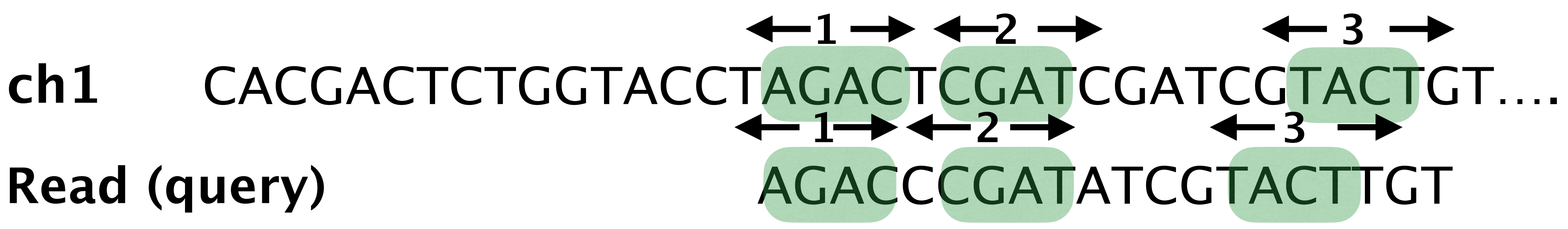
3. Extension

- Grow out base-level pairwise *extension alignment* from seeds
- Get an alignment score



3. Extension

- Grow out base-level pairwise *extension alignment* from seeds
- Get an alignment score



3. Extension

- Grow out base-level pairwise *extension alignment* from seeds
- Get an alignment score

```
ch1           CACGACTCTGGTACCTAGACTCGATCGATCGTACT-GT...
                              |||| ||||  |||||||| ||
Read (query)                  AGACCCGAT--ATCGTACTTGT
```

Alignment score (AS): $Am - Bx - \sum_i O \cdot Eg_i$

A: Match score (2)	*m*: #matches
B: Mismatch penalty (4)	*x*: #mismatches
O: Gap open penalty (4)	g_i: length of gap *i*
E: Gap extension penalty (2)	

3. Extension

- Grow out base-level pairwise *extension alignment* from seeds
- Get an alignment score

```
ch1           CACGACTCTGGTACCTAGACTCGATCGATCGTACT-GT...
                              |||| ||||  |||||||| ||
Read (query)                  AGACCCGAT--ATCGTACTTGT
```

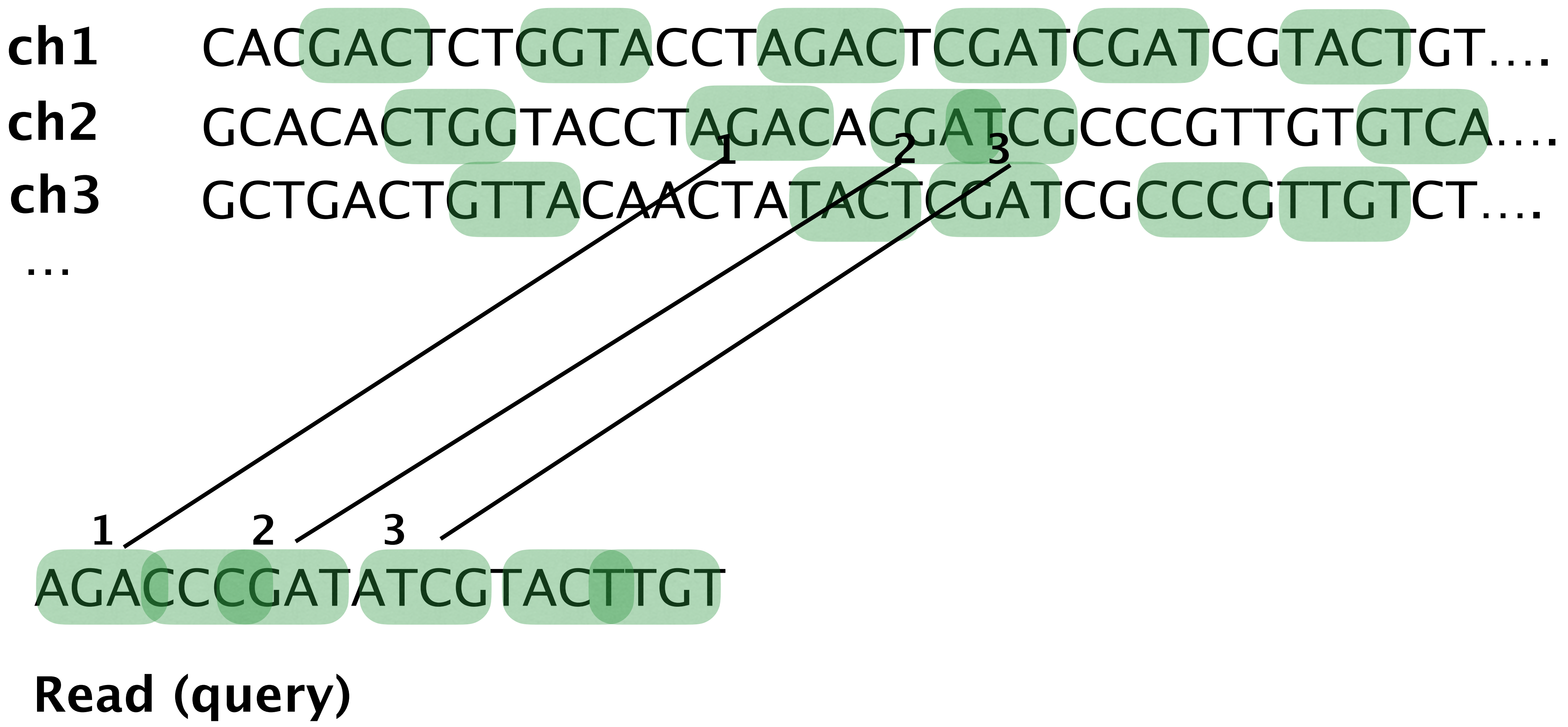
Alignment score (AS): $Am - Bx - \sum_i O \cdot Eg_i$

A: Match score (2)	*m*: #matches
B: Mismatch penalty (4)	*x*: #mismatches
O: Gap open penalty (4)	g_i: length of gap *i*
E: Gap extension penalty (2)	

$$AS = 2 \cdot 18 - 4 \cdot 1 - (4 + 2 \cdot 2) - (4 + 2 \cdot 1) = 18$$

3. Extension

- Grow out base-level pairwise *extension alignment* from seeds
- Get an alignment score



ch1 CACGACTCTGGTACCTAGACTCGATCGATCGTACTGT….
ch2 GCACACTGGTACCTAGACACGATCGCCCGTTGTGTCA….
ch3 GCTGACTGTTACAACTATACTCGATCGCCCGTTGTCT….
…
1
2
3
1
2
3
AGACCCGATATCGTACTTGT
Read (query)

3. Extension

- Grow out base-level pairwise *extension alignment* from seeds
- Get an alignment score

```
ch2            GCACACTGGTACCTAGACACGATCGCCCGTTGTGTCA....

Read (query)                 AGACCCGATATCGTACTTGT
```

3. Extension

- Grow out base-level pairwise *extension alignment* from seeds
- Get an alignment score

```
ch2   GCACACTGGTACCTAGACACGAT--CGCCCGTTGTGTCA....
                    |||| ||||  ||  | ||||
                    AGACCCGATATCGTAC-TTGT
```

Alignment score (AS): $Am - Bx - \sum_i O \cdot Eg_i$

A: Match score (2)
B: Mismatch penalty (4)
O: Gap open penalty (4)
E: Gap extension penalty (2)

m: #matches
x: #mismatches
g_i: length of gap *i*

$$AS = ?$$

Long-read splice alignment

- Still seed-chain-extend, but:
  - Several collinear chains (think one per exon)
  - ‘Local’ extension alignments around exon sites
  - Splice site specific extension penalty (canonical site?)
- For small introns (same local chain): gap or intron penalty

Long-read splice alignment

- Still seed-chain-extend, but:
  - Several collinear chains (think one per exon)
  - 'Local' extension alignments around exon sites
  - Splice site specific extension penalty (canonical site?)
- For small introns (same local chain): gap or intron penalty

Reference

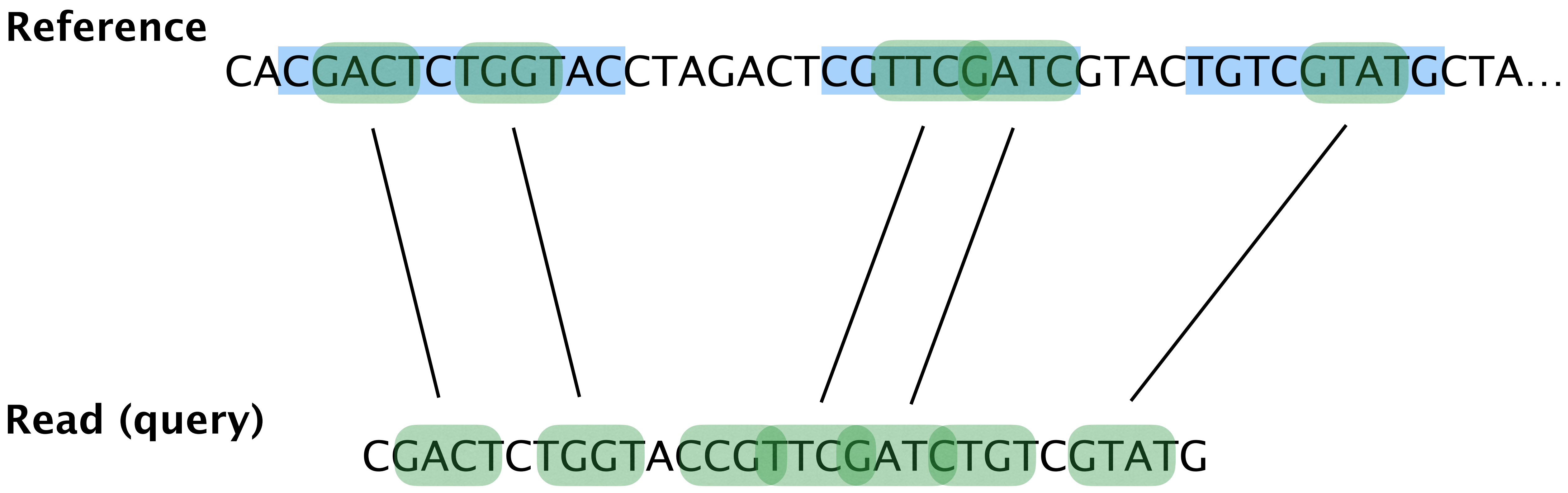
CACGACTCTGGTACCTAGACTCGATCGATCGTACTGTCGTATGCTA…

Read (query)

CGACTCTGGTACCGATCGATCTGTCGTATG

Long-read splice alignment

- Still seed-chain-extend, but:
  - Several collinear chains (think one per exon)
  - ‘Local’ extension alignments around exon sites
  - Splice site specific extension penalty (canonical site?)
- For small introns (same local chain): gap or intron penalty


Reference
CACGACTCTGGTACCTAGACTCGTTCGATCGTACTGTCGTATGCTA…
Read (query)
CGACTCTGGTACCGTTCGATCTGTCGTATG

Parameters (Minimap2)

Many parameters!
(and many 'hidden' parameters https://lh3.github.io/minimap2/minimap2.html)

- Seeding:
  - -k: size of seeds
  - -w: density of seeds
  - -f: fraction filtering repetitive of seeds
- Chaining:
  - -n Minimum seeds to include in a chain
  - -m minimum chaining score
- Alignment:
  - -a: extension alignment
  - Is the base level alignment off? Parameters -A -B -O -E

- For splice alignment:
  - The ensemble '-ax splice' flag typically take care of suitable parameter choices (canonical splice sites)
  - -u: how to find GT-AG
  - -G Maximum gap on the reference (long introns)
  - -C Cost for a non-canonical GT-AG splicing (Hidden)

Parameters (Minimap2)

Many parameters!
(and many ‘hidden’ parameters https://lh3.github.io/minimap2/minimap2.html)

Poor alignments?

- Error rate/sequence diversity
- Short exons
- Your reference characteristics (e.g., repetitiveness)
- Forgot ‘cleaning’ (trimming) reads before alignment
    - pychopper (ONT)
    - lima (PacBio)

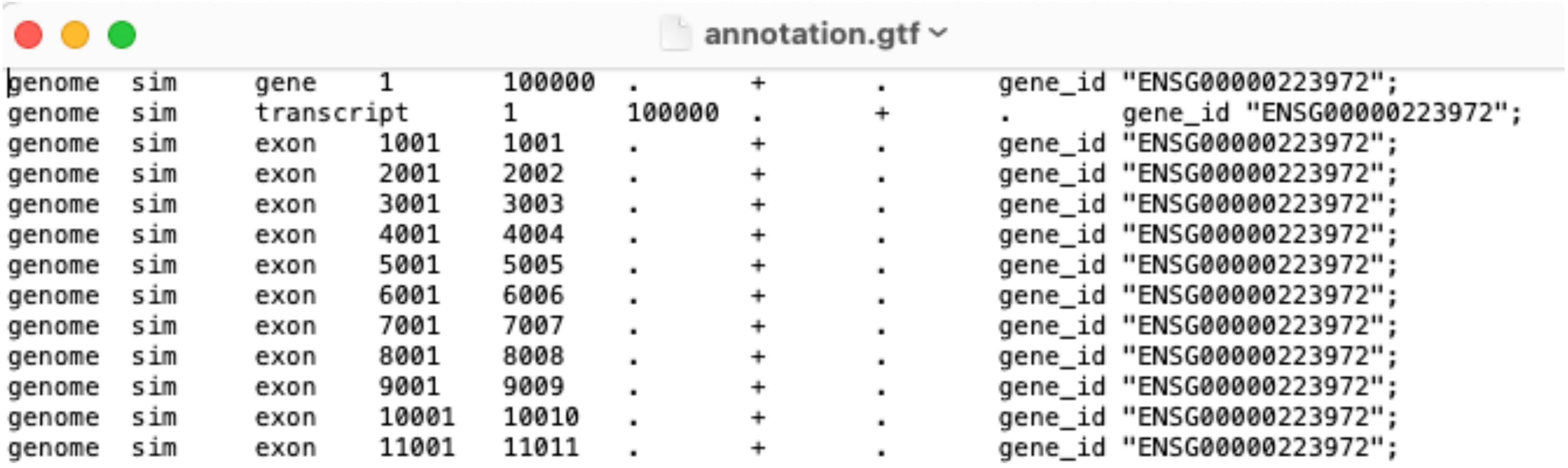
Challenge: Get as many exons correctly aligned as possible with minimap2!

Prerequisites

- Install minimap2
- Install Python
- Download data:
  - git clone https://github.com/ksahlin/misc.git
  - (or download 'evaluate_instance.py' and the the three files in dataset1 folder at https://github.com/ksahlin/misc/tree/main/LongTREC/examples)

Data

- 100kbp simulated 'genome' - nucleotides A,C,G,T *randomly simulated*
- Exons of size 1nt, 2nt, 3nt,… 99nt simulated.
- Challenge: Map an error-free transcript containing all exons


annotation.gtf

```
genome  sim     gene    1       100000  .       +       .       gene_id "ENSG00000223972";
genome  sim     transcript      1       100000  .       +       .       gene_id "ENSG00000223972";
genome  sim     exon    1001    1001    .       +       .       gene_id "ENSG00000223972";
genome  sim     exon    2001    2002    .       +       .       gene_id "ENSG00000223972";
genome  sim     exon    3001    3003    .       +       .       gene_id "ENSG00000223972";
genome  sim     exon    4001    4004    .       +       .       gene_id "ENSG00000223972";
genome  sim     exon    5001    5005    .       +       .       gene_id "ENSG00000223972";
genome  sim     exon    6001    6006    .       +       .       gene_id "ENSG00000223972";
genome  sim     exon    7001    7007    .       +       .       gene_id "ENSG00000223972";
genome  sim     exon    8001    8008    .       +       .       gene_id "ENSG00000223972";
genome  sim     exon    9001    9009    .       +       .       gene_id "ENSG00000223972";
genome  sim     exon    10001   10010   .       +       .       gene_id "ENSG00000223972";
genome  sim     exon    11001   11011   .       +       .       gene_id "ENSG00000223972";
```

Challenge: Get as many exons correctly aligned as possible with minimap2!

SETUP

1. git clone https://github.com/ksahlin/misc.git
2. cd misc/LongTREC/examples/
3. minimap2 --eqx -a **[PARAMS]** dataset1/genome.fa dataset1/query.fa > mm2.sam
4. *python evaluate_instance.py --gtf dataset1/annotation.gtf --samfile mm2.sam*

Challenge: Get as many exons correctly aligned as possible with minimap2!

SETUP

1. git clone https://github.com/ksahlin/misc.git
2. cd misc/LongTREC/examples/
3. minimap2 --eqx -a **[PARAMS]** dataset1/genome.fa dataset1/query.fa > mm2.sam
4. *python evaluate_instance.py --gtf dataset1/annotation.gtf --samfile mm2.sam*

Try steps 3 and 4 on **dataset1** with various parameters

Challenge: Get as many exons correctly aligned as possible with minimap2!

SETUP

1. git clone https://github.com/ksahlin/misc.git
2. cd misc/LongTREC/examples/
3. minimap2 --eqx -a **[PARAMS]** dataset1/genome.fa dataset1/query.fa > mm2.sam
4. *python evaluate_instance.py --gtf dataset1/annotation.gtf --samfile mm2.sam*

Try steps 3 and 4 on **dataset1** with various parameters

OUTPUT

#Exact	#Approx	min-E	min-A
7	25	68	37

Challenge: Get as many exons correctly aligned as possible with minimap2!

SETUP

1. git clone https://github.com/ksahlin/misc.git
2. cd misc/LongTREC/examples/
3. minimap2 --eqx -a **[PARAMS]** dataset1/genome.fa dataset1/query.fa > mm2.sam
4. *python evaluate_instance.py --gtf dataset1/annotation.gtf --samfile mm2.sam*

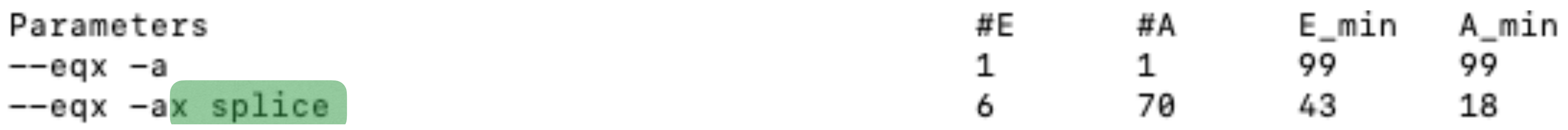
Try steps 3 and 4 on **dataset1** with various parameters

OUTPUT

```
#Exact      #Approx      min-E      min-A
7           25           68         37
```

#Exact:	Number of exact exon alignments
#Approx:	Number of exact OR approximate exon alignments (alnmt overlaps true exon)
min-E:	Smallest exon with an exact exon alignment
min-A:	Smallest exon with an exact OR approximate exon alignment nr_exact

Challenge: My attempts



```
Parameters                                                     #E        #A        E_min     A_min
--eqx -a                                                       1         1         99        99
--eqx -ax splice                                               6         70        43        18
```

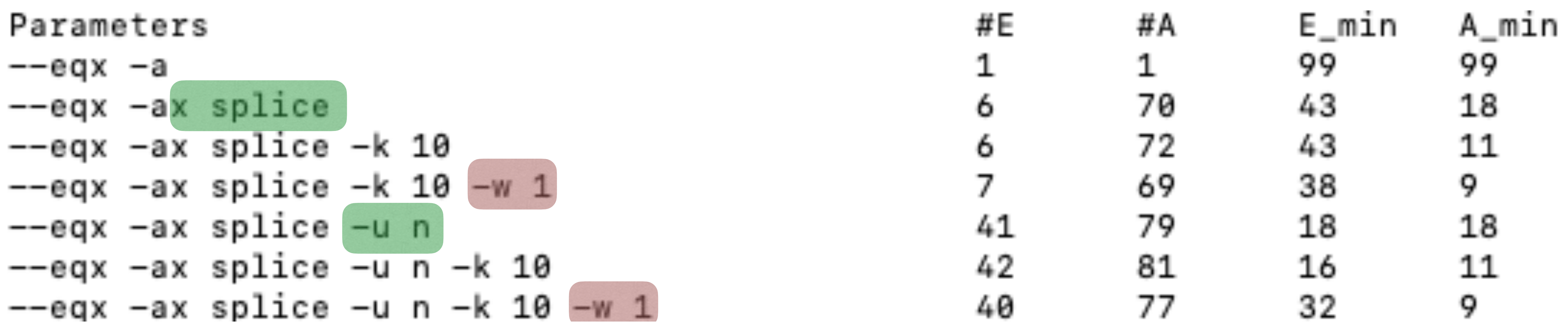
Challenge: My attempts

```
Parameters                                   #E      #A      E_min    A_min
--eqx -a                                     1       1       99       99
--eqx -ax splice                             6       70      43       18
--eqx -ax splice -k 10                       6       72      43       11
--eqx -ax splice -k 10 -w 1                  7       69      38       9
```

SciLifeLab

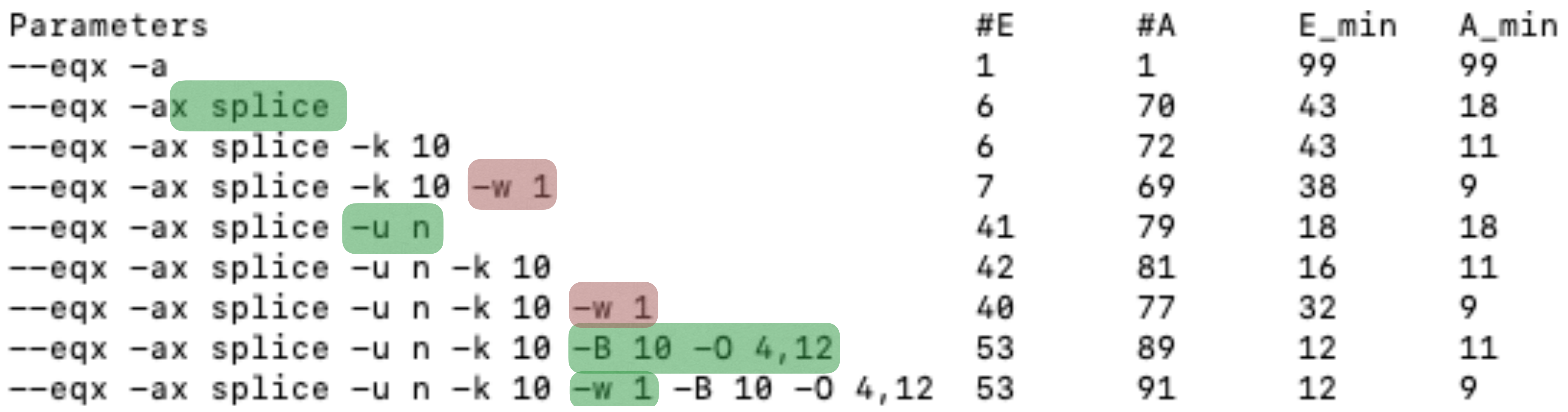
Stockholm University

Challenge: My attempts



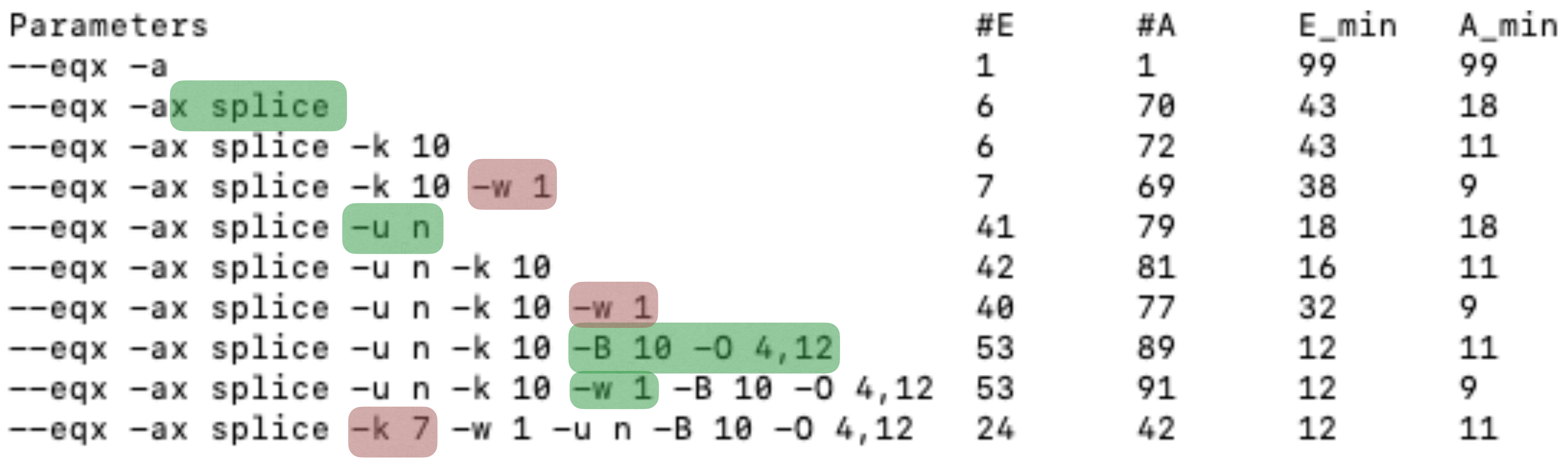
```
Parameters                                   #E      #A      E_min   A_min
--eqx -a                                     1       1       99      99
--eqx -ax splice                             6       70      43      18
--eqx -ax splice -k 10                       6       72      43      11
--eqx -ax splice -k 10 -w 1                  7       69      38      9
--eqx -ax splice -u n                        41      79      18      18
--eqx -ax splice -u n -k 10                  42      81      16      11
--eqx -ax splice -u n -k 10 -w 1             40      77      32      9
```

Challenge: My attempts



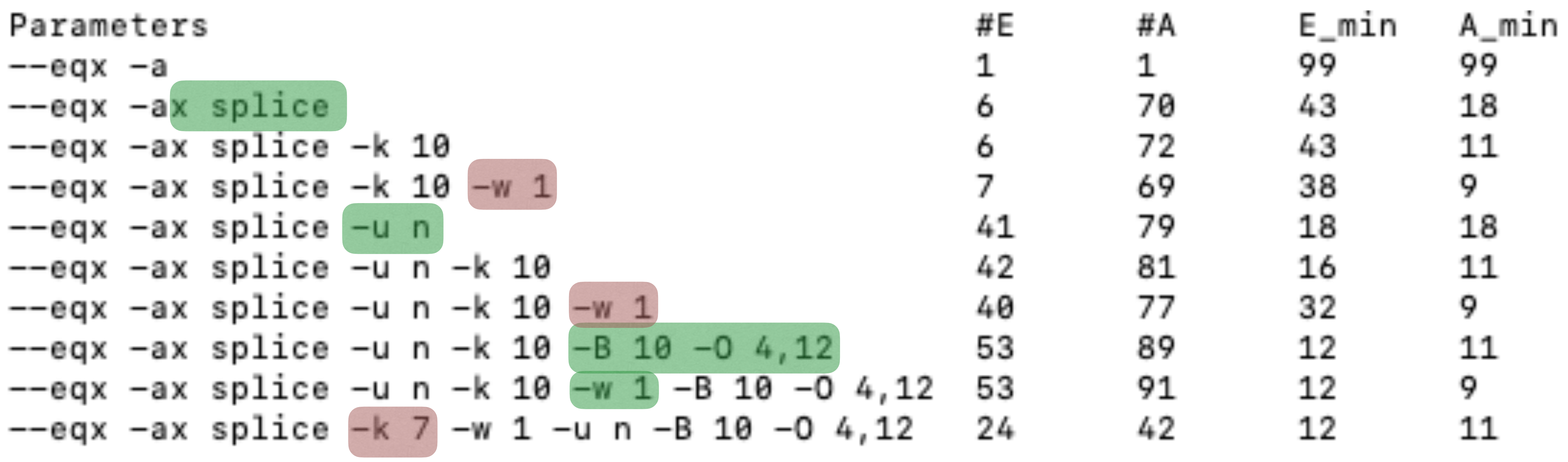
```
Parameters                                                 #E      #A      E_min    A_min
--eqx -a                                                   1       1       99       99
--eqx -ax splice                                           6       70      43       18
--eqx -ax splice -k 10                                     6       72      43       11
--eqx -ax splice -k 10 -w 1                                7       69      38       9
--eqx -ax splice -u n                                      41      79      18       18
--eqx -ax splice -u n -k 10                                42      81      16       11
--eqx -ax splice -u n -k 10 -w 1                           40      77      32       9
--eqx -ax splice -u n -k 10 -B 10 -O 4,12                  53      89      12       11
--eqx -ax splice -u n -k 10 -w 1 -B 10 -O 4,12             53      91      12       9
```

Challenge: My attempts



```
Parameters                                                  #E       #A       E_min    A_min
--eqx -a                                                    1        1        99       99
--eqx -ax splice                                            6        70       43       18
--eqx -ax splice -k 10                                      6        72       43       11
--eqx -ax splice -k 10 -w 1                                 7        69       38       9
--eqx -ax splice -u n                                       41       79       18       18
--eqx -ax splice -u n -k 10                                 42       81       16       11
--eqx -ax splice -u n -k 10 -w 1                            40       77       32       9
--eqx -ax splice -u n -k 10 -B 10 -O 4,12                   53       89       12       11
--eqx -ax splice -u n -k 10 -w 1 -B 10 -O 4,12              53       91       12       9
--eqx -ax splice -k 7 -w 1 -u n -B 10 -O 4,12               24       42       12       11
```

Challenge: My attempts



```
Parameters                                           #E     #A     E_min   A_min
--eqx -a                                             1      1      99      99
--eqx -ax splice                                     6      70     43      18
--eqx -ax splice -k 10                               6      72     43      11
--eqx -ax splice -k 10 -w 1                          7      69     38      9
--eqx -ax splice -u n                                41     79     18      18
--eqx -ax splice -u n -k 10                          42     81     16      11
--eqx -ax splice -u n -k 10 -w 1                     40     77     32      9
--eqx -ax splice -u n -k 10 -B 10 -O 4,12            53     89     12      11
--eqx -ax splice -u n -k 10 -w 1 -B 10 -O 4,12       53     91     12      9
--eqx -ax splice -k 7 -w 1 -u n -B 10 -O 4,12        24     42     12      11
```

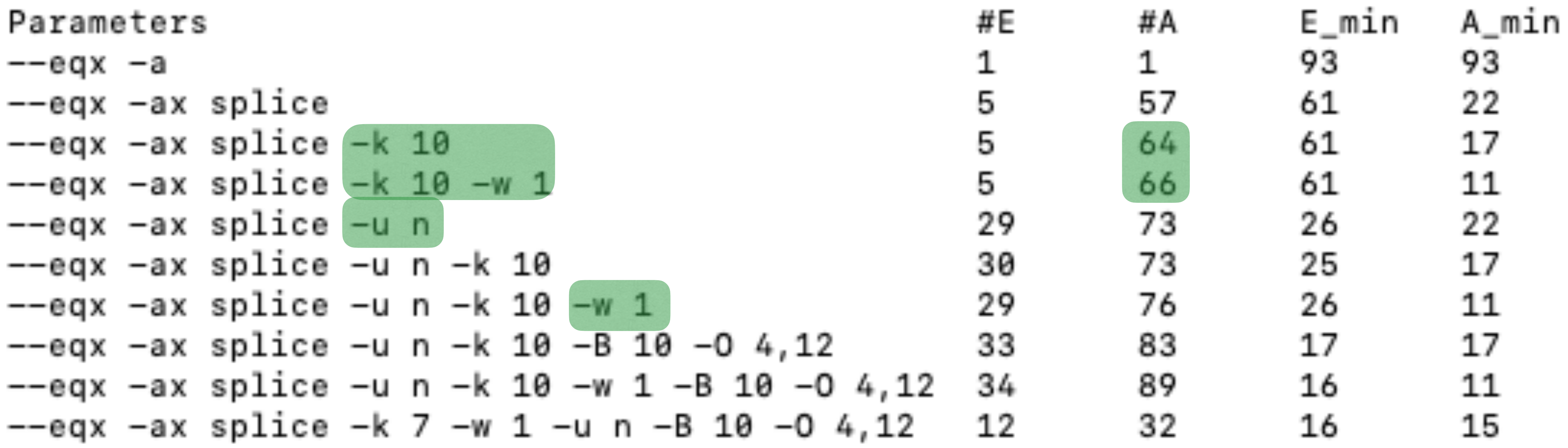
- Difficult to predict final outcome due to all heuristics and parameters (-w 1)
- Useful to know the basics: k-mer size, fitting to canonical splice sites.
- Extension alignment parameters (A, B, O, E) remains guesswork to this day.
    - But: lower O, E penalties prefers to open gaps - helpful to splice alignments

Three additional datasets

- Dataset2: 7% errors
- Dataset3: No errors, only GT-AG
- Dataset4: 7% errors, only GT-AG

Three additional datasets

- **Dataset2: 7% errors**
- Dataset3: No errors, only GT-AG
- Dataset4: 7% errors, only GT-AG

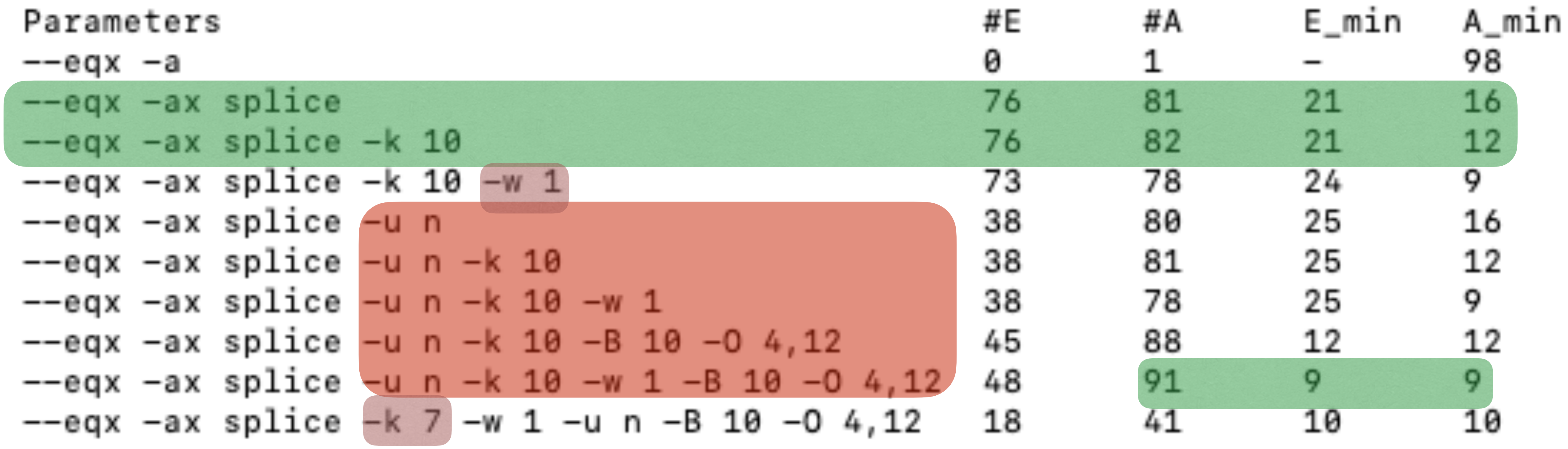


```
Parameters                                                #E       #A       E_min    A_min
--eqx -a                                                  1        1        93       93
--eqx -ax splice                                          5        57       61       22
--eqx -ax splice -k 10                                    5        64       61       17
--eqx -ax splice -k 10 -w 1                               5        66       61       11
--eqx -ax splice -u n                                     29       73       26       22
--eqx -ax splice -u n -k 10                               30       73       25       17
--eqx -ax splice -u n -k 10 -w 1                          29       76       26       11
--eqx -ax splice -u n -k 10 -B 10 -O 4,12                 33       83       17       17
--eqx -ax splice -u n -k 10 -w 1 -B 10 -O 4,12            34       89       16       11
--eqx -ax splice -k 7 -w 1 -u n -B 10 -O 4,12             12       32       16       15
```

- With errors/mutations - k and w is more influential
- Less exact matches - but almost the same amount of exons with approximate mappings

Three additional datasets

- Dataset2: 7% errors
- **Dataset3: No errors, only GT-AG**
- Dataset4: 7% errors, only GT-AG

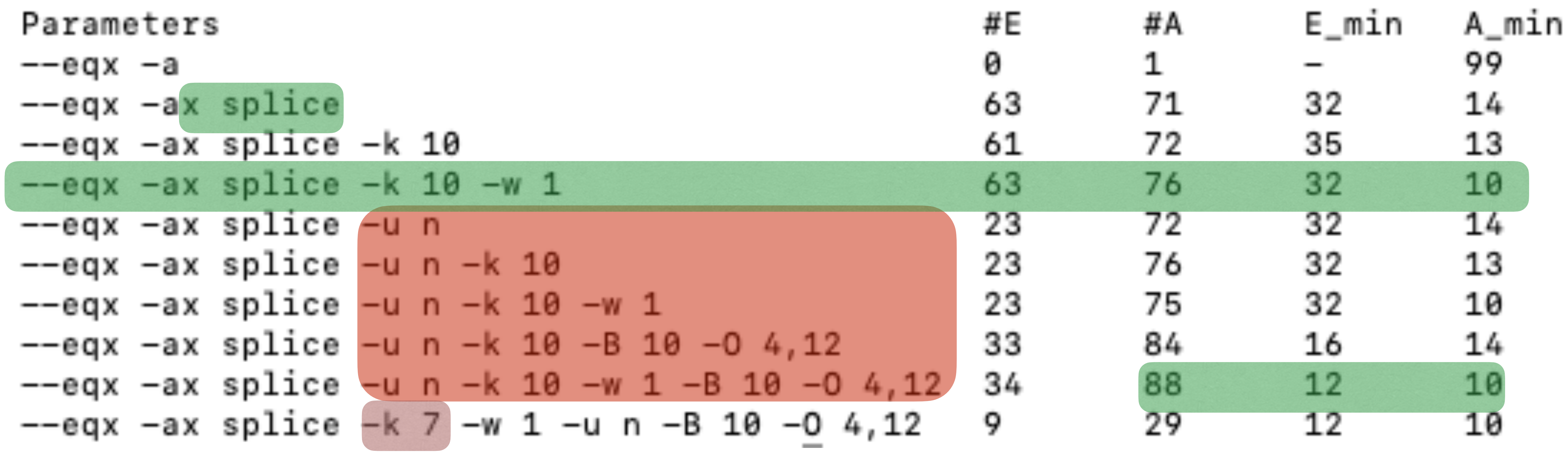


```
Parameters                                               #E      #A      E_min    A_min
--eqx -a                                                 0       1       -        98
--eqx -ax splice                                         76      81      21       16
--eqx -ax splice -k 10                                   76      82      21       12
--eqx -ax splice -k 10 -w 1                              73      78      24       9
--eqx -ax splice -u n                                    38      80      25       16
--eqx -ax splice -u n -k 10                              38      81      25       12
--eqx -ax splice -u n -k 10 -w 1                         38      78      25       9
--eqx -ax splice -u n -k 10 -B 10 -O 4,12                45      88      12       12
--eqx -ax splice -u n -k 10 -w 1 -B 10 -O 4,12           48      91      9        9
--eqx -ax splice -k 7 -w 1 -u n -B 10 -O 4,12            18      41      10       10
```

- Minimap2 is optimised for canonical splice sites!
- However, tuned alignment parameters still finds the most exon sites (91)

Three additional datasets

- Dataset2: 7% errors
- Dataset3: No errors, only GT-AG
- **Dataset4: 7% errors, only GT-AG**



```
Parameters                                               #E      #A      E_min   A_min
--eqx -a                                                 0       1       -       99
--eqx -ax splice                                         63      71      32      14
--eqx -ax splice -k 10                                   61      72      35      13
--eqx -ax splice -k 10 -w 1                              63      76      32      10
--eqx -ax splice -u n                                    23      72      32      14
--eqx -ax splice -u n -k 10                              23      76      32      13
--eqx -ax splice -u n -k 10 -w 1                         23      75      32      10
--eqx -ax splice -u n -k 10 -B 10 -O 4,12                33      84      16      14
--eqx -ax splice -u n -k 10 -w 1 -B 10 -O 4,12           34      88      12      10
--eqx -ax splice -k 7 -w 1 -u n -B 10 -O 4,12            9       29      12      10
```

- With errors/mutations - k and w is more influential
- Minimap2 is optimised for canonical splice sites!
- However, tuned alignment parameters still finds the most exon sites (88)

Take-home of the analysis

minimap2

✓ Minimap2 ‘just works’ (installation and running easy compared to competition)
✓ There is likely a parameter combination that is suitable for your needs
- The problem is how to find it
- Minimap2 works better with canonical splice sites

General splice alignment

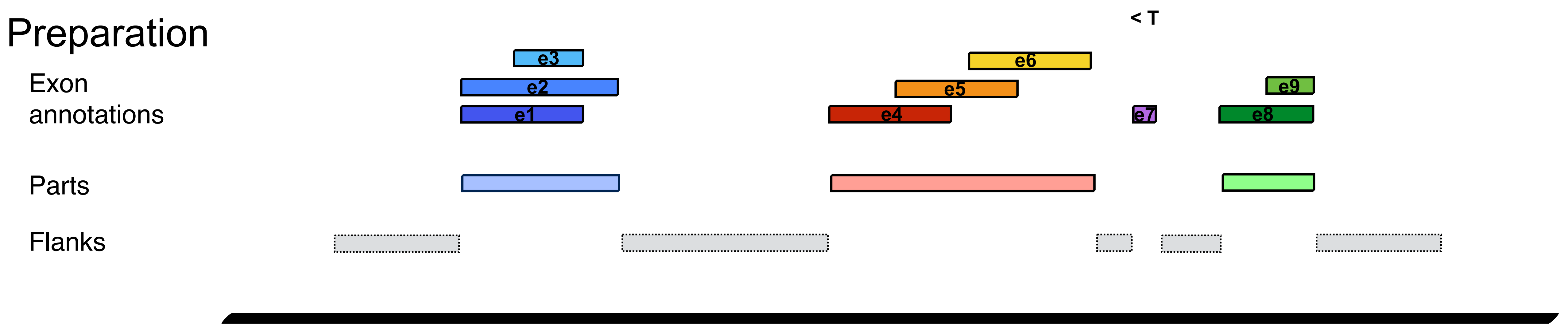
- Don’t blindly trust the input alignments for your isoform detection
- Know your dataset and genome
- If you are interested in smaller exons - there are better tools (deSALT, uLTRA)

Alternative aligners: uLTRA

+ uLTRA runs minimap2 under-the-hood to find alignments un unannotated regions
+ It picks the best alignment (alignment score) between uLTRA and minimap2

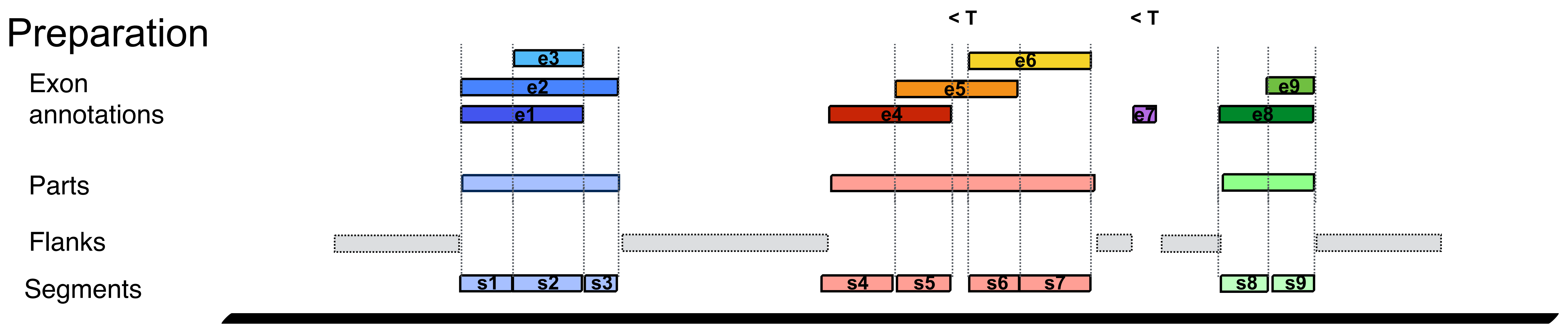
Alternative aligners: uLTRA

Preparation



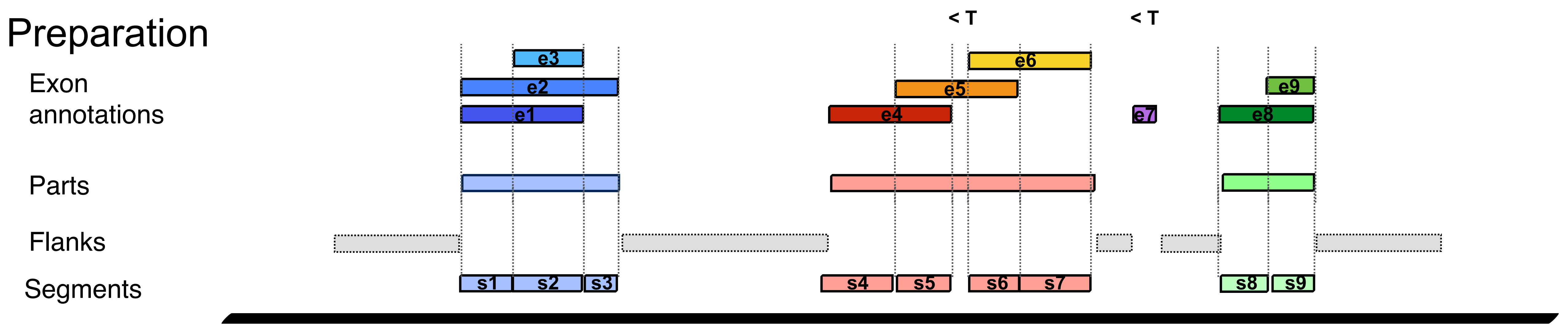
Alternative aligners: uLTRA

Preparation

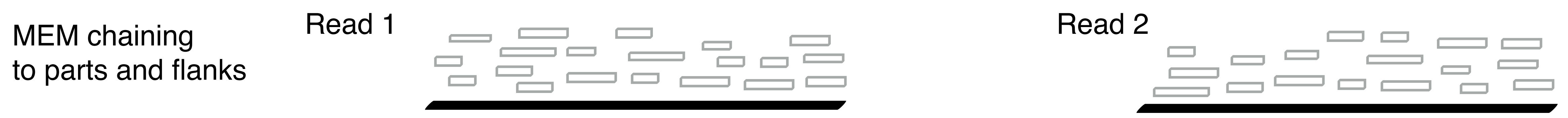


Alternative aligners: uLTRA

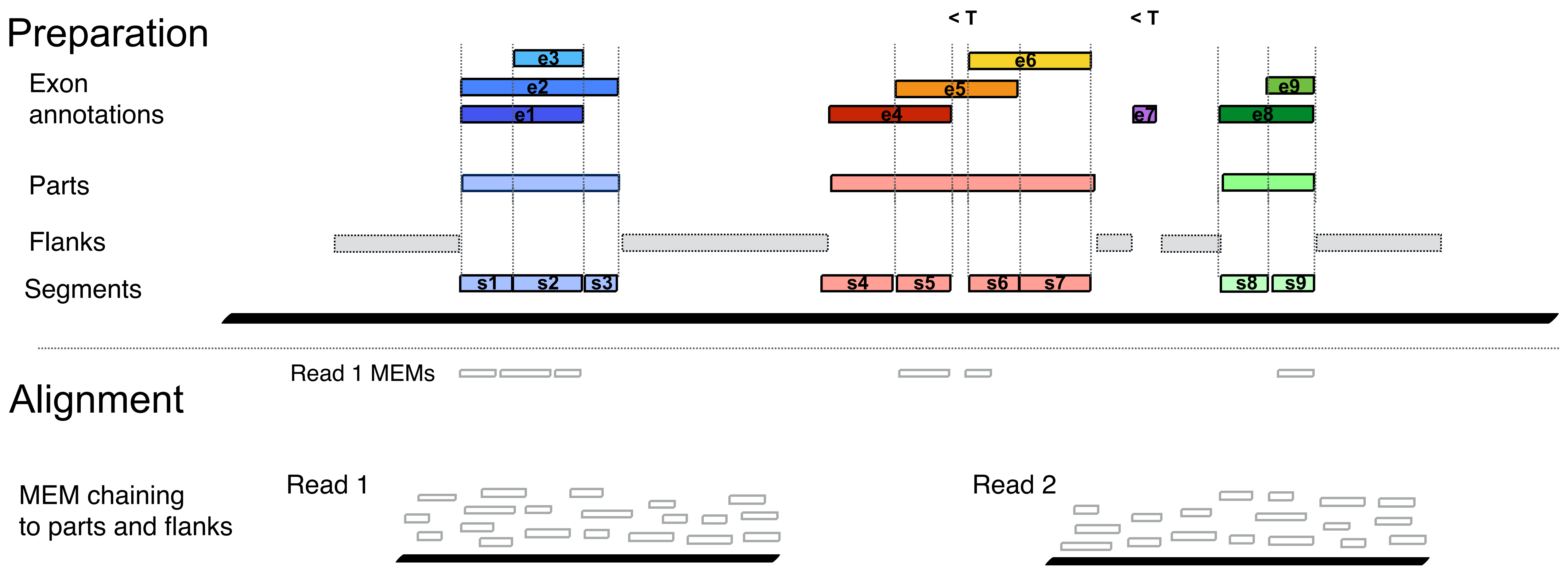
Preparation



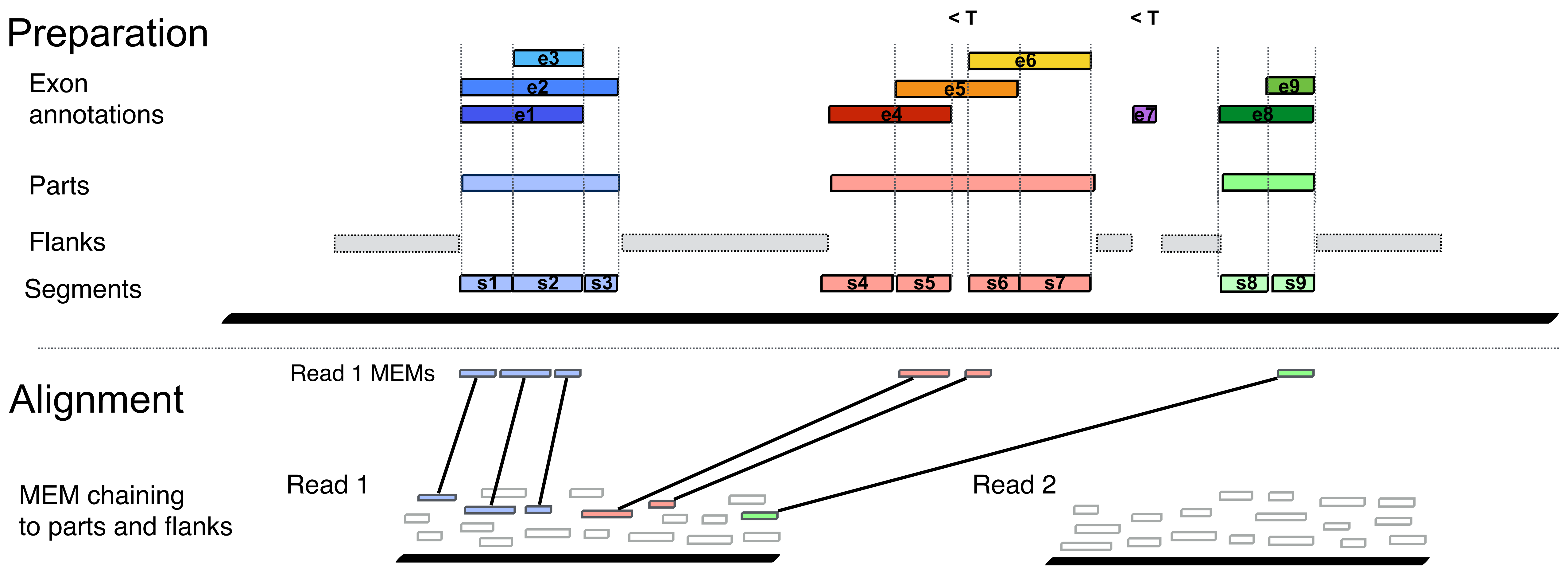
Alignment



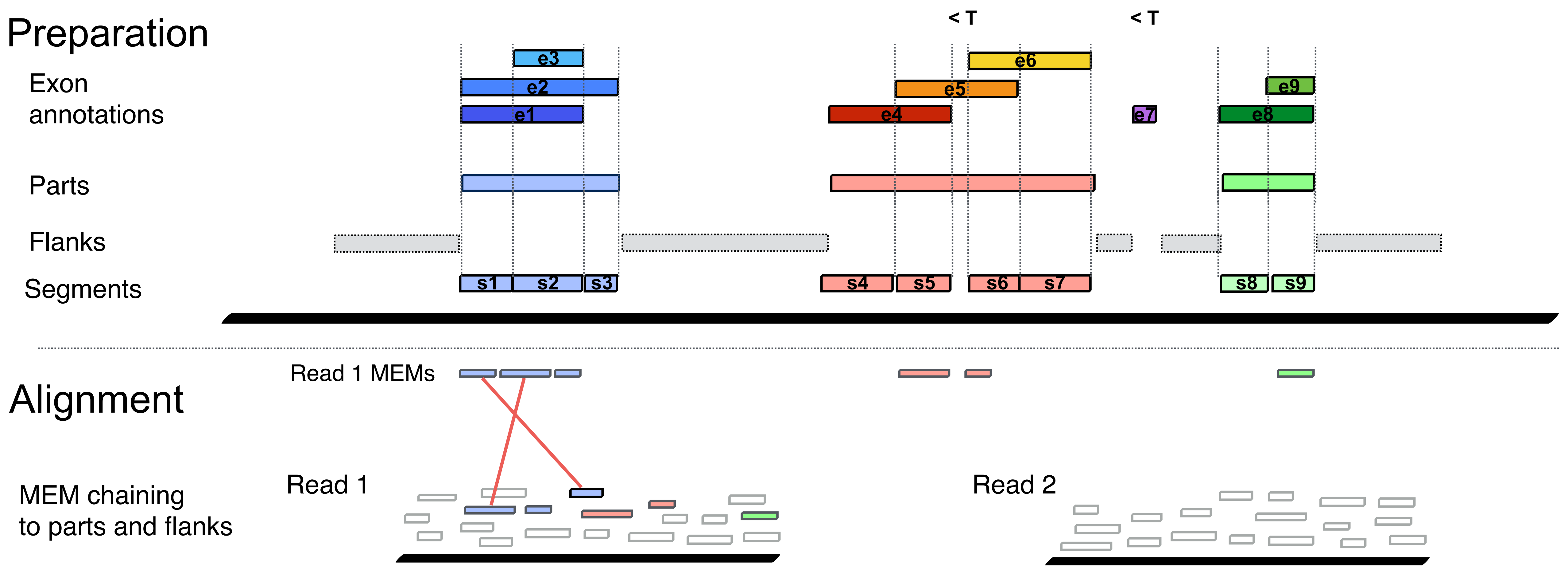
Alternative aligners: uLTRA



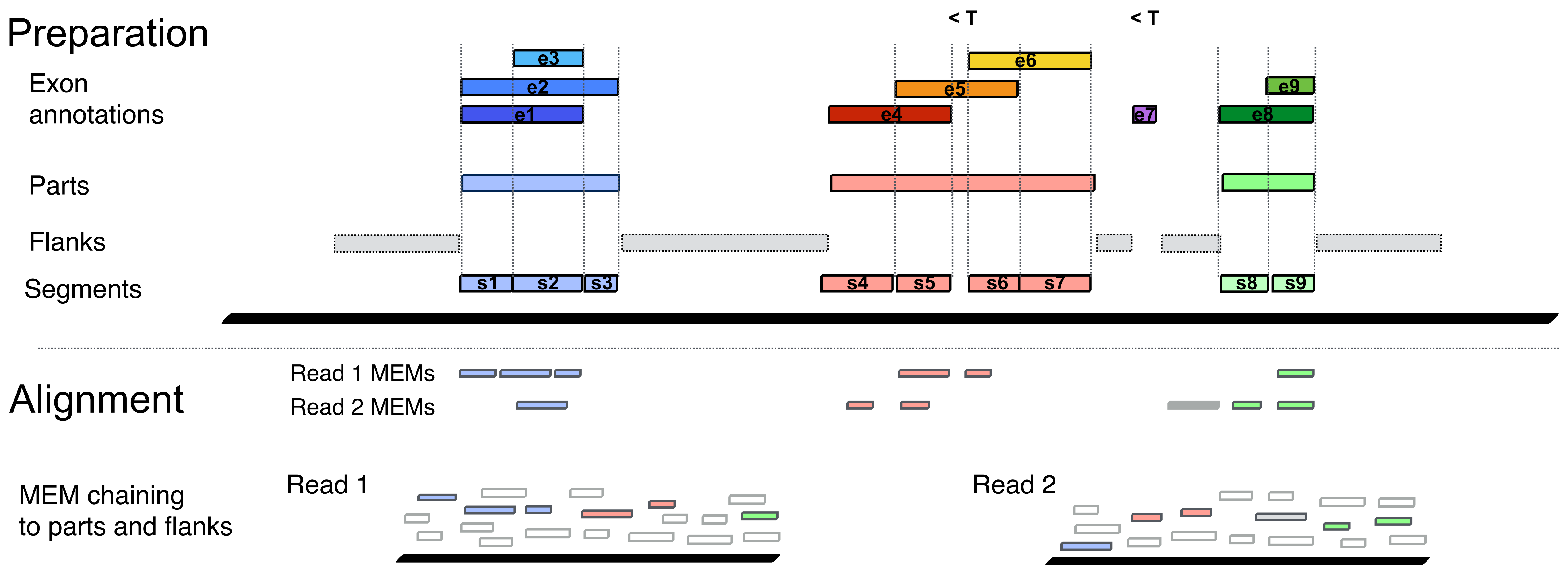
Alternative aligners: uLTRA



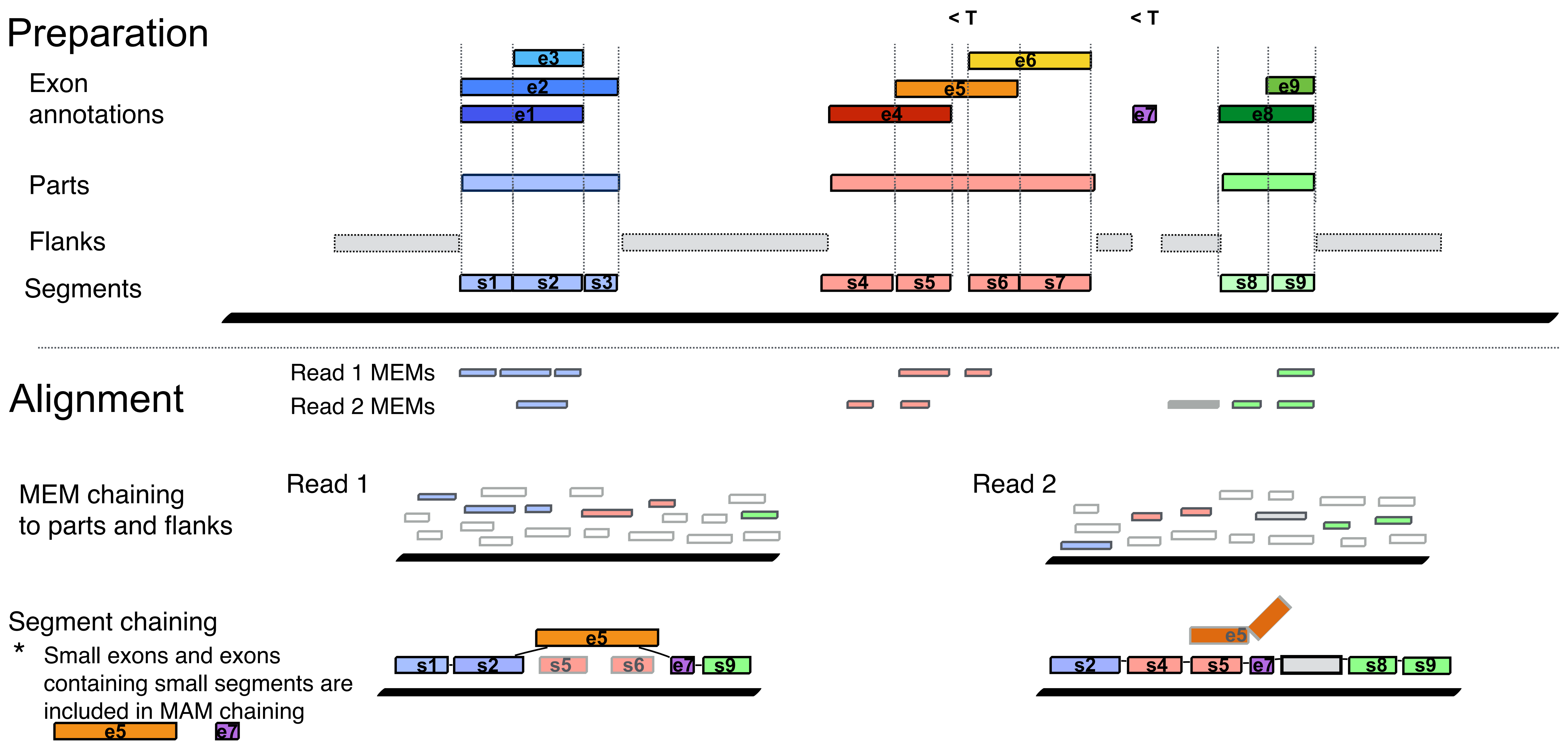
Alternative aligners: uLTRA



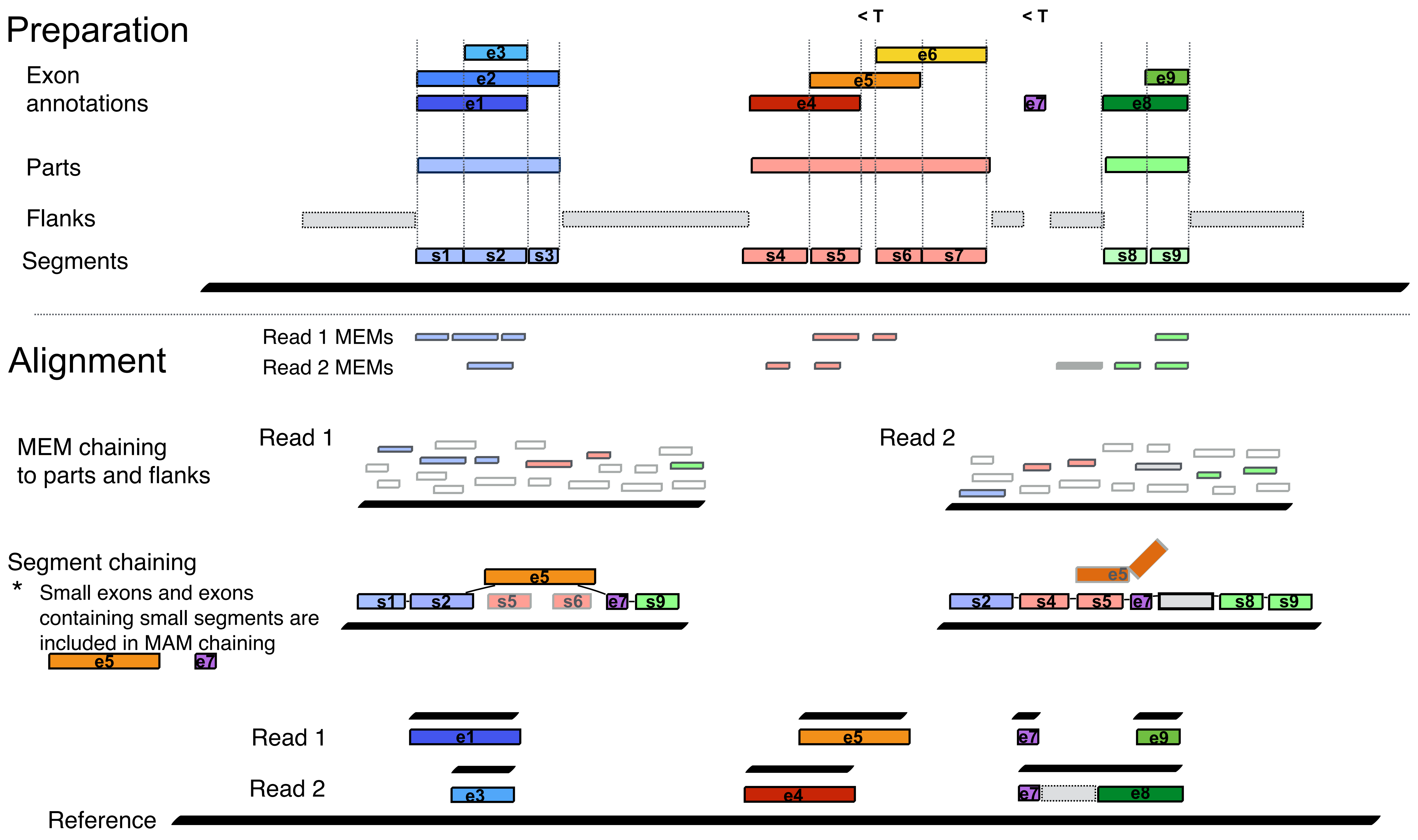
Alternative aligners: uLTRA



Alternative aligners: uLTRA



Alternative aligners: uLTRA



* Small exons and exons containing small segments are included in MAM chaining

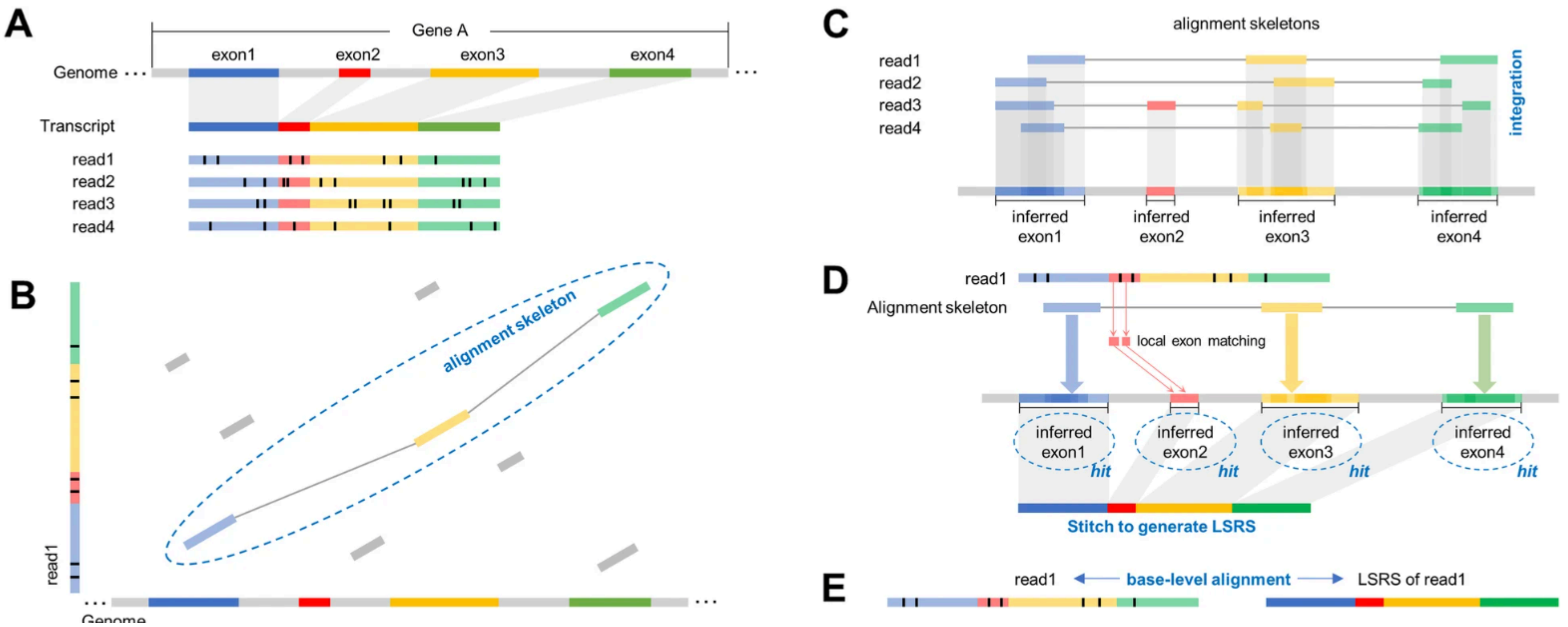
uLTRA v0.0.4.1 results

	Parameters	#E	#A	E_min	A_min
Dataset 1	default	94	95	5	5
Dataset 2	default	94	95	5	5
Dataset 4	default	94	95	5	5
Dataset 3	default	93	94	5	5

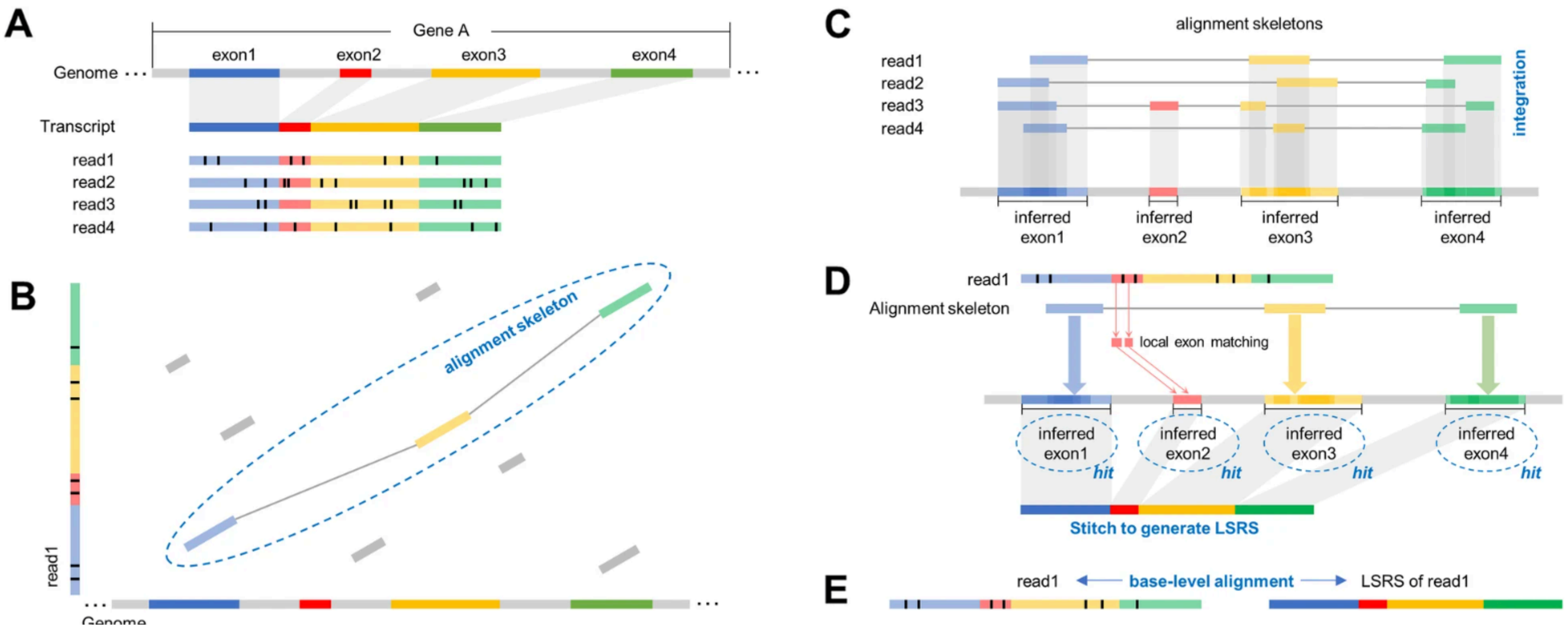
Take-home uLTRA

- ✓ Uses annotation to refine alignments
- ✓ Robust against different datasets (error rate, canonical or not..)
- ✓ Good for finding small exons (that are part of the annotation)
- ✓ Has minimap2's accuracy in unannotated regions
- Slower than minimap2 (at least 2 times slower)
- Disk space: prints a lot of intermediate/temporary files

Alternative aligners: deSALT



Alternative aligners: deSALT



- Could not install on my MacBooks (Intel and M1 chip)
- *Prediction based on previous analysis*: places somewhere between minimap2 and uLTRA in accuracy on these datasets

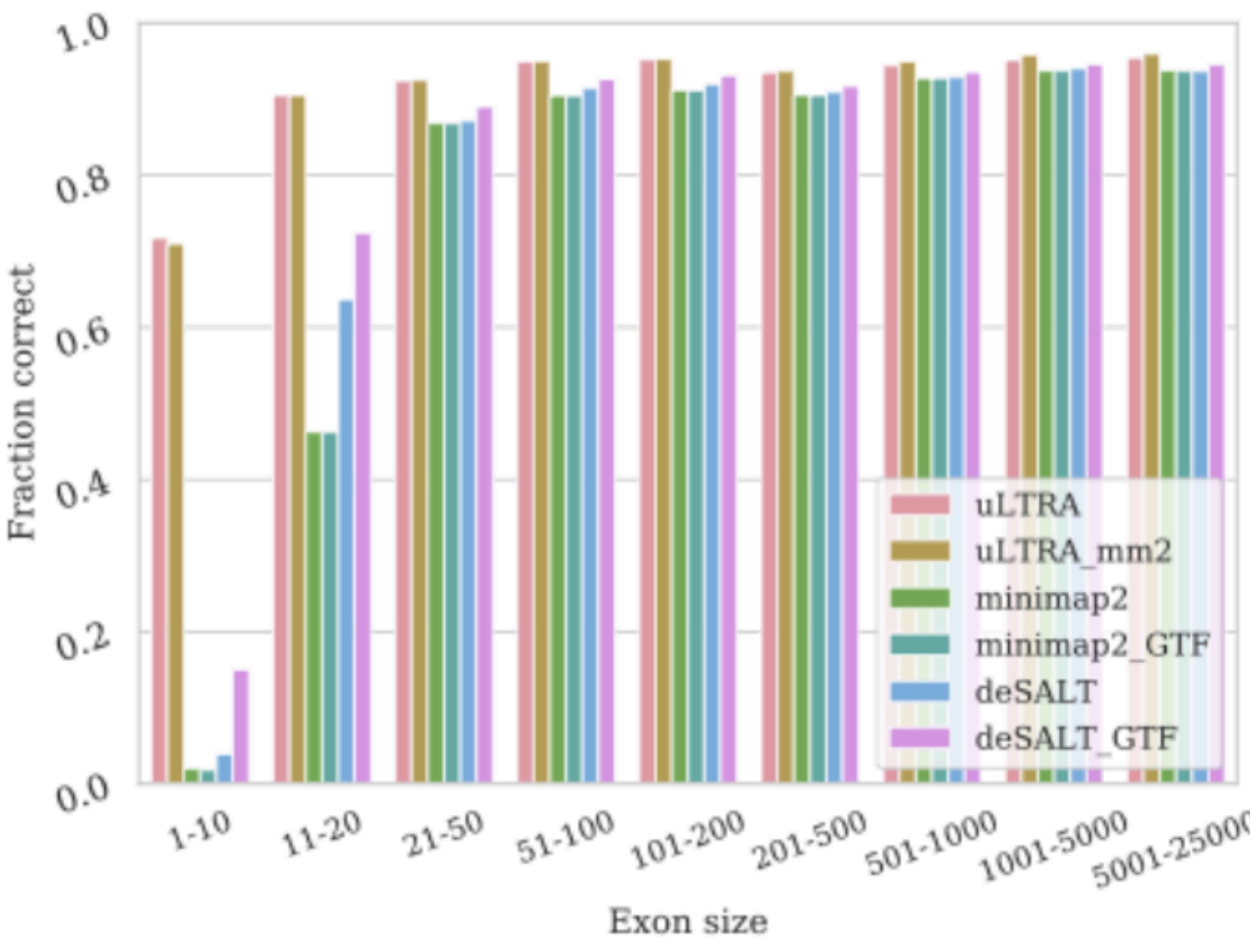
SciLifeLab

Stockholm University

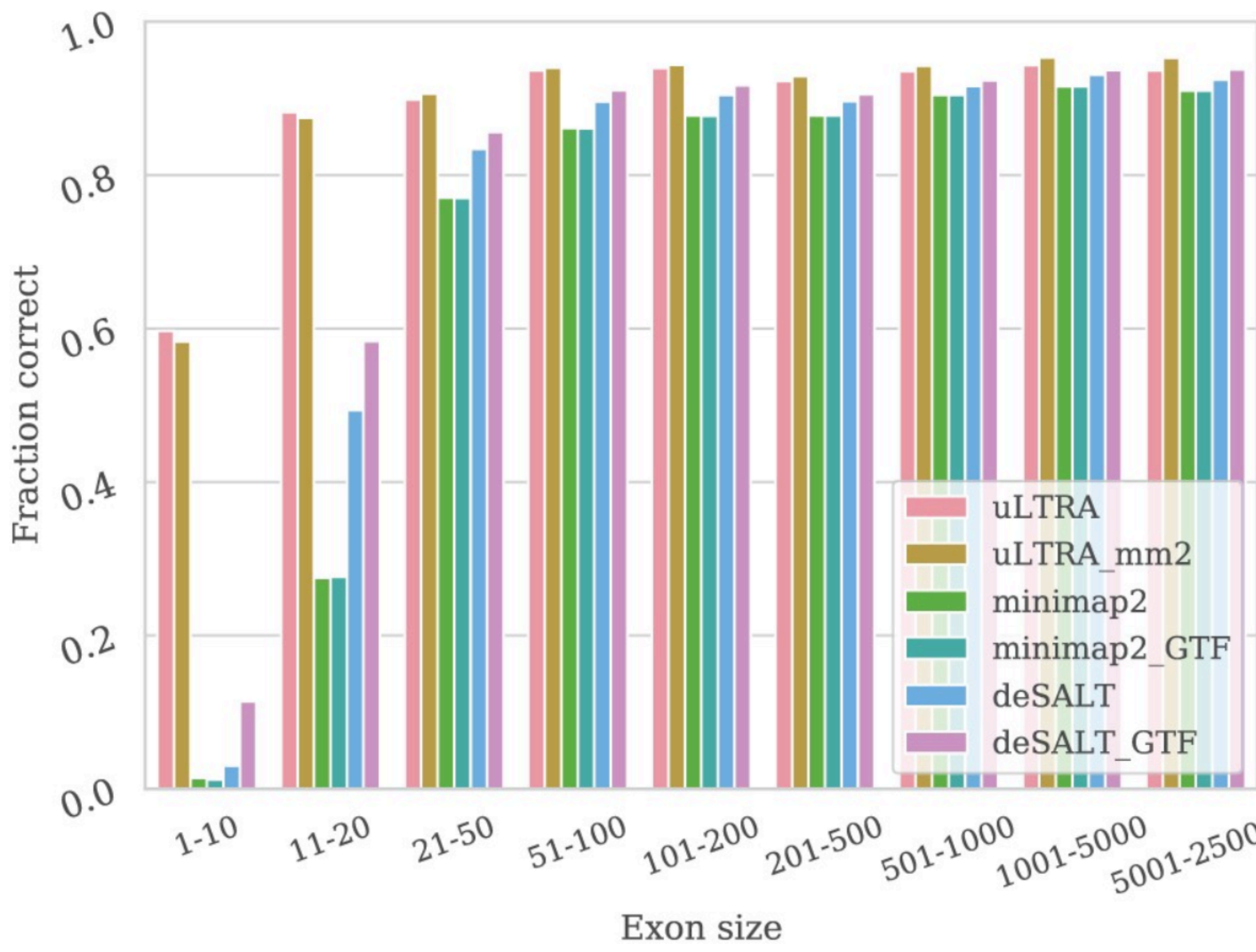
Accuracy binned by exon size (uLTRA, deSALT, minimap2)

Data from uLTRA paper

No errors (mapping transcripts to genome)



Transcript annotations with 7-8% errors



Interpreting the output (SAM)

The alignment of reads:

```
ref      AGCATGTTAGATAA**GATAGCTGTGCTAGTAGGCAGTCAGCGCCAT
+r002         aaaAGATAA*GGATA
```

Interpreting the output (SAM)

The alignment of reads:

```
ref       AGCATGTTAGATAA**GATAGCTGTGCTAGTAGGCAGTCAGCGCCAT
+r002          aaaAGATAA*GGATA
```

The resulting SAM file:

```
@HD VN:1.6 SO:coordinate
@SQ SN:ref LN:45
r002    0 ref  9 30 3S6M1P1I4M *  0    0 AAAAGATAAGGATA    *
```

Interpreting the output (SAM)

The alignment of reads:

```
ref      AGCATGTTAGATAA**GATAGCTGTGCTAGTAGGCAGTCAGCGCCAT
+r002          aaaAGATAA*GGATA
```

A ‘flag’ Ref start MAPQ CIGAR

The resulting SAM file:

```
@HD VN:1.6 SO:coordinate
@SQ SN:ref LN:45
r002     0 ref   9 30 3S6M1P1I4M *  0   0 AAAAGATAAGGATA    *
```

Interpreting the output (SAM)

The alignment of reads:

```
ref      AGCATGTTAGATAA**GATAGCTGTGCTAGTAGGCAGTCAGCGCCAT
+r002          aaaAGATAA*GGATA
+r003        gcctaAGCTAA
```

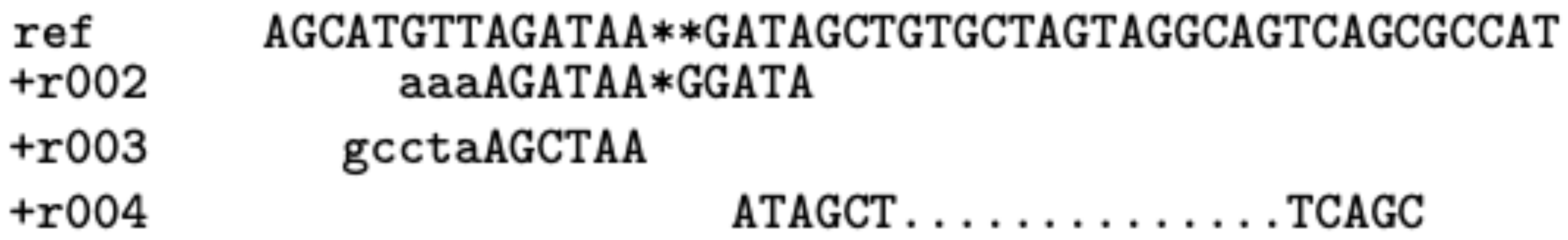
A 'flag'   Ref start   MAPQ   CIGAR

The resulting SAM file:

```
@HD VN:1.6 SO:coordinate
@SQ SN:ref LN:45
r002    0 ref  9 30 3S6M1P1I4M *  0   0 AAAAGATAAGGATA    *
r003    0 ref  9 30 5S6M       *  0   0 GCCTAAGCTAA       * SA:Z:ref,29,-,6H5M,17,0;
```

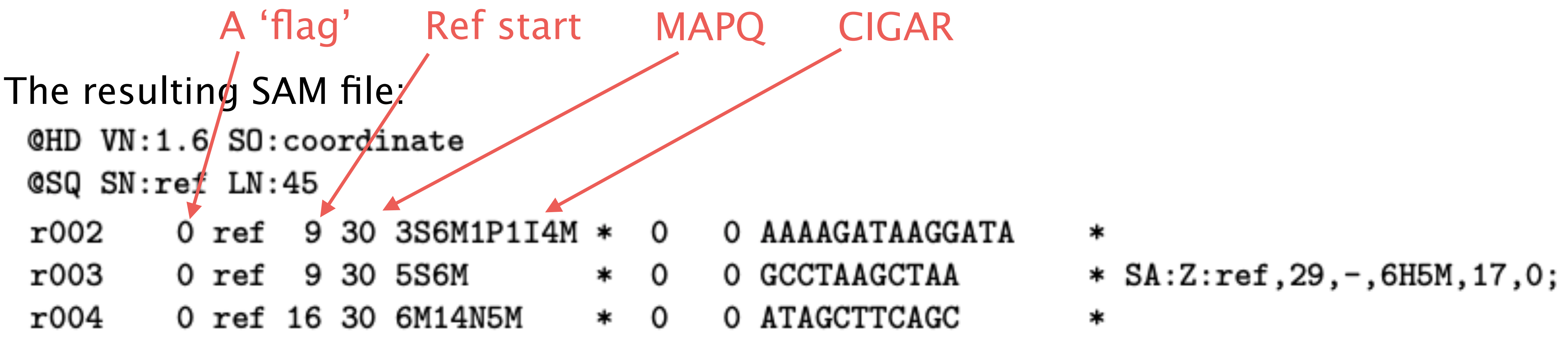
Interpreting the output (SAM)

The alignment of reads:



```
ref      AGCATGTTAGATAA**GATAGCTGTGCTAGTAGGCAGTCAGCGCCAT
+r002         aaaAGATAA*GGATA
+r003       gcctaAGCTAA
+r004                     ATAGCT..............TCAGC
```

The resulting SAM file:



```
@HD VN:1.6 SO:coordinate
@SQ SN:ref LN:45
r002    0 ref  9 30 3S6M1P1I4M *  0   0 AAAAGATAAGGATA    *
r003    0 ref  9 30 5S6M       *  0   0 GCCTAAGCTAA       * SA:Z:ref,29,-,6H5M,17,0;
r004    0 ref 16 30 6M14N5M    *  0   0 ATAGCTTCAGC       *
```

Interpreting the output (SAM)

The alignment of reads:

```
ref      AGCATGTTAGATAA**GATAGCTGTGCTAGTAGGCAGTCAGCGCCAT
+r002         aaaAGATAA*GGATA
+r003       gcctaAGCTAA
+r004                 ATAGCT..............TCAGC
-r003                        ttagctTAGGC
```

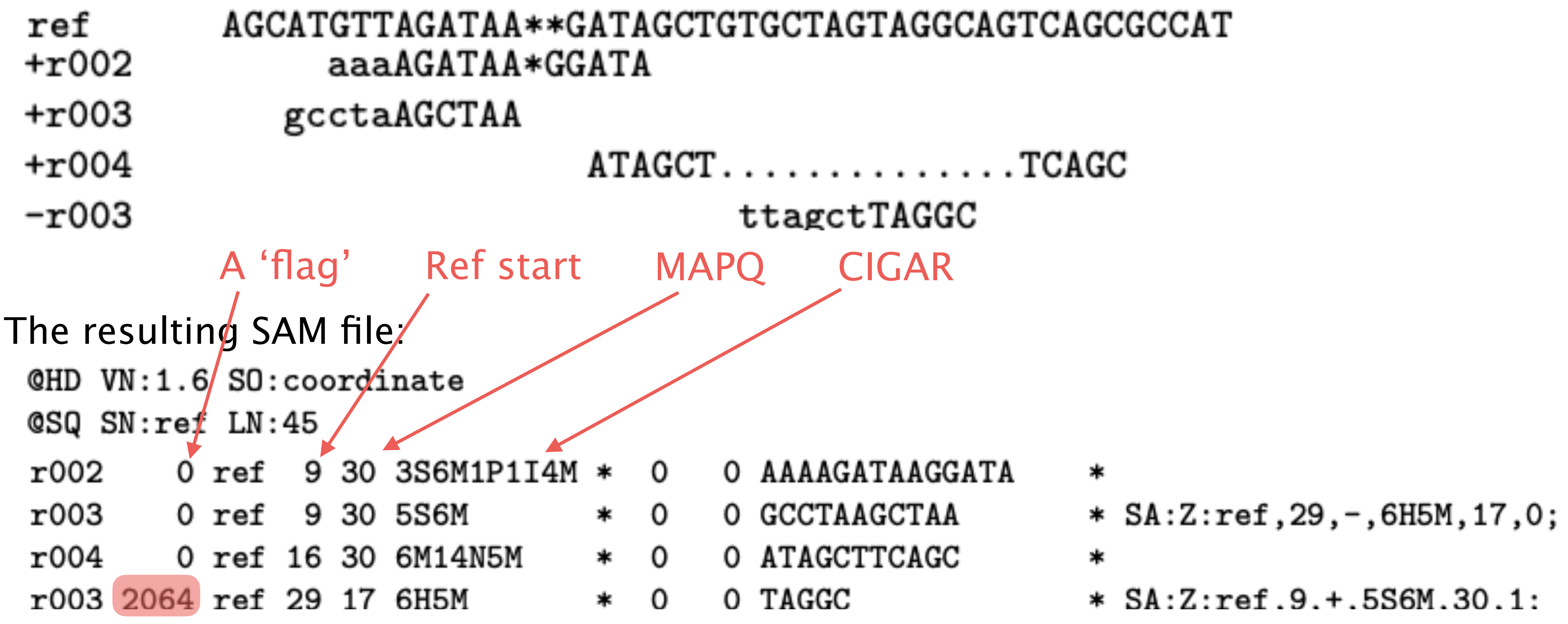
A ‘flag’   Ref start   MAPQ   CIGAR

The resulting SAM file:

```
@HD VN:1.6 SO:coordinate
@SQ SN:ref LN:45
r002    0 ref  9 30 3S6M1P1I4M *  0  0 AAAAGATAAGGATA  *
r003    0 ref  9 30 5S6M       *  0  0 GCCTAAGCTAA     * SA:Z:ref,29,-,6H5M,17,0;
r004    0 ref 16 30 6M14N5M    *  0  0 ATAGCTTCAGC     *
r003 2064 ref 29 17 6H5M       *  0  0 TAGGC           * SA:Z:ref,9,+,5S6M,30,1;
```

Interpreting the output (SAM)

The alignment of reads:



```
ref      AGCATGTTAGATAA**GATAGCTGTGCTAGTAGGCAGTCAGCGCCAT
+r002          aaaAGATAA*GGATA
+r003        gcctaAGCTAA
+r004                    ATAGCT..............TCAGC
-r003                           ttagctTAGGC
```

A ‘flag’
Ref start
MAPQ
CIGAR

The resulting SAM file:

```
@HD VN:1.6 SO:coordinate
@SQ SN:ref LN:45
r002    0 ref  9 30 3S6M1P1I4M *  0  0 AAAAGATAAGGATA   *
r003    0 ref  9 30 5S6M       *  0  0 GCCTAAGCTAA      * SA:Z:ref,29,-,6H5M,17,0;
r004    0 ref 16 30 6M14N5M    *  0  0 ATAGCTTCAGC      *
r003 2064 ref 29 17 6H5M       *  0  0 TAGGC            * SA:Z:ref,9,+,5S6M,30,1;
```

Useful links:
- SAM format: https://samtools.github.io/hts-specs/SAMv1.pdf
- Explain SAM flags: https://broadinstitute.github.io/picard/explain-flags.html

Trouble shooting and visualisation

- **Samtools**:
  - quick sanity check statistics (% aligned reads etc)
- **BLAT** - server version (https://genome.ucsc.edu/cgi-bin/hgBlat)
  - align a single or handful of reads with the absolute best accuracy
  - Extremely slow
  - Not the latest references
  - good for checking selected reads
- **Seaview (https://doua.prabi.fr/software/seaview)**
  - Align reads against the selves - transcript redundancy redundancy checking etc
- **IGV (https://igv.org/doc/desktop/)**:
  - Stacked read view - *can sometimes be* good for gene/transcript level visualisation (SNPs etc)

References

- **Seed-chain-extend:** Sahlin, K., Baudeau, T., Cazaux, B. *et al.* A survey of mapping algorithms in the long-reads era. *Genome Biol* **24**, 133 (2023). https://doi.org/10.1186/s13059-023-02972-3
- **uLTRA**: Kristoffer Sahlin, Veli Mäkinen, Accurate spliced alignment of long RNA sequencing reads, *Bioinformatics*, Volume 37, Issue 24, December 2021, Pages 4643–4651, https://doi.org/10.1093/bioinformatics/btab540
- **Minimap2**: Li H. (2018) Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics*, 34, 3094–3100.
- **deSALT:** Liu, B., Liu, Y., Li, J. et al. deSALT: fast and accurate long transcriptomic read alignment with de Bruijn graph-based index. Genome Biol 20, 274 (2019) doi:10.1186/s13059-019-1895-9
- **SAM format:** Heng Li, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, Richard Durbin, 1000 Genome Project Data Processing Subgroup, The Sequence Alignment/Map format and SAMtools, Bioinformatics, Volume 25, Issue 16, August 2009, Pages 2078–2079, https://doi.org/10.1093/bioinformatics/btp352
- **Samtools:** Heng Li, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, Richard Durbin, 1000 Genome Project Data Processing Subgroup, The Sequence Alignment/Map format and SAMtools, Bioinformatics, Volume 25, Issue 16, August 2009, Pages 2078–2079, https://doi.org/10.1093/bioinformatics/btp352
- **BLAT:** https://genome.ucsc.edu/cgi-bin/hgBlat