Sequencing Data Processing

D. Lavenier

Principle of DNA data storage

00 → A
01 → C
10 → G
11 → T

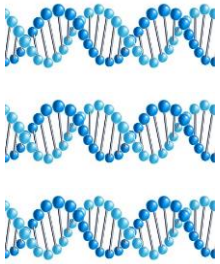
A → 00
C → 01
G → 10
T → 11

Coding | Synthesis | Storage | Retrieval | Sequencing | Decoding

Digital Files → DNA Sequences → DNA Molecules → Capsule → DNA Molecules → DNA Sequences → Digital Files

010100101
101010011
010001010
001010010

ATGAGGACCG
TTAGGACCAG
GCCATATAAC
GGTTATATAA



ATGAGGACCG
TTAGGACCAG

010100101
101010011

Processing of sequencing data

```
A → 00
C → 01
G → 10
T → 11
```

Sequencing

Sequencing Data Processing

Decoding

DNA Molecules

Noisy DNA Sequences

```
ATGAGGACCG
TTAGGACCAG
GTAGGACAGA
TTAGGACCCA
GGACCAGGAC
```

READS

Clean DNA Sequences

```
ATGAGGACCG
TTAGGACCAG
```

SEQUENCES

Digital Files

```
010100101
101010011
```

Errors

Synthesis

- Oligonucleotides of different sizes
- Missing oligonucleotides

Storage

- Broken oligonucleotides

Retrieval

- Erroneous oligonucleotides

Sequencing

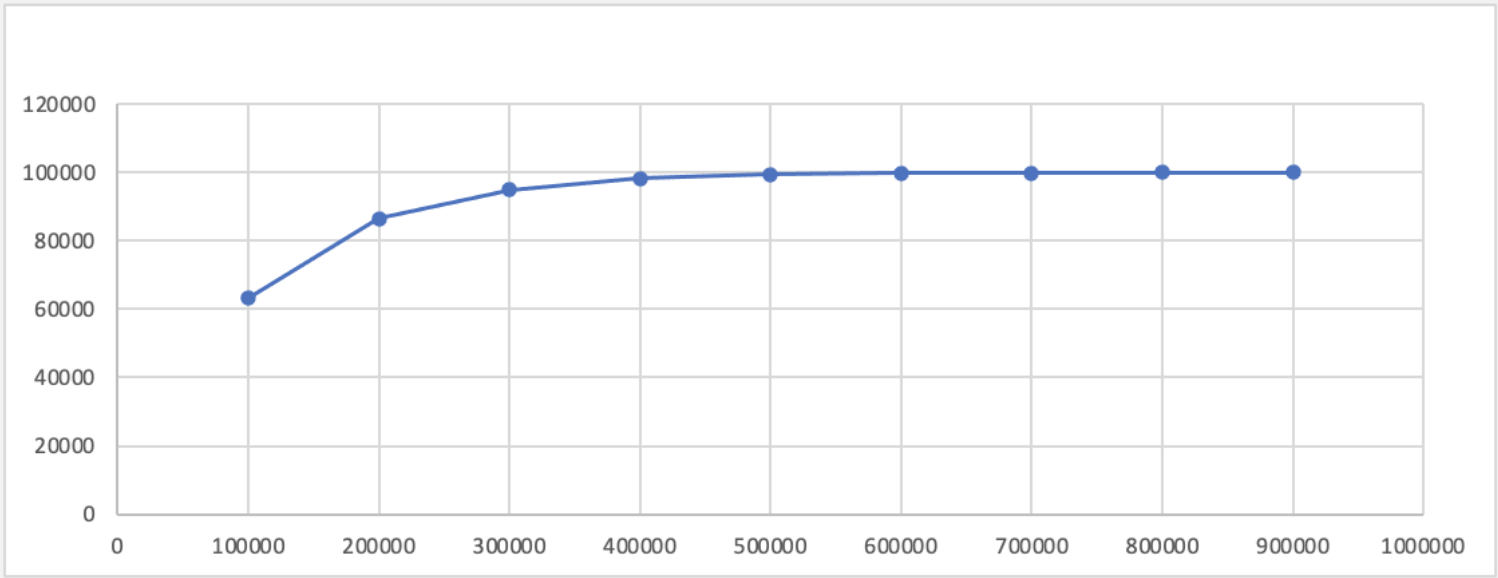
- Substitution / insertion / deletions of nucleotides

Sequencing

Random process

100 000 reference sequences

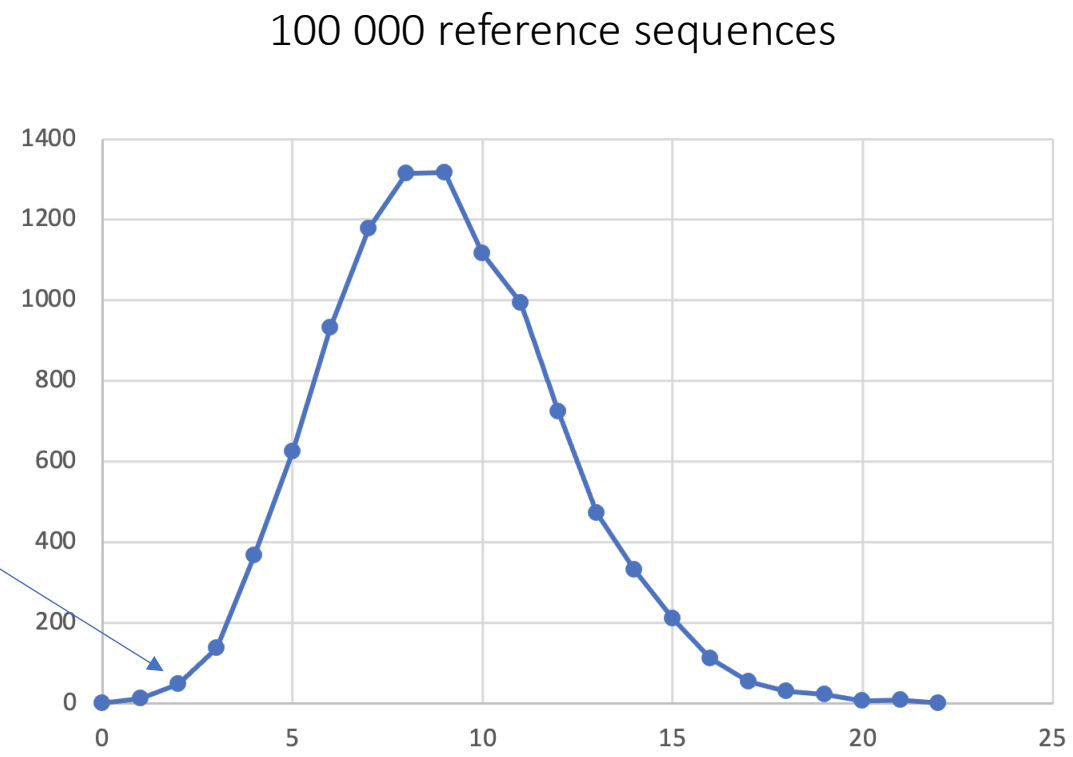
0	0
100000	63178
200000	86495
300000	94993
400000	98150
500000	99334
600000	99744
700000	99898
800000	99957
900000	99983



Sequencing

Coverage: 10X

48 sequences are in only 2 copies



0	0
1	11
2	48
3	136
4	366
5	624
6	932
7	1178
8	1315
9	1317
10	1116
11	994
12	723
13	471
14	331
15	210
16	111
17	54
18	29
19	21
20	5
21	8
22	0

Objective

Use redundancy of the sequencing data to provide good quality sequence to the decoding stage

Input:

Noisy sequences in many copies

Output

One representant of each sequences with correct size and as few errors as possible

- Correction capability → strong impact on the error correcting codes

Standard method

TGGCCACAGAGATGGATTAG
TTAGAGAGACAAGAGGACACG
GGATTATAGACCAATGGGACAC
GGATGAGACACAGATTAGGAG
TAAAAGATACCAATTAGGATAA
TTAGAGAGACAGAGGACCACG
GGATAAGACCAATCGGACAC
TGGCACACCGAGATGGATTAG
TTAGAGAGACAGAGGACACG
GGATATAGACCATTGGGACAC
TAAAGATACCAATTAGGTTAA
TGGCAGCCAGAGATGGATTAG
TTAGAGTGACCAGAGGCACACG
GGATATAGTCCAATGGGACAC
TAAAATACCAATTAGGATTAA
GGATACAGACCAATGCGACAC
TAATAGATACCAATTAGGATAA
TTAGAGAGACAGAGGTCCACG

Reads
Sequencing data

CLUSTERING

TGGCCACAGAGATGGATTAG
TGGCACACCGAGATGGATTAG
TGGCAGCCAGAGATGGATTAG

TTAGAGAGACAAGAGGACACG
TTAGAGAGACAGAGGACCACG
TTAGAGAGACAGAGGACACG
TTAGAGTGACCAGAGGCACACG
TTAGAGAGACAGAGGTCCACG

GGATTATAGACCAATGGGACAC
GGATAAGACCAATCGGACAC
GGATATAGACCATTGGGACAC
GGATATAGTCCAATGGGACAC
GGATACAGACCAATGCGACAC

GGATGAGACACAGATTAGGAG

TAAAGATACCAATTAGGATAA
TAAGATACCAATTAGGTTAA
TAAATACCAATTAGGATTAA
TATAGATACCAATTAGGATAA

clusters

CONSENSUS

TGGC CA CAGAGATGGATTAG
TGGCACACC GAGATGGATTAG
TGGCAG CCAGAGATGGATTAG
TGGCACACCAGAGATGGATTAG

TTAGAGAGACAAGAGGAC ACG
TTAGAGAGACA GAGGACCACG
TTAGAGAGACA GAGGAC ACG
TTAGAGTGACCAGAGGCACACG
TTAGAGAGACA GAGGTCCACG
TTAGAGAGACAAGAGGTCCACG

GGATTATAGACCAATGGGACAC
GGATAA GA CCAATCGGACAC
GGA ATAGAACCATTGGGACAC
GGATATAGTACCAATGGGACAC
GGATACAGA CCAATGCGACAC
GGATACAGAGCCAATGCGACAC

GGATGAGACACAGATTAGGAG

TAAAGATACCAATTAGGA TAA
TAA GATACCAATTAGG TTAA
TAAA TACCAATTAGGATTAA
TATAGATACCAATTAGGAT AA
TAAAGATACCAATTAGGATTAA

sequences

Clustering

Any clustering algorithms can be used

Need to define a distance between sequences

- Group of sequences with small distance put inside the same cluster

Hamming distance

sequences of identical size

Hamming distance = 2

```
ATAGGAGACACCAGGATTTAGAGCCAGATAACACAGATATTAAG
ATAGGAGACACCATGATTTAGAGCCAGATTACACAGATATTAAG
```

Levenshtein distance

Minimum number of operations to transform one string into another, using the three permitted operations: substitution, deletion and insertion.

Levenshtein distance = 4

```
ATAGGAGACACCAGGATTTAGAGCCAGATAACAC-GATAT-AAG
ATAGGAGACACCA-GATTTAGAGCCAGATTACACAGATATTAAG
```

Leveinshtein distance

Dynamic programming algorithm

Sequence A

Sequence B			C	T	C	G	A	T
		0	1	2	3	4	5	6
C	1	0	1	2	3	4	5
C	2	1	1	2	3	4	5
T	3	2	1	2	3	4	4
C	4	3	2	1	2	3	4
T	5	4	3	3	3	4	3

$$D_{i,j} = \min \begin{cases} D_{i,j-1} + 1 \\ D_{i-1,j} + 1 \\ D_{i-1,j-1} + C \end{cases}$$

C = 0 if A[i] = B[j] , otherwise C = 1

Leveinshtein distance

Dynamic programming algorithm

Sequence A

Sequence B			C	T	C	G	A	T
		0	1	2	3	4	5	6
C	1	0	1	2	3	4	5
C	2	1	1	2	3	4	5
T	3	2	1	2	3	4	4
C	4	3	2	1	2	3	4
T	5	4	3	3	3	4	3

initialisation

distance

$$D_{i,j} = \min \begin{cases} D_{i,j-1} + 1 \\ D_{i-1,j} + 1 \\ D_{i-1,j-1} + \mathbf{C} \end{cases}$$

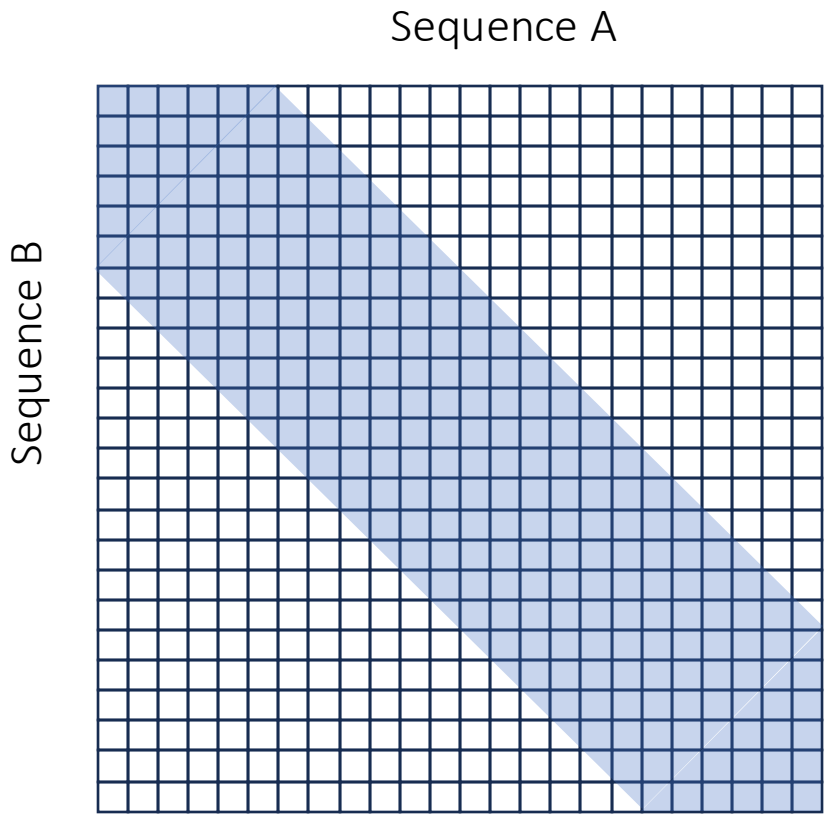
C = 0 if A[i] = B[j] , otherwise C = 1

Complexity: $0(N^2)$, N = size of the sequences

```
C - T C G A T
C C T C - - T
```

Leveinshtein distance - optimization

Dynamic programming algorithm



Similar sequences have small distance between them

Computation can be performed on the diagonal cells

All-vs-All sequence comparison

Too much computation

Input = 10^9 sequences → 10^{18} / 2 comparisons

Heuristics

indexing based on k-mers

Indexing

Based on K-mers

- Words of length K

```
1: ATGGACCAGGATTAGGAT
```

Create dictionary

```
ACCAGG 1,4
AGGATT 1,7
ATGGAC 1,0
ATTAGG 1,10
CAGGAT 1,6
CCAGGA 1,5
GACCAG 1,3
GATTAG 1,9
GGACCA 1,2
GGATTA 1,8
TAGGAT 1,12
TGGACC 1,1
TTAGGA 1,11
```

Indexing

```
1: ATGGACCAGGATTAGGAT
2: TGGATTAGAGGATTAGAC
```

Create dictionary

```
ACCAGG 1,4
AGAGGA 2,6
AGGATT 1,7 - 2,8
ATGGAC 1,0
ATTAGA 2,3 - 2,11
ATTAGG 1,10
CAGGAT 1,6
CCAGGA 1,5
GACCAG 1,3
GAGGAT 2,7
GATTAG 1,9 - 2,2 - 2,10
GGACCA 1,2
GGATTA 1,8 - 2,1 - 2,9
TAGAGG 2,5
TAGGAT 1,12
TGGACC 1,1
TGGATT 2,0
TTAGAC 2,12
TTAGAG 2,4
TTAGGA 1,11
```

K = 6

Seed & Extend heuristic

ATTGACCAGGATTAGGAT

not in dictionary

```
1: ATGGACCAGGATTAGGAT
2: TGGATTAGAGGATTAGAC
```

```
ACCAGG 1,4
AGAGGA 2,6
AGGATT 1,7 - 2,8
ATGGAC 1,0
ATTAGA 2,3 - 2,11
ATTAGG 1,10
CAGGAT 1,6
CCAGGA 1,5
GACCAG 1,3
GAGGAT 2,7
GATTAG 1,9 - 2,2 - 2,10
GGACCA 1,2
GGATTA 1,8 - 2,1 - 2,9
TAGAGG 2,5
TAGGAT 1,12
TGGACC 1,1
TGGATT 2,0
TTAGAC 2,12
TTAGAG 2,4
TTAGGA 1,11
```

Seed & Extend heuristic

A**TTGACC**AGGATTAGGAT

not in dictionary

```
1: ATGGACCAGGATTAGGAT
2: TGGATTAGAGGATTAGAC
```

```
ACCAGG 1,4
AGAGGA 2,6
AGGATT 1,7 - 2,8
ATGGAC 1,0
ATTAGA 2,3 - 2,11
ATTAGG 1,10
CAGGAT 1,6
CCAGGA 1,5
GACCAG 1,3
GAGGAT 2,7
GATTAG 1,9 - 2,2 - 2,10
GGACCA 1,2
GGATTA 1,8 - 2,1 - 2,9
TAGAGG 2,5
TAGGAT 1,12
TGGACC 1,1
TGGATT 2,0
TTAGAC 2,12
TTAGAG 2,4
TTAGGA 1,11
```

Seed & Extend heuristic

1: ATGGACCAGGATTAGGAT
2: TGGATTAGAGGATTAGAC

AT**TGACCA**GGATTAGGAT

not in dictionary

```
ACCAGG 1,4
AGAGGA 2,6
AGGATT 1,7 - 2,8
ATGGAC 1,0
ATTAGA 2,3 - 2,11
ATTAGG 1,10
CAGGAT 1,6
CCAGGA 1,5
GACCAG 1,3
GAGGAT 2,7
GATTAG 1,9 - 2,2 - 2,10
GGACCA 1,2
GGATTA 1,8 - 2,1 - 2,9
TAGAGG 2,5
TAGGAT 1,12
TGGACC 1,1
TGGATT 2,0
TTAGAC 2,12
TTAGAG 2,4
TTAGGA 1,11
```

Seed & Extend heuristic

```
1: ATGGACCAGGATTAGGAT
2: TGGATTAGAGGATTAGAC
```

ATT**GACCAG**GATTAGGAT

```
ACCAGG 1,4
AGAGGA 2,6
AGGATT 1,7 - 2,8
ATGGAC 1,0
ATTAGA 2,3 - 2,11
ATTAGG 1,10
CAGGAT 1,6
CCAGGA 1,5
GACCAG 1,3
GAGGAT 2,7
GATTAG 1,9 - 2,2 - 2,10
GGACCA 1,2
GGATTA 1,8 - 2,1 - 2,9
TAGAGG 2,5
TAGGAT 1,12
TGGACC 1,1
TGGATT 2,0
TTAGAC 2,12
TTAGAG 2,4
TTAGGA 1,11
```

```
   ATTGACCAGGATTAGGAT
1: ATGGACCAGGATTAGGAT
```

Seed & Extend heuristic

```
1: ATGGACCAGGATTAGGAT
2: TGGATTAGAGGATTAGAC
```

```
ACCAGG 1,4
AGAGGA 2,6
AGGATT 1,7 - 2,8
ATGGAC 1,0
ATTAGA 2,3 - 2,11
ATTAGG 1,10
CAGGAT 1,6
CCAGGA 1,5
GACCAG 1,3
GAGGAT 2,7
GATTAG 1,9 - 2,2 - 2,10
GGACCA 1,2
GGATTA 1,8 - 2,1 - 2,9
TAGAGG 2,5
TAGGAT 1,12
TGGACC 1,1
TGGATT 2,0
TTAGAC 2,12
TTAGAG 2,4
TTAGGA 1,11
```

ATTGACCAG**GATTAG**GAT

```
    ATTGACCAGGATTAGGAT
2:        TGGATTAGAGGATTAGAC
```

Seed & Extend heuristic

```
1: ATGGACCAGGATTAGGAT
2: TGGATTAGAGGATTAGAC
```

ATTGACCAG**GATTAG**GAT

```
ACCAGG 1,4
AGAGGA 2,6
AGGATT 1,7 - 2,8
ATGGAC 1,0
ATTAGA 2,3 - 2,11
ATTAGG 1,10
CAGGAT 1,6
CCAGGA 1,5
GACCAG 1,3
GAGGAT 2,7
GATTAG 1,9 - 2,2 - 2,10
GGACCA 1,2
GGATTA 1,8 - 2,1 - 2,9
TAGAGG 2,5
TAGGAT 1,12
TGGACC 1,1
TGGATT 2,0
TTAGAC 2,12
TTAGAG 2,4
TTAGGA 1,11
```

```
    ATTGACCAG GATTAG GAT
2:  TGGATTAGAG GATTAG AC
```

Seed & Extend heuristic

```
1: ATGGACCAGGATTAGGAT
2: TGGATTAGAGGATTAGAC
```

ATTGACCAG**GATTAG**GAT

```
ACCAGG 1,4
AGAGGA 2,6
AGGATT 1,7 - 2,8
ATGGAC 1,0
ATTAGA 2,3 - 2,11
ATTAGG 1,10
CAGGAT 1,6
CCAGGA 1,5
GACCAG 1,3
GAGGAT 2,7
GATTAG 1,9 - 2,2 - 2,10
GGACCA 1,2
GGATTA 1,8 - 2,1 - 2,9
TAGAGG 2,5
TAGGAT 1,12
TGGACC 1,1
TGGATT 2,0
TTAGAC 2,12
TTAGAG 2,4
TTAGGA 1,11
```

ATTGACCAG**GATTAG**GAT

i: ATTGACCA**GATTAG**GAT

alignment

ATTGACCAG**GATTAG**GAT

ATTGACCA**-GATTAG**GAT

Seed & Extend heuristic

Does not guaranty to find all alignments

In practice: very few false negatives

All-vs-all complexity:

- Create dictionary
- O(N) distance calculations

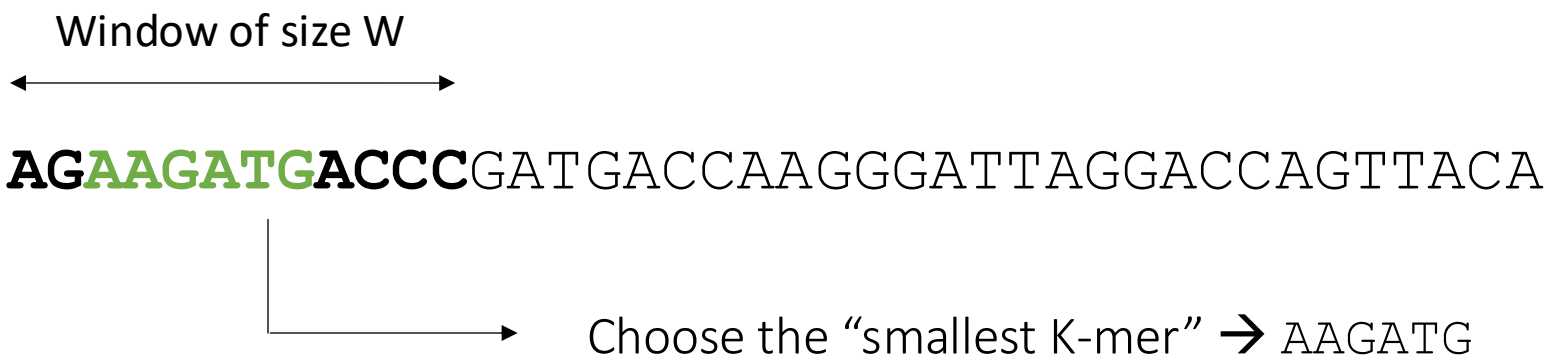
Seed & Extend heuristic optimization

Does not index all K-mers

- Aim: reduce space search, reduce dictionary

Selection of chosen K-mers → minimizers

K=6
W=12



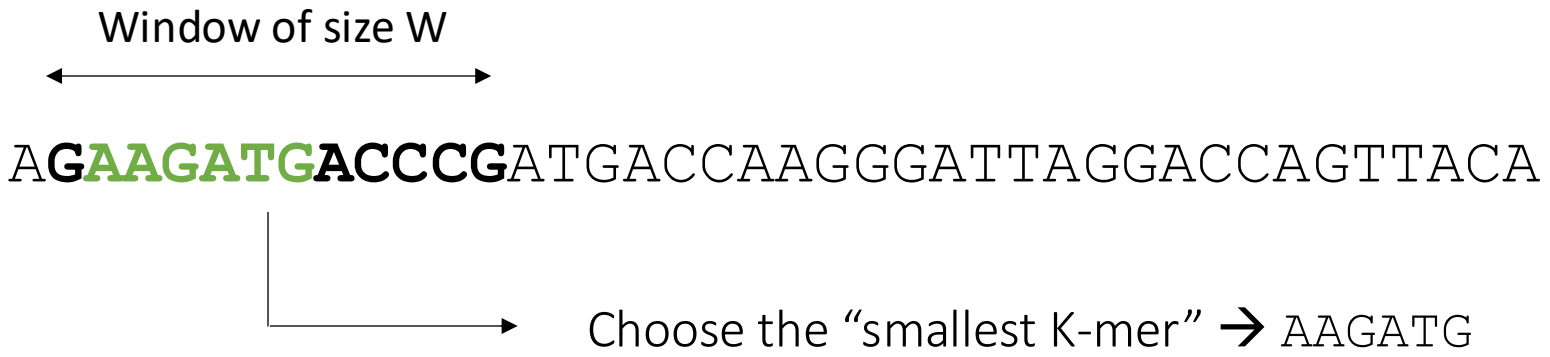
Seed & Extend heuristic optimization

Does not index all K-mers

- Aim: reduce space search, reduce dictionary

Selection of chosen K-mers → minimizers

K=6
W=12



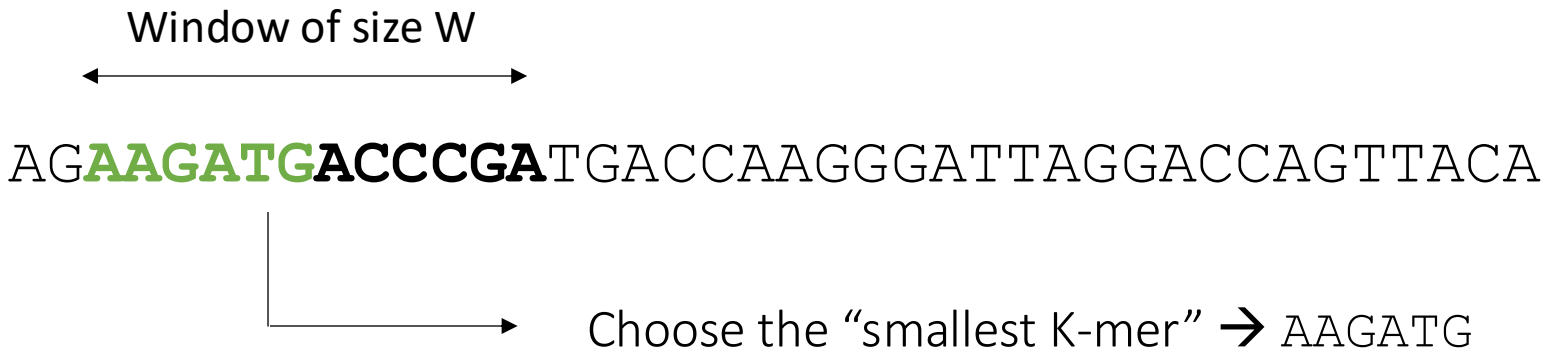
Seed & Extend heuristic optimization

Does not index all K-mers

- Aim: reduce space search, reduce dictionary

Selection of chosen K-mers → minimizers

K=6
W=12



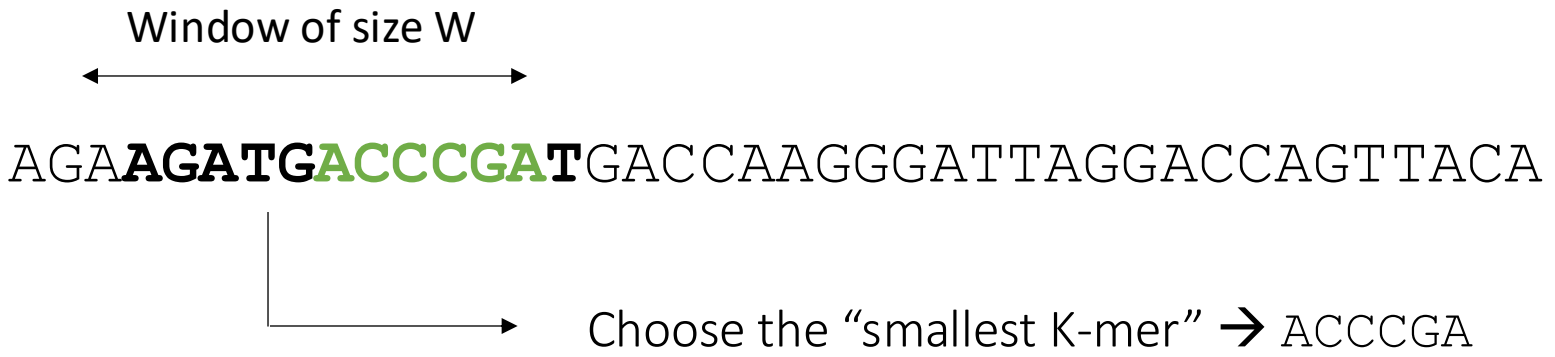
Seed & Extend heuristic optimization

Does not index all K-mers

- Aim: reduce space search, reduce dictionary

Selection of chosen K-mers → minimizers

K=6
W=12



Standard method

TGGCCACAGAGATGGATTAG
TTAGAGAGACAAGAGGACACG
GGATTATAGACCAATGGGACAC
GGATGAGACACAGATTAGGAG
TAAAAGATACCAATTAGGATAA
TTAGAGAGACAGAGGACCACG
GGATAAGACCAATCGGACAC
TGGCACACCGAGATGGATTAG
TTAGAGAGACAGAGGACACG
GGATATAGACCATTGGGACAC
TAAAGATACCAATTAGGTTAA
TGGCAGCCAGAGATGGATTAG
TTAGAGTGACCAGAGGCACACG
GGATATAGTCCAATGGGACAC
TAAAATACCAATTAGGATTAA
GGATACAGACCAATGCGACAC
TAATAGATACCAATTAGGATAA
TTAGAGAGACAGAGGTCCACG

Reads
Sequencing data

CLUSTERING

TGGCCACAGAGATGGATTAG
TGGCACACCGAGATGGATTAG
TGGCAGCCAGAGATGGATTAG

TTAGAGAGACAAGAGGACACG
TTAGAGAGACAGAGGACCACG
TTAGAGAGACAGAGGACACG
TTAGAGTGACCAGAGGCACACG
TTAGAGAGACAGAGGTCCACG

GGATTATAGACCAATGGGACAC
GGATAAGACCAATCGGACAC
GGATATAGACCATTGGGACAC
GGATATAGTCCAATGGGACAC
GGATACAGACCAATGCGACAC

GGATGAGACACAGATTAGGAG

TAAAGATACCAATTAGGATAA
TAAGATACCAATTAGGTTAA
TAAATACCAATTAGGATTAA
TATAGATACCAATTAGGATAA

clusters

CONSENSUS

TGGC CA CAGAGATGGATTAG
TGGCACACC GAGATGGATTAG
TGGCAG CCAGAGATGGATTAG
TGGCACACCAGAGATGGATTAG

TTAGAGAGACAAGAGGAC ACG
TTAGAGAGACA GAGGACCACG
TTAGAGAGACA GAGGAC ACG
TTAGAGTGACCAGAGGCACACG
TTAGAGAGACA GAGGTCCACG
TTAGAGAGACAAGAGGTCCACG

GGATTATAGACCAATGGGACAC
GGATAA GA CCAATCGGACAC
GGA ATAGAACCATTGGGACAC
GGATATAGTACCAATGGGACAC
GGATACAGA CCAATGCGACAC
GGATACAGAGCCAATGCGACAC

GGATGAGACACAGATTAGGAG

TAAAGATACCAATTAGGA TAA
TAA GATACCAATTAGG TTAA
TAAA TACCAATTAGGATTAA
TATAGATACCAATTAGGAT AA
TAAAGATACCAATTAGGATTAA

sequences

Consensus

cluster

GGATTATAGACCAATGGGACAC
GGATAAGACCAATCGGACAC
GGAATAGAACCATTGGGACAC
GGATATAGTACCAATGGGACAC
GGATACAGACCAATGCGACAC

consensus

Consensus sequence

GGATACAGAGCCAATGCGACAC

Consensus by multiple alignment

Cluster

```
GGATTAAAGACCAATGGGACAG
GGATAAGACCAATCGGACAG
GGAATAGAACCATTGGGACAC
GGATAAAGTACCAATGGGACAT
GGATAAAGACCAATGCGACAT
```

Multiple alignment algorithms

```
GGATTAAAGACCAATGGGACAG
GGATAA-GA-CCAATCGGACAG
GGA-ATAGAACCATTGGGACAC
GGATAAAGTACCAATGGGACAT
GGATAAAGA-CCAATGCGACAT
```

Majority vote

```
GGAAACAGAGCCAATGCGACAC
```

Consensus using de-Bruijn graph

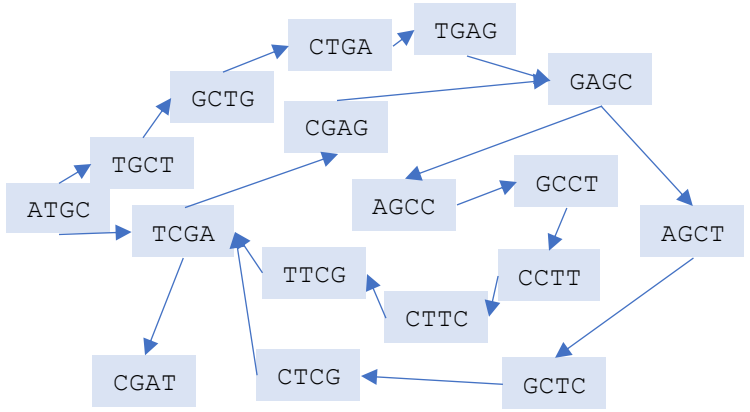
Cluster

GGATTAAAGACCAATGGGACAG
GGATAAGACCAATCGGACAG
GGAATAGAACCATTGGGACAC
GGATAAAGTACCAATGGGACAT
GGATAAAGACCAATGCGACAT

Extract k-mers

AAAGA
AAGAC
AATCG
…

Construct
De-Bruijn graph

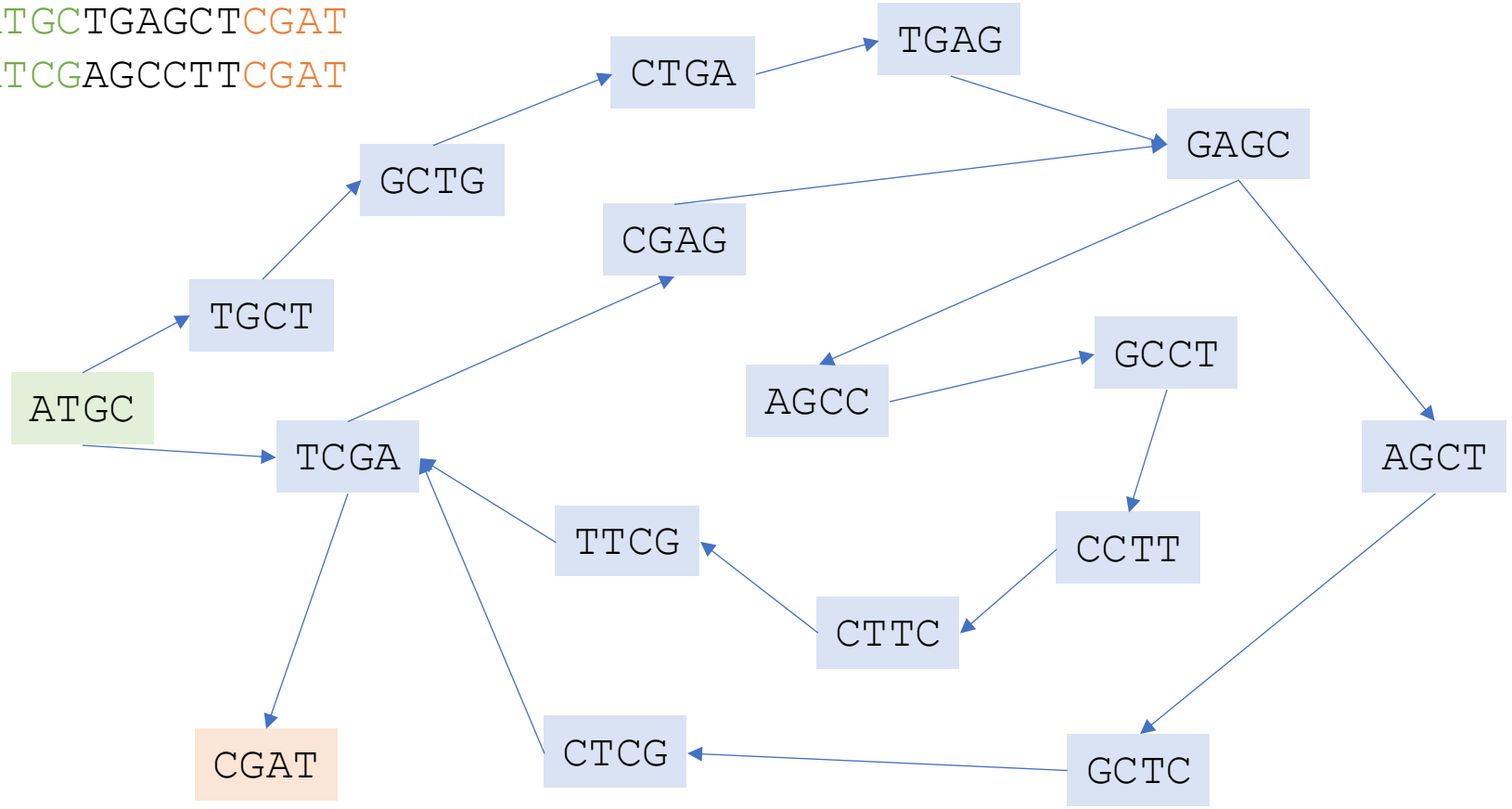


Find path

GGAAACAGAGCCAATGCGACAC

de-Bruijn graph

ATGCTGAGCTCGAT
ATCGAGCCTTCGAT



Kmer size = 4

Consensus using de-Bruijn graph

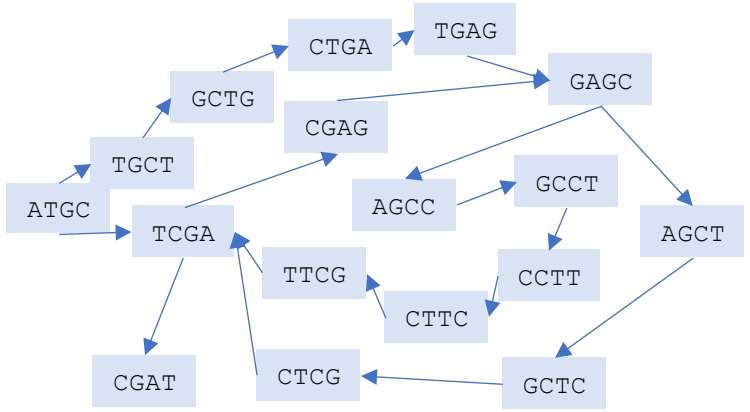
GGATTAAAGACCAATGGGACAC
GGATAAGACCAATCGGACAC
GGAATAGAACCATTGGGACAC
GGATAAAGTACCAATGGGACAT
GGATAAAGACCAATGCGACAT

Extract k-mers

SOLID K-MERS

AAAGA
AAGAC
AATCG
…

Construct
De-Bruijn graph



Find path

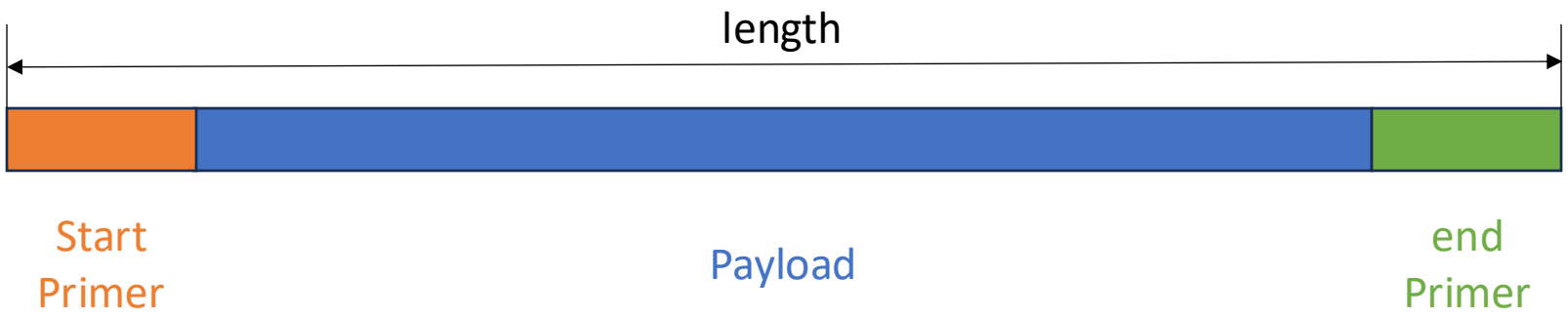
GGAAACAGAGCCAATGCGACAC

Alternative approach

Merge clustering & consensus steps using a global de-Bruijn graph constructed from all sequencing data

Use known information of oligonucleotides

- Start primer / End primer
- Length



Alternative approach

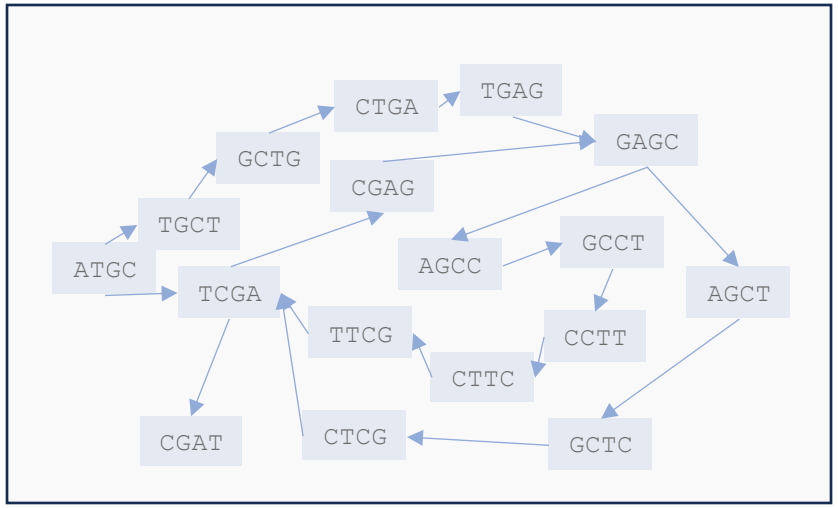
TGGCCACAGAGATGGATTAG
TTAGAGAGACAAGAGGACACG
GGATTATAGACCAATGGGACAC
GGATGAGACACAGATTAGGAG
TAAAAGATACCAATTAGGATAA
TTAGAGAGACAGAGGACCACG
GGATAAGACCAATCGGACAC
TGGCACACCGAGATGGATTAG
TTAGAGAGACAGAGGACACG
GGATATAGACCATTGGGACAC
TAAAGATACCAATTAGGTTAA
TGGCAGCCAGAGATGGATTAG
TTAGAGTGACCAGAGGCACACG
GGATATAGTCCAATGGGACAC
TAAAATACCAATTAGGATTAA
GGATACAGACCAATGCGACAC
TAATAGATACCAATTAGGATAA
TTAGAGAGACAGAGGTCCACG

Reads

Sequencing data

Construct
de-Bruijn graph

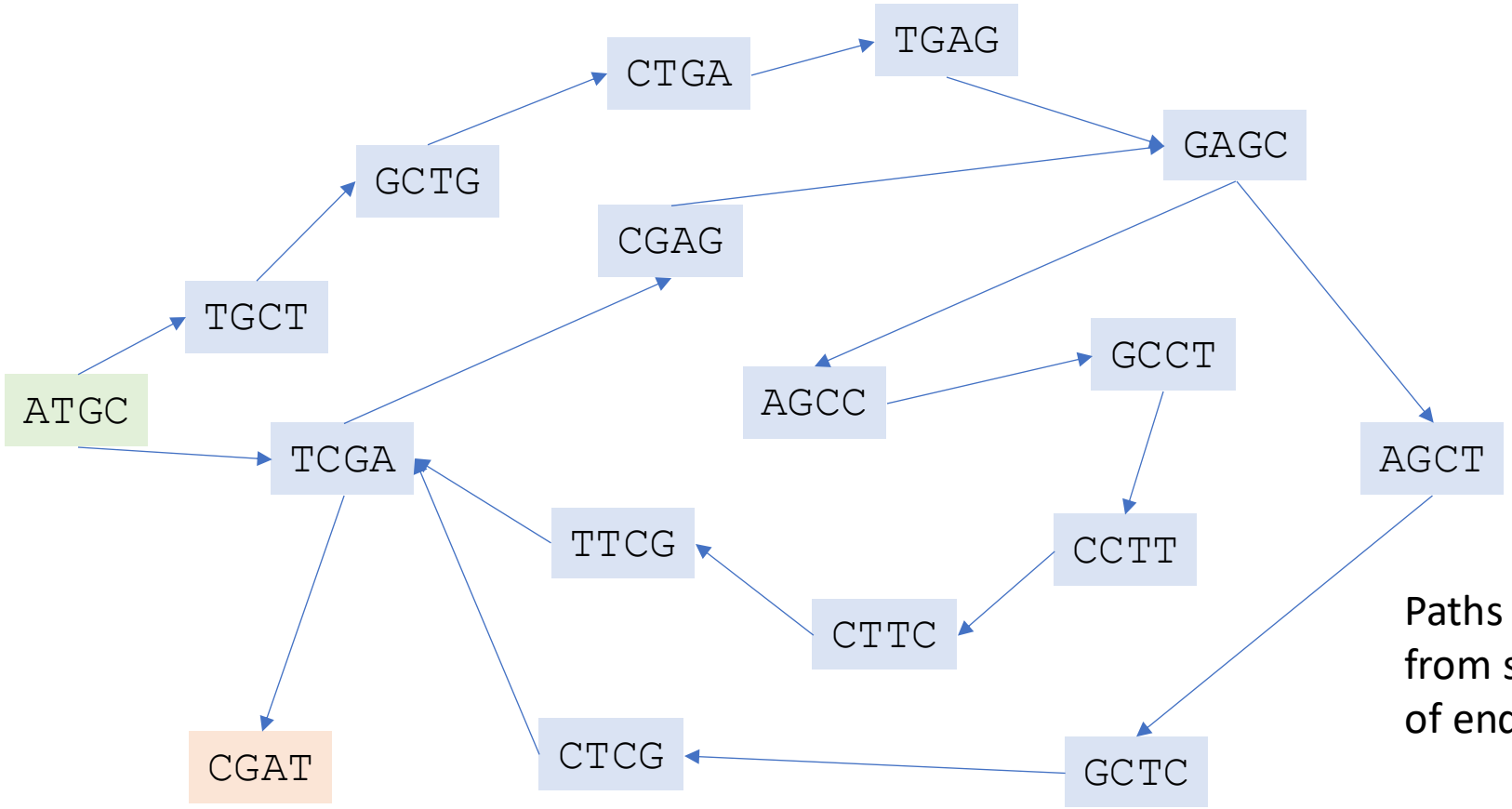
K-mer size = 15



TGGCACACCAGAGATGGATTAG
TTAGAGAGACAAGAGGTCCACG
GGATACAGAGCCAATGCGACAC
TAAAGATACCAATTAGGATTAA

Extract
sequences

Extract sequences

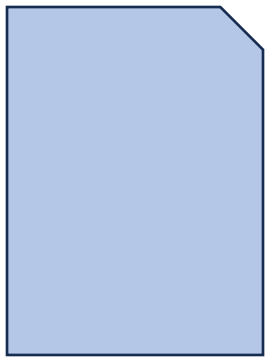


ATGCTGAGCTCGAT
ATCGAGCCTTCGAT

Paths of length N starting with k-mer from start primer and ending with k-mer of end primers are potential sequences

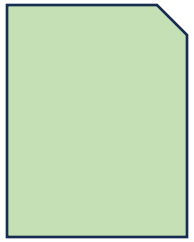
Scalability

Complex graph: too many cycles → extraction process difficult
- long execution time
- many false positives



$10^7 – 10^{10}$ reads

De-Bruijn graph



sequences

Partitioning

$10^7 – 10^{10}$ reads

~ 10^4 reads

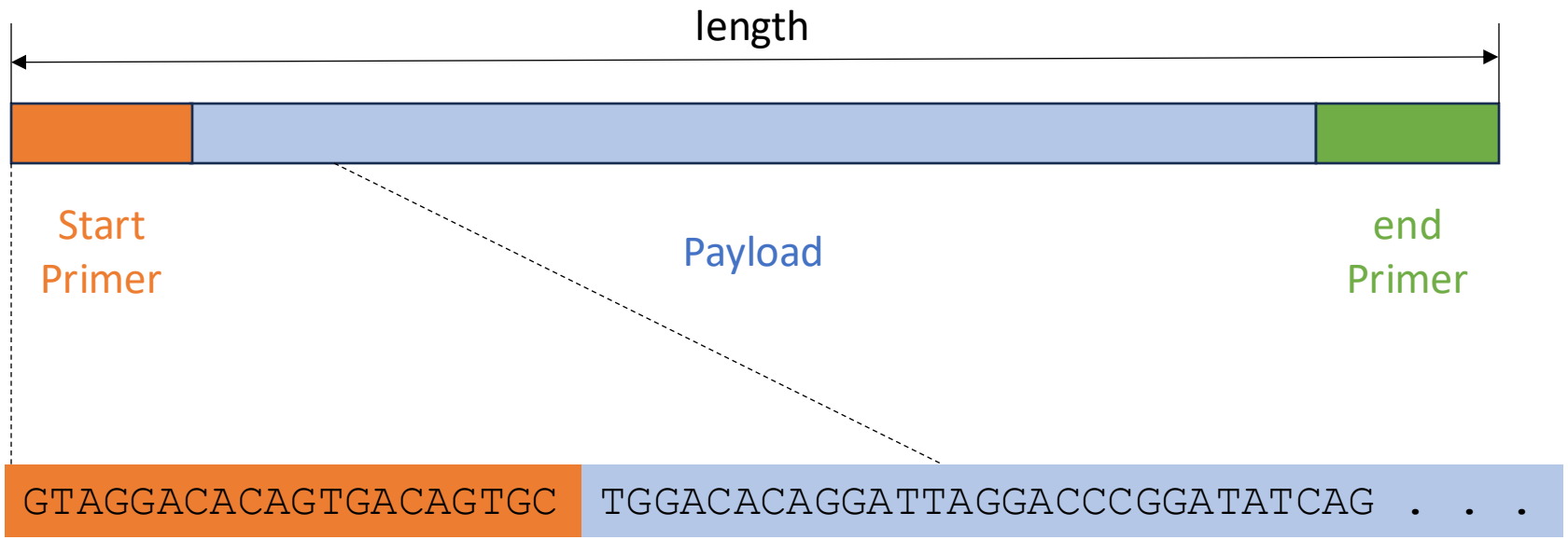
de-Bruijn graph → sequences

de-Bruijn graph → sequences

de-Bruijn graph → sequences

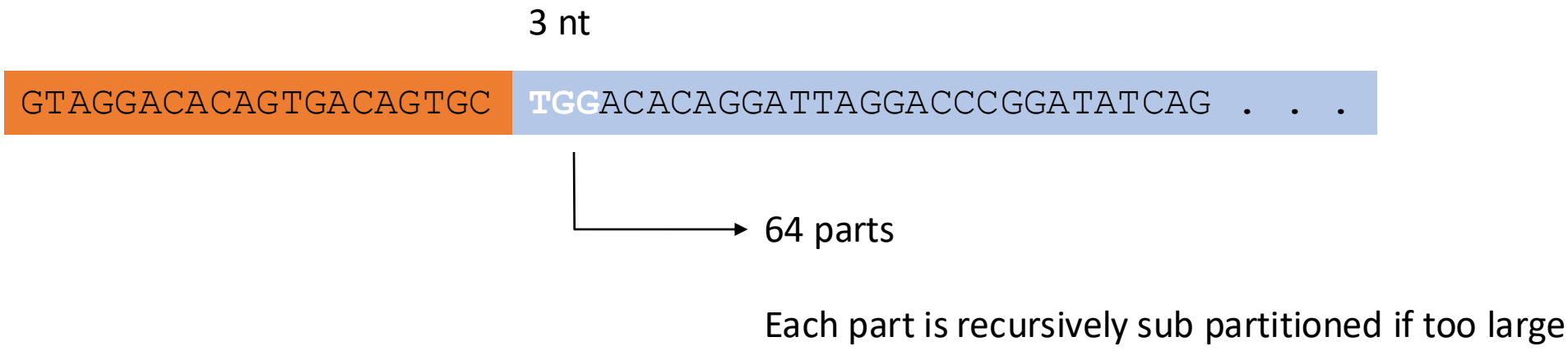
Partitioning

Use start primer as reference



First nucleotides of the payloads → used as index for partition

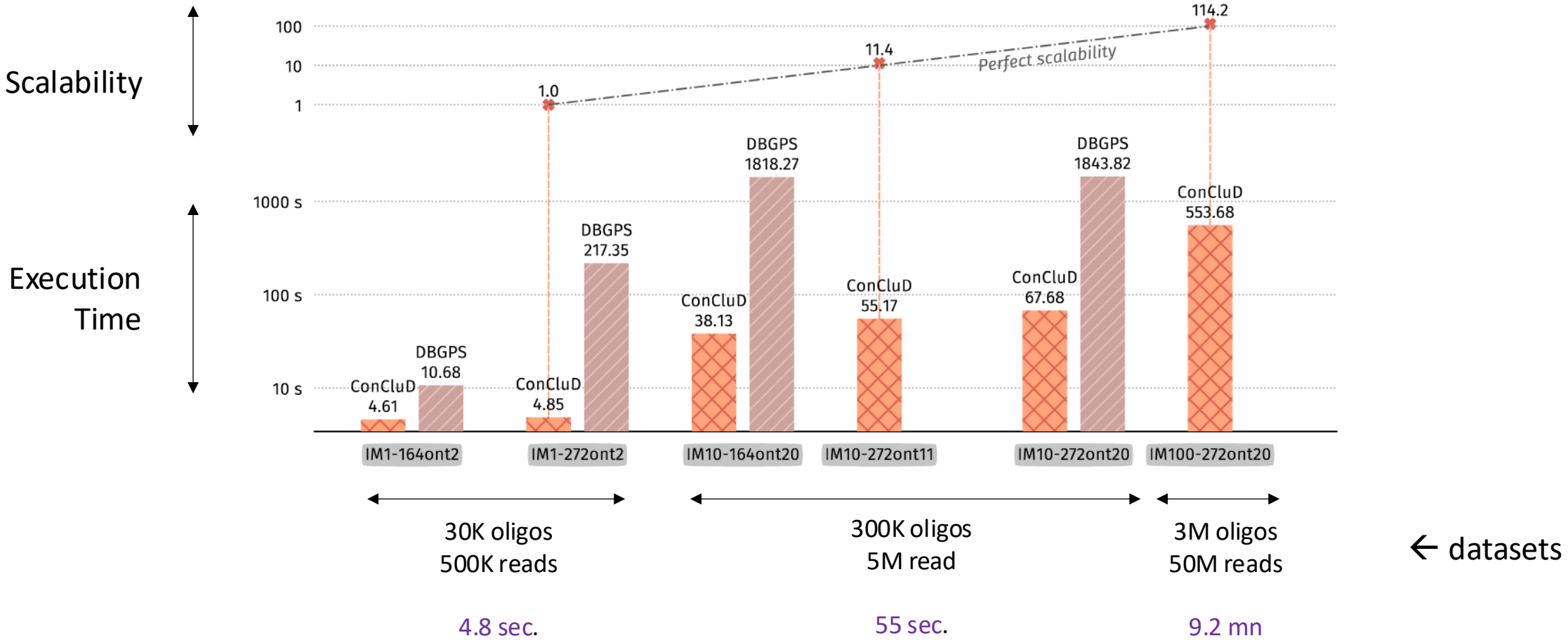
Partitioning strategy



Advantages

- Very fast process
- Size of the de-Bruijn graphs fully controlled
- Well suited for parallelization

Implementation: ConCluD



Conclusion

Sequencing data processing

- Needed to clean the data before the decoding step
- Use redundancy of sequencing devices
- Cannot recover missing oligonucleotides
- Correct sequencing errors
- Better the cleaning, lighter the error correcting codes
- Algorithms adapted from genomics
  - Known meta data from coding strategy can be used

Perspectives

Omit/simplify the clustering/consensus step

- Quality of sequencing is increasing

Merge sequencing data process and decoding stage

- Optimization of path search in the de-Bruijn graph

  → Design codes that:

    - Allow in-line sequence verification
    - Generate set of sequences with huge Hamming distance between them