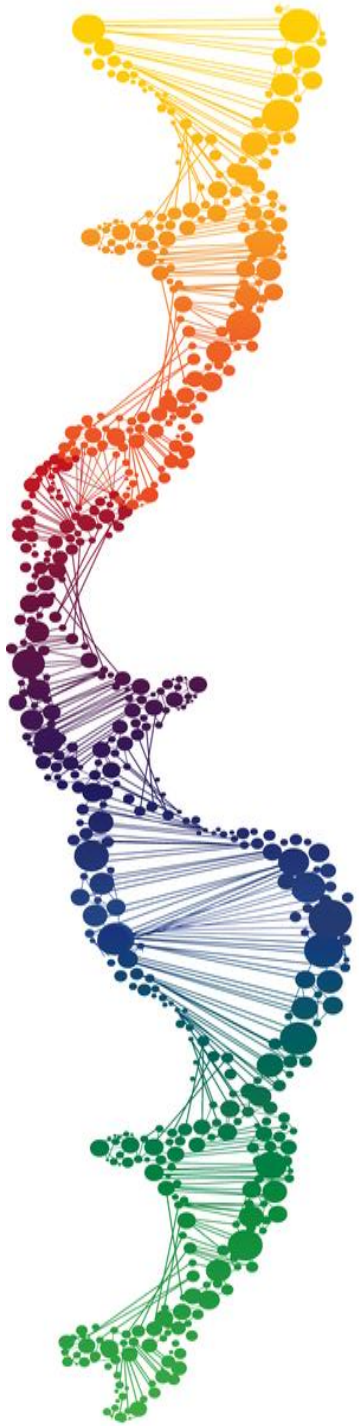
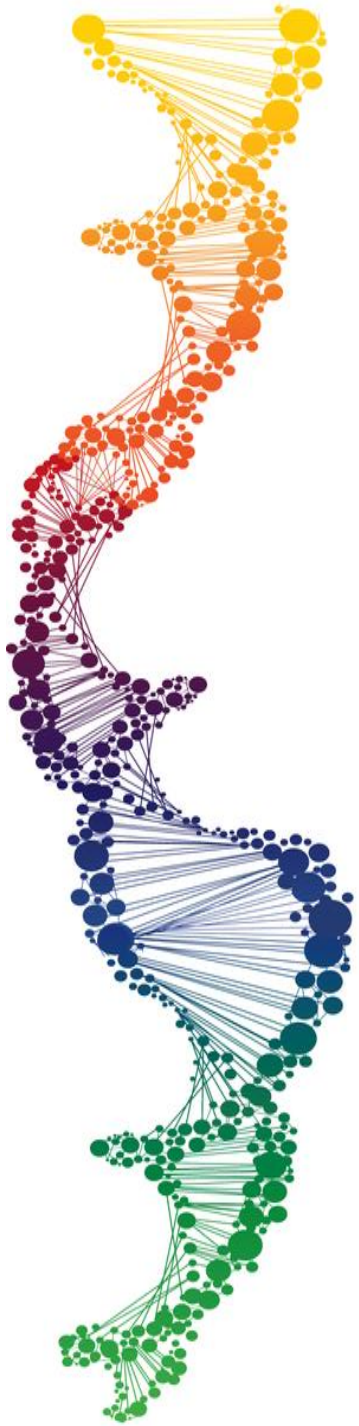


Algorithms for Third-Generation Sequencing Reads Mapping

Bioinformatics

Emanuele Parisi

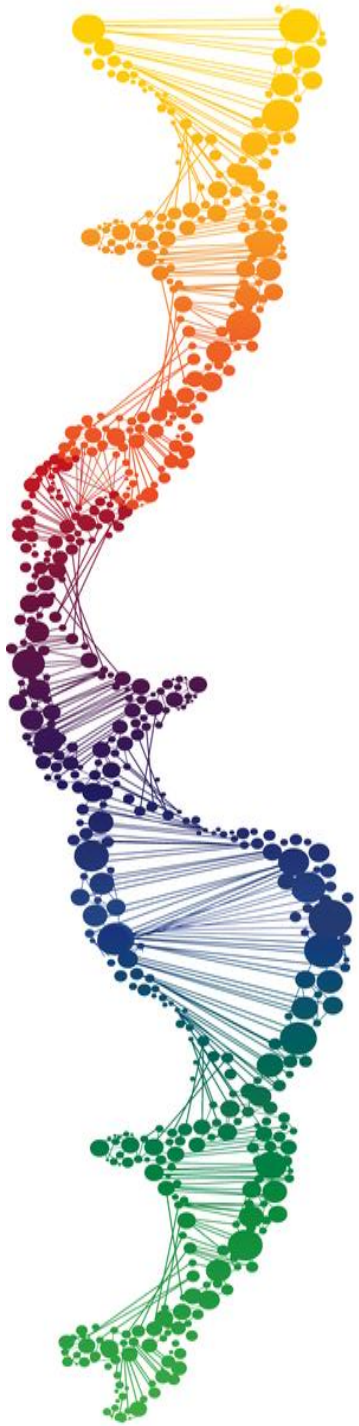


Third-Generation Sequencing recap.

- Third-Generation Sequencing aims at overcoming the limitations of Next-Generation Sequencing
- Two main technologies available:
  - SMRT sequencing
  - Nanopore sequencing

PACIFIC BIOSCIENCES®

Oxford NANOPORE Technologies



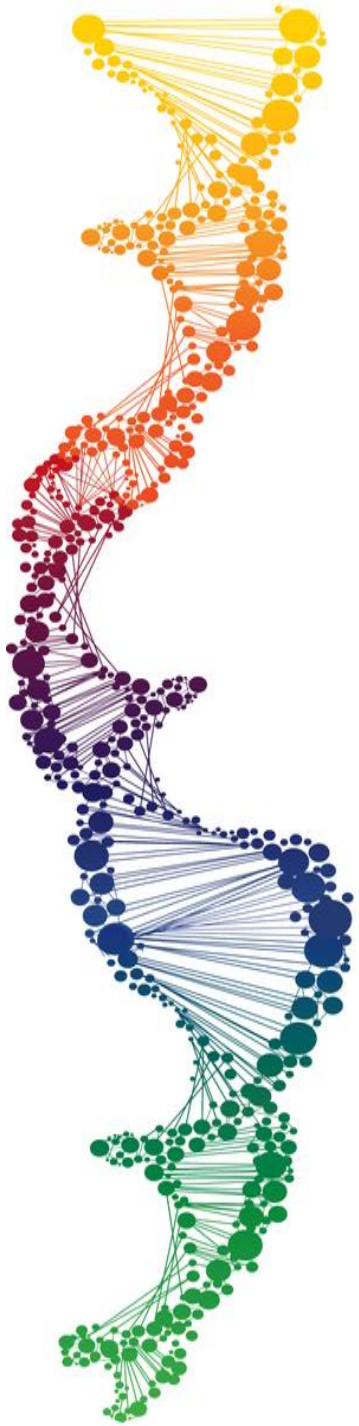
Third-Generation Sequencing recap.

- Long-Reads have different properties than short one
  - Higher error-rate
  - Higher length
  - Different length distribution

Third-Generation Sequencing recap.

Read type	Error rate (Proportion of overall error) (%)			
	Overall	Insertion	Deletion	Mismatch
PacBio CCS	1.72	0.087 (5.06)	0.34 (19.48)	1.30 (**75.46**)
PacBio subread	14.20	5.92 (**41.71**)	3.01 (21.17)	5.27 (37.12)
ONT 2D	13.40	3.12 (23.30)	4.79 (35.70)	5.50 (**40.99**)
ONT 1D	20.19	2.93 (14.51)	7.52 (37.24)	9.74 (**48.25**)

- Next-Generation Sequencing: ~1% error rate
- Third-Generation Sequencing: ~15-20% error rate



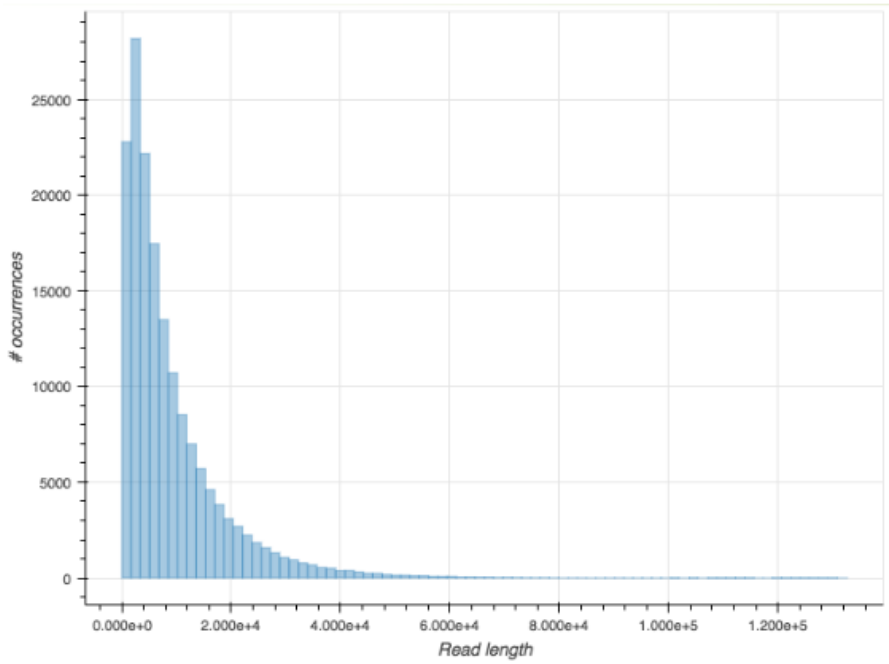
Third-Generation Sequencing recap.

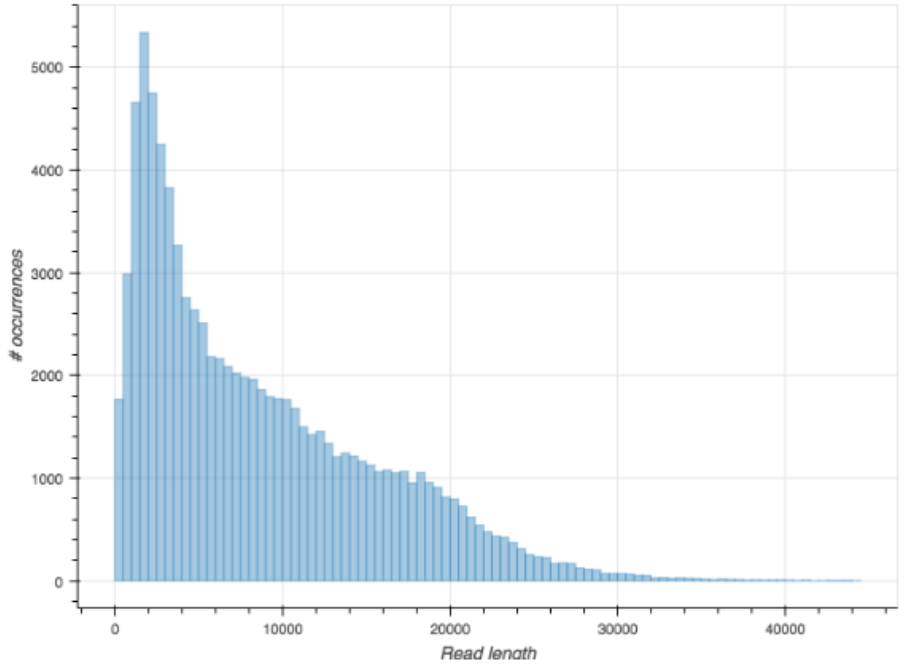
Read type	Error rate (Proportion of overall error) (%)			
	Overall	Insertion	Deletion	Mismatch
PacBio CCS	1.72	0.087 (5.06)	0.34 (19.48)	1.30 (**75.46**)
PacBio subread	14.20	5.92 (**41.71**)	3.01 (21.17)	5.27 (37.12)
ONT 2D	13.40	3.12 (23.30)	4.79 (35.70)	5.50 (**40.99**)
ONT 1D	20.19	2.93 (14.51)	7.52 (37.24)	9.74 (**48.25**)

- Next-Generation Sequencing: ~1% error rate
- Third-Generation Sequencing: ~15-20% error rate

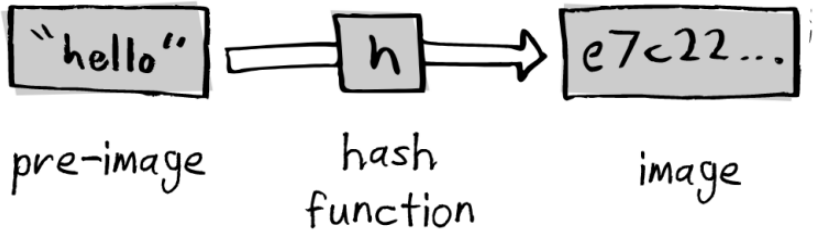
Third-Generation Sequencing recap.

- Read length distribution





Some vocabulary…

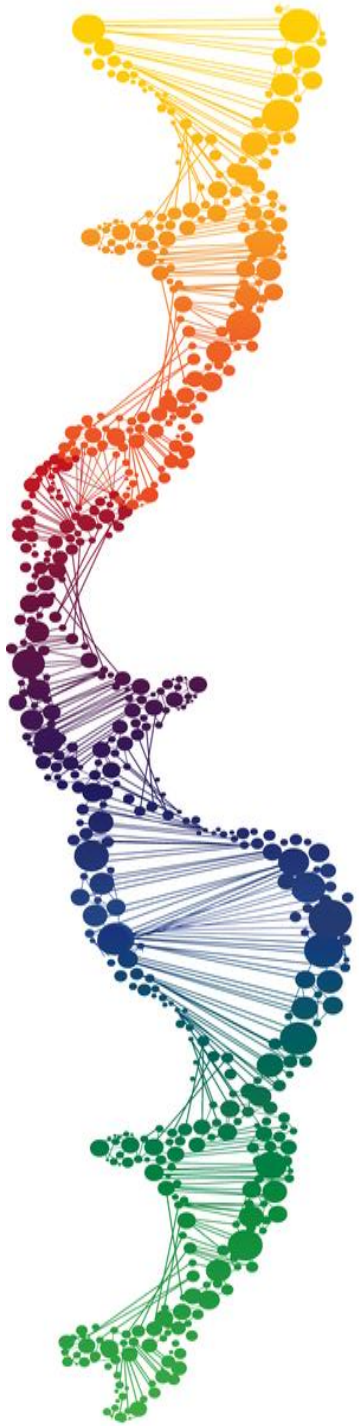


Hash: numerical representation of an object

***K*-mer**: sub-string of length *k*

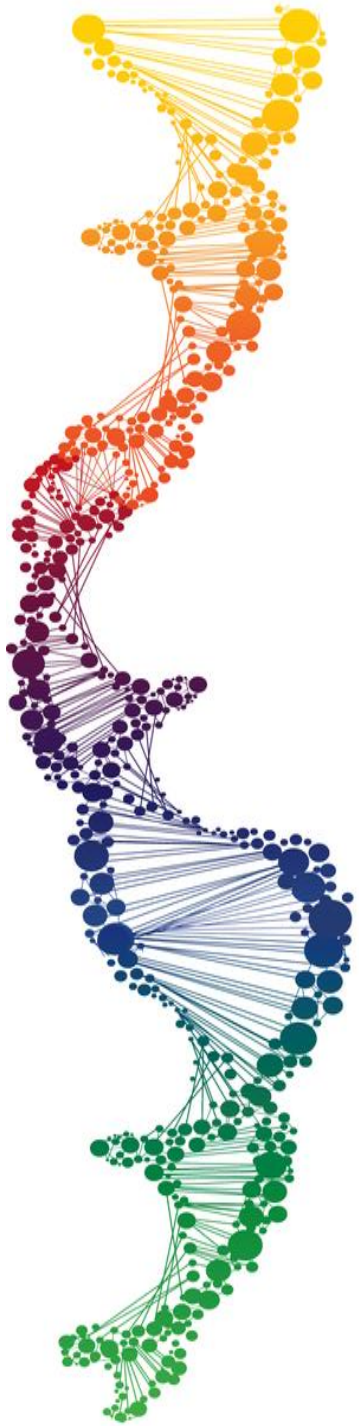
Window: a set of successive *k*-mers

```
AACGGCATTCGGCA
AAC GCA TCG
 ACG CAT CGG
  CGG ATT GGC
   GGC TTC GCA
```



Introduction

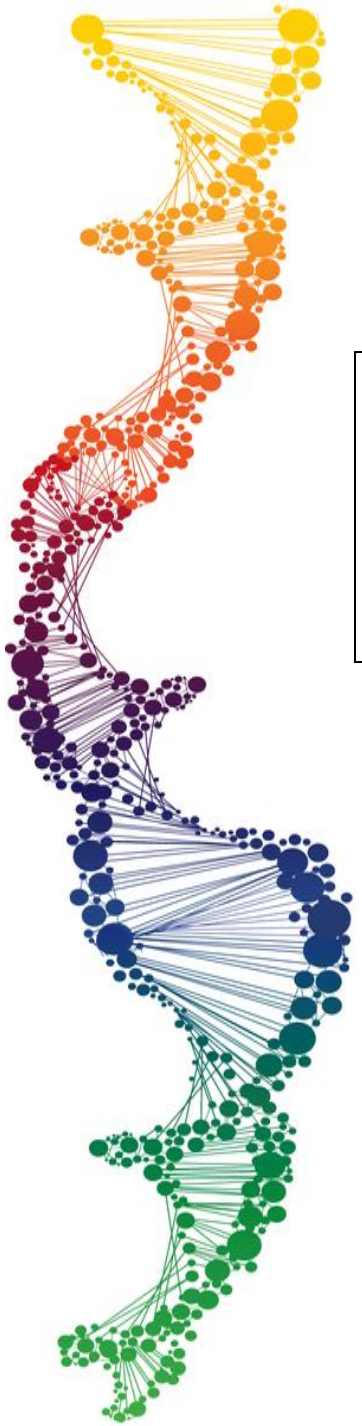
- Long-reads may be challenging to handle as they are
- Reduce data dimensionality by:
  - *K*-mer frequency array
  - *K*-mer selection procedures
- Evaluate sequence similarity looking at the reduced representation

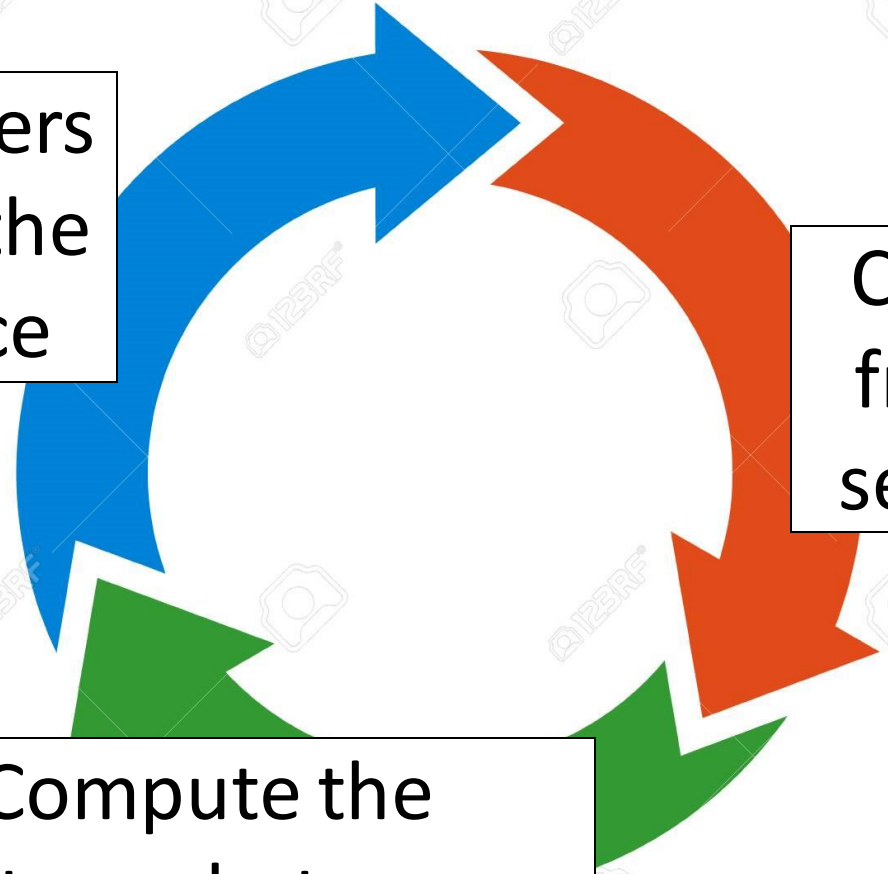


K-mer frequency arrays

- Infer sequences similarities by looking at their *k*-mer distribution
- Define a distance metric over frequency arrays
  - Euclidean distance, cosine distance…
- Easy to compute and to store for small *k*

K-mer frequency arrays



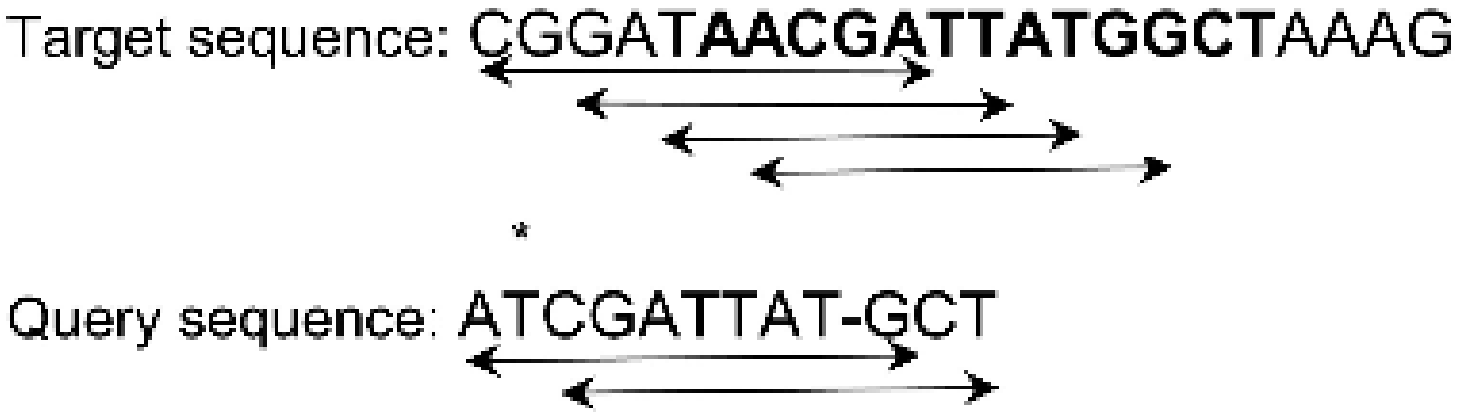


Compute *k*-mers frequency in the first sequence

Compute *k*-mers frequency in the second sequence

Compute the distance between the frequency arrays

An example…




```
CGGATAACGATTATGGCTAAAG
|*|||||||  |||
ATCGATTAT-GCT
```

An example…

Unique k-mers (k = 2)	Number of k-mer occurrences in windows starting at offsets j	
	j = 0	j = 2
AA	0	0
CA	0	0
GA	1	1
TA	1	1
AC	0	0
CC	0	0
GC	0	1
TC	1	0
AG	0	0
CG	1	1
GG	0	0
TG	1	1
AT	3	2
CT	0	1
GT	0	0
TT	1	1

An example…

Unique k-mers (k = 2)	Number of k-mer occurrences in windows starting at offsets i						
	i = 0	i = 2	i = 4	i = 6	i = 8	i = 10	i = 12
AA	1	1	1	0	0	1	2
CA	0	0	0	0	0	0	0
GA	2	2	1	1	1	0	0
TA	1	1	2	1	1	2	1
AC	1	1	1	1	0	0	0
CC	0	0	0	0	0	0	0
GC	0	0	0	0	1	1	1
TC	0	0	0	0	0	0	0
AG	0	0	0	0	0	0	1
CG	2	1	1	1	0	0	0
GG	1	0	0	1	1	1	1
TG	0	0	0	1	1	1	1
AT	1	2	2	2	2	1	1
CT	0	0	0	0	1	1	1
GT	0	0	0	0	0	0	0
TT	0	1	1	1	1	1	0

Target sequence: CGGAT**AACGATTATGGCT**AAAG

An example…

Unique k-mers (k = 2)	Number of k-mer occurrences in windows starting at offsets i						
	i = 0	i = 2	i = 4	i = 6	i = 8	i = 10	i = 12
AA	1	1	1	0	0	1	2
CA	0	0	0	0	0	0	0
GA	2	2	1	1	1	0	0
TA	1	1	2	1	1	2	1
AC	1	1	1	1	0	0	0
CC	0	0	0	0	0	0	0
GC	0	0	0	0	1	1	1
TC	0	0	0	0	0	0	0
AG	0	0	0	0	0	0	1
CG	2	1	1	1	0	0	0
GG	1	0	0	1	1	1	1
TG	0	0	0	1	1	1	1
AT	1	2	2	2	2	1	1
CT	0	0	0	0	1	1	1
GT	0	0	0	0	0	0	0
TT	0	1	1	1	1	1	0

Unique k-mers (k = 2)	Number of k-mer occurrences in windows starting at offsets j	
	j = 0	j = 2
AA	0	0
CA	0	0
GA	1	1
TA	1	1
AC	0	0
CC	0	0
GC	0	1
TC	1	0
AG	0	0
CG	1	1
GG	0	0
TG	1	1
AT	3	2
CT	0	1
GT	0	0
TT	1	1

Euclidean distance between pair of vectors in query and target sequence		
target\query offset	j = 0	j = 2
i = 0	3.46	3.16
i = 2	2.45	2.45
i = 4	2.45	2.45
i = 6	**2.00**	2.00
i = 8	2.45	**1.41**
i = 10	3.46	2.45
i = 12	4.00	3.16

COSINE mapping tool

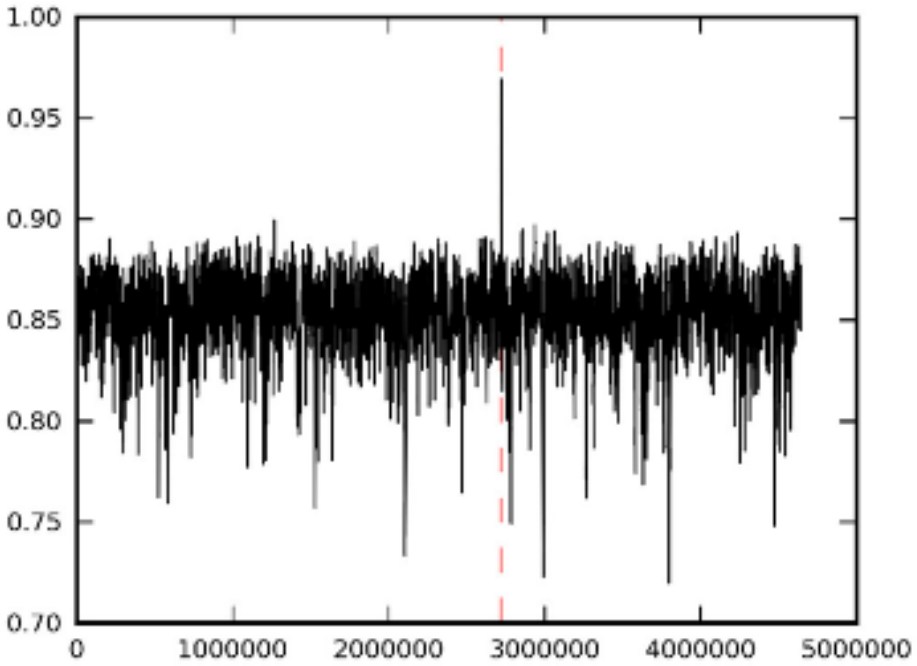
COSINE: non-seeding method for mapping long noisy sequences

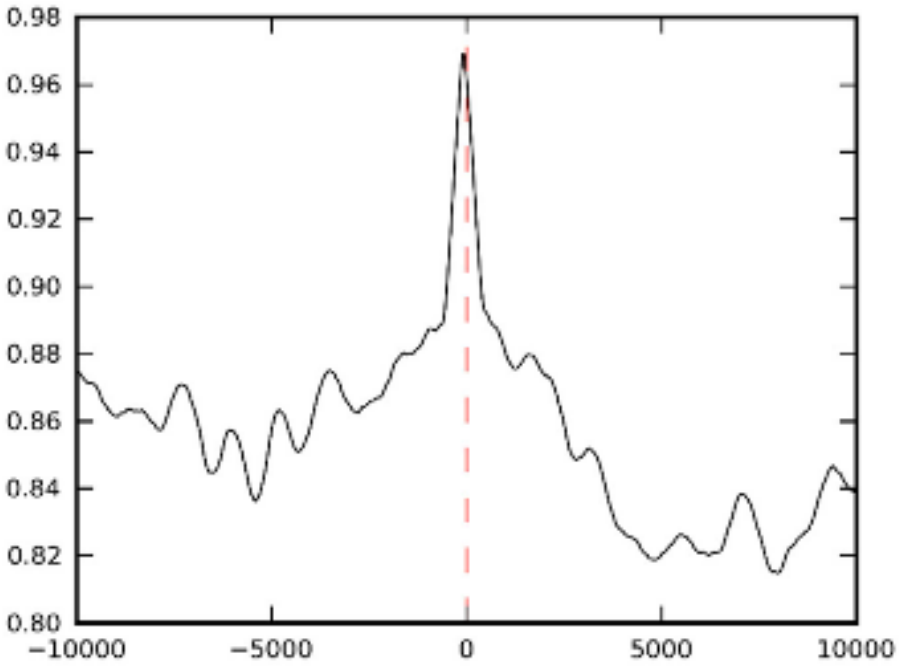
Pegah Tootoonchi Afshar[1] and Wing Hung Wong[2,*]

[1]Department of Electrical Engineering, School of Engineering, Stanford University, Stanford, CA 94305, USA and
[2]Department of Statistics and Department of Biomedical Data Science, Stanford University, Stanford, CA 94305, USA

COSINE computes the context similarity of two stretches of nucleobases given the similarity over distributions of their short *k*-mers ($k = 3$–4) along the sequences.

COSINE mapping tool





The *Winnowing* algorithm

Winnowing: Local Algorithms for Document Fingerprinting

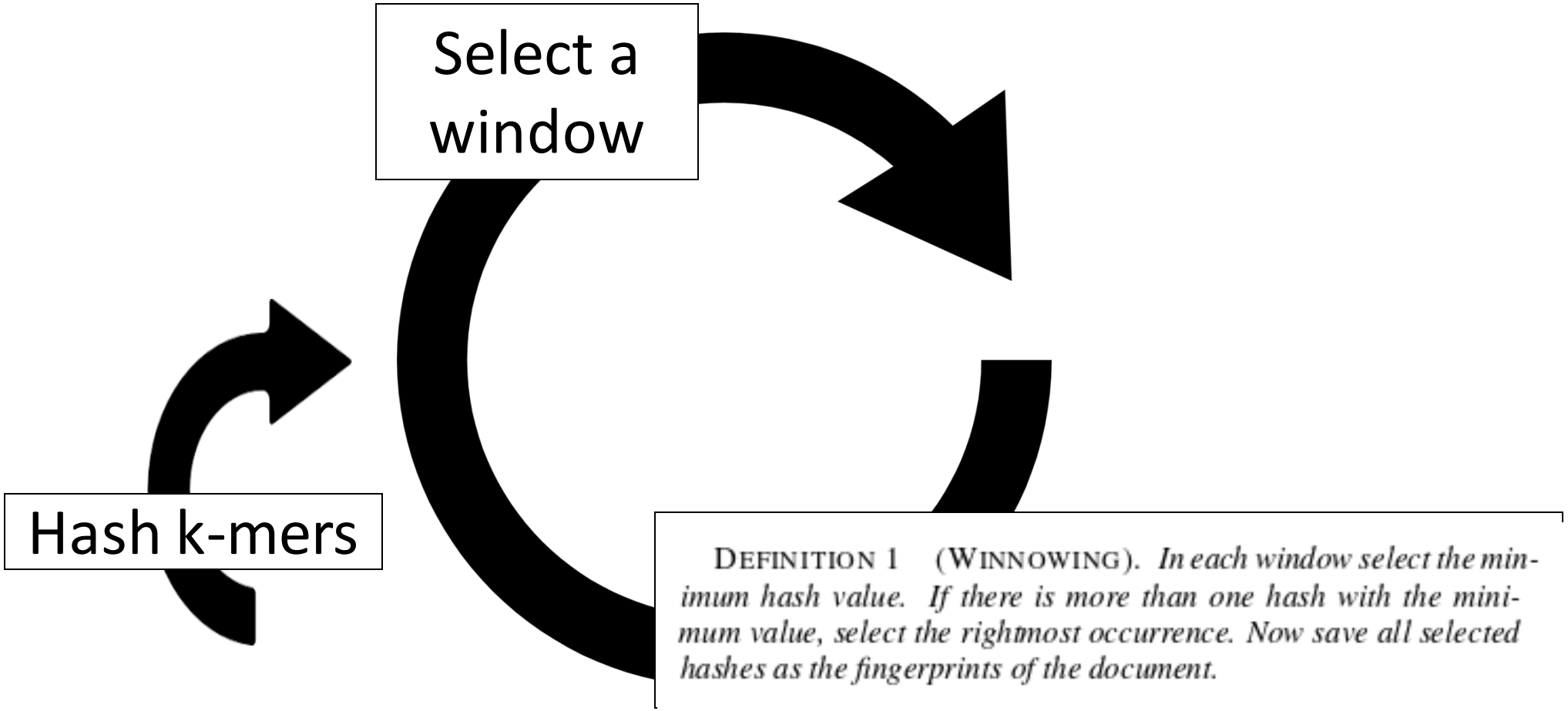
Saul Schleimer
MSCS
University of Illinois, Chicago
saul@math.uic.edu

Daniel S. Wilkerson
Computer Science Division
UC Berkeley
dsw@cs.berkeley.edu

Alex Aiken
Computer Science Division
UC Berkeley
aiken@cs.berkeley.edu

- Pre-processing step for text similarity detection
- Define a procedure for extracting locally-related features from the sequence

The *Winnowing* algorithm



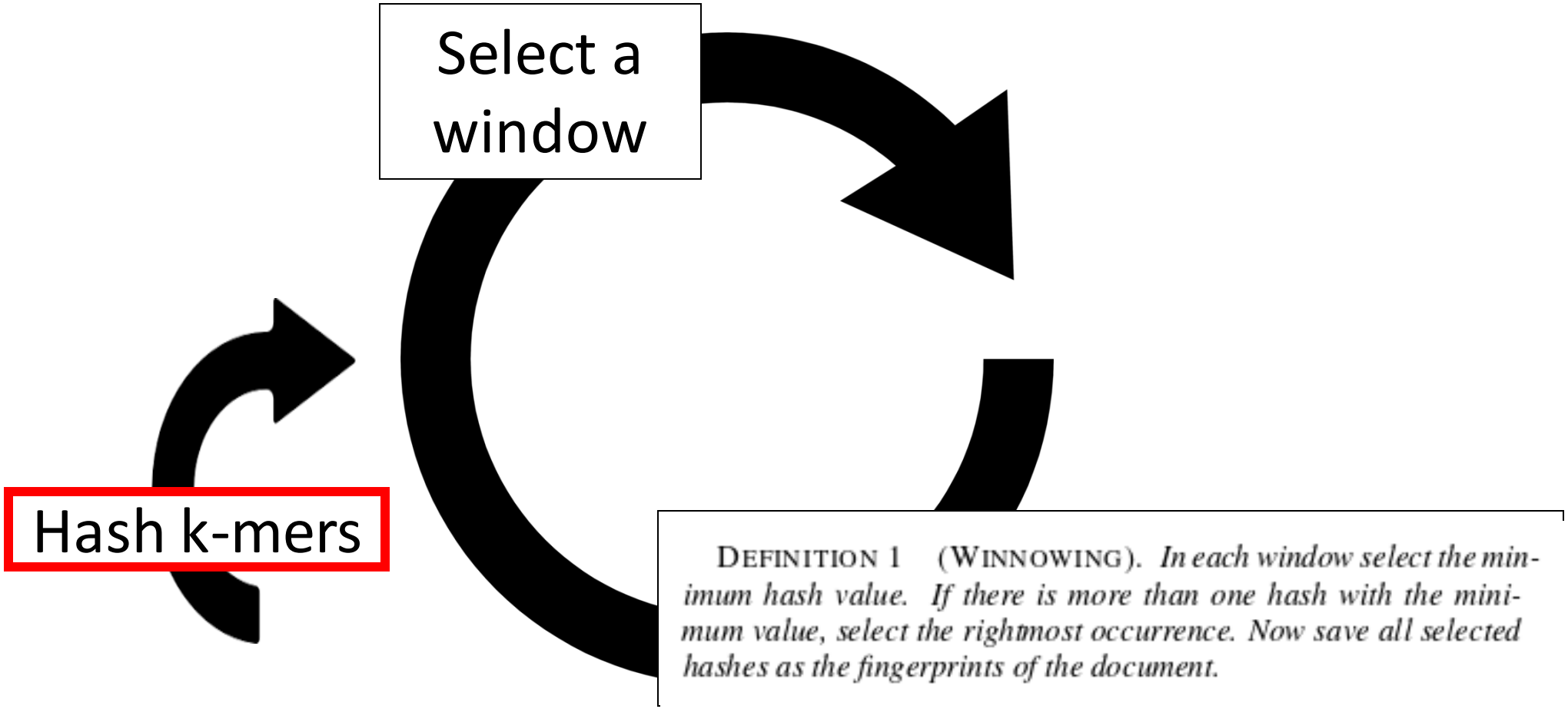
DEFINITION 1 (WINNOWING). *In each window select the minimum hash value. If there is more than one hash with the minimum value, select the rightmost occurrence. Now save all selected hashes as the fingerprints of the document.*

The *Winnowing* algorithm

Compute the Winnowing-features of a sequence, using k-mers of size 3 and windows of size 5

A C G A T C A T C T A T C G G C T T T T

The *Winnowing* algorithm



DEFINITION 1 (WINNOWING). *In each window select the minimum hash value. If there is more than one hash with the minimum value, select the rightmost occurrence. Now save all selected hashes as the fingerprints of the document.*

The *Winnowing* algorithm

Compute the Winnowing-features of a sequence, using k-mers of size 3 and windows of size 5

```
A C G A T C A T C T A T C G G C T T T T
 ACG
```

The *Winnowing* algorithm

Compute the Winnowing-features of a sequence, using k-mers of size 3 and windows of size 5

```
A C G A T C A T C T A T C G G C T T T T
 ACG
   CGA
```

The *Winnowing* algorithm

Compute the Winnowing-features of a sequence, using k-mers of size 3 and windows of size 5

```
A C G A T C A T C T A T C G G C T T T T
 ACG
   CGA
      GAT
```

The *Winnowing* algorithm

Compute the Winnowing-features of a sequence, using k-mers of size 3 and windows of size 5

```
A C G A T C A T C T A T C G G C T T T T
 ACG         ATC         CGG
   CGA         TCT         GGC
     GAT         CTA         GCT
       ATC         TAT         CTT
         TCA         ATC         TTT
           CAT         TCG         TTT
```

The *Winnowing* algorithm

Compute the Winnowing-features of a sequence, using k-mers of size 3 and windows of size 5

```
A C G A T C A T C T A T C G G C T T T T
 132          ATC          CGG
   CGA          TCT          GGC
     GAT          CTA          GCT
       ATC          TAT          CTT
         TCA          ATC          TTT
           CAT          TCG          TTT
```

The *Winnowing* algorithm

Compute the Winnowing-features of a sequence, using k-mers of size 3 and windows of size 5

```
A C G A T C A T C T A T C G G C T T T T
 132         ATC         CGG
   634         TCT         GGC
     GAT         CTA         GCT
       ATC         TAT         CTT
         TCA         ATC         TTT
           CAT         TCG         TTT
```

The *Winnowing* algorithm

Compute the Winnowing-features of a sequence, using k-mers of size 3 and windows of size 5

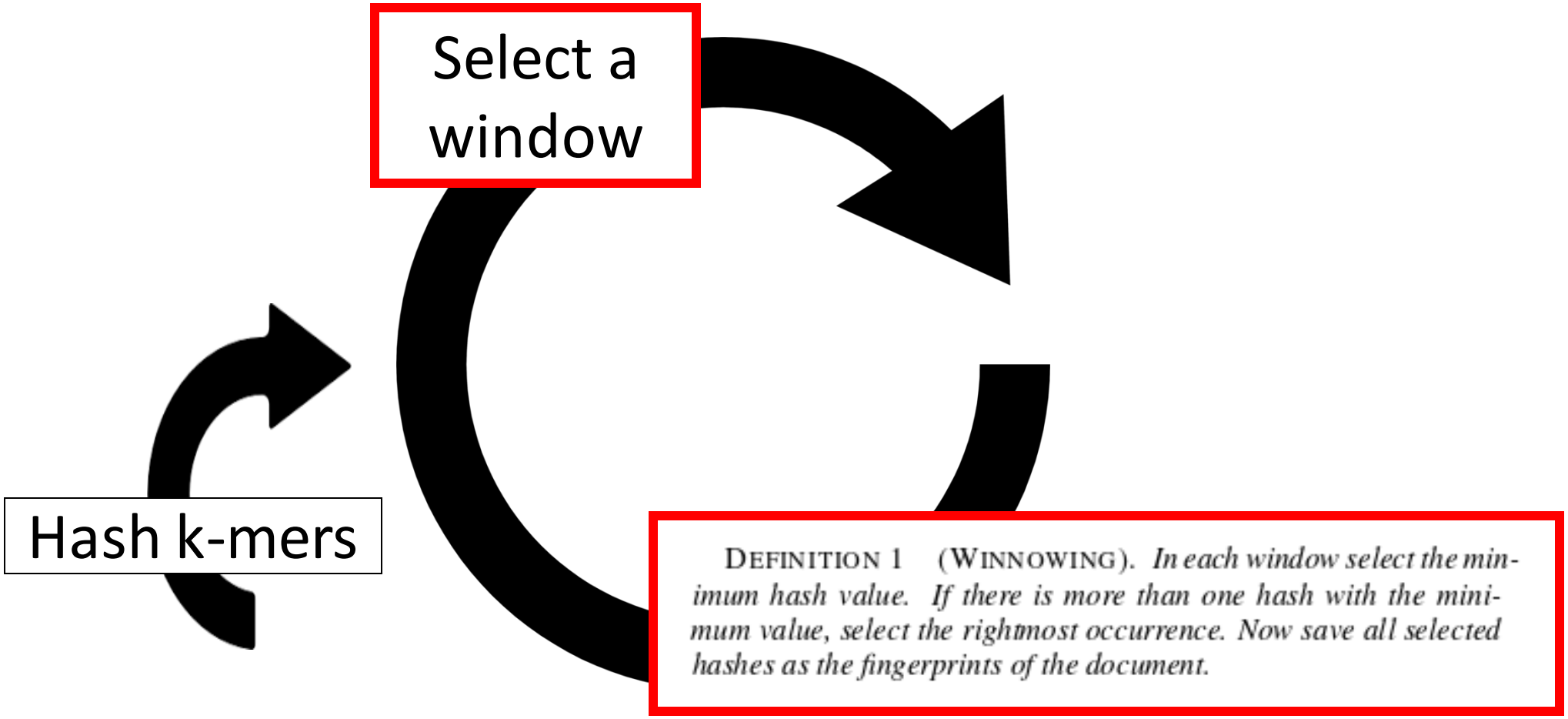
```
A C G A T C A T C T A T C G G C T T T T
 132         ATC         CGG
   634         TCT         GGC
     735         CTA         GCT
       ATC         TAT         CTT
         TCA         ATC         TTT
           CAT         TCG         TTT
```

The *Winnowing* algorithm

Compute the Winnowing-features of a sequence, using k-mers of size 3 and windows of size 5

```
A C G A T C A T C T A T C G G C T T T T
 132         034         111
   634         284         110
     735         333         534
       034         121         221
         373         034         046
           199         840         046
```

The *Winnowing* algorithm



The *Winnowing* algorithm

Compute the Winnowing-features of a sequence, using k-mers of size 3 and windows of size 5

```
A C G A T C A T C T A T C G G C T T T T
 132         034         111
   634         284         110
     735         333         534
       034         121         221
         373         034         046
           199         840         046
```

The *Winnowing* algorithm

Compute the Winnowing-features of a sequence, using k-mers of size 3 and windows of size 5

```
A C G A T C A T C T A T C G G C T T T T
132         034         111
  634         284         110
    735         333         534
      034         121         221
        373         034         046
          199         840         046
```

The *Winnowing* algorithm

Compute the Winnowing-features of a sequence, using k-mers of size 3 and windows of size 5

```
A C G A T C A T C T A T C G G C T T T T
132         034         111
  634         284         110
    735         333         534
      034         121         221
        373         034         046
          199         840         046
```

The *Winnowing* algorithm

Compute the Winnowing-features of a sequence, using k-mers of size 3 and windows of size 5

```
A C G A T C A T C T A T C G G C T T T T
 132         034         111
   634         284         110
     735         333         534
       034         121         221
         373         034         046
           199         840         046
```

The *Winnowing* algorithm

Compute the Winnowing-features of a sequence, using k-mers of size 3 and windows of size 5

```
A C G A T C A T C T A T C G G C T T T T
 132         034         111
   634         284         110
     735         333         534
       034         121         221
         373         034         046
           199         840         046
```

The *Winnowing* algorithm

Compute the Winnowing-features of a sequence, using k-mers of size 3 and windows of size 5

```
A C G A T C A T C T A T C G G C T T T T
132         034         111
  634         284         110
    735         333         534
      034         121         221
        373         034         046
          199         840         046
```

The *Winnowing* algorithm

Compute the Winnowing-features of a sequence, using k-mers of size 3 and windows of size 5

```
A C G A T C A T C T A T C G G C T T T T
 132         034         111
   634         284         110
     735         333         534
       034         121         221
         373         034         046
           199         840         046
```

The *Winnowing* algorithm

Compute the Winnowing-features of a sequence, using k-mers of size 3 and windows of size 5

```
A C G A T C A T C T A T C G G C T T T T
132         034         111
  634         284         110
    735         333         534
      034         121         221
        373         034         046
          199         840         046
```

The *Winnowing* algorithm

Compute the Winnowing-features of a sequence, using k-mers of size 3 and windows of size 5

```
A C G A T C A T C T A T C G G C T T T T
 132         034         111
   634         284         110
     735         333         534
       034         121         221
         373         034         046
           199         840         046
```

The *Winnowing* algorithm

Compute the Winnowing-features of a sequence, using k-mers of size 3 and windows of size 5

```
A C G A T C A T C T A T C G G C T T T T
132         034         111
  634         284         110
    735         333         534
      034         121         221
        373         034         046
          199         840         046
```

The *Winnowing* algorithm

Compute the Winnowing-features of a sequence, using k-mers of size 3 and windows of size 5

```
A C G A T C A T C T A T C G G C T T T T
 132         034         111
   634         284         110
     735         333         534
       034         121         221
         373         034         046
           199         840         046
```

The *Winnowing* algorithm

Compute the Winnowing-features of a sequence, using k-mers of size 3 and windows of size 5

```
A C G A T C A T C T A T C G G C T T T T
 132         034         111
   634         284         110
     735         333         534
       034         121         221
         373         034         046
           199         840         046
```

The *Winnowing* algorithm

Compute the Winnowing-features of a sequence, using k-mers of size 3 and windows of size 5

```
A C G A T C A T C T A T C G G C T T T T
 132         034         111
   634         284         110
     735         333         534
       034         121         221
         373         034         046
           199         840         046
```

The *Winnowing* algorithm

Compute the Winnowing-features of a sequence, using k-mers of size 3 and windows of size 5

```
A C G A T C A T C T A T C G G C T T T T
 132         034         111
   634         284         110
     735         333         534
       034         121         221
         373         034         046
           199         840         046
```

The *Winnowing* algorithm

Compute the Winnowing-features of a sequence, using k-mers of size 3 and windows of size 5

```
A C G A T C A T C T A T C G G C T T T T
132         034         111
  634         284         110
    735         333         534
      034         121         221
        373         034         046
          199         840         046
```

The *Winnowing* algorithm

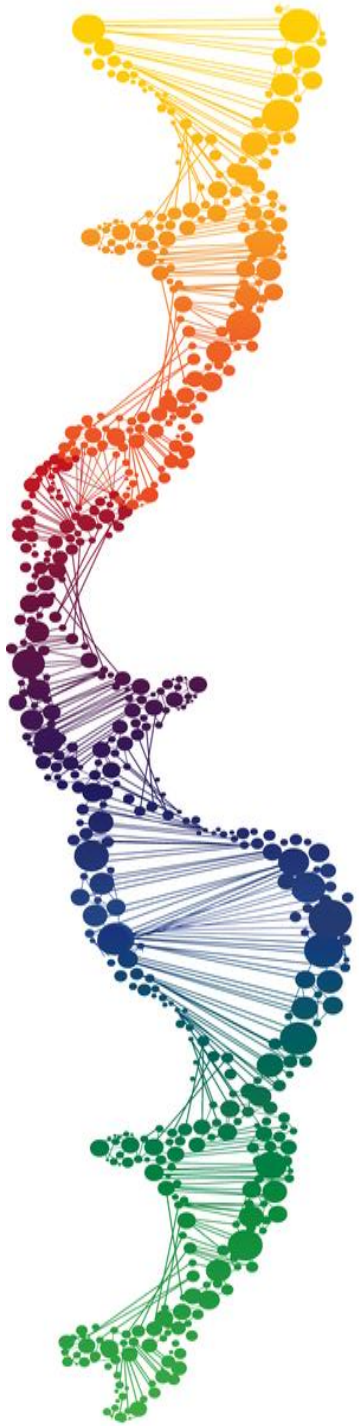
Compute the Winnowing-features of a sequence, using k-mers of size 3 and windows of size 5

```
A C G A T C A T C T A T C G G C T T T T
 132         034         111
   634         284         110
     735         333         534
       034         121         221
         373         034         046
           199         840         046
```

The *Winnowing* algorithm

Compute the Winnowing-features of a sequence, using k-mers of size 3 and windows of size 5

```
A C G A T C A T C T A T C G G C T T T T
 132         034         111
   634         284         110
     735         333         534
       034         121         221
         373         034         046
           199         840         046
```



Treating the *Winnowing* features

- Chaining
  - Detect similarity by looking for successive colinear matches
- Set-similarity
  - Detect similarity by counting common *k*-mers

Chaining of *Winnowing* matches

Minimap2: pairwise alignment for nucleotide sequences

Heng Li
Broad Institute, 415 Main Street, Cambridge, MA 02142, USA

- Use Winnowing for computing a set of hash values for the sequences to be compared
- Find sequences of co-linear hash matches

Chaining of *Winnowing* matches

- A chain of features with the same hash, accounts for a local similarity

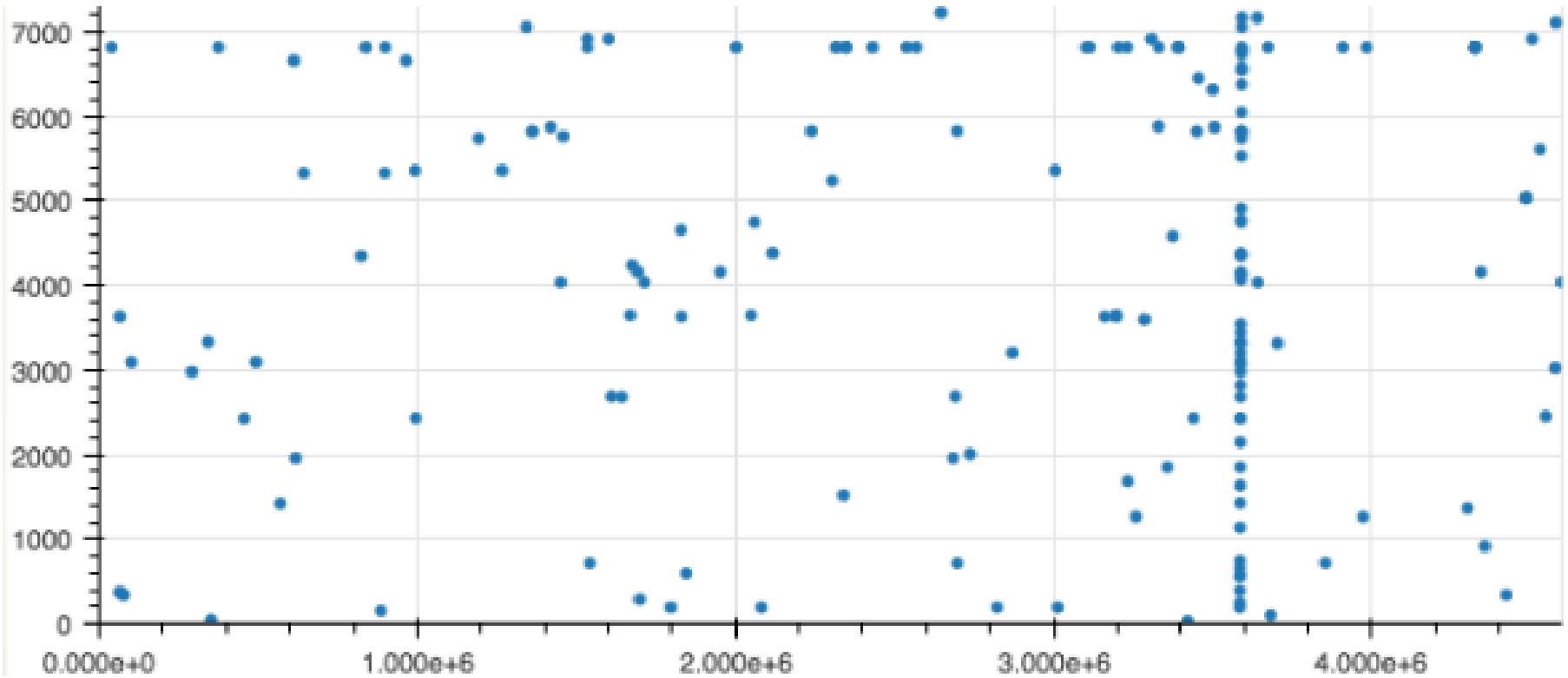
```
Fingerprint 1: {87, 6}, {12, 9}, {87, 11}, {23, 15}, {7, 18}

Fingerprint 2: {12, 0}, {87, 3}, {23, 7}, {0, 10}
```

Chaining of *Winnowing* matches

- A chain of features with the same hash, accounts for a local similarity

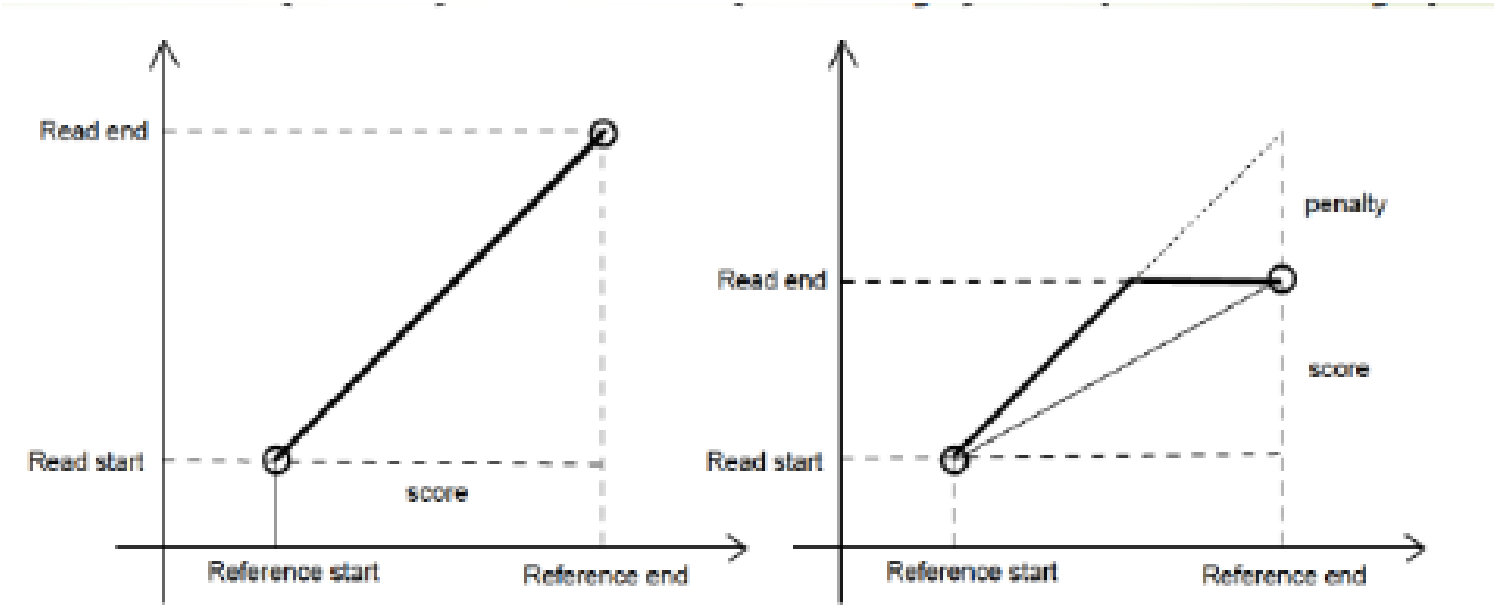
```
Fingerprint 1: {87, 6}, {12, 9}, {87, 11}, {23, 15}, {7, 18}
Fingerprint 2: {12, 0}, {87, 3}, {23, 7}, {0, 10}
```

Chaining of *Winnowing* matches



Chaining of *Winnowing* matches

- Detect co-linearity by assigning edge scores



Set-similarity of *Winnowing* hashes

A Fast Approximate Algorithm for Mapping Long Reads to Large Reference Databases

CHIRAG JAIN,[1,2] ALEXANDER DILTHEY,[2] SERGEY KOREN,[2] SRINIVAS ALURU,[1] and ADAM M. PHILLIPPY[2]

- Use Winnowing for computing a set of hash values for the sequences to be compared
- Measure the set-similarity with the Jaccard index

Set-similarity of *Winnowing* hashes

- Example with Jaccard index

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}.$$

```
Fingerprint 1: {87, 6}, {12, 9}, {87, 11}, {23, 15}, {7, 18}
Fingerprint 2: {12, 0}, {87, 3}, {23, 7}, {0, 10}

Jaccard-index: 3 / 5 = 0.6
```

Libraries

https://github.com/seqan/seqan
https://seqan.readthedocs.io/en/master/index.html