Efficient DNA/RNA sequence clustering

Using *k*-mers as an approximation for sequence similarity

Anders Kiel Hovgaard

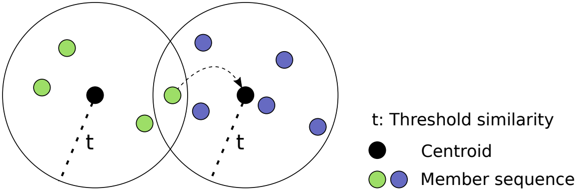
Department of Computer Science, University of Copenhagen

June 18, 2015

Introduction

Defining the clustering problem to be solved

Partitioning of sequences into a minimal number of clusters based on a measure of similarity between sequences.



- How to measure similarity/distance between sequences?
- How to cluster sequences based on such a measure?

Distance metrics

There are various distance metrics:

- edit distance, Levenshtein
- sequence alignment
- feature based distance, k-mer counting
    - K-DIST uses a kind of k-mer counting

The simple *d*2 algorithm

$$S_1 = ACTACAC$$
$$S_2 = ACAGAT$$

- Fill vectors with *k*-mer counts

	AC	AG	AT	CA	CT	GA	TA
S_1	3			1	1		1
S_2	1	1	1	1		1	

- Calculate the Euclidean distance

$$d2_2(S_1, S_2) = \sqrt{(3-1)^2 - 1^2 - 1^2 + (1-1)^2 + 1^2 - 1^2 + 1^2}$$
$$= \sqrt{9} = 3$$

The K-DIST algorithm

A variant of the simple *d*2 algorithm:

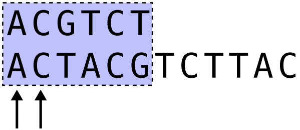
- a single *k*-mer vector
- Manhattan distance

$$\sum_{i=1}^{n} |u_i - v_i|$$

- window calculation

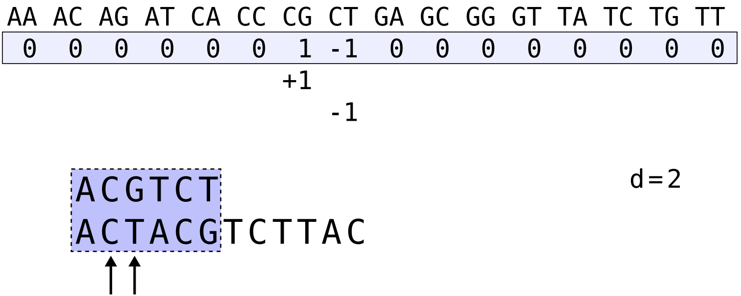
The K-Dist algorithm

AA	AC	AG	AT	CA	CC	CG	CT	GA	GC	GG	GT	TA	TC	TG	TT
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	+1														
	-1														


ACGTCT
ACTACGTCTTAC

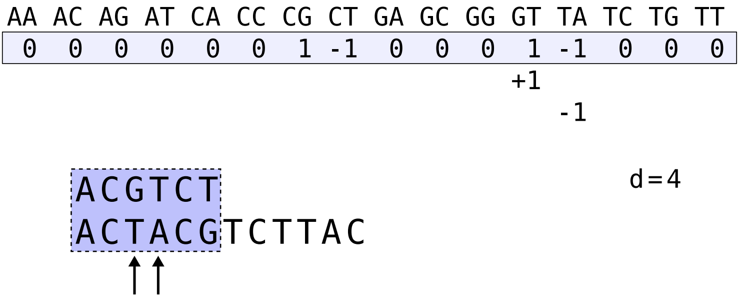
d=0

The K-DIST algorithm



AA	AC	AG	AT	CA	CC	CG	CT	GA	GC	GG	GT	TA	TC	TG	TT
0	0	0	0	0	0	1	-1	0	0	0	0	0	0	0	0

The K-DIST algorithm



AA	AC	AG	AT	CA	CC	CG	CT	GA	GC	GG	GT	TA	TC	TG	TT
0	0	0	0	0	0	1	-1	0	0	0	1	-1	0	0	0

+1

-1

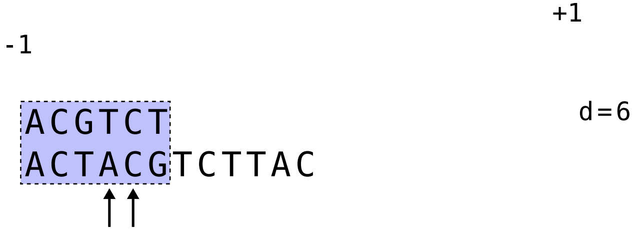
ACGTCT

ACTACGTCTTAC

d=4

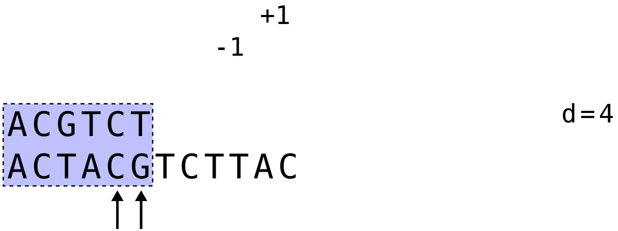
The K-DIST algorithm

AA	AC	AG	AT	CA	CC	CG	CT	GA	GC	GG	GT	TA	TC	TG	TT
0	-1	0	0	0	0	1	-1	0	0	0	1	-1	1	0	0



The K-Dist algorithm

AA	AC	AG	AT	CA	CC	CG	CT	GA	GC	GG	GT	TA	TC	TG	TT
0	-1	0	0	0	0	0	0	0	0	0	1	-1	1	0	0



The K-DIST algorithm

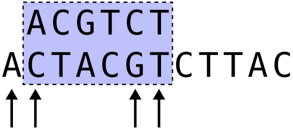
AA	AC	AG	AT	CA	CC	CG	CT	GA	GC	GG	GT	TA	TC	TG	TT
0	-1	0	0	0	0	0	0	0	0	0	1	-1	1	0	0

```
ACGTCT
ACTACGTCTTAC
```

win	0	1	2	3	4	5	6
d	4						

The K-DIST algorithm

AA	AC	AG	AT	CA	CC	CG	CT	GA	GC	GG	GT	TA	TC	TG	TT
0	0	0	0	0	0	0	0	0	0	0	0	-1	1	0	0
	+1														
											-1				



ACGTCT

ACTACGTCTTAC

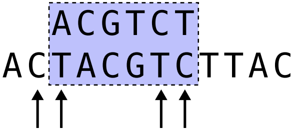
win	0	1	2	3	4	5	6
d	4	2					

The K-DIST algorithm

AA	AC	AG	AT	CA	CC	CG	CT	GA	GC	GG	GT	TA	TC	TG	TT
0	0	0	0	0	0	0	1	0	0	0	0	-1	0	0	0

+1

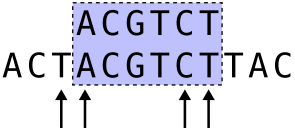
-1



win	0	1	2	3	4	5	6
d	4	2	2				

The K-DIST algorithm

AA	AC	AG	AT	CA	CC	CG	CT	GA	GC	GG	GT	TA	TC	TG	TT
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
												+1			
							-1								



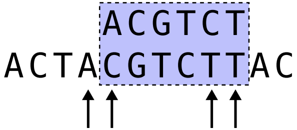
ACGTCT

ACTACGTCTTAC

win	0	1	2	3	4	5	6
d	4	2	2	0			

The K-DIST algorithm

AA	AC	AG	AT	CA	CC	CG	CT	GA	GC	GG	GT	TA	TC	TG	TT
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	-1
	+1														
															-1



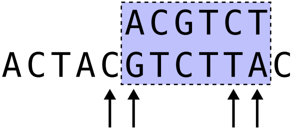
win	0	1	2	3	4	5	6
d	4	2	2	0	2		

The K-DIST algorithm

AA	AC	AG	AT	CA	CC	CG	CT	GA	GC	GG	GT	TA	TC	TG	TT
0	1	0	0	0	0	1	0	0	0	0	0	-1	0	0	-1

+1

-1



ACGTCT

ACTACGTCTTAC

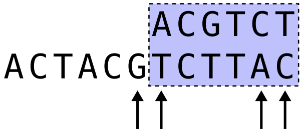
win	0	1	2	3	4	5	6
d	4	2	2	0	2	4	

The K-DIST algorithm

AA	AC	AG	AT	CA	CC	CG	CT	GA	GC	GG	GT	TA	TC	TG	TT
0	0	0	0	0	0	1	0	0	0	0	1	-1	0	0	-1

+1

-1



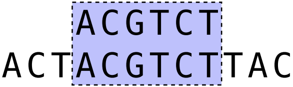
ACGTCT

ACTACGTCTTAC

win	0	1	2	3	4	5	6
d	4	2	2	0	2	4	4

The K-DIST algorithm

AA	AC	AG	AT	CA	CC	CG	CT	GA	GC	GG	GT	TA	TC	TG	TT
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



```
   ACGTCT
ACTACGTCTTAC
```

win	0	1	2	3	4	5	6
d	4	2	2	0	2	4	4

Time complexity of K-Dist

```
for $i \leftarrow 0$ to $|s| - k$ do
    $s_i \leftarrow s.substring(i, k)$
    $t_i \leftarrow t.substring(i, k)$
    update $cur\_dist$, kmers[$s_i$] and kmers[$t_i$]
```

$\Theta\left(|s| - k\right)$

$+$

```
for $i \leftarrow 0$ to $|t| - |s|$ do
    $kmer_{out} \leftarrow t.substring(i, k)$
    $kmer_{in} \leftarrow t.substring(|s| - k + i + 1, k)$
    update $cur\_dist$, kmers[$kmer_{out}$] and kmers[$kmer_{in}$]
    $min\_dist \leftarrow min(min\_dist, cur\_dist)$
```

$\Theta\left(|t| - |s|\right)$

Total: $\Theta\left(|t| - k\right)$

Cluster analysis algorithms

Various approaches to clustering:

- hierarchical clustering
- graph-based clustering
- greedy clustering

A greedy approach is necessary due to the sizes of the data.

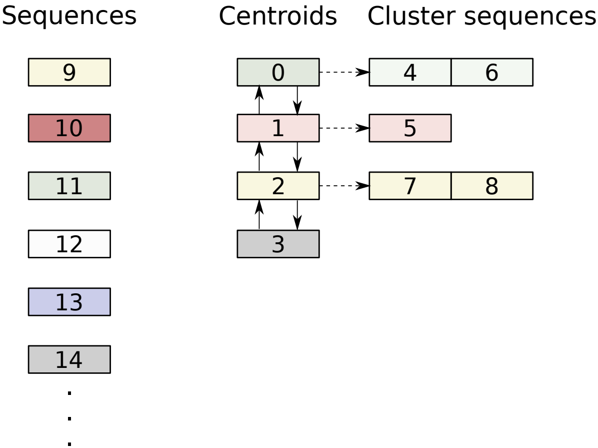
The K-Clust algorithm

The clustering algorithm used in `klust`.

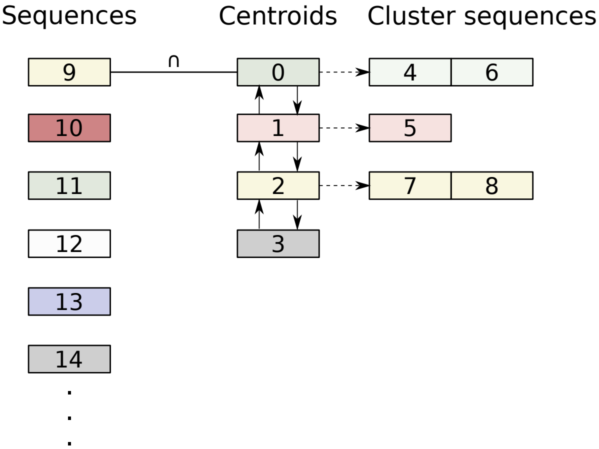
The intersection criterion

$$|K(s) \cap K(c)| \geq |K(c)| \cdot id$$

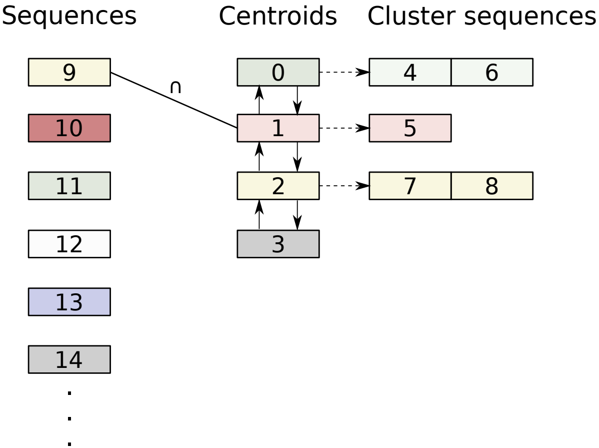
The K-CLUST algorithm



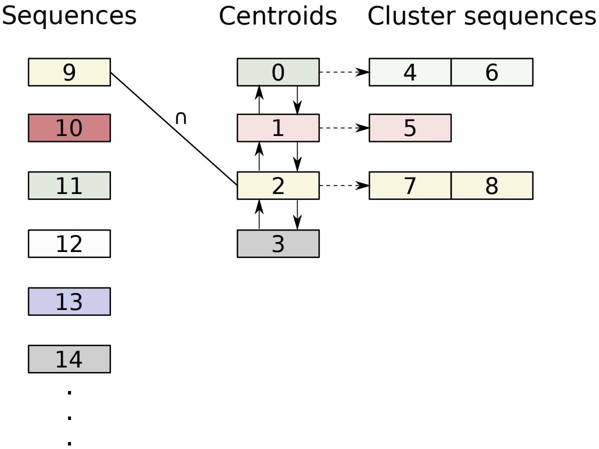
The K-Clust algorithm



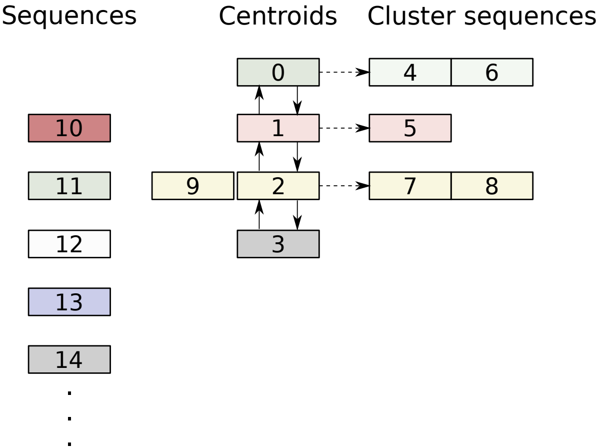
The K-Clust algorithm



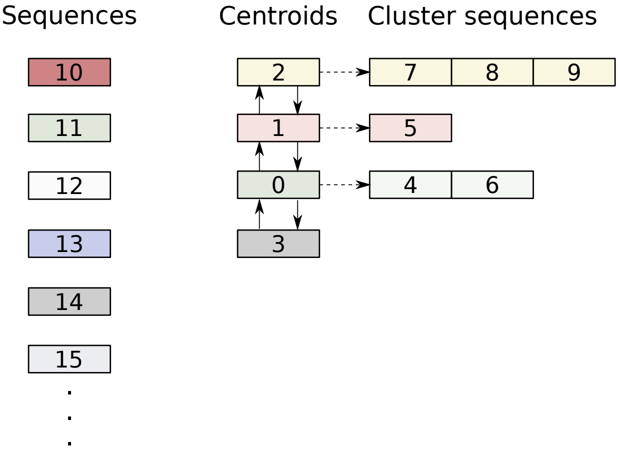
The K-Clust algorithm



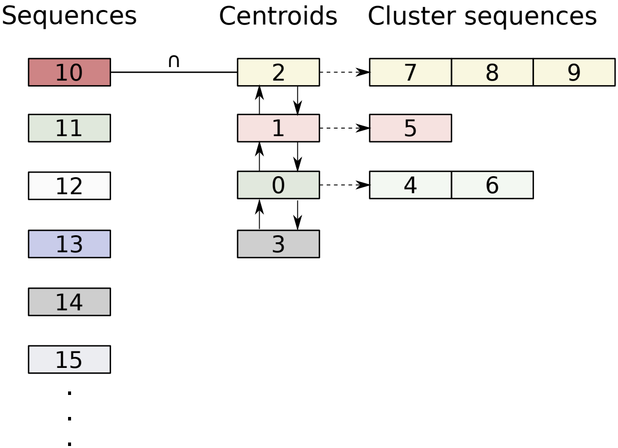
The K-Clust algorithm



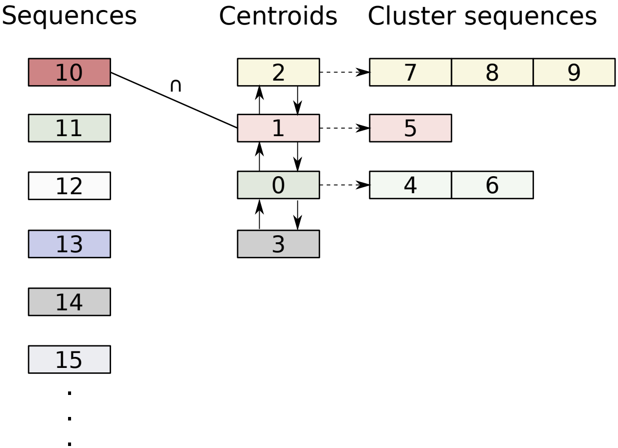
The K-CLUST algorithm



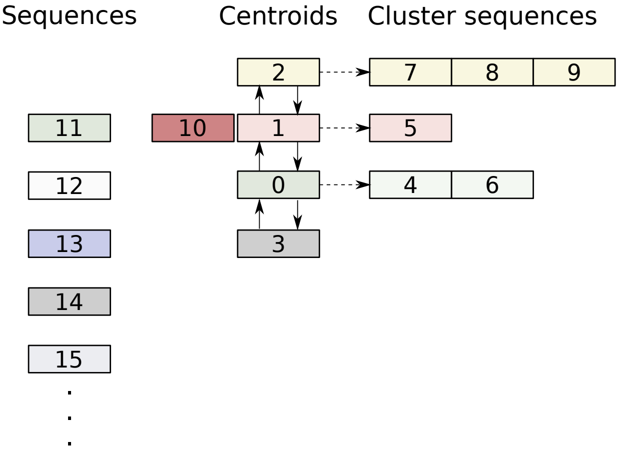
The K-Clust algorithm



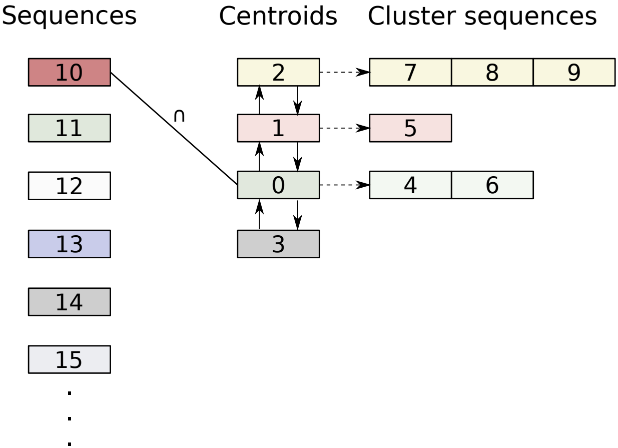
The K-Clust algorithm



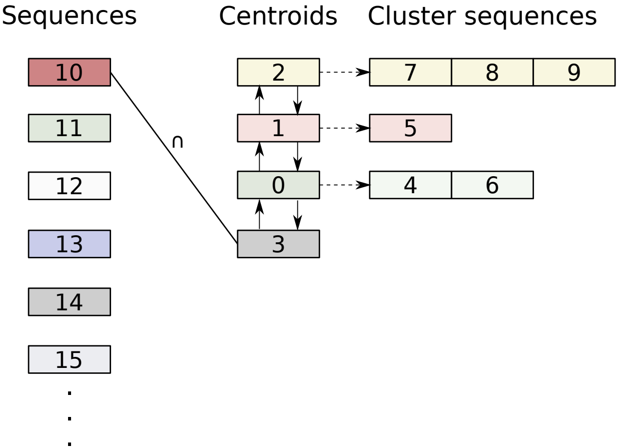
The K-Clust algorithm



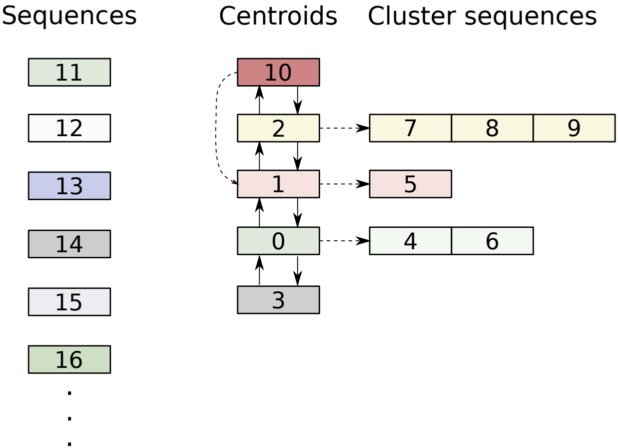
The K-Clust algorithm



The K-Clust algorithm

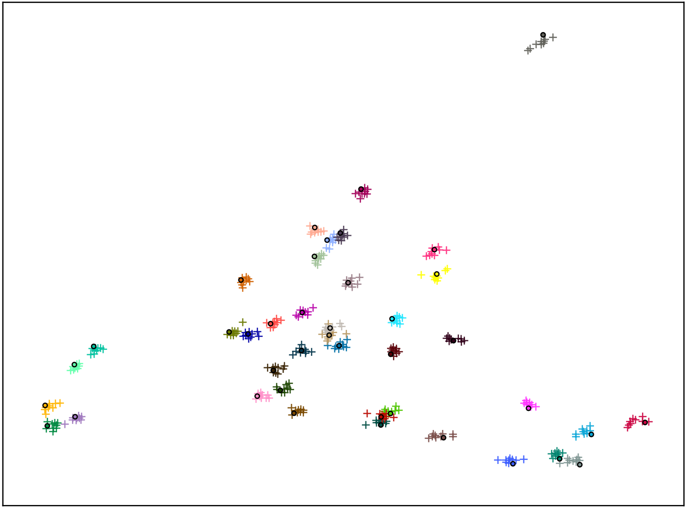


The K-CLUST algorithm



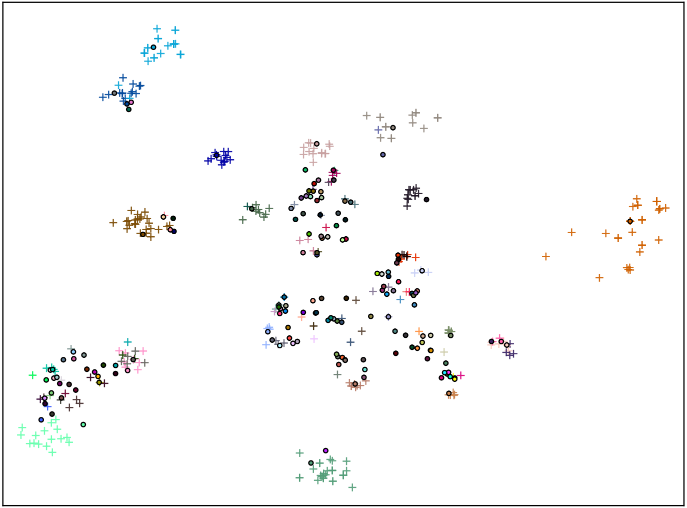
Clustering the SILVA RNA dataset

Clustering program	Time	Throughput (seqs./sec.)	Clusters	Max. memory
klust, $id = 0.90$, $k = 5$	0:42:59	614.16	159,812	$\approx$ 1021 MB
USEARCH, $id = 0.97$, -cluster_smallmem	1:04:10	411.10	221,040	$\approx$ 2048 MB



K-CLUST
$k = 5$
$id = 0.85$

clusters: 40
max. size: 10
avg. size: 10
singletons: 0



K-Clust
$k = 5$
$id = 0.85$
sort: incr.

clusters: 157
max. size: 43
avg. size: 3.18
singletons: 89