

5eme Arrondissement Paris

Graines et YASS

Quelques slides sur les graines espacées ...

Laurent Noé

16 Rue Claude Bernard, 75005 Paris

LIFL, Université Lille 1 - INRIA

Journées ANR Cocogen

10-11 mars 2008 - Paris

Image © 2008 The GeoInformation Group, InterAtlas
© 2008 Tele Atlas

© 2008 Google™

My Talk

in a few words ...

Motivation : pairwise sequence alignment.

Seeds : filtration to speed up the alignment.

Filtration :

- the *lossy* case
- the *lossless* case

Sequence Alignment

on a very small DNA example

```
TTTTGAACTGGGACGAAAGTGCATCAGTGCAAATGCGCAAGAAAAA  
CGCCGAACGCTTCAGATCAGCGCAAATGCTCAAGAGGTCTCGTCGC  
TGAGGCACTACGGCCAGCCGAGCCAGTCAT
```

Sequence Alignment

on a very small DNA example

```
TTTTGAACTGGGACGAAAGTGCATCAGTGCAAATGCGCAAGAAAA  
CGCCGAACGCTTCAGATCAGCGCAAATGCTCAAGAGGTCTCGTCGC  
TGAGGCACTACGGCCAGCCGAGCCAGTCAT
```

Sequence Alignment

on a very small DNA example

```
TTTTGAACTGGGACGAAAGTGCATCAGTGCAAATGCGCAAGAAAAA  
CGCCGAACGCTTCAGATCAGCGCAAATGCTCAAGAGGTCTCGTCGC  
TGAGGCACTACGGCCAGCCGAGCCAGTCAT
```

Sequence Alignment

on a very small DNA example

```
TTTTGAACTGGGACGAAAGTGCATCAGTGCAAATGCGCAAGAAAA  
CGCCGAACGCTTCAGATCAGCGCAAATGCTCAAGAGGTCTCGTCGC  
TGAGGCACTACGGCCAGCCGAGCCAGTCAT
```

ATCAGTGCAAATGCGCAAGA

ATCAGCGCAAATGCTCAAGA

Sequence Alignment

on a very small DNA example

```
TTTTGAACTGGGACGAAAGTGCATCAGTGCAAATGCGCAAGAAAAA  
CGCCGAACGCTTCAGATCAGCGCAAATGCTCAAGAGGTCTCGTCGC  
TGAGGCACTACGGCCAGCCGAGCCAGTCAT
```

```
ATCAGTGCAAATGCGCAAGA  
|||||:|||||||.|||||  
ATCAGCGCAAATGCTCAAGA
```

Sequence Alignment

methods used to solve this problem ...

Algorithm: Smith-Waterman algorithm (in $\mathcal{O}(n^2)$).

Heuristic: Filtration principle

- (1) some *clues* are detected using **seeds**.
- (2) these clues are extended by local dynamic programming.

Contiguous Seeds

(Fasta 85, Blast 91, Gapped-Blast 97, ...)

Principle: A contiguous seed π detects one alignment motif of size k .

Notation: π is represented by a (fixed length) word over alphabet $\{\#\}$.
($\#$ only accepts the | symbol from an alignment).

Example

seed pattern : $\pi = \#\#\#\#\#$

```
ATCAGTGCAAATGCGCAAGA
||| |:|||||.|||||
ATCAGCGCAAATGCTCAAGA
```

Contiguous Seeds

(Fasta 85, Blast 91, Gapped-Blast 97, ...)

Principle: A contiguous seed π detects one alignment motif of size k .

Notation: π is represented by a (fixed length) word over alphabet $\{\#\}$.
($\#$ only accepts the | symbol from an alignment).

Example

seed pattern : $\pi = \#\#\#\#\#\#$

$\#\#\#\#\#\#$

```
ATCAGTGCAAAATGCGCAAGA
| | | | : | | | | | | | | | |
ATCAGCGGCAAATGCTTCAAGA
```

Contiguous Seeds

(Fasta 85, Blast 91, Gapped-Blast 97, ...)

Principle: A contiguous seed π detects one alignment motif of size k .

Notation: π is represented by a (fixed length) word over alphabet $\{\#\}$.
($\#$ only accepts the | symbol from an alignment).

Example

seed pattern : $\pi = \#\#\#\#\#$

$\#\#\#\#\#$

```
ATCAGTGCAAATGCGCAAGA
| | | | : | | | | | | | | | |
ATCAGCGCAAATGCTCAAGA
```

Contiguous Seeds

(Fasta 85, Blast 91, Gapped-Blast 97, ...)

Principle: A contiguous seed π detects one alignment motif of size k .

Notation: π is represented by a (fixed length) word over alphabet $\{\#\}$.
($\#$ only accepts the | symbol from an alignment).

Example

seed pattern : $\pi = \#\#\#\#\#$

$\#\#\#\#\#$

```
ATCAGTGCAAAATGCGCAAGA
| | | | : | | | | | | | | | |
ATCAGCGCAAATGCTTCAAGA
```

Contiguous Seeds

(Fasta 85, Blast 91, Gapped-Blast 97, ...)

Principle: A contiguous seed π detects one alignment motif of size k .

Notation: π is represented by a (fixed length) word over alphabet $\{\#\}$.
($\#$ only accepts the | symbol from an alignment).

Example

seed pattern : $\pi = \#\#\#\#\#$

$\#\#\#\#\#$

```
ATCAGTGCAAAATGCGCAAGA
| | | | : | | | | | | | | | |
ATCAGCGGCAAATGCTTCAAGA
```

Contiguous Seeds

(Fasta 85, Blast 91, Gapped-Blast 97, ...)

Principle: A contiguous seed π detects one alignment motif of size k .

Notation: π is represented by a (fixed length) word over alphabet $\{\#\}$.
($\#$ only accepts the | symbol from an alignment).

Example

seed pattern : $\pi = \#\#\#\#\#$

$\#\#\#\#\#$

```
ATCAGTGCAAAATGCGACAAGA
| | | | : | | | | | | | | | |
ATCAGCGCAAATGCTCAAGA
```

Spaced Seeds

(PatternHunter 02, Burkhardt et al 01, BLASTz 03, YASS 04)

Definition

A spaced seed π is defined as a binary word over the alphabet $\{\#, -\}$:

- $\#$: accepts only match symbol | ,
- $-$: accepts all alignment symbols (*joker*) .

s : *span* (length), w : *weight* (number of $\#$).

Example

seed pattern : $\pi = \#\#\#-\#-\#\#$

```
ATCAGTGC GAATGCGCAAGA
| | | | : | | | | . | | | |
ATCAGCGC AAATGCTCAAGA
```

Spaced Seeds

(PatternHunter 02, Burkhardt et al 01, BLASTz 03, YASS 04)

Definition

A spaced seed π is defined as a binary word over the alphabet $\{\#, -\}$:

- $\#$: accepts only match symbol | ,
- $-$: accepts all alignment symbols (*joker*) .

s : *span* (length), w : *weight* (number of $\#$).

Example

seed pattern : $\pi = \#\#\#-\#-\#\#$

$\#\#\#-\#-\#\#$

```
ATCAGTGC GAATGCGCAAGA
||| |:| |:| |. ||| |
ATCAGCGCAAATGCTCAAGA
```

Spaced Seeds

(PatternHunter 02, Burkhardt et al 01, BLASTz 03, YASS 04)

Definition

A spaced seed π is defined as a binary word over the alphabet $\{\#, -\}$:

- $\#$: accepts only match symbol | ,
- $-$: accepts all alignment symbols (*joker*) .

s : *span* (length), w : *weight* (number of $\#$).

Example

seed pattern : $\pi = \#\#\#-\#-\#\#$

$\#\#\#-\#-\#\#$

```
ATCAGTGC GAATGCGCAAGA
| | | | : | | | . | | | |
ATCAGCGCAAATGCTCAAGA
```

Spaced Seeds

(PatternHunter 02, Burkhardt et al 01, BLASTz 03, YASS 04)

Definition

A spaced seed π is defined as a binary word over the alphabet $\{\#, -\}$:

- $\#$: accepts only match symbol | ,
- $-$: accepts all alignment symbols (*joker*) .

s : *span* (length), w : *weight* (number of $\#$).

Example

seed pattern : $\pi = \#\#\#-\#-\#\#$

$\#\#\#-\#-\#\#$

```
ATCAGTGC GAATGCGCAAGA
||| |:| :||| |. ||| |
ATCAGCGCAAATGCTCAAGA
```

Spaced Seeds

(PatternHunter 02, Burkhardt et al 01, BLASTz 03, YASS 04)

Definition

A spaced seed π is defined as a binary word over the alphabet $\{\#, -\}$:

- $\#$: accepts only match symbol | ,
- $-$: accepts all alignment symbols (*joker*) .

s : *span* (length), w : *weight* (number of $\#$).

Example

seed pattern : $\pi = \#\#\#-\#-\#\#$

$\#\#\#-\#-\#\#$

```
ATCAGTGC GAATGCGCAAGA
| | | | : | | | . | | | |
ATCAGCGCAAATGCTCAAGA
```

Spaced Seeds

(PatternHunter 02, Burkhardt et al 01, BLASTz 03, YASS 04)

Definition

A spaced seed π is defined as a binary word over the alphabet $\{\#, -\}$:

- $\#$: accepts only match symbol | ,
- $-$: accepts all alignment symbols (*joker*) .

s : *span* (length), w : *weight* (number of $\#$).

Example

seed pattern : $\pi = \#\#\#-\#-\#\#$

$\#\#\#-\#-\#\#$

```
ATCAGTGC GAATGCGCAAGA
| | | | : | | | . | | | |
ATCAGCGC AAATGCTCAAGA
```

Spaced Seeds

(PatternHunter 02, Burkhardt et al 01, BLASTz 03, YASS 04)

Definition

A spaced seed π is defined as a binary word over the alphabet $\{\#, -\}$:

- $\#$: accepts only match symbol | ,
- $-$: accepts all alignment symbols (*joker*) .

s : *span* (length), w : *weight* (number of $\#$).

Example

seed pattern : $\pi = \#\#\#-\#-\#\#$

$\#\#\#-\#-\#\#$

```
ATCAGTGC GAATGCGCAAGA
| | | | : | | | . | | | |
ATCAGCGC AAATGCTCAAGA
```

Example

Example

$\pi_c = #####$

$\pi_s = ###-#-##$

Example

$\pi_c = #####$

$\pi_s = ###-#-##$

$\alpha =$

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

Example

$\pi_c = #####$

$\pi_s = ###-#-##$

$\alpha =$

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

Example

$\pi_c = #####$

$\pi_s = ###-#-##$

$\alpha =$

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

Example

$\pi_c = #####$

$\pi_s = ###-#-##$

$\alpha =$

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

Example

$\pi_c = #####$

$\pi_s = ###-#-##$

$\alpha =$

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

Example

$\pi_c = #####$

$\pi_s = ###-#-##$

$\alpha =$

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

Example

$\pi_c = #####$

$\pi_s = ###-#-##$

$\alpha =$

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

Example

Example

ATCAGTGC^TCAAATGCTCAAGA

||||| |||||||||||||||

ATCAGCGCAAATGCTCAAGA

#####

#####

#####

#####

#####

#####

#####

#####

#####

#####

#####

#####

#####

#####

#####

ATCAGTGC^TCAAATGCTCAAGA

||||| |||||||||||||||

ATCAGCGCAAATGCTCAAGA

###-#-##

###-#-##

###-#-##

###-#-##

###-#-##

###-#-##

###-#-##

###-#-##

###-#-##

###-#-##

###-#-##

###-#-##

Example

ATCAGTGC^TCAAATGCTCAAGA
||||| ||||||||||||||||
ATCAGCGCAAATGCTCAAGA

#####

#####

#####

#####

#####

#####

#####

#####

#####

#####

#####

#####

#####

#####

#####

ATCAGTGC^TCAAATGCTCAAGA
||||| ||||||||||||||||
ATCAGCGCAAATGCTCAAGA

###--#-##

###-#-##

###-#-##

###--#-##

###--#-##

###--#-##

###--#-##

###--#-##

###--#-##

###--#-##

###--#-##

###--#-##

Example

ATCAGTGC^AAAATGCTCAAGA
| | | | | | | | | | | | | | | |
ATCAGCGCAAATGCTCAAGA

#####

#####

#####

#####

#####

#####

#####

#####

#####

#####

#####

#####

#####

#####

#####

ATCAGTGC^AAAATGCTCAAGA
| | | | | | | | | | | | | | | |
ATCAGCGCAAATGCTCAAGA

###--#-##

###-#-##

###-#-##

###--#-##

###--#-##

###--#-##

###--#-##

###--#-##

###--#-##

###--#-##

###--#-##

###--#-##

Example

ATCAGTGCGAATGCGCAAGA
||||| || ||||| |||||
ATCAGCGCAAATGCTCAAGA

#####

ATCAGTGCGAATGCGCAAGA
||||| || ||||| |||||
ATCAGCGCAAATGCTCAAGA

###-#-##
###-#-##
###-#-##
###-#-##
###-#-##
###-#-##
###-#-##
###-#-##
###-#-##
###-#-##
###-#-##
###-#-##
###-#-##
###-#-##
###-#-##
###-#-##
###-#-##

Example

ATCAGTGCGAATGCGCAAGA
||||| || ||||| |||||
ATCAGCGCAAATGCTCAAGA

#####

ATCAGTGCGAATGCGCAAGA
||||| || ||||| |||||
ATCAGCGCAAATGCTCAAGA

###-#-##
###-#-##
###-#-##
###-#-##
###-#-##
###-#-##
###-#-##
###-#-##
###-#-##
###-#-##
###-#-##
###-#-##
###-#-##
###-#-##
###-#-##
###-#-##
###-#-##
###-#-##
###-#-##
###-#-##
###-#-##

Example

#####

###--#-##

Example

#####

(5)

###--#-##

###--#-## (3)

Example

#####

(5,4)

###--#-##

###--#-## (3,2)

Example

#####

(5,4,3)

###--#-##

###--#-## (3,2,2)

Example

#####

(5,4,3,2)

###--#-##

###--#-## (3,2,2,1)

Example

#####

(5,4,3,2,1)

###--#-##

###--#-## (3,2,2,1,2)

Example

#####

(5,4,3,2,1)

###--#-##

###--#-## (3,2,2,1,2,2)

Example

#####

(5,4,3,2,1)

###--#-##

###--#-## (3,2,2,1,2,2,2)

Example

#####

(5,4,3,2,1)

###--#-##

###--#-## (3,2,2,1,2,2,2,1)

Example

#####

(5,4,3,2,1)

###--#-##

(3,2,2,1,2,2,2,1)

- Burkhardt, Karkkainen, CPM 2001: *spaced seeds for (lossless) approximate pattern matching*
- Ma, Tromp, Li 2002 (*PatternHunter*): *spaced seeds for (lossy) similarity search*
- Califano, Rigoutsos 1993 (*FLASH*), Buhler 2001 (*LSH*)

Spaced Seeds

Research threads (cont.)

- **Estimating the sensitivity of a seed:** Keich et al 2002, Buhler et al 2003, Brejova et al 2003, Choi et al 2004, Kucherov et al 2004, Mak&Benson 2007
- **Extended seed models:** BLASTZ 2003, Brejova et al 2003, Chen&Sung 2003, Noe&Kucherov 2004, Sun&Buhler 2006, Mak et al 2006, Zhou&Florea 2007
- **Statistical foundations:** Choi&Zhang 2004, Zhang 2005, Kong 2007, Ma&Yao 2008
- **Efficient implementation of spaced seeds:** Csuros 2004, Csuros&Ma 2004
- **Multiple spaced seeds:** Li et al 2004 (PatternHunter II), Sun&Buhler 2004, Kong 2007
- **Designing (multiple) seeds:** Xu et al 2004, Brown 2004, Ilie&Ilie 2007
- **Lossless (multiple) seeds:** Burkhardt&Karkkainen 2001, Kucherov et al 2004, Farach et al 2004, Fontaine et al 2004, Nicolas&Rivals 2005
- **Surveys:** Brown&Li&Ma 2005, Brown 2008

Spaced Seeds

How to choose the best one

The main question in (most of) these papers: how to choose the best *seed* ...

Spaced Seeds

How to choose the best one

The main question in (most of) these papers: how to choose the best *seed* ...

Sensitivity : defined as the *probability* to have at least one *hit* (seed occurrence) inside an alignment.

Spaced Seeds

How to choose the best one

The main question in (most of) these papers: how to choose the best *seed* ...

Sensitivity : defined as the *probability* to have at least one *hit* (seed occurrence) inside an alignment.

Best Seed : defined as the one that maximize the sensitivity (among the seeds of a given class).

Mutations on DNA

Transitions and Transversions ...

Two kinds of mismatches : *transitions* and *transversions*

Definition

Transitions are substitutions between **purins** ($A \leftrightarrow G$) or between **pyrimidins** ($T \leftrightarrow C$). Transitions are usually overrepresented mutations ...

Example

```
ATCAGTGC GAATGCCAAGA
| | | | : | | : | | | | . | | | |
ATCAGCGC AAATGCTCAAGA
```

- **:** is a transition symbol.
- **.** is a transversion symbol.

Transition Constrained Seeds

(YASS 04, Zhou&Florea 07)

Definition

A transition constrained seed π is defined as a ternary word over the alphabet $\{\#, @, -\}$ with :

- # : accepts only match symbol |,
- - : accepts all alignment symbols (*joker*),
- @ : accepts match symbol | or transition mismatch symbol :,

Example

seed pattern : $\pi = \# \# @ \# - @ \# \#$

```
ATCAGTGC GAATGC CAAGA
| | | | : | | : | | | | . | | | |
ATCAGCGC AAATGCT CAAGA
```

Transition Constrained Seeds

(YASS 04, Zhou&Florea 07)

Definition

A transition constrained seed π is defined as a ternary word over the alphabet $\{\#, @, -\}$ with :

- $\#$: accepts only match symbol | ,
- $-$: accepts all alignment symbols (*joker*),
- $@$: accepts match symbol | or transition mismatch symbol :,

Example

seed pattern : $\pi = \#\#@\#-\#@\#$

$\#\#@\#-\#@\#$

```
ATCAGTGCGAATGCGCAAGA
| | | | : | | : | | | | . | | | |
ATCAGCGCAAATGCTTCAAGA
```

Transition Constrained Seeds

(YASS 04, Zhou&Florea 07)

Definition

A transition constrained seed π is defined as a ternary word over the alphabet $\{\#, @, -\}$ with :

- $\#$: accepts only match symbol $|$,
- $-$: accepts all alignment symbols (*joker*),
- $@$: accepts match symbol $|$ or transition mismatch symbol $:$,

Example

seed pattern : $\pi = \#\#@\#-\#@\#$

$\#\#@\#-\#@\#$

```
ATCAGTGCGAATGCGCAAGA
| | | | : | | : | | | | . | | | |
ATCAGCGCAAATGCTTCAAGA
```

Transition Constrained Seeds

(YASS 04, Zhou&Florea 07)

Definition

A transition constrained seed π is defined as a ternary word over the alphabet $\{\#, @, -\}$ with :

- $\#$: accepts only match symbol $|$,
- $-$: accepts all alignment symbols (*joker*),
- $@$: accepts match symbol $|$ or transition mismatch symbol $:$,

Example

seed pattern : $\pi = \#\#@\#-\#@\#$

$\#\#@\#-\#@\#$

```
ATCAGTGCGAATGCGCAAGA
||| |:| |:| |. ||| |
ATCAGCGCAAATGCTTCAAGA
```

Transition Constrained Seeds

(YASS 04, Zhou&Florea 07)

Definition

A transition constrained seed π is defined as a ternary word over the alphabet $\{\#, @, -\}$ with :

- $\#$: accepts only match symbol | ,
- $-$: accepts all alignment symbols (*joker*),
- $@$: accepts match symbol | or transition mismatch symbol :,

Example

seed pattern : $\pi = \#\#@\#-\#@\#$

$\#\#@\#-\#@\#$

```
ATCAGTGCGAATGCGCAAGA
| | | | : | | : | | | | . | | | |
ATCAGCGCAAATGCTTCAAGA
```

Transition Constrained Seeds

(YASS 04, Zhou&Florea 07)

Definition

A transition constrained seed π is defined as a ternary word over the alphabet $\{\#, @, -\}$ with :

- $\#$: accepts only match symbol |,
- $-$: accepts all alignment symbols (*joker*),
- $@$: accepts match symbol | or transition mismatch symbol :,

Example

seed pattern : $\pi = \#\#@\#-\#@\#$

$\#\#@\#-\#@\#$

```
ATCAGTGCGAATGCGCAAGA
||| |:| |:| |. ||| |
ATCAGCGCAAATGCTTCAAGA
```

Transition Constrained Seeds

(YASS 04, Zhou&Florea 07)

Definition

A transition constrained seed π is defined as a ternary word over the alphabet $\{\#, @, -\}$ with :

- $\#$: accepts only match symbol $|$,
- $-$: accepts all alignment symbols (*joker*),
- $@$: accepts match symbol $|$ or transition mismatch symbol $:$,

Example

seed pattern : $\pi = \#\#@\#-\@##$

$\#\#@\#-\@##$

```
ATCAGTGC GAATGCGCAAGA
||| |:| |:| |. |||
ATCAGCGCAAATGCTCAAGA
```

Seed examples

- *spaced* and *transition constrained* seeds.
- *multiple* seeds.

Example

parameters :

Alignment length : 64

Alignment model : *Bernoulli* (*match:0.7,transition:0.15,transversion:0.15*)

results :

- | | |
|---|----------|
| • ##### | 0.412080 |
| • ##-##---##-##-### | 0.595740 |
| • #@#-##---##-@-@#@# | 0.603236 |
| • ####---#--#---#--###,##-##---#-##-### | 0.748452 |
| • #@##@----@-@#-@@#-##,##@-#@#-#-#### | 0.756022 |

Problem :

seed sensitivity IS difficult, seed design MUST BE difficult

- *Computing sensitivity is NP-Hard* (Li&Ma 04, Nicolas&Rivals 05).
- *Golomb ruler and seeds* (Nicolas&Rivals 05, Ma&Yao 08).

(m, k) -similarities

Definition

Given two integers m, k , find **all** the similarities of length $\geq m$ with $\leq k$ mismatches.

Example

if $m = 18, k = 3$, then

111011110111101111

and

111110101111011111

are similarities that must be detected by the filter.

(m, k) -similarities and lossless seeds

(Burkhardt&Karkkainen 01)

one can use

- a contiguous seed filter $\pi = \#\#\#\#$,
- a spaced seed filter $\pi = \#\#-\#\#\#$.

Example

111110101111011111

(m, k) -similarities and lossless seeds

(Burkhardt&Karkkainen 01)

one can use

- a contiguous seed filter $\pi = \#\#\#\#$,
- a spaced seed filter $\pi = \#\#-\#\#\#$.

Example

$\#\#\#\#$

111110101111011111

(m, k) -similarities and lossless seeds

(Burkhardt&Karkkainen 01)

one can use

- a contiguous seed filter $\pi = \text{####}$,
- a spaced seed filter $\pi = \text{##-###}$.

Example

####

111110101111011111

(m, k) -similarities and lossless seeds

(Burkhardt&Karkkainen 01)

one can use

- a contiguous seed filter $\pi = \#\#\#\#$,
- a spaced seed filter $\pi = \#\#-\#\#\#$.

Example

$\#\#-\#\#\#$

111110101111011111

(m, k) -similarities and lossless seeds

(Burkhardt&Karkkainen 01)

one can use

- a contiguous seed filter $\pi = \#\#\#\#$,
- a spaced seed filter $\pi = \#\#-\#\#\#$.

Example

$\#\#-\#\#\#$

111110101111011111

(m, k) -similarities and lossless seeds

(Burkhardt&Karkkainen 01)

one can use

- a contiguous seed filter $\pi = \#\#\#\#$,
- a spaced seed filter $\pi = \#\#-\#\#\#$.

Example

$\#\#-\#\#\#$

111110101111011111

(m, k) -similarities and lossless seeds

(Burkhardt&Karkkainen 01)

one can use

- a contiguous seed filter $\pi = \#\#\#\#$,
- a spaced seed filter $\pi = \#\#-\#\#\#$.

Example

$\#\#-\#\#\#$

111110101111011111

$\#\#-\#\#\#$

(m, k) -similarities and lossless seeds

(Burkhardt&Karkkainen 01)

one can use

- a contiguous seed filter $\pi = \#\#\#\#$,
- a spaced seed filter $\pi = \#\#-\#\#\#$.

Example

$\#\#-\#\#\#$

111110101111011111

$\#\#-\#\#\#$

(m, k) -similarities and lossless seeds

(Burkhardt&Karkkainen 01)

one can use

- a contiguous seed filter $\pi = \#\#\#\#$,
- a spaced seed filter $\pi = \#\#-\#\#\#$.

Example

$\#\#-\#\#\#$

111110101111011111

$\#\#-\#\#\#$

(m, k) -similarities and lossless seeds

(Burkhardt&Karkkainen 01)

one can use

- a contiguous seed filter $\pi = \#\#\#\#$,
- a spaced seed filter $\pi = \#\#-\#\#\#$.

Example

$\#\#-\#\#\#$

111110101111011111

$\#\#-\#\#\#$

(m, k) -similarities and lossless seeds

(Burkhardt&Karkkainen 01)

one can use

- a contiguous seed filter $\pi = #####$,
- a spaced seed filter $\pi = ##-###$.

Example

##-###

111110101111011111

##-###

(m, k) -similarities and lossless seeds

(Burkhardt&Karkkainen 01)

one can use

- a contiguous seed filter $\pi = \#\#\#\#$,
- a spaced seed filter $\pi = \#\#-\#\#\#$.

Example

$\#\#-\#\#\#$

111110101111011111

$\#\#-\#\#\#$

(m, k) -similarities and lossless seeds

(Burkhardt&Karkkainen 01)

one can use

- a contiguous seed filter $\pi = \#\#\#\#$,
- a spaced seed filter $\pi = \#\#-\#\#\#$.

Example

$\#\#-\#\#\#$

111110101111011111

$\#\#-\#\#\#$

(m, k) -similarities and lossless seeds

(Burkhardt&Karkkainen 01)

one can use

- a contiguous seed filter $\pi = \#\#\#\#$,
- a spaced seed filter $\pi = \#\#-\#\#\#$.

Example

```
          ##-###  
111110101111011111  
          ##-###
```

(m, k) -similarities and lossless seeds

(Burkhardt&Karkkainen 01)

one can use

- a contiguous seed filter $\pi = \#\#\#\#$,
- a spaced seed filter $\pi = \#\#-\#\#\#$.

Example

```
          ##-###  
111110101111011111  
          ##-###
```

(m, k) -similarities and lossless seeds

(Burkhardt&Karkkainen 01)

one can use

- a contiguous seed filter $\pi = \#\#\#\#$,
- a spaced seed filter $\pi = \#\#-\#\#\#$.

Example

```
          ##-###  
11111010111101111  
          ##-###
```

(m, k) -similarities and lossless seeds

(Burkhardt&Karkkainen 01)

one can use

- a contiguous seed filter $\pi = \#\#\#\#$,
- a spaced seed filter $\pi = \#\#-\#\#\#$.

Example

```
          ##-###  
111110101111011111  
          ##-###
```

(m, k) -similarities and lossless seeds

(Burkhardt&Karkkainen 01)

one can use

- a contiguous seed filter $\pi = \#\#\#\#$,
- a spaced seed filter $\pi = \#\#-\#\#\#$.

Example

$\#\#-\#\#\#$

111110101111011111

$\#\#-\#\#\#$

(m, k) -similarities and lossless seeds

(Burkhardt&Karkkainen 01)

one can use

- a contiguous seed filter $\pi = \#\#\#\#$,
- a spaced seed filter $\pi = \#\#-\#\#\#$.

Example

$\#\#-\#\#\#$

111110101111011111

$\#\#-\#\#\#$

(m, k) -similarities and lossless multiple seeds

(Kucherov et al 04, Farach et al 04)

using more than one *single* seed improves the filter ...

Example

$$\pi = \{ \text{\#}\# - \text{\#}\#\#\}$$

(m, k) -similarities and lossless multiple seeds

(Kucherov et al 04, Farach et al 04)

using more than one *single* seed improves the filter ...

Example

$$\pi = \{ \text{\#}\text{\#}\text{-}\text{\#}\text{\#}\text{\#} \}$$

$$\pi_2 = \left\{ \begin{array}{l} \text{\#}\text{\#}\text{-}\text{\#}\text{-}\text{\#}\text{\#}\text{\#}\text{\#}\text{\#} \\ \text{\#}\text{\#}\text{\#}\text{-}\text{-}\text{-}\text{\#}\text{-}\text{-}\text{\#}\text{\#}\text{-}\text{\#} \end{array} \right.$$

(m, k) -similarities and lossless multiple seeds

Example

1111111101110111011

(m, k) -similarities and lossless multiple seeds

Example

1111111101110111011

$$\pi_2 = \begin{cases} \#\#-\#\-##### \\ \###---\#\--\#\-#\end{cases}$$

(m, k) -similarities and lossless multiple seeds

Example

1111111101110111011

###--#--##-#

###--#--##-#

$$\pi_2 = \begin{cases} \#\#-\#\-##### \\ \#\#\--\#\--\#\-\# \end{cases}$$

(m, k) -similarities and lossless multiple seeds

Example

1111111101110111011

$$\pi_6 = \left\{ \begin{array}{l} \#\#-\#\#-\#\#\#\#\#, \\ \#\#\#-\#\#\#\#--\#\#, \\ \#\#\#-\#\#\#---\#\#\#\#, \\ \#\#\#---\#\#\#\#-\#\#\#, \\ \#\#\#---\#\-\#\-\#\#\#-\#\#, \\ \#\#\#-\#\-\#\-\#\#-----\#\#\# \end{array} \right.$$

(m, k) -similarities and lossless multiple seeds

Example

1111111101110111011

###-##-##-##-##

$$\pi_6 = \left\{ \begin{array}{l} \text{##-##-#####}, \\ \text{###-#####-##}, \\ \text{###-##-##-##-###}, \\ \text{##-##-#####-###}, \\ \text{###-##-##-##-##}, \\ \text{###-##-##-##-##-###} \end{array} \right.$$

Potential applications

- *oligonucleotide* selection
- *loop* detection ?
- ...