

Efficient seeding techniques for protein similarity search

Mikhail Roytberg¹, Anna Gambin², **Laurent Noé**³, Sławomir Lasota², Eugenia Furletova¹, Ewa Szczurek⁴, Gregory Kucherov³

¹ Institute of Mathematical Problems in Biology, Pushchino (Moscow region)

² Institute of Informatics, Warsaw University

³ LIFL, Université Lille 1 - CNRS - INRIA (France)

⁴ Max Planck Institute for Molecular Genetics

ALBIO 2008, Technical University of Vienna,
July 7-9, 2008

My talk

in a few words ...

Motivation : pairwise sequence alignment.

Seeds : filtration to speed-up sequence alignment.

Subset Seeds :

- a new model of seed.
- **applied on protein alignments.**

My talk

in a few words ...

Motivation : pairwise sequence alignment.

Seeds : filtration to speed-up sequence alignment.

Subset Seeds :

- a new model of seed.
- **applied on protein alignments.**



first slides give examples of **nucleic** (\neq **protein**) alignments

Sequence Alignment

on a very small nucleic example

```
TTTTGAACTGGGACGAAAGTGCATCAGTGCAAATGCGCAAGAAAAA  
CGCCGAACGCTTCAGATCAGCGCAAATGCTCAAGAGGTCTCGTCGC  
TGAGGCACTACGGCCAGCCGAGCCAGTCAT
```

Sequence Alignment

on a very small nucleic example

```
TTTTGAACTGGGACGAAAGTGCATCAGTGCAAATGCGCAAGAAAAA  
CGCCGAACGCTTCAGATCAGCGCAAATGCTCAAGAGGTCTCGTCGC  
TGAGGCACTACGGCCAGCCGAGCCAGTCAT
```

Sequence Alignment

on a very small nucleic example

```
TTTTGAACTGGGACGAAAGTGCATCAGTGCAAATGCGCAAGAAAAA  
CGCCGAACGCTTCAGATCAGCGCAAATGCTCAAGAGGTCTCGTCGC  
TGAGGCACTACGGCCAGCCGAGCCAGTCAT
```

Sequence Alignment

on a very small nucleic example

```
TTTTGAACTGGGACGAAAGTGCATCAGTGCAAATGCGCAAGAAAA  
CGCCGAACGCTTCAGATCAGCGCAAATGCTCAAGAGGTCTCGTCGC  
TGAGGCACTACGGCCAGCCGAGCCAGTCAT
```

ATCAGTGCAAATGCGCAAGA

ATCAGCGCAAATGCTCAAGA

Sequence Alignment

on a very small nucleic 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^[30], Blast 90^[1], Gapped-Blast 97^[2], ...)

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 : $\pi = \#\#\#\#\#$

```
ATCAGTGCAAATGCCAAGA
||| | : ||| | | | | . | | | |
ATCAGCGCAAATGCTCAAGA
```

Contiguous Seeds

(Fasta 85^[30], Blast 90^[1], Gapped-Blast 97^[2], ...)

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 : $\pi = \#\#\#\#\#$

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

```
ATCAGTGC AAATGC GCAAGA
| | | | : | | | | | | . | | | |
ATCAGCGCAAATGCTCAAGA
```

Contiguous Seeds

(Fasta 85^[30], Blast 90^[1], Gapped-Blast 97^[2], ...)

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 : $\pi = \#\#\#\#\#$

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

```
ATCAGTGC AAATGC GCAAGA
||| |: ||| | | | | | | | |
ATCAGCGCAAATGCTCAAGA
```

Contiguous Seeds

(Fasta 85^[30], Blast 90^[1], Gapped-Blast 97^[2], ...)

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 : $\pi = \#\#\#\#\#$

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

```
ATCAGTGC AAATGC GCAAGA
| | | | : | | | | | | . | | | |
ATCAGCGCAAATGCTCAAGA
```

Contiguous Seeds

(Fasta 85^[30], Blast 90^[1], Gapped-Blast 97^[2], ...)

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 : $\pi = \#\#\#\#\#$

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

```
ATCAGTGC AAATGC GCAAGA
| | | | : | | | | | | . | | | |
ATCAGCGCAAATGCTCAAGA
```

Contiguous Seeds

(Fasta 85^[30], Blast 90^[1], Gapped-Blast 97^[2], ...)

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 : $\pi = \#\#\#\#\#$

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

ATCAGTGC^AAAATGC^GCAAGA

| | | | : | | | | | | . | | | |

ATCAGC^GGCAAATGCT^TCAAGA

Spaced Seeds

(PatternHunter 02^[31], Burkhardt & al. 01^[10], BLASTz 03^[40], YASS 04^[38])

Definition

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

- $\#$: 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^[31], Burkhardt & al. 01^[10], BLASTz 03^[40], YASS 04^[38])

Definition

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

- $\#$: 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^[31], Burkhardt & al. 01^[10], BLASTz 03^[40], YASS 04^[38])

Definition

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

- $\#$: 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^[31], Burkhardt & al. 01^[10], BLASTz 03^[40], YASS 04^[38])

Definition

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

- $\#$: 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^[31], Burkhardt & al. 01^[10], BLASTz 03^[40], YASS 04^[38])

Definition

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

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

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

Example

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

$\###-##$

```
ATCAGTGCGAATGCGCAAGA
| | | | : | | : | | | | . | | | |
ATCAGCGCAAATGCTCAAGA
```

Spaced Seeds

(PatternHunter 02^[31], Burkhardt & al. 01^[10], BLASTz 03^[40], YASS 04^[38])

Definition

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

- $\#$: 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^[31], Burkhardt & al. 01^[10], BLASTz 03^[40], YASS 04^[38])

Definition

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

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

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

Example

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

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

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

- Burkhardt & Karkkainen 2001^[10]:
spaced seeds for (lossless) approximate pattern matching
- Ma, Tromp & Li 2002^[31] (*PatternHunter*):
spaced seeds for (lossy) similarity search
- Califano & Rigoutsos 1993^[12] (*FLASH*), Buhler 2001^[8] (*LSH*)
randomly chosen *spaced seeds for (lossy) similarity search*

Spaced Seeds

Research threads (cont.)

- **Estimating the sensitivity of a seed:** Keich et al 2002^[23], Buhler et al 2003^[9], Brejova et al 2003^[3], Choi et al 2004^[14], Kucherov et al 2004^[26], Mak&Benson 2007^[33], Gao et al 2007^[20]
- **Extended seed models:** BLASTZ 2003^[40], Brejova et al 2003^[4], Chen&Sung 2003^[13], Noe&Kucherov 2004^[38], Sun&Buhler 2006^[42], Mak et al 2006^[34], Zhou&Florea 2007^[46], Yang&Zhang 2008^[44]
- **Statistical foundations:** Nicodeme et al 2002^[36], Choi&Zhang 2004^[14], Zhang 2007^[45], Kong 2007^[25], Ma&Yao 2008^[32]
- **Efficient implementation of spaced seeds:** Csuros 2004^[15], Csuros&Ma 2005^[16, 17]
- **Multiple spaced seeds:** Li et al 2004^[28] (PatternHunter II), Sun&Buhler 2004^[41], Kong 2007^[25]
- **Designing (multiple) seeds:** Xu et al 2004^[43], Brown 2004^[5], Ilie&Ilie 2007^[22]
- **Lossless (multiple) seeds:** Burkhardt&Karkkainen 2001^[11], Kucherov et al 2004^[27], Farach et al 2004^[18], Fontaine et al 2004^[19], Nicolas&Rivals 2005^[37]
- **Surveys:** Brown&Li&Ma 2004^[6], Brown 2008^[7]

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* ?
- Another question in these papers: can we extend the *spaced seed* model ?

Nucleic Mutations

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 ...

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

Example

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

Transition Constrained Seed

(Chen&Sung 03^[13], YASS 04^[38], Sun&Buhler 06^[42], Zhou&Florea 07^[46], Yang&Zhang 08^[44])

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
```

Transition Constrained Seed

(Chen&Sung 03^[13], YASS 04^[38], Sun&Buhler 06^[42], Zhou&Florea 07^[46], Yang&Zhang 08^[44])

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

@ # - @ #

ATCAG**T**G**C**GAATG**C**CAAGA

| | | | : | | : | | | | . | | | |

ATCAG**C**G**C**AAATG**C**TCAAGA

Transition Constrained Seed

(Chen&Sung 03^[13], YASS 04^[38], Sun&Buhler 06^[42], Zhou&Florea 07^[46], Yang&Zhang 08^[44])

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
```

Transition Constrained Seed

(Chen&Sung 03^[13], YASS 04^[38], Sun&Buhler 06^[42], Zhou&Florea 07^[46], Yang&Zhang 08^[44])

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
```

Transition Constrained Seed

(Chen&Sung 03^[13], YASS 04^[38], Sun&Buhler 06^[42], Zhou&Florea 07^[46], Yang&Zhang 08^[44])

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
```

Transition Constrained Seed

(Chen&Sung 03^[13], YASS 04^[38], Sun&Buhler 06^[42], Zhou&Florea 07^[46], Yang&Zhang 08^[44])

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
```

Transition Constrained Seed

(Chen&Sung 03^[13], YASS 04^[38], Sun&Buhler 06^[42], Zhou&Florea 07^[46], Yang&Zhang 08^[44])

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
```

Nucleic subset seeds

Example of seed pattern and seed alphabet

Nucleic subset seeds

Example of seed pattern and seed alphabet

Example

pattern: ###-#@@-##@#--#-@## , #@#@-##-#--@##-@###

Nucleic subset seeds

Example of seed pattern and seed alphabet

Example

pattern: ###-#@@-##@#--#-@## , #@#@-##-#--@##-@###

alphabet:

-	{AGCT}
@	{AG} {CT}
#	{A} {G} {C} {T}

Nucleic subset seeds

Example of seed pattern and seed alphabet

Example

pattern: ###-#@@-##@#--#-@## , #@#@-##-#--@##-@###

alphabet:

-	{AGCT}
@	{AG} {CT}
#	{A} {G} {C} {T}

Nucleic subset seeds

Example of seed pattern and seed alphabet

Example

pattern: ###-#@@-##@#--#-@## , #@#@-##-#--@##-@###

alphabet:

-	{AGCT}
@	{AG} {CT}
#	{A} {G} {C} {T}

Nucleic subset seeds

Example of seed pattern and seed alphabet

Example

pattern: ###-#@@-##@#--#-@## , #@#@-##-#--@##-@###

alphabet:

-	{AGCT}
@	{AG} {CT}
#	{A} {G} {C} {T}

Nucleic subset seeds

Example of seed pattern and seed alphabet

Nucleic subset seeds

Example of seed pattern and seed alphabet

Protein subset seeds

Example of seed pattern and seed alphabet

Protein subset seeds

Example of seed pattern and seed alphabet

Protein subset seeds

Example of seed pattern and seed alphabet

Example

pattern: JEQE , H064H , H0DI , I2005J , IBH

Protein subset seeds

Example of seed pattern and seed alphabet

Example

pattern: JEQE , H064H , H0DI , I2005J , IBH
alphabet:

```
0 {CFYWHMLIVPGQERKNDATS}
1 {CFYWHMLIV} {PGQERKNDATS}
2 {C} {FYWHMLIV} {PGQERKNDATS}
3 {C} {FYWHMLIV} {P} {QERKNDATS}
4 {C} {FYWH} {MLIV} {P} {QERKNDATS}
5 {C} {FYWH} {MLIV} {P} {GATS} {QERKND}
6 {C} {FYWH} {MLIV} {P} {G} {ATS} {QERKND}
7 {C} {FYWH} {MLIV} {P} {G} {ATS} {QERK} {ND}
8 {C} {FYW} {H} {MLIV} {P} {G} {ATS} {QERK} {ND}
9 {C} {FYW} {H} {MLIV} {P} {G} {A} {TS} {QERK} {ND}
A {C} {FYW} {H} {MLIV} {P} {G} {A} {TS} {QE} {RK} {ND}
B {C} {FYW} {H} {ML} {IV} {P} {G} {A} {TS} {QE} {RK} {ND}
C {C} {FYW} {H} {ML} {IV} {P} {G} {A} {TS} {QE} {RK} {N} {D}
D {C} {FYW} {H} {ML} {IV} {P} {G} {A} {T} {S} {QE} {RK} {N} {D}
E {C} {FY} {W} {H} {ML} {IV} {P} {G} {A} {T} {S} {QE} {RK} {N} {D}
F {C} {FY} {W} {H} {ML} {IV} {P} {G} {A} {T} {S} {Q} {E} {RK} {N} {D}
G {C} {FY} {W} {H} {M} {L} {IV} {P} {G} {A} {T} {S} {Q} {E} {RK} {N} {D}
H {C} {FY} {W} {H} {M} {L} {I} {V} {P} {G} {A} {T} {S} {Q} {E} {RK} {N} {D}
I {C} {F} {Y} {W} {H} {M} {L} {I} {V} {P} {G} {A} {T} {S} {Q} {E} {RK} {N} {D}
J {C} {F} {Y} {W} {H} {M} {L} {I} {V} {P} {G} {A} {T} {S} {Q} {E} {R} {K} {N} {D}
```

Protein subset seeds

Example of seed pattern and seed alphabet

Example

pattern: JE0E , H064H , H0DI , I2005J , IBH
alphabet:

```
0 {CFYWHMLIVPGQERKNDATS}
1 {C}{FYWHMLIV}{PGQERKNDATS}
2 {C}{FYWHMLIV}{PGQERKNDATS}
3 {C}{FYWHMLIV}{P}{GQERKNDATS}
4 {C}{FYWH}{MLIV}{P}{GQERKNDATS}
5 {C}{FYWH}{MLIV}{P}{GATS}{QERKND}
6 {C}{FYWH}{MLIV}{P}{G}{ATS}{QERKND}
7 {C}{FYWH}{MLIV}{P}{G}{ATS}{QERK}{ND}
8 {C}{FYW}{H}{MLIV}{P}{G}{ATS}{QERK}{ND}
9 {C}{FYW}{H}{MLIV}{P}{G}{A}{TS}{QERK}{ND}
A {C}{FYW}{H}{MLIV}{P}{G}{A}{TS}{QE}{RK}{ND}
B {C}{FYW}{H}{ML}{IV}{P}{G}{A}{TS}{QE}{RK}{ND}
C {C}{FYW}{H}{ML}{IV}{P}{G}{A}{TS}{QE}{RK}{N}{D}
D {C}{FYW}{H}{ML}{IV}{P}{G}{A}{T}{S}{QE}{RK}{N}{D}
E {C}{FY}{W}{H}{ML}{IV}{P}{G}{A}{T}{S}{QE}{RK}{N}{D}
F {C}{FY}{W}{H}{ML}{IV}{P}{G}{A}{T}{S}{Q}{E}{RK}{N}{D}
G {C}{FY}{W}{H}{M}{L}{IV}{P}{G}{A}{T}{S}{Q}{E}{RK}{N}{D}
H {C}{FY}{W}{H}{M}{L}{I}{V}{P}{G}{A}{T}{S}{Q}{E}{RK}{N}{D}
I {C}{F}{Y}{W}{H}{M}{L}{I}{V}{P}{G}{A}{T}{S}{Q}{E}{RK}{N}{D}
J {C}{F}{Y}{W}{H}{M}{L}{I}{V}{P}{G}{A}{T}{S}{Q}{E}{R}{K}{N}{D}
```

Protein subset seeds

Example of seed pattern and seed alphabet

Example

pattern: JEQE , H064H , HODI , I2005J , IBH
alphabet:

```
0 {CFYWHMLIVPGQERKNDATS}
1 {CFYWHMLIV} {PGQERKNDATS}
2 {C} {FYWHMLIV} {PGQERKNDATS}
3 {C} {FYWHMLIV} {P} {QERKNDATS}
4 {C} {FYWH} {MLIV} {P} {QERKNDATS}
5 {C} {FYWH} {MLIV} {P} {GATS} {QERKND}
6 {C} {FYWH} {MLIV} {P} {G} {ATS} {QERKND}
7 {C} {FYWH} {MLIV} {P} {G} {ATS} {QERK} {ND}
8 {C} {FYW} {H} {MLIV} {P} {G} {ATS} {QERK} {ND}
9 {C} {FYW} {H} {MLIV} {P} {G} {A} {TS} {QERK} {ND}
A {C} {FYW} {H} {MLIV} {P} {G} {A} {TS} {QE} {RK} {ND}
B {C} {FYW} {H} {ML} {IV} {P} {G} {A} {TS} {QE} {RK} {ND}
C {C} {FYW} {H} {ML} {IV} {P} {G} {A} {TS} {QE} {RK} {N} {D}
D {C} {FYW} {H} {ML} {IV} {P} {G} {A} {T} {S} {QE} {RK} {N} {D}
E {C} {FY} {W} {H} {ML} {IV} {P} {G} {A} {T} {S} {QE} {RK} {N} {D}
F {C} {FY} {W} {H} {ML} {IV} {P} {G} {A} {T} {S} {Q} {E} {RK} {N} {D}
G {C} {FY} {W} {H} {M} {L} {IV} {P} {G} {A} {T} {S} {Q} {E} {RK} {N} {D}
H {C} {FY} {W} {H} {M} {L} {I} {V} {P} {G} {A} {T} {S} {Q} {E} {RK} {N} {D}
I {C} {F} {Y} {W} {H} {M} {L} {I} {V} {P} {G} {A} {T} {S} {Q} {E} {RK} {N} {D}
J {C} {F} {Y} {W} {H} {M} {L} {I} {V} {P} {G} {A} {T} {S} {Q} {E} {R} {K} {N} {D}
```

Protein subset seeds

Example of seed pattern and seed alphabet

Example

pattern: JEQE , H064H , HODI , I2005J , IBH
alphabet:

```
0 {CFYWHMLIVPGQERKNDATS}
1 {CFYWHMLIV} {PGQERKNDATS}
2 {C} {FYWHMLIV} {PGQERKNDATS}
3 {C} {FYWHMLIV} {P} {GQERKNDATS}
4 {C} {FYWH} {MLIV} {P} {GQERKNDATS}
5 {C} {FYWH} {MLIV} {P} {GATS} {QERKND}
6 {C} {FYWH} {MLIV} {P} {G} {ATS} {QERKND}
7 {C} {FYWH} {MLIV} {P} {G} {ATS} {QERK} {ND}
8 {C} {FYW} {H} {MLIV} {P} {G} {ATS} {QERK} {ND}
9 {C} {FYW} {H} {MLIV} {P} {G} {A} {TS} {QERK} {ND}
A {C} {FYW} {H} {MLIV} {P} {G} {A} {TS} {QE} {RK} {ND}
B {C} {FYW} {H} {ML} {IV} {P} {G} {A} {TS} {QE} {RK} {ND}
C {C} {FYW} {H} {ML} {IV} {P} {G} {A} {TS} {QE} {RK} {N} {D}
D {C} {FYW} {H} {ML} {IV} {P} {G} {A} {T} {S} {QE} {RK} {N} {D}
E {C} {FY} {W} {H} {ML} {IV} {P} {G} {A} {T} {S} {QE} {RK} {N} {D}
F {C} {FY} {W} {H} {ML} {IV} {P} {G} {A} {T} {S} {Q} {E} {RK} {N} {D}
G {C} {FY} {W} {H} {M} {L} {IV} {P} {G} {A} {T} {S} {Q} {E} {RK} {N} {D}
H {C} {FY} {W} {H} {M} {L} {I} {V} {P} {G} {A} {T} {S} {Q} {E} {RK} {N} {D}
I {C} {F} {Y} {W} {H} {M} {L} {I} {V} {P} {G} {A} {T} {S} {Q} {E} {RK} {N} {D}
J {C} {F} {Y} {W} {H} {M} {L} {I} {V} {P} {G} {A} {T} {S} {Q} {E} {R} {K} {N} {D}
```

Protein subset seeds

Example of seed pattern and seed alphabet

Example

pattern: JE0E , H064H , H0DI , I2005J , IBH
alphabet:

```
0 {CFYWHMLIVPGQERKNDATS}
1 {CFYWHMLIV} {PGQERKNDATS}
2 {C} {FYWHMLIV} {PGQERKNDATS}
3 {C} {FYWHMLIV} {P} {QERKNDATS}
4 {C} {FYWH} {MLIV} {P} {QERKNDATS}
5 {C} {FYWH} {MLIV} {P} {GATS} {QERKND}
6 {C} {FYWH} {MLIV} {P} {G} {ATS} {QERKND}
7 {C} {FYWH} {MLIV} {P} {G} {ATS} {QERK} {ND}
8 {C} {FYW} {H} {MLIV} {P} {G} {ATS} {QERK} {ND}
9 {C} {FYW} {H} {MLIV} {P} {G} {A} {TS} {QERK} {ND}
A {C} {FYW} {H} {MLIV} {P} {G} {A} {TS} {QE} {RK} {ND}
B {C} {FYW} {H} {ML} {IV} {P} {G} {A} {TS} {QE} {RK} {ND}
C {C} {FYW} {H} {ML} {IV} {P} {G} {A} {TS} {QE} {RK} {N} {D}
D {C} {FYW} {H} {ML} {IV} {P} {G} {A} {T} {S} {QE} {RK} {N} {D}
E {C} {FY} {W} {H} {ML} {IV} {P} {G} {A} {T} {S} {QE} {RK} {N} {D}
F {C} {FY} {W} {H} {ML} {IV} {P} {G} {A} {T} {S} {Q} {E} {RK} {N} {D}
G {C} {FY} {W} {H} {M} {L} {IV} {P} {G} {A} {T} {S} {Q} {E} {RK} {N} {D}
H {C} {FY} {W} {H} {M} {L} {I} {V} {P} {G} {A} {T} {S} {Q} {E} {RK} {N} {D}
I {C} {F} {Y} {W} {H} {M} {L} {I} {V} {P} {G} {A} {T} {S} {Q} {E} {RK} {N} {D}
J {C} {F} {Y} {W} {H} {M} {L} {I} {V} {P} {G} {A} {T} {S} {Q} {E} {R} {K} {N} {D}
```

Protein subset seeds

Example of seed pattern and seed alphabet

Example

pattern: JEQE , H064H , HODI , I2005J , IBH
alphabet:

```
0 {CFYWHMLIVPGQERKNDATS}
1 {CFYWHMLIV} {PGQERKNDATS}
2 {C} {FYWHMLIV} {PGQERKNDATS}
3 {C} {FYWHMLIV} {P} {QERKNDATS}
4 {C} {FYWH} {MLIV} {P} {QERKNDATS}
5 {C} {FYWH} {MLIV} {P} {GATS} {QERKND}
6 {C} {FYWH} {MLIV} {P} {G} {ATS} {QERKND}
7 {C} {FYWH} {MLIV} {P} {G} {ATS} {QERK} {ND}
8 {C} {FYW} {H} {MLIV} {P} {G} {ATS} {QERK} {ND}
9 {C} {FYW} {H} {MLIV} {P} {G} {A} {TS} {QERK} {ND}
A {C} {FYW} {H} {MLIV} {P} {G} {A} {TS} {QE} {RK} {ND}
B {C} {FYW} {H} {ML} {IV} {P} {G} {A} {TS} {QE} {RK} {ND}
C {C} {FYW} {H} {ML} {IV} {P} {G} {A} {TS} {QE} {RK} {N} {D}
D {C} {FYW} {H} {ML} {IV} {P} {G} {A} {T} {S} {QE} {RK} {N} {D}
E {C} {FY} {W} {H} {ML} {IV} {P} {G} {A} {T} {S} {QE} {RK} {N} {D}
F {C} {FY} {W} {H} {ML} {IV} {P} {G} {A} {T} {S} {Q} {E} {RK} {N} {D}
G {C} {FY} {W} {H} {M} {L} {IV} {P} {G} {A} {T} {S} {Q} {E} {RK} {N} {D}
H {C} {FY} {W} {H} {M} {L} {I} {V} {P} {G} {A} {T} {S} {Q} {E} {RK} {N} {D}
I {C} {F} {Y} {W} {H} {M} {L} {I} {V} {P} {G} {A} {T} {S} {Q} {E} {RK} {N} {D}
J {C} {F} {Y} {W} {H} {M} {L} {I} {V} {P} {G} {A} {T} {S} {Q} {E} {R} {K} {N} {D}
```

Protein subset seeds

Example of seed pattern and seed alphabet

Example

pattern: JE OE , H O64H , H ODI , I 2005J , IBH
alphabet:

```
0 {CFYWHMLIVPGQERKNDATS}
1 {CFYWHMLIV} {PGQERKNDATS}
2 {C} {FYWHMLIV} {PGQERKNDATS}
3 {C} {FYWHMLIV} {P} {QERKNDATS}
4 {C} {FYWH} {MLIV} {P} {QERKNDATS}
5 {C} {FYWH} {MLIV} {P} {GATS} {QERKND}
6 {C} {FYWH} {MLIV} {P} {G} {ATS} {QERKND}
7 {C} {FYWH} {MLIV} {P} {G} {ATS} {QERK} {ND}
8 {C} {FYW} {H} {MLIV} {P} {G} {ATS} {QERK} {ND}
9 {C} {FYW} {H} {MLIV} {P} {G} {A} {TS} {QERK} {ND}
A {C} {FYW} {H} {MLIV} {P} {G} {A} {TS} {QE} {RK} {ND}
B {C} {FYW} {H} {ML} {IV} {P} {G} {A} {TS} {QE} {RK} {ND}
C {C} {FYW} {H} {ML} {IV} {P} {G} {A} {TS} {QE} {RK} {N} {D}
D {C} {FYW} {H} {ML} {IV} {P} {G} {A} {T} {S} {QE} {RK} {N} {D}
E {C} {FY} {W} {H} {ML} {IV} {P} {G} {A} {T} {S} {QE} {RK} {N} {D}
F {C} {FY} {W} {H} {ML} {IV} {P} {G} {A} {T} {S} {Q} {E} {RK} {N} {D}
G {C} {FY} {W} {H} {M} {L} {IV} {P} {G} {A} {T} {S} {Q} {E} {RK} {N} {D}
H {C} {FY} {W} {H} {M} {L} {I} {V} {P} {G} {A} {T} {S} {Q} {E} {RK} {N} {D}
I {C} {F} {Y} {W} {H} {M} {L} {I} {V} {P} {G} {A} {T} {S} {Q} {E} {RK} {N} {D}
J {C} {F} {Y} {W} {H} {M} {L} {I} {V} {P} {G} {A} {T} {S} {Q} {E} {R} {K} {N} {D}
```

Protein subset seeds

Implied Problems

Protein subset seeds

Implied Problems

pattern: how to choose “good” seed patterns ?

why choosing JE0E,H064H,H0DI,I205J,IBH ?

Protein subset seeds

Implied Problems

pattern: how to choose “good” seed patterns ?

why choosing JE0E,H064H,H0DI,I205J,IBH ?

alphabet: how to choose “good” alphabets ?

why choosing

```
0 {CFYWHMLIVPGQERKNDATS}
1 {CFYWHMLIV} {PGQERKNDATS}
2 {C} {FYWHMLIV} {PGQERKNDATS}
3 {C} {FYWHMLIV} {P} {GQERKNDATS}
4 {C} {FYWH} {MLIV} {P} {GQERKNDATS}
5 {C} {FYWH} {MLIV} {P} {GATS} {QERKND}
6 {C} {FYWH} {MLIV} {P} {G} {ATS} {QERKND}
7 {C} {FYWH} {MLIV} {P} {G} {ATS} {QERK} {ND}
8 {C} {FYW} {H} {MLIV} {P} {G} {ATS} {QERK} {ND}
9 {C} {FYW} {H} {MLIV} {P} {G} {A} {TS} {QERK} {ND}
A {C} {FYW} {H} {MLIV} {P} {G} {A} {TS} {QE} {RK} {ND}
B {C} {FYW} {H} {ML} {IV} {P} {G} {A} {TS} {QE} {RK} {ND}
C {C} {FYW} {H} {ML} {IV} {P} {G} {A} {TS} {QE} {RK} {N} {D}
D {C} {FYW} {H} {ML} {IV} {P} {G} {A} {T} {S} {QE} {RK} {N} {D}
E {C} {FY} {W} {H} {ML} {IV} {P} {G} {A} {T} {S} {QE} {RK} {N} {D}
F {C} {FY} {W} {H} {ML} {IV} {P} {G} {A} {T} {S} {Q} {E} {RK} {N} {D}
G {C} {FY} {W} {H} {M} {L} {IV} {P} {G} {A} {T} {S} {Q} {E} {RK} {N} {D}
H {C} {FY} {W} {H} {M} {L} {I} {V} {P} {G} {A} {T} {S} {Q} {E} {RK} {N} {D}
I {C} {F} {Y} {W} {H} {M} {L} {I} {V} {P} {G} {A} {T} {S} {Q} {E} {RK} {N} {D}
J {C} {F} {Y} {W} {H} {M} {L} {I} {V} {P} {G} {A} {T} {S} {Q} {E} {R} {K} {N} {D}
```

Protein subset seeds

Implied Problems

Protein subset seeds

Implied Problems

pattern: how to choose “good” seed patterns ?

No efficient method known (?^[37, 32])

Use *hill climbing*^[42, 21, 47] heuristics

Protein subset seeds

Implied Problems

pattern: how to choose “good” seed patterns ?

No efficient method known (?^[37, 32])

Use *hill climbing*^[42, 21, 47] heuristics

alphabet: how to choose “good” alphabets ?

No efficient algorithm known (?)

Use 3 different *clustering* methods ...

Protein subset seeds

Implied Problems

pattern: how to choose “good” seed patterns ?

No efficient method known (?^[37, 32])

Use *hill climbing*^[42, 21, 47] heuristics

alphabet: how to choose “good” alphabets ?

No efficient algorithm known (?)

Use 3 different *clustering* methods ...

Protein subset seeds

Related work on seed alphabets

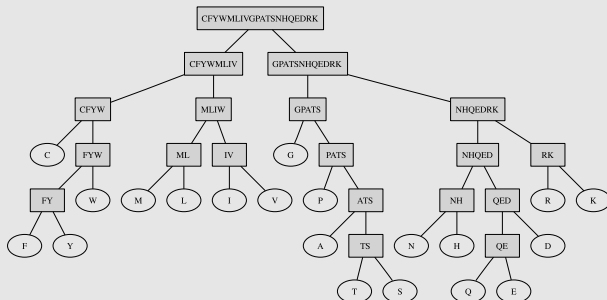
Protein alphabet Reduction: Li et al. 2003^[29], Murphy et al. 2000^[35]

Protein subset seeds

Related work on seed alphabets

Protein alphabet Reduction: Li et al. 2003^[29], Murphy et al. 2000^[35]

Example



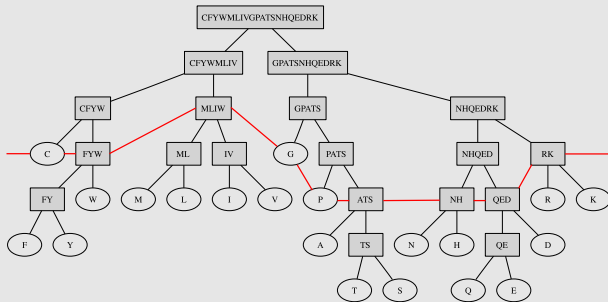
Tree defined in Li et al. 2003^[29]

Protein subset seeds

Related work on seed alphabets

Protein alphabet Reduction: Li et al. 2003^[29], Murphy et al. 2000^[35]

Example



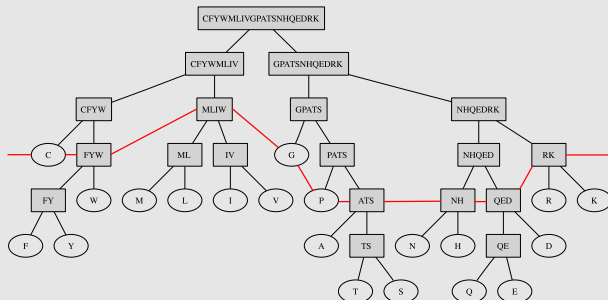
A **cut** in the tree defines a **seed letter**
(here {C} {FYW} {MLIW} {G} {P} {ATS} {NH} {QED} {RK})

Protein subset seeds

Related work on seed alphabets

Protein alphabet Reduction: Li et al. 2003^[29], Murphy et al. 2000^[35]

Example



“cut” clustering → seed alphabet (transitive-predefined)

Protein subset seeds

Clustering algorithms for seed alphabets

Transitive :

`transitive-predefined`: use the previous tree^[29].

`transitive-ab-initio`: BLOSUM pairing frequencies.
bridge likelihood optimization.
greedy algorithm.

`non-tree-transitive`: same as `transitive-ab-initio`
without hierarchical clustering constraint.

Protein subset seeds

Clustering algorithms for seed alphabets

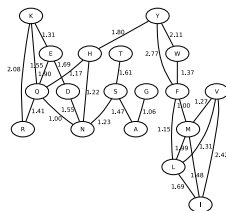
Transitive :

transitive-predefined: use the previous tree^[29].

transitive-ab-initio: BLOSUM pairing frequencies.
bridge likelihood optimization.
greedy algorithm.

non-tree-transitive: same as transitive-ab-initio
without hierarchical clustering constraint.

Non-Transitive : use the BLOSUM likelihood graph.



Protein subset seeds

Clustering algorithms for seed alphabets

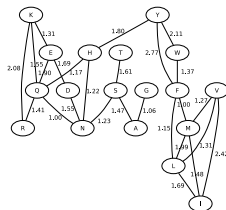
Transitive :

transitive-predefined: use the previous tree^[29].

transitive-ab-initio: BLOSUM pairing frequencies.
bridge likelihood optimization.
greedy algorithm.

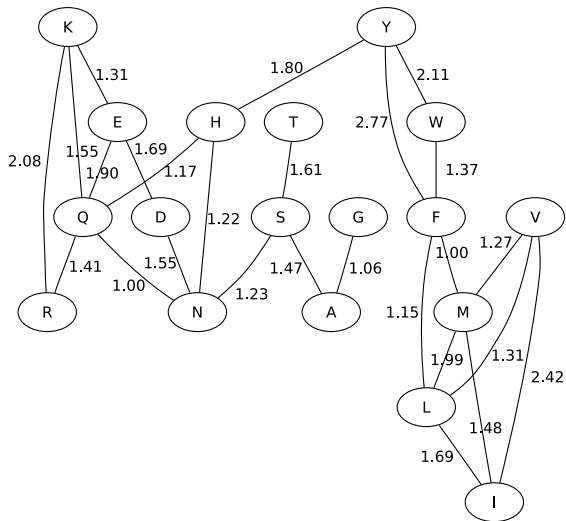
non-tree-transitive: same as transitive-ab-initio
without hierarchical clustering constraint.

Non-Transitive : use the BLOSUM likelihood graph.



Protein subset seeds

Clustering algorithms for seed alphabets



Protein subset seeds

Related work on seed models

Protein subset seeds

Related work on seed models

Protein seed models :

- 1 BLASTP^[2],
- 2 Vector Seeds (*Brown 2004^[5], tPatternHunter 2005^[24]*)

Protein seed models :

- 1 BLASTP^[2],
- 2 Vector Seeds (*Brown 2004*^[5], *tPatternHunter 2005*^[24])

Definition

- 1 BLASTP : **hit** at positions i on sequence A and j on sequence B iif

$$\sum_{p=0}^2 \text{BLOSUM}[A_{[i+p]}, B_{[j+p]}] \geq 11$$

$$\overline{\text{###}} \geq 11$$

Protein subset seeds

Related work on seed models

Protein seed models :

- 1 BLASTP^[2],
- 2 Vector Seeds (*Brown 2004^[5], tPatternHunter 2005^[24]*)

Definition

- 1 BLASTP : **hit** at positions i on sequence A and j on sequence B iif

$$\sum_{p=0}^2 \text{BLOSUM}[A_{[i+p]}, B_{[j+p]}] \geq 11$$

$$\overline{\text{###}} \geq 11$$

- 2 Vector Seeds : *extension using spaced seed shapes*

$$\overline{\text{###}} \geq 11 \rightarrow \overline{\text{\#-\#\#-\#}} \geq 18$$

Protein seed models :

Definition

- ① BLASTP : **hit** at positions i on sequence A and j on sequence B iff

$$\sum_{p=0}^2 \text{BLOSUM}[A_{[i+p]}, B_{[j+p]}] \geq 11$$

$$\overline{\text{###}} \geq 11$$

- ② Vector Seeds : *extension using spaced seed shapes*

$$\overline{\text{###}} \geq 11 \rightarrow \overline{\text{\#-\#\#-\#}} \geq 18$$

*Need to precompute, for each word, the list of **neighbor** words*

Protein subset seeds

Seed design using *iedera*

the *iedera* software was used to :

- 1 compute sensitivity/selectivity of
 - BLASTP seeds,
 - Vector seeds,
 - Protein subset seeds.
- 2 optimize the shapes of
 - Vector seeds,
 - Protein subset seeds.

with the *hill climbing*^[42, 21, 47] method.

<http://bioinfo.lifl.fr/yass/iedera.php>



Protein subset seeds

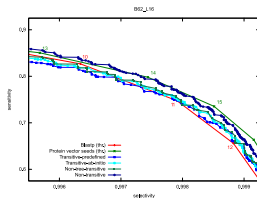
Experiments

- Seed design \rightarrow (*sensitivity,selectivity*) pair for each seed *pattern*.

Protein subset seeds

Experiments

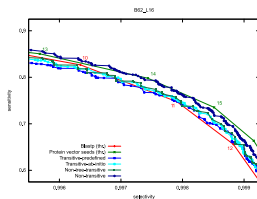
- 1 Seed design \rightarrow (*sensitivity, selectivity*) pair for each seed *pattern*.
- 2 Theoretical ROC curves.



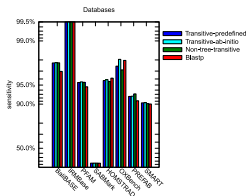
Protein subset seeds

Experiments

- 1 Seed design \rightarrow (*sensitivity, selectivity*) pair for each seed *pattern*.
- 2 Theoretical ROC curves.

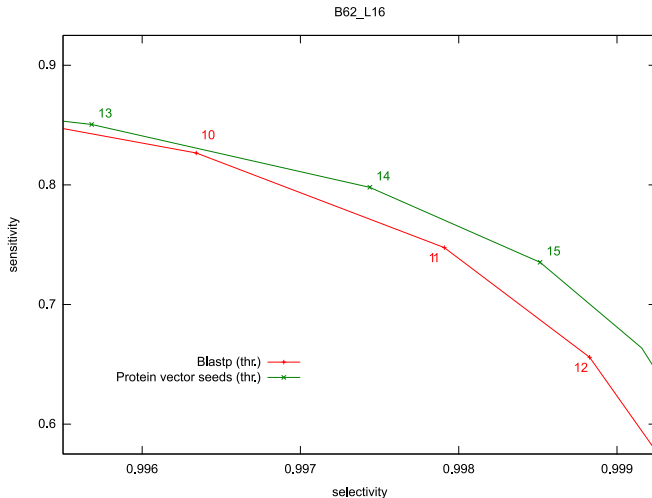


- 3 Real DATABASE hits.



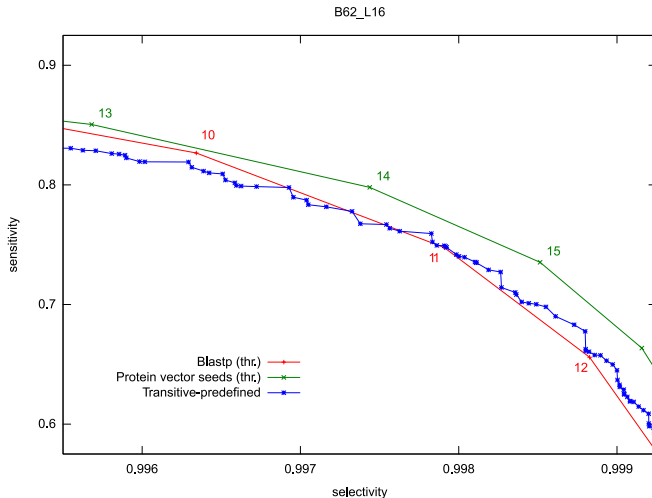
Protein subset seeds

Results on theoretical model (length 16)



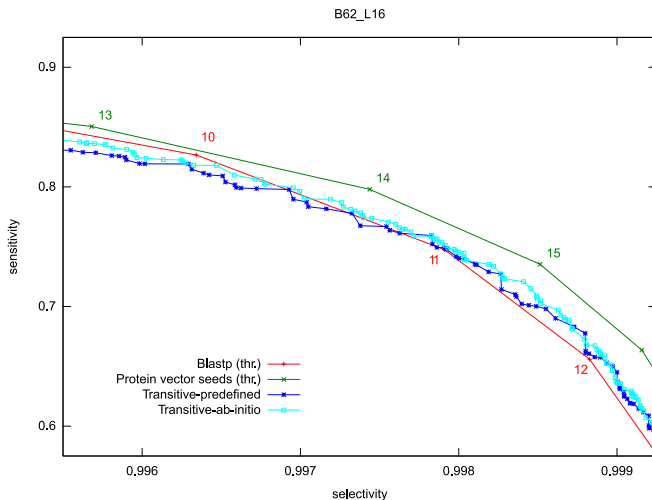
Protein subset seeds

Results on theoretical model (length 16)



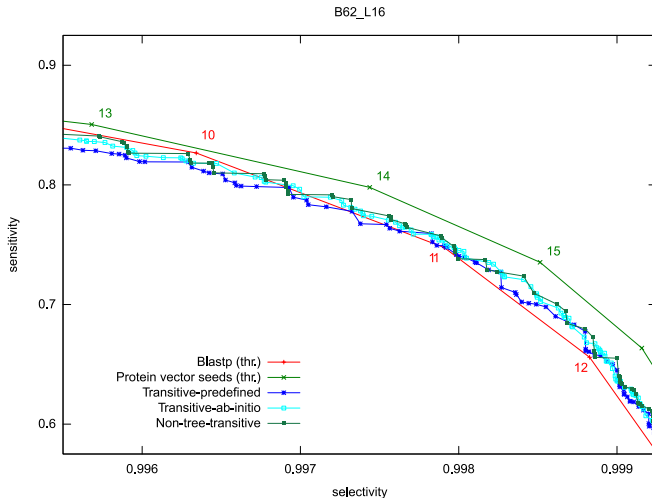
Protein subset seeds

Results on theoretical model (length 16)



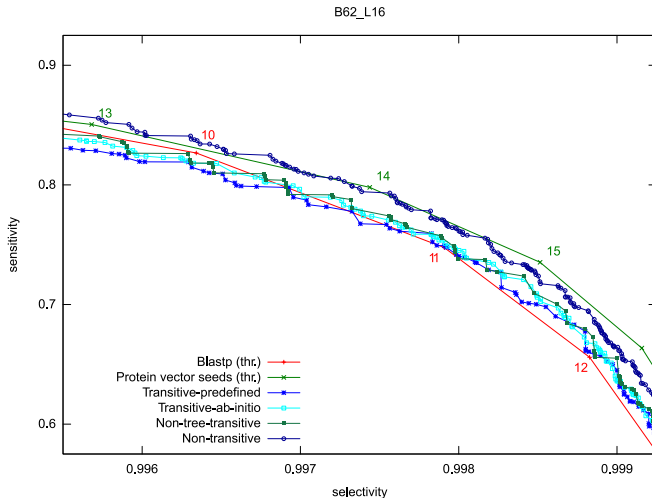
Protein subset seeds

Results on theoretical model (length 16)



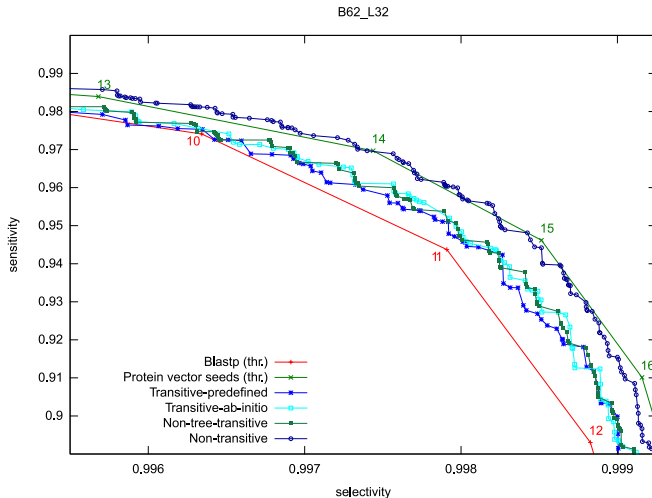
Protein subset seeds

Results on theoretical model (length 16)



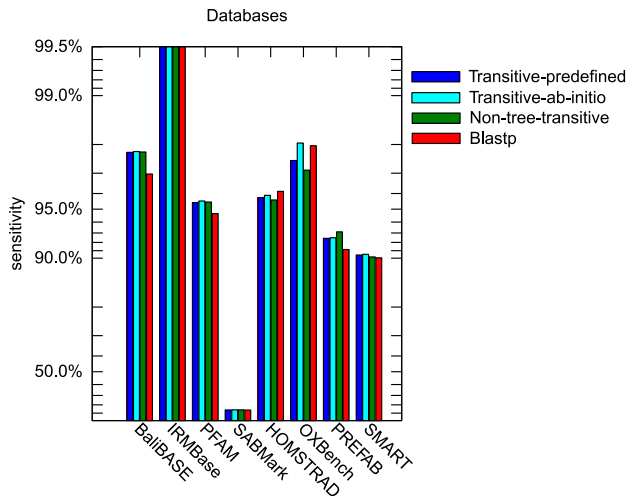
Protein subset seeds

Results on theoretical model (length 32)



Protein subset seeds

Results on real datasets



Protein subset seeds

conclusion & open questions

- 1 a *simple* way to index proteins sequences, based on carefully designed seeds.
- 2 a *cache efficient* way to perform protein similarity search.
already probed > 20% faster on FPGA device with on-board Flash memory [39].

Protein subset seeds

conclusion & open questions

- ① a *simple* way to index proteins sequences, based on carefully designed seeds.
- ② a *cache efficient* way to perform protein similarity search.
already probed > 20% faster on FPGA device with on-board Flash memory [39].

However, during the *precomputing step* ...

Protein subset seeds

conclusion & open questions

- 1 a *simple* way to index proteins sequences, based on carefully designed seeds.
- 2 a *cache efficient* way to perform protein similarity search.
already probed > 20% faster on FPGA device with on-board Flash memory [39].

However, during the *precomputing step* ...

- seed *design* is a *CPU-consuming task*.
- seed alphabet *choice* is also *non-trivial*.

References I



S. Altschul, W. Gish, W. Miller, E. Myers, and D. Lipman.

Basic Local Alignment Search Tool.

Journal of Molecular Biology, 215:403–410, 1990.



S. Altschul, T. Madden, A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. Lipman.

Gapped BLAST and PSI-BLAST: a new generation of protein database search programs.

Nucleic Acids Research, 25(17):3389–3402, 1997.



B. Brejová, D. Brown, and T. Vinar.

Optimal spaced seeds for Hidden Markov Models, with application to homologous coding regions.

In M. Crochemore R. Baeza-Yates, E. Chavez, editor, *Proceedings of the 14th Symposium on Combinatorial Pattern Matching (CPM), Morelia (Mexico)*, volume 2676 of *Lecture Notes in Computer Science*, pages 42–54. Springer-Verlag, June 2003.



B. Brejová, D. Brown, and T. Vinar.

Vector seeds: an extension to spaced seeds allows substantial improvements in sensitivity and specificity.

In *WABI*, volume 2812 of *Lecture Notes in Computer Science*, pages 39–54. Springer-Verlag, September 2003.



D. Brown.

Multiple vector seeds for protein alignment.

In I. Jonassen and J. Kim, editors, *Proceedings of the 4th International Workshop in Algorithms in Bioinformatics (WABI), Bergen (Norway)*, volume 3240 of *Lecture Notes in Bioinformatics*, pages 170–181. Springer-Verlag, September 2004.

References II



D. Brown, M. Li, and B. Ma.

A tutorial of recent developments in the seeding of local alignment.
Journal of Bioinformatics and Computational Biology, 2(4):819–842, 2004.



D. G. Brown.

Bioinformatics Algorithms: Techniques and Applications, chapter A survey of seeding for sequence alignment, pages 126–152.
Wiley-Interscience (I. Mandoiu, A. Zelikovsky), Feb. 2008.



J. Buhler.

Efficient large-scale sequence comparison by locality-sensitive hashing.
Bioinformatics, 17(5):419–428, 2001.



J. Buhler, U. Keich, and Y. Sun.

Designing seeds for similarity search in genomic DNA.
In *Proceedings of the 7th Annual International Conference on Computational Molecular Biology (RECOMB), Berlin (Germany)*, pages 67–75. ACM Press, April 2003.



S. Burkhardt and J. Kärkkäinen.

Better filtering with gapped q-grams.
In *Proceedings of the 12th Symposium on Combinatorial Pattern Matching (CPM)*, volume 2089 of *Lecture Notes in Computer Science*, pages 73–85. Springer-Verlag, July 2001.



S. Burkhardt and J. Kärkkäinen.

Better filtering with gapped q-grams.
Fundamenta Informaticae, 56(1-2):51–70, 2003.
Preliminary version in *Combinatorial Pattern Matching* 2001.

References III



A. Califano and I. Rigoutsos.

Flash: A fast look-up algorithm for string homology.

In *Proceedings of the 1st International Conference on Intelligent Systems for Molecular Biology (ISMB)*, pages 56–64, July 1993.



W. Chen and W. Sung.

On half gapped seed.

Genome Informatics, 14:176–185, 2003.

preliminary version in the 14th International Conference on Genome Informatics (GIW).



K.P. Choi and L. Zhang.

Sensitivity analysis and efficient method for identifying optimal spaced seeds.

Journal of Computer and System Sciences, 68(1):22–40, 2004.



M. Csürös.

Performing local similarity searches with variable length seeds.

In S.C. Sahinalp, S. Muthukrishnan, and U. Dogrusoz, editors, *Proceedings of the 15th Annual Combinatorial Pattern Matching Symposium (CPM), Istanbul (Turkey)*, volume 3109 of *Lecture Notes in Computer Science*, pages 373–387. Springer-Verlag, 2004.



M. Csürös and B. Ma.

Rapid homology search with two-stage extension and daughter seeds.

In *Proceedings of the 11th International Computing and Combinatorics Conference (COCOON)*, volume 3595 of *Lecture Notes in Computer Science*, pages 104–114. Springer-Verlag, 2005.

References IV



M. Csürös and B. Ma.
Rapid homology search with neighbor seeds.
Algorithmica, 48(2):187–202, June 2007.



M. Farach-Colton, G. Landau, S.C. Sahinalp, and D. Tsur.
Optimal spaced seeds for faster approximate string matching.
Journal of Computer and System Sciences, 73(7):1035–1044, March 2007.



M. Fontaine, S. Burkhardt, and J. Kärkkäinen.
BDD-based analysis of gapped q -gram filters.
International Journal of Foundations of Computer Science, 6(16):1121–1134, 2005.



X. Gao, S.C. Li, and Y. Lu.
New algorithms for the spaced seeds.
In *Frontiers of Algorithmic Workshop 2007 (FAW2007)*, volume 4613 of *Lecture Notes in Computer Science*, pages 51–61. Springer-Verlag, 2007.



L. Ilie and S. Ilie.
Fast computation of good multiple spaced seeds.
In *Proceedings of the 7th International Workshop in Algorithms in Bioinformatics (WABI), Philadelphia (USA)*, volume 4645 of *Lecture Notes in Bioinformatics*, pages 346–358. Springer-Verlag, Sept. 2007.



L. Ilie and S. Ilie.
Long spaced seeds for finding similarities between biological sequences.
In *Proceedings of the 2nd International Conference on Bioinformatics & Computational Biology (BIOCOMP)*, pages 3–8, 2007.



U. Keich, M. Li, B. Ma, and J. Tromp.

On spaced seeds for similarity search.

Discrete Applied Mathematics, 138(3):253–263, 2004.

preliminary version in 2002.



D. Kisman, M. Li, B. Ma, and W. Li.

tPatternhunter: gapped, fast and sensitive translated homology search.

Bioinformatics, 21(4):542–544, February 2005.



H. Kong.

Generalized correlation functions and their applications in selection of optimal multiple spaced seeds for homology search.

Journal of Computational Biology, 14(2):238–254, Mar. 2007.



G. Kucherov, L. Noé, and Y. Ponty.

Estimating seed sensitivity on homogeneous alignments.

In *Proceedings of the IEEE 4th Symposium on Bioinformatics and Bioengineering (BIBE), May 19–21, 2004, Taichung (Taiwan)*, pages 387–394. IEEE Computer Society Press, April 2004.



G. Kucherov, L. Noé, and M. Roytberg.

Multiseed lossless filtration.

IEEE Transactions on Computational Biology and Bioinformatics (TCBB), 2(1):51–61, january 2005.

References VI



M. Li, B. Ma, D. Kisman, and J. Tromp.

PatternHunter II: Highly sensitive and fast homology search.

Journal of Bioinformatics and Computational Biology, 2(3):417–439, 2004.
(earlier version in GIW 2003).



T. Li, K. Fan, J. Wang, and W. Wang.

Reduction of protein sequence complexity by residue grouping.

Journal of Protein Engineering, 16:323–330, 2003.



D. Lipman and W. Pearson.

Rapid and sensitive protein similarity searches.

Science, 227:1435–1441, 1985.



B. Ma, J. Tromp, and M. Li.

PatternHunter: Faster and more sensitive homology search.

Bioinformatics, 18(3):440–445, 2002.



B. Ma and H. Yao.

Seed optimization is no easier than optimal Golomb ruler design.

In *Proceedings of the 6th Asia Pacific Bioinformatics Conference (APBC)*, pages 133–144,
january 2008.



D. Mak and G. Benson.

All hits all the time: parameter free calculation of seed sensitivity.

In *Proceedings of the 5th Asia Pacific Bioinformatics Conference (APBC)*, pages 317–326,
2007.

References VII



D. Mak, Y. Gelfand, and G. Benson.
Indel seeds for homology search.
Bioinformatics, 22(14):e341–e349, 2006.



L. Murphy, A. Wallqvist, and R Levy.
Simplified amino acid alphabets for protein fold recognition and implications for folding.
Journal of Protein Engineering, 13:149–152, 2000.



P. Nicode me, B. Salvy, and P. Flajolet.
Motif statistics.
Theoretical Computer Science, 287(2):593–617, 2002.



F. Nicolas and E. Rivals.
Hardness of optimal spaced seed design.
In A. Apostolico, M. Crochemore, and K. Park, editors, *Proceedings of the 16th Annual Symposium on Combinatorial Pattern Matching (CPM), Jeju Island (Korea)*, volume 3537 of *Lecture Notes in Computer Science*, pages 144–155. Springer-Verlag, 2005.
(earlier version in RECOMB 2004).



L. No e and G. Kucherov.
Improved hit criteria for DNA local alignment.
BMC Bioinformatics, 5(149), october 2004.

References VIII



P. Peterlongo, L. Noé, D. Lavenier, G. Georges, J. Jacques, G. Kucherov, and M. Giraud.
Protein similarity search with subset seeds on a dedicated reconfigurable hardware.
In R. Wyrzykowski, J. Dongarra, K. Karczewski, and J. Wasniewski, editors, *Proceedings of the 2nd Workshop on Parallel Bio-Computing Workshop (PBC'07), Gdansk (Poland), September 9-12, 2007*, volume 4967 of *Lecture Notes in Computer Science*, pages 1240–1248. Springer Verlag, 2008.



S. Schwartz, J. Kent, A. Smit, Z. Zhang, R. Baertsch, R. Hardison, D. Haussler, and W. Miller.
Human–mouse alignments with BLASTZ.
Genome Research, 13:103–107, 2003.



Y. Sun and J. Buhler.
Designing multiple simultaneous seeds for DNA similarity search.
In *Proceedings of the 8th Annual International Conference on Computational Molecular Biology (RECOMB), San Diego (California)*, pages 76–84, March 2004.



Y Sun and J. Buhler.
Choosing the best heuristic for seeded alignment of DNA sequences.
BMC Bioinformatics, 7(133), march 2006.



J. Xu, D. Brown, M. Li, and B. Ma.
Optimizing multiple spaced seeds for homology search.
Journal of Computational Biology, 13(7):1355–1368, 2006.



J. Yang and L. Zhang.

Run probability of high-order seed patterns and its applications to finding good transition seeds.

In *Proceedings of the 6th Asia Pacific Bioinformatics Conference (APBC)*, pages 123–132, january 2008.



L. Zhang.

Superiority of spaced seeds for homology search.

IEEE Transactions on Computational Biology and Bioinformatics (IEEE TCBB), 4(3):496–505, 2007.



L. Zhou and L. Florea.

Designing sensitive and specific spaced seeds for cross-species mRNA-to-genome alignment.

Journal of Computational Biology, 14(2):113–130, Mar. 2007.



L. Zhou, J. Stanton, and L. Florea.

Universal seeds for cDNA-to-genome comparison.

BMC Bioinformatics, 9(36), 2008.