SpEED
Fast combinatorial computation of good seeds for genome alignment

Lucian Ilie

Department of Computer Science
University of Western Ontario
A Revolution in Biology

- computer revolution – all areas affected
- biology – discoveries are produced with incredible speed
- great ability to create, store and process huge amounts of data
- bioinformatics algorithms
- biological molecules are seen as sequences of letters
  AGCTTTTTCATTCTGACTGCAACGGGCAATATGTCTCTCT...
- sequence similarity often implies functional similarity
- finding similarities – the most important problem in bioinformatics
- also called local alignment
Similarity search – Dynamic Programming

- [Smith, Waterman, 1981]
- example: local alignment of AGGCGG and GGGCTGGCGA

<table>
<thead>
<tr>
<th></th>
<th>G</th>
<th>G</th>
<th>G</th>
<th>C</th>
<th>T</th>
<th>G</th>
<th>G</th>
<th>C</th>
<th>G</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>7</td>
</tr>
</tbody>
</table>

optimal local alignment: G G G C T G G C G A

A G G C G G
Similarity search – Dynamic Programming

- [Smith, Waterman, 1981]
- Example: local alignment of AGGCGG and GGGCTGGCGA

```
<table>
<thead>
<tr>
<th></th>
<th>G</th>
<th>G</th>
<th>G</th>
<th>C</th>
<th>T</th>
<th>G</th>
<th>G</th>
<th>C</th>
<th>G</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>7</td>
</tr>
</tbody>
</table>
```

- Optimal local alignment: G G G C T G G C G A
- Problem: quadratic time – can handle only short sequences
  - Too slow for whole genome alignment or large database search
Similarity Search – BLAST

- **BLAST** (Basic Local Alignment Search Tool) – heuristic program
- **hit and extend**
  - hit: search for 11 consecutive matches
  - extend: check the surrounding area for similarity
Similarity Search – BLAST

- **BLAST** (Basic Local Alignment Search Tool) – heuristic program
- **hit and extend**
  - **hit**: search for 11 consecutive matches
  - **extend**: check the surrounding area for similarity

```
CCCTTGCTGGCAATATGTCTCTGTGGATTAAAAAAACAGTGTCACTGATACCACTACTCCCAGTTCCAGAT
CAATATCAGCCTAAAAAAAGAGTGTCTGATACCAACCTGCTTCTGAACGTGGTACCCTGCGTGAGTAAATTAAT
```
**Similarity Search – BLAST**

- **BLAST** (Basic Local Alignment Search Tool) – heuristic program
- **hit and extend**
  - hit: search for 11 consecutive matches
  - extend: check the surrounding area for similarity

```plaintext
CCCTTGCTGGCAATATGTCCTCTGTGGATTTAAAAAACAGTGTCAGATACCCAGCTACTCCCAGTTCCAGATCAATATCAGCCTAAATAAAAGAGTGTCCTGATACCCAGCTTCTGAACTGGTTACCTGCCGTCGAGTAAATTAAAAATT
```

- the most widely used bioinformatics software
  - [Altschul, Gish, Miller, Myers, Lipman, 1990]  
    - the initial BLAST paper – 34,733 citations
  - [Altschul, Madden, Schäffer, Zhang, Zhang, Miller, Lipman, 1997]  
    - gapped BLAST paper – 35,615 citations
Similarity Search – Seeds

- **seed** – the 11 consecutive matches of BLAST: 11111111111

- spaced seed – matches not consecutive: 111*1**1*1**11*111

1 = match, * = don't care; weight = number of 1's

- spaced seeds have higher chance of finding similarities

- [Califano, Rigoutsos, 1993], [Pevzner, Waterman, 1995], [Buhler, 2001], [Ma, Tromp, Li, 2002] – optimized spaced seeds

- PatternHunter's seed: 111*1**1*1**11*111

- multiple spaced seeds – much better

- also [Brejova, Brown, Vinar, 2003], [Buhler, Keich, Sun, 2003]

- PatternHunterII: 16 seeds of weight 11

- more space needed (more hash tables)

- space reduction – neighbour seeds – [Cs˝ ur˝ os, Ma, 2007]
Similiarity Search – Seeds

- **seed** – the 11 consecutive matches of BLAST: 11111111111
- **spaced seed** – matches not consecutive: 111*1**1*1**11*111
  - 1 = match, * = don’t care; **weight** = number of 1’s

- spaced seeds have higher chance of finding similarities
- [Califano, Rigoutsos, 1993], [Pevzner, Waterman, 1995], [Buhler, 2001], [Ma, Tromp, Li, 2002] – optimized spaced seeds
- PatternHunter’s seed: 111*1**1*1**11*111
- multiple spaced seeds – much better
  - also [Brejova, Brown, Vinar, 2003], [Buhler, Keich, Sun, 2003]
- PatternHunterII: 16 seeds of weight 11
  - more space needed (more hash tables)
- space reduction – neighbour seeds – [Csürös, Ma, 2007]
Similarity Search – Seeds

- **seed** – the 11 consecutive matches of BLAST: 11111111111
- **spaced seed** – matches not consecutive: 111*1**1*1**11*111
  - 1 = match, * = don’t care; weight = number of 1’s
- spaced seeds have higher chance of finding similarities
  - [Califano, Rigoutsos, 1993], [Pevzner, Waterman, 1995], [Buhler, 2001]
Similarity Search – Seeds

- **seed** – the 11 consecutive matches of BLAST: 11111111111
- **spaced seed** – matches not consecutive: 111*1**1*1**11*111
  - 1 = match, * = don’t care; weight = number of 1’s
- spaced seeds have higher chance of finding similarities
  - [Califano, Rigoutsos, 1993], [Pevzner, Waterman, 1995], [Buhler, 2001]
  - [Ma, Tromp, Li, 2002] – optimized spaced seeds
    - PatternHunter’s seed: 111*1**1*1**11*111
Similarity Search – Seeds

- **seed** – the 11 consecutive matches of BLAST: 11111111111
- **spaced seed** – matches not consecutive: 111*1**1*1**11*111
  - 1 = match, * = don’t care; **weight** = number of 1’s
- spaced seeds have higher chance of finding similarities
  - [Califano, Rigoutsos, 1993], [Pevzner, Waterman, 1995], [Buhler, 2001]
- [Ma, Tromp, Li, 2002] – optimized spaced seeds
  - PatternHunter’s seed: 111*1**1*1**11*111
  - multiple spaced seeds – much better
  - also [Brejova, Brown, Vinar, 2003], [Buhler, Keich, Sun, 2003]
Similarity Search – Seeds

- **seed** – the 11 consecutive matches of BLAST: 11111111111
- **spaced seed** – matches not consecutive: 111*1**1*1**11*111
  - 1 = match, * = don’t care; weight = number of 1’s
- spaced seeds have higher chance of finding similarities
  - [Califano, Rigoutsos, 1993], [Pevzner, Waterman, 1995], [Buhler, 2001]
  - [Ma, Tromp, Li, 2002] – optimized spaced seeds
    - PatternHunter’s seed: 111*1**1*1**11*111
    - multiple spaced seeds – much better
      - also [Brejova, Brown, Vinar, 2003], [Buhler, Keich, Sun, 2003]
- PatternHunterII: 16 seeds of weight 11
Similarity Search – Seeds

- **seed** – the 11 consecutive matches of BLAST: 11111111111
- **spaced seed** – matches not consecutive: 111*1**1*1**11*111
  - 1 = match, * = don’t care; **weight** = number of 1’s
- spaced seeds have higher chance of finding similarities
  - [Califano, Rigoutsos, 1993], [Pevzner, Waterman, 1995], [Buhler, 2001]
- [Ma, Tromp, Li, 2002] – optimized spaced seeds
  - PatternHunter’s seed: 111*1**1*1**11*111
  - multiple spaced seeds – much better
  - also [Brejova, Brown, Vinar, 2003], [Buhler, Keich, Sun, 2003]
- **PatternHunterII**: 16 seeds of weight 11
  - more space needed (more hash tables)
  - space reduction – neighbour seeds – [Csűrös, Ma, 2007]
Sensitivity of a Seed

- [Li, Ma, Kisman, Tromp, 2004]

DNA seq. $S_1$: A C G A G G C A C T G T A T G T A T A T C T A

DNA seq. $S_2$: A G T A G G C A A T G C A T T T A A A T C T C

matches/mism.: $\neq \neq \neq \neq \neq \neq \neq \neq \neq \neq \neq \neq \neq \neq \neq \neq \neq \neq \neq \neq \neq \neq \neq \neq \neq \neq \neq \neq \neq \neq \neq \neq $  

Bernoulli seq. $R$: 1 0 0 1 1 1 1 1 0 1 1 0 1 0 1 1 0 1 1 1 1 0

spaced seed $s$: 1 1 1 * 1 * * 1 * 1 * * 1 * 1 1 1 1

- an example of a hit using PatternHunter’s seed

- similarity $= \text{probability } p \text{ of equality (probab. of 1 in } R)$

- sensitivity $= \text{probability that } s \text{ hits } R \text{ at or before position } n$
  - depends on $p$ and length of random region $n$
Sensitivity of a Seed

- [Li, Ma, Kisman, Tromp, 2004]

DNA seq. $S_1$: A C G A G G C A C T G T A T G T A T A T C T A
DNA seq. $S_2$: A G T A G G C A A T G C A T T T A A A T C T C
matches/mism.: $= \neq = = = = = \neq = = \neq = = \neq = = \neq = = = = = \neq$
Bernoulli seq. $R$: 1 0 0 1 1 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 1 1 0
spaced seed $s$: 1 1 1 * 1 * * 1 * 1 * * 1 1 * 1 1 1

- an example of a hit using PatternHunter’s seed
- similarity $= \text{probability } p \text{ of equality (probab. of 1 in } R)$
- sensitivity $= \text{probability that } s \text{ hits } R \text{ at or before position } n$
  - depends on $p$ and length of random region $n$
- for $p = 0.70, n = 64$:
  - BLAST: 0.30; PatternHunter: 0.466; PatternHunterII: 0.924
Speed vs Sensitivity

- two competing goals for the initial matches (hits)
  - high sensitivity – find many similar regions
  - high speed – find few non-similar regions (random hits)
- more initial matches (higher seed weight)
  - higher speed
  - many similar regions missed
  - lower sensitivity
- fewer initial matches (lower seed weight)
  - higher sensitivity
  - too many random hits
  - lower speed
- PatternHunter – increases both speed and sensitivity
Papers on (Multiple) Spaced Seeds

- advantages of spaced seeds over consecutive ones
  - [BKS03, KLMT04, ChZh04, LMZ06]
- relevant problems are NP-hard
  - [LMKT04, LMZ06, MaYa07]
- exact (exponential) algorithms for computing sensitivity
  - [BKS03, LMKT04, KLMT04, ChZh04, CZZ04]
- polynomial time approximation schemes
  - [LMZ06]
- heuristic algorithms
  - [LMKT04, CZZ04, Yaea04, PZC05, IlIl07, Koxx]
- adapting the seeds for specific biological tasks
  - [BBV04, KNP04, SuBu04, NoKu05]
- models for the mechanism that makes spaced seeds powerful
  - [BKS03, SuBu04, PZC05]
Applications

- **Similarity search**
  - WABA – [Kent,Zahler,2000]
  - PatternHunter – [Ma,Tromp,Li,2002]
  - YASS – [Kucherov,Noe,2005]

- **Read mapping**
  - MAQ – [Li,Ruan,Durbin,2008]
  - BFAST – [Homer,Merriman,Nelson,2009]
  - SHRiMP – [David,Dzamba,Lister,Ilie,Brudno,2010]

- **Oligonucleotide design**
  - ProDesign – [Feng,Tillier,2007]
Computing Good Seeds

- finding optimal seeds – hard
- exhaustive search – two exponential steps:
  - exponentially many seeds
    - single seeds: $O(2^\ell)$
    - multiple seeds: $O\left(2^{\sum_{i=1}^k \ell_i} \max L^k\right)$
  - computing sensitivity of each is exponential
    - single seeds: $O(nw2^{\ell-w})$
    - multiple seeds: $O\left((k + L + n)\sum_{i=1}^k \ell_i 2^{\ell_i-w}\right)$
- feasible only for single seeds – [Choi, Zeng, Zhang, 2004]
- for multiple seeds – heuristic algorithms – all exponential
Computing Good Seeds

- finding optimal seeds – hard
- exhaustive search – two exponential steps:
  - exponentially many seeds
    - single seeds: $\mathcal{O}(2^\ell)$
    - multiple seeds: $\mathcal{O}(2^{\sum_{i=1}^k \ell_i})\max L^k$
  - computing sensitivity of each is exponential
    - single seeds: $\mathcal{O}(nw2^{\ell-w})$
    - multiple seeds: $\mathcal{O}((k + L + n)\sum_{i=1}^k \ell_i2^{\ell_i-w})$
- feasible only for single seeds – [Choi, Zeng, Zhang, 2004]
- for multiple seeds – heuristic algorithms – all exponential
- a polynomial-time algorithm must
  - avoid considering all seeds
  - avoid computing sensitivity (!)
Avoiding Computing Sensitivity

- the hits of uniformly spaced seeds are more clustered
  - consecutive seed is uniformly spaced

- clustered $\Rightarrow$ overlapping

- idea: low overlap complexity $\Leftrightarrow$ high sensitivity
Overlap complexity (OC)

\[ \sigma[i] = \text{pairs of 1's aligned together} \]

\[ \text{OC}(s_1, s_2) = \sum_{i=1-|s_2|}^{|s_1|-1} 2^{\sigma[i]} \]

for multiple \((k)\) seeds:

\[ \text{OC}(S) = \sum_{1 \leq i \leq j \leq k} \text{OC}(s_i, s_j) \]

invar. w.r.t order, reversal

intuitive reason behind definition

\[ \text{OC}(11**1*1, 1*11) = 25 \]
Correlation with Sensitivity

- sensitivity of top overlap complexity seeds for weights 9..18, similarity 70%, and length of random region 64

<table>
<thead>
<tr>
<th>weight</th>
<th>optimal sensitivity</th>
<th>sensitivity of a top overlap seed</th>
<th>difference to optimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>0.729156</td>
<td>0.729156</td>
<td>0.000000</td>
</tr>
<tr>
<td>10</td>
<td>0.595740</td>
<td>0.595740</td>
<td>0.000000</td>
</tr>
<tr>
<td>11</td>
<td>0.467122</td>
<td>0.467122</td>
<td>0.000000</td>
</tr>
<tr>
<td>12</td>
<td>0.356430</td>
<td>0.356430</td>
<td>0.000000</td>
</tr>
<tr>
<td>13</td>
<td>0.264750</td>
<td>0.264750</td>
<td>0.000000</td>
</tr>
<tr>
<td>14</td>
<td>0.193514</td>
<td>0.193514</td>
<td>0.000000</td>
</tr>
<tr>
<td>15</td>
<td>0.138660</td>
<td>0.138333</td>
<td>0.000327</td>
</tr>
<tr>
<td>16</td>
<td>0.098942</td>
<td>0.098942</td>
<td>0.000000</td>
</tr>
<tr>
<td>17</td>
<td>0.070004</td>
<td>0.070004</td>
<td>0.000000</td>
</tr>
<tr>
<td>18</td>
<td>0.049146</td>
<td>0.049146</td>
<td>0.000000</td>
</tr>
</tbody>
</table>
Avoid Considering All Seeds

- Hill Climbing based on overlap complexity
  - start with \(*\ldots*11\ldots1\) (very bad seed!)
  - swap 1 ↔ * to cause the greatest decrease in OC
  - at most \(kw\) swaps
- complexity: \(O(w^5)\)
Avoid Considering All Seeds

- Hill Climbing based on overlap complexity
  - start with **...*11...1 (very bad seed!)
  - swap 1 ↔ * to cause the greatest decrease in OC
  - at most $kw$ swaps

- complexity: $O(w^5)$

- example: PatternHunter seed

<table>
<thead>
<tr>
<th>intermediate seeds</th>
<th>pairs swapped</th>
</tr>
</thead>
<tbody>
<tr>
<td>* * * * * * * 1 1 1 1 1 1 1 1</td>
<td>(1, 12)</td>
</tr>
</tbody>
</table>
Avoid Considering All Seeds

- Hill Climbing based on overlap complexity
  - start with $**\ldots **11\ldots 1$ (very bad seed!)
  - swap $1 \leftrightarrow *$ to cause the greatest decrease in OC
  - at most $kw$ swaps

- complexity: $O(w^5)$

- example: PatternHunter seed

<table>
<thead>
<tr>
<th>intermediate seeds</th>
<th>pairs swapped</th>
</tr>
</thead>
<tbody>
<tr>
<td>$**\ldots **11\ldots 1$</td>
<td>(1, 12)</td>
</tr>
<tr>
<td>$1**\ldots **11\ldots 1$</td>
<td>(3, 15)</td>
</tr>
</tbody>
</table>
Avoid Considering All Seeds

- Hill Climbing based on overlap complexity
  - start with **...*11...1 (very bad seed!)
  - swap 1 ↔ * to cause the greatest decrease in OC
  - at most \( kw \) swaps
- complexity: \( O(w^5) \)
- example: PatternHunter seed

<table>
<thead>
<tr>
<th>intermediate seeds</th>
<th>pairs swapped</th>
</tr>
</thead>
<tbody>
<tr>
<td>** ** ** ** ** 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1</td>
<td>(1, 12)</td>
</tr>
<tr>
<td>1 ** ** ** ** ** 1 1 1 1 1 * 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1</td>
<td>(3, 15)</td>
</tr>
<tr>
<td>1 * 1 ** ** ** ** 1 1 1 1 1 * 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1</td>
<td>(2, 9)</td>
</tr>
<tr>
<td>1 1 1 ** ** ** ** 1 * 1 1 * 1 1 1 * 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1</td>
<td>(5, 11)</td>
</tr>
<tr>
<td>1 1 1 * 1 ** ** ** 1 1 1 * 1 1 1 * 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1</td>
<td></td>
</tr>
</tbody>
</table>
Avoid Considering All Seeds

- Hill Climbing based on overlap complexity
  - start with **...*11...1 (very bad seed!)
  - swap 1 ↔ * to cause the greatest decrease in OC
  - at most $kw$ swaps

- complexity: $O(w^5)$

- example: PatternHunter seed

<table>
<thead>
<tr>
<th>intermediate seeds</th>
<th>pairs swapped</th>
</tr>
</thead>
<tbody>
<tr>
<td>* * * * * * * 1 1 1 1 1 1 1</td>
<td>(1, 12)</td>
</tr>
<tr>
<td>1 * * * * * * 1 1 1 1 * 1 1 1 1 1</td>
<td>(3, 15)</td>
</tr>
<tr>
<td>1 * 1 * * * * 1 1 1 1 * 1 1 * 1 1 1 1</td>
<td>(2, 9)</td>
</tr>
<tr>
<td>1 1 1 * * * * 1 * 1 1 * 1 1 * 1 1 1 1</td>
<td>(5, 11)</td>
</tr>
<tr>
<td>1 1 1 * 1 * * 1 * 1 1 * 1 1 * 1 1 1 1</td>
<td></td>
</tr>
</tbody>
</table>

- PatternHunter seed found using 4 swaps
- this can be done by hand!!
Sensitivity of Heuristic Seeds

- sensitivity of heuristic seeds, weights 9..18, similarity 70%, and length of random region 64

<table>
<thead>
<tr>
<th>weight</th>
<th>optimal sensitivity</th>
<th>swap sensitivity</th>
<th>difference to optimal</th>
<th>time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>0.729156</td>
<td>0.726279</td>
<td>0.002877</td>
<td>0.01</td>
</tr>
<tr>
<td>10</td>
<td>0.595740</td>
<td>0.594758</td>
<td>0.000981</td>
<td>0.01</td>
</tr>
<tr>
<td>11</td>
<td>0.467122</td>
<td>0.467122</td>
<td>0.000000</td>
<td>0.01</td>
</tr>
<tr>
<td>12</td>
<td>0.356430</td>
<td>0.354035</td>
<td>0.002395</td>
<td>0.04</td>
</tr>
<tr>
<td>13</td>
<td>0.264750</td>
<td>0.264512</td>
<td>0.000238</td>
<td>0.04</td>
</tr>
<tr>
<td>14</td>
<td>0.193514</td>
<td>0.192711</td>
<td>0.000803</td>
<td>0.09</td>
</tr>
<tr>
<td>15</td>
<td>0.138660</td>
<td>0.138333</td>
<td>0.000327</td>
<td>0.16</td>
</tr>
<tr>
<td>16</td>
<td>0.098942</td>
<td>0.098865</td>
<td>0.000076</td>
<td>0.17</td>
</tr>
<tr>
<td>17</td>
<td>0.070004</td>
<td>0.069874</td>
<td>0.000130</td>
<td>0.33</td>
</tr>
<tr>
<td>18</td>
<td>0.049146</td>
<td>0.048946</td>
<td>0.000200</td>
<td>0.58</td>
</tr>
</tbody>
</table>
### Comparison with PatternHunter II

<table>
<thead>
<tr>
<th>similarity</th>
<th>sensitivity of the 16 seeds of PatternHunter II</th>
<th>sensitivity of our 16 seeds</th>
<th>sensitivity of our 32 seeds</th>
</tr>
</thead>
<tbody>
<tr>
<td>60%</td>
<td>0.566640</td>
<td>0.582399</td>
<td>0.703620</td>
</tr>
<tr>
<td>65%</td>
<td>0.781508</td>
<td>0.795993</td>
<td>0.880050</td>
</tr>
<tr>
<td>70%</td>
<td>0.924114</td>
<td>0.932308</td>
<td>0.968727</td>
</tr>
<tr>
<td>75%</td>
<td>0.984289</td>
<td>0.986850</td>
<td>0.995516</td>
</tr>
<tr>
<td>80%</td>
<td>0.998449</td>
<td>0.998813</td>
<td>0.999725</td>
</tr>
<tr>
<td>85%</td>
<td>0.999951</td>
<td>0.999967</td>
<td>0.999995</td>
</tr>
<tr>
<td>90%</td>
<td>1.000000</td>
<td>1.000000</td>
<td>1.000000</td>
</tr>
</tbody>
</table>

**time**

- 12 days
- 10 s
- 85 s
Comparison with PatternHunter II

<table>
<thead>
<tr>
<th>similarity</th>
<th>sensitivity of the 16 seeds of PatternHunter II</th>
<th>sensitivity of our 16 seeds</th>
<th>sensitivity of our 32 seeds</th>
</tr>
</thead>
<tbody>
<tr>
<td>60%</td>
<td>0.566640</td>
<td>0.582399</td>
<td>0.703620</td>
</tr>
<tr>
<td>65%</td>
<td>0.781508</td>
<td>0.795993</td>
<td>0.880050</td>
</tr>
<tr>
<td>70%</td>
<td>0.924114</td>
<td>0.932308</td>
<td>0.968727</td>
</tr>
<tr>
<td>75%</td>
<td>0.984289</td>
<td>0.986850</td>
<td>0.995516</td>
</tr>
<tr>
<td>80%</td>
<td>0.998449</td>
<td>0.998813</td>
<td>0.999725</td>
</tr>
<tr>
<td>85%</td>
<td>0.999951</td>
<td>0.999967</td>
<td>0.999995</td>
</tr>
<tr>
<td>90%</td>
<td>1.000000</td>
<td>1.000000</td>
<td>1.000000</td>
</tr>
</tbody>
</table>

| time       | 12 days                                        | 10 s                        | 85 s                        |

**better multiple spaced seed; 100,000 faster!**
SpEED

- **Spaced SEED** computing software
  - engineering the overlap complexity
- **Hill climbing + restart**
Spaced SEED computing software
- engineering the overlap complexity

Hill climbing + restart

repeat a fixed number of times
- guess good seed lengths
- apply overlap complexity
- use sensitivity to keep the best seed
SpEED

- **Spaced SEED** computing software
  - engineering the overlap complexity
- **Hill climbing + restart**
- repeat a fixed number of times
  - guess good seed lengths
  - apply overlap complexity
  - use sensitivity to keep the best seed
- **SpEED fast** – using only overlap complexity – polynomial
- **SpEED best** – the best seed at the end – exponential
Spaced SEED computing software
  engineering the overlap complexity

Hill climbing + restart
repeat a fixed number of times
  guess good seed lengths
  apply overlap complexity
  use sensitivity to keep the best seed

SpEED fast – using only overlap complexity – polynomial

SpEED best – the best seed at the end – exponential

compared against the best existing software for computing seeds:
  Mandala – Buhler, Keich, Sun
  Iedera – Kucherov, Noé, Roytberg
<table>
<thead>
<tr>
<th>$w$</th>
<th>$p$</th>
<th>original seeds</th>
<th>Mandala</th>
<th>Iedera</th>
<th>SpEED</th>
<th>first</th>
<th>best</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.75</td>
<td>89.6113</td>
<td>90.6608</td>
<td>90.6802</td>
<td>90.6835</td>
<td>90.9098</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.80</td>
<td>97.3159</td>
<td>97.7316</td>
<td>97.7586</td>
<td>97.7436</td>
<td>97.8337</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.85</td>
<td>99.6613</td>
<td>99.7283</td>
<td>99.7437</td>
<td>99.7414</td>
<td>99.7569</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.75</td>
<td>261</td>
<td>2,706</td>
<td>0.06</td>
<td>1,300</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.80</td>
<td>81.6772</td>
<td>83.0512</td>
<td>83.2413</td>
<td>83.1190</td>
<td>83.3793</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.85</td>
<td>94.1141</td>
<td>94.7845</td>
<td>94.9350</td>
<td>94.8619</td>
<td>94.9861</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>714</td>
<td>5,355</td>
<td>0.07</td>
<td>1,658</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.80</td>
<td>89.3037</td>
<td>90.2580</td>
<td>90.3934</td>
<td>90.2786</td>
<td>90.5750</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.85</td>
<td>97.7253</td>
<td>98.0786</td>
<td>98.0781</td>
<td>98.1239</td>
<td>98.1589</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.90</td>
<td>99.8330</td>
<td>99.8633</td>
<td>99.8773</td>
<td>99.8777</td>
<td>99.8821</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2,092</td>
<td>10,476</td>
<td>0.1</td>
<td>2,772</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.85</td>
<td>84.0995</td>
<td>84.3838</td>
<td>84.5795</td>
<td>84.5476</td>
<td>84.8212</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.90</td>
<td>97.1676</td>
<td>97.3023</td>
<td>97.2806</td>
<td>97.3299</td>
<td>97.4321</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.95</td>
<td>99.9260</td>
<td>99.9287</td>
<td>99.9331</td>
<td>99.9338</td>
<td>99.9388</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>10,218</td>
<td>10,220</td>
<td>0.3</td>
<td>2,279</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.85</td>
<td>71.1961</td>
<td>72.1954</td>
<td>72.1695</td>
<td>72.8024</td>
<td>73.1664</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.90</td>
<td>92.5652</td>
<td>93.0855</td>
<td>93.0442</td>
<td>93.5595</td>
<td>93.7120</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.95</td>
<td>99.6299</td>
<td>99.6603</td>
<td>99.6690</td>
<td>99.7372</td>
<td>99.7500</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1,604</td>
<td>5,432</td>
<td>0.6</td>
<td>31,374</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>0.70</td>
<td>92.4114</td>
<td>92.3811</td>
<td>92.0708</td>
<td>92.9759</td>
<td>93.2526</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.75</td>
<td>98.4289</td>
<td>98.4320</td>
<td>98.3391</td>
<td>98.5971</td>
<td>98.6882</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.80</td>
<td>99.8449</td>
<td>99.8448</td>
<td>99.8366</td>
<td>99.8675</td>
<td>99.8820</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>37,806</td>
<td>12,326</td>
<td>13.1</td>
<td>44,651</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.85</td>
<td>92.4114</td>
<td>92.3811</td>
<td>92.0708</td>
<td>92.9759</td>
<td>93.2526</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.90</td>
<td>98.4289</td>
<td>98.4320</td>
<td>98.3391</td>
<td>98.5971</td>
<td>98.6882</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.95</td>
<td>99.8449</td>
<td>99.8448</td>
<td>99.8366</td>
<td>99.8675</td>
<td>99.8820</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>37,806</td>
<td>12,326</td>
<td>13.1</td>
<td>44,651</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>0.85</td>
<td>58.6907</td>
<td>—</td>
<td>60.1535</td>
<td>60.3534</td>
<td>60.8127</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.90</td>
<td>87.3359</td>
<td>—</td>
<td>87.9894</td>
<td>88.3817</td>
<td>88.5969</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.95</td>
<td>99.2249</td>
<td>—</td>
<td>99.2196</td>
<td>99.3524</td>
<td>99.3659</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>&gt; 1 day</td>
<td>62,683</td>
<td>18.8</td>
<td>29,298</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**SHRiMP: 4 seeds ($N = 50$)**

**PatternHunter II: 16 seeds ($N = 64$)**

**BFAST: 10 seeds ($N = 50$)**
References
