Optimal Spaced Seeds for Hidden Markov Models, with Applications to Homologous Coding Regions

Broňa Brejová, Daniel G. Brown, Tomáš Vinař

School of Computer Science, University of Waterloo

Large-scale pairwise local alignment

- Problem: Given two long genomic sequences, identify and align similar regions
- Fundamental tool in comparative genomics
- Full $O(nm)$ dynamic programing not feasible
- Fast heuristic methods are used (e.g. BLAST)

BLAST [S. F. Altschul et al. 1990]

Algorithm:

1. Find exact matches of length w (**hits**)
2. Extend hits to alignments

Problems:

- **False positives:** spurious hits that cannot be extended to significant alignments; they increase running time
- **False negatives:** real alignments without a hit; they are not found
- **Sensitivity:** fraction of real alignments with a hit
- Decrease $w \rightarrow$ increase sensitivity but also running time

  Increase $w \rightarrow$ decrease running time but also sensitivity

Spaced seeds – PatternHunter [Ma, Tromp, Li, 2002]

Increases sensitivity without increasing running time

BLAST hit: w consecutive matches

PatternHunter hit: w non-consecutive matches in a prescribed pattern, called **spaced seed** (w – weight of the seed)

Spaced seed as a binary string:

1 = position with required match; 0 = "don't care" position

Example: hits of seed 1101

```
GTGGTGCTCTCTGACAAAGCC
ATTGTTCTTAATGAGAAAGAA
   1101     1101
            1101
```

PatternHunter (cont.)

- All seeds of weight $w \rightarrow$ roughly same running time
- Seeds differ in sensitivity
- Want: most sensitive seed of a given weight

  $\rightarrow$ How to predict seed sensitivity?

Prediction of seed sensitivity

- **Predicted sensitivity:** probability that a **random alignment** has at least one seed hit
- **Random alignment:** positions are independent, match with probability p, mismatch with probability $1 - p$
- This specifies a simple **probabilistic model of alignments**

PatternHunter (cont.)

Assume: alignment length 80, probability of match 0.7

BLAST	**PatternHunter**
`1111111111`	`111001001001010111`
Expected number of hits:	
$71 \cdot 0.7^{10} = 2.01$	$63 \cdot 0.7^{10} = 1.78$
Probability of hit at position $i+1$ if hit at position i:	
0.7	$0.7^6 = 0.12$
`1111111111` `  1111111111`	`111001001001010111` `  111001001001010111`
Hits clustered together	Hits more “independent”
Sensitivity	
0.5	0.7

Seed sensitivity on protein coding regions

PatternHunter seed performs well for general alignments

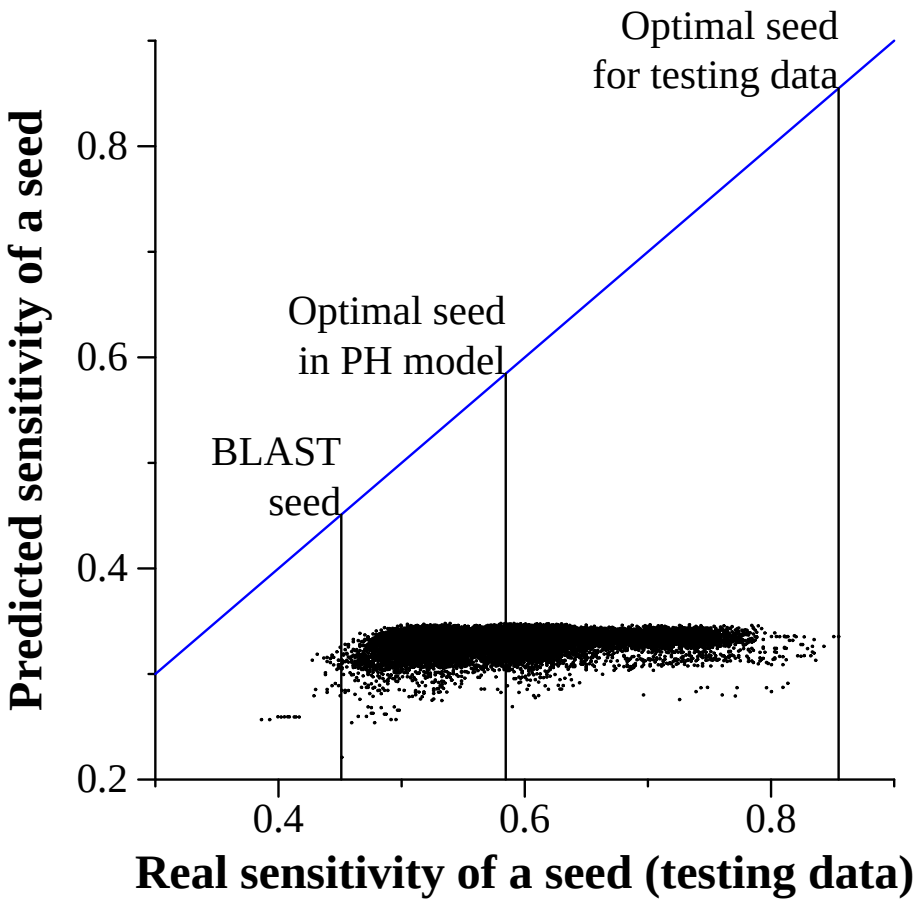
Protein coding regions

- Genome regions encoding proteins
- Alignments between protein coding regions:
  - Help to locate protein coding regions
  - Most alignments between distant species are inside protein coding regions

How PatternHunter seeds perform on coding region alignments?

Seed sensitivity on protein coding regions

Testing data: coding region alignments between human and *Drosophila*



Why the difference between prediction and reality?

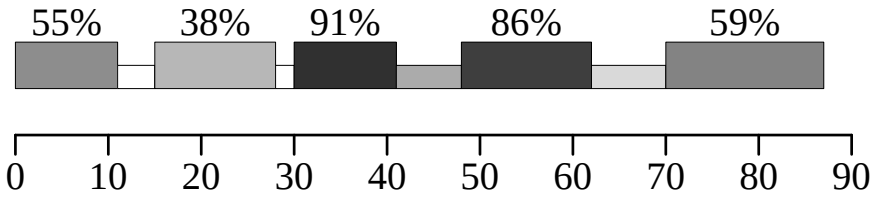
Special properties of coding regions:

- Sequence of triplets (codons). Each encodes one amino acid.

  Mutation rate depends on position within codon:

Position within codon:	first	second	third
Probability of match in our set:	0.54	0.64	0.34

- Dependencies among codon positions.
- Similarity level is not uniform:



Our approach:

- Better model for random alignments of coding regions
- Algorithm to predict sensitivity in such a model

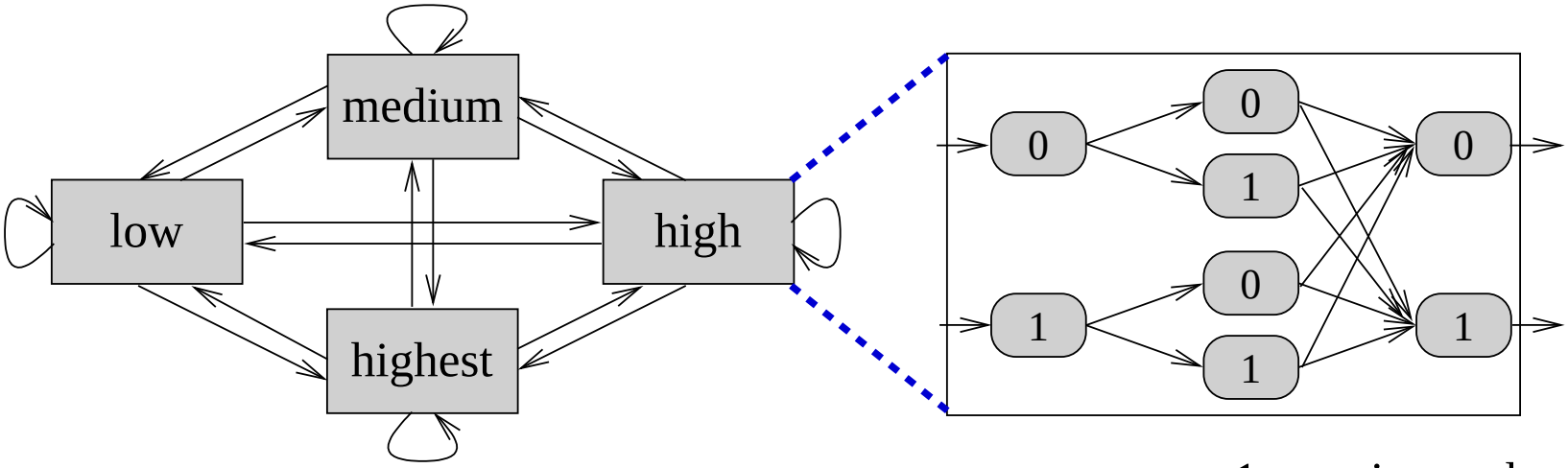
Models for random alignments of coding regions

$M^{(3)}$ model:

- parameters p_0, p_1, p_2
- alignment positions independent
- position i is a match with probability $p_{i \bmod 3}$
- models 3-periodic codon structure

Models for random alignments of coding regions (cont.)

Hidden Markov model (HMM):

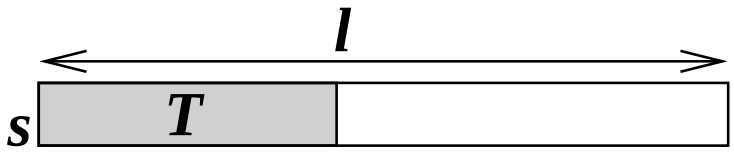


1 − emit match
0 − emit mismatch

- Four blocks, each with different similarity level
- Each block emits one or several codons
- Allow dependencies between codon positions

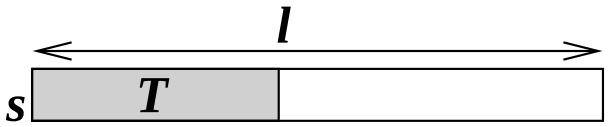
Prediction of seed sensitivity under HMM

- Extension of algorithm from [Keich, Li, Ma, Tromp].
- Given seed Q and HMM emitting binary (match/mismatch) sequence, compute sensitivity by dynamic programming
- **Subproblem:** $A(l, T, s) =$
  $\Pr[\text{hit in first } l \text{ characters} \mid \text{HMM starts in state } s \text{ by generating } T]$

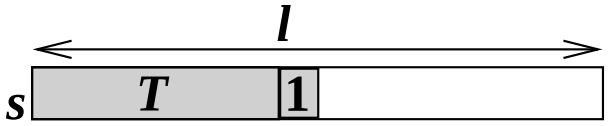


- Let H be set of all possible hits of Q (strings of length $|Q|$)
- Let H_P be set of prefixes of H
- **Base cases:** If $l < |Q|$, then $A(l, T, s) = 0$ (seq. too short)
  If $T \in H$, then $A(l, T, s) = 1$ (guaranteed hit)

Prediction of seed sensitivity under HMM (cont.)



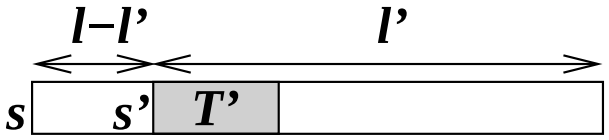
- If $T \in H_P \setminus H$, then



$$A(l,T,s) = q \cdot A(l,T1,s) + (1-q) \cdot A(l,T0,s)$$

$$q = \Pr[\text{HMM emits 1 after } T]$$

- If $T \notin H_P$, then

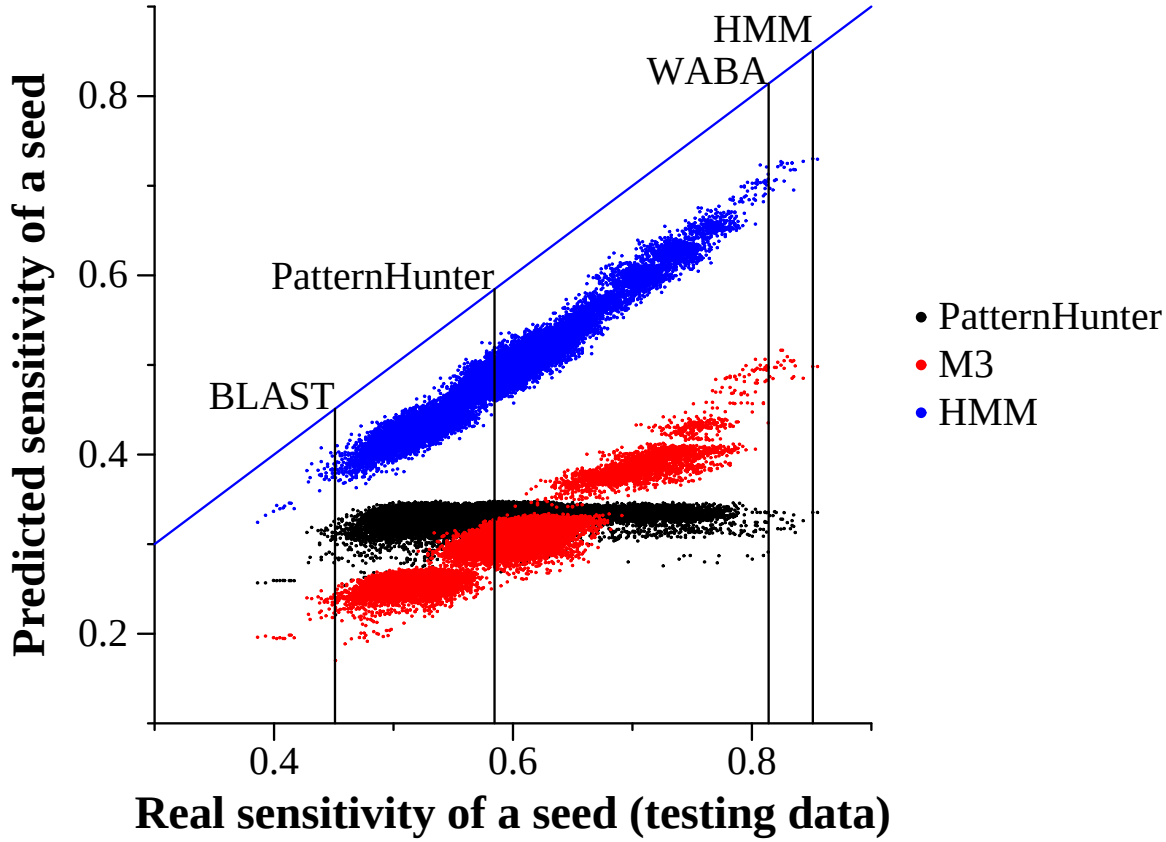


$$A(l,T,s) = \sum_{\text{state } s'} p_{s \rightsquigarrow s'} \cdot A(l',T',s')$$

T' = longest suffix of T in H_P

$p_{s \rightsquigarrow s'} = \Pr[\text{HMM starts } T' \text{ in state } s']$

Seed sensitivity on protein coding regions II.



Data opt.: 110 110 110 000 110 11

HMM: 110 110 000 110 110 11

WABA: 110 110 110 110 11

PatternHunter: 111001001001010111

BLAST: 1111111111

Observations from data

- Best seed matches 85% of alignments
- Top 3 actual seeds also top 3 seeds in HMM.
- BLAST seed matches only 45% of alignments
- Other programs for similar task:
  - TBLASTX: Translates both sequences into protein, then compares. Very slow.
  - BLAT: Seeds can have mismatches, but not to a pattern.
  - WABA: Seeds of form 110110110110
- Our method as much as 40 x as fast as TBLASTX, while essentially as sensitive. Much faster than BLAT, more sensitive than WABA.

Conclusion

- HMMs model coding regions alignments well.
- We developed an algorithm for computing sensitivity of a seed in such a model.
- Optimal seeds for these models highly sensitive in practice.

Future work:

- Faster algorithms for computing sensitivity
- Modeling other types of alignments
- Applications to other areas