Algorithms in Genome Analysis, Spring 2023

Veli Mäkinen

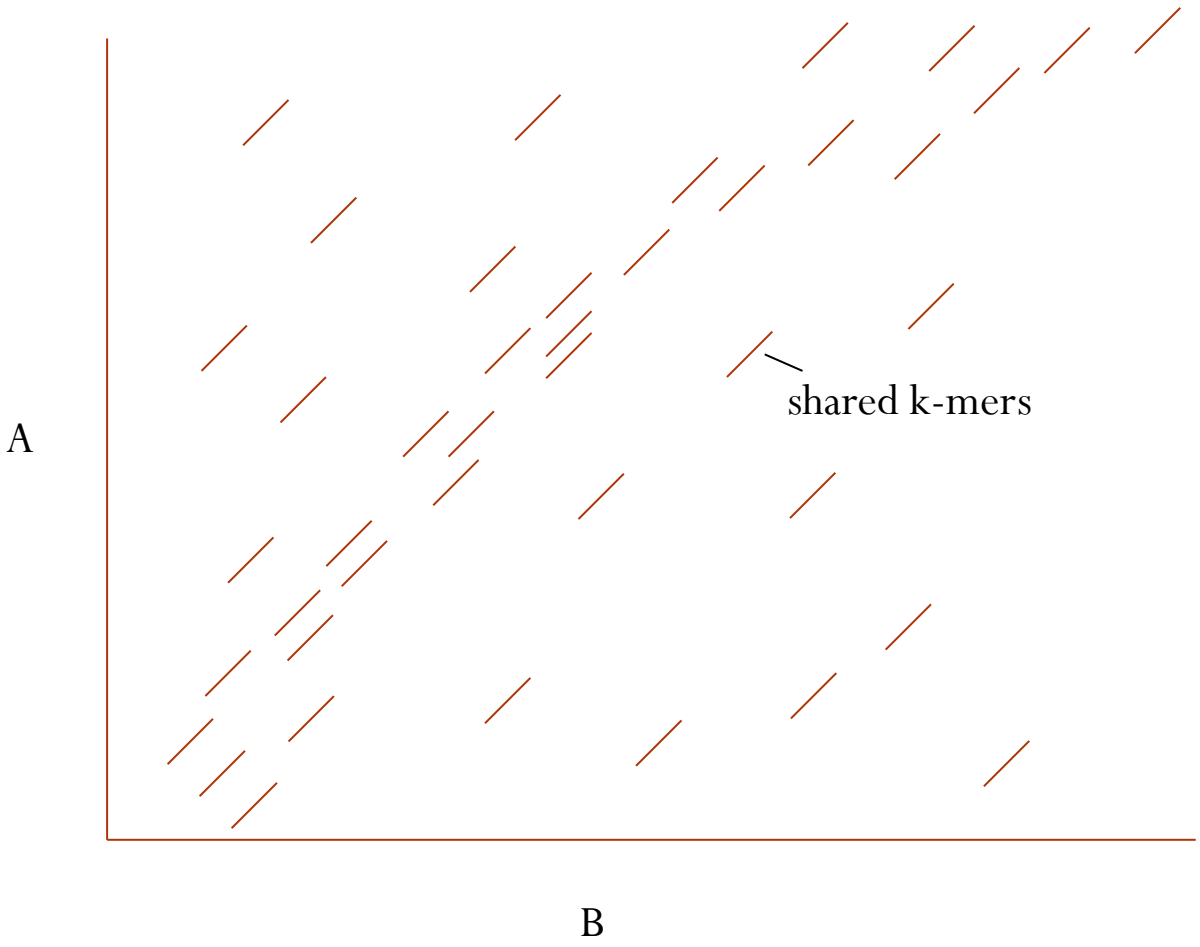
Lecture 1: Minimizers

Min-hashing based routine to speed-up alignments

Alignments

- Sequence alignment is the core routine in bioinformatics to estimate how one sequence has evolved from another:
  - It gives a similarity measure S(A,B) for two sequences A and B, with high value meaning that A and B are likely to be closely related
- Next week video lecture ”Alignment scores” introduces how this similarity measure S(A,B) is derived
- Later we learn that it takes quadratic time to compute S(A,B)
- With high-throughput sequencing data, such quadratic computations are usually impossible, so in practice, heuristics like BLAST are used instead.

BLAST main idea



See also ”dot-plot”

Alignment-free

- An alternative to heuristically computing S(A,B), is to look at other similarity measures, and compute those exactly
- Such methods are typically categorized as *alignment-free measures*

Jaccard similarity

- Can be used as alignment-free similarity measure between two sequences S and T.
- Let X and Y be the sets of k-mers of S and T, respectively.
- $J(S,T) = \frac{|X \cap Y|}{|X \cup Y|}$
- Can be computed using *bidirectional BWT*, see course book.
- Consider we want to compute the similarity between all pairs of sequences from a large collection.
- Approximating J(S,T) is fine, we just want to find all S that are candidates of being close to T, then align them.

Approximating Jaccard similarity

- Consider a random perfect hash function h() applied to all elements of X and Y, sorting the elements to sequences of k-mers X' and Y'.
- Value $J^p(X,Y) = \frac{|M \cap X'[1..p] \cap Y'[1..p]|}{|M|}$ approximates Jaccard similarity, where M is the set of p smallest elements of $X'[1..p] \cup Y'[1..p]$.
- The length p of the *min-sketch* or *fingerprint* (set of p minimum elements) determines how good the approximation is.
- Instead of single h() with extracted p-sketch, one can use p independent hash functions and take the minimum element from each as the p-sketch.
- (For probabilistic analysis of the approximation, see A. Z. Broder. On the resemblance and containment of documents. *Proceedings. Compression and Complexity of SEQUENCES 1997.)*

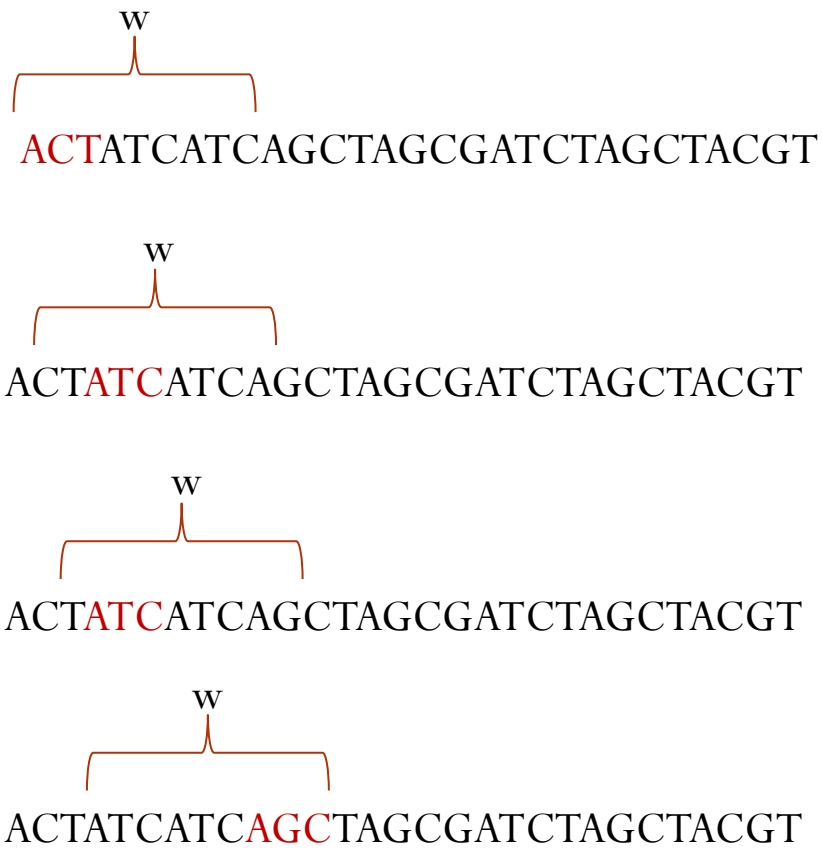
Example

- S=CAGCTAGCTAC, T=TAGGCTAGCTA, k=3
- X={CAG,AGC,GCT,CTA,TAG,TAC}
- Y={TAG,AGG,GGC,GCT,CTA,AGC}
- J(X,Y)=4/8=50%
- h(X)={35,24,62,5,12,41}, X'={5,12,24,35,41,62}
- h(Y)={12,8,19,62,5,24}, Y'={5,8,12,19,24,62}
- p=4, M={5,8,12,19}
- $J^4(X,Y) = \frac{2}{4} = 50\%, J^3(X,Y) = \frac{2}{3} = 66.6\%$
- $J^2(X,Y) = \frac{1}{2} = 50\%, J^1(X,Y) = \frac{1}{1} = 100\%$

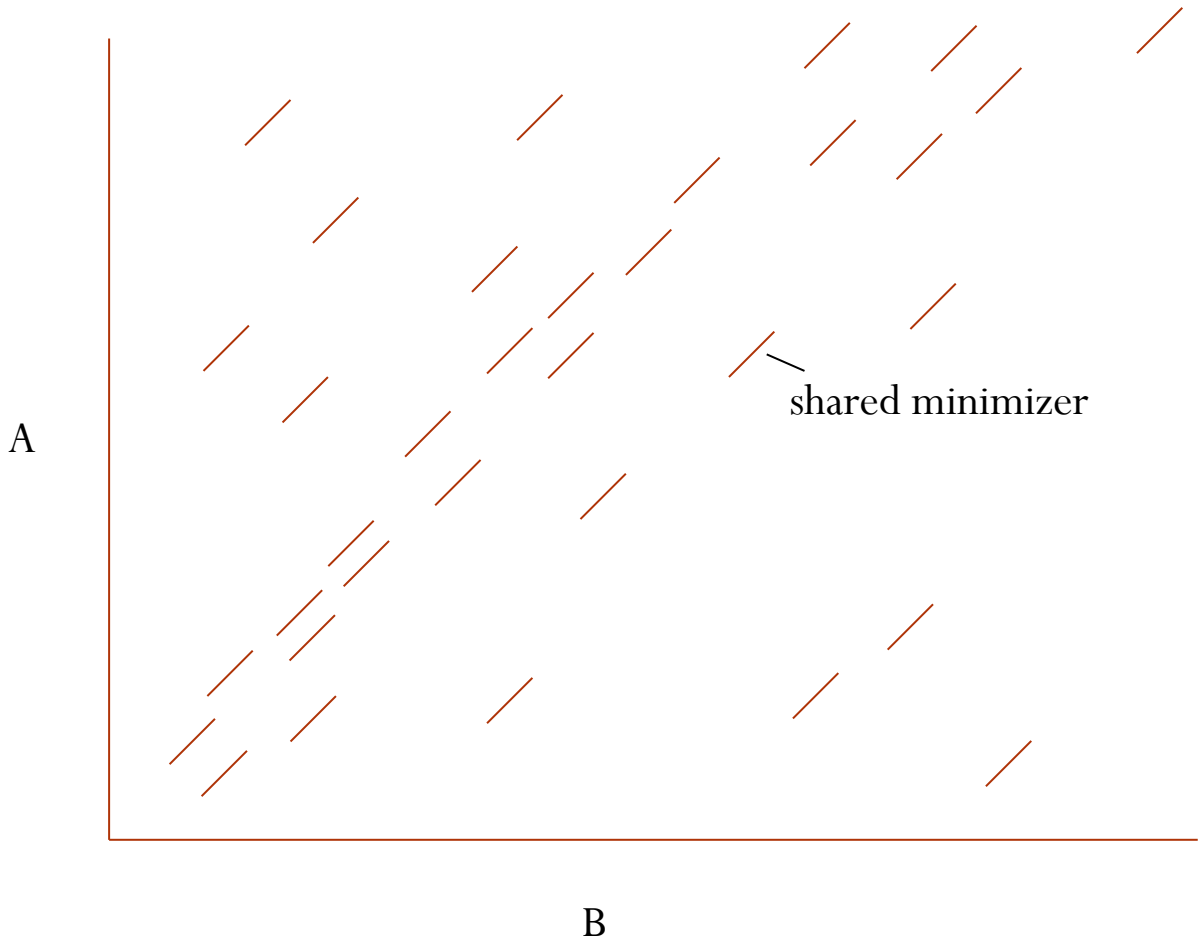
Minimizers

- Consider sliding a window of length w over a sequence S.
- For each window, find the minimum k-mer.
- The set of these over all windows form the set of minimizers of S.
  - Typically adjacent windows have the same minimizer.
  - Positions of the minimizers can be associated with the set.
- What is minimum k-mer?
  - One with minimum hash value
  - Or simply the lexicographically smallest
- How to use them?
  - Pair-up identical minimizers from two sequences to form alignment anchors
  - (For more applications, see papers citing Roberts, M., Hayes, W., Hunt, B.R., Mount, S.M. & Yorke, J.A. Reducing storage requirements for biological sequence comparison. *Bioinformatics*, 20:3363- 3369 (2004).)

Example of minimizers



Minimizer alignment idea



Minimizers in linear time

- Assume your hash function h(X) is string X=$x_1 x_2 \ldots x_k$ interpreted as a k-digit integer in base c, where {0,1,2,..,c-1} is the alphabet of X
  - DNA k-mer X can be interpreted with, e.g., 0=A, 1=C, 2=G, 3=T.
  - h(X)=$x_1 c^{k-1} + x_2 c^{k-2} + \cdots + x_k$
- Taking h(X) mod N limits the domain to [0..N-1] and one can add randomness to create a family of universal hash functions (details omitted here, see *Karp-Rabin fingerprints*)
- h($x_2 x_3 \ldots x_k\ a$)= $h(X)c - x_1 c^k + a$, thus each k-mer value can be computed in constant time.
- Exercise: How to get all the minimizers in linear time?