DNA-Shazam

The fingerprinting process was split in four parts, including a „search and score“ part.

The original algorithm, which was intended for audio recognition, works with frequency and time values to characterize a song. While the time values are easily replaced by position values, it is harder to find a DNA equivalent to frequency values. Nevertheless, the Shannon entropy may serve as a good replacement to frequency values because it combines two important k-mer properties in one value – the position and the number of occurences in a given sequence. Figure 1 demonstrates the calculation algorithm of the Shannon entropy (Wei, 2012) (Chun, 2005).

First, the provided sequence, regardless whether it is intended as query or database, is divided into smaller parts by means of a sliding window (e.g. window size 150 nucleotides (nt)). Within each window all possible k-mers of particular length (e.g. 3-mers) are generated and for each k-mer the Shannon entropy is calculated. We obtain entropy vectors e_i ordered by the position of the sliding window from which they were calculated.

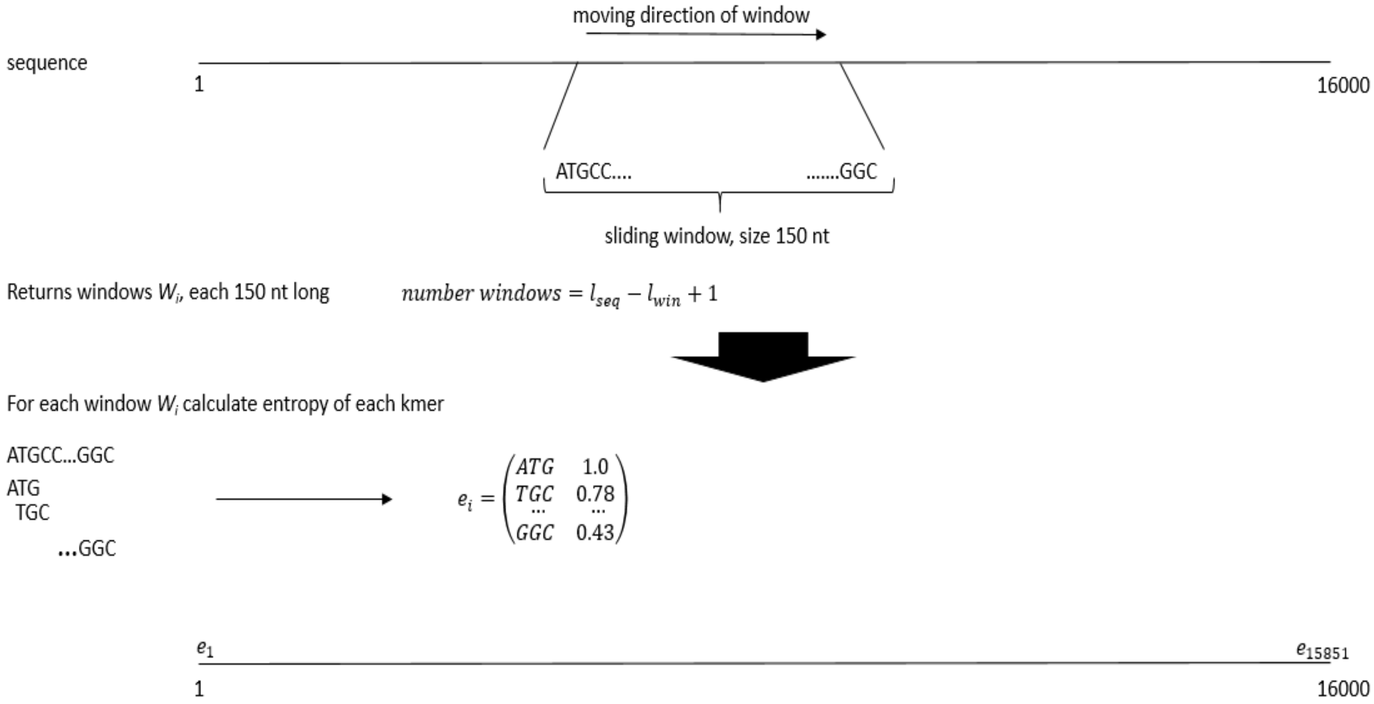

moving direction of window
sequence
1
16000
ATGCC....
.......GGC
sliding window, size 150 nt
Returns windows W_i, each 150 nt long
$number\ windows = l_{seq} - l_{win} + 1$
For each window W_i calculate entropy of each kmer
ATGCC...GGC
ATG
TGC
...GGC
$e_i = \begin{pmatrix} ATG & 1.0 \\ TGC & 0.78 \\ \cdots & \cdots \\ GGC & 0.43 \end{pmatrix}$
e_1
e_{15851}
1
16000

Figure 1: fingerprint generation step 1. Calculation of Shannon entropy for each kmer in the sliding window; process results in entropy vectors e_i ordered by position of sliding window from which they were derived; the process is shown exemplary for k-mer length=3, lseq=length of sequence, lwin= length of sliding window

In Figure 2 the feature extraction is shown. First, a number of entropy vectors (e.g. 100) is concatenated into a matrix M, with one vector e_i as one column. The mean of all entropy values in the matrix M is calculated (M_{mean}). From the matrix M, the new matrix M‘ is inferred where an element m_{ij} is kept only if m_{ij} is above M_{mean} and m_{ij} is set to zero if m_{ij} is less than M_{mean}. This step, of discarding elements with lower entropy, should increase the robustness of the fingerprints identification process since only the elements containing the most information are kept. Next, the M‘ matrix is divided into non-overlapping submatrices s_{ij} (e.g. size(s_{ij})=4x4) starting from element m'_{11}. For each submatrix s_{ij} the sum of its elements is calculated to account for local variances in the matrix M’. These sum(s_{ij}) are saved into a vector v_i . The index of the six highest elements in v_i is saved to the f_i as the feature vector. From a sequence of 16000 nts one would obatin 160 feature vectors, ordered by a position with the shown parameters

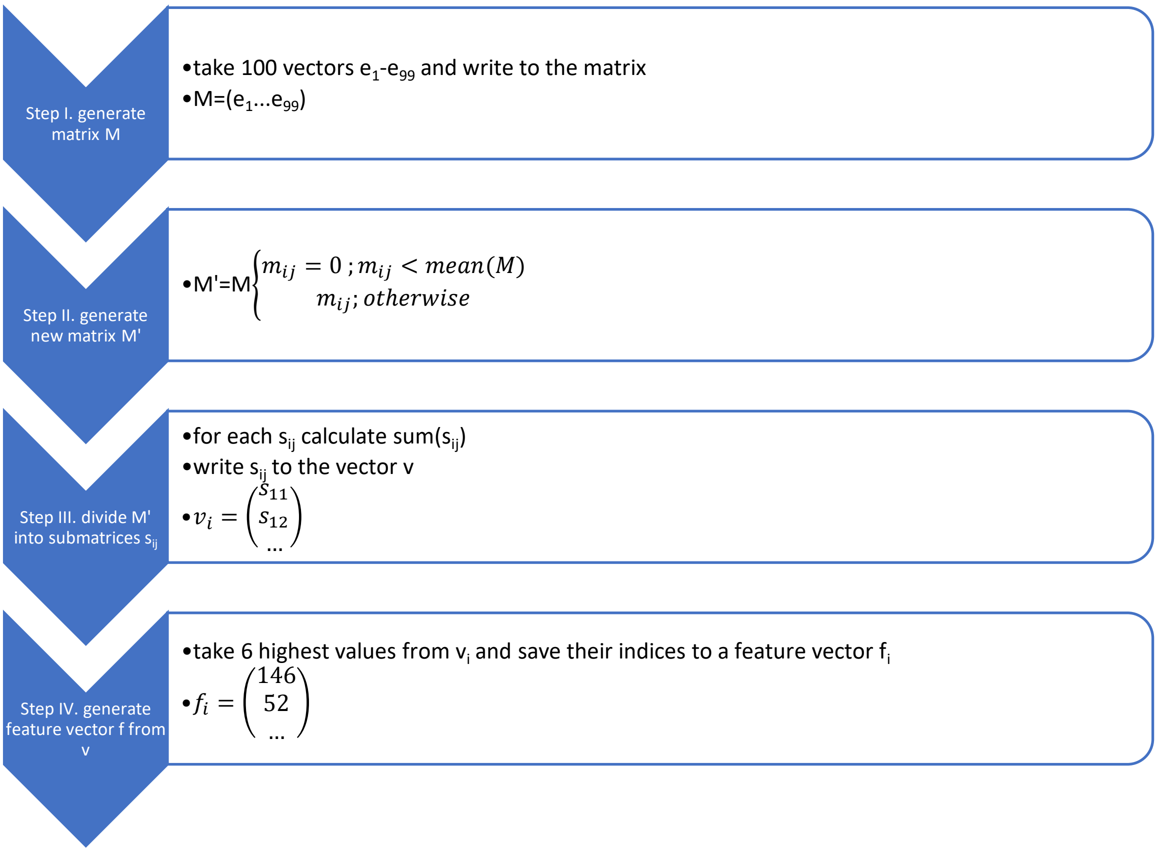

Step I. generate matrix M
•take 100 vectors e_1-e_{99} and write to the matrix
•M=(e_1…e_{99})
Step II. generate new matrix M'
•M'=M$\begin{cases} m_{ij} = 0\,; m_{ij} < mean(M) \\ m_{ij}; otherwise \end{cases}$
Step III. divide M' into submatrices s_{ij}
•for each s_{ij} calculate sum(s_{ij})
•write s_{ij} to the vector v
•$v_i = \begin{pmatrix} s_{11} \\ s_{12} \\ \dots \end{pmatrix}$
Step IV. generate feature vector f from v
•take 6 highest values from v_i and save their indices to a feature vector f_i
•$f_i = \begin{pmatrix} 146 \\ 52 \\ \dots \end{pmatrix}$

Figure 2: fingerprint generation step 2. The feature extraction performed for each block of entropy vectors e_i via concatenation of these vectors to a matrix M (step I); inferring of a matrix M’ from M to select most information, based on the mean entropy value (step II); division of M’ into submatrices s_{ij} to account for a local variances (step III); generate feature vectors (step IV); s_{ij} is a 4x4-matrix in this example

In the next step, an address for each feature vector is generated as shown in Figure 3. For this purpose all feature vectors are ordered by the position and k-mer length, inherited from the feature vector extraction (Figure 2), and assigned a corresponding index (the red number in Figure 3). The position is related to the position of the sliding window, e.g. featurevector 1 represents the first 100 entropy vectors which where inferred from the first 100 positions of the sliding window covering the first 249 bases of the original sequence. Next, feature vectors are combined in target zones by order of index, with 5 feature vectors for each target zone (Figure 3 panel B). Target zone 1 consist of feature vectors with index 0 to 4, target zone 2 consists of feature vectors with index 1 to 5 etc. Each target zone is paired with an anchor point (Figure 3 panel C). An anchor point is the third point before the very first point of a target zone.

Adresses of feature vectors with their corresponding anchor points and difference of position between feature vector and anchor are used as an unique identification template for matching sequences. For example the address for point 4 would be $[feature\ vector\ 1; feature\ vector\ 4; delta]$, with the delta equal to a difference between $position(featurevector\ 1)\ and\ position(featurevector\ 4)$. In the case when particular point is included into multiple target zones (e.g. point 6), it will have multiple addresses. These adresses are linked with vectors which will be referred to as "couples". Corresponding couples will be the value returned if an address-match is made during the searching process. In the case when a database is processed, the couple consists of the absolute position of the anchor in the sequence and the ID of the sequence. If a query is processed the couple is a vector with one entry: the absolute position of the anchor in the query.

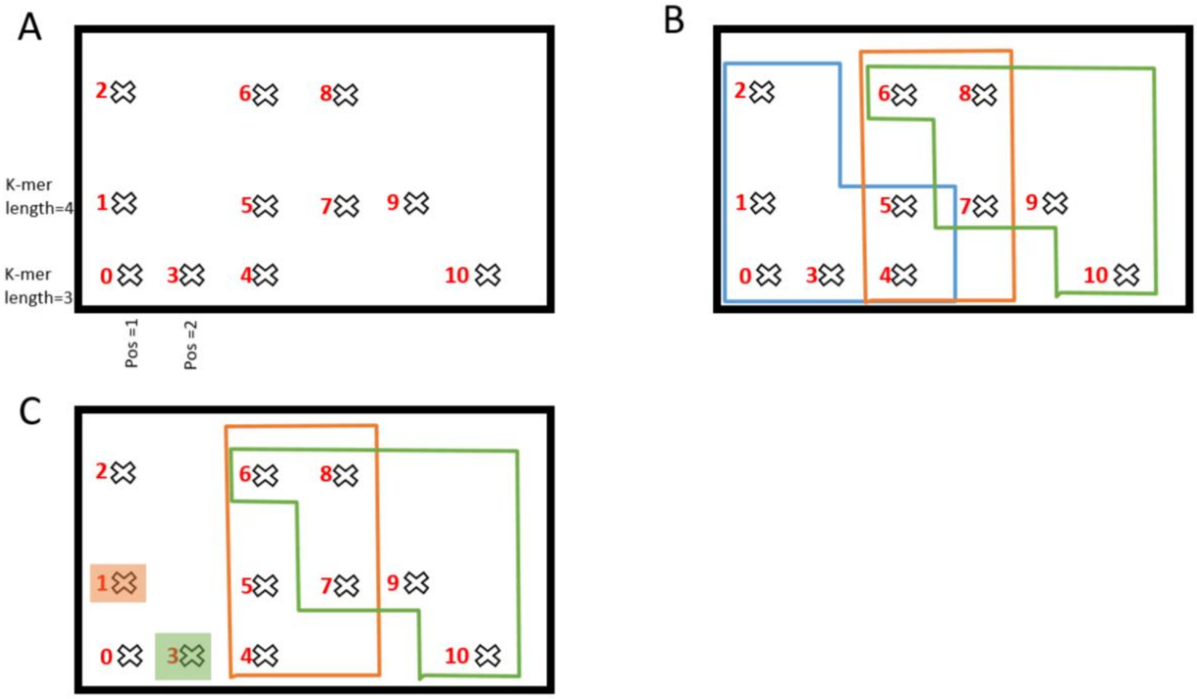
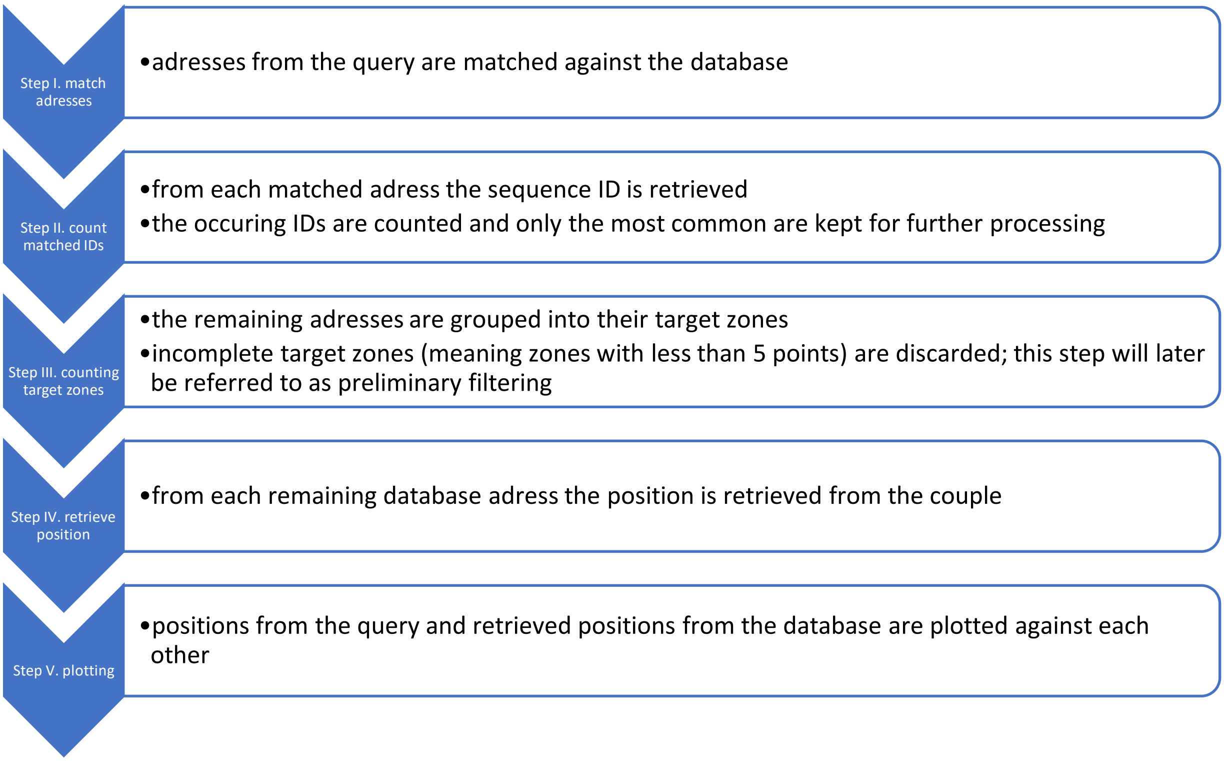


Figure 3: adress generation for feature vectors. In the panel A, the ordered feature vectors with their assigned indices (red numbers) are shown. Feature vectors are initially ordered by position and for the same position they are ordered by k-mer length. In the panel B, three target zones are shown exemplarily. Each target zone consists of 5 adjacent feature vectors grouped together by index. In the panel C, two target zones are shown together with their respective anchor. An anchor is the 3rd point upstream from the very first point of a target zone.

After accomplishing the fingerprinting process for both query and reference sequence, the searchprocess of a perfectly matching fingerprints is performed as shown in Figure 4.



Step I. match adresses

- adresses from the query are matched against the database

Step II. count matched IDs

- from each matched adress the sequence ID is retrieved
- the occuring IDs are counted and only the most common are kept for further processing

Step III. counting target zones

- the remaining adresses are grouped into their target zones
- incomplete target zones (meaning zones with less than 5 points) are discarded; this step will later be referred to as preliminary filtering

Step IV. retrieve position

- from each remaining database adress the position is retrieved from the couple

Step V. plotting

- positions from the query and retrieved positions from the database are plotted against each other

Figure 4: searching process for matching adresses after fingerprint generation. First query and database addresses are matched (Step I.). The retrieved IDs are counted and the most frequent (most frequent as specified by input) IDs and their adresses are kept for further processing (Step II.). The remaining addresses are scanned for complete target zones and only complete target zones are kept (Step III.). Positions are retrieved from the addresses still remaining and plotted against the positions of the query (Step IV. And Step V.).

If the query is a part of the database, matching adresses and therefore positions retrieved from the database, should occur in the same order as the matching adresses and therefore positions in the query. This should result in a perfect diagonal line if a match is found, when plotting query match positions against database match positions.

Wei, e. (2012). A novel hierarchical clustering algorithm for gene sequences. *BMC Bioinformatics*. doi:10.1186/1471-2105-13-174

Chun, e. (2005). Relative entropy of DNA and its application. *Physica A*(347), S. 465-471.

Wei, e. (2012). A novel hierarchical clustering algorithm for gene sequences. *BMC Bioinformatics*. doi:10.1186/1471-2105-13-174