ON USING DNA DISTANCES AND CONSENSUS IN REPEATS DETECTION

Petre G. POP
Technical University of Cluj-Napoca, Romania
petre.pop@com.utcluj.ro

Abstract: Sequence repeats are the simplest form of regularity and analyzing repeats can lead to first clues to evidencing new biological phenomena. Many of the methods for detecting repeated sequences are part of the digital signal processing (DSP) field and, therefore, the numerical representation of genomic signals is very important. Most of these methods use distances and consensus sequences to generate candidate sequences. This paper presents results obtained using a dedicated numerical representation and a mapping algorithm with different distances and consensus types to isolate the position of DNA repeats with specific lengths.

Keywords: *genomic signal processing, sequence repeats, DNA numerical representations, DNA distances, DNA consensus.*

I. INTRODUCTION

The existence of DNA repeated sequences is a fundamental feature of all genomes. A repeat is the simplest form of regularity and the detection of repeats is important in biology and medicine as it can be used for phylogenic studies and disease diagnosis. A major difficulty in identification of repeats is caused by the fact that the repeat units can be of unknown length and either in tandem or dispersed or exact or imperfect.

Nucleotide sequences are represented by character strings consists of the letters A, T, C and G, corresponding to Adenine, Thymine, Cytosine and Guanine nucleotides. A perfect (exact) repeat is a string that can be represented as a smaller string repeated contiguously at least twice. Repeats, whose copies are distant in the genome, whether or not located on the same chromosome, are called distant/dispersed repeats. Among those, biologists distinguish micro-satellites, mini-satellites, and satellites, according to the length of their repeated unit. However, perfect tandem repeats are of limited biological interest, since different biological events will often render the copies imperfect [1]. The result is an approximate repeat, defined as a string of nucleotides repeated consecutively at least twice with small differences between the instances.

The numerical representation of genomic signals becomes very important as almost all DSP techniques require two parts: mapping the symbolic data (symbols for nucleotides) into a numeric form in a non-arbitrary manner and calculating a kind of transform of that numeric sequence [2].

Most of the numerical representations used for repeats detection associate a numerical value to one position in the sequence using numerical values associated to each nucleotide and, finally, reflect the presence or the absence of a certain nucleotide in a specific position. In order to include information about the number of consecutive nucleotides and to generate only one numerical sequence for each DNA subsequence which may be associated with a repeat, we've introduced a novel representation and a mapping algorithm which takes into account the length of the expected repeats and the number of possible mismatches due to point mutations, based on polynomial-like representation [3]. Like many methods for detecting repeated sequences, we used distances and then evaluate a consensus sequence to generate candidate sequences.

This paper presents results obtained using this dedicated numerical representation with associated mapping algorithm and different DNA distances and consensus types to isolate the position of repeats in DNA sequences, having a specified length.

II. NUMERICAL REPRESENTATION AND MAPPING ALGORITHM

We've proposed a numerical representation and a mapping algorithm, which takes into account the length of the expected repeats and the number of possible mismatches because of point mutations [3].

For a DNA sequence of length L a numerical value is associated in a polynomial-like representation:

$$V = \sum_{k=0}^{L-1} V_{\alpha_k} 10^{k}, \quad \alpha \in \{A, G, C, T\} \qquad (1)$$

Where V_α is the value of a single nucleotide. These coefficients should be different integer values such that the resulting numerical value is unique for a subsequence. One possibility is to use consecutive natural numbers, preserving DNA's reverse complementary properties (A+T = C+G), such as A=1, G=2, C=3, T=4.

But for two very similar sequences (which differ, for instance, by a single nucleotide) will get two very different numbers. So it takes an algorithm that allows finding similar sequences and then generates single numerical values for these sequences.

The following input values are needed:

- A DNA sequence of length N;
- The length of expected repeated sequence, L;
- The maximum number of mismatches in the repeated

Manuscript received September 18, 2012; revised November 15

sequences, M_m.

To pass from DNA sequence to numerical values, a kind of distance and consensus value are needed:

- The distance measures the number of mismatches between sequences of the same length; if two sequences are identical, this distance should be zero;
- Given a number of sequences of the same length, the consensus sequence is a sequence formed by the nucleotide most likely to occur at each position in analyzed sequences.

The mapping algorithm is summarized below:

- Step-1: Consider all successive subsequences of length L in the initial DNA sequence;
- Step-2: Determine all the positions (and the associated subsequences of length L) in the original sequence for which the distance (against a sequence from Step-1) is less or equal to the prefixed mismatches number M_m;
- Step-3: Determine the consensus sequence for all (similar) subsequences from Step-2; Calculate the distance between the consensus sequence and each associated subsequence; those subsequences whose distance is greater than M_m must be reassign; Re-compute the consensus sequence for remaining sequences.
- Step-4: Compute the numerical value for consensus sequence (using (1)) and assign this value to all these positions.

As output, the algorithm generates a single vector, *SeqVal[ ],* of *(N-L)* structures; each structure is associated to a unique subsequence of length L (possible a repeat unit) and contains the position, the associated numerical value and the repetition number. The array will contain values for only those similar subsequences that occur in a number larger than a certain threshold. This array of structures can be further processed conveniently, for example can be sorted descending by number of repetitions or can be graphically represented using a dot-plot approach.

An important property of this mapping algorithm is that if the L value is a prime factor of repeated sequence length then the entire repeated sequence will be emphasized. This allows a significant reduction of the computational effort in case of long repeats.

III. DISTANCES AND CONSENSUS SEQUENCE

Determination of similar sequences in Step-2 and evaluation of consensus sequence requires evaluating the distance between two sequences. In our experiments we used two types of distances, used in string matching. The first type includes edit distances: Hamming distance, Levenshtein distance and Damerau-Levenshtein distance. The second category includes Jaro distance.

Distances from the first category evaluate conversion costs from a sequence to other using editing operations like substitution, insertion, deletion or transposition between adjacent positions. The simplest, Hamming distance [4] determines the number of different nucleotides between two equal length subsequences (i.e. measures only substitutions). Levenshtein distance [5] consider the insertion operation, substitution and deletion and is defined as the minimal number of characters you have to replace, insert or delete to transform string s into string t. Implementation is done using dynamic programming based on next formula:

$$c_{s,t}(i,\ j) = \begin{cases} 0, s_i = t_j \\ 1, s_i \neq t_j \end{cases}$$

$$Lev_{s,t}(i,\ j) = \begin{cases} 0 & ,i = j = 0 \\ i & ,j = 0,\ i > 0 \\ j & ,i = 0,\ j > 0 \\ \min \begin{cases} Lev_{s,t}(i-1,\ j) + 1 \\ Lev_{s,t}(i,\ j-1) + 1 \\ Lev_{s,t}(i-1,\ j-1) + c(i,\ j) \end{cases} \end{cases} \quad (2)$$

Damerau-Levenshtein distances [6] consider, in addition, transposition operation between adjacent positions. All these distances give a value of 0 (minimum value) for identical sequences and the maximum value is given by length of sequences.

Jaro distance [7] consider the number and order of the common nucleotides between two sequences. Let s^* be the nucleotides in s that are common with t (in the same order they appear in s), let t^* be analogous for t. Nucleotides considered to be common in the sequence s and t if they appear nearer than $Min(|s|,|t|)/2$. Let $T(s^*,t^*)$ be one-half of the number of transpositions, i.e., the number of positions where the nucleotides in s^* and t^* do not match. Jaro distance is then defined as following:

$$Jaro\ (s,t) = \frac{1}{3}\left(\frac{|s^*|}{|s|} + \frac{|t^*|}{|t|} + \frac{|s^*| - T(s^*,t^*)}{|s^*|} \right) \quad (3)$$

Values computed with (3) are between 0 (complete different sequences) and 1 (identical sequences).

In Step-3 we determine the consensus sequence for all similar subsequences determined in Step-2 and, on this basis, we calculate the associated numerical value (using (1)). Here, a consensus sequence is a sequence pattern derived from multiple, similar sequences that represents the nucleotide most likely to occur at each position in analyzed sequences [8]

We used the following types of consensus:

- Most frequently occurring nucleotide in each column, even if it is not the majority (MC).
- Majority with fixed cutoff: use the fraction of nucleotides in a column to establish majority for that column, provided that the fraction is greater than the cutoff parameter (MFCF).
- Majority with global appearing frequency cutoff: same as previous case but the cutoff for each nucleotide is computed as the appearing frequency in the original sequence (MGCF).
- Majority with local appearing frequency cutoff: same as previous case but the cutoff for each nucleotide is computed as the appearing frequency in the analyzed similar sub-sequences (MLCF).

For last three consensus types, if there is no nucleotide that exceeds the threshold we consider that we have no valid

consensus sequence and those subsequences must be reassign. In case that more than one nucleotide is calculated to have the same confidence, and this exceeds the consensus threshold, the nucleotides are assigned in descending order of their global appearing frequency precedence.

IV. EXPERIMENTS AND RESULTS

The experiments were performed in two stages. In the first stage we used a short test sequence with short repeated sequences, well characterized (location, length and pattern) to validate the numerical representation and the mapping algorithm.

In the second stage we used two long test sequences containing alpha satellite DNA of known length to test the performance of our method in case of long sequences.
Our case study was the human microsatellite sequence M65145 (GenBank [9]) which exhibits repeats of the 11-mer TGACTTTGGGG [10] and DNA alpha satellites in AC010523 Homo sapiens chromosome 19 (GenBank) and in AC136363 Homo sapiens chromosome 17 which contain dispersed alphoid sequences, both higher-order and monomeric alpha-satellite of approximately 171 bp (base pairs), tandemly arranged in a head-to-tail fashion [11].

For first experiments (M65145), we have created an application that implements the mapping algorithm and polynomial numerical representation and allows the introduction of input data as well as selecting the type of distance that is used to determine similar sequences (in Step-2) and to evaluate the consensus sequence (in Step-3) and also, selecting the type of consensus (in Step-3). Output data (numerical values and associated DNA sequences) are sorted descendant by the number of detected repetitions of the same type. For each combination of computation parameters (distances and consensus type), we used parameters *L=11* (repeat length), M_m*=2, 3, 4* (number of admissible mismatches) and determined the number of repetitions and the associated consensus sequence. We selected only those sequences that match the reported consensus sequence (TGACTTTGGGG). Detected number of repetitions for each value of the parameter M_m and for each combination of distances and consensus type, is represented in the next figures (1 …9).

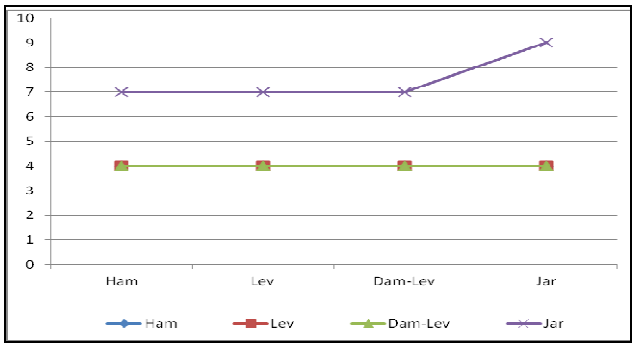

Figure 1. M_m=2; all types of consensus

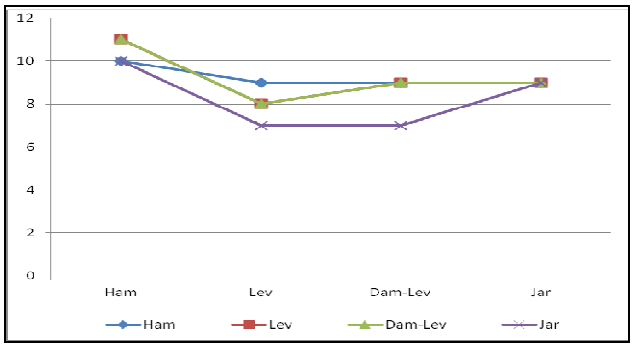

Figure 2. M_m=3;MC consensus.

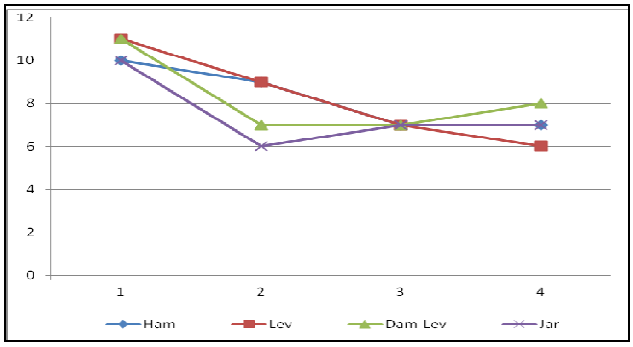

Figure 3. M_m=3;MFCF consensus.

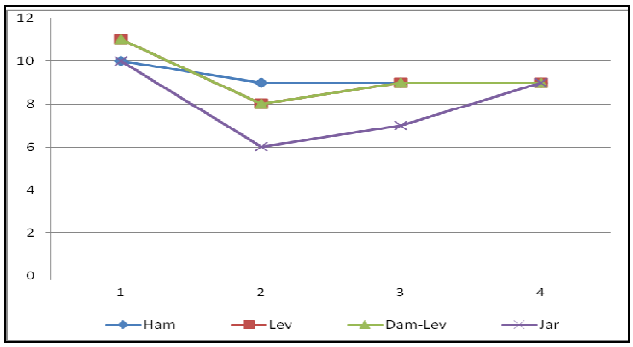

Figure 4. M_m=3; MGCF consensus.

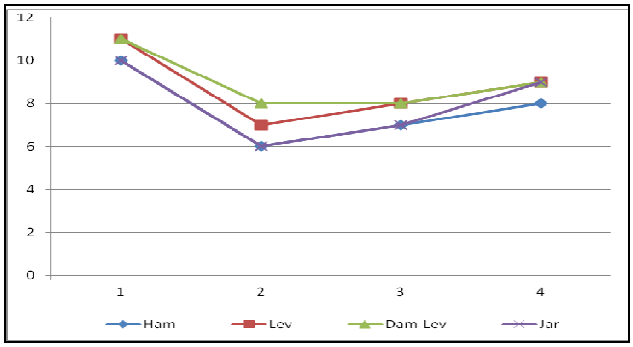

Figure 5. M_m=3; MLCF consensus.

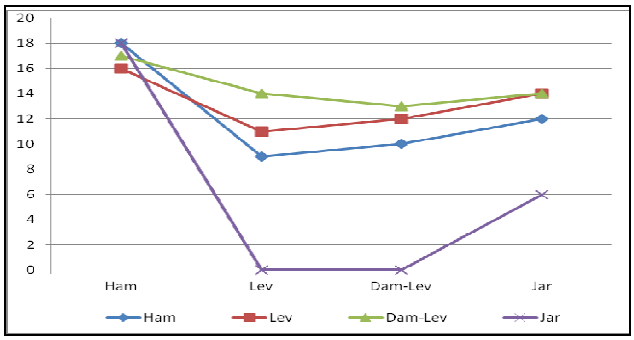

Figure 6. M_m=4; MC consensus.

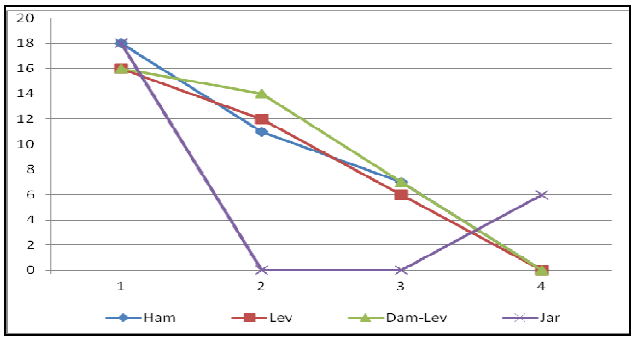

Figure 7. M_m=4; MFCF consensus.

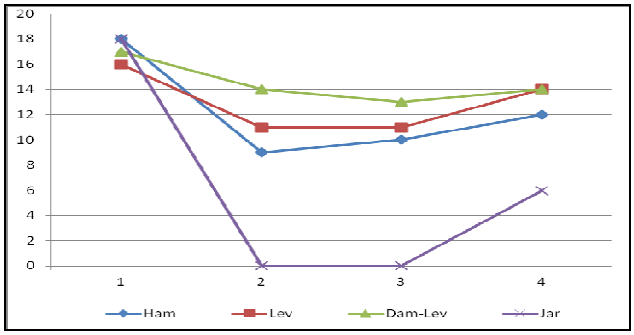

Figure 8. M_m=4; MGCF consensus.

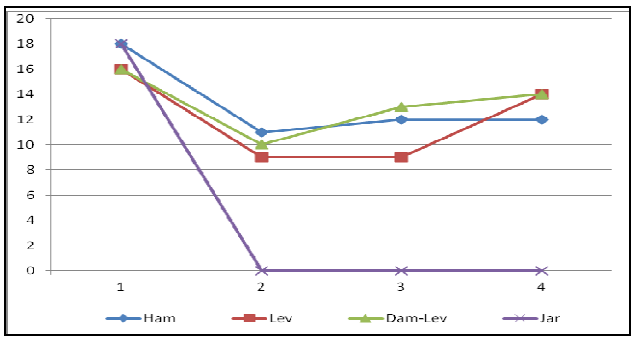

Figure 9. M_m=4; MLCF consensus.

Based on these experiments, some preliminary conclusions can be drawn:

- We have repeats detected for M_m=2 (which is a serious constraint in this case). This is an inherent advantage of the numerical representation and the associated mapping algorithm.
- Jaro distance in Step-2 lead to better results for M_m=2.
- In case of M_m=2, the method used to calculate the consensus sequence does not influence the results.
- Levenshtein distance or Damerau-Levenshtein distance used in Step-2 gives better results for M_m=3, M_m=4. Most common based consensus and majority consensus with global appearing frequency cutoff give the best results.
- Using the Hamming distance in Step-3 lead to better results for M_m=3, M_m=4, regardless of the method used to assess the consensus sequence.
- Jaro distance use in Step-2, for M_m=4 leads to worst results.

For the second stage of experiments (AC010523 and AC136363) we used dot plot analysis. Dot plots are two-dimensional representations where the x-axis and y-axis each represents a sequence and the plot itself shows a comparison of these two sequences by a calculated score for each position of the sequence. If a window of fixed size on one sequence (one axis) match to the other sequence a dot is drawn at the plot.
Some important characteristics of patterns appearing in dot plots are [12]:

- Parallels to the main diagonal indicate repeated regions in the same reading direction on different parts of the sequences.
- Blocks of parallel lines indicate tandem repeats of a larger motif in both sequences. The distance between the diagonals equals the distance of the motif.

For evaluating the results of these experiments we developed a customized dot-plot analysis as:

- The analyzed sequence is a numerical one (the output of mapping algorithm) and not a symbolic one.
- Most times the length of analyzed sequence far exceeds the number of points on each axis (in our case, the first analyzed sequence is around 40,000 bp length and the second sequence is around 85,000 bp while number of points on one axis is around 1,000).
- Due to the large number of nucleotides we need to determine the degree of similarity between subsequences of different lengths to decide if a dot will be plot or not.

To determine the degree of similarity between two numerical subsequences of different lengths, *m* and *n*, we used correlation coefficient calculated for two sequences of length *n*, *(m-n)* times (using a sliding window), then determine the average coefficient.
Several experiments were conducted using the following parameter combinations: *L=9* (divisor of alpha satellite length, 171 bp), M_m=*1, 2, 3, and 4*. To mention that having to do with a graphical representation for very long sequences, we performed the results evaluation in a visual manner, by identifying areas with specific visual patterns of repeated sequences (based on intensity and image contrast).
Figures 10-17 shows best dot plots results obtained for sequence AC010523 (10-13) and AC136363 (14-17) considering each value for M_m and possible combinations for distances (in Step-2 and Step-3) and consensus type (on each axis is represented position in the analyzed sequence).

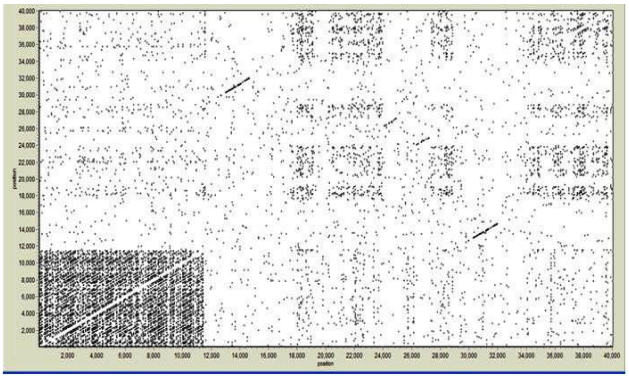

Figure 10. AC010523, M_m=1; Hamming-Jaro, MC consensus.

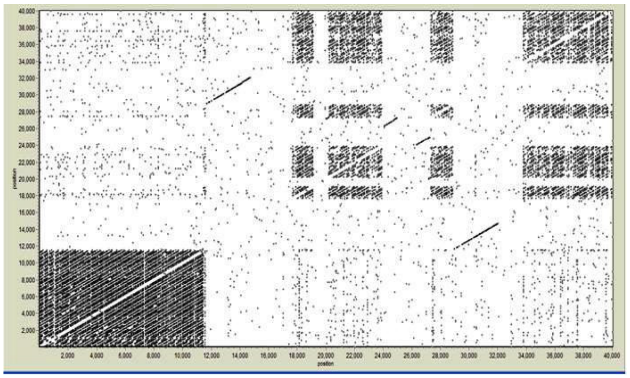

Figure 11. AC010523, M_m=2; Hamming-Damerau Levenshtein, MGCF consensus.

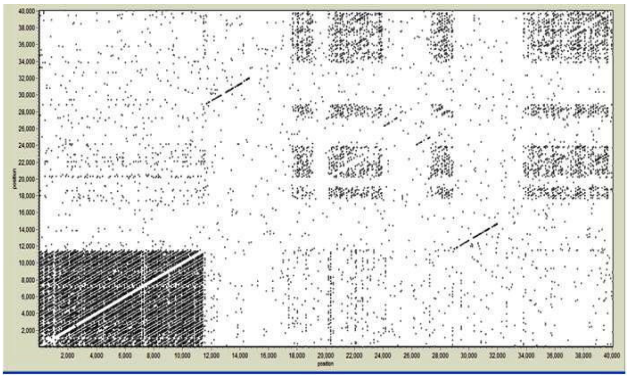

Figure 12. AC010523, M_m=3; Hamming-Levenshtein, MC consensus.

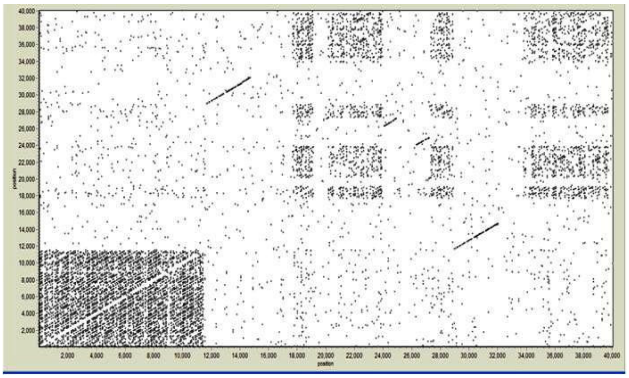

Figure 13. AC010523, M_m=4; Damerau Levenshtein-Hamming, MGCF consensus.

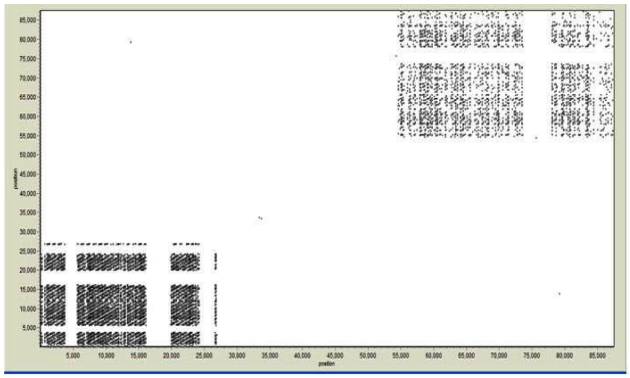

Figure 14. AC136363, M_m=1; Hamming-Hamming, MGCF consensus.

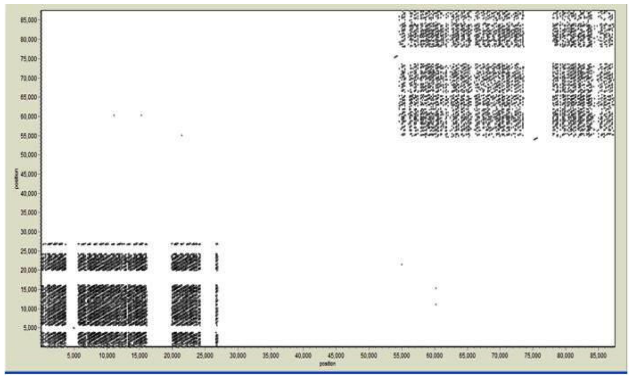

Figure 15. AC136363, M_m=2; Hamming-Hamming, MGCF consensus.

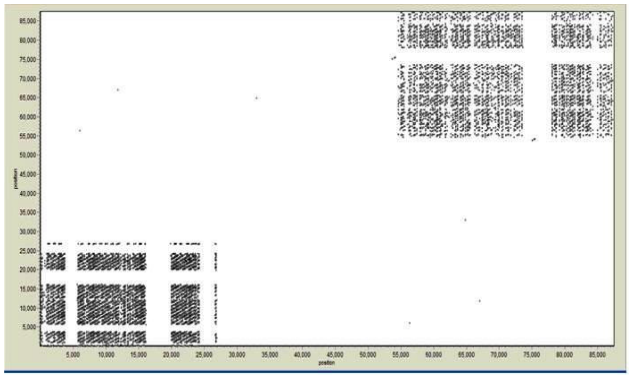

Figure 16. AC136363, M_m=3; Levenshtein-Hamming, MC consensus.

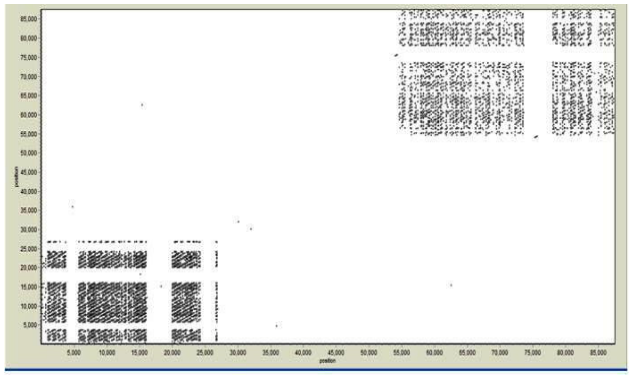

Figure 17. AC136363, M_m=4; Jaro-Damerau Levenshtein, MC consensus.

As one can see:

- We have repeats detected for $M_m=1$ (which is a serious constraint in this case). Using Jaro distance in Step-3 can benefit in quality of results.
- In case of $M_m=2$, Hamming distance in Step-2 and Hamming or Damerau-Levenshtein distance in Step-3 gives best results.
- In case of $M_m=3$, combination of Hamming-Levenshtein distances lead to better results.
- In case of $M_m=4$, using Damerau-Levenshtein distance in Step-2 or Step-3 can benefit in quality of results.
- Most common based consensus and majority consensus with global appearing frequency cutoff give the best results.
- Jaro distance use in Step-3, for $M_m=3, 4$ leads to worst results.

Finally, some considerations related to execution time:

- In case of short sequences, the influence of different distances and consensus sequence over execution time is not noticeable.
- For long sequences, the situation changes significantly:
  - Combinations of Levenshtein and Damerau-Levenshtein distances lead to higher execution times by up to an order of magnitude;
  - Evaluation of the consensus sequence with local appearing frequency cutoff leads to execution times by 10-20% higher.

IV. CONCLUSIONS

An original nucleotide sequence representation and a mapping algorithm are used to provide a single numerical sequence for DNA repeats detection which includes information about repeats length. The mapping algorithm uses DNA distances to determine similar sub-sequences, then to evaluate the distance between the consensus sequence of similar sub-sequences and each sub-sequence.

Several experiments were done using four distances in each stage and four ways to calculate the consensus sequence.

There is no single distance leading to good results in all conditions. However, some conclusions can be drawn from these experiments.

In stage of determination of similar sub-sequences:

- For a lower number of admissible mismatches ($M_m=10$-15% of L), using Jaro distance can benefit in quality of results. This can be explained by the fact that this type of distance is based on common groups of nucleotides which must be present for few admissible mismatches.
- For medium values of admissible mismatches ($M_m=15$-25% of L), Hamming distance gives good and very good results. Similar results are obtained with other distances but with increasing execution time.
- For a higher number of admissible mismatches ($M_m>25\%$ of L), for which number of candidate sequences is high, other distances should be used (Levenshtein, Damerau-Levenshtein or Jaro distance) for better results. This can be explained by the fact that these types of distances consider, additional, the insertion, and substitution, deletion and transposition operations.

For the second stage (determination of the distance between the consensus sequence of similar sub-sequences and each sub-sequence), Hamming distance gives best results followed by Levenshtein and Damerau-Levenshtein distance. This is probably because, at this stage, other operations outside of the substitutions are less useful.

In terms of the calculation of the consensus sequence, majority consensus with global appearing frequency cutoff and most common based consensus give the best results.

The results are not uniform and depend on the characteristics of searched repeats: length, number of admissible mismatches. If you know what you are looking for then you can choose the distance and type of consensus sequence which gives the best results.

REFERENCES

[1] Y.Wexler, Z.Yakkini, Y.Kashi, D.Geiger, "Finding Approximate Tandem Repeats in Genomic Sequences", RECOMB'04, March 27–31, San Diego, California, USA, 2004.
[2] Lim KG, Kwoh CK, Hsu LY, Wirawan A., "Review of tandem repeat search tools: a systematic approach to evaluating algorithmic performance", Brief Bioinform. 2012 May 29 (on line).
[3] P.G.Pop, A.Voina, "Numerical Representations Involved in DNA Repeats Detection Using Spectral Analysis", Studies in Informatics and Control, vol. 20, no. 2, pp.163-180, 2011.
[4] R.W.Hamming, "Error detecting and error correcting codes", Bell System Tech. J., 29 (2): 147–160, 1950.
[5] V.I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals", Soviet Physics Doklady 10, pp. 707–10, 1966.
[6] F.J.Damerau, "A technique for computer detection and correction of spelling errors", Communications of the ACM, vol.7, no. 3, pp. 171-176, 1964.
[7] M.A.Jaro, "Probabilistic linkage of large public health data files", Statistics in Medicine 14, pp. 491–498, 1995.
[8] T.D. Schneider, "Consensus Sequence Zen", Applied Bioinformatics 1(3) 111-119, 2002.
[9] http://www.ncbi.nlm.nih.gov/genbank/
[10] D. Sharma, B.Issac, G.P.S Raghva, R.Ramaswamy, "Spectral Repeat Finder (SRF): identification of repetitive sequences using Fourier transformation", Bioinformatics, 20(9), pp. 1405-1411, 2004.
[11] Alkan C, Ventura M, Archidiacono N, Rocchi M, Sahinalp SC, Eichler EE. (2007), Organization and evolution of primate centromeric DNA from whole-genome shotgun sequence data, PLoS Comput Biol. 2007 Sep; 3(9):1807-18.
[12] Yankov, D., Keogh, E., Lonardi, S., "Dot plots for time series analysis", Proc. 17th IEEE International Conference on Tools with Artificial Intelligence, 2005, pp 159 – 168.