Convolution and Fast Fourier Transform to Compare Symbol Sequences*

Anna Molyavko[1[0000–0003–1016–0131]], Evgenia Karepova[2[0000–0002–6515–2932]], Mikhail Sadovsky[2[0000–0002–1807–0715]], Vladimir Shaidurov[2[0000–0002–7883–5804]] and Igor Borovikov[3[0000–0002–0861–0001]]

[1] Siberian Federal University, 79 Svobodny st., Krasnoyarsk, 660041, Russia
[2] Institute of computational modelling of the Siberian Branch of the Russian Academy of Sciences, 50/44 Akademgorodok, Krasnoyarsk, 660036, Russia
[3] Electronic Arts, Redwood City, California, USA
{msad, e.d.karepova}@icm.krasn.ru

Abstract. We propose a new method to compare and analyze symbol sequences based on the convolution function calculation, where the latter is defined over the binary numeric sequences obtained by a specific transformation of the original symbol sequence. The method allows highly parallel implementation and it is of great value for the insertion/deletion mutations search. To calculate the convolution function, a fast discrete Fourier transform is implemented. Some genomic applications are provided and discussed. The applications are used to illustrate and overcome the problem of signal/noise selection, and alignment localization.

Keywords: Pattern Recognition, Anomaly Detection, Parallel Computation, InDel, Genome Comparison, Knowledge Retrieval.

1 Introduction

Here, we present a new method to compare and/or search for common subsequences in symbol sequences based on convolution. Additionally, fast Fourier transformation is used to compute the convolution, as well as a special representation of the symbol sequences under consideration. We illustrate it with a few biologically inspired examples. Also, the studies reveal some difficulties in the method implementation, signal/noise extraction and localization of the coinciding subsequences being the most acute among them.

Currently, alignment is the most popular method to compare two (or several) sequences, either exact matching, or with some errors. In spite of tremendous investigations both in hardware and software for alignment, this method is still very complicated and has a number of drawbacks which are impossible to eliminate. The worst of them are divergence, arbitrariness in the fine function determination, and

* Copyright c 2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR-WS.org/Vol-2727/paper14.pdf

very low efficiency for InDel mismatches search (see, e. g. [1, 2, 8, 7] for details). The proposed method is free from these crucial disadvantages. Moreover, it is highly potential for coarse grained parallelism in various implementations.

In brief, the method implies the following steps.

Preprocessing. Each symbol sequence under consideration must be converted into K binary ones, where $K = |\aleph|$ is the capacity of the alphabet. Next, one sequence ($\mathfrak{T}_2$, for certainty) must be inverted. Then, expand each binary sequence to the length $L = N_1 + N_2 - 1$, adding zeros (upright, for certainty). Finally, expand each binary sequence of the lenth L to the nearest upper power of 2, to proceed to a fast Fourier transform (FFT).

Processing. Calculation of Fourier images of those binary sequences.

Postprocessing. Term by term multiplication of the images, thus obtaining a new sequence.

Result. Calculation of the inverse Fourier transform to obtain a convolution.

2 Method Description

2.1 Basic Concepts

Convolution $\mathrm{S} = \mathrm{A} * \mathrm{B}$ of two number sequences $\mathrm{A} = \{a_i\}_{i=0}^{N-1}$ and $\mathrm{B} = \{b_i\}_{i=0}^{L-1}$ is the sequence $\mathrm{S}\{s_i\}_{i=0}^{N+L-2}$ with

$$s_i = \begin{cases} \sum_{k=0}^{i} a_k b_{i-k}\,, i < L\,, \\ \sum_{k=0}^{L} a_k b_{i-k}\,, L \le i < N\,, \\ \sum_{k=0}^{N+L-1-i} a_{N-1-k} b_{i-N+1+k}\,, N \le i\,. \end{cases} \quad (1)$$

The convolution is a key tool to compare two sequences, to seek for a specific subsequence within the latter. To do the search, we must introduce the convolution $\mathrm{S} = \mathrm{A} * \widetilde{\mathrm{B}}$ of the inversed sequence $\widetilde{\mathrm{B}} = \{b_{L-1-i}\}_{i=0}^{L-1}$. Following (1), one obtains $s_0 = a_0 b_{L-1}$, $s_1 = a_0 b_{L-2} + a_1 b_{L-1}$, $s_2 = a_0 b_{L-3} + a_1 b_{L-2} + a_2 b_{L-1}$, and so on.

The convolution is a key tool to compare two sequences, to seek for a specific subsequence within the latter. To do the search, we must introduce the convolution $\mathrm{S} = \mathrm{A} * \widetilde{\mathrm{B}}$ of the inversed sequence $\widetilde{\mathrm{B}} = \{b_{L-1-i}\}_{i=0}^{L-1}$. Following (1), one obtains $s_0 = a_0 b_{L-1}$, $s_1 = a_0 b_{L-2} + a_1 b_{L-1}$, $s_2 = a_0 b_{L-3} + a_1 b_{L-2} + a_2 b_{L-1}$, and so on.

A brute force way to calculate the convolution of two sequences is rather hard. To overcome this problem, we consider the convolution as a product of two polynomials (of the power $L - 1$ and $N - 1$, respectively). In other words, we consider two number sequences A and B as sets of coefficients of the corresponding polynomials. Thus, the convolution is converted to a product of two polynomials.

The next step comes from the well-known theorem stating that Fourier transform of a convolution is a product of Fourier transforms of the convolution of functions (sequences, in our case) under consideration. Hence, an idea is to apply (fast) Fourier transform to both sequences, multiply the Fourier images, and then to apply the inverse Fourier transform to obtain the convolution of the original sequences. Fourier transform is, in turn, the convolution. Meanwhile, there is a specific algorithm of a

very fast calculation of Fourier image of any number sequence called fast Fourier transform (FFT).

Let $\mathbb{F}$ denote FFT; it transforms a number sequence A into a sequence $\mathrm{A}' = \mathbb{F}(\mathrm{A})$ of the same length $N-1$. Let $\mathbb{F}^{-1}(\mathrm{A}') = \mathrm{A}$ denote the inverse FFT. Let now introduce the operation $\mathrm{X} \bullet \mathrm{Y}$ for two number sequences $\mathrm{X} = \{x_i\}_{i=0}^{N-1}$ and $\mathrm{Y} = \{y_i\}_{i=0}^{N-1}$ of the same length:

$$\mathrm{X} \bullet \mathrm{Y} = \{x_i y_i\}_{i=0}^{N-1} \tag{2}$$

2.2 Algorithm Description

Consider two finite symbol sequences $\mathrm{P} = \{p_i\}_{i=0}^{N-1}$ and $\mathrm{Q} = \{q_i\}_{i=0}^{L-1}$ from the alphabet $\aleph = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$. The algorithm comprises the following steps.

1. Inverse the sequence Q, yielding $\widetilde{\mathrm{Q}} = \{q_{L-1-i}\}_{i=0}^{L-1}$.
2. Change P and Q into $|\aleph|$ (that is 4, in our case) binary sequences, provided by the following:

— $(0,1)$ sequence $\mathrm{P_A}$ is obtained by the substitution of all the symbols A in P with 1, while all the others are changed for 0;
— $(0,1)$ sequence $\mathrm{P_C}$ is obtained by the substitution of all the symbols C in P with 1, while all the others are changed for 0;
— $(0,1)$ sequence $\mathrm{P_G}$ is obtained by the substitution of all the symbols G in P with 1, while all the others are changed for 0; finally
— $(0,1)$ sequence $\mathrm{P_T}$ is obtained by the substitution of all the symbols T in P with 1, while all the others are changed for 0.

Similarly, $\mathrm{Q} = \{q_i\}_{i=0}^{L-1}$ must be changed for $\mathrm{Q_A}$, $\mathrm{Q_C}$, $\mathrm{Q_G}$ and $\mathrm{Q_T}$. It should be kept in mind that here $\widetilde{\mathrm{Q}}$ sequence must be used.

3. Expand the sequences with zeros for further application of FFT to obtain a sequence of the length $N + L - 1$. To do this, all $2 \times |\aleph|$ binary sequences must be accomplished with zeros (upright, for certainty) to that length. Additionally, an effective implementation of FFT requires the sequence to be as long as the power of 2, so we must add zeros to obtain the length $\widetilde{N} = 2^{\lceil \log_2(N+L-1) \rceil}$.
4. Apply FFT to each of the binary sequences:

$$\mathrm{P_A}' = \mathbb{F}(\mathrm{P_A}),\quad \mathrm{P_C}' = \mathbb{F}(\mathrm{P_C}),\quad \mathrm{P_G}' = \mathbb{F}(\mathrm{P_G}),\quad \mathrm{P_T}' = \mathbb{F}(\mathrm{P_T}),$$
$$\mathrm{Q_A}' = \mathbb{F}(\mathrm{Q_A}),\quad \mathrm{Q_C}' = \mathbb{F}(\mathrm{Q_C}),\quad \mathrm{Q_G}' = \mathbb{F}(\mathrm{Q_G}),\quad \mathrm{Q_T}' = \mathbb{F}(\mathrm{Q_T}).$$

5. Following (2), multiply the relevant $\mathrm{Q_v}'$ (here v runs A, C, G and T) and sum them up:

$$\mathrm{S}' = \mathrm{P_A}' \bullet \mathrm{Q_A}' + \mathrm{P_C}' \bullet \mathrm{Q_C}' + \mathrm{P_G}' \bullet \mathrm{Q_G}' + \mathrm{P_T}' \bullet \mathrm{Q_T}'.$$

6. Apply the inverse FFT to S' to obtaining the convolution $\mathrm{S} = \mathbb{F}^{-1}(\mathrm{S}')$.

3 Results

We illustrate the method efficiency by applicaton to the search of transposons in plant mitochondrial genomes. We use the well-established results of the search provided by the Censor software which was carried out at the Siberian Federal University as a diploma project [3]. Transposon is defined as a chromosomal segment which can undergo transposition, especially in a bacterial DNA that can be translocated as a whole between the chromosomal, phage, and plasmid DNA in the absence of a complementary sequence in the host DNA. The typical length of the transposon ranges from 30 to 500 nucleotides. Meanwhile, the typical length of a mitochondrial DNA of a plant is about $\sim 10^6$ nucleotides. Fig. 1(a) shows the total pattern, for the entire chromosome, and Fig. 1(b) shows the detailed site of the exact matching transposon. InDel detection with the convolution comparison technique is shown in Fig. 1(e) (the total pattern, for the entire chromosome), and in Fig. 1(f) (the detailed site of the InDel mismatch). Figs 1(c) and 1(d) show the total (left) and the detailed (right) patterns for the case of 18 point mismatches.

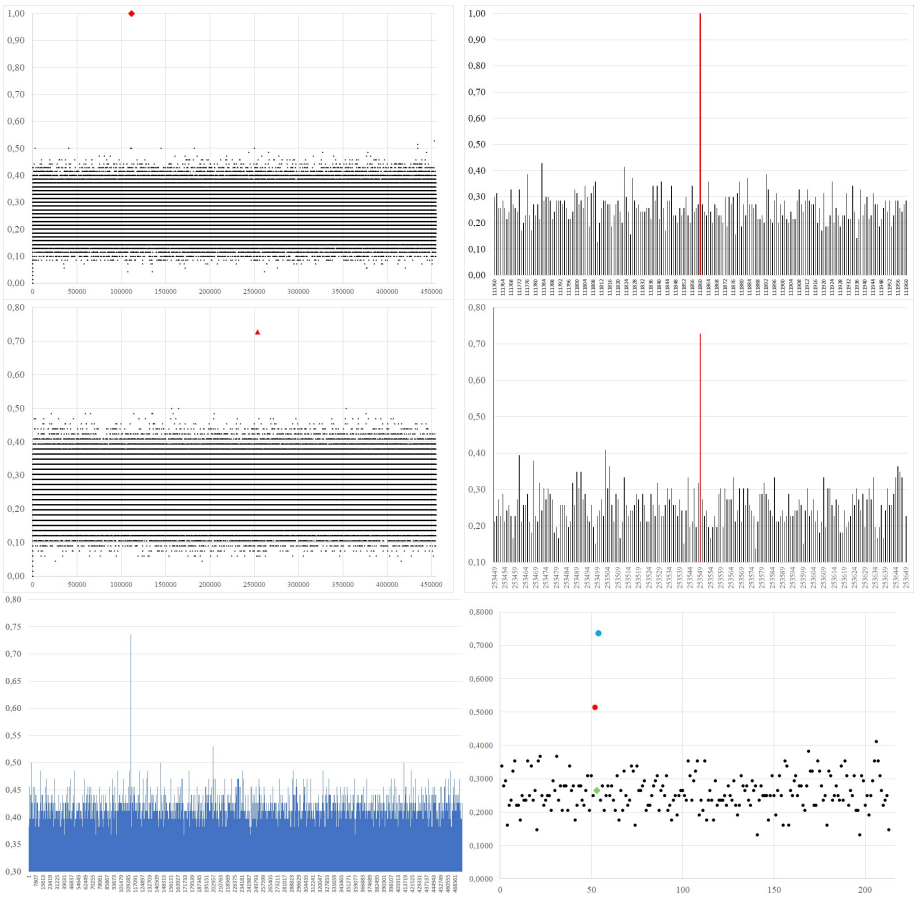

Fig. 1. Detection of exact matching with the convolution comparison technique.

Since the convolution yields the sequence $N_1 + N_2 - 1$ symbols long, then two essential problems arise: the former is localization, and the latter is signal/noise distinction [6]. Let us explain the first problem in few detail. Suppose, there is a common subsequence of the length $L \approx \sqrt{N}$, where $N = \min\{N_1, N_2\}$. Then, the peak indicating the exact (or highly likely) coincidence of two copies of the subsequence appears somewhere at the convolution. The point is that the exact location of this highly scored coincidence could be located within the resulting convolution with accuracy of $\pm L$, depending on the location of a copy of the subsequence in each sequence under comparison. To avoid this discrepancy, we illustrate the efficiency of the method with the search of very small (in terms of the localization problem) template subsequences that are the transposons.

Similarly, the noise-to-signal problem is also closely related to the length of the sequences under comparison. The worst case from the point of view of the signal deterioration occurs if two sequences of close lengths are compared; again, the genetic entities we use to illustrate the method allow one to bypass this problem at the moment.

Thus, below we provide three typical situations of a search for homology in two genetic sequences: these are exact matching, subsec. 3.1; point mutations (point mismatches), subsec. 3.2 and insertion/deletion mismatch, subsec. 3.3. Yet, a combination of these three situations might bring some problems; thus, some further studies should address them. To show that all these three types of mismatches can be detected by the convolution technique, we use *Aegilops speltoides* mitochondrial genome (AC NC_022666.1 in GenBank).

Three different transposons have been tested against the mitochondrial genome mentioned above. Fig. 1 illustrates the feasibility of the convolution based homology search for these transposons over the mitochondrial genome. In this figure, the left subfigures show the entire convolution trend; the right subfigures show the detailed pattern of the convolution variation at the site of homology.

3.1 Exact Matching Search

Exact matching search is the classical problem of pure and applied mathematics. A lot has been done in this area. Figs. 1(a) and 1(b) show the case of exact matching search. Due to the low resolution of the picture (the convolution total length is about 455 000) a point diagram is used in the left part (i. e., for the total convolution trend); the red dot shows the location of the site exactly matching the transposon. Fig. 1(b) shows an inset with the convolution, so that only 200 values are shown which are presented in bars. Obviously, the jump in the convolution is evident, which proves the feasibility of the method.

3.2 Point Mutations

Point mutation search has made great progress in the editing distance methodology [4, 5]. Probably, this type of mutations is the most suitable for alignment and relevant approaches. Figs. 1(c) and 1(d) show the result of the convolution based search of a

template with 18 point mismatches. Again, Fig. 1(c) shows the entire genome, and the red dot indicates the location of the highest homology to the transposon. Fig. 1(d) shows the detailed pattern of the convolution behavior; here the red bar represents the coincidence of the template and the site in the genome.

3.3 Insertion/Deletion

This is the hardest mismatch type from the point of view of detection with regular alignment tools. Figs. 1(e) and 1(f) show the result of the convolution calculation to detect the transposon. Note that the location of the site corresponding to the transposon is absolutely the same, as in the case of the exact search (see Figs. 1(a) and 1(b)). It should be said that here we tested another transposon differing in two insertions.

4 Discussion and Conclusion

Here, we present a new method of the homology search in symbol sequences for genetic applications. The method is based on the convolution calculation for digital sequences obtained from the symbol ones through special transformation. The method enables an innovative application of the well-known Fourier transform to dramatically speed up computations for important class of bioinformatics problems.

The results shown above demostrates the feasibility and efficiency of the new method to search for homologies in extended genetic sequences. Certainly, the method could be applied for the analysis of sequences of any nature ranging from linguistics to financial time series. Meanwhile, some further improvements are expected: signal/noise discrimination and localization of the site with the homology are the most challenging ones among them.

Also, it should be stressed that the method allows coarse grained parallelism:

1. the sequences P_ν and Q_ν where ν runs A, C, G and T could be treated simultaneously, and in parallel;
2. similarly, the greater is the number of the sequences to be compared pairwise, the faster could be the software implementation of the method;
3. parallelism grows up, as the capacity of an alphabet increases. Indeed, for amino acid sequences (that is very important in a number of applications), parallelism may accelerate the methods ten times and faster;
4. finally, very long biological sequences (up to 10^{12} symbols) could be treated by parts, with the subsequent concatenation of the convolutions.

The detailed discussion of these issues falls beyond the scope of this paper.

Acknowledgments. This work is supported by the Krasnoyarsk Mathematical Center and financed by the Ministry of Science and Higher Education of the Russian Federation in the framework of the establishment and development of regional Centers for Mathematics Research and Education (Agreement No. 075-02-2020-1631).

References

1. Kawam, A., Khatri, S., Datta, A.: A survey of software and hardware approaches to performing read alignment in next generatiob sequencing. IEEE/ACM transactions on computational biology and bioinfirmatics **14(6)**, 1202–1213 (2017)
2. Kaur, Y., Sohi, N.: Comparison of different sequence alignment methods – A survey. International Journal of Advanced Research in Computer Science **8(5)** (2017)
3. Kulishina, Y.: Mobile elements in mitochondrion genomes of *Poacea*. Bachelor diploma. Siberian Federal University (June 2020)
4. Levenshtein, V.I.: On perfect codes in deletion and insertion metric. Discrete Mathematics and Applications **2(3)**, 241–258 (1992)
5. Miller, F., Vandome, A., McBrewster, J.: Levenshtein Distance. VDM Publishing (2009), https://books.google.ru/books?id=TTzhQgAACAAJ
6. Molyavko, A., Shaidurov, V., Karepova, E., Sadovsky, M.: Highly parallel convolution mathod to compare DNA sequences with enforced in/del and mutation tolerance. In: International Work-Conference on Bioinfirmatics and Biomedical Engineering. pp. 472–481. Springer (2020)
7. Ng, H.C., Liu, S., Luk, W.: Reconfigurable acceleration of genetic sequence aignment: A survey of two decades of efforts. In: 2017 27th International Conference on Field Programmable Logic and Applications (FPL). pp. 1–8. IEEE (2017)
8. Wang, X.D., Liu, J.X., Xu, Y., Zhang, J.: A survey of multiple sequence alignment techniques. In: International Conference on Intelligent Computing. pp. 529–538. Springer (2015)