A Compact 2D Representation and Visualization of Large Symbolic Sequences and Applications for Comparative Genome Studies

Lubica Benuskova
Knowledge Engineering and
Discovery Research Institute
Auckland University of Technology
Auckland, New Zealand
E-mail: lbenusko@aut.ac.nz

Rene W. Kroon
University of Twente
Faculty of Electrical Engineering,
Mathematics and Computer Science,
7522 NB Enschede, The Netherlands
E-mail: r.w.kroon@student.utwente.nl

Ilkka Havukkala
Knowledge Engineering and
Discovery Research Institute
Auckland University of Technology
Auckland, New Zealand
E-mail: ilkka.havukkala@aut.ac.nz

***Abstract*—We show that representation of DNA sequences by means of Iterated Function Systems (IFS) can be used as a fast and useful 2D visualization tool for bioinformatics analysis and comparison of genomic data. The methods could be applied to any long symbolic sequences as well, with potential for fast searching as well.**

I. INTRODUCTION

The appealing visual structure of fractals and fractal-like structures has been used to visualize DNA sequences. Especially the Chaos Game Representation (CGR) is a method described in the literature in this context [1, 2]. The new contribution of this paper is to show the potential of Iterated Function System (IFS) as a fast visualization tool for long biological sequences like DNA and proteins for quick visual comparison of their whole sequences or subsequences. This is useful for a variety of comparative genome studies, and could be applied to text analysis as well.

A. Iterated Function Systems (IFS)

Definition of IFS [3]: IFS is a finite set of contractive transformations, such that

$$\Omega = \{\omega_i \mid \omega_i : X \to X, i \le n\} \qquad (1)$$

where $X = \langle 0,1 \rangle$. Starting point can be an arbitrary point in X. Fig. 1 contains the illustration of the following IFS of four transformations in the space $X = \langle 0,1 \rangle^2$:

$$\begin{aligned} \omega_T(x,y) &= (0.5x + 0.5, 0.5y) \\ \omega_A(x,y) &= (0.5x, 0.5y + 0.5) \\ \omega_G(x,y) &= (0.5x, 0.5y) \\ \omega_C(x,y) &= (0.5x + 0.5, 0.5y + 0.5) \end{aligned} \qquad (2)$$

Every transformation contracts X to its quarter of a unit square. A limit set of points emerging from an infinite application of the IFS is called the IFS attractor. For long complex series of symbols, like DNA sequences, it can have a fractal-like self-similar appearance.

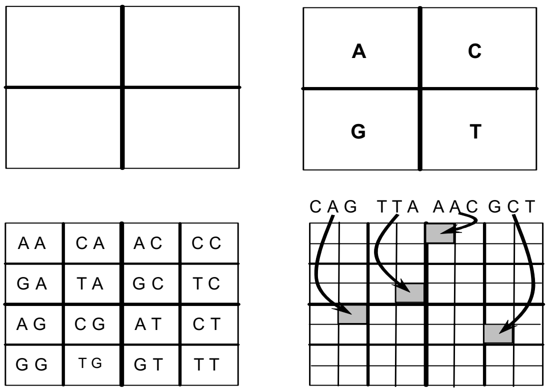

Fig. 1. Illustration of IFS Mapping. Modified from [4].

II. RESULTS

A. IFS method of 2D representation of DNA data

For each successive nucleotide encountered in a DNA sequence the transformation set (2) is applied. The starting point can be arbitrary, but should be within the unit space. First characters will not have the same coordinates, but for very large sequences this does not matter as we are interested in the IFS attractor. The fractal trajectory of a DNA sequence can be visualized by putting a dot in the unit space X at every coordinate that is the result of the transformation of successive nucleotides by means of the above IFS. The space displayed in Fig. 1 can in theory display an infinite amount of different sequences. In practice this is limited by representation of floating-point numbers and display resolution.

The set of transformations (2) groups a set of strings by their suffix, meaning that strings with a longer corresponding ending are closer to each other in the space. By choosing slightly different formulae it is also possible to organize the space so that similar prefixes are ordered close to each other. This can be of interest for other applications than DNA research, *e.g.* searches of large text databases. The symmetry of the formulae, however, is relevant to DNA research, as forward and reverse strands lead to mirrored points, but maintain the same overall structure.

To overcome the resolution problem we have discretized the unit space into a finite number of regions, where each region represents a suffix of length N. The resulting array is then of size 2^N by 2^N. With this table ready we can instead of just plotting the points keep counts of the occurrence of each suffix of length N. This means that we now have an alignment-free method [5] to compare DNA sequences. More importantly, the resulting figure remains readable and one can easily visually see and compare different structures and subsequence occurrences in the genome.

B. Example of IFS visualization of DNA data

An example of visualization of human chromosome 4 in a log scaled histogram can be seen in Fig. 2.

The resulting histogram in Fig. 2 still shows structure in DNA even after capturing 190 million bases. The big white area left and below the center diagonal line and other white areas indicate that the suffix ‘CG’ is sparse in human chromosome 4. The A to T diagonal indicates a lot of repeats of various combinations of A and T. Visual comparison of this histogram with that of chromosome 4 of mouse (M*us musculus*) showed a remarkable similarity. In the more remote pufferfish (T*etraodon nigroviridis*) the ‘CG’ sparseness was also observable, but less obvious, but bee (A*pis mellifera*) data showed totally different patterns. So at a glance we can put up the hypothesis that sparseness of ‘CG’ in DNA sequences could be a vertebrate property. If we had to rely on probability tables this conclusion might not have been so obvious. Thus, IFS method of visualization can contribute to knowledge discovery in comparative genome studies. The main advantage of this method is its speed and compact representation of genomic information in a visual form.

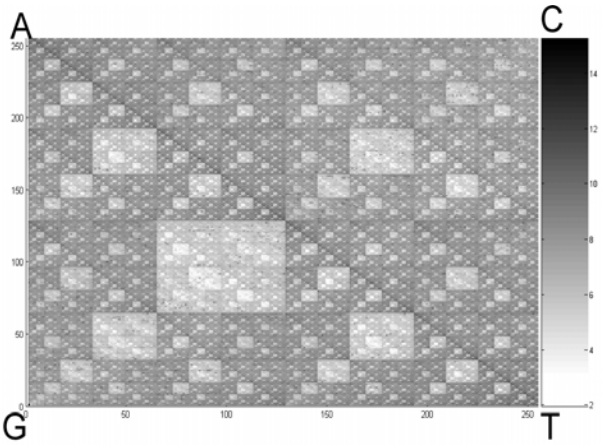

Fig. 2. IFS Chart Of The Human Chromosome 4. Scale On The Right Indicates Log(occurrence+1) Of The Substring. Axes Indicate Suffices Of Length N.

III. CONCLUSIONS

The presented method of visualization by means of IFS looks very promising. It is not limited to just four-letter alphabet of DNA, but can be extended into a P-dimensional hypercube, to represent an alphabet with 2^P letters. This higher dimensionality is more difficult to visualize, so alternative fractal methods should be studied that map for example to portions of a circle or other geometric structures.

Another useful implementation for DNA would be a program that takes a window ranging from a thousand to a million bases from a genomic sequence and then ‘slides’ through that sequence, displaying the histogram per region so that for instance non-coding and coding regions could be recognized, or other new features of particular regions discovered. Comparison of DNA coding and non-coding regions by means of CGR and IFS can be found in [4].

An application that can render the histogram of a certain zoomed region in real-time would be most useful, as creating the histogram is computationally cheap and therefore suitable for real-time interaction. Due to the nature of our IFS method it is possible to display the histogram of a selected area in higher resolution than the original full map.

In conclusion, the IFS is a versatile system that has great potential to improve DNA research by offering a powerful visualization method, and which can be extended into other alphabets as well. Software implementations of the method should be easy and could be very helpful for genome studies. Extending the visualization into texts with bigger alphabets requires a different iterated/fractal mappings of a dataset onto a two dimensional space. Additional applications for e.g. accelerated similarity searches of long genomic sequences or other symbolic texts seem also attractive.

ACKNOWLEDGMENTS

The authors wish to thank Prof. N. Kasabov for helpful comments. This work was supported by the NERF grant no. AUTX02001 and KEDRI AUT funds (L.B., I. H.) and by the Dutch internship funds (R. W. K.).

REFERENCES

[1] H. J. Jeffrey, "Chaos game representation of gene structure," Nucleic Acids Res., vol. 18, pp. 2163-2170, 1990.

[2] J. S. Almeida, J. A. Carrico, A. Maretzek, P. A. Noble, and M. Fletcher, "Analysis of genomic sequences by Chaos Game Representation," Bioinformatics, vol. 17, pp. 429-437, 2001.

[3] P. Tino, "Spatial representation of symbolic sequences through Iterative Function Systems," IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans, vol. 29, pp. 386-392, 1999.

[4] M. Cernansky, "Recurrent Neural Networks. PhD Thesis.," in http://www.ii.fmph.uniba.sk/~benus/courses/RNN-Cernansky.ps. Bratislava: Slovak Technical University, 2001.

[5] S. Vinga and J. Almeida, "Alignment-free sequence comparison-a review," Bioinformatics, vol. 19, pp. 513-523, 2003.