TCGR: A Novel DNA/RNA Visualization Technique

Donya Quick and Margaret H. Dunham
Department of Computer Science and Engineering
Southern Methodist University
Dallas, Texas 75275
dquick@mail.smu.edu, mhd@engr.smu.edu

Abstract

TCGR is a new method for analyzing DNA/RNA sequences by pattern distribution and for assessing similarity between multiple sequences. Previous approaches to analyzing DNA sequences ignore the temporal distribution drifts that occur along the DNA/RNA sequence. We argue for the development of innovative techniques to analyze sequences. We feel strongly that these approaches must involve a visualization component and be tied to new similarity techniques to aid in motif identification. DNA sequence investigation is a fuzzy classification problem. That is, two DNA strands may be slightly different in both length and structure, yet serve the same biological function.

1. Introduction

The examination of DNA/RNA sequences has been an area of strong research by biologists, computer scientists, and statisticians. Historically, comparative genomics has placed more focus on those components of the genome that are conserved between distant species. In contrast, the recent completion of the chimpanzee genome sparked new interest in identifying non-conserved genomic differences between closely related species, such as chimps and humans. Alignment of the chimp and human genomes reveals an overall identity of greater than 98.8% [1], but the divergent 1.2% still translates into tens of millions of differences given the 30 billion base pair content of these genomes. We assert that some of the key non-conserved differences between chimps and human biology are due to species-specific miRNAs and targets that direct key behavioral and physiological differences. *Micro-RNAs (miRNA)* are short non-coding RNA sequences that have been shown to regulate gene expression.

Recent studies on the basic structure of miRNA sequences, target sites, and pre-miRNA sequences intimate that there is something unique about the overall 3-dimensional fold of the polynucleotide structure that is key to miRNA biogenesis and target-site function [2]. However, simple sequence conservation and alignment is not sufficient to predict the existence of new animal miRNAs and thus new algorithms must be developed that are capable of 'capturing' subtle sequence composition, frequency, distribution, and other features that impact RNA folding and the potential to act as a miRNA.

The initial focus of our DNA/RNA sequence research is on miRNA visualization and prediction. However the scalability and universality of our approach is such that it is applicable to other sequence investigation applications. The algorithms themselves are generic. They learn what the DNA/RNA structure is from provided training data (using the wealth of information already uncovered about miRNAs).

2. Overview of TCGR

The original application of *Chaos Game Representation (CGR)* to visualize DNA nucleotide distributions and sequences first appeared in 1990 [3]. The *Frequency CGR (FCGR)* shows the frequencies of oligonucleotides using a color scheme normalized to the distribution of frequency of occurrence of associated patterns. The FCGR was first investigated by Deschavanne in 1999 [4].

Our contribution centers around a novel visualization approach that captures temporal nucleotide distribution shifts that occur along the length of DNA/RNA. ***TCGR (Temporal Chaos Game Representation)*** is an innovative approach to analyzing and visualizing DNA sequence data that involves creating a generalized overview of the content of a set of sequences [5]. Unlike other algorithms that are dependant on the exact order of

nucleotides, TCGR shows the structure of the data set by showing the distribution of short nucleotide patterns along the length of the data set. Proximity of the subsequences is more important in TCGR than exact order. The nucleotide subpatterns are counted in all sequences in a sliding window of fixed length that moves down the data set, generating a row of data each time it shifts. TCGR can also be used to show the overall structure of sets of sequences and is not strongly affected by insertions, deletions, or SNPs (single nucleotide polymorphims) contained in related sequences. Each of the parameters supplied to the TCGR algorithm has a significant effect on the output.

3. Visualization Algorithm

To produce a TCGR for a set of one or more DNA sequences, a sliding window is used to count sub-patterns, which are referred to as subsequences. The window has a fixed length and starts at the beginning of the sequence set. Within the window, all subsequences of a specified, fixed length are counted and tallied for all sequences. Once counting is complete within a window, the window is moved down by an offset specified by the degree of window overlap. The counting process is repeated, and the window continues to move down the data set until it either reaches the end or is only partially filled (extending beyond the end of the data set).

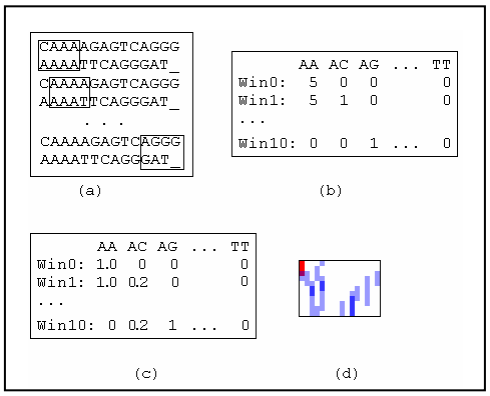

Figure 1. A small data set of two sequences is analyzed with a sliding window (a) converted to counts (b), then frequencies (c), and finally a TCGR (d).

When counting is finished, a 2-D matrix of integer values will have been formed, with one row representing a window and each column representing the counts for a specific sub-pattern. The maximum count value is found for the data set, or the user supplies a maximum value, and all cells are divided by that value to produce a matrix of frequencies.

The matrix of frequencies is visualized using a color scheme or grayscale gradient to represent frequencies on a scale of 0.0 to 1.0 (Figure 1). The simplest scheme is to use grayscale frequency conversions where 0.0 is white and 1.0 is black. An alternative that makes hot/cold spots more visible is to use a color scheme where 0.0 is white, 0.50 is blue, and 1.0 is red (Table 1).

Color Schemes for TCGR Visualization

Frequency	Color	Grayscale
0.00	White	White
0.50	Blue	Gray (50%)
1.00	Red	Black

Table 1. Two simple color schemes for representing data in TCGR.

TCGR does not consider the order of the subsequences as they are represented in columns. We have analyzed data with the convention that the columns for single nucleotides will correspond to A, C, G, and T in that order. For subsequences of size two, the columns will be in the order of AA, AC, AG, AT, CA, CC, …, TT. This pattern is used for all subsequence sizes.

3.1. Effects of Parameters and Alignment

The properties of TCGR visualization make it a novel method of analyzing DNA sequences, since the data is analyzed in sections rather than as a continuous string and the analysis method can be applied to both single sequences and sets of multiple sequences. The visualization process enables the easy identification of distribution differences along the sequence, particularly when the data is aligned prior to applying TCGR. Applications of TCGR include the identification of CG-islands, motifs, and binding sites.

The effects of subsequence size, window length, manual scaling, and window overlap, as well as the interaction between some of these parameters can cause large differences in TCGR output. Additionally, results can be affected and improved using sequence alignment algorithms, and by choosing subsequence sizes and a window length suitable for the data set. Changes in the parameters and the use of alignment can lead to more accurate data representation, but if improperly applied, can also distort or over-simplify the data.

3.2. Subsequence Size

Increasing the subsequence size, n, increases the number of columns by 4^n. Larger subsequence sizes also cause the output to become sparser for constant window lengths (Figure 2). Increased sparseness in the data is due partly to the fact that there are fewer opportunities to record larger subsequences within a window than smaller subsequences, and longer subsequences also have a lower probability of occurring in the same abundance as smaller ones. Hotspots become isolated and easier to identify in certain subsequence size ranges depending on the sequences and the window length used. Since hotspot distribution and the width of the TCGR are affected with larger subsequence sizes, keeping a constant subsequence size when comparing data sets is necessary.

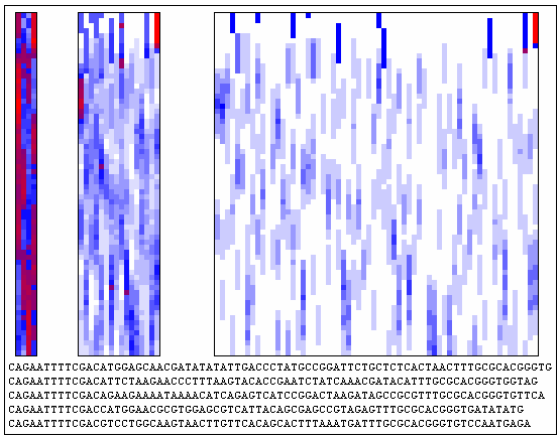

Figure 2. Effect of subsequence sizes 1, 2, and 3 from left to right respectively on synthetic data with a constant window size of 10, and a constant overlap of 9. The input data set is shown below the image.

3.3. Maximum Count Value

After the initial counting of subsequences has taken place, the integer counts are converted to frequency values based on either the largest count contained in the data set or a user-supplied value greater than or equal to that. The effect of supplying a maximum count value is manual scaling of the frequencies. When comparing data sets with the same number of sequences for hot spots, manual scaling should be used such that the counts are subjected to the same maximum value when converted to frequencies. If comparing for general structure, manual scaling may not be necessary, especially if output from a single sequence is compared to the output from a large set of multiple sequences. However, when comparing two same-size sets, not using manual scaling may not give an accurate idea of the degree of similarity between the two sets.

3.4. Window Length

For a constant subsequence size and with maximal window overlap, as window length increases, the data becomes denser (Figure 3). The window length to subsequence size becomes important since changing the ratio causes hot spots and cold spots to appear with different intensity. With small window lengths, hotspots will stand out, but cold spots may not be as meaningful. More hotspots may appear with higher window length to subsequence size ratios.

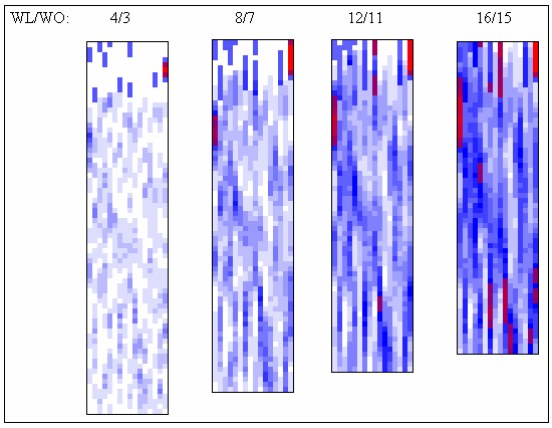

Figure 3. The data set from figure 2 with subsequence size 2 showing the effect of increased window length (WL) with maximized window overlap (WO).

3.5. Window Overlap

When window overlap is maximized such that the window only moves by one nucleotide each time, adjacent windows will show redundant data. This redundancy may be removed by decreasing the amount of window overlap, but this should be done with caution. The degree of overlap present between windows is also an important parameter, because decreasing the overlap between windows can cause data loss or distortion.

Despite redundancy, when the window overlap is close to the maximum possible, it will give a more accurate representation of the data than TCGRs done with progressively less window overlap. Problems posed by shrinking window overlap include the

merging of nearby hotspots in the same column, as well as the loss of very small hotspots occurring only in a small number of consecutive windows. Too little window overlap will result in over-generalized data that does not accurately represent the sequences, and therefore, window overlap should be maximized.

3.6. Sequence Alignment

Alignment is also an important factor in TCGR visualization of DNA sequence sets. Different degrees of alignment in the sequences will produce different results with TCGR, affecting the location and strength of hotspots. Hotspots may appear in aligned data that are not seen in unaligned data due to small differences in sequence length. Combined effect of parameters also impacts the representation of data sets with TCGR, so the parameters need to be considered relative to each other (such as the window length to subsequence size ratio) as well as individually. If data purely aligned to the starting points of the sequences, common regions become progressively less likely to show up in TCGR. Poorly aligned conserved regions downstream will not show up as prominently as those upstream unless alignment

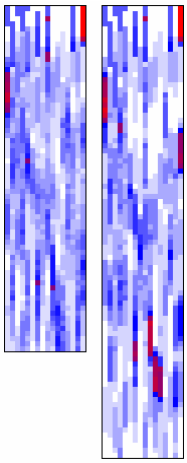

Figure 4. Effect of alignment on the same data set from figure 2 for subsequence size 2, window length 10, and window overlap 9. The alignment was produced with G-Align using a gap penalty of 135 and the default scoring matrix.

is used (Figure 4). Therefore, alignment is important to maximize the appearance of conserved regions.

4. Prediction

TCGRs provide a great visualization tool for DNA/RNA sequences. However, they serve other functions as well. Being based on the existence of count vectors, these mathematical versions of the images provide the basis for prediction techniques.

The *Extensible Markov Model (EMM)* is a very powerful modeling tool. The time series view of TCGR easily lends itself to be modeled by an EMM. An EMM is a time varying Markov Chain (MC). At any point in time, when viewed as a static graph it is a MC. However, over time the structure of the graph changes. Both the number of nodes in the graph as well as the transition probabilities vary. Thus the EMM is both a graph and a learning algorithm. The EMM takes the advantage of distance-based clustering for spatial data as well as that of the Markov chain for temporality. EMM achieves an efficient modeling by mapping groups of closely located real world events to states of a Markov chain. Further information concerning EMM can be found in [6].

The TCGR count input can be viewed as both spatial and temporal. Spatiality is defined by the sub-patterns being examined. Temporality is based on the sliding window. With the EMM model, at any time t a probability of a target event E occurring at some time in the future can be calculated. At any time, t, we can view the input as represented by a vector of n numeric values: $E_t = <S_{1t}, S_{2t}, \ldots, S_{nt}>$. For the miRNA prediction problem, t is actually the sliding window number. Each element, S_{it}, contains the count vector for sub-pattern *i* at time *t*. The miRNA prediction problem can be viewed as predicting whether the EMM (which was constructed using known miRNAs as training data) accurately models a given input sequence. Given an input sequence, the likelihood that it is modeled by an EMM can be determined by multiplying the transition probabilities found along the path constructed in the EMM as the input sequence is mapped (clustered) to EMM nodes.

Over the past several years, scientists have identified numerous miRNAs that exist in many different species. In most cases, biologists find miRNAs by molecular biology techniques that biochemically enrich for and then sequence small RNAs extracted from tissues or organisms of interest [7]. Recent studies have also exploited bioinformatics algorithms to predict miRNAs based on the presence of hairpins, other structures associated with the presence of miRNAs and conserved sequences across species. Most of these *in-silico* miRNA prediction algorithms rely heavily on cross-species sequence conservation. However, it is hypothesized that species-specific miRNAs contribute and explain the biological diversity. Recent experiments have discovered and confirmed a

number of miRNAs that do not have close homologs in the sequenced genomes available. Recently, Bentwich et al. [8] proposed a miRNA prediction method to find both conserved and nonconserved human miRNAs. They used structural features including hairpin length, loop length, stability score, free energy per nucleotide, number of matching base pairs and bulge size, and sequence features including sequence repetitiveness, regular and inverted internal repeats and free energy per nucleotide composition. However, their method still uses cross-species sequence conservation to make predictions. Nam et al. [9] proposed a paired hidden Markov model (HMM) as a general miRNA prediction method to identify close homologs as well as distant homologs. Conceptually, our approach has the following advantages:

1. It lends itself to capturing potential *long-range interactions* between nucleotides. One well-known drawback of the Markov Models for sequence classification problem is their inability of capturing long-range interactions. Although higher order MMs can be developed, they are difficult to train and computationally expensive. However, because the EMMs in our case take as input not the sequence directly, but the TCGR count vectors, which are computed from the sequences, we can use a first order MM and simply let the TCGR algorithm capture potential interactions within the sliding window.
2. The EMMs can reduce the number of states by merging "similar" states. This is particularly appealing for miRNA classification. Currently, there are only a small number of experimentally validated miRNAs, thus the size of the training set is very small. A MM with fewer states means that it will require less training data.

5. Conclusion

TCGRs represent a new class of DNA/RNA visualization tools. Careful use of parameters and alignment in TCGR can generate a more accurate data representation. The parameters need to be chosen carefully to have the best data representation, and alignment is important for doing TCGR with multiple sequences. The ability of alignment to facilitate the location of hotspots indicating conserved regions is especially important.

TCGRs provide the basis for a novel miRNA prediction technique that is completely independent of cross-species similarities. As such its applicability is general and quite novel. Ongoing studies have demonstrated the potential benefit of this approach [5].

6. Related Links

The java-based TCGR visualization program, count generator, and alignment tool G-Align can be found at http://engr.smu.edu/~dquick

7. Acknowledgements

The authors want to acknowledge the contributions of Jim Waddle, Monnie McGee, and Yuhang Wang to some of the wording provided in this abstract as well as invaluable contributions to background ideas for initial TCGR research.

7. References

[1] C. S. a. A. Consortium, "Initial sequence of the chimpanzee genome and comparison with the human genome," *Nature*, vol. 437, pp. 69-87, 2005.

[2] N. Rajewsky, "microRNA target predictions in animals," *Nat Genet*, vol. 38 Suppl 1, pp. S8-S13, 2006.

[3] H. J. Jeffrey, "Chaos Game Representation of Gene Structure*," Nucleic Acids Research*, 1990, vol 18, pp 2163-2170.

[4] P.J., Deschavanne, A. Giron, J. Vilain, G. Fagot and B. Fertil, Genomic Signature: Characterization and Classification of Species Assessed by Chaos Game Representation of Sequences," *Molecular Biol. Evol, 1999, vol 16, pp 1391-1399.*

[5] Dunham et al, "Visualization of DNA/RNA Structure using Temporal CGRs," *IEEE BIBE Conference Proceedings,* pp171-178, 2006.

[6] Margaret Dunham, Yu Meng, and Jie Huang, "Extensible Markov Model", *Proc. IEEE Int'l Conf. Data Mining* (ICDM 04), 2004.

[7] E. Berezikov, E. Cuppen, and R. H. Plasterk, "Approaches to microRNA discovery," *Nat Genet*, vol. 38 Suppl 1, pp. S2-7, 2006.

[8] I. Bentwich, A. Avniel, Y. Karov, R. Aharonov, S. Gilad, O. Barad, A. Barzilai, P. Einat, U. Einav, E. Meiri, E. Sharon, Y. Spector, and Z. Bentwich, "Identification of hundreds of conserved and nonconserved human microRNAs," *Nat Genet*, vol. 37, pp. 766-70, 2005.

[9] J. W. Nam, K. R. Shin, J. Han, Y. Lee, V. N. Kim, and B. T. Zhang, "Human microRNA prediction through a probabilistic co-learning model of sequence and structure," *Nucleic Acids Res*, vol. 33, pp. 3570-81, 2005.