Chaos game representation of gene structure

H.Joel Jeffrey
Northern Illinois University, DeKalb, IL, USA

Received June 5, 1989; Revised and Accepted March 20, 1990

ABSTRACT

This paper presents a new method for representing DNA sequences. It permits the representation and investigation of patterns in sequences, visually revealing previously unknown structures. Based on a technique from chaotic dynamics, the method produces a picture of a gene sequence which displays both local and global patterns. The pictures have a complex structure which varies depending on the sequence. The method is termed *Chaos Game Representation (CGR)*. CGR raises a new set of questions about the structure of DNA sequences, and is a new tool for investigating gene structure.

INTRODUCTION

The Chaos Game

During the past 15 years a new field of physics has developed, known as 'non-linear dynamics', 'chaotic dynamical systems', or simply 'chaos' [1,2]. Central to much of the field are questions of the structure of certain complex curves known as 'fractals'. These curves, which are, in a certain mathematical sense 'infinitely complex', have become quite well-known in the last few years, especially with the appearance of articles accessible to the non-mathematician, such as those by Dewdney in *Scientific American* [3,4].

The Chaos Game is an algorithm which allows one to produce pictures of fractal structures, using paper and pencil or, obviously, a computer. In simplest form, it proceeds as follows:

1. Locate three dots on a piece of paper. They can be anywhere, as long as they are not all on a line. We will call these dots vertices (for reasons that will become clear in shortly).
2. Label one vertex with the numerals 1 and 2, one of the others with the numerals 3 and 4, and the third with the numerals 5 and 6.
3. Pick a point anywhere on the paper, and mark it. This is the initial point.
4. Roll a 6-sided die. Since in Step 2 the vertices were labelled, the number that comes up on the die is a label on a vertex. Thus, the number rolled on the die picks out a vertex. On the paper, place a mark half way between the previous point and the indicated vertex. (The first time the die is rolled, the 'previous point' is the initial point picked in Step 3.) For example, if 3 is rolled, place a mark on the paper half way between the previous point and the vertex labelled '3'.
5. Continue to roll the die, on each roll marking the paper at the point half way between the previous point and the indicated vertex.

One might expect that this procedure, if repeated many times, would yield a paper covered with random dots or, perhaps, a triangle filled with random dots. Such is not the case. In fact, if the Chaos Game is written on a computer (using a random number generator for the 'die'), and is run for several thousand points, the result is as shown in Figure 1. This figure has been known in mathematics for many years, and is termed the 'Sierpinski triangle', after the mathematician who first defined it.

On seeing this result, one obvious question is, 'What if you start with a different number of initial points?'

For five points, six, or seven initial points the chaos game produces a figure with visible patterns (pentagons within pentagons, a striated hexagon, or heptagons within heptagons), but for eight or more point the game yields essentially a filled-in polygon, except that the center is empty.

With four initial points, however, the result is different. It is not squares within squares, as one might expect; in fact there is no pattern at all. The chaos game on four points produces a square uniformly and randomly filled with dots.

The picture produced by the chaos game is known as the *attractor*

Iterated Function Systems

Mathematically, the chaos game is described by an *iterated function system* (IFS). An IFS is a set of pairs of linear equations, each pair of the form $x = ax + by + e$, $y = cx + dy + f$. Each pair of equations gives the formula for computing the new value of the x and y coordinates. For example, the chaos game on three points is played by marking the new point in the paper half way between the previous point and the particular vertex. Suppose coordinates of the vertices are (0, 0), (0, 1), *and* (1, 0). Then, if 3 is rolled, vertex 2 is indicated, and the coordinates of the new point are given by $x = 0.5 \cdot (x + 0) = 0.5x$, and $y = 0.5 \cdot (y + 1.0) = 0.5y + 0.5$, and similarly for the other vertices.

With three vertices, and one equation per coordinate, we need six equations. A more compact notation is to write $w(x, y) = (ax + by + e, cx + dy + f)$.

Thus, there is one equation (usually called a 'mapping' or 'map') for each, and each map is given by the 6 coefficients a through f. For the Sierpinski triangle, the maps are:

w_1	0.5	0	0	0.5	0	0
w_2	0.5	0	0	0.5	0	0.5
w_3	0.5	0	0	0.5	0.5	0.5

Since the choice of map is determined by a die (or random number generator), each map has an associated probability, all

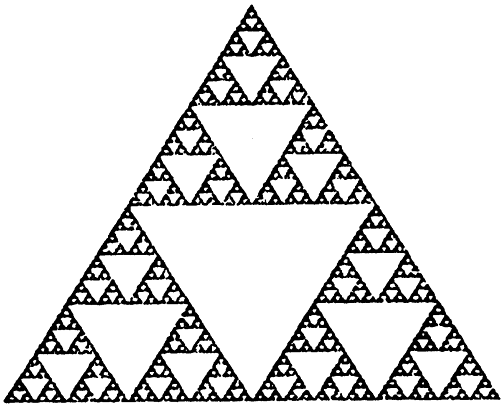

Figure 1. The Result of the Chaos Game on Three Points.

equal in the case of the unloaded die. If the probabilities are not equal, the shape of the attractor is unchanged, but the shading may be [1].

In tabular form, including the probabilities, we can use the following compact notation, which is known as the *the IFS code*:

Table 1: IFS Code for the Sierpinski Triangle

w	a	b	c	d	e	f	p
1	0.5	0	0	0.5	0	0	0.33
2	0.5	0	0	0.5	0	0.5	0.33
3	0.5	0	0	0.5	0.5	0.5	0.33

The IFS code for the filled-in square is:

Table 2: IFS Code for the Filled-in Square

w	a	b	c	d	e	f	p
1	0.5	0	0	0.5	0	0	0.25
2	0.5	0	0	0.5	0	0.5	0.25
3	0.5	0	0	0.5	0.5	0	0.25
4	0.5	0	0	0.5	0.5	0.5	0.25

Non-random Sequences

When the chaos game is played with 3, 5, 6, or 7 points, the quality of the random number generator is not very important; the same figure is produced, although it may take longer to 'fill out'. With 8 or 16 (or, to a lesser extent, 4) points, such is not the case.

Quite by chance, the author and a colleague (G. M. Henry) discovered that in these cases the random number generator can make a very significant difference. As noted above, with a good random number generator, the Chaos Game on 8 points produces an almost-filled octagonal. However, when the game is played using Turbo Pascal 3.0, which has a flawed random number generator [9], elaborate patterns are visible, resembling a circle within a circle, the circles connected by 8 (or, respectively, 16) spidery lines.

Further, not all flawed random number generators produce a visible pattern from Chaos Game; using DOS Basic (Version 2.1) RND, which is quite a poor random number generator, the Chaos Game on eight points produces no visible patterns.

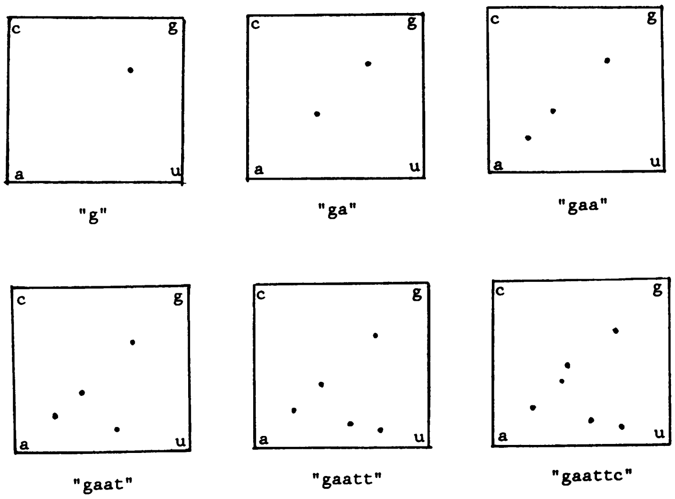

Figure 2. CGRs of the first 6 bases of HUMHBB.

Chaos Game Representation of DNA Sequences

Intuitively, non-randomness means that a sequence has 'structure'. If a sequence of numbers is used to produce an attractor for an IFS code, as described above, and that attractor has visually observable then we have, intuitively, revealed some underlying structure in the sequence of numbers.

The experiments desribed in the above section had shown that the Chaos Game can be used to display certain kinds of non-randomness visually.

This led to the following question:

Since a genetic sequence can be treated formally as a string composed from the four letters 'a', 'c', 'g', and 't' (or 'u'), suppose that, rather than random numbers, we control the Chaos Game with DNA sequences? Instead of 'rolling a 4-sided die', use the next base (a, c, g, t/u) to pick the next point. Each of the four corners of the square is labelled 'a', 'c', 'g', or 'u'; if a 'c', for example, is the next base, then a point is plotted half way between the previous point and the 'c' corner.

Example: The first 6 bases of the GenBank sequence HUMHBB (human beta globin region, chromosome 11) are 'gaattc'.

1. The first 'g' is plotted half way between the center of the square and the 'g' corner.

2. The next base, 'a', is plotted half way between the point just plotted and the 'a' corner.

3. The base 'a' is plotted half way between the previous point and 'a' corner.

4. Next, 't' is plotted half way between the previous point and the 't' corner.

etc.

Plotting these six bases, we obtain Figure 2.

As with the initial points of the Sierpinski triangle, little significance is visible. However, if we continue for the entire 73,357 bases of HUMHBB, we obtain Figure 3.

We have termed the resulting picture the *Chaos Game Representation* (CGR) of the sequence.

HUMHBB exemplifies a number of the characteristics of CGRs in general, and of vertebrate sequences in particular.

1. Perhaps the most obvious characteristic of this CGR is the almost empty area in the upper right quadrant (the g-quadrant). A smaller copy of this 'scoop' appears in the upper left, or c-

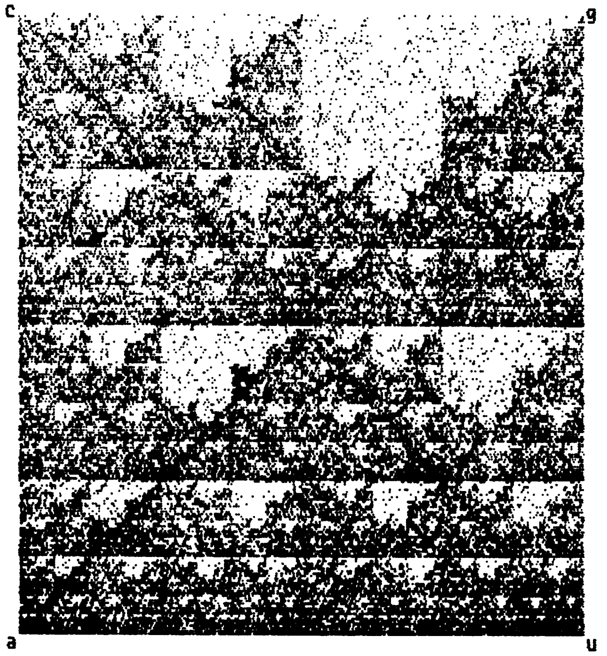

Figure 3. CGR of Human Beta Globin Region on Chromosome 11 (HUMHBB) (73,357 bases).

quadrant, presenting a double-scoop appearance. As discussed in the next section, each point in the CGR correpsonds to exactly one subsequence (starting from the first base), up to resolution of the screen. Therefore this graphic pattern indicates repeated patterns in the gene sequence. The same is true of any other visible pattern.

Crudely, the 'double-scoop' corresponds to a comparative sparseness of guanine following cytosine in the gene sequence; a 'g' is plotted half way between the previous point and the 'g' corner. (This is dicussed in more detail below.)

2. Note that any base will always be plotted somewhere in the quadrant with its label, since a base is always plotted half way toward its corner.

3. Copies of the double-scoop, one in the t-quadrant and one in the a-quadrant.

Looking at the top of the lower half of the picture, there are two copies of the double-scoop, one in the t-quadrant and one in the a-quadrant. Further, this continues: If we examine the picture in horizontal 'strips' (in halves, quarters, etc.), we see that at the top of each quarter-strip there are four copies; at the top of each eighth-strip there are eight, and so forth. It thus exhibits the property of *self-similarity* a concept very important in the study of fractals and chaotic dynamics. Formally, a figure is self-similar if a subset of it, with appropriate change of scale, has the same shape as the overall figure. ('Same shape' is formalized using the Hausdorff distance).

4. A rather noticeable feature of this CGR is the set of curves on the top of the upper left quadrant of the plot, curving from the solid block up and the the right.

5. Two visible characteristics of Figure 3 are related to the underlying mathematics: (1) The division into squares, sub-squares, etc., and (2) the self-similarity noted. If, for example, we use the IFS code for the filled-in square is used, but with the non-uniform probabilities shown in Table 3, we obtain Figure 4.

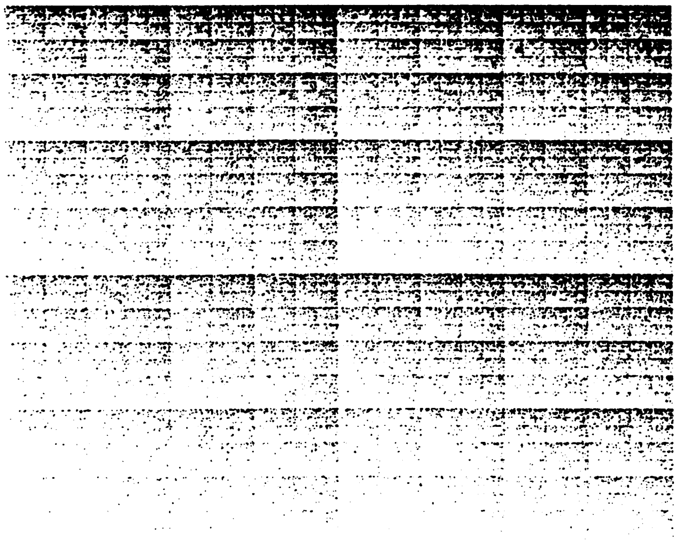

Figure 4. Square Attractor with Non-Uniform Probabilities.

This, however, is only part of the explanation; the other part is a characterization of the subsequences that are missing, or sparse, that lead to the structures shown in Figure 3. A partial characterization of these sequences is presented below.

The features found in the CGR of HUMHBB are found in a number of other genetic sequences. HUMALBGC (human serum albumin gene, complete) and HUMADAG (human adenosine deaminase gene, complete) are excellent examples. This pattern has been found (to date) only in vertebrate sequences and those of certain viruses such as HIV, hepatitis, and yellow fever.

Table 3: IFS Code for the Shaded Fill-in Square

w	a	b	c	d	e	f	p
1	0.5	0	0	0.5	0	0	0.1
2	0.5	0	0	0.5	0	0.5	0.2
3	0.5	0	0	0.5	0.5	0	0.3
4	0.5	0	0	0.5	0.5	0.5	0.4

Properties of the CGR of a DNA Sequence

The relation between the CGR and the DNA sequence is of course the central issue. Many of the aspects of that relation are unknown, or lack a mathematical characterization, at this time. Certain fundamental facts, however, can be noted.

1. The k-th point plotted on the CGR of a sequence corresponds to the first k-long initial subsequence of the sequence, *and no other subsequence* (up to the resolution of the screen). Thus, there is a one-to-one correspondence between the subsequences (anchored at the start) of a gene and points of the CGR.

2. Therefore any visible pattern in the CGR corresponds to some pattern in the sequence of bases.

3. As noted, the resolution of the screen limits the detail that may be shown on any one CGR. However, as with all fractals, including those generated by IFS codes, any portion of the picture may be magnified, revealing finer structure [1] Thus, if there is an area of interest in which suspected structure is obscured, it can be magnified to show the fine structure of the points and, therefore, the structure of the sequences yielding the points. This magnification is without limit (as long as there are more bases in the sequence).

4. Adjacent bases in the sequence are not plotted adjacent to

each other (except when the first point is close to a corner and the next base is the same); being close in the CGR does not mean being close in the sequence. Euclidean distance in the CGR therefore implies a new metric on subsequences, or bases.

5. The question of when two points close in the CGR represent similar sequences is a bit more complicated. In general, two close points may correspond to different sequences. For example, in Figure 2, note that the final point plotted, corresponding to the final 'c', is spatially close to the second point plotted, but between the first 'a' and the final 'c' is the sequence 'att'.

However, this situation can only occur if the two points, although close, are in different quadrants of the picture. Since a base is always plotted in its quadrant, any sequence will always be plotted somewhere in the quadrant of its last base, and conversely any two points in the same quadrant must have the same last base. Further, the notion of quadrant is recursive; each quadrant can be divided into quadrants, etc.

Thus, in Figure 2, 'g' is plotted in the g-quadrant. Then 'a' is plotted in the upper right of the a-quadrant (the one with its upper right vertex at the center of the entire figure), or what might be called the 'ga' sub-quadrant. Any point in the g-quadrant would be mapped to this subquadrant. Thus, 'a' produces a copy of the g-quadrant that is one-half the size (side length) of the g-quadrant, or one-fourth of the size of the entire picture. The next 'a' then produces a one-half size copy of the 'ga' sub-quadrant, the 'gaa' sub-sub-quadrant, and then 't' produces a 'gaat' sub-sub-sub-quadrant.

Further, due again to fact that a base is plotted in its quadrant, the converse holds as well: If two points are within the same quadrant, they correspond to sequences with the same last base; if they are in the same sub-quadrant, the sequences have the same last two bases; if they are in the same sub-sub-quadrant the have the same last three bases, etc.

Thus, we have the following:

Theorem 1. *In a CGR whose side is of length 1, two sequences with suffix of length k are contained within the square with side of length 2^{-k}. Further, the center of the square is give by the following recursive definition:*

(a) The center of the suffix of 0 *length is* (1/2, 1/2)

(b) The if the center of the square containing sequences with suffix w is at (x, y), then

i. the center of the square containing sequences with suffix wa is (x/2, y/2);

ii. the center of the square containing sequences with suffix wc is (x/2, (y+1)/2);

iii. the center of the square containing sequences with suffix wg is at ((x+1)/2, (y+1)/2);

iv. the center of the square containing sequences with suffix wt (or wu) is at ((x+1)/2, y/2).

Conversely, all points within this square correspond to sequences with this suffix.

6. As a consequence of 4 and 5, the visible patterns in the CGR represent global as well as local patterns. A density (or paucity) of points in a region corresponds to a large (or small) number of sequences with suffixes corresponding to the region. Also, since each square region (sub-, sub-sub-, etc. quadrant) corresponds to a particular suffix, any dense (or sparse) region corresponds the union of S_1, S_2, $\cdots$, in which S_i is the set of sequences with suffix i.

The resolution of most monitors or printers is such that the points for sequences with identical suffixes of length over 10 (maximum) are superimposed, in any given picture. However, the magnification capability noted in Point 3 means that there is no lower limit to the size of the square that can be displayed, and therefore no limit on the length of suffixes represented.

The CGR method thus provides a graphic way of displaying the composition of a sequence. The information is displayed so that interesting features can be noticed by eye.

7. Due to the correspondence between points on the CGR and the sequence, any mathematical characterization of the CGR is a characterization of the underlying sequence.

For example, it may be possible to find a technique for producing a mathematical description of the CGR of a sequence, using concepts from the IFS theory. If such a technique can be found, it will be a technique for producing a description of the DNA sequence.

As a result of these observations, we can say that in an intuitive sense the CGR represents both statistical properties of frequencies of bases as well as sequentiality properties— i.e., which bases follow others, immediately or later in the gene.

Generally, about 4000 base pairs are necessary for a sharply defined picture, although in many cases 2000 give a reasonably good approximation. To date we have not observed any CGRs in which the double scoop pattern appeared early, as the sequence was plotted, and was then covered up by further dots. Further, we have not found any cases in which one pattern began to emerge and then changed, as further bases were plotted, to another pattern. This seems to be significant, for it would indicate that many features of the genetic sequence are exhibited by an initial subsequence, and thus the examining the entire sequence may add no new information.

In the next section we present Chaos Game Representations of several different types of genes.

The CGR of Certain Groups of Genes

The key question, of course, about this or any other representation, is whether it yields biologically interesting observations. It appears at this point that it does. We have found several distinctive patterns, by examininng the CGR of a number of groups of genes. Using genetic sequence data from the Genbank data base (Release 55), we have discovered some characteristic patterns presented below.

Vertebrate CGRs

With two exceptions, every vertebrate sequence examined so far exhibits the characteristic pattern of HUMHBB (Figure 3). This pattern has not been found in any group other than vertebrates, with the exception of certain viruses (such as the HIV viruses). This is discussed below in the section on viral CGRs.

The results of Section 4 permit a partial characterization of the double-scoop pattern of Figure 3. Examining the figure, there is the general paucity of points in the cg-subquadrant, which corresponds to a paucity of subsequences ending in 'cg'. However, the sparse area is more complex geometrically; it can be decomposed into sub-sub-quadrants, etc., producing the rounded bottom of the scoop. Each of these smaller sparse areas corresponds to longer sparse suffixes. The paucity of one particular suffix produces a square sparse area in the CGR (as discussed below).

The characterization is partial in that we have as yet no mathematical description of the scarce suffixes producing the complex outline of the scoop.

The two exceptions mentioned are oncogenes and human

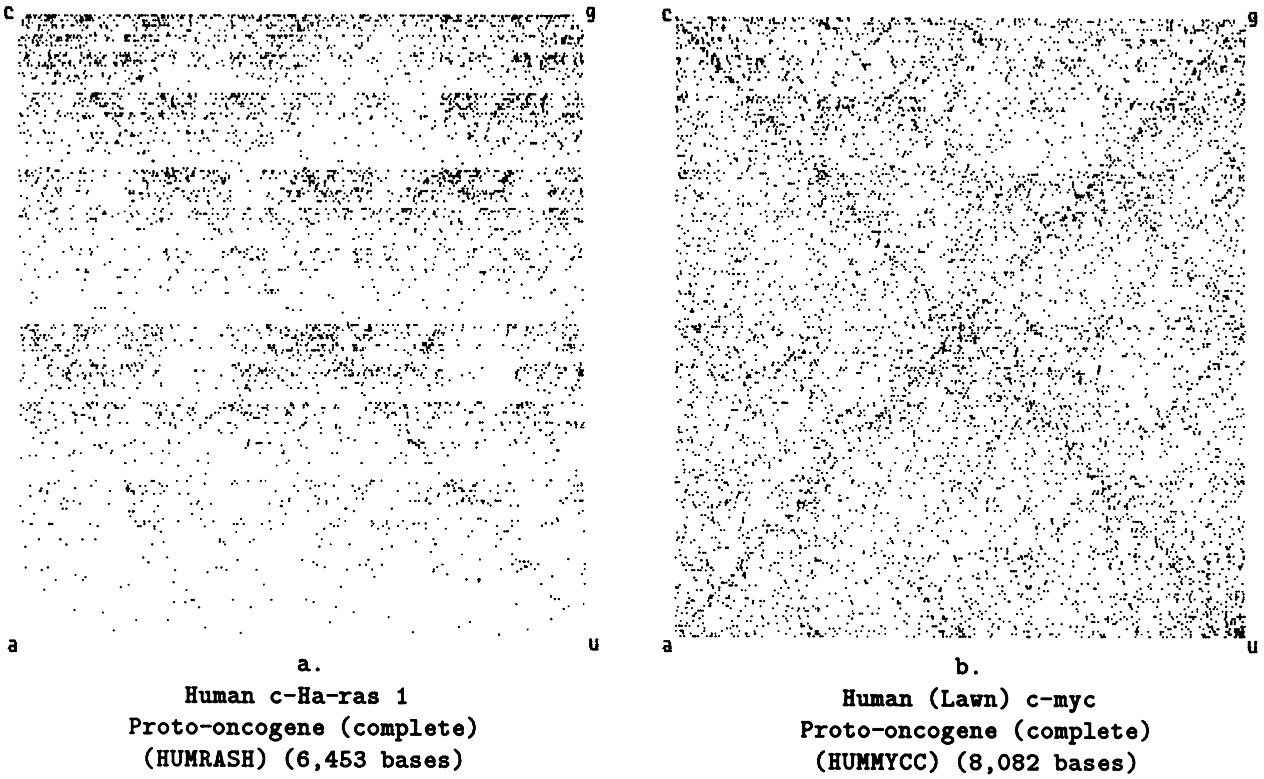

Figure 5. Oncogene CGRs.

ribosomal RNA. Oncogenes display an entirely different pattern, discussed below in the section of oncogene CGRs. The CGR of Human ribosomal RNA displays few distinctive features, although it is clearly unlike the randomly filled-in square. It does reveal a high density of c-g pairs, as shown by a densely filled-in c-g line.

Oncogenes examined to date have exhibited two characteristic patterns, shown in Figure 5.

Invertebrates

Invertebrate sequences in general exhibit less structure than those of vertebrates, the CGR appearing to be almost an even distribution of points.

Certain patterns are found, however, as in the nematode (C. elegans) major myosin heavy chain isozyme unc-54 I gene (CELMYUNC), which displays some clsutering and a diffuse band of points along the a-g diagonal.

Plants and Slime Molds

Several distinct types of CGRs may be found in plants, none like the vertebrate pattern. POTPATG, the solanum tuberosum gene for patatin, shows horizontal striations, with a sparse area near the upper portion of each striation, and a high density of points on the a-u axis.

Many other plant sequences show little discernible pattern. MZESUSYSG (Maize sucrose synthase, complete) and barley aleurain gene (BLYALR) are good examples.

Yeast genes show a different characteristic pattern, in many cases. It is more diffuse, although it displays a somewhat higher density of points along the lower portion of the a-g diagonal.

A quite distinctive pattern, is exhibited by several of the slime mold sequences. That for SLMMYHC (D. discoideum myosin heavy chain gene, complete) (Figure 6).

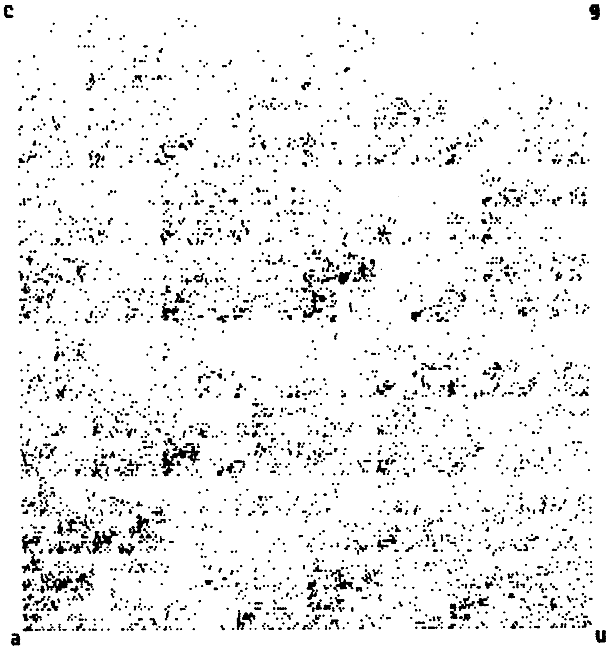

Figure 6. D. Discoideum Myosin Heavy Chain Gene (complete) (SLMMYHC) (6,680 bases).

This pattern is also exhibited, for example, by SLMDIRS1A (D. Discoideum transposon DIRS-1, complete).

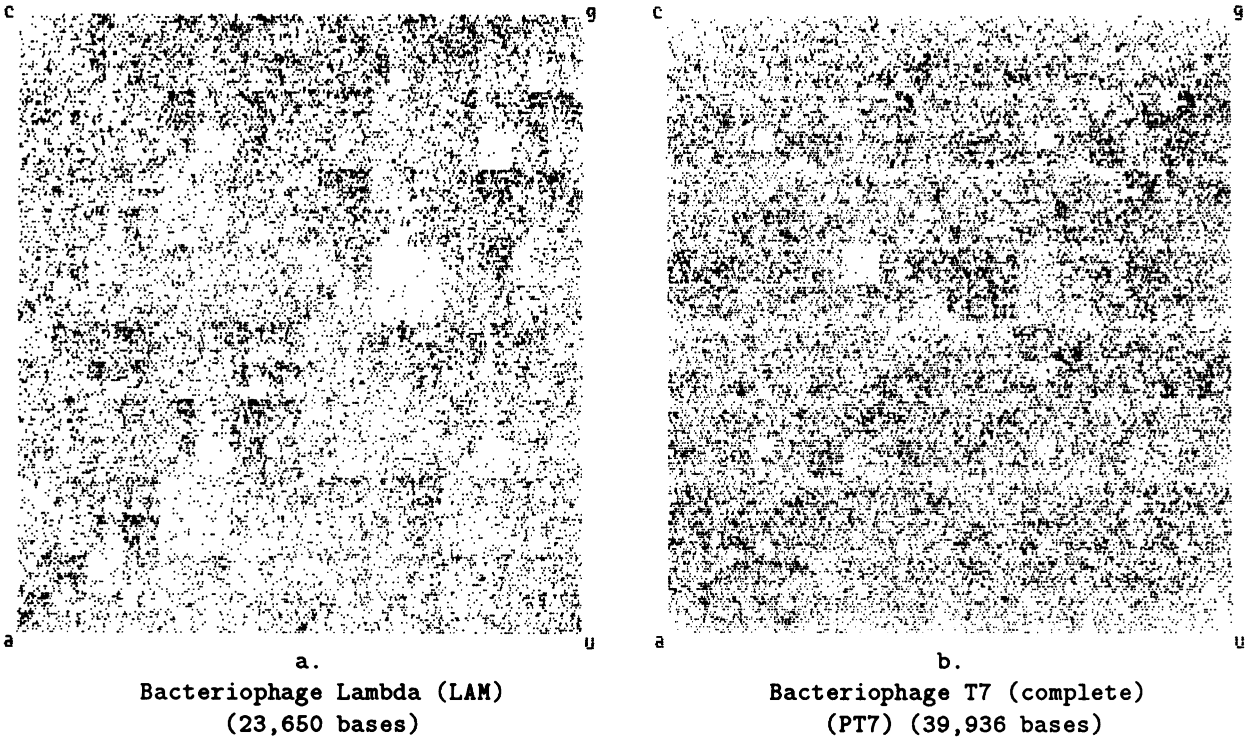

Figure 7. Phage CGRs.

Phages

Phage CGRs display some visually striking patterns, as shown in Figure 7.

Note, for example, the series of very sparsely populated squares below the a-g diagonal in Fig. 7a, and the almost empty square near the middle of Fig. 7b.

The square near the middle of Fig. 7b indicates the lack of subsequences ending in 'gatc'. As discussed in Section 4, every 'gatc' sequence produces a point in this square, and only 'gatc' sequences do. (And, in fact, there are exactly six 'gatc' sequences in PT7, and six dots inside the square.)

Figure 7a contains a similar situation: a larger very sparse square just below the a-g diagonal. This sparse sub-sub-quadrant indicates a lack of 'tag' sequences in lambda. Note that the largest sparse square is one-eighth the size of the entire CGR, corresponding to a suffix of length 3, while the sparse square in Fig. 7b is one- sixteenth the size of the CGR, corresponding to a suffix of length 4.

Bacteria

The CGR for bacteria investigated to date are in general fairly uniformly filled in, with the exception of a series of diffuse sparsely filled squares below the a-g diagonal. No non-bacterial sequences to date have shown this pattern. The CGR for ECOUNCC (E. coli operon encoding 8 subunits of ATP) is typical of this group.

Viruses

Viral CGRs show several well-defined patterns. Quite intriguing is that displayed by the sequence for the human T-cell lymphotropic virus (type III) (Figure 8). The double-scoop pattern is quite evident, along with diagonal 'striations' along the a-g axis. The pattern is similar to that of Fig. 3, but the sparse area is a union of a smaller number of squares than that of Fig. 3, indicating a simpler set of scarce sequences.

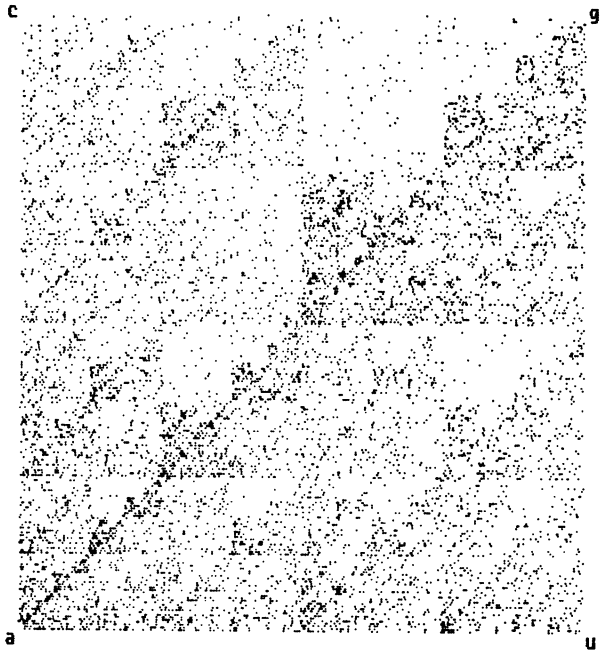

Figure 8. Human T-cell Lymphotropic Virus (Type III) (complete) (HIVPV22) (9,770 bases).

The same pattern is displayed by the CGR for HIVZ6 (human immunodeficiency virus type 1), HIVBRUCG (human lymphadenopathy-associated virus), HIVELICG (human

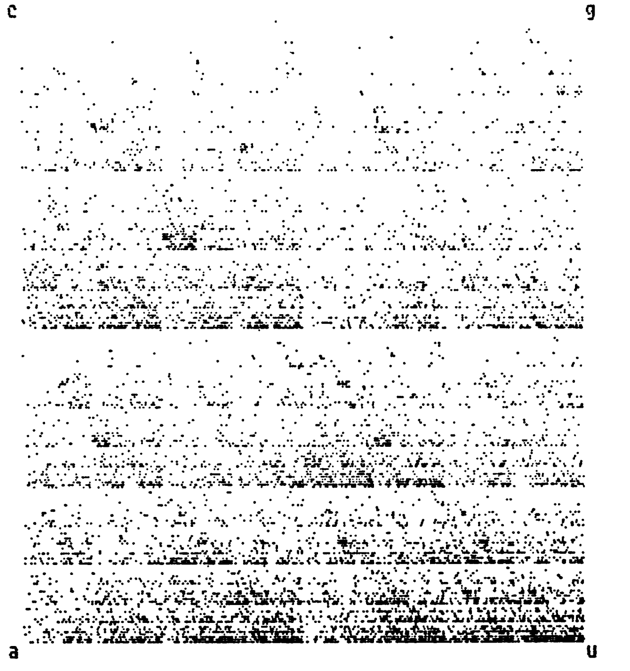

Figure 9. 'Theory and Practice of Knowledge Engineering' (12,508 bases).

lymphadenapthy virus), and HIV2ROD (human immunodeficiency virus type 2), YFV (yellow fever), and HPA (hepatitis).

Non-genetic Sequences

The CGR algorithm produces a CGR for any sequences of letters, plotting 'a', 'c', 'g', and 't' or 'u', while ignoring any other intervening characters. This leaves open the question of whether the CGRs we have observed, intriguing as they may seem, were simply a mathematical oddity, displayed by any sequence of letters. To investigate this possibility, we produced CGRs of several text files, of this and previous papers.

English text files do in fact produce CGRs (rather than, for example, simply the filled-in square), but they are unlike those of any genetic sequence found to date.

Figure 9 is the CGR of a paper the author has previously published on knowledge engineering.

Related Work

As iterated function systems are intimately related to non-linear dynamics and chaos [1], there are number of connections to that field. A. Mandell has used non-linear dynamics to analyze protein structure [7] and coding problems [8]. A number of related papers appear in [6].

The CGR involves treating a genetic sequence as an abstract string of symbols. As such, symbolic dynamics is clearly relevant. Symbolic dynamics [2], a topic in dynamical systems theory, associates strings of symbols with orbits of a dynamical system. It is a powerful tool for analyzing the orbits. It may be that this approach can be reversed: we may be able to find dynamical systems whose behavior is represented by a given DNA sequence.

Subshifts of finite type [2] may be relevant here, as they are a technique for analyzing the orbits of systems under restrictions as to which symbols can follow others. In most DNA sequences any base may follow any other, but it may be that we can find subsequences (such as exons) for which this does not hold. In such cases this approach may prove valuable.

Broadly, the CGR is a technique for studying the 'non-randomness' of genetic sequences. Statistical analysis of DNA sequences has been explored by a number of researchers; [10] and [5] are good examples.

To date we have relied solely on visual characterization of the patterns found in CGRs, both as to recognizing features and judging similarity of features. An objective, mathematical measure is needed. The pattern recognition literature reveals little in the way of a formal definition or characterization of 'pattern' that is directly applicable. However, digital image enhancement techniques are clearly applicable. The Hausdorff distance [1] has been used to formalize similarity of patterns, and may well prove valuable here.

Research into using particular sequences of numbers (or symbols) to control the chaos game (which is the concept of the CGR), and thereby analyze/characterize the sequences themselves, would be highly relevant. No such other work has been found. This appears to be a new area of study, in need of further investigation.

Open Questions

Chaos Game Representation has revealed an entirely new set of questions, most of which are unanswered at this pint. In this section we present a representative sample. It is intended to be provocative, rather than in any way complete.

The overall question of course is whether CGR can serve as a useful tool for investigating DNA sequences, and if so in what ways. The answer will depend in large measure on results obtained in addressing these and other specific questions.

1. The patterns found so far appear to be biologically meaningful, due to the 1 − 1 representation and to the fact that observable patterns in CGRs vary across groups of genes. The work on this topic, though, is just beginning. What correlations are there between observable patterns and CGRs and biologically interesting gene categories?

2. Mathematically characterize the sparse sequences that produce the 'double scoop' pattern (Fig. 3), which to date has been found only in non-oncogene vertebrate sequences and in some genes from viruses that can infect vertebrates? A partial answer is given in *The GCR of Certain Groups of Genes*, but it seems likely that considerably more can be said, both mathematically and biologically.

3. Since the double scoop is so common in vertebrate sequences, what can we say about the points that appear within the scoop—its sparse filling?

4. Can we find a mathematical representation of the CGR that makes it easier to represent/recogize biologically interesting groups of genes?

5. Suppose we calculate the successive points for each base, but plot only those bases in exons. Thus, the shape and location of the exons would be visible, without intron sequences 'covering them up'. What does this reveal?

6. The same question as 5, but rendering invisible all but the introns. (This question is related to [10], in which the randomness of introns is a focus of study.)

7. For any screen resolution, if the gene sequence is long enough many areas of the CGR become completely filled in. This suggests two possible enhancements. First divide the picture into small areas, and color the area according to how many points of the CGR appear in it. Second, plot three-dimensionally, where

the third dimension represents frequency of a CGR point's occurrence.

8. (Proposed by P. Senapathy [11]). Suppose we simply color exons differently from introns, or color each exon differently. What then is revealed about the intron/exon structure in the gene?

9. It is possible to apply the CGR algorithm to the codons, or to the amino acids they code for. However, doing so does not yield visible patterns; starting with so many vertices yields a figure almost randomly filled. However, additional techniques may yield insight into the structure of the amino acid sequences themselves. Three such that are under investigation are (1) Using four vertices, as with base CGRs, but plotting four acids per CGR; (2) Representing the number of the acid in numerical base 4, so that, for example, amino acid 18 is represented as the string of base-3 digits '20'. Any codon (or amino acid) would thus have a representation using only 4 distinct digits, and we could therefore apply the CGR algorithm to it; (3) Use one vertex per codon, but use the IFS code that tiles the polygon with copies of itself (as the IFS code for the square tiles the square with copies of itself).

In addition to these questions, several extensions to the technique suggest themselves, such as comparison of CGRs, displaying only the points in common (or not in common) between several CGRs. These may of course lead to further open questions themselves.

ACKNOWLEDGEMENTS

The interest, encouragement, and expertise of K. Palaniappan and P. Senapathy have been invaluable. The comments of the referees on earlier versions of this paper were extremely helpful. A particularly valuable contribution was made by one referee in pointing out the correspondence of sparsely populated square areas to sparse sequences with particular suffix.

REFERENCES

1. M. F. Barnsley, *Fractals Everywhere*. Springer-Verlag, New York, 1988.
2. R. L. Devaney, *An Introduction to Chaotic Dynamical Systems*. Addison Wesley, Redwood City, California, 1989.
3. A. K. Dewdney, Computer Recreations. *Scientific American*, August, 1985, pp. 16–20.
4. A. K. Dewdney, Computer Recreations. *Scientific American*, November, 1987 pp. 140–144.
5. G. Fichant and C. Gautier, 'Statistical Method for predicting protein coding regions in nucleic acid sequences,' *Computer Applications in the Biological Sciences*, Vol. 3 no. 4, 1987, pp. 287–295.
6. J. A. S. Kelso, A. J. Mandell, and M. F. Shlesinger (eds.), *Dynamic Patterns in Complex Systems*, World Scientific, Singapore, 1988.
7. A. J. Mandell, P. V. Russo, and B. W. Blomgren, 'Geometric Universality if Brain Alosteric Protein Dynamics: Complex Hydrophobic Transformation Predicts Mutual Recognition by Polypeptides and Proteins,' *Perspectives in Biological Dynamics and Theoretical Medicine: Annals of the New York Academy of Sciences*, V. 504, 1987, pp. 88–115.
8. A. J. Mandell, 'An Unstable Singularity Theory of Molecular Biological Coding; Calcitonin's Structures and Potencies,' *Dynamic Patterns in Complex Systems*, in Kelso et al. (1988), pp. 219–235.
9. S.K. Park and K. W. Miller, 'Random Number Generators: Good Ones are Hard to Find,', *Communications of the ACM*, Vol. 31, No. 10, October, 1988, pp. 1192–1201.
10. P. Senapathy, 'Origin of eukaryotic introns: A Hypothesis, based on codon distribution statistics in genes, and its implications.' *Proceedings of the National Academy of Science, USA*, Vol. 83, pp. 2133–2137, April, 1986
11. P. Senapathy, Personal Communication, April 17, 1989.