J. theor. Biol. (1998) **193,** 257–285
Article No. jt980698

Studying Artificial Life with a Molecular Automaton

LOUIS VARETTO

Centre for Protein Engineering and Laboratoire d'Enzymologie, Université de Liège, B6 Sart-Tilman, B4000 Liège, Belgium

(*Received on* 11 *September* 1996, *Accepted in revised form on* 23 *February* 1998)

In a previous paper, we proposed a molecular automaton which was an attempt to implement the "molecular logic of the living state" in an artificial biochemistry. This automaton is an artificial genetic system composed of two classes of interacting molecules, the informants and the transformers. In the present paper, we show that such a tangled hierarchy is susceptible to give rise to general hypercycles we called tanglecycles. These tanglecycles are shown to be self-reproducing autocatalytic metabolisms. They can be considered as genuine "acellular beings" in which the information is conserved in a dynamical loop. Our phenomenological study of tanglecycles reveals the emergence of properties usually associated with life, including self-reproduction, selection and chaos. In the field of the autocatalytic networks it represents a new approach in studying the self-organization of the matter with a prebiotic point of view.

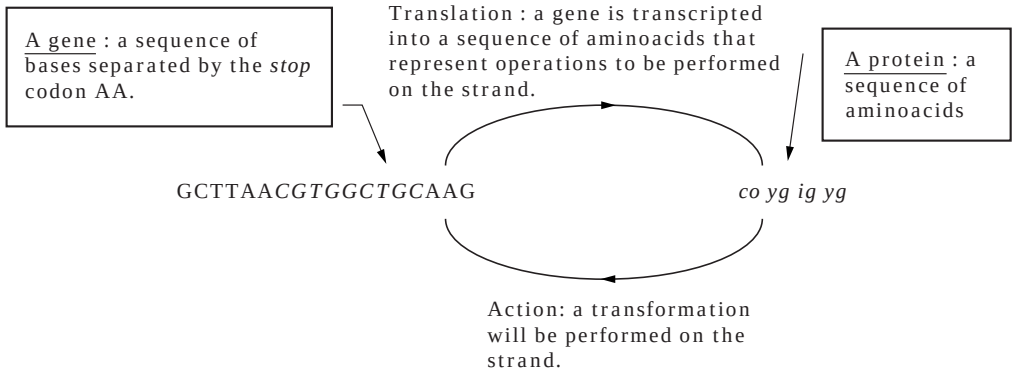
1. Introduction

1.1. TYPOGENETICS

Typogenetics: *an artificial genetic system* was presented in a preceding paper (Varetto, 1993). First introduced by Hofstadter (1979) in connection with his discussion on the "tangled hierarchy" of the DNA replication processes, we have computer-programmed an artificial genetic system equivalent to the actual natural genetic system in his principles. Typogenetics were also developed by Morris (1989), but only as a logical system.

Typogenetics includes two classes of strands, on one hand nucleic acids, formed from the characters A, C, G and T, and on the other hand proteins (typoenzymes) coded by the nucleic acids via a typogenetical code composed of 16 doublets (for more details about Typogenetics and our specific automaton, see the Appendix).

The typoenzymes are constituted from 15 different aminoacids, the 16-th doublet being a STOP codon. These aminoacids perform activities such as cutting, lengthening, copying, etc.



0022–5193/98/140257 + 29 $30.00/0

Each typoenzyme possesses a tertiary structure encoded by the program and determined by its primary structure. This tertiary structure dictates the enzyme specificity for the nucleic acid it originates from, i.e. it determines how the typoenzyme first interacts with the nucleic acid. Then, the nucleic acid–typoenzyme complex evolves under the catalytic effect of the typoenzyme, yielding the next generation of the nucleic acid. It can be lengthened, shortened, partially or totally copied according to the classical A–T and C–G pairing rules, or simply destroyed.

The different general transformations that a strand could undergo are summarized hereafter:

(i) condensation reactions of a polymer A with one or more monomers a, b . . .:

—lengthening:

$A + a + b + \ldots \rightarrow B$ with $B > A$

Example:

$$GC + G \xrightarrow{ig} GGC$$

This reaction indicates that a G base is introduced on the right of the binding unit (the binding unit is G according to the enzyme specificity) by the *ig* enzyme coded by the GC strand itself.

—formation of 2 or more polymers by complementarity (copy mode):

$$A + a + b + \cdots \rightarrow B + C$$

Example:

$$CCCGAGCATATC + 2T + G \xrightarrow{dg\text{–}co\text{–}el\text{–}dd\text{–}yol\text{–}ud} {}^{5'}CCGAGCATATC^{3'} \atop {}_{3'}\square\square\square TCGT\square\square\square_{5'} \Rightarrow$$

CCGAGCATATC + TGCT after separation

(ii) lysis reactions:

—shortening: $A \rightarrow B + a + b + \cdots$ with $B < A$

Example:

$$TGTAAG \xrightarrow{yg\text{–}yd\text{–}el} TGTAG + A$$

—cutting: $A \rightarrow B + C$

Example:

$$ACACTATGG \xrightarrow{cp\text{–}dd\text{–}yd\text{–}yg} ACAC + TATGG$$

With this system—a program written in a compiled basic language—one can start from any nucleic acid and then follow its evolution from generation to generation. It must be stressed that nucleic acids undergo deterministic transformations: everything is determined by the sequence of the starting nucleic acid and the typogenetics rules.

These rules being established, we tried to find a *self-replicator*, i.e. an original strand that reappears in duplicate (or more) in some future generation. We called such a strand a *selfR*. The *GC* strand was found to be a selfR, in fact the shortest member of a selfR family. In the preceding paper (Varetto, 1993), we determined some properties of this family, notably its exponential biphasic growth, its growing parameters (growth factor f_g), the influence of the maximum allowed strand length (ML) and the necessity of the presence of the selfRs for an exponential growth.

1.2. THE TANGLECYCLES OF TYPOGENETICS

When the generations are computed, some strands extend indefinitely. To avoid this problem, the length of any strand is limited. In the present work a limit of 80 bases was chosen. If a strand becomes longer, it is eliminated from the following generations. Under these conditions, it could be shown that the number of different possible strands amounts approximately to 2.10^{48} for the nucleic acids and to 10^{94} for the typoenzymes. Among all these strands a selfR family exists from which the shortest strand is GC. Taken as the original strand, GC gives two copies of itself in the same generation (generation 75). We can represent the GC ensemble by tangled cycles. These cycles include at least two closed selfRs loops, beginning and finishing by the same selfR I_1 [Fig. 1(a)].

In an article entitled *Selforganization of Matter and the Evolution of Biological Macromolecules*, Eigen (1971) proposed a "self-reproductive *hypercycle*". Hypercycles are multilevel hierarchical catalytic reactions. Eigen & Schuster (1979) introduced their theory as part of their study of the driving forces sustaining the prebiotic molecular evolution. The Eigen model [Fig. 2(a)] is a "cyclic hierarchy" in which many nucleotides consisting of a "positive" and a "negative" chain that reproduce themselves, are related together by an enzymatic hypercycle. Owing to this secondary loop, the different I_i cooperate, which enables information conservation (Eigen *et al.*, 1981; Eigen & Schuster, 1982). In fact, it can be considered that the Eigen hypercycle is a particular case of what we will call a *tanglecycle*.

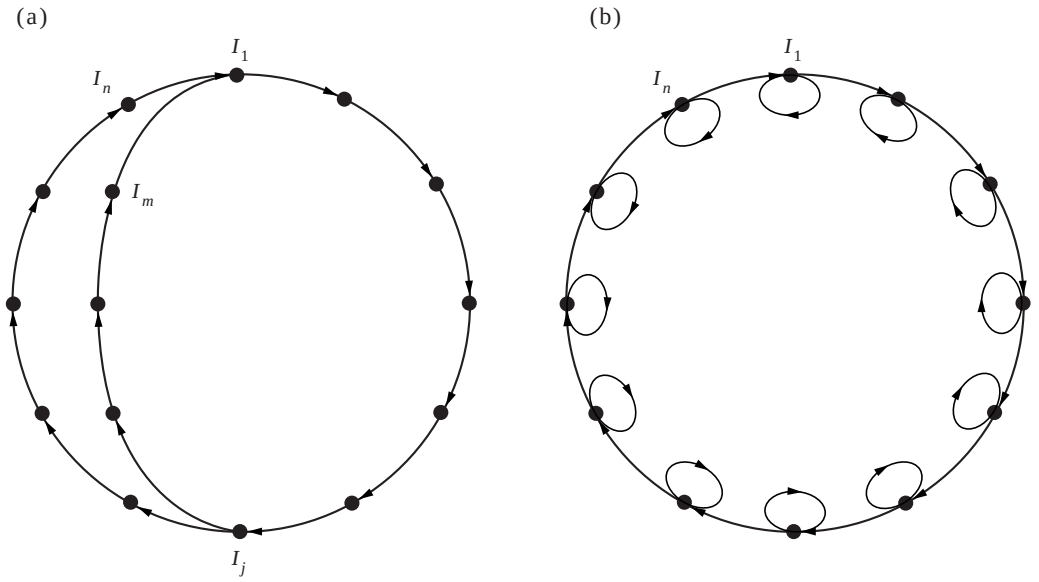


FIG. 1. (a) Minimal tanglecycle. I_i represents the different selfRs. For self-reproducing and growing, at least two tangled loops are required. For the GC family there are many tangled loops and the two shortest ones include, respectively, 28 and 47 steps. The selfRs of the shortest loop ($I_1, \ldots, I_j, \ldots, I_m, \ldots, I_1$) are successively: GC, GGC, GTGC, GTGCGA, GCGATGCCGA, GCGACGTGCGA, GCGCGACGTGCGA, GCGCGACGGCGTGCGA, GCGCGACGGCGTCGAGCGGGCGA, GATGCGGGCGACGGCGTCGAGCGGGCGA, GCACGTGAA, TCAGGAA, TCGCGAA, TCGCGCGAA, TCGCGCGAACGG, GTCGGCGCGAACGG, CTCGGAGGCGCGAACGG, CTCGGATCGGCGCGAACGG, CTCCGTGATCGGCGCGAACGG, CCATTCCGTGATCGGCGCCGAACGC, GTCGGC, GTGACGGC, GTGAGCACGGC, GTGAG, GTGGACAG, GTGACCAAG, GTGACCACAAG, GCATGACCACAAG, and again GC. Note that each generation can contain many strands, but at least one of them (the I_i) must belong to the cycle. The secondary strands (I_i') are not represented on this figure; (b) Eigen hypercycle in which complementarity is equivalent to a "one loop per step" situation.

In our tanglecycles, there are two different classes of entities. These are nucleic acids and proteins but more generally they could be called informants and transformers. The informants code for the transformers that in turn transform the informants. The system will be self-reproducing if it can be represented by diagram (b) of Fig. 2. In the inner loop, I_1 reappears after m steps and in the outer loop after n steps. Two copies of I_1 will arise at the same generation after $m + n$ steps (self-replication). Figure 1(b) represents the Eigen hypercycle (b) as a particular case of a tanglecycle (a). The hypercycle can be considered as

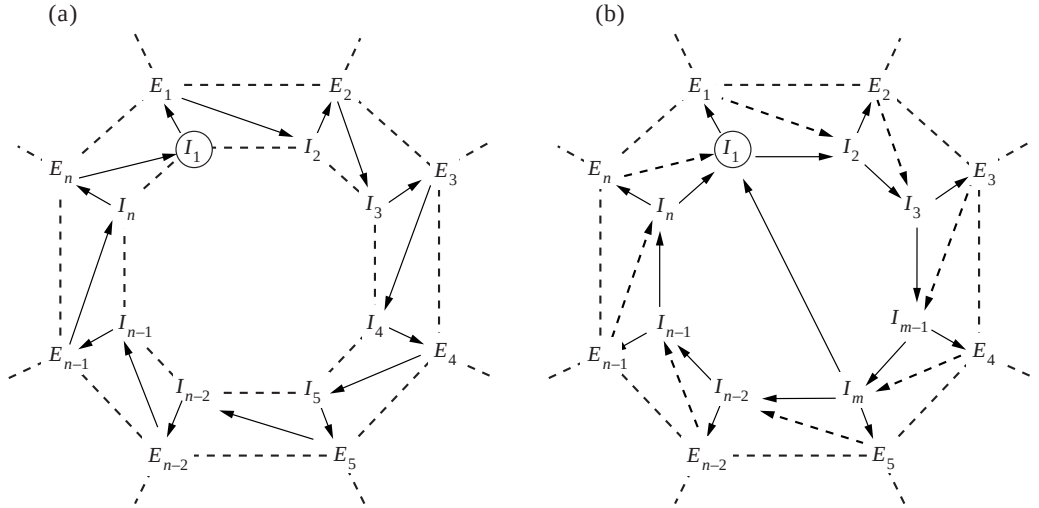


FIG. 2. Eigen (a) and typogenetics (b) hypercycles. I_i represents information carriers, i.e. strands of nucleic acids. E_i (encoded by I_i) represents catalytic function. In the Eigen hypercycle, each I_i is conserved after translation into I_{i+1}. With typogenetics, each I_i is modified to I_{i+1} and then each I_i disappears and self-reproducing requires that two I_i of the cycle (I_m and I_n in the figure) must be transformed into I_1.

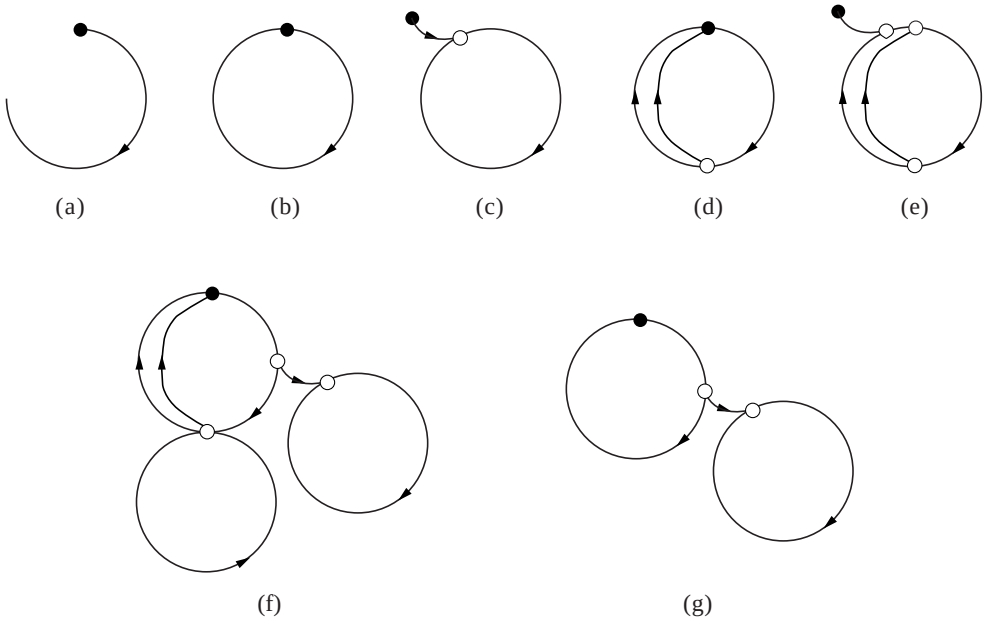


FIG. 3. In typogenetics, a starting strand (represented by ● on the figures) can be: (a) a short-lived strand, i.e. a strand that dies out after some generation [in this figure, the circle (or part of circle) represents the succeeding generations originating from the initial strand]; (b) a *selfM*, i.e. a strand that is part of a monocycle; (c) a *proselfM*, i.e. a strand that is transformed into a selfM; (d) a *selfR*, i.e. a strand that is part of a tanglecycle, and (e) a *proselfR*, i.e. a strand that is transformed into a selfR. Complex combinations of these possibilities can arise. For example, (f) represents a monocycle including two proselfRs and (g) represents a monocycle including a proselfM. The GC strand gives rise to very complex associations of such cycles (including a tanglecycle). The population generated by a selfR or a proselfR is exponential. Examples are GC and TCGAG (selfRs). The population generated by a selfM or a proselfM gives a steady-state. Examples are the GAGA strand (a proselfM) and the ATGAG strand (a selfM above-mentioned). For coupled monocycles, [an example is shown in (g)], a linear growing can be obtained. For example, the GTGC monocycle (121 steps) computed with a modified program instructed to eliminate the dimers and trimers (suppressing the appearance of the GC tanglecycle) gives rise to a biphasic growth from which the second phase is linear and can be fitted with the equation ($N = 0.132\ Ge + 11.0$), indicating that the monocycle in which GTGC is a selfM is coupled with one or some other monocycles. A short-lived strand gives a population that reaches a maximum before falling to zero.

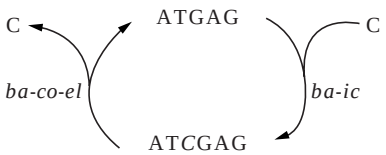
a tanglecycle including as many inner loops as I_i informants in the cycle, so that at each step the following loop appears:

$$I_i \rightarrow I_i + I_{i+1}$$

With this system, two copies of I_1 arise after n steps.

1.3. SELF-REPRODUCING, SELF-MAINTAINING AND SHORT-LIVED STRANDS

In addition to self-reproducing strands like GC, self-maintaining strands (*selfM*) were also found. Whereas a selfR strand belongs to a tanglecycle, a selfM belongs to a monocycle. The following monocycle is a very simple one



but other ones can be very large. For example the GTGC strand belongs to a monocycle in which it reappears every 121 generations.

Finally, many strands die out after some generations, they are short-lived strands. The different possibilities are represented in Fig. 3.

In our preceding paper we stressed that self-replication in typogenetics was an emerging property, in that it is nowhere explicitly coded. Self-replication is therefore a property of typogenetics but it should occur in any system based on the same principles.It is different from hypercycles simulated by cellular automata (Boerlijst & Hogeweg, 1991), in which self-replication exists as a transformation rule of the automaton, or from self-replicating systems simulated by equations (Schnabl *et al*., 1991; Chacon & Nuño, 1995) in which self-replication is also explicitly included in the equation.

In this paper we studied the tanglecycle features. After a research of other tanglecycles in addition to GC, we have investigated their internal structures.

Since typogenetics tanglecycles intrinsically exhibit the essential properties of life, they were submitted to classical experiments usually carried out on living world populations:

- restriction of available resources;
- population dynamics studies;

- competition between identical or different "species";
- behaviour of initial random population.

Finally a comparison was made with other works in this area: cellular automata, hypercycles and autocatalytic reaction networks.

2. Experimental Study of Tanglecycles

2.1. SEARCHING FOR OTHER TANGLECYCLES

Among the 16 possible doublets consisting of the bases A, C, G and T, GC is the only selfR. The other doublets die out rapidly. The shortest member of a given tanglecycle will be called a *primary selfR* (selfR I). For example, GC is a primary selfR whereas GGC (the GC's daughter) is a selfR but not a primary one.

No primary selfR was found among the 64 triplets and the 256 quadruplets. The GGC, GGA, GTGC and GTGA strands are selfRs but belong to the GC tanglecycle. Among the 1024 quintuplets we discovered another selfR I, the TCGAG strand. This tanglecycle was found to have three tangled loops with 126, 132 and 252 selfRs. The growth is also exponential and biphasic but the growth factor for TCGAG is almost 4 times lower than for GC (Fig. 4).

We did not continue the tedious systematic study of the 4096 sextuplets, 16384 septuplets, etc., because it would be necessary to try the strands one by one (what was made for the 1360 doublets, triplets and quintuplets). It must be noted that entering one initial strand in the computer, computing the different generations, storing them on a hard disk and finally running the program that searches the presence of an

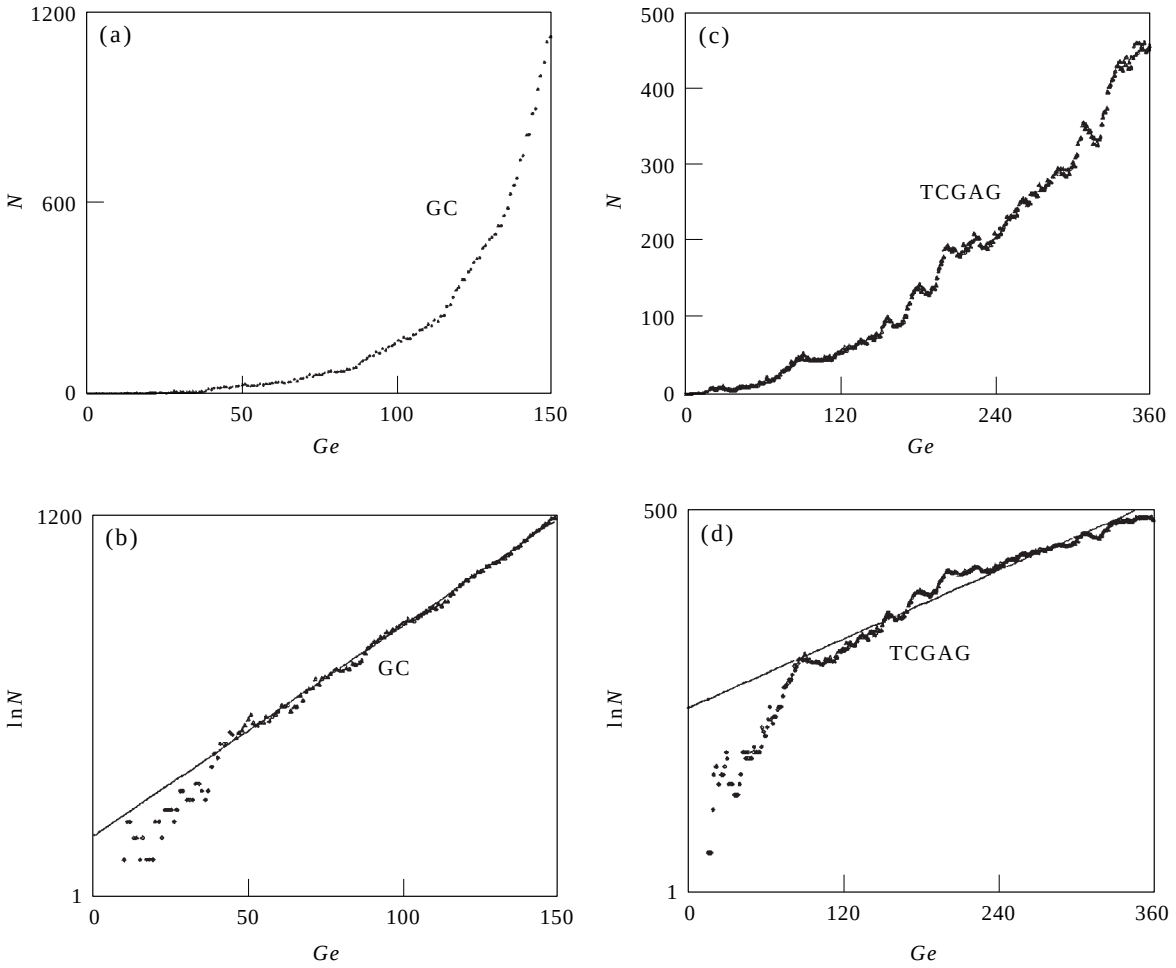


FIG. 4. Exponential growth for GC and TCGAG. The logarithmic plot ln N (strands) vs. Ge (generations) [curves (b) and (d)] shows the biphasic aspect. The growth factor f_g (second phase) amounts to 0.0387 G^{-1} for GC and to 0.0092 G^{-1} for TCGAG. The population doubles every 17.9 and 75.3, generations respectively.

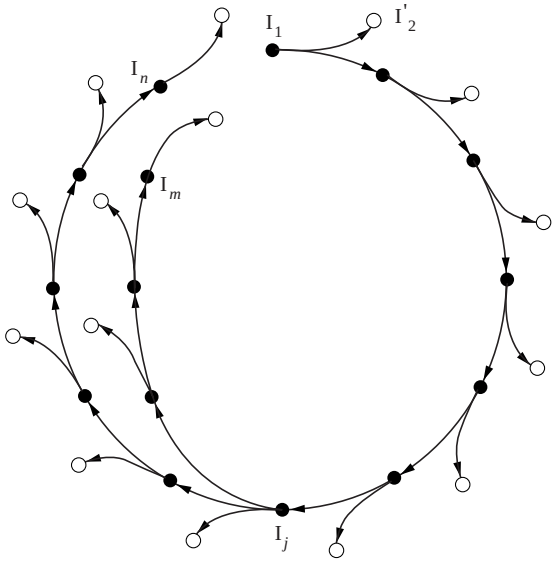


FIG. 5. Interrupted tanglecycle. The empty circles ○ represent the covering strands I_i', i.e. the strands that are not selfRs. These strands evolve among the generations (but not self-replicate).

initial strand duplicate in any generation can take up to several hours.

So, among the 1360 tested strands, two were primary selfRs. How many are they among the 2×10^{48} possible strands limited to 80 bases? The probability to be a primary selfR should decrease with the length of the strand so that we cannot apply the ratio of 2/1360 to the strands as a whole (which would give 3×10^{45} selfR Is). The actual value is certainly much lower but the emergence of self-replication in a tanglecycle forbids a forecasting of the number of selfR Is. In consequence our work will concern the two tanglecycles GC and TCGAG.

2.2. STUDYING THE COMPOSITION OF THE GENERATIONS

In order to study the composition of the generations the computer was instructed to compute the tanglecycle but to remove each GC (or TCGAG) as soon as they appear, so that it computed the evolution of the covering strands I_i (Fig. 5). The difference as compared to the uninterrupted cycle begins at the generation 28 for GC and 126 for TCGAG (the generations at which the first primary selfR reappears). In the two cases, an oscillating steady state is observed (Fig. 6).

GC tanglecycle: the steady state is established at generation 297. There is an oscillation from 15 to 16 strands, indicating that only some little monocycles are still running. Consequently, the whole strands from generation 1 to 297 represent a collection (estimated to approximately 5000 strands), from which only some hundreds are necessary for the replication (the selfRs, i.e. black circles in Fig. 5). This collection is a limited set in that it is possible to distinguish it from the rest of the 2.10^{48} possible strands.

Therefore, the exponential growth comes from the multiplication of a "being" (a tanglecycle), separated from the rest of the world in that it includes a limited number of transformers and informants, to the exclusion of the other possible ones.

After reproduction, such beings reach a steady state that produces no more descent, the overall exponential growth coming from the multiplication of these overlapping strand populations.

TCGAG tanglecycle: the steady state appears at generation 430. There is an oscillation from 83 to 86 strands. The TCGAG "being" contains almost 35000 strands of which almost 500 are selfRs.

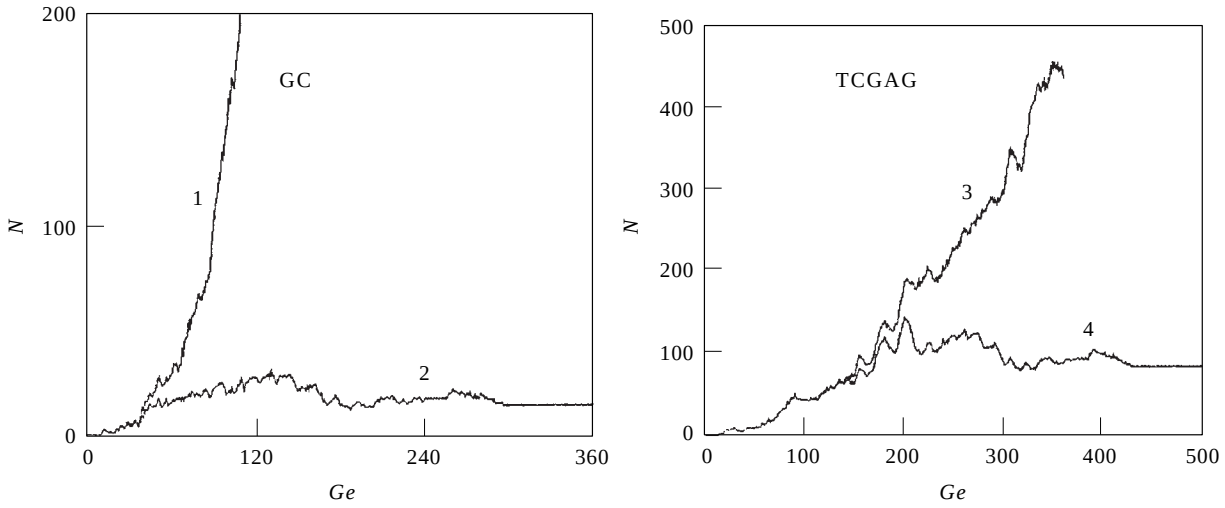


FIG. 6. Evolution of the population for the original (1 and 3) and interrupted (2 and 4) cycles.

One can summarize a tanglecycle fate by the three following phases:

- childhood, the first development phase, before self-reproduction (from generation 1 to 125 for TCGAG);
- maturity, in the course of which self-reproduction occurs (two copies of I_1 at the same generation) and the development of the covering strands continues (innovation production, from generations 127 to 251 for TCGAG);
- the elderly phase, with a steady state and no production of new strands.

3. Experiments on Tanglecycles

3.1. Limitation of the available bases

The growth of a tanglecycle is exponential if the amount of building blocks (the bases A, C, G and T as well as the 16 aminoacids) is not limited. We decided to study the population with a limited number of each of 999 bases. In practice we modified the program so that when it acts on a strand, its bases first return to a pool (999 bases of each kind at generation 0). After treatment, the new strand is inspected and the pool is decreased by the number of corresponding A, C, G and T. At this moment, if the pool contains at least a base of each kind, the program continues normally, otherwise the last strand is eliminated and its bases returned to the pool. It must be noted that the strand is not randomly eliminated but when it needs a base that is not present in the pool. The system is equivalent to a well-stirred flow reactor that can be initialized with any starting population.

3.1.1. *GC in the reactor*

Figure 7(a) shows the evolution of the population for 2000 generations if the reactor is initialized with a GC strand. In the beginning the growth is exponential but from the generation 117, the G base becomes a limiting substrate resulting in a suppression of some strands. Since this moment, the number of Gs in the pool is between 0 and about 250 and the total number of strands fluctuates strongly. At times the population sharply decreases (for example it falls to seven strands at generation 1362) and also increases sharply. The general behaviour becomes chaotic although a certain periodicity can be detected as will be shown later.

In order to measure how the composition of the generations evolves the following equation was found to calculate the homogeneity (Hg) of the population at a given generation:

$$Hg = \frac{1}{n}\sqrt{\sum_{i=1}^{e} S_i^2}$$

n: total number of strands in a given generation;

e: number of different strand species in this generation ($1 \leqslant e \leqslant n$);

s_i: number of identical strands of the i species in this generation.

The homogeneity is comprised between 0 (all the strands are different) and 1 (all the strands are the same).

Figure 7(b) shows the evolution of Hg. When the number of strands begins to sharply fluctuate an important increase of the homogeneity can be observed. From the generation 1000, Hg has an average value of 0.46, with maxima greater than 0.9 (90% of identical strands!). On the other hand, a 0.07 Hg value was calculated during the exponential phase. Thus a selection is made because of the limitation of the building blocks (increase of Hg by a factor $0.46/0.07 = 6.6$).

Some strands are eliminated in favour of other ones and the generations became much more homogeneous so that a great number of strands in the same generation will have the same fate. If this fate is to disappear (for example reaching the maximum length) then an abrupt decrease will be observed, but if on the contrary this fate is to breed, then a great number of strands will reproduce at the same generation. This behaviour is confirmed by the number of GC appearing at each generation [Figure 8(a)].

For almost the first 600 generations the graph shows an irregular appearance of the GC strands. Nevertheless a certain periodicity (with a period of 50 generations) becomes gradually clear although irregularity islets continue to appear, for example around the generation 2500.

For generations 1000–2000, there are almost 1.19 GC per generation but for certain generations this number can be larger than several tens. The irregular or chaotic nature of this population will be analysed in a following chapter.

3.1.2. *TCGAG in the reactor*

Figure 7(c) shows the population for 1000 generations if the reactor is initialized with a TCGAG strand. Unlike what was observed for GC, an oscillating steady state is reached. The TCGAG strand appears for the last time at generation 384,

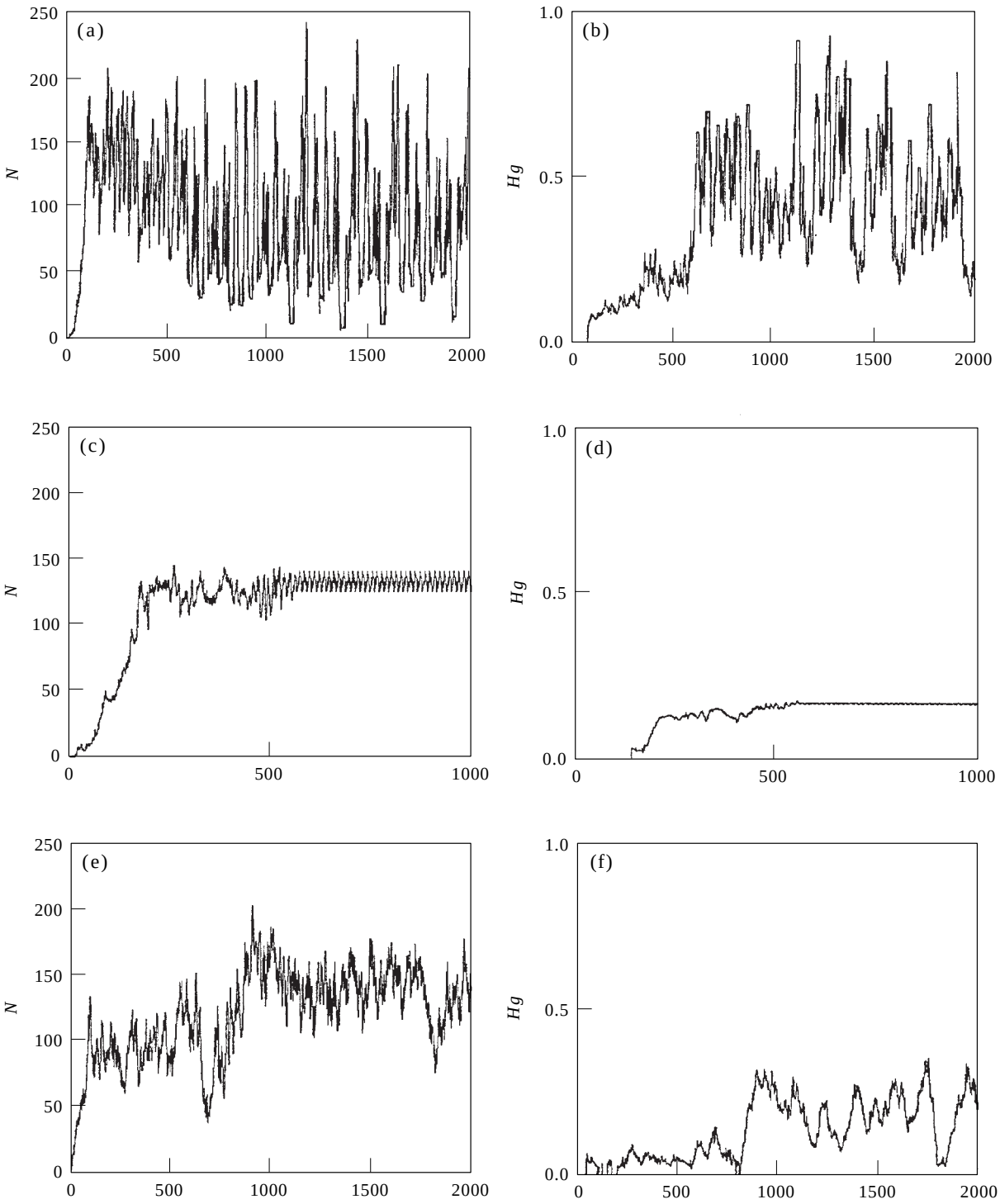


FIG. 7. Evolution of the population and homogeneity with a limited amount of 999 bases A, C, G and T. (a) GC evolution; (b) GC homogeneity; (c) TCGAG evolution; (d) TCGAG homogeneity; (e) random population evolution; (f) random population homogeneity.

which means that the limitation to 999 bases particularly affects the I_i selfRs of the TCGAG tanglecycle, leaving the sole I_i' secondary strands that give finally a group of selfM oscillating strands for which the homogeneity becomes stable at almost 0.16 [Fig. 7(d)], contrary to 0.09 for TCGAG in the

exponential phase (multiplication factor of 0.16/0.09 = 1.8).

We also made trials with a pool containing up to 7000 bases of each kind instead of 999 but the same phenomenon is always observed: the TCGAG strand ceases appearing and a steady state is reached.

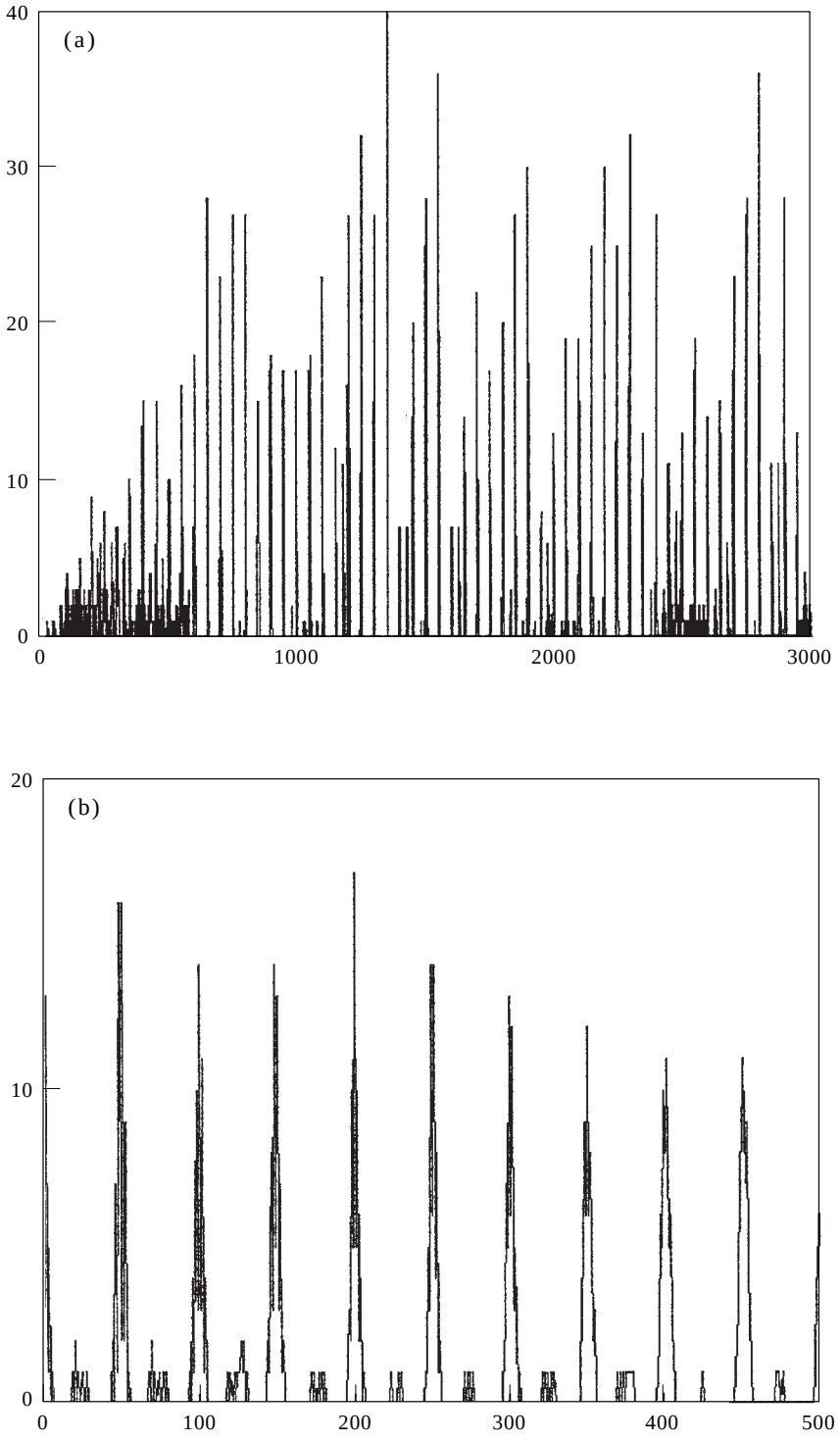


FIG. 8. (a) Number of GC from generation 1 to 3000; (b) autocorrelation for the number of GC from generations 1001 to 2000.

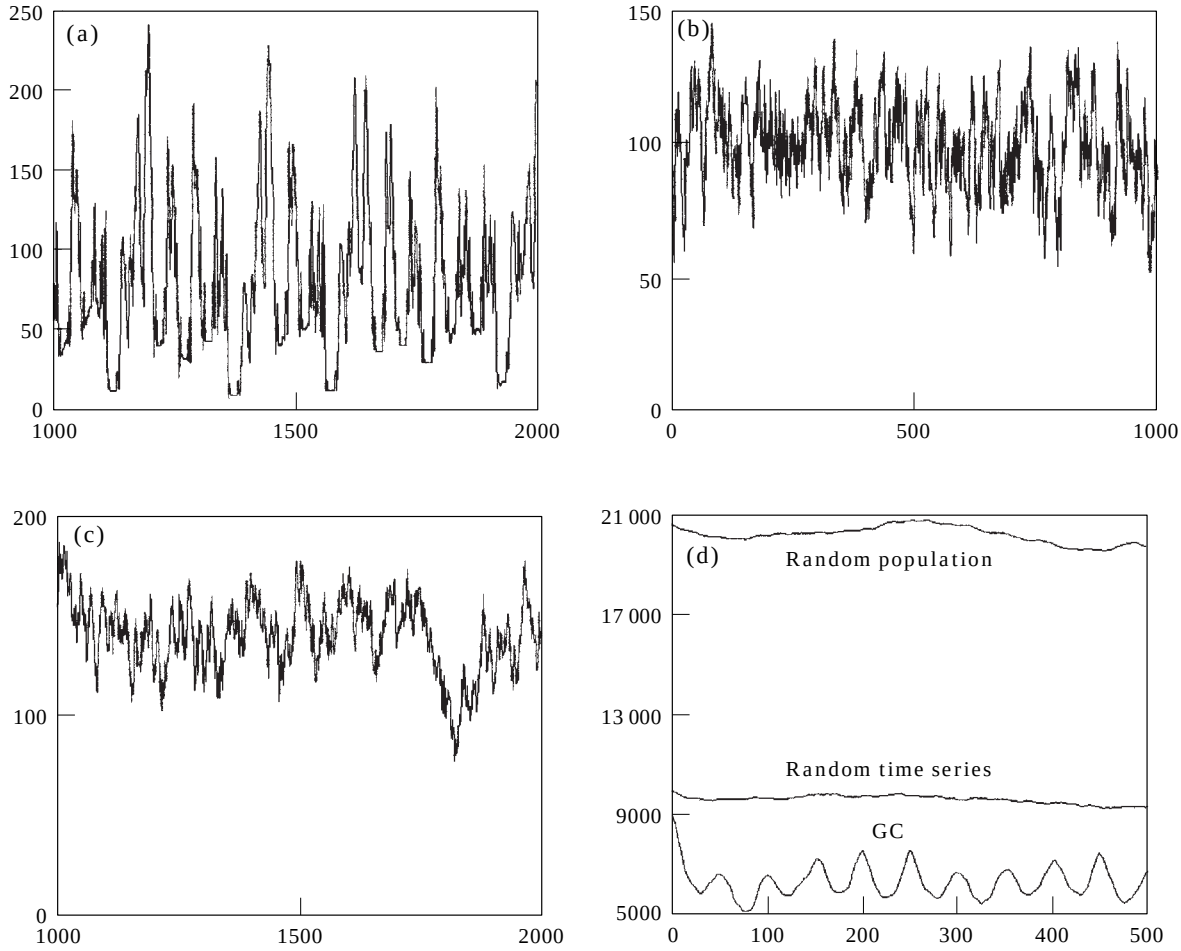


FIG. 9. (a) The GC tanglecycle from generation 1001 to 2000; (b) random time series; (c) random population; (d) autocorrelation functions.

3.1.3. *Random population without tanglecycles*

We also computed the evolution of a random population by introducing at each generation a strand of random length and composition. Nevertheless a length greater than six was chosen to avoid the appearance of a GC or TCGAG tanglecycle. This evolution [Fig. 7(e)] is supposed to mimic that of a world which would be a "soup" of strands in which no selfR would be present. One can observe that the changes are less abrupt than for GC and that the homogeneity stays low (average value of Hg = 0.18 for the generations 1001 to 2000).

3.2. TANGLECYCLE DYNAMICS STUDIES

The evolution of the typogenetics population with a limited pool can be considered as a discrete dynamic system. It is a deterministic system since all the rules are well established and known, but its evolution is unpredictable.

Figure 9(a) represents the limited GC evolution for the generations 1000–2000. In recent years such graphs have been encountered in many publications. These are time series concerning many domains: meteorologic data (Nicolis & Nicolis, 1984, 1987; Grassberger, 1986; Essex *et al*., 1987; May, 1987; Tsonis & Elsner, 1988), epidemiologic data (Schaffer & Kot, 1985; Pool, 1989; Olsen & Schaffer, 1990), biological populations (Sugihara & May, 1990; Solé & Valls, 1992), quantum chaos (Gutzwiller, 1992), physiological systems (Vasilakos & Beuter, 1993), ecology (Solé & Bascompte, 1994), stock exchange disorders (Orléan, 1991), solar system evolution (Laskar & Froeschlé, 1991; Laskar, 1995), etc.

The recent chaos theory shows us that very simple and determinist systems can yield unpredictable behaviours.

The reverse reasoning has been made and it was asked if apparently chaotic and unpredictable

phenomena could not hide a structure governed by simple underlying laws. According to the theory of dynamical systems (see for example Mullin, 1993), the evolution of a system can be described by trajectories in the phase space in which the coordinates are defined by the variables needed to completely describe the evolution of the system. Each trajectory in the phase space represents the evolution of the system from some initial conditions. The system can exhibit an attractor, i.e. a local region of available space that attracts the trajectories. Attractors can be fixed points, limit cycles, toruses or non-topological regions characterized by a fractal dimension, i.e. a dimension that is not an integer. The latter are called "strange" attractors. An important property of these attractors is the divergence of initially nearby trajectories that prevents long-term predictability. The dimension of an attractor, whether strange or not, supply information on the number of variables describing the system. Therefore, the determination of the dimension of an attractor yields the minimum number of variables that should be included in a model designed to predict the evolution of the system.

For actual systems, a trajectory in the phase space can be produced using a single record of some observable variable of the system. These observables could be for example the observed temperature or pressure for a meteorological system, the number of infected individuals for an epidemiological system, the cardiac frequency in a cardiological system, etc. In our tanglecycle system we will use N, the number of strands at each generation, as an observable of the system.

But unlike the preceding physical or biological systems, all is known about the production of our time series. We created the system, established the laws and have a computer which supplies the energy necessary to produce the successive generations without random factor. Nevertheless, typogenetics exhibit a totally unpredictable behaviour, as unpredictable as the classical systems which can be found in the publications dealing with chaos, such as the excited pendulum, the logistic equation, etc. These systems are quite simple, they are determinist and yet chaotic. Typogenetics is not simple and is closely related to the natural systems like climate, biological populations, etc.

Considering that the evolution of the number of strands in a non-limited medium can be represented (Varetto, 1993) by a simple exponential law

$$N = e^{T_c \cdot Ge}$$

can the evolution in a limited medium (999 bases) be described by a relatively simple law that would be hidden in the apparent complexity? In other words does an attractor of small dimensions exist? Is this attractor strange? Is the GC tanglecycle chaotic when the food supply is restricted?

3.2.1. *Sensitivity to initial conditions*

Sensitivity to initial conditions is a major characteristic of chaos. An infinitesimal variation $dx(0)$ in the initial conditions will become $dx(t)$ at time t. Sensitivity to initial conditions pertains if $dx(t)$ increases exponentially with t: $|dx(t)| \sim |dx(0)|.e^{\lambda t}$ where λ (>0) is called the Liapounov exponent.

Starting again from GC with a number of 998 bases instead of 999 the evolution exhibits a difference between GC998 and GC999 from generation 107. This difference is increasingly growing so that (Fig. 10) the evolutions become completely different.

The Liapounov exponent was computed by measuring the variation of the difference between the two evolutions from generation 107. This variation is exponential and it was found that $\lambda = 0.023 \pm 0.002\ G^{-1}$, so that it can be estimated that the two evolutions diverge completely after almost 120 generations.

3.2.2. *GC*999 *as a time series*

In order to detect the presence of a n-dimensional chaotic attractor from a signal, a time delays method can be used (Eckmann & Ruelle, 1985). From a time signal $S(t)$ the trajectory of a point $X(t)$ of coordinates $X_1 = S(t)$, $X_2 = S(t+\tau)$, $X_3 = S(t+2\tau), \ldots, X_n = S(t+(n-1)\tau)$, where τ is a time delay, is plotted in a n-dimensional phase space. The obtained trajectory contains the dynamics of the system. If the phase space is more than 3-dimensional, it is not possible to directly study the geometric structure of the attractor and we must be satisfied with more global informations like the *correlation dimension*. The most prevalent algorithm for computing the correlation dimension from an experimental time series is that of Grassberger & Procaccia (1983).

Assume a hypersphere of radius r whose centre is on an object composed of discrete points. The number of points N_r in the hypersphere varies as r^n, where n is the dimension of the object. The correlation dimension is defined as

$$D_c = \lim_{r \to 0} \frac{\ln C_r}{\ln r}$$

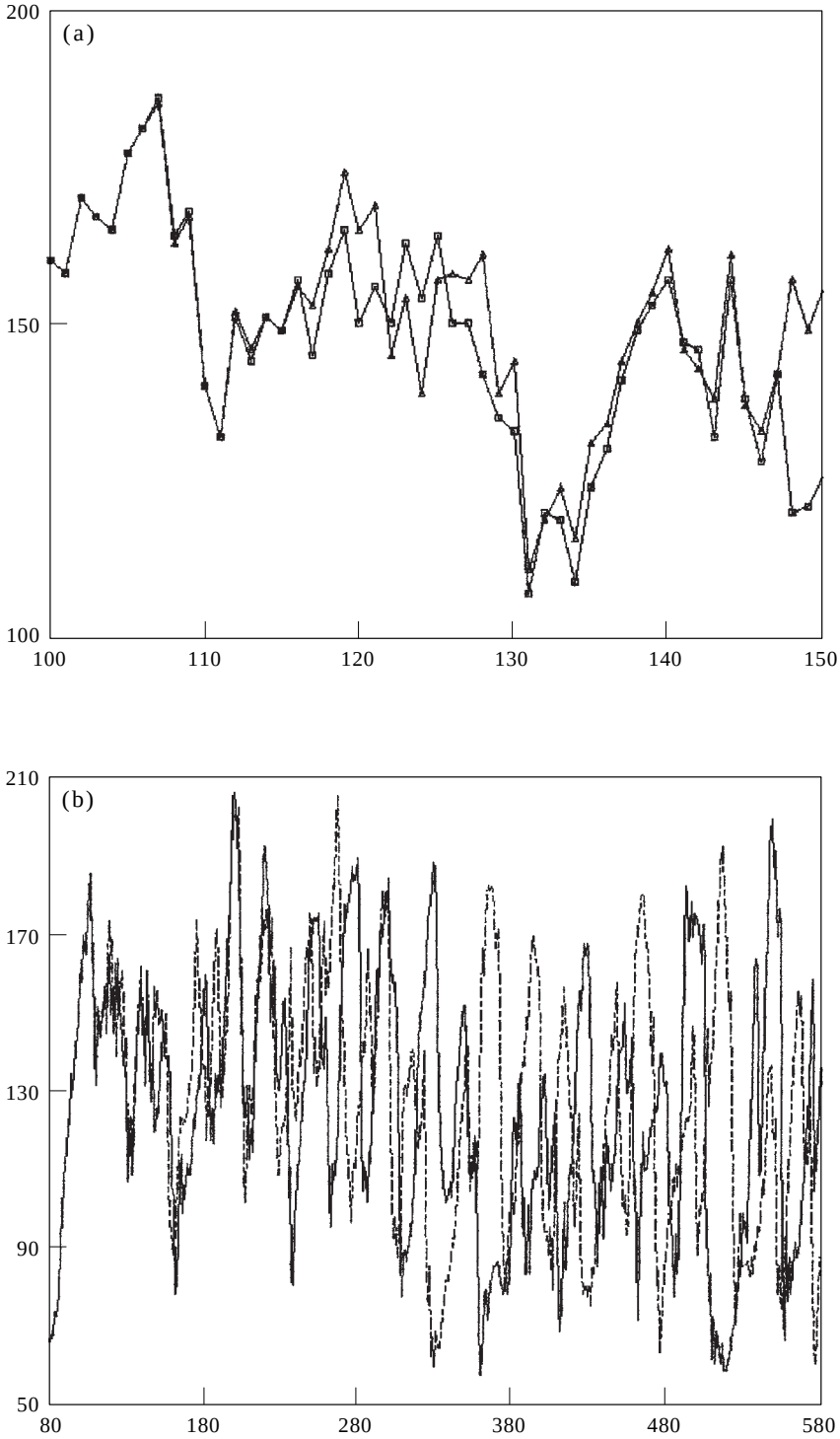


FIG. 10. Sensitivity to initial conditions. Differences between GC999 and GC998. (a) Generations 100 to 150; (b) generations 80 to 580.

where

$$C_r = \lim_{r\to\infty} \frac{1}{N^2}[\text{number of pairs } (x_i, x_j)$$

such as

$$|x_i - x_j| < r]$$

It could be shown that C_r is proportional to r^{D_c} where r, the radius of the hypersphere, is small compared to the size of the attractor. Assume a set of N points on an attractor in a n-dimension phase space:

$X_0(t_1), \ldots, X_0(t_N)$

$X_0(t_1 + \tau), \ldots, X_0(t_N + \tau)$

...

$X_0(t_1 + (n-1)\tau), \ldots, X_0(t_N + (n-1)\tau)$

A point X_i in the phase space will have the coordinates $\{X_0(t_i), \ldots, X_0(t_i + (n-1)\tau)\}$. A reference point X_i is selected in the data and all the distances $|X_i - X_j|$ from the $N-1$ other points are computed, which allows us to know the number of points located at less than a certain distance r from point X_i. Repeating the process for all the i values yields

$$C(r, n) = \frac{1}{N^2} \sum_{\substack{i,j=1 \\ i\neq j}}^{N} \Theta(r - |x_i - x_j|) \qquad (1)$$

where Θ is the Heaviside function, $\Theta(x) = 0$ if $x < 0$, $\Theta(x) = 1$ if $x > 0$.

For relatively small r values the correlation dimension should vary as

$$C(r, n) = r^d \qquad (2)$$

The correlation dimension d is then given by the slope of log C_r vs. r. The linearity is then tested for increasing values of n. If the value of d becomes independent of n, reaching a saturation value d_s, then the system represented by the time series should possess an attractor with a dimension equal to d_s. If this dimension is not an integer, then the attractor is fractal and the dynamics chaotic. Otherwise, if d increases indefinitely with n one can conclude that the time series is just random.

Some authors (Grassberger, 1986; Tsonis & Elsner, 1988) recommended to include in the sum of eqn (1) only the pairs of points (i, j) separated by a time period greater than the correlation time. Consequently, the following autocorrelation function was computed [Fig. 9(d)]

$$R(n, \tau) = \frac{1}{N-1} \sum_{i=0}^{N-1} x_i x_{i+n\tau} \qquad (3)$$

In order to test the programs a random time series of 1000 points $\{y; i = 1, \ldots 1000\}$ was also constructed [Fig. 9(b)] such as $y_i = (5y_{i-1} + 200r_i)/6$ where r_i is uniformly in the [0, 1] interval (modified from Grassberger, 1986). The same procedure was also applied to the random population from generations 1001 to 2000 [Fig. 9(c)].

Figure 11(a), (b) and (c) shows ln $C(r, n)$ vs. r for $\tau = 8$ generations. As a poor choice of τ can produce a bad convergence in the estimation of the correlation dimension (Wu, 1995), the values of $\tau = 4$, 16 and 32 were also tried but without significant modification. We retained the value of 8 since it appeared to give the best linear portion in the graphs. For each dimension n we determined the linear region such that the error on the linear regression of the slope did not exceed 1%.

As indicated in Fig. 11(d) the variation of the correlation dimension d with the embedding dimension n shows that for the tanglecycle GC the correlation dimension d is reaching a saturation value: $d_s \sim 3.7$. It is the case neither for the random time series nor for the random population for which d increases in a continuous manner. Some authors have shown that the method of Grassberger and Procaccia was not reliable if the number of points in the time series was small. In particular Ruelle (1990; Eckman & Ruelle, 1992) indicated that a dimension higher than 2 $\log_{10} N$ should not be estimated where N is the number of points in the sample. As we find a dimension of 3.7 with 1000 points we are well beyond the limit of 2 $\log_{10} N = 6$.

Therefore it can be concluded that the system represented by the GC tanglecycle with a limited pool of bases possesses an attractor. It is a 3.7-dimension strange attractor and the value of 3.7 suggests that the evolution could be represented by a dynamical system with four degrees of freedom (four differential equations).

3.3. COMPETITION BETWEEN TANGLECYCLES

3.3.1. *GC and TCGAG with a limited pool of bases*

Figure 12 depicts the situation when the tanglecycles GC and TCGAG compete in the reactor. We modified the program so that it was able to distinguish the strands originating from either the initial GC (upper curve) or the initial TCGAG in any generation.

The GC tanglecycle having a growth factor four times higher than the TCGAG one, it was expected that GC would win the competition. It is not the case, and although the two cycles influence each other, the strands coming from TCGAG do not completely

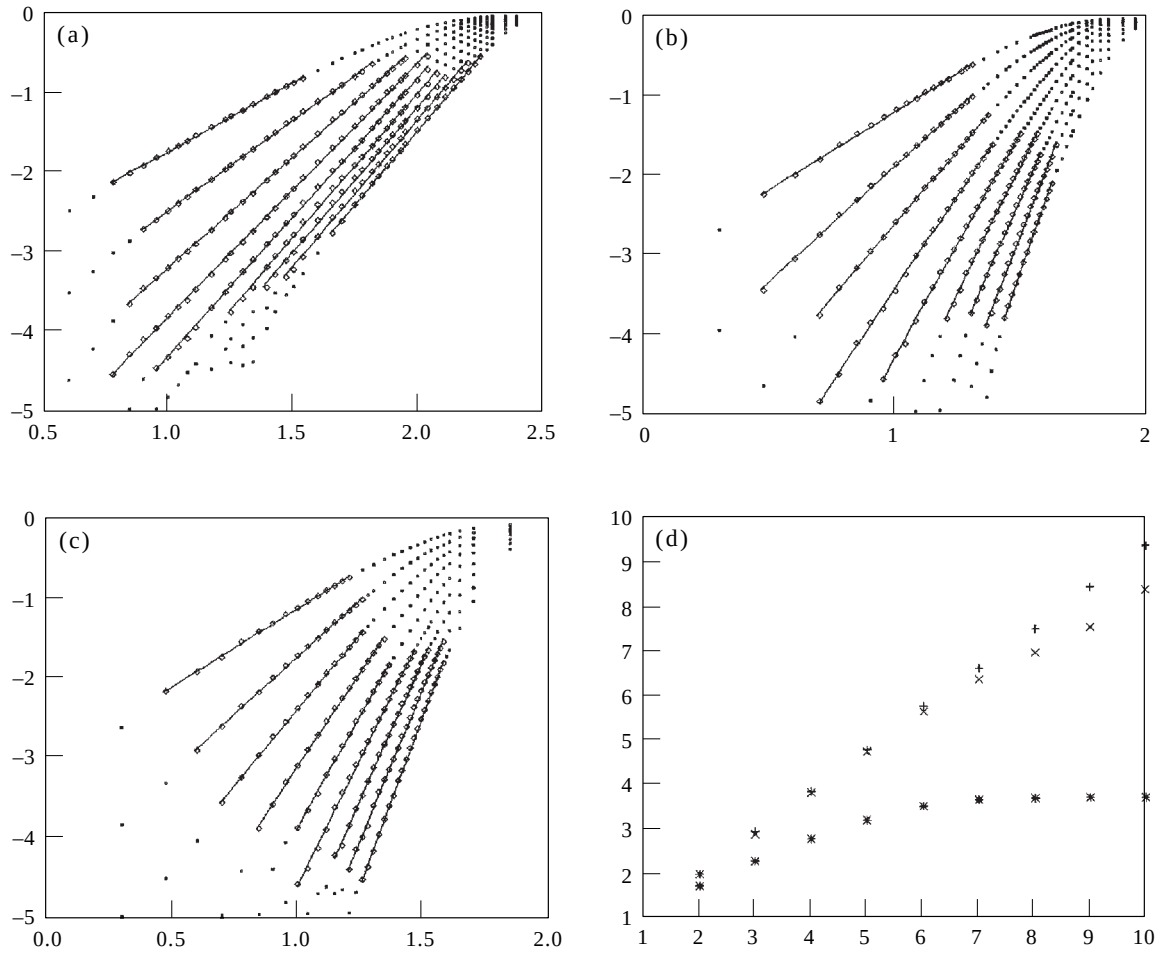


FIG. 11. Values of ln $C(r, n)$ vs. r for the GC tanglecycle (a), the random time series (b) and the random population (c); (d) shows the values of the slope d vs. dimension n [eqn $C(r, n) = r^d$]: (+) random time series, (x) random population, (*) GC tanglecycle.

disappear. In fact, for the GC tanglecycle the G base is a limiting substrate, whereas for the TCGAG cycle it is the T base. Nevertheless one cannot consider that the two tanglecycles of different "species" coexist independently. Figure 12 shows that the fluctuations of the population are even more drastic in the presence of the competing TCGAG species than in its absence. So the population originating from GC declines to a single strand at generation 600 (in fact, GC came very close to dying out!). This strand is CTCGGATCGGCGCGAACGG that is evidently a selfR since the exponential growth later resumes. Concerning TCGAG the situation is more stable, the number of strands stabilizing at approximately 20 with an homogeneity of zero. A steady state is recovered whenever the population originating from GC happens to decrease. So GC is chaotic, very sensitive and unpredictable whereas TCGAG is not chaotic, not very sensitive and stays in a quite stable situation.

In fact, it is not a true competition between two tanglecycles, since the TCGAG is not able to develop because of the limitation of the number of bases. The GC strands simply appear to be evolving in a more hostile medium.

3.3.2. *GC and GC with a limited pool of bases*

Figure 13 shows the competition between two identical GCs, named GC1 and GC2. As soon as the bases happen to be exhausted the fate of the two strands begins to differ. How can two identical strands processed by the same and deterministic way have a different fate? In fact the strands are successively computed by the program, therefore the limitation of the number of bases can affect a

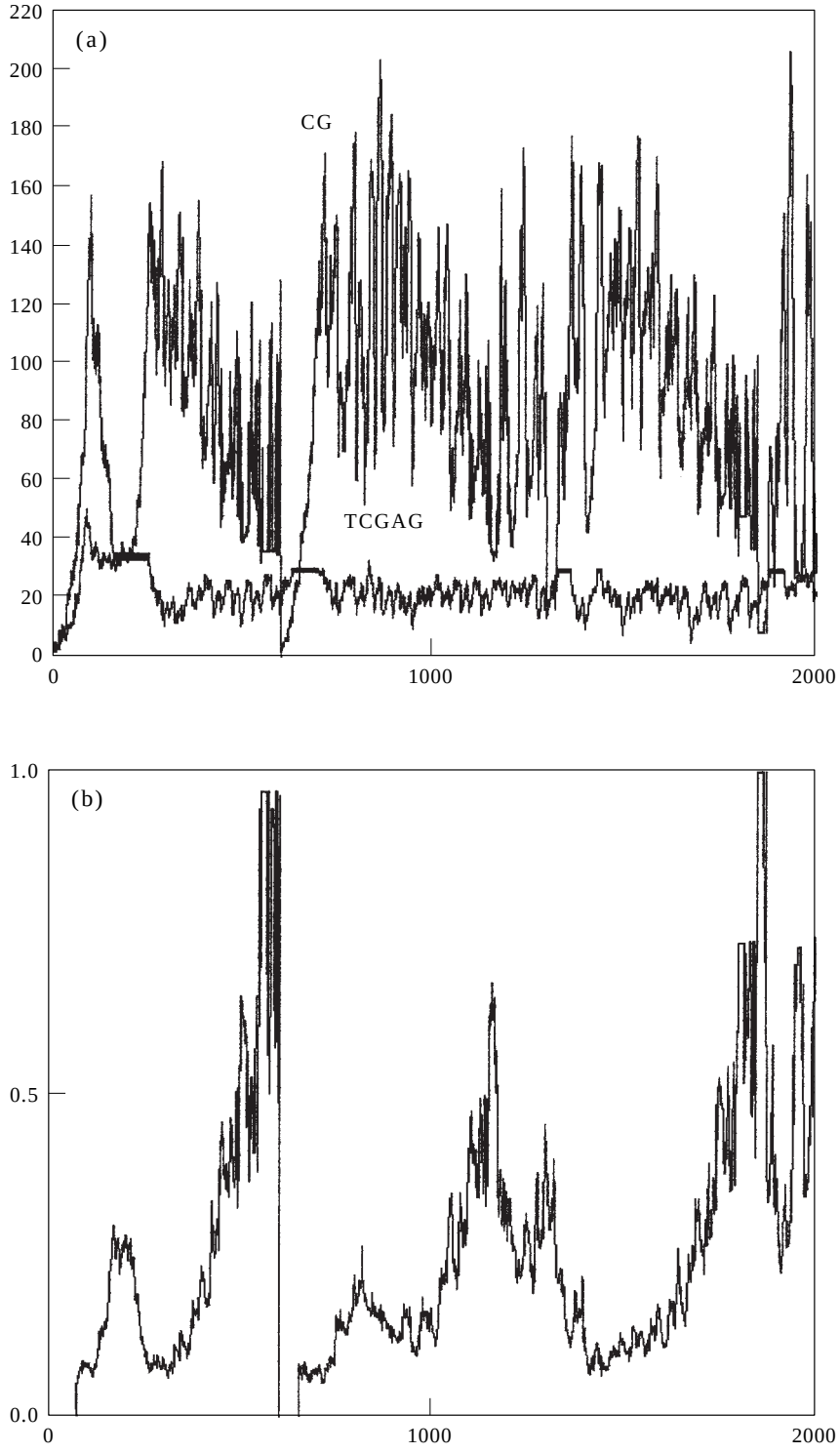


FIG. 12. (a) Evolution of GC and TCGAG with a limited amount of 999 A, C, G and T; (b) homogeneity of the GC population in the same conditions.

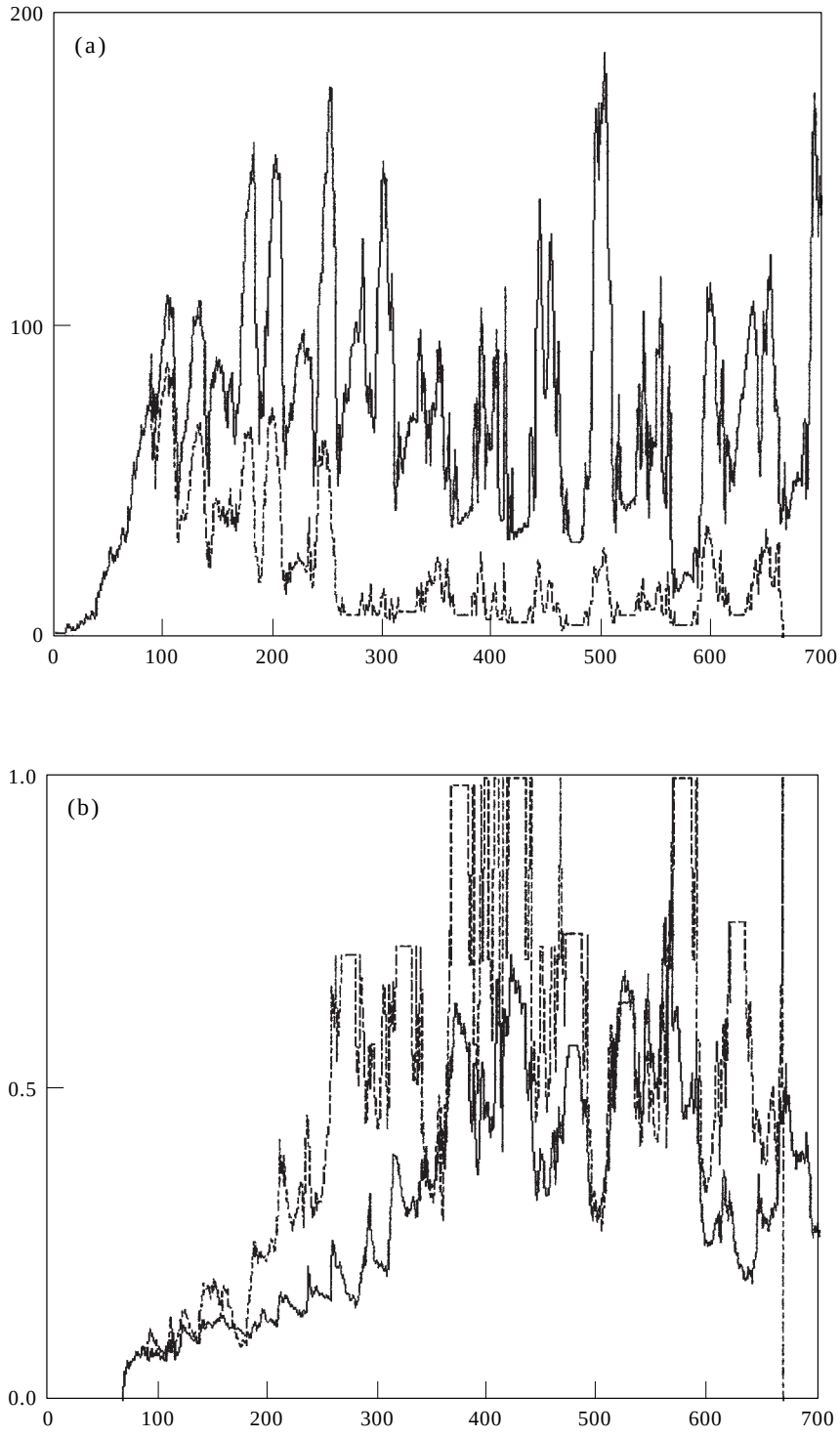


FIG. 13. Competition among two GCs: (a) number of strands; (b) homogeneity.

particular strand and not the previous one. In our case, the affected strand belongs to the second GC tanglecycle but the situation could have been reversed. At generation 88 we have 88 GC1 strands and only 74 GC2 strands. This difference further amplifies until the GC2 strands die out (would one dare to compare it with the victory of the L-aminoacids on the D-aminoacids in nature?). The

relative decrease of the number of GC2 strands is accompanied by an increase of the homogeneity of the GC2 strands. It should be noted again that an increase of the pressure applied to the system gives a higher instability. So GC has not been able to "beat" TCGAG which was however a poorer selfR (we have seen that it was because TCGAG was not a true tanglecycle in these conditions) but GC has been able to "beat" its own twin.

These results agree with the theory according to which any competition among (true) hypercycles leads to the survival of only one of them ("Once-and-forever" selection, Eigen & Schuster, 1979, 1982).

3.3.3. *Reactor initialized with a random population*

In these experiments (four trials), the reactor was initialized with a strand population of random length (from 2 to 80 bases) until consumption of the free A, C, G and T monomers and computed up to 4000 generations.

In all cases the appearance of GC strands was observed after some generations and the GC tanglecycles progressively invaded the reactor. The invasion was not necessarily total, some other (mono)cycles could continue running. A partial synchronization takes place. The 50 generations period seems to be a characteristic of the GC tanglecycle in this reactor but in general the synchronization is less pronounced than for the experiment in which the reactor was initialized with a unique GC. The homogeneity averages 0.2 but large peaks of 0.5–0.8 arise. In addition to these general properties, each individual run can give rise to specific results.

In the first of the four trials the entire population finally dies out! At generation 1115 the homogeneity becomes equal to 1 and all strands become too long at generation 1116, with total dying out. One can conclude that all the "young" tanglecycles (i.e. the tanglecycles in their childhood and maturity phase) have been eliminated at the same time by the selection. Only the tanglecycles in the elderly phase were conserved, with a subsequent inability to breed and a final dying out. One could say that we witnessed the suicide of the species. Another phenomenon was observed in our third trial: TCGAG appeared in addition to GC strands. The first GC was produced at generation 2 and the first TCGAG at generation 19. After that, a regular appearance of TCGAG strands was observed. It has been shown that this phenomenon was caused by the fact that new TCGAG strands are continuously produced by a four membered monocycle. This monocycle gives two TCGAG strands per revolution, as in Fig. 3(f).

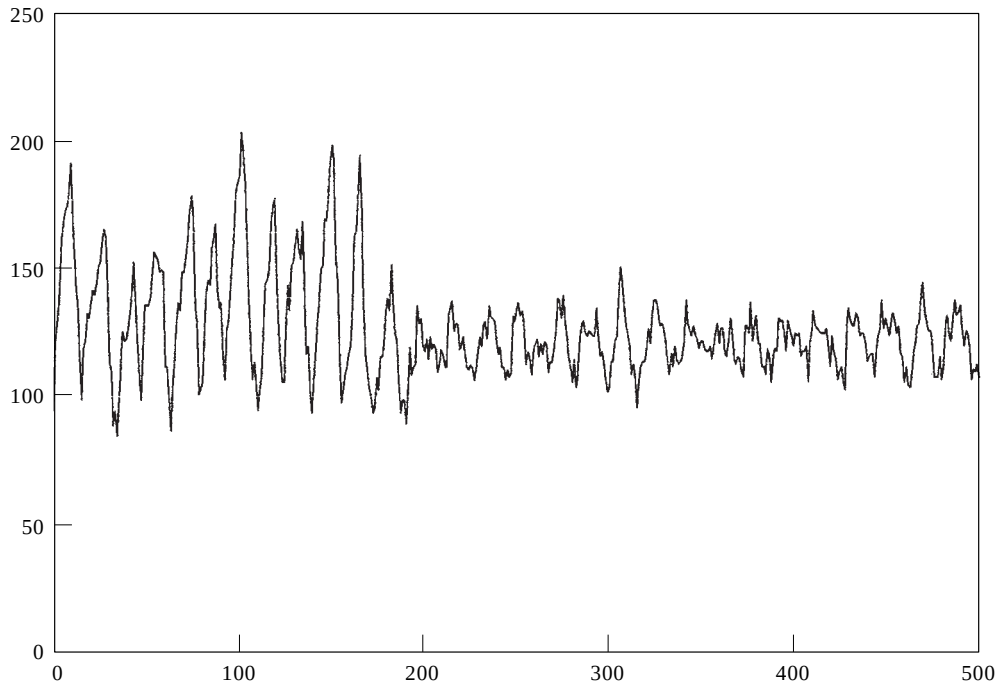


FIG. 14. The reactor was initialized with a random population: transition between a first phase during which the GC tanglecycles prevail and a second one during which the TCGAG tanglecycles prevail.

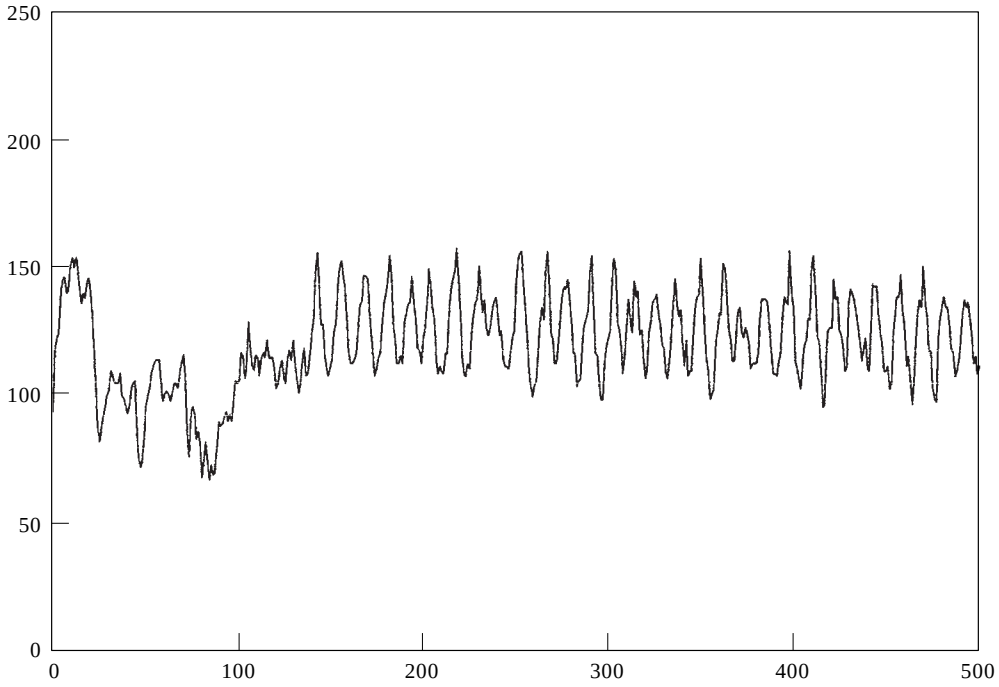


FIG. 15. The reactor was initialized with the same population as in Fig. 14 but here the GC tanglecycles were prohibited. A chaotic evolution takes place with TCGAG tanglecycles coupled with monocycles producing TCGAG strands.

In the beginning of the evolution, the GC tanglecycles are predominant and the limiting base is G, but progressively the T base becomes the limiting monomer and the TCGAG overwhelm the GC tanglecycles but without eliminating them. The transition is obvious on Fig. 14. During this second phase the homogeneity averages 0.2.

These results do not seem in agreement with the "once-and-forever" selection theory, GC and TCGAG being able to live together, and if it is remembered that the TCGAG tanglecycle was not viable when it was alone in the reactor, it could be hypothesized that the two tanglecycles can really cooperate and that the presence of GC allows TCGAG to survive. In fact, it will be shown in the following experiment that TCGAG can survive even without to the help of the GC tanglecycle, owing to the four membered monocycle.

3.3.4. *Reactor initialized with a random population but with prohibition of the GC tanglecycles*

The same four initial populations were computed again but the program was instructed to eliminate the strands shorter than 4, so that no GC tanglecycles could arise in these conditions. In all cases a steady state was reached, except for that for which TCGAG strands were observed (trial 3). In trials 1, 2 and 4, the (oscillating) steady state indicates that the reactor contains only self-maintaining monocycles, as expected.

In trial 3, there is no steady state and the evolution of the population is chaotic (Fig. 15). The reason is the presence of the TCGAG tanglecycles. These are not eliminated, in contrast to the experiment in which the reactor was initialized with a unique TCGAG strand. As stated above, the TCGAG tanglecycles manage to survive thanks to a monocycle acting as a TCGAG strands producer.

It can be concluded that very complex organizations can exist in which tanglecycles can be maintained by coupled monocycles. Thus the situation is complicated by these coupled cycles and it could just be hypothesized that the "once-and-forever" selection is confirmed by the experiments if the reactor contains only tanglecycles.

4. Discussion

4.1. TANGLECYCLES AND CELLULAR AUTOMATA

In our preceding paper we proposed that typogenetics was a molecular automaton that could represent a model for an artificial prebiotic universe in the same way that cellular automata could represent a model for an artificial biological universe. A classification of the cellular automata was made by

Wolfram (1984). He has identified the following four qualitative classes of cellular automaton behaviour:

class 1 evolves to a homogeneous state;
class 2 evolves to simple separated periodic structures;
class 3 yields chaotic aperiodic patterns;
class 4 yields complex patterns of localized structures, including propagating structures.

Moreover, a parallelism has been made between the cellular automata classes and the different kinds of attractors observed in physical systems. A class 1 automaton is similar to a continuous system with the simplest attractor: a limit point invariably leading to the same final state. The evolution of a class 2 automaton is rather like a system with a limit cycle, a set of configurations repeating themselves indefinitely. The class 3 automata, with their chaotic patterns can be associated with strange attractors. The class 4 automata should have no analogues in continuous dynamic systems, but Langton (1986) suggested that class 4 could correspond to systems on the onset of the chaos, namely located between class 2 and class 3. According to its behaviour, it appears that the TCGAG tanglecycle under limited conditions is analogous to a class 2 automaton, with a set of oscillating strands repeating themselves indefinitely. Any starting strand giving a monocycle or set of monocycles is to be classified in the same category. The GC tanglecycle which was shown to exhibit a life-like behaviour is similar to a class 4 automaton, with a periodic structure interrupted by chaotic patterns. This is in agreement with the idea that life seems to be at the border between order and chaos (McIntosh, 1990; Langton, 1990; Kauffman & Johnsen, 1991; Ito & Gunji, 1992). It reinforces the idea that tanglecycles could be used as a model for a universe between life and non-life, as a prebiotic universe is expected to behave.

4.2. Tanglecycles and hypercycles

4.2.1. *Similarities and differences*

We said at the outset that a hypercycle can be considered as a tanglecycle including as many inner loops as I_i informants in the cycle. An important characteristic that is shared by these cycles is the existence of two kinds of molecules. The transformations of the strands are catalysed by enzymes. However, an important difference is that the Eigen model is centered on template directed replication of complementary + and − RNA strands. Therefore complementarity is a necessary condition for self-reproduction. On the other hand, with typogenetics, complementary copying is not essential and represents just a silico-chemical property, like the other operations of moving, cutting, lengthening or shortening. For example, in the GC tanglecycle, the GC strand reappears for the first time after a cutting operation of the GCATGACCACAAG strand as follows (and not by a copying operation):

$$\mathrm{GCATGACCACAAG} + G + C \xrightarrow{ig\text{–}ba\text{–}ic\text{–}dg\text{–}cp} \mathrm{GC} + \mathrm{AT}CG\mathrm{GACCACAAG}$$

Typogenetics is not a system based on self-replication through templating, it is a system in which objects repeatedly construct specific new objects. Thanks to this construction, a given strand can be either short-lived, self-maintaining or self-reproducing. Thus, strands like GC and TCGAG are special not because of any programmed specific rules of the automaton but because they give rise to special self-reproducing tangled cycles.

Another difference between hypercycles and tanglecycles is that in a hypercycle, the different I_i are conserved after each step, whereas in a tanglecycle, the different I_i are converted after each step. The different strands of a given tanglecycle do not exist simultaneously (the GC strand exists at time t_1, the GGC strand at time t_2, etc. . . .), in contrast to the different strands of a hypercycle, which exist at the same place and at the same time. In fact, a typogenetics cycle can be compared to a *metabolism*.

4.2.2. *Tanglecycles as metabolisms*

The well known citric acid cycle (Krebs cycle) is a series of 10 reactions that result in the oxidation of acetyl residues to CO_2, liberating hydrogen equivalent. In the same way, a monocycle or a tanglecycle uses monomers (bases) or strands and produces other bases or strands.

More, some constituents can be removed from the Krebs cycle (gluconeogenesis, transamination) and others can be added to the cycle from the outside (transamination). For example, oxaloacetate takes part in gluconeogenesis when it is converted to phosphoenolpyruvate that in turn can yield pyruvate (end product of glycolysis). Pyruvate can also be transformed into oxaloacetate under the action of pyruvate carboxylase. These reactions are equivalent to two connected cycles. The situation is the same as

for the GTGC strand that was shown to act as a selfR in the GC tanglecycle and as a selfM in a big 121-member monocycle. In a cell, there is indeed a great number of "individual Krebs cycles" that are running at the same time so that the simultaneous existence of each of the 10 intermediates is not doubtful. In the same way, in the set of strands present in the reactor at a given generation, there are strands that belong to many different individual tanglecycles. These overlapping generations give the tanglecycles a local permanence (in the same place) and a temporal permanence (at the same time).

4.3. TANGLECYCLES AND AUTOCATALYTIC NETWORKS

4.3.1. *Tanglecycles as autocatalytic reaction networks*

An autocatalytic reaction network (ARN) contains molecules and reactions between molecules, with each reaction catalysed by some molecule in the network (Bagley *et al*., 1989). These networks, introduced by Stuart Kauffman, were first modeled in terms of random graphs (Kauffman, 1986). The random graph model was developed into a kinetic model (Farmer *et al*., 1986) that was simulated by computer (Bagley *et al*., 1989; Bagley & Farmer, 1991) in order to provide a plausible mechanism for the emergence of a polymer network, as well as to explore the collective behavior of the network.

Other models were also developed. Fontana & Buss (1994a,b) used an abstract chemistry based on λ-calculus, and Banzhaf (1994) on a system of binary numbers. In these systems, there is no clear separation of the molecular species into substrates, products and enzymes, a given species can play all three roles in different reactions. The network is called autocatalytic because every reaction between polymers is catalysed by a polymer in the network.

On the contrary with typogenetics there are two categories of molecules. Informants (typonucleic acids) are converted into other informants with the help of transformers (typoenzymes). The informants never become transformers and vice versa. However, each transformer is obtained from an informant, according to rules encoded in the latter. For example the first reaction of the GC tanglecycle

$$\mathrm{GC} + \mathrm{G} \xrightarrow{ig} \mathrm{GGC}$$

can be considered as the transformation of a GC substrate into a GGC product catalysed by the GC substrate itself (through the medium of a catalyst *ig* synthesized by GC). Therefore, it can be said that the GC $\rightarrow$ GGC reaction is catalysed by a molecule of the system, namely GC, and that the sequence of the succeeding reactions is an autocatalytic reaction network provided that GC is itself synthesized by another substrate of the system. This is the case if GC is part of a cycle. Any monocycle or tanglecycle or complex combinations of them can be considered as autocatalytic reaction networks.

So, among the possible strands, some give rise to self-reproducing ARNs (GC, TCGAG and the other I_i of these tanglecycles), others give rise to self-maintaining ARNs (GTGC, ATGAG and the other I_i of these cycles) and finally others give rise to sequences of reactions that cannot, strictly speaking, be called autocatalytic networks (the short-lived-strands).

4.3.2. *Tanglecycles as self-reproducing autocatalytic metabolisms*

An important result emphasized by the different authors dealing with these networks is that, under appropriate conditions, a catalytic reaction network can focus the material of its environment into a few chemical species. This feature is shared by our system. If our reactor is initialized with random strands, the homogeneity increases progressively and becomes stable at average values of 0.26, 0.15, 0.18 and 0.30 for the four trials, respectively. The reason is that tanglecycles are spreading out at the expense of the other strands. Like a metabolism, the tanglecycle "digests" the material of its environment, incorporating it into its own form. Bagley & Farmer (1991) called such a system *an autocatalytic metabolism*. Therefore, the tanglecycles are *self-reproducing autocatalytic metabolisms*.

4.3.3. *Tanglecycles and the λ-organizations*

An abstract chemistry implemented in a λ-calculus-based modeling platform was developed by Fontana & Buss (1994a,b). In this system, the objects combine with other objects to produce new objects which are transformed to achieve a stable form (called the *normal form* of the new object). In this model, a universe is specified by the axioms of λ-calculus, which define the nature of objects and the manner in which objects are transformed syntactically. The basic event is the interaction among two objects, A and B, upon collision:

$$\text{collision between } A \text{ and } B \rightarrow A + B + \text{normal form of } (A)B \quad (4)$$

In this interaction scheme, the assumption is made that the colliding objects are not consumed in the collision process.

The system is a well-stirred flow reactor that is initialized with 1000 randomly generated objects. A

pair of objects is chosen at random for collision according to the above interaction scheme. The number of objects is kept constant at any time by eliminating an object at random. The whole procedure is reiterated with the help of a computer.

In many instances the system reduces to just one object species that is a self-copier. In other cases the system contains a small stable ecology of objects engaged in mutual copy-actions, i.e. elementary hypercycles. The results have been summarized by Fontana & Buss (1994b) as follows:

(i) hypercycles of self-reproducing objects arise (level 0 organizations);
(ii) when replication is prohibited or inhibited, self-maintaining organizations of considerable complexity emerge (level 1 organizations);
(iii) organizations can be hierarchically combined to produce new self-maintaining organizations that contain the low-level organizations as self-maintaining components (level 2 organizations).

These three findings can be adapted to typogenetics:

(i) self-reproducing organizations of considerable complexity emerge (selfR organizations);
(ii) when tanglecycles are prohibited, self-maintaining organizations arise (selfM organizations);
(iii) there is no level 2 organization in typogenetics because of the absence of random rules.

Thus, a feature shared by the λ-system and typogenetics is the appearance of self-reproducing entities, but it must be noted that Fontana and Buss used the word "arise" and not the word "emerge" for their self-reproducing objects. This can be explained by the fact that the λ-system is very favourable to self-reproduction. If in reaction (4) the normal form of $(A)B$ is confined to A (or to B), the self-reprodution of A (or of B) is obtained:

$$A + B \rightarrow A + B + A \tag{5}$$

In fact, such a reaction is equivalent to each elementary step of a hypercycle, the complementary strings I^+/ I^- copying each other. The following reaction is equivalent to reaction (5) and represents the copying of the I^+ strand with conservation of the initial double strand

$$I^+ + I^- \rightarrow I^+ + I^- + I^- \tag{6}$$

So, the common property of the hypercycles and the λ-system is that the reactants are not consumed by the reaction and that they produce a new object at each step. Then, it is sufficient for this new object to be identical to one of the reactants involved in any of the preceding steps to give rise to self-reproduction. Thus, although the self-reproduction is not really explicitly coded in the λ-system, it could be predicted that self-copying objects will arise. As stated by Darley (1994), "emergent systems are those in which even perfect knowledge and understanding may give us no predictive information". We can now understand why it is difficult to say that (level 0) self-reproducing objects emerge in the λ-systems. Even if the probability for an object to be a self-replicator were very low, the appearance in the reactor of only one of them is sufficient to make it invade the reactor and eliminate the other organizations that just maintain without self-reproduction. For this reason, Fontana and Buss had to artificially prohibit these simple self-reproducing objects to make the study of the (level 1) self-maintaining organizations possible.

It was impossible to make predictions about the evolution of these level 1 organizations, they can be said to emerge. Consequently, for these organizations the optimal means of prediction was simulation.

With typogenetics, the self-reproducing organizations can be said to emerge, as well as the self-maintaining ones and moreover simulation was shown to be the only means of prediction. The λ-system level 2 organizations are not present in typogenetics. With the λ-system, when two different level 1 organizations are introduced in the same reactor, interaction between them can arise because of the random assignment of reactions. An object being part of a given organization can react with an object of the other organization, with a subsequent possibility for a new organization. With typogenetics, a given object (a strand) contains all the information needed to specify its chemical properties and thus this strand can belong to only one organization. For example, whereas the GTGC strand was seen to belong to a monocycle as well as to a tanglecycle, these two cycles cannot be called different organizations. If a GTGC strand is introduced in the reactor, it will generate the two cycles simultaneously and these two cycles represent the same organization.

4.4. THE EMERGING PROPERTIES OF TANGLECYCLES

Typogenetics, such as programmed here, is a quite complicated system from which the basic laws are defined in a computer program. Two $\times 10^{48}$ different informants (nucleic acids) can exist that code, with the help of 16 doublets, for almost 10^{94} possible transformers (typoenzymes). The basic laws are exactly defined, nothing being left to chance. In these conditions we have shown the existence of strands like GC, TCGAG, etc. . . . which grow exponentially by self-reproduction. This self-reproduction is an emergent phenomenon in that it is not explicitly coded and

even if we are the creator of the system and have a perfect knowledge and understanding of the elementary interactions, it gives us no predictive information about the exact evolution of the system.

We suggest that such an emergent self-reproduction property could be achieved by any *constructive dynamical system*, i.e. any finite set in which interactions among objects repeatedly construct specific new objects (Fontana & Buss, 1994b) provided the system uses *deterministic rules for assigning reactions* and evolves under *non-equilibrium-conditions*.

4.4.1. *Tanglecycle within an autocatalytic reaction network*

In Fig. 16(a) a tanglecycle is proposed that could emerge in a system for which we assume a two-letter alphabet consisting of *a* and *b*. In this tanglecycle, the polymers *abb*, *abbb*, *abbbb* and *bb* are not only selfRs but also catalysts. Here, the non-equilibrium condition is satisfied by assuming irreversible individual reactions but could also be satisfied by assuming reversible reactions in a reactor driven away from equilibrium by the flow of mass. No random assignment of reactions is supposed, each reaction being catalysed by a polymer of the network according to complementarity rules: it is supposed that the *abbbb* strand is the only one able to catalyse the transformation $abb + b \rightarrow abbb$, that the *bb* strand is the only one able to catalyse the reaction $abbb + b \rightarrow abbbb$, and so on.

If all these strands were present in the same reactor, a self-reproducing tanglecycle would arise that could be studied as we did for our typogenetics tanglecycles.

Typogenetics can indeed be considered as a constructive dynamical system in which assignment of irreversible reactions is realized by a deterministic rule: a given enzyme only acts on the strand it originates from. It must be noted that even if a reaction $A \rightarrow B$ is irreversible it does not mean that the *B* product cannot be transformed into *A*. For example the typogenetics reaction

$$\mathrm{ATGAG + C} \overset{ba\text{–}ic}{\rightarrow} \mathrm{ATCGAG}$$

is not the reverse reaction of

$$\mathrm{ATCGAG} \overset{ba\text{–}co\text{–}el}{\rightarrow} \mathrm{ATGAG + C}$$

in that the two reactions are not catalysed by the same enzyme. These two reactions must be considered as a monocycle, as we have seen before, instead of a reversible reaction. Such situations are not rare in the actual biochemistry, for example in glycolysis the transformation of fructose 6-phosphate into fructose 1,6-bisphosphate is catalysed by the enzyme phosphofructokinase and the transformation of fructose 1,6-bisphosphate into fructose 6-phosphate is catalysed by fructose 1,6-bisphosphatase.

Typogenetics could be modified into a more complicated system so that an enzyme produced by a strand could catalyse the transformation of any strand provided it possesses a sequence that would be complementary to the enzyme. It would be still deterministic, and we expect that self-reproducing autocatalytic metabolisms will emerge. A possible tanglecycle produced in these conditions is shown in

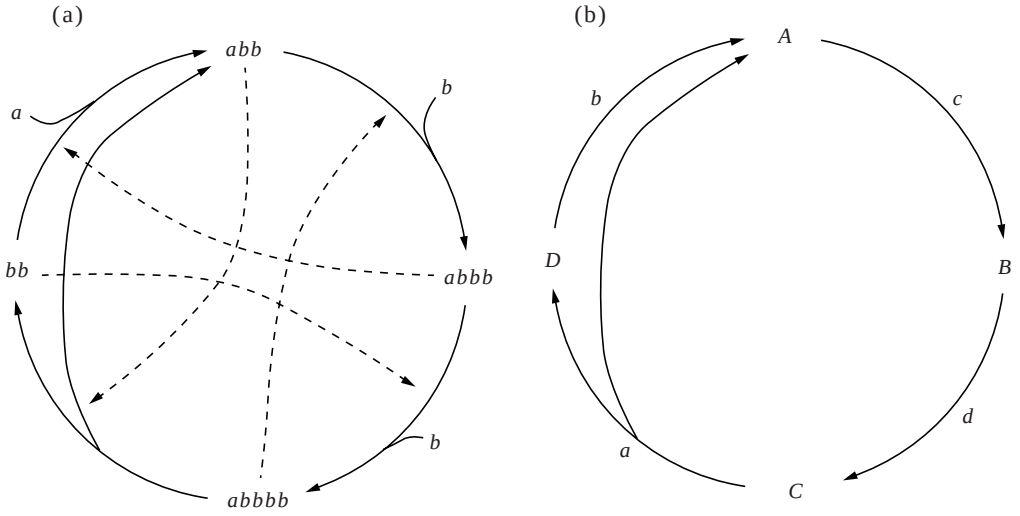


FIG. 16. (a) Possible tanglecycle in an autocatalytic reaction network. (– – –) indicates catalysis. Each reaction is not catalysed at random but according to strict match rules; (b) possible tanglecycle in modified typogenetics. Enzymes *a*, *b*, *c*, etc. are produced by strands *A*, *B*, *C*, etc. An enzyme encoded by a given strand catalyses the transformation of the only strand to which it is complementary according to strict match rules.

Fig. 16(b). The structures of the tanglecycles of Fig. 16(a) and (b) are equivalent. Their dynamical structure is not influenced by the fact that in modified typogenetics a substrate catalyses a reaction via an encoded enzyme and not directly. It must be noted that the computation of such modified typogenetics should require important computation time.

Non-equilibrium conditions are necessary because nothing interesting could occur at equilibrium. The distribution of material would be constant and self-reproduction would not be possible.

4.4.2. *The "beings" of typogenetics*

A random assignment of the reactions could be applied in typogenetics, for example a given enzyme could catalyse the transformation of a strand chosen at random. This case should approximate to the λ-system, for which the first step of the iterative process is to choose pairs of objects at random, or to the simple but also interesting system of binary strings developed by Banzhaf (1994) that behaves as a metabolic network in which binary strings are able to react at random with each other by means of a spatial reorganization of the information. On the contrary, typogenetics is completely deterministic and generates a unique "chemistry". It is the reason why the self-reproducing families can be considered as "beings", i.e. sets of strands, informants and transformers, distinguishable from all the other possible strands. So, the GC tanglecycle is composed of almost 5000 informants and at least as many transformers (an informant encodes at least one typoenzyme). For the TCGAG tanglecycle, there are almost 35000 informants. These "beings" are more than simple metabolisms, they are similar to cells without membrane in which the two classes of strands influence each other. Whatever the polymer content of the reactor, the composition of a tanglecycle will be the same, provided the reactor is supplied with a sufficient amount of monomers. The information contained in a string contains all the information needed to specify its properties. With random rules, the emerging organizations cannot be considered as "beings" for which an independent existence could exist, because these organizations depend on chance as well as on the chemical properties of the molecules that form these organizations. The random rule generates a set of possible chemistries, corresponding to all possible sequences of random choices (Bagley & Farmer, 1991).

4.4.3. *The action possibilities of typogenetics*

A feature of typogenetics that is not shared by the other autocatalytic networks is the existence of two classes of (molecular) structures acting on each other according to a code. In the λ-system for example, a molecule is an object with both a structure and an implied function. Its function, encoded by its structure, is revealed by the reactions in which it takes part. With typogenetics, this structure/function duality is achieved by two different kinds of molecules, the informants and the transformers, the function of the latter being coded by the structure of the former. But all transformers are not used for self-reproduction, for example, some of them do not bind to their strand or transform strands that are not members of families of self-replicators (the I_i'). In the same way, all the genes of a cell are not translated into enzymes which directly serve to reproduce them. Thus, typogenetics could be extended to give some phenotypic aspects to enzymes that are not required for self-replication.

Consequently, a tanglecycle possesses the property of the essential ingredient of life, namely self-replication associated with action possibilities. As written by Eigen, "Only part of the information stored in each I_i has to be used for coding the enhancing function in favor of formation of the next information carrier; other parts may be left for the coding of general enzymic functions such as translation, polymerization, control functions etc."

This dual information is, of course, found in the natural self-reproduction process. The information in the DNA is transcribed into messenger RNA and translated into proteins. This involves interpreting the information as instructions for constructing a protein. The information in the DNA is also replicated to form two copies of the original information. This involves simply copying the information without interpretation. With typogenetics the "DNA" is also translated into proteins, which involves interpreting the information, but here the self-reproduction is not explicit, it is not strictly speaking a replication (there is no DNA gyrase, DNA polymerase, DNA ligase, etc.). Self-reproduction is an "automatic" consequence of the translation provided it is followed by a feedback action of the transformers on the informants. Under these conditions, there are strands for which self-reproducing is literally included in the translation–retroaction couple. Such a strand will belong to a tanglecycle in which the information for self-reproduction is included in the strand itself. In our preceding paper we called typogenetics a self-reproducing molecular automaton referring to the self-reproducing cellular automata. The tanglecycles of typogenetics as well as some cellular automata like that of Langton (1984) are self-reproducing structures that achieve their simplicity by storing their

description in a dynamic loop rather than on a static tape. Therefore, with respect to the prebiotic molecular evolution, a tanglecycle is a more primitive model than a hypercycle because the replication is not built in but is an emergent property. The situation is the same when hypercycles are simulated by cellular automata, as in the work of Boerlijst & Hogeweg (1991) who simulated hypercycles with the help of a 300×300 cells cellular automaton in order to study their stability in the presence of parasites. Again, in this work, the replication phenomenon is implemented, it represents one of the rules of the automaton.

4.5. TANGLECYCLES POPULATIONS DYNAMICS

The notion of selection is a difficult problem to analyse, specially in the prebiotic stages. The approach taken by Eigen & Schuster (1982) and by Holland (1984) consists of a competition for raw material among different chemical structures. We used the same approach when we imposed a limited number of A, C, G and T bases. In these conditions, a tanglecycle can have a special behaviour if the constraints on the system do not suppress self-reproduction. It is the case for the GC tanglecycle that achieves two properties: an especially high homogeneity and the presence of a strange attractor.

	GC	TCGAG	Random
Homogeneity	0.46	0.16	0.18
Attractor	Strange	Steady state	—

A random population (in which GC and TCGAG tanglecycles were prohibited) subjected to the same constraints as the GC tanglecycle shows a low homogeneity and no attractor. Such a population is composed of a mixture of almost 150 strands among the possible $2 \cdot 10^{48}$ strands modifying themselves continuously according to the typogenetics rules. The autocorrelation function does not reveal any particular pattern and the algorithm of Grassberger–Procaccia does not allow to make a difference between this evolution and random series. If the evolution of such a ''soup'' could be followed for many generations it might be possible to observe some regularity. It would then be necessary to suspect the soup to have generated a tanglecycle.

There is indeed a certain probability that the new strand created at each generation could give a tanglecycle. Nevertheless, to create a tanglecycle is not sufficient, the latter must in addition resist to the 999 bases limitation, something the TCGAG tanglecycle cannot do for which the informants I_i and the self-reproduction disappear under these constraints. The GC tanglecycle preserves its self-reproduction properties in spite of the constraints. The autocorrelation function, applied to the number of GC strands [Fig. 8(b)] and to the total number of strands [Fig. 9(d)], shows that the system evolves to a certain periodicity. There is a peak centered on the generation numbers which are multiples of 50. From generation 107, when the limitation begins to be effective, the population is composed of a certain number of tanglecycles shifted in time with respect to each other (in addition to the initial GC, 36 GCs are appearing between generations 1 and 107). After that one can observe a certain synchronization of the tanglecycles. For example around generation 1500 (from 1491 to 1502), 71 GCs appear and thereafter none before the next peak beginning at generation 1546. The setting of the periodicity is accompanied by an increase of the homogeneity. The latter amounts to an average of 0.46, with a minimum of 0.14 (generations 1980 to 1983) and a maximum of 0.92 (generations 1117 to 1130). Hence, the pressure exerted on the system results in a selection. We are aware that selection must not be confused with sorting, as quoted by Fontana & Buss (1994b). Darwinian selection, for example, acts upon entities capable of reproduction (Vrba & Gould, 1986). Tanglecycles are capable of reproduction, and when, in our reactor, the pressure removes a strand, an entire tanglecycle with its descent can be removed if this strand is a selfR. But among all the tanglecycle strands, some are more sensitive than others to the base limitation, and the population, involving a high quantity of identical strands (up to 92%) is subjected to sharp fluctuations. However, in spite of the symmetric pattern, the evolution remains unpredictable, with chaos islets (for example from the generations 2400 to 2600 with many peaks). There is indeed an attractor but a strange one, with a dimension $d_s \sim 3.7$.

Similar results were reported for actual animal populations. So, Nicholson's (1954) studies of Australian blowflies under constant conditions were analysed by Brillinger *et al*. (1980). These authors showed that the blowflies exhibit chaotic dynamic behavior: most of the time the chaotic behavior has an almost periodic structure, but occasionally episodes in which the dynamics appear to be random are observed.

By using a phenomenological analysis based on the study of a time series, Schaffer & Kot (1985)

suggested that the frequency of the measles cases in New York and Baltimore from 1928 to 1963 (before the introduction of the vaccine) was exhibiting a chaotic behaviour with an underlying strange attractor. More recently Costantino *et al*. (1995) showed that it was possible to induce transitions to chaos by experimentally manipulating the adult mortality of the flour beetle *Tribolium* populations. Moreover, their results are in agreement with a mathematical model that is a system of three differential equations involving larvae, pupae and adults.

These dynamical similarities between tanglecycles and biological populations reinforce the idea that the reactor containing tanglecycles can be considered as a model for a test tube containing self-reproducing entities that behave like individuals.

A lot of work has been done (see for example May, 1987) in order to find models for the evolution of biological population. These models can be simple like those involving discrete non-overlapping generations (first-order differential equations) or more complex with discrete but overlapping generations (higher-order differential equations) or with continuous growth where regulatory effects contain time lags (time-delayed differential equations). So, a real (or artificial) population can be not only compared to a mathematical model but also analysed phenomenologically by methods based on the chaos theory. These two approaches are complementary and tend to show the relevance of the nonlinear mathematical models in populations biology.

5. Conclusion

As part of the works dealing with artificial life, especially in the prebiotic field, the emergence of higher organization levels from simple elements is clearly of great importance. So, self-replication has been, since von Neumann, a property subjected to intensive studies. The experimental emergence of self-replication was achieved by means of cellular automata (Langton, 1984) and we also achieved it with our typogenetics molecular automaton. With the Langton automaton a collection of cells is self-reproducing and could be assimilated to an organism. With the tanglecycles a collection of molecules is self-reproducing and could be assimilated to a cell without membrane or if preferred, to an ''acellular being'' that preserves its information in a dynamic loop.

Therefore, if we consider the hierarchic sequence of life

$$molecule \rightarrow cell \rightarrow organism \rightarrow society$$

a tanglecycle is at a lower level than a cellular automaton.

Molecular automata like tanglecycles represent another approach of the self-organization processes of matter. In our model, molecules self-organize through a tangled hierarchy according to the transforming codes

$$informants \rightarrow transformers$$

and

$$transformers \rightarrow informants$$

This structure from which self-reproduction originates can also give rise to other classical behaviour of life, for example a selection of strands, a quasi-periodic organization of the population or the presence of attractors (strange or not).

In the present living world, informants are nucleic acids and transformers are proteins. The coding *informants* → *transformers* involves many physico-chemical interactions (Van der Waals, hydrogen, hydrostatic bonds) that we are used to summarize in the genetic correspondence code *set of codons* → *set of aminoacids*, whereas coding *transformers* → *informants* also involves physico-chemical interactions as for example those taking place in the formation of enzyme–substrate complexes (DNA polymerase–DNA, etc.). When we simulate *in silico* such interactions (typogenetic code, tertiary structure of typoenzymes, etc.), some strands give tanglecycles in which self-reproduction and other emerging properties appear without being explicitly programmed.

In the future tanglecycles could be the object of many studies, notably concerning their action possibilities. We also suggest that the time series generated by tanglecycles under constant conditions could be used as data to improve the techniques utilized to distinguish chaos from noise, as for example the nonlinear forecasting techniques proposed by Sugihara & May (1990).

As we pointed out above, the self-reproducing function of tanglecycles requires just a fraction of the enzymes encoded by the informants. Then, it would be possible to program other silico-chemical elementary interactions, to let ''nature'' follow its course and to observe the probable emergence of new properties.

In the same way as solar energy allows a local decrease of entropy when the matter self-organizes and becomes more and more complex according to its physico-chemical properties, the electric energy of the computer allows virtual matter to self-organize

provided it possesses silico-chemical features allowing this self-organization.

That life began with the appearance of an autocatalytic self-replicating molecule is an hypothesis used as the "minimal assumption" (Elitzur, 1994) from which the emergence of life from inanimate matter should follow as a natural consequence. We agree with this proposition and we suggest that, as in our tanglecycles, the features allowing *informants–transformers* self-referent loops could play an essential role in the way leading from inanimate matter to life.

We wish to thank J. M. Frère (University of Liège) for his interest in this work. This work was supported by the Belgian Government in the frame of the Pôles d'Attraction Interuniversitaires (PAI P4/03).

REFERENCES

Bagley, R. J. & Farmer, J. D. (1991). Spontaneous emergence of a metabolism. In: *Artificial Life II* (Langton, C. G., Taylor, C., Farmer, J. D. & Rasmussen, S., eds), pp.93–140. Redwood City: Addison-Wesley.

Bagley, R. J., Farmer, J. D., Kauffman, S. A., Packard, N. H., Perelson, A. S. & Stadnyk, I. M. (1989). Modeling adaptive biological systems. *Biosystems* **23,** 113–137.

Banzhaf, W. (1994). Self-organization in a system of binary strings. In: *Artificial Life IV* (Brooks, R. A. & Maes, P., eds), pp.109–118. Cambridge, MA: MIT Press.

Boerlijst, M. C. & Hogeweg, P. (1991). Spiral wave structure in pre-biotic evolution: hypercycles stable against parasites. *Physica D* **48,** 17–28.

Brillinger, D. R., Guckenheimer, J., Guttorp, P. & Oster, G. F. (1980). Empirical modeling of population time series data: the case of age and density dependent vital rates. In: *Some Mathematical Questions in Biology*, Vol. 13 (Oster, G. F., ed.), pp. 65–90. Providence, RI: AMS.

Chacon, P. & Nuño, J. C. (1995). Spatial dynamics of a model for prebiotic evolution. *Physica D* **81,** 398–410.

Costantino, R. F., Cushing, J. M., Dennis, B. & Desharnais, R. A. (1995). Experimentally induced transitions in the dynamic behaviour of insect populations. *Nature* **375,** 227–230.

Darley, V. (1994). Emergent phenomena and complexity. In: *Artificial Life IV* (Brooks, R. A. & Maes, P., eds), pp. 411–416. Cambridge, MA: MIT Press.

Eckmann, J. P. & Ruelle, D. (1985). Ergodic theory of chaos and strange attractors. *Rev. Mod. Phys.* **57,** 917–656.

Eckmann, J. P. & Ruelle, D. (1992). Fundamental limitations for estimating dimensions and Lyapounov exponents in dynamical systems. *Physica D* **56,** 185–187.

Eigen, M. (1971). Selforganization of matter and the evolution of biological macromolecules. *Naturwissenschaften* **58,** 465–523.

Eigen, M. & Schuster, P. (1979). *The Hypercycle*: *a Principle of Natural Self-organization*. Berlin: Springer.

Eigen, M. & Schuster, P. (1982). Stages of emerging life—five principles of early organization. *J. Mol. Evol.* **19**(1), 47–61.

Eigen, M., Gardiner, W., Schuster, P. & Winkler-Oswatisch, R. (1981). The origin of genetic information. *Sci. Am.* **244,** 78–94.

Elitzur, A. C. (1994). Let there be life. Thermodynamic reflections on biogenesis and evolution. *J. theor. Biol.* **168**(4), 429–459.

Essex, C., Lookman, T. & Nerenberg, M. A. H. (1987). The climate attractor over short timescale. *Nature* **326,** 64–66.

Farmer, J. D., Kauffman, S. A. & Packard, N. H. (1986). Autocatalytic replication of polymers. *Physica D* **22,** 50–67.

Fontana, W. & Buss, L. W. (1994). What would be conserved if "the tape were played twice"? *Proc. Natl. Acad. Sci. U.S.A.* **91**(2), 757–761.

Fontana, W. & Buss, L. W. (1994). "The arrival of the fittest": toward a theory of biological organization. *BULL. MATH. BIOL.* **56,** 1–64.

Grassberger, P. (1986). Do climatic attractors exist? *Nature* **323,** 609–611.

Grassberger, P. & Procaccia, I. (1983). Measuring the strangeness of strange attractors. *Physica D* **9,** 189–208.

Gutzwiller, M. C. (1992). Quantum chaos. *Sci. Am.* **266**(1), 26–32.

Hofstadter, D. (1979). *Gödel*, *Escher*, *Bach*: *An Eternal Golden Braid*. New York: Basic Books.

Holland, J. H. (1976). In: *Automata*, *Languages*, *Development* (Lindenmayer, A. & Rozenberg, G., eds), pp. 385–404. Amsterdam: North-Holland.

Ito, K. & Gunji, Y. P. (1992). Self-organization toward criticality in the Game of Life. *Biosystems* **26,** 135–138.

Kauffman, S. A. (1986). Autocatalytic sets of proteins. *J.theor. Biol.* **119,** 1–24.

Kauffman, S. A. & Johnsen, S. (1991). Coevolution to the edge of chaos: coupled fitness landscapes, poised states, and coevolutionary avalanches. *J. theor. Biol.* **149,** 467–505.

Langton, C. G. (1984). Self-reproduction in cellular automata. *Physica D* **10,** 135–144.

Langton, C. G. (1986). Studying artificial life with cellular automata. *Physica D* **22,** 120–149.

Langton, C. G. (1990). Computation at the edge of chaos: phase transitions and emergent computation. *Physica D* **42,** 12–37.

Laskar, J. (1995). La stabilité du système solaire. *Pour la science* **Janvier,** 45–47.

Laskar, J. & Froeschlé, C. (1991). Le chaos dans le système solaire. *La Recherche* **232,** 572–582.

May, R. M. (1987). Chaos and the dynamics of biological populations. *Proc. R. Soc. London* **413A,** 27–44.

McIntosh, H. V. (1990). Wolfram's class IV automata and a good life. *Physica D*, **45,** 105–121.

Morris, H. (1989). Typogenetics: a logic for artificial life. In: *Artificial Life* (Langton, C. G., ed.), pp. 369–395. Redwood City: Addison-Wesley.

Mullin, T. (ed.) (1993). *The Nature of Chaos*. Oxford: Oxford Science Publications.

Nicholson, A. J. (1954). An outline of the dynamics of animal populations. *Austral. J. Zool.* **2,** 9–65.

Nicolis, C. & Nicolis, G. (1984). Is there a climatic attractor? *Nature* **311,** 529–532.

Nicolis, C. & Nicolis, G. (1987). Evidence for climatic attractors. *Nature* **326**, 523–523.

Olsen, L. F. & Schaffer, W. M. (1990). Chaos versus noisy periodicity: alternative hypothesis for childhood epidemics. *Science* **249,** 499–504.

Orléan, A. (1991). Les désordres boursiers. *La Recherche* **232,** 668–672.

Pool, R. (1989). Is it chaos, or is it just noise? *Science* **243**, 25–28.

Ruelle, D. (1990). Deterministic chaos: the science and the fiction. *Proc. R. Soc. London* **427A,** 241–248.

Schaffer, W. M. & Kot, M. (1985). Nearly one dimensional dynamics in an epidemic. *J. theor. Biol.* **112,** 403–427.

Schnabl, W., Stadler, P. F., Forst, C. & Schuster, P. (1991). Full characterization of a strange attractor. *Physica D* **48,** 65–90.

Sole, R. V. & Bascompte, J. (1994). Ecological chaos. *Nature* **367,** 418.

Sole, R. V. & Valls, J. (1992). On structural stability and chaos in biological systems. *J. theor. Biol.* **155,** 87–102.

Sugihara, G. & May, R. M. (1990). Nonlinear forecasting as a way of distinguishing chaos from measurement error in time series. *Nature* **344,** 734–741.

Tsonis, A. A. & Elsner, J. B. (1988). The weather attractor over very short timescales. *Nature* **333,** 545-547.

Varetto, L. (1993). Typogenetics: an artificial genetic system. *J.theor. Biol.* **160,** 185–205.

Vasilakos, K. & Beuter, A. (1993). Effects of noise on a delayed visual feedback system. *J. theor. Biol.* **165,** 389–407.

VRBA, E. & GOULD, S. J. (1986). Sorting is not selection. *Paleobiology* **12,** 217–228.
WOLFRAM, S. (1984). Universality and complexity in cellular automata. *Physica D* **10,** 1–35.
WU, Z. B. (1995). Remark on metric analysis of reconstructeddynamics from chaotic time series. *Physica D* **85,** 485–495.

APPENDIX

Typogenetics

STRANDS, BASES AND ENZYMES

Typogenetics introduced by Hofstadter consists in performing typographic manipulations on some sequences of letters, namely sequences of the four letters A, C, G and T. Arbitrary sequences of these letters, e.g. CCA, ACGTTACG or GTCTGAATCGTACACGTGACT, are called strands. The letters are called bases and the positions they occupy units. One strand can be modified in different ways, it can be extended, shortened, cut or copied. Those operations can be performed by typographic enzymes or typoenzymes. The typoenzymes bind to the strands and modify them. Each enzyme has a binding specificity according to rules that will be explained later. So, enzymes which bind, respectively, to an A or a C base will be called an A- or a C-enzyme.

THE COPY MODE, THE DOUBLE STRANDS AND THE 5′ AND 3′ENDS

A whole strand or a part of a strand can be copied by some typoenzymes. An A and a C base will be copied to a T and a G base, respectively. This is called base pairing. A and G are called purines and C and T pyrimidines.

base pairing

$$A \leftrightarrow T$$

$$G \leftrightarrow C$$

Thus copying a strand consists, in fact, in creating a complementary strand, yielding a double strand such as

TACT

ATGA

The TACT strand is read from the left (5′) to the right (3′) end, and the complementary strand from the right to the left end.

$^{5'}$TACT$^{3'}$

$_{3'}$ATGA$_{5'}$

Thus, after separation, the resulting strands are TACT and AGTA.

AMINOACIDS, TRANSLATION AND TYPOGENETIC CODE

A typoenzyme consists of sequences of two letter commands called aminoacids. The aminoacids can perform 15 different operations on the strands (Table A1). When an enzyme binds to a strand, the aminoacid commands apply in turn on the strand. Where do the typoenzymes come from? The answer is: they come from the strands themselves. Every

TABLE A1
The different aminoacid (AA) commands and their associated functions

Command	Doublet/AA	Function
Cutting	AC/cp	Cutting between the binding unit and the preceding base.* The left part of the cut strand is no longer modified by the enzyme which remains bound to the right part
Elimination	AG/el	The base to which the enzyme is bound is eliminated as well as the complementary base and the enzyme binds to the following base
Switching	AT/ba	The enzyme switches from the main strand to the complementary strand. When it happens, the strands are inverted: the main strand becomes the complementary strand
Moving	CA/dd	The enzyme moves one unit to the right (dd) or to the left (dg)
	CC/dg	
Copying	CG/co	Copy mode turned ON (co) or OFF (fi): the operations of moving,
	CT/fi	introducing and searching create the complementary strand or not
Introducing	GA/ic	Introduction of the specified base on the right of the binding unit
	GC/ig	
	GG/it	
	GT/ia	
Searching	TA/yd	The enzyme moves on the right (yd) or on the left (yg) until a pyrimidine is founded
	TG/yg	
	TC/ud	The enzyme moves on the right (ud) or on the left (ug) until a purine is founded
	TT/ug	

* If a preceding base does not exist (beginning of the strand) the command is ignored. It is a general rule: if a command is impossible to execute, it is simply ignored.

TABLE A2
The typogenetic code

First base \ Second base	A	C	G	T
A		cp *s*	el *s*	ba *g*
C	dd *s*	dg *s*	co *g*	fi *d*
G	ic *s*	ig *g*	it *g*	ia *d*
T	yd *g*	ud *d*	yg *d*	ug *d*

Each doublet is translated into an aminoacid. For example the ATCGGTTCG strand is translated into the *ba–co–ia–ud* typoenzyme. Note that this strand contains an odd number of bases, therefore the last base does not belong to a doublet and has no corresponding aminoacid.

strand can be translated into one or more typoenzymes. This translation from the strands into the enzymes follows a typogenetic code (Table A2) according to which every *doublet* of adjacent bases is coding for an aminoacid. The string is partitioned in doublets prior to translation and the reading frame is strictly defined by the first base of the strand (5′ end) so that we have no multiple usage of the genetic information in different reading frames.

THE TERTIARY STRUCTURE AND SPECIFICITY OFENZYMES

The letters *s*, *g* and *d* in the lower right corner of each code square have an important function, because they dictate the enzyme folding that gives to the enzyme a binding preference (specificity) for an A, C, G or T base. The aminoacid sequence dictates folding and specificity (Table A3).

TABLE A3
The rules of folding and specificity of a typoenzyme

(1) given an initial direction ⇒ to which points the first amino acid;
(2) the *g* letter prescribes a left turn in the typoenzyme, the *d* letter a right turn and the *s* letter prescribes to reverse the initial direction;
(3) operation 2 must be repeated for each aminoacid;
(4) finally, the enzyme binds to the base A, C, G or T if the angle value (in a trigonometric sense) between the directions towards which the first and the last aminoacid point is 0, 90, 180 or 270°, respectively
For example the letters associated to the *ba–co–ia–ud* enzyme are respectively *g*, *g d* and *d*. Its tertiary structure is

```
ud—
|     that gives an angle value of 0°
ia—co     and a specificity for an A base
     |
    ⇒ ba
```

PUNCTUATION, GENES AND RIBOSOMES

The AA doublet serves as a punctuator between the genes of a multi-gene strand. The strand

```
CGATTTCACTA  AGAAGGGTC
←———————→ ←———————→
  Gene 1      Gene 2
```

will be translated into two typoenzymes, respectively *co–ba–ug–dd–fi* and *ic–el–it–ud*. The machinery which reads and translates the strands is called a typoribosome, but in fact it is the person (or the computer) deriving the strands who plays that role. It is important to see that this person or computer is not responsible for the tertiary structure of the enzymes which is completely defined by the base sequence of the strand.

FROM GENERATION TO GENERATION

Now consider typogenetics as a formal system, beginning with a single strand, for example CGATTTCACTAAGAAGGGTC. This strand is translated with the help of the typogenetic code into the two typoenzymes *co–ba–ug–dd–fi* and *ic–el–it–ud*. Consequently, these enzymes can bind to the strand and then execute the various operations associated with each of their aminoacids, modifying (or not) the original strand to give one (or several) new strand(s). The same process applies to the daughter strand(s) and so on. It should be stressed that, when several strands code for several enzymes, one given enzyme only acts on the strand it originates from.

Typogenetics Computerization

UNAMBIGUOUS RULES FOR A COMPUTER PROGRAM

The characteristic of a formal system is that it uses selected symbols according to unambiguous rules, and that the application of these rules yields well-defined results. Typogenetics as presented by Hofstadter was incomplete because some rules and definitions were lacking. It was then necessary to complete the system in order to allow a computer aided derivation of the strands:

- the basic program will consider the strands and the enzymes as character string variables;
- the folding letters *s*, *g* and *d* will be replaced, respectively, by the numerals 1, 2 and 3. Moreover, 0 will correspond to the AA doublet. Then, the angle values (0, 90, 180 and 270°) will be replaced by four numbers: 0, 1, 2 and 3. These

numbers will be called the specificity numbers of each aminoacid. The typoenzyme specificity is found by adding up the specificity numbers, but in such a way that the result will be also 0, 1, 2 or 3. Nothing is changed, each aminoacid still contributes to the specificity but a computer is now able to calculate it. The binding preference is stated as a base. When more than one unit in the strand is of that base type, the enzyme binds to the first one starting from the 5′ end;

- when many enzymes are produced by a strand, they each bind to the strand, in turn. The first one binds, works on the strand and is released. Thereafter the second enzyme binds to the modified strand and so on. If two AA doublets are consecutive, one of them is removed, (CTAAAACG becomes CTAACGC) which avoids the proliferation of strands which would only differ by the number of AA doublets and would thus have the same "meaning";
- some strands are translated into enzymes which do not modify them, for example an A-enzyme and a strand devoid of A base. Those strands are not further modified and the program eliminates them from the following generations;
- the strands are stored on a computer disk. Each generation is stored as a relative file and the different strands of a generation constitute different records of that file. But some strands could become longer and longer, so it is necessary to choose a maximum length (ML). If a strand becomes longer than this ML value, it is eliminated from the following generations. It can be assumed that a too long strand becomes unstable and dies out.

THE PROGRAM: OVERALL VIEW

The program treats the strands one by one. A strand i (record i) of generation n (file n) is called from the disk and after treatment by the program the daughter strand(s) will be saved as a record $i, j, k, l \ldots$) of the generation $n + 1$. The different versions of the main program as well as the programs intended to compute homogeneity, autocorrelation and other algorithms were written for the Commodore AMIGA computers in AMOS (compiled basic language) and computed on AMIGA 2500 (Motorola 68020/68881, 14 MHz), AMIGA 4000/30 (Motorola 68030/68882, 25 MHz) and AMIGA 4000/60 (Cyberstorm Card with Motorola 68060, 50 MHz). A version of the main program is also available from the author for the IBM PC and compatible computers in GW BASIC.