FALSIFICATION OF THE EVOLUTIONARY THEORY WITH FLIPP

José del Carmen Rodríguez Santamaría
https://www.ejristos.com
Bogotá, Colombia

December 14, 2021

Abstract

A theory is false when it produces wrong predictions. We exhibit a very important and generic prediction of the Evolutionary Theory (ET) that is contradicted by Nature. We find that the Evolution of Perfection is mandatory and that underperfection is immanent to Evolution in the face of complexity, the natural enemy of Evolution. Since this is not observed in Nature, we conclude: Natural Evolution has nothing to do with our origins. Our falsification is final because it captures the essence of Natural Evolution, which rests upon two pillars: the Genome is software, a set of encoded instructions, and Evolution is a software developer. That is why Genetic Programming (GP), the use of Evolution to design software, was used as the testbed. Our GP uses Flipp, a programming language especially suited to print binary strings, for which complexity and length are synonyms.

Contents

1 OVERVIEW **2**

2 GENETIC ALGORITHMS (GAs) **3**
2.1 The historical perspective 3
2.2 A mighty sophism 5
2.3 Objection to GAs 5
2.4 Our purpose 6

3 THE GENOME IS SOFTWARE **6**
3.1 The power of the word 6
3.2 Computers and software 6
3.3 Informatics of the genome 8

4 ALGORITHMS EVERYWHERE **9**
4.1 Animals must solve algorithmic tasks 9

5 EVOLUTION IS A SOFTWARE DEVELOPER **10**
5.1 Algorithmic interpretation of the human diversity 10

6 THE VERBAL ET **10**
6.1 Formulation 10
6.2 Is the Verbal ET natural? 11
6.3 The falsifying challenge 12

7 TESTING THE VERBAL ET **12**
7.1 An incredible prediction 12
7.2 High quality and astonishing complexity as test 13
7.3 Ubiquitous and total perfection as a test 13

8 THE FLIPP COMPUTER **15**

9 THE COST OF PERFECTION **16**
9.1 SPEC as an abstraction 16
9.2 The cost of SPEC 17
9.3 The evolutionary arisen of genes 18

10 THE COST OF IMPERFECTION **18**
10.1 Our simulations 18

11 THE CONUNDRUM OF TELEONOMY **19**

12 BUGS **21**
12.1 Random walks in Flipp GP 22
12.2 The dynamics in Flipp GP 22
12.3 The tapestry of perfection 23

13 CONTEXTUALIZATION **24**
13.1 Relation with creationism 24
13.2 Relation with Science 24
13.3 Relation with persons 25
13.4 Jesus comes! 25

14 CONCLUSION **25**

1 OVERVIEW

How nice it is to look at champions in their exhibitions! They show us the impressive power of the species. At the root of these incredible realizations resides Evolution: two changes here, a rearrangement there and the result might eventually represent an improvement. Sometimes a trend arises, improvements accumulate and wonders appear. Marvelous. This involves our personal life and habits, the Culture around us, and the Genome. So, Evolution Everywhere is the air we breathe.

The accumulation of improvements might be explained by our strong instinct of detecting and picking the best. We are so used to this that a genius was needed to bring forth tens of observations to sustain the idea that in Nature no picking agent is necessary because picking of the best happens spontaneously:

In the first place, there are always more offspring than can survive, and those that recurrently do it carry special traits, which may vary from individual to individual. Secondly, survivors reproduce, and transmit their winning traits. Third: when the process

runs generation over generation, perfection and complexity arise. The overall process is called Evolution while the emerging process in which the reproduction of the best is favored is named Natural Selection. This was the fundamental stone where Darwin built his Evolutionary Theory (ET) of the origin of species ([4] 1859).

In regard to the corroboration of the ET, Natural Biological Evolution is part of our daily reality as the Evolution of viruses shows it. Besides, it has been applied in many, many diverse fields. Actually, simulations of Evolution in the form of Genetic Algorithms (GAs) are ordinary tools to find approximated solutions to optimization problems. Our particular interest here is in a special type of GAs in which Evolution is applied to the development of software, a flourishing and challenging discipline that is called Genetic Programming (GP). Let us recall that software is the generic name given to texts that carry instructions that can be decoded and executed by the CPU of a computer. Thus, the essence of GP is that these texts also can undergo Evolution just as those in DNA. So, the ET predicts that the creation of software by means of Evolution shall be fruitful. This is at present a thoroughly verified prediction and our own simulations here make its verification accessible for desk computers and cell phones.

The relation of software with Biology is that Natural Evolution is indeed Natural GP. This idea comes from a simple observation: DNA texts convey instructions to be interpreted and executed by the ribosome to sustain and to continue life. Our point is that there is an equivalence of the ribosome with a CPU of a computer, and of the Genome with Software. Thus, the Evolution of the Genome is indeed Natural GP. We can now update the ET in terms of Software development. The result is called the **Verbal ET** that claims: *Evolution is the one and only software developer that is responsible for the creation of the software in the Genomes of all species*. To test it, we calculate predictions with simulations in JavaScript and Flipp, a programming language especially suited to print binary strings. Our aim is to see how is the battle in GP of Evolution against complexity. After a tedious but at last simple research, we find that under-perfection must be ubiquitous and that the Evolution of Perfection is mandatory in GP as a permanent observable. This amounts to a prediction of the Verbal ET: our bodies shall be full of imperfections and our physiology of malfunctions. Since this is not found in our bodies, we conclude that Evolution cannot be the explanation of our existence. In the philosophy of science, this is called a falsification (McLeod [23] 2020): The ET has been falsified by GP over Flipp.

Let us develop now our insight.

2 GENETIC ALGORITHMS (GAs)

The question that orients us is whether the complexity and perfection of living beings can be explained by Nature. Many thinkers have helped us with incredible discussions. In our opinion, the summit of clarity is captured by the GAs.

2.1 The historical perspective

Human beings are christened with two names, one from the family plus another that is personal. Linnaeus proposed in 1753 to name species just in the same way: a genus, the general group in which they fall in - such as `pine` or `Pinus` -, and a personal name, such as `lambertiana` (Pearson [24] 2003). Behind this anthropological accounting idea, there is a powerful conceptualization: creatures can be organized according to

common characteristics into a nested hierarchy (BerkeleyUE [3] 2010?). To pass from it to trees was for Lamarck a sort of duty with the plus of interpreting the tree in terms of descent, of biological evolution. He proposed in the early 1800s an Evolutionary Theory (ET) in which Evolution was unable to overcome complexity alone and thus needed to be guided by alchemical forces. In fact, Lamarck was adept at alchemy that he preferred instead of the quantitative chemistry of Lavoiser (Wikipedia [36] 2015). Nevertheless, when one searches for the mechanism as alchemical forces interact with living matter to guide its Evolution, the only possible hint is in the nervous fluxes, which he emphasizes too much as the key to his theory (Lamarck [19] 1809).

Next, Darwin ([4] l.c., 1859) proposed the Evolutionary Theory (ET) in which Evolution was postulated as a complexity killer thanks to Natural Selection, the differential reproduction of individuals. He lacked a mechanism of heredity. Later, Mendel was an idealist that around 1865 forced field experiments with peas to sustain his thought that inheritance is tied to the transmission of material items (Science Learning Hub – Pokapū Akoranga Pūtaiao [31] 2011). His idea resulted to be correct and derived in modern Molecular Biology with DNA, genes, and all that.

The ideas of Darwin can be variously blended with the genetics of Mendel. In the first place, the expectations of Darwin can be realized in Genetics if one assumes that mutation can be stopped at those sites that already gained a correct value: we have here a guided mutation, i.e., Lamarckism. The resulting Evolutionary Theory is miraculous, just as expected by Darwin (Rodríguez, [29] 2021). In hindsight, we can say *Darwin + Mendel = Lamarck*. Nevertheless, the idea of a guided mutation was seriously combated by de Vries in 1901, who worked mainly with plants. He claimed that the environment cannot guide the direction of mutation, which occurs in jumps. In that way, he freed Evolution from the ideas of Lamarck and opposed the gradualism of Darwin (Hall [9] 1935). Another form of blending Darwin to Mendel was proposed in the Synthetic Theory, formulated around 1930, and that for all effects is currently equivalent to actual Population Genetics (Jorde [14] 2016). In it, the mutation is not guided, rather it is at random. This new reformulation of the ET is accepted by Modern Science and is heralded to be the explanation of our existence.

In the second half of the XXth century, Computer Scientists and Engineers worked out the Genetic Algorithm (GA) which is a simulation of the ET in computers (Rodríguez [28] 2019). According to my experience, no one can be sure of his or her correct understanding of the ET unless he or she executes with pencil and paper a GA. Very difficult is for students to simulate mutation at random because they inadvertently tend to guide it. Because they think that mutation must be guided, all of them accept that we are here because of Evolution. By contrast, GAs use generators of random numbers that are sufficiently good, and with them, one easily grasps various conclusions:

- Evolution in GAs is abstracted in terms of mutation, recombination, differential surviving, reproduction, and death. The fundamental idea of GAs is that Evolution can solve optimization problems if individuals represent approximated solutions and if these can reproduce in proportion to the quality of the approximation.
- GAs have proved to be a powerful method to find approximated solutions to optimization problems of whatever nature. This has created a sophism, see below.

- Evolution can be found everywhere because the mechanism depicted by GAs is simple and ubiquitous.
- The difference between controllable and random mutation is that in the first case Evolution goes upwards always miraculously. In the second, mutation undoes at night what it does in the day. The result is that Evolution goes upwards thanks to Natural Selection, but so slowly that it cannot be a complexity killer any further. Rather:
- Complexity is a burden for Evolution. Thus, if solutions are improved, this might happen very, very slowly.

2.2 A mighty sophism

GAs are used every day all through the world to attack problems of Science and Engineering. That is why it is universally believed and taught that the GAs are a verification of the ET and that therefore the ET is true. Nevertheless, this is a sophism. Instead, the truth is this:

- The ET would be immediately falsified if the GAs would not function because the GAS implement the main idea of the ET.
- The GAs work and hence, this is indeed a verification of the ET.
- If a theory is verified, the theory can anyway be false. In the present case, the verification is very powerful and hence, it is predicted that many features of life can be correctly explained and reproduced by Evolution. It is the task of Science to elucidate by means of the Scientific Method which features fit into this class.
- In the light of the GAs, the ET has a mandatory prediction that is false. The mandatory prediction is that when Evolution can solve a complex problem, the Evolution of Perfection must necessarily exist: solutions synthesized at random are rather crude approximations that are turned by Evolution into mediocre solutions that next evolve into good solutions that next battle for perfection. This takes time. This must take a long time, so much that it is predicted by the ET that under-perfection must abound everywhere as in the fossil record as in our own body, anatomy, and function. Since this is not seen in the immense majority of babies of every species, the ET is false (Rodríguez l.c. [29] 2021).

I learned the virtues and pitfalls of Evolution from GAs. While most people present them as powerful proof of the ET, we demand to make it clear that GA's prove that Evolution is indeed useful but not the explanation of our existence because Evolution is far from green: the Universe provides not enough resources to run GAs to explain our complexity. Now we ask ourselves: How strong is the falsification of the ET by means of the GAs?

2.3 Objection to GAs

One spontaneously imagines that GAs have something to do with Nature. Such a belief is darkened by observing that GAs imperfectly reflect Natural Evolution. The correct, fair, and transparent insight on Natural Evolution takes into account two additional factors. First, Natural Evolution deals with encoded instructions to continue life. Thus, it

is verbal, algorithmic by its very nature. Second: GAs have as a purpose the optimization of the function that encodes the problem to be solved, but Nature has no purpose at all according to the materialistic philosophy that Science follows. So and strictly, Science cannot use GAs either to support the ET or to reject it. That is why we need a new frame, the correct one.

2.4 Our purpose

The DNA carries encoded instructions for the continuation of extant life. So, the Genome is software and the Evolution of these texts is indeed the Evolution of Software. We are now in the position of reformulating the ET in conceptual, essential terms. The result is the **Verbal ET** that we elaborate on below. It is very human and transparently reflects Evolution as a computing phenomenon. Moreover, if clarity is worked out carefully, the falsification of the Theory must spontaneously arise in the mind of the Reader. Our responsibility here is to prove that this is reproducible for every person with a desktop and along with the standards of the Scientific Method. Our programs run over web browsers even over cell phones, if only a low level of complexity is selected.

We enter now to develop the algorithmic part of the **Verbal ET** and to enable the formulation of tests according to the Scientific Method.

3 THE GENOME IS SOFTWARE

The DNA encodes for verbal instructions to make proteins, i.e., it is software.

3.1 The power of the word

We can converse. Simple as this might look, it is a powerful miracle. Let us see.

The teacher says to his pupil: *Raise your right hand up*. The boy raises his right hand. The teacher says: *sing your beloved song*. The child sings a lullaby. The teacher says: *eat and drink*. The kid eats an apple. The teacher says: *add 2 plus 2*. The infant answers: *four*. The teacher says: *we are done, go home*. The little one ends every task, makes the backpack, and goes home.

The incredible finding is that verbal communication can be encoded and automated. It suffices to look at its mechanics: an instruction is encoded in a string of verbal sounds or phonemes, it is interpreted according to a given table or code, and the instruction is executed. This triplet is the trademark of verbal communication, see figure 1.

3.2 Computers and software

Our term **verbal** refers to a string of sounds that conveys information. Many cultures found that verbal communication can be encoded into texts that consist of chains of chars or symbols. The relation of texts with verbal utterances is mediated by a code of interpretation that is capricious. Nevertheless, the whole package is variously optimized, say, external noise must cause low interference. Texts may contain instructions that conform to a procedure, a program that upon execution produces actions.

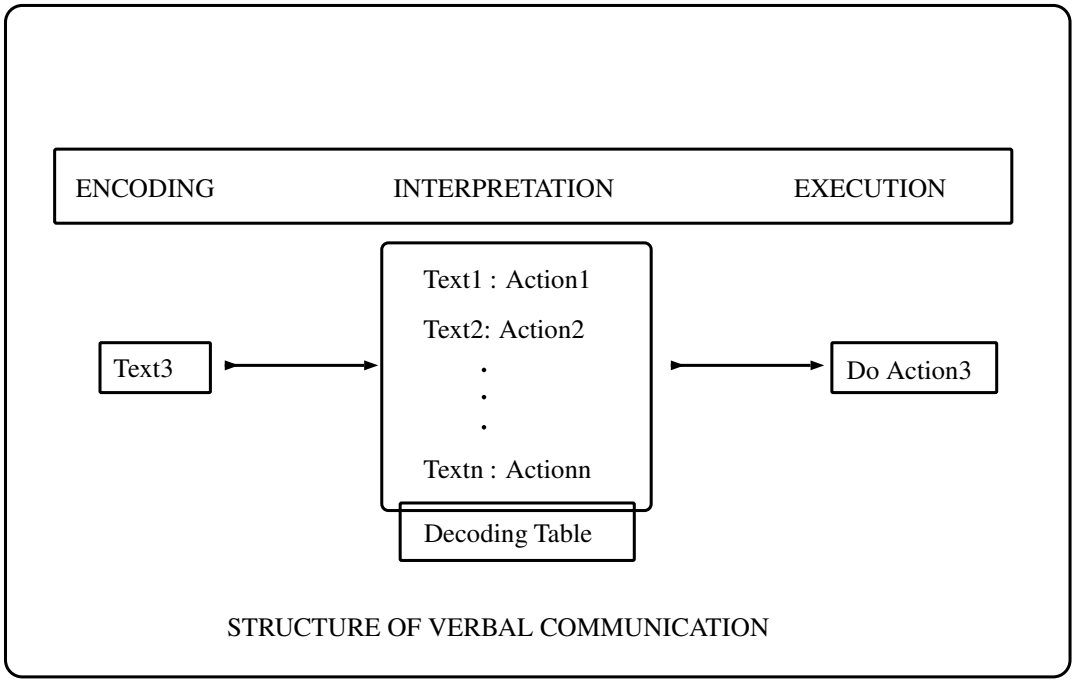


Figure 1: *A verbal instruction is encoded in a text, it is interpreted according to a given table or code, and the instruction is executed. This triplet is universal and has allowed the design of every type of computers.*

We immediately catch a connection with computers:

A computer is a machine that can execute verbal tasks of the most varied nature, just as we do. For instance, one can command to send an e-mail, pass next to play music, to continue with calculus of the financial balance of a company that is ensued with the simulation of a flying aircraft, to end with the shutdown of all operations.

The elementary definitions of Computer Science are the following:

- The hardware is the set of physical components of a computer.
- A program is a list of verbal instructions, functions, methods, or procedures to operate a computer.
- The software consists of the programs to operate it.
- A robot is a computer capable of taking decisions and converting them into actions. It might enjoy an outstanding degree of autonomy.
- A book is a software package to help human beings to program his or her mind for the better. Human beings are quite extraneous robots, so much that they love to claim that they are free and that henceforth they are responsible for their actions.

Now, a connection to Biology.

3.3 Informatics of the genome

The cell is the fundamental building block of life. The DNA of the cell functions on verbal processing. For instance, the DNA of the cells of the stomach is starting to assemble a protein and it says:

Initialize a polypeptide chain with `Ala`. The cell initializes a polypeptide chain with Alaline. The DNA says: *add* `Leu`. The cell adds an amino acid of Leucine to the chain. The DNA says: *add* `Ile`. The cell adds an amino acid of Isoleucine...The cell says: *add* `Ala` and Alaline is added to the chain. The cell says: *mark the chain to be exported out of the cell*. The cell marks the chain with appropriate signaling. The DNA says *end*. The cell ends the chain and releases it for possible further processing. And all this would have been triggered by the need to make more pepsin to digest food in the stomach of a pig.

The way commands are encoded in the DNA is called the **Genetic Code** (NIH, [22] 2021).

1 ***Exercise.*** *Show that the Genetic Code fits the general structure of verbal communication. Answer*

Thus, we see that a cell is a robotic entity. This means that the cell has an inner computer that is based on verbal processing. It is programmed to allow for autonomous but interactive decision-making, such as self-feeding and reproduction.

The more astonishing proof that the cell relies on verbal processing is that it can follow on the fly the more strange commands.

Example with bacteria. A phage, a bacteria killer virus, might arrive that invades the bacteria cell, unfolds its own genetic material, the cell grasps it, and immediately the cell begins to synthesize those proteins and genetic material that enable the spontaneous assembling of more viruses of the same class and that very soon will destroy the cell. This is used in medicine as phage therapy against the power of Evolution that comes here in the form of antibiotic resistance (Kurzgesagt [18] 2018). In regard to a specific mammal, tens of different types of viruses might infect its cells. For example, bats are extremely tolerable of viruses and some 50 different types infect them (Racaniello [25] 2013).

If the cell is a robot, what about its computer?

Some details of the cell supercomputer are the following: the DNA molecules serve as *read only random access memory*, and keep the genetic messages or genes. A DNA molecule is a double strand, each one of which is a string of chemical bases and that functions just as our written texts but with only 4 letters: ATCG. Genetic texts might be organized in books or rolls that we call chromosomes. These form the **Genome.** Genetic strings to be useful need a powerful machinery that decodes the verbal information they contain according to a given code. To do that, DNA molecules are transcribed to mRNA strings, and the decoding machinery groups the residues by 3, substrings that receive the name of codons, and that constitute elementary verbal commands. The information is read and executed by ribosomes which function as a multicore CPU of a computer. The products of ribosomes are chains of peptides, some of which serve as structural items and others as enzymes. Ribosomes use a table for translation, the

Genetic Code, that plays the role of an interpreter hash table. The code is written in the tRNAs, which are synthesized by the aminoacyl-tRNA synthetases. So, these enzymes are at last the carriers of the Genetic Code.

The diversity of individuals relies on the verbal nature of the genome: some set of instructions produces an individual but a slightly different set produces another one that possibly is very distinct. The verbal nature of the genome is also exploited by genetic engineering that produces localized modifications that affect specific messages.

With this, we have argued that **the Genome is software and that the cell is a robotic entity**.

The word `software` is basic in Computer Science, but we need to recognize that the software deals with algorithms, procedures or receipts, and that therefore our lives are filled in them.

4 ALGORITHMS EVERYWHERE

Humans beings continuously develop software, as for example, procedures or receipts to cook. We are developing software every time we ask ourselves: How can I do this? But we are not unique.

4.1 Animals must solve algorithmic tasks

When we plan a trip to go from home to downtown we must design an algorithm that consists of a list of instructions, some in series, when they must be executed one after the other, others in parallel, when one must consider alternate paths or transport means. This shows that algorithms are constitutive to our lives. Animals are the same. Precisely, mice have proven to excel at the task of exploring labyrinths. Does this imply that the animal brain uses neural networks to develop software? This possibility is just a permanent quest in the trade (Nahla Davies [5] 2020).

2 ***Exercise.*** *It is important to realize that the whole of our lives is made of software, of instructions to do things, of know-how. This is also valid for animals. For instance, buffalo herds are so good at the optimal design of routes that the early rails across the USA were traced following their paths in their migration. Does this imply that the animal brain can simulate genetic algorithms? In the meantime, play with the computer game developed by* `w3schools` *and that they so gently provide online (w3schools [35] 2020). The game consists in developing the software to design the route to get the award. Complexity grows softly. Pay attention to the following setting to decide when we are dealing with software: There is a set of symbols or of reserved words that form the* `programming language`*. A list of symbols or reserved words is a* `program`*. The table that assigns commands to symbols or words is the* `code of interpretation`*. An interpretation symbol by symbol or word by word is done by an* `interpreter`*. When the whole program is taken as a unity, with possible syntactic determinants or annotations, we deal with a* `compiler`*.* `Execution` *is done by a Central Processing Unit (CPU) or by a microprocessor.*

Claim. We are all very busy in the creation of software, receipts, or lists of instructions, at every moment and everywhere. Our lives are so hard because they are

filled with the execution of algorithmic tasks and/or their development. That implies that every human being is an expert in software development and henceforth he or she has the authority to judge what we are saying (if only we can be clear).

5 EVOLUTION IS A SOFTWARE DEVELOPER

Let us see now that Molecular Biology also deals with software development.

5.1 Algorithmic interpretation of the human diversity

Our DNA content is different from person to person. Thus, we all have different Genomes, a different set of encoded instructions that allow us to be what we are. Such differentiation is what enables to make paternity tests (Adams [1] 08). Even clones have different DNA than their mothers (Jiang, [13] 2011). On the other hand, differences amount to a minor fraction of the whole genome and are explained mainly by synonymous substitutions (Thaler, [34]).

Let us notice now: human diversity amounts to algorithmic development because our sets of genetic instructions, our software, have evolved, have proved to be viable, and we have been able to survive and have filled the Earth. This proves that it is a fact of Nature that **Evolution is a software developer:** from software, Evolution has produced more software. We find this lovely and marvelous.

6 THE VERBAL ET

If we consider the incredible likeness of the entrails of a hen to those of a pig, we would feel that we have the sacred duty of reinventing the Evolutionary Theory (ET) that must claim: the hen and the pig have a common ancestor. That theory was abstracted to GAs that represented a descriptive formulation of the ET. To pass from it to the conceptual quintessence of that theory is immediate. Given that the Genome and Evolution are both algorithmic, the correct formulation of the ET must be also verbal, algorithmic.

We presented that new formulation in 2010 (Rodríguez, [26]), to which we appended our reasons, most of the common sense, to think that this new stand furnishes researches with an immediate falsification of the ET. Here we repeat the formulation of the theory, the Verbal ET, plus a laboratory in GP to test it. All tests were done in JS and Flipp, a programming language to print binary strings. All programs were run over web browsers Firefox or Chrome under Ubuntu 20.04, Windows 10, and Android 10 and 11. They have been run over desktops with 8 or 16 GB in ram and cell phones with 4 MB. And with 8 or 12 nuclei at nearly 2GH, but this is less important. GPUs were not used. The time of duration of each run lasted from seconds to hours and days. One can explore diverse levels of complexity to see which is best suited for own possibilities. By augmenting complexity a bit, a program would run for weeks instead of hours.

6.1 Formulation

The **Verbal ET** is the direct completion of a transparent description of the Genome and of Natural Evolution. Let us see how it emerges from a review of our material:

- Living beings are objects that exist thanks to the execution of a software program that is written in the DNA of a germinal cell that for humans results from the fusion of a spermatozoon to an ovule. The whole program is called the **Genome**. We are saying that **the Genome is software**. But a language has more much than receipts to make things, it has poetry so, DNA also has much more than genes to make enzymes in a regulated, interactive way. For example, DNA must be compatible with housekeeping and logistics operations, a duty that poses too many demands on the DNA strands. It is with this caution that we say the Genome is software.

- At the molecular level, **Evolution consists of mutation and recombination** of DNA texts. The mutation changes texts and recombination composes new texts by concatenating parts of others. Some proportion of evolved texts are accepted by the executing cell machinery and so, these classify as new programming instructions, as new algorithms.

- Given that **The Genome is software,** and since Evolution can make changes in the genetic algorithms to produce new instructions that eventually can be beneficial, we must conclude that **Evolution is a software developer**. This is not a theory. This is a fact of Nature thanks to which we exist and are unique.

- Over this basis, **the Verbal Evolutionary Theory (Verbal ET) reads: Evolution is the software developer that is responsible for the origin of the genomes of all species.** The Verbal ET drags with it all studies and beliefs of the ordinary ET. It is an updating that is here more technical than philosophical. In particular, the Evolutionary Process is natural, automatic and materialist.

6.2 Is the Verbal ET natural?

We must ask ourselves now in loud voice:

Is our algorithmic formulation just an interesting academic point of view or is it natural, with surviving and industrial implications?

Our answer is:

In the first place, the algorithmic point of view over Evolution is the correct description of the Genetic world: the Genome is a bunch of verbal instructions to continue life. To be verbal also means that the Genetic Code looks like an arbitrary association. Nevertheless, it has been claimed that the Code optimizes very important properties (Freeland, [7] 1998). Anyway, it would be very difficult to show that some permutations inside the Genetic Code would not do it as well as the extant one.

In the second place, the algorithmic view offers a simple explanation of the fact that the pharmaceutical industry is always at the border of being defeated by the Evolution of bacteria and viruses. Why? Concretely and in regard to bacteria, we have:

With possibly one single mutation, a bacteria might cease to be a target for a given antibiotic. It acquires antibiotic resistance. Now, why does not the pharmaceutical company make a tiny change in the antibiotic to remedy the situation? Because for the company there is no tiny change: antibiotics are the fruit of a complicated chain of reactions, or, in our words, any product in the company is the result of an algorithm. Hence, to make a tiny change the company must make great correlated changes

all around in its algorithm. And this takes time, long times and very high expenses. Developing a new antibiotic costs billions.

In third place, the previous reasoning applies to the whole of life in regard to adaptation and very specially to our present time when human activity causes climate warming and wandering: some changes in the habitat can be countered by some tiny changes in some few enzymes. No problem. But this adaptive capability breaks rather soon than later because correlated changes become necessary to survive. This needs time that nobody has. The result is a permanent loss of species.

6.3 The falsifying challenge

Let us argue now that the Verbal ET is obviously false and that every person is an authorized judge to give that verdict. The proof is simple: we make many tries, and surpass many failures before success, if any. Where are the tries and failures of Evolution before it made a cat? More extended:

Verbal operations are extremely complex and always involve imperfections. This means that everyone can express him or herself but that to do that correctly and attractively is something reserved to very few persons. And even these need to work hard and long to achieve something interesting. So, perfection is reached as the summit of a process of Evolution of Perfection. By the same token, living beings are the epitome of complexity and perfection, hence if they appeared by Evolution, a mandatory prediction arises: our bodies, anatomy, and physiology, as well as the fossil record, all must have an abundant witness of the battle of Evolution with complexity in the form of every type of malformations and malfunctions. Since most babies of every species are absolutely perfect, the Verbal ET is obviously false.

Our subjective appreciations produce **the Evolutionary Falsifying challenge**: *Find the completest, fairest, clearest, and most discriminating theoretical frame that renders the Evolutionary Theory (ET) to be obviously false.*

Until here, we have proposed a solution to this challenge: it is the Verbal ET. How good is that theory as frame to falsify the ET? We need to know it.

7 TESTING THE VERBAL ET

In the Scientific Method, theories are done to be tested. The procedure is simply to formulate a prediction of the theory and to compare it with observations. If there is no great difference, another challenge is invented, and so on until the theory breaks down and the need to invent a new one arises. Eventually, a theory withstands various important tests and is adopted as a dogma or, in any case, as a highly successful theory with great predictive power. All this is just common sense, no rocket science at all except because tests could be very naive and do not catch fragrant falsity.

Let us apply this to the Verbal ET.

7.1 An incredible prediction

It is difficult to formulate important predictions. We have one here.

Immediate incredible Prediction of the Verbal ET:

If Evolution is responsible for the algorithm that describes the construction of our body, then Evolution must be useful for the synthesis of software for human use as in the Academy as in the Industry. All that is needed is a GA whose selection procedure is targeted to filter computer programs with the required characteristics. So, this is realizable and is crucial. In official terms, the Verbal ET is falsifiable, and therefore scientific.

This prediction is incredible because software development is among the hardest tasks ever invented by mankind as a whole. And, that is what we do at every moment. That is why our lives are so burdensome. Is the prediction correct?

Yes, it is.

Computing Science has thoroughly worked out this test and the result is a challenging discipline that is called **Genetic Programming (GP)**, the synthesis of software by means of Evolution (John Koza et al. [17] 2005, whose first works were published in 1988). It brings forth year after year impressive code that successfully competes against human-made programs (Genetic Programming [8] 2021). This is an extremely difficult and expensive discipline but an understandable introduction for the Java and Clojure languages is available (Rodríguez [27] 2017).

7.2 High quality and astonishing complexity as test

Biological beings from viruses to humans are all astonishingly complex and incredibly perfect. This is naturally done into a test.

The prediction of the Glorious GP:

If Evolution is responsible for the algorithm that describes the construction of our bodies, which are so complex and so functional, then Evolution must be useful for the synthesis of very complex and high-quality software for human use.

The test asks about the worth of the programs made by GP. In this regard, one success of GP that I admire is the programming of a quantum computer for a certain task (Spector et al.[32] 1998). That is why we with joy declare:

Verdict: The ET withstands the test of the Glorious GP: the glory of Applied Evolution is in general that of your imagination.

The natural reaction to this impressive success is to unconditionally conclude that the ET is the correct explanation of our existence. Nevertheless, a liar always says many truths. That is why to pass a sharp test does not mean that a theory is correct. More tests and more insights are needed, and when to stop is well known: never. That is why we go further.

7.3 Ubiquitous and total perfection as a test

I was invited to visit a mother and her newborn daughter. The mother allowed me to hold the baby and with joy, I admired her tiny fingers and nails and I was so stumbled with her perfection that I exclaimed: *She is perfect!* When the baby listened to me, she opened her deep blue eyes, looked at me for one second, and closed them again. Next, we altogether thanked the Lord for that miracle.

Wherever I turn my eyes in the living kingdom, I see perfection. And if I look into fossils, say, of various types of dinosaurs, they also look perfect.

Subjective as my appreciations are, they have programmed my mind to reject the ET with all my heart. And the great glory of Science has not been able to stagger my convictions. Why? Because Science is wrong: the perfection I see comes out of the blue, but the complexity of life demands from the ET to predict the Evolution of Perfection. But this is not found. To see why, it suffices to look at our own light:

The design of our face is very attractive: 2 eyes, 2 eyebrows, 2 nostrils, one mouth. To explain this, the Verbal ET argues that it was originated by Evolution: the pattern of our face is just the same as in all mammals, birds, dinosaurs. So, its origin and fixation should have occurred a long time ago. But, where are the tracks of that fixation? Sharks have tissues on the lateral part of their bodies that can taste possible prays: where are the ancestors of mammals with tasting tissues in the arms? Many spiders have 4 pairs of eyes (different than having 8 eyes): where are our ancestors with 2 pairs of eyes, one in the front and one in the nape? Do you consider that our questions are silly? In that case, think of the disposition of the various items in the face. Did it appear by randomness and never evolved? No? Was it fixed by Evolution that awards the reproduction of the fittest? Yes? Where are the ancestors with the other variations, say, with eyes in the front and the nostrils in the chin? Where is something in Nature that may convince me that the pattern of my face was the product of Evolution?

Nowhere.

Now, you have only one possible riposte:

The Evolutionary tree connects archea to the human being. Thus Perfection also evolved because the perfection of archea is not the same as that of complex organisms. Now, in regard to a given species, the Evolution of Perfection indeed happened but no fossil record exists because of three reasons: most changes leave no fossil record, sampling effects, and silent Evolution in junk DNA or in duplicated material.

Our decision is to demand to be distrustful of this and of every justification of the absence of the Evolution of Perfection: The number of species is above 7 and no fossil record of the Evolution of their Perfection has ever been found. Is trustworthy to say that they appeared by Evolution?

Now, we refuse to feed our suspicion with more rhetoric. We already have enough. Instead, we must provide experimental research. So, let us turn to GP to see how the Evolution of Perfection looks like.

Objection. Everyone is sick of the term Perfection that might mean anything. Can somebody provide an operational definition?

3 ***Exercise.*** *Resolve the objection.* *Answer*

Let us turn now to experimental GP. The problem is that it has been an impenetrable mystery for almost everybody.

Anymore.

In what follows, we will propose a game for children that will allow us to make GP to see the battle of Evolution against complexity in regard to the development of software. But, in the midst of so many details, it is important to keep in mind what are we doing: we are incorporating quantitative studies to rule out some lies. We do this all the time. Now, since the information in our DNA is similar to the richness, the following popular example will help us to fix ideas.

Example: Joe used to normally earn a living, but in the last days he appeared with expensive cars and people that served him, some dressed in fine black. He said that all this is the fruit of his work and wisdom. But since he is suddenly squandering big money, the Authorities accuse him of washing money. He denies every charge and does not accept the evidence. In the discussion, they make a prediction: if you are clean, you must have enough written documents as well as persons ready to testify. No documents about big money were found and that is why they called Peter and James. Peter said: I have transported goods for him from the factory to his stores for 20 years. James also said the same. After some calculations, the Authorities replied: Where are the other 1200 employees working during 20 years that were needed to assure net gains as big as those you report? Joe was unable to say something coherent and that is why he was condemned. He has endured 9 years in jail and soon will end his service.

To start with our accounting task, we will introduce a very simple computer together with a language to operate it. Your children will find them very amusing. And, it will be sufficient for our GP tests in which we confront Evolution to complexity.

8 THE FLIPP COMPUTER

Program `FlippComputer.html` shows us a `Flipp computer` with various peripherals:

- A `Flip Computer` has one memory cell that can keep a bit, a 0 else a 1. To start or after clearing, the cell contains the value 0.
- `Flipp` is a programming language for a Flip Computer. Flipp is a shorthand for flip and print.
- A `Flipp computer` is a Flip machine that is programmed with Flipp.
- The value of the memory cell is modified by the commands of Flipp, which are only two. Use the `dot .` to print the content of the cell to the OUTPUT text area. Use the `exclamation mark !` to flip and print: Flipping turns 0 into 1, and 1 into 0.
- Typed commands form a program that appears in the PROGRAM text area.
- Use this computer to design programs to print specific binary texts. This is the task of every program on every computer.
- Train yourself by looking at what the commands do. When you get ready, take a test: If you check the CHALLENGE checkBox, a binary string will appear to bold you: Your mission, if you accept it, is to develop the Flipp program that reproduces the given string. To end your programming, click the EXECUTE PROGRAM button. You must pass 7 levels.

- Would you consider that programming in Flipp is easy? Would you recommend a metric to support your answer?

Example. Some passwords and their generating Flipp programs:

```
Password = 0
Flipp Pr = .

Password = 1
Flipp Pr = !

Password = 0000
Flipp Pr = ....

Password = 11111
Flipp Pr = !....

Password = 000110001001
Flipp Pr = ...!.!..!!.!

Password = 1000110001001
Flipp Pr = !!..!.!..!!.!
```

4 ***Exercise.*** *Show with a diagram that Flipp is just another implementation of the general verbal communication scheme. Answer*

Note. There is only one Flipp program that can output a given string. Such a program can be calculated automatically as program `FlippMaker.html` shows. The `Exclamation Mark Operator` can be called the `Jump Operator` and the `Dot` the `Identity Operator`. The Flipp program can be understood as a `symbolic discrete derivative` of the posed string.

9 THE COST OF PERFECTION

When a developer is contacted, the client says: `I want a program that under these circumstances output this and under those others that.` The written memory of the transaction is consigned in a document that is called `Spec`. Our preoccupation here deals with a spec problem and the cost of finding a perfect solution.

9.1 SPEC as an abstraction

Can Evolution deal with the complexity of the design of living beings? In the frame of Genetics, this problem reads: Can Evolution specify the DNA of our cells? It is further abstracted here in the frame of the Flipp Verbal ET as follows:

SPEC, the Spec Binary Problem in Flipp GP:
Can Evolution match a random binary string with the output of a program in Flipp?

5 ***Exercise.*** *Would it be not enough to restrict ourselves to matching binary strings with binary strings without mentioning any program at all?* Answer

We see that no condition is referred to in the formulation of SPEC so, our problem is indeed simple, possibly the simplest one. Can Flipp-GP solve it? Of course, it can. But, at what cost? This question is important for us because, in regard to combinatorics, the computing resources of the Universe as a whole are fantastically low.

6 ***Exercise.*** *Prove else refute: Flipp is the simplest language that outputs binary strings at the lowest cost. So, the estimations we will find are the cheapest ones.* Answer

9.2 The cost of SPEC

To calculate the cost of SPEC, we adopt the methodology used in Mathematics and Computer Science for the calculation of the algorithmic complexity of a task (Arora and Barak, [2] 2007). The difference is that we do not use mathematical calculations but simulations instead. A **challenge for everybody** *is to use our estimations to build robust mathematical models.*

This is what we will do: We give a binary string of length x, match it with Flipp GP, measure the cost in individuals as an exponential function in x, and pose a new random string of length $x+1$. In our experiments, we begin with length 1 and we go on until we see a stable tendency in the exponential expression. The cost rapidly grows as x increases so, we are unable to reach a large number. In most cases, the number 30 is already unbearable, but exponential expressions stabilize around 15.

What are we doing? We are pitting Evolution against complexity because as the length x of a string increases, so does its information content, and also the difficulty to match it.

Program `GPwithFlipp.html` allows us to estimate the cost of matching a binary string of length x as $49.7e^{0.3x}$. This estimation resulted from solving SPEC in the interval $[1,30)$, where 30 was not included. For extrapolations, the exponent is what matters, unless compared expressions have the same exponent. The extrapolated cost of Evolution was less than that of randomness, which in its turn was less than that of the searching worst case.

Our results allow us to claim: The cost of SPEC is exponential in the length of the input, where the length of the input is x.

Objection: The procedure to extrapolate the cost as an exponential function was obtained thanks to a finite number of points. But the natural fitting in such a case is a polynomial so, an exponential dependence is left without a place here.

7 ***Exercise.*** *Resolve the objection.* Answer

8 ***Exercise.*** *Compare the cost of SPEC on Flipp GP with that of GAs.* Answer

Challenge. For a human being, it is immediate to design a Flipp program whose output matches a given binary string. Nevertheless, for Evolution, it has an exponential

cost in the length of the input. This disequilibrium says that Evolution must not be run in the space of programs but in the space of AI systems. Implement this.

Challenge. In the XVIII-th Volume of *Java for the Study of Evolution*, a program showed that one can arrange things to increase in one the number of matched sites per generation. This seems to contradict our present results. Clearly state the contradiction in quantitative terms and resolve it.

9.3 The evolutionary arisen of genes

The exponential cost of SPEC is readily made into a falsification of the Verbal ET:

Claim. According to our simulations, Evolution needs an exponential number of individuals to produce a Flipp program whose output matches a given binary string. Therefore, Evolution cannot be the cause of our existence because of its definitive lack of resources. In fact, the exponential cost of SPEC means that there is no universe with a finite velocity of information transfer in which Evolution could specify our DNA. For instance, the innocent cost $49.7e^{0.3x}$ evaluated for $x = 50$ renders 162 millions and for $x = 500$ more than 10^{68}. This implies that the computer power of the whole of mankind is null in comparison with what is needed to simulate GP in the long-range. Actually, our troubles use to begin around $x = 25$. Nevertheless, the program `GPwithFlipp.html` also shows that Evolution is better than randomness, which is much better than the searching worst case. So, Evolution can be useful to GP more than those and by looking to the appropriate problems, GP would continue to surprise us. In general, we preach: Your imagination is the only bound to the power of Applied Evolution.

9 ***Exercise.*** *Find an evolutionary exit to the claimed falsification of the Vebal ET.* *Answer*

We have the suspicion that the exponential cost, more than being a rejection of the Verbal ET, is one more of its nuances that makes it interesting. We arrive at the conclusion that we must explore other insights. Let us present a very natural one: the exponential cost of Perfection is no problem, precisely because Evolution makes the best of Imperfection.

10 THE COST OF IMPERFECTION

From our experience with GAs, we know that Evolution also must pay an exponential cost to find approximated solutions. Which is the case here?

10.1 Our simulations

Program `FlippCostOfApprox.html` shows that the cost of imperfection behaves as shown in Table 1.

We say simply: Approximating with the output of Flipp programs the 70% of a binary string has an exponential cost in the length of the input with exponent 0.06.

In all cases, the extrapolated cost of Evolution was less than that of randomness, which in its turn was less than that of the searching worst case.

Table 1: Exponential cost of approximations.

Degree	Evolution	Randomness	Worse case	Range	Duration
0.70	$30.7e^{0.06x}$	$2.3e^{0.09x}$	$1.4e^{0.49x}$	[1-150)	7 min
0.80	$44.7e^{0.1x}$	$2.3e^{0.21x}$	$1.3e^{0.55x}$	[1-75)	12min
0.90	$28e^{0.22x}$	$1.8e^{0.41x}$	$1.3e^{0.63x}$	[1-50)	13 hours
0.95	$81.1e^{0.22x}$	$3.8e^{0.5x}$	$1.2e^{0.66x}$	[1-30]	1 minute
0.95	$58.8e^{0.25x}$	$3.3e^{0.51x}$	$1.2e^{0.67x}$	[1-37]	31 hours
0.95	$47.5e^{0.27x}$	$2.9e^{0.51x}$	$1.2e^{0.67x}$	[1-38]	48 hours
0.95	$41.8e^{0.28x}$	$2.9e^{0.52x}$	$1.2e^{0.67x}$	[1-40)	54 hours
1.00	$49.7e^{0.3x}$	$0.2e^{0.74x}$	$1e^{0.69x}$	[1, 30)	20min
1.00	$47.5e^{0.28x}$	$0.3e^{0.74x}$	$1e^{0.69x}$	[1, 30)	55min
1.00	$35e^{0.3x}$	$0.6e^{0.71x}$	$1e^{0.69x}$	[1, 30)	30min

Program `FlippCostOfApprox.html` *predicts the exponential dependence of the cost of approximating a random binary string of length x according to Flipp GP, Randomness, and the Worst Case Searching. This table must be verbalized as follows: For the approximation degree* 0.70*, the expected averaged cost of finding by Evolution the first Flipp program that matches a string of length x is* $30.7e^{0.06x}$ *that was established in the range [1-150). The last column reports the needed time in a desktop computer that exclusively runs the program. Repetition causes as the coefficient as the exponent to vary such as is shown by the last three rows (obtained with program* `GPwithFlipp.html`*). But, variations are the less the wider is the range of the researched range of password lengths.*

If one repeats the simulation, a tendency to stability is observed but to slightly shifted values. This might eventually mean that to achieve stability and convergence to the unique limit, wider ranges are needed. The option of fluctuations that never die is correct but their magnitude diminishes as the range is increased. That is why stability is expected.

Overall conclusion. The cost of matching SPEC is exponential in the length of the input as for Perfection as for Imperfection. The degree of approximation is naturally reflected in the exponent.

10 ***Exercise.*** *Propose some implications of our conclusion in regard to Natural Evolution.* Answer

11 ***Exercise.*** *Our simulations say that the cost of both Perfection and Imperfection in Flipp GP is exponential in the length of the input. Would you add the calculation of confidence intervals to make that assertion more serious?* Answer

Thus far, we have made a good deal of progress. Is there any complaint?

11 THE CONUNDRUM OF TELEONOMY

The following is a terrible criticism of our work until here:

General Objection. All of our GP tests are more for Lamarckism than for Darwinism.

In fact, Lamarck thought that matter is dressed in an alchemist progressive force for Evolution, while Darwin proposed that progress is a byproduct of the struggle for life. Now, in our simulations we followed Mendel, De Vries, and Lamarck:

We followed Mendel because we used discrete inheritable items. We followed de Vries because our mutation was at random. We followed Lamarck because all our programs have a goal that is both imposed from the outside and enforced at every step in reproduction. So, we permanently compare evolved strings with that declared as the target. Thus, our GP is indeed Directed GP because always assumes a purpose.

By contrast, according to the Verbal ET, which is a materialistic extension of the ET, Natural Evolution has no goal at all. The force of this doctrine resides in the great idea of Darwin that the struggle for life in the midst of siblings singles out the fittest that leaves more offspring. This process leads to perfection when it is iterated generation over generation.

Challenge. Enjoy the stupendous simulation of Evolution that was evolved by Ingo Berg ([12] 2019) and that runs automatically online. Decide where it follows Mendel, de Vries, Lamarck, or Darwin. The code can be downloaded for detailed examination. You might ask yourself, for instance: to what degree is the selection procedure defined by hand rather than by self-organization, by an emerging property of the environment-organism relationship?

The situation when a predefined goal is actively pursued from the exterior will be referred to as **teleonomy** (compare Henning [10] 2011). Thus, we have a question: How will we simulate Evolution without teleonomy?

We feel that the question of teleonomy is a conundrum that never will be sufficiently enlightened by simulations because of the eternal lack of computing power. Therefore, we must resort to indirect reasoning. Our first point is an invitation to look in hindsight:

The problem of surviving is the only important problem in Darwinism, which has possibly myriads of correct solutions. Nevertheless, we can observe that in general, the success in the struggle for life can be deconstructed, for instance, as follows: good mobility, high intelligence to understand the environment, good profiting of resources, victorious behavior with siblings, excellent approach with mates. All this demands appropriate constructive information. Our point is that this information must be encoded in the DNA, be it directly or somehow indirectly. And when it is, it is just a string that is as frozen as the ice in the eyes of an uncomprehended person, which anyway conserves some portion of life or variation. This seems to imply that the cost of matching a string is the same in both cases, with or without teleonomy. Is that indeed really so?

12 ***Exercise.*** *We know that a degree of approximation in Directed GP has an exponential cost with consequently growing exponents. Show that the corresponding cost in Self-Organized Evolution, without teleonomy, cannot be lesser.* *Answer*

Objection. Recombination with strange experiments could produce solutions faster than recombination with more of the same. So, the cost under self-organization might be lower than in Directed GP.

13 ***Exercise.*** *Deal with the previous objection.* *Answer*

We have reasoned guided by common sense that seems rooted in Darwinian gradualism when population size is rather large. Is that all to the Evolution of Software?

12 BUGS

Some 15 years ago I arrived at the conclusion that to falsify the ET, one needs the Scientific Method. And, that moreover, simulations were the easiest way to calculate predictions. At that time Java became open source, and so, I decided to learn Java to program simulations of Evolution. In that way, after some 10 years of not programming, I began to work and I got astounded at the immense quantity of errors that I committed. At the same time, I also realized that the Genome is software. In the trade, errors in programming are called **bugs**. That is why I found myself asking: **If Evolution created me, where are its bugs?** For an unknown reason and since my twenties, I use to look at my left hand when I ask myself things like that. I open and close it and I perceive how glad I feel with it. And from my heart, a glaring flame arises that burns any desire of believing that I am here because of Evolution. And a desire of glorifying God my Creator flourishes in my mind and soul.

I am aware that God the Creator is not a matter of Science, i.e., of experiments with the Scientific Method so, I classify myself as irrational. But, What about the ET? The Scientific Method demands to calculate predictions. To do that I implemented simulations in Java. Nevertheless, I noticed that running programs in Java was impossible for most people. Then, I learned to program in JavaScript that everyone could run programs with one click if one unzips them. In fact, JavaScript programs can be opened with any browser and run immediately: over a new tab in your browser, tap `Ctrl + O`, select your program, and open it. Otherwise, one must revise the configuration of the browser to enable JS programs to run.

Our task now is to transform a subjective experience with bugs of the rejection of the ET into something objective and transparent that could be fairly simulated. To begin with, the concept of bug is irremediably tied to teleonomy, because bugs are errors that are done against a target. In the second place, we prefer operational questions:

Why are bugs so frequent?

That depends on both the developer and the nature of the Software. We forget the developer because we seek for easiness to detect objectiveness. In regard to Software, the cause is that it is discontinuous, i.e., it is not amenable to gradualism, to gathering small progress. In fact, since Software is verbal, it cannot obey gradualism. Consider, for instance, the following expression that is valid in most high level languages:

$$s = g(t)$$

Since t can be anything processable by function g, one can vary t to obtain possibly very diverse, dissimilar outputs. So, gradualism is not the rule in GP. How is the situation with Flipp? Let us study the outputs of random walks in the space of programs. Random walks are the very epitome of action without a goal.

12.1 Random walks in Flipp GP

A **random walk in GP** starts with the generation of a program at random, which is mutated recurrently to study how the output changes after each mutation at random. Now, what do we expect in a random walk?

Flipp admits mutations that produce soft changes.

Example. The program ..... prints 000000000 that is interpreted as the number 0 and the mutant .....! produces 00000001 than in decimal interpretation is 1. The mutant is as close to its parent as possible.

Nevertheless, abrupt changes are also possible:

Example. The program ..... prints 000000000 that is interpreted as the number 0 and the mutant !..... produces 111111111. If we add 1 to this number, we get 100000000 that in decimal notation is 2^9. So the mutant is $2^9 - 1$. We see that one mutation is enough to cause the largest possible change. Thus, Flipp is discontinuous by its very nature.

What does happen in general in random walks?

Program `FlippGPRndmWalk.html` presents a systematic study of the changes along random walks. We conclude that jumps of all sizes are the normal dynamics in random walks of Flipp programs.

Now, What effects does the discontinuity of Flipp have over ordinary GP?

12.2 The dynamics in Flipp GP

Program `FlippDynamics.html` shows us how GP deals along time with the problem of matching a password that represents a chessboard. A square in gray is represented by 0, and one in black by 1. So, a board is represented by a binary string. The program traces the dynamics through the report of the best fitness of each generation. We see no needles here because we pay attention not to all kinds of jumps but only to those in the best fitness from one generation to another. The program says us that the dynamics in GP are very similar to those of GAs: after an initial value that is produced by randomness and that clearly classifies as mediocre, the best solution ascends a bit forming a shoulder that levels itself into an equilibrium value. Let us remark that that value is in the fitness, but the corresponding strings could have fluid dynamics. The reason is that many strings can have the same degree of matching of the chosen target. Fluctuations exist and are wider than in GAs. These are larger for low complexity but diminish as the length of passwords increases.

We have now a question:

Is the extrapolation of the exponential cost correct? That extrapolation assumes that GP can indeed get to Perfection. But here we see that Evolution creates an equilibrium value rather close to mediocrity and that fluctuations departing from it are enough to solve problems of small complexity but when this increases, the size of the fluctua-

tions diminishes. So, from this perspective, it is not at all clear whether Perfection is reachable in Flipp GP.

Our interpretation is that fluctuations never die and that these have no mystery, neither to create miracles nor to impede their appearance. Rather, they obey the statistical laws and here we would say that the law of large numbers is in action. So, we believe that Natural Evolution can indeed attain Perfection although its cost is exponential in the length of the input, such as it was extrapolated from a previous program.

Let us notice that the equilibrium value is stable: if a fluctuation rises up, its decaying is the most probable event. And, if a fluctuation goes down, Evolution rises the fitness until leveling. That is why we identify the stable state with the effect of a deterministic process, that receives the name of Selection. Therefore, we are inferring that selection is in general an anti-evolutionary force, while fluctuations can be both reactionary or progressive. Nevertheless, continue reading.

Challenge. We have said that the equilibrium value is stable because we observe that fluctuations dance around it. Would you do a direct proof of this by programming suitable initial conditions and following their further Evolution?

Challenge. Which is at last the cause of stagnation? Let us suggest that mutations with slight effects are responsible for the deterministic component of GP, which is stagnating and frustrating in terms of perfection. Prove or refute this.

Challenge. Which is the cause of reaching success in Flipp GP? Prove or refute: GP reaches success thanks to fluctuations but not alone, rather fluctuations and selection work together somehow to accelerate Evolution to success. We think this because GP is faster than randomness. That is why GP has a secure place in Science and Engineering.

14 ***Exercise.*** *When GP battles for matching a password, it finds partial matches, first of length 1, then of length 2, and so on until at last, it makes a bull's eye. Is this a constructive trend in which GP uses achieved results? Or, is it destructive in which GP cannot use achieved results? If one graphics the cost of reaching the partial matching of length k, what would be the functional expression? Make a guess and test it.* *Answer*

12.3 The tapestry of perfection

The discussion about the role of fluctuations and of the stagnating force of selection around the equilibrium state is for us fascinating. With the hope of understanding a bit better that interaction, the program `FlippEvoOfPerfection.html` depicts the Evolution of Perfection in Flipp GP in battle against SPEC.

The program presents us 3 graphics that show the functional expression of the cost of reaching the partial matching of length k when GP solves a given password. In the second place, it shows us a tapestry of tiny, colored squares. The color represents the fitness of the corresponding Flipp program. Each row represents a generation, which is numbered. Fitness increases from dark green (zeroth class) to light green (eighth class), then to violet which represents Quasi-Perfection, and then to red, which is Perfection. The perfect Genome is joined by two segments to its parents, one just below it and the other somewhere in the inferior row. Every column denotes direct descent.

We observe that almost all generations have light green colored squares, meaning that for GP it is quite easy to build good quality. Quasi-Perfection, in violet, is rarely seen. So, how is Perfection fabricated? The red square almost never comes from violet ones but more usually comes from parents that output rather incipient solutions.

We conclude that the tapestry shows us that Evolution causes Flipp programs to produce outputs whose fitness preferentially changes by jumps.

Objection. The results of the program `FlippEvoOfPerfection.html` have been used to claim that the discontinuity of Flipp GP is due to the discontinuous nature of Software. But this is false: it is recombination that is discontinuous. Precisely, recombination is the cause that GA's, which contains no software, can escape from local, stagnating maxima.

15 ***Exercise.*** *Resolve the objection.* *Answer*

Our conclusion is that in Directed Flipp GP, Perfection is preceded by long runs of under-perfection. We extend this conclusion to Natural Evolution because of the emerging of a Self-Organized Selection, and thus we get another falsification of the Verbal ET: Before perfection, imperfection must be abundantly found in the fossil record. Since this is false, the Verbal ET has been falsified.

16 ***Exercise.*** *Show that the Verbal ET predicts that the effects of bugs must be abundantly visible in present populations. Compare the prediction with what you observe in your own body. Decide whether the Verbal ET has been falsified.* *Answer*

13 CONTEXTUALIZATION

How is our falsification of the ET related to our world and lives? Let us see:

13.1 Relation with creationism

Our work is a scientific falsification of the ET. From this, it cannot be deduced that God created us. Nevertheless, if someone has faith that God created us and that He will judge every hidden thought we have, Science cannot use Evolution to reject that belief.

13.2 Relation with Science

Materialist Philosophy forces Science to affirm eternally against Christian believers: `It is impossible to detect your soul, hence it does not exist, and all your beliefs are flawed`. Defend not yourself, rather enjoy your faith and treat persons of lower rank as equal to you because they are also the image of God. And remember that your enemies have a permanent battle against depression.

At the operational level, we have a change of paradigm:

"Nothing in Biology Makes Sense Except in the Light of Evolution". This is what everyone accepted until now. Everything was explained in Science by Evolution, and for each instance of speciation, to get done you just needed to invent a true-like mechanism.

But we have here a falsification of the ET. What will be next?

The falsifying of the ET is similar to the induced explosion of a meteorite because to save our lives we must care afterward of its debris. In the same way, the approach to evolutionary hypotheses must change. The dictum that illuminates our work is from now on:

Nothing in Biology Makes Sense Except in the Light of Complexity. Thus, no relation between species can be explained by Evolution unless you can prove it by means of the Scientific Method, whose first requirement is replicability. There is no excuse for Science because the number of species of microorganisms ranges in the order of millions or, maybe trillions? (Kenneth et al. [15] 2016).

We can put everything in a slogan: **The ET is dead, long live the ET!** Since the ET is dead, the project of Evolution as a Cosmogony fades away. This means that each cosmological theory that uses Evolution as a fundamental building block is also dead. Thus, you are invited to inject new life to Cosmology by inventing a fresh start.

Our demands on speciation are extremely difficult and very expensive in every sense. Possibly the whole humanity must joint efforts to make a single point beyond the easy ones, say, the origin of some species of flies.

17 ***Exercise.*** *Propose concrete realizations of the Scientific Method that test the hypothesis that two given species are related by descent.* *Answer*

13.3 Relation with persons

Human beings love dogmas. But one might decide which are his or her dogmas and which beliefs are questionable. Precisely, that makes him or her human because this is always done on irrational grounds, say, it is part of own religion, culture, or family inheritance. We have exhibited here our irrationality in making the ET a questionable theory that has proved to be false. Nevertheless, let us remind everybody that Evolution is everywhere and that our imagination is the higher barrier for a powerful application of Evolution.

Every person has now a question: if Evolution is not the cause of my existence, what is it, and what does it want from me?

13.4 Jesus comes!

This work obeys the command of the Lord given through Prophet Isaiah: " Prepare ye the way of the LORD, make straight in the desert a highway for our God."

14 CONCLUSION

We have completely specified the `Evolutionary Theory (ET)` by formulating the `Verbal ET`, which is mechanistic, modern, final, transparent, simple, and clear. First: *the genome is software*. It is a fact of Nature. Second: *Evolution is a software developer*. It is a fact of Nature and of engineering as Genetic Programming (GP), which is the development of software by means of Evolution. Third, the `Verbal ET` reads: *Evolution is the sole software developer that is responsible for the creation of all species on Earth.* To test that theory with the **Scientific Method**, we have built the

simplest testbed: `Flipp GP`, where `Flipp` is a programming language specifically suited to print binary strings. It happens that `Evolution` is beaten by its natural enemy, `Complexity`. That is, the `Verbal ET` predicts mandatorily that an **Evolution of Perfection must exist and that under-perfection must be immanent to Evolution**. Operationally, this means that malformations must be abundant as in the fossil record as in our own bodies, and malfunctions must fill our own physiology. No such thing is found in our species. Thus, the `ET` is false, it is dead as a dogma, but it lives as a scientific inquiry: *The ET is dead, long live the ET!*

ANSWERS

Exercise 1, page 8. The Genetic Code fits the general structure of verbal communication, such as it is shown in Figure 2.

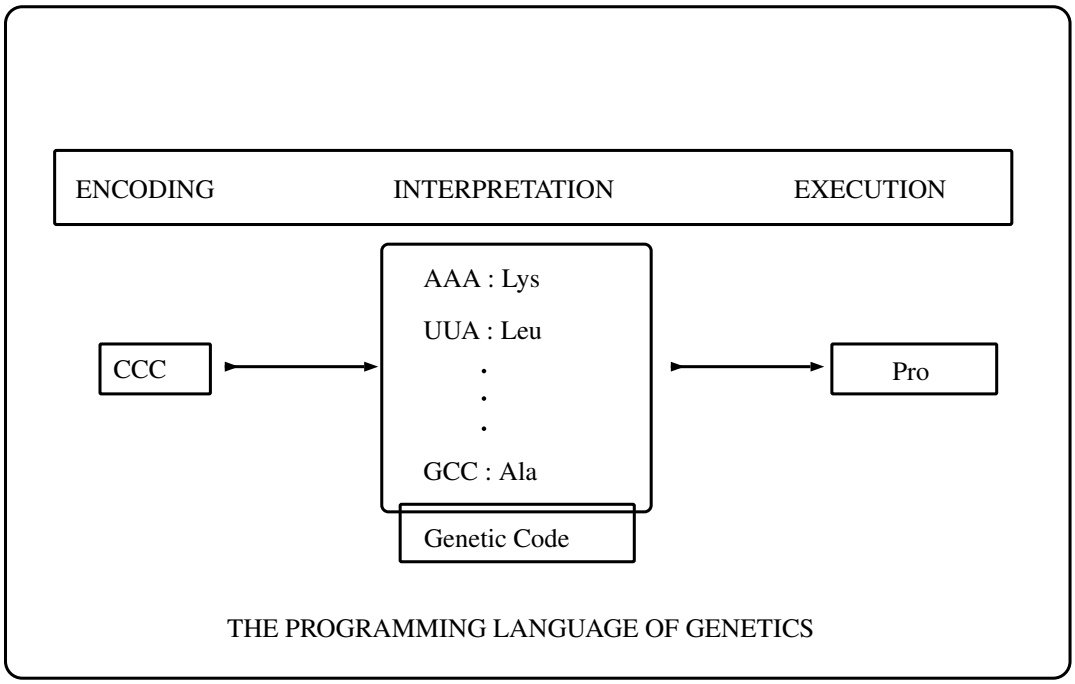


Figure 2: *DNA contains codons that are triplets of symbols. Each triplet is a verbal instruction that is interpreted according to the Genetic Code, and the instruction is executed by a ribosome, in the cytoplasm, that adds the corresponding amino acid to a polypeptide chain.*

Exercise 3, page 14. It seems to me that Perfection is something ethereal, i.e., a mix of art and design, of complexity and function. So, we will work with an operational simplification: perfection is the exact matching of a set of instructions and restrictions. Since every text can always be encoded in binary format, we end with a definition of Perfection as the exact matching of a binary string. In GP, the string must be the product of a program. This will appear explicitly in our codes, where the string to be matched can be anyone.

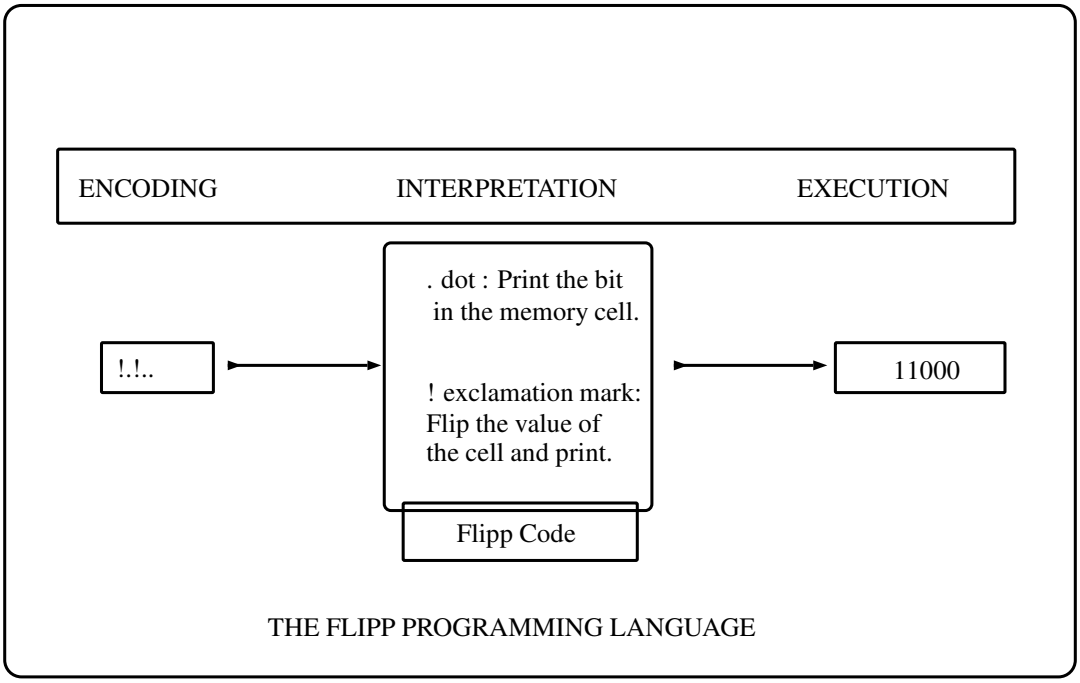


Figure 3: *Flipp is a programming language with two commands, dot, and the exclamation mark. Dot . means:* `print the content of the memory cell`*. The exclamation mark ! means:* `flip the value of the memory cell and print`*.*

Exercise 4, page 16. The diagram in figure 3 shows that Flipp allows making programs that fit into the general scheme of Verbal communication.

Exercise 5, page 17. DNA is not selected by itself but by its fruits, i.e., by the phenotype. As a first and possible crude approximation, the phenotype of a structural gene could be the enzyme or polypeptide that it encodes for. In our abstraction, the phenotype of a Flipp program is a binary string. So, the score of a Flipp program depends on the string it produces.

Exercise 6, page 17. Flipp is not the simplest language that can output binary strings of whatever content. The simplest one is the language that takes the very binary string and immediately interprets it as a program. Thus, suppose that the string is 0101. That language, the `Identity Language`, would say: the first 0 is a command that means `Print a zero`. The following 1 means `Print a 1`. The second 0 means `Append a zero`. The following 1 means `Print a 1`. And so on. Nevertheless, both languages, Identity and Flipp, pay the same cost on the same string, which coincides with the length of the string. Thus, for the string 0101, both need the execution of 4 instructions.

Exercise 7, page 17. We calculate in our simulations the exponential regression for all lengths of passwords, one after another in the considered range, and observe the

tendency: if after initial fluctuations the exponential dependence tends to get stable, the exponential claim is robust against polynomial pretensions. Otherwise, it is weak. This criterion is due to the fact that no polynomial can fit an exponential in the long range, instead, the degree of the polynomial shall increase as more points are incorporated. By contrast, our exponential regressions tend to stabilize.

Exercise 8, page 17. Program `GAcost.html` shows that the cost of SPEC for GAs is also exponential but by far lesser than that of GP, given by program `GPwithFlipp.html`. Why? Our opinion is that it is because the Flipp Exclamation command creates interference. In fact, it flips the memory value and that changes what the dot command prints next. So, it is more difficult to separate chars in GP than in GAs.

Exercise 9, page 18. The exponential result has been made by the aforementioned claim into a `no-go theorem for Evolution`. This contrasts with what Science shows us every day: Evolution is a powerful tool as for design as for finding approximate solutions to optimization problems. So, the stand of a no-go theorem is inappropriate: it tends to blind our imagination. More to the point, let us consider the following scenario: Evolution can match short strings but cannot match large ones exactly. The Evolutionary solution to this tension is simple: small chunks of DNA will acquire function and evolve consequently into the modern genes of present organisms. The complexity analysis of this idea goes as follows: While binary strings, as well as the Flipp language, have both an alphabet with two symbols, DNA has four. But once we consider genes, we enter into another language with an unbounded number of symbols. In an Evolutionary Environment, shorter genes might be preferred. So, we might consider an alphabet with a finite but large number of genes that eventually fill in different functions. With this new alphabet, we again find the problem of matching a given high order function, say surviving, and we expect an exponential cost, just as before. Thus, the arisen of genes is not a guaranteed solution to the falsification of the ET forwarded by the exponential cost of GP. But a doubt remains.

Exercise 10, page 19. We find the same tension as before: it is too costly to develop imperfect long binary strings so, half-cooked genes will arise that in random combinations will co-evolve into great functional units. And, since the cost of ascending the ladder of Perfection is exponential, Imperfection is mandatory: it must be present both as in the fossil record as in my own body. This is false. The perfection of skeletons of recently dead dogs awakes in me so a great admiration as those of dinosaurs and of fossils of leaves in the rock. And in regard with me, being aware of all my defects, I declare myself perfect: I continuously wonder at my heart that still works and works and works after 67 years of ruthless service. Simply marvelous!

Exercise 11, page 19. Confidence intervals predict observations upon repetitions and beyond actual data. The narrower they are, the more compelling are the estimations. It happens that confidence intervals are the wider the farther are from the range of registered data. Precisely, our purpose here is to extrapolate values from the range [1, 30] to the realm of billions (our DNA contains 3 billion bases). So, confidence intervals make no sense for that extrapolation because they will be too wide to be of any interest. Even the simplest assertion of an exponential cost would be deprived of value by some purists. Then, we ask ourselves: Is our work useless? No, it is not. In the first place, criticisms would be important in the only case in which the cost of Binary GP results to be lower than exponential, i.e., polynomial of low degree. Such a

result would propel a revolution in the software industry and most engineers would be left unemployed. But this is unexpected by most people even if quantum computing is employed. In the second place, our results would guide the formulation of mathematical models that would produce by themselves lower bounds of the cost for any range. Daunting as this task could be, program your mind with the conviction that you can do something serious. And let yourself give it a try. Various cues are provided along with this work.

Exercise 12, page 20. When Evolution goes on without external goals, all possible functions begin to appear until some few start to be perfect enough to generate individuals that are fitter than the rest. It is like this that a directional selective force emerges and guides the Evolutionary Process. And while they evolve into Perfection, anyway, too many other parallel experiments are running. Thus, Self-Organized Evolution is by far more expensive than the Directed one, which awards the fittest. But, if we look carefully, our discussion shows that Evolution without teleonomy is not good for producing specific strings but that it is good in the quest for complex organism. The problem is that one would need more than 7 eons to see something interesting.

Exercise 13, page 21. The Scientific Culture recognizes that graceful surprises might come from that world that is not the focus of our attention. A name has been given to this situation: serendipity. But, it is clear that Universities do not expect such surprises to give them status. Rather they work orderly over research programs with a clear objective. This invites me to believe that the cost of Evolution by pure serendipity is tremendously expensive in comparison with Directed Evolution and GP. So, the cost of both Perfection and Imperfection in aimless Evolution is anyway exponential in the length of the input. This implies that in Darwinism, which imposes no goal, mediocrity and Imperfection must abound to leave its place to eternal under-perfection until the eventual arriving of Perfection. This is not found in Nature.

Exercise 14, page 23. Given that GP is the Evolution of software, we expect discontinuities to play a dominant role and therefore the trend to perfection is destructive. On the other hand, our guess on the cost of reaching the partial matching of length k is that it is exponential and with exactly the same expression as the cost of matching a password of length k found previously. We test our belief with a program that appears a bit later.

Exercise 15, page 24. The theme is difficult. To say the least, GAs also contain software: when one encodes the solutions to a problem as strings, evolves them, and decodes them into solutions, we use translation rules, i.e., we are dealing with Software, although of low expressiveness. Therefore, there must also be discontinuity in the GAs. Indeed, a GA can mutate the string 0000 to 1000, which is a huge leap. With recombination something very similar happens. What is the effect of this discontinuity on the dynamics of a GA simulation? The program `GAevoOfPerfection.html` presents the tapestry of perfection for GAs. We found that gradualism is definitely more visible to GAs than to Flipp PG. However, there are also cases of jumps in which the red square is not connected to a purple one. Thus, one can not ignore the disturbing role neither of mutation, nor of recombination, nor of the interpretation or semantics of strings when translating them into numbers. But our program does not give a clue on how to separate these effects.

Exercise 16, page 24. Fluctuations are the fuel that drives Evolution to Perfection. Favorable fluctuations are multiplied by selection and eventually another improvement can be made by mutation or recombination. But, the magnitude of fluctuations diminishes as the length of the input grows. This implies that Perfection is the more costly, the higher is complexity. More precisely, its cost is exponential. Therefore, Evolution is both open and permanently dressed in imperfect products. It is open because nobody can predict what miracle would happen next. On the other hand, Evolution undoes in an instant what possibly took centuries to accomplish. Thus, Imperfection reigns supreme. Eternally. That is why my own body must be filled in every sort of imperfections of anatomy and function. By opening and closing my left hand, I found that this is false. The Verbal ET has been falsified in my own body. Wonderful!

Exercise 17, page 25. Since from now on Evolution is scientific, all claims of evolutionary events belong in the realm of Science. This means that they can now be submitted to test. We present some examples to illustrate how to exit from Mythology to celebrate the empowering of the Scientific Method into the realm of Theoretical Evolution:

- Let us consider the chimp-human case. We have more than one billion reasons to consider that they are related by descent, i.e., that they have a common ancestor. So, it is a reasonable hypothesis and is worth a scientific investigation. To fix ideas, let us consider that the common ancestor is the chimp. Now, use artificial selection to evolve the chimp to walk upright. This must be a very easy challenge because chimps occasionally walk like that. A zoo is the most appropriate place to carry this experiment. It might happen that the chimp is not the common ancestor so, calculate it accurately, make a regression of the chimp to that ancestor, and flip Evolution towards the human being.
- Use Directed Evolution to evolve the hand of the chimp into a human one and get ready to register changes in its brain and behavior. This is a study of the Evolution of anatomy, but you can change to a functional one: select for behavior say, the use of tools in a human way, and register the Evolution of the hand and of neural networks in the brain and dorsal spine.
- Test the idea that the mitochondrion and the chloroplast resulted from the engulfing of a cell by another (Scitable [30] 2014). To that aim, select engulfed bacteria that provide the host with energy, sugars, and other substances. The symbiosis is promised to be so fruitful that the hosting cell sacrifices its integrity in favor of a more powerful organization in which the guest cell is anymore a parasite but an integrated and productive part of the whole system. Actual mitochondria are not expected to appear, only useful replacements.
- The genetic code has natural variation (Kooning and Novoshilov, [16] 2009) and theoretical analysis seems to imply that there are plenty of possibilities for its Evolution (Xu and Zhang [37] 2021) although not as much as desired (Terres et al [33] 2000). Now, use artificial selection to connect the standard genetic code in bacteria with that of mitochondria: inject a bacterium into the plasma of another one and select next for the genetic code of a mitochondrion and continue with selection for a mitochondrion to the full. Wisely choose your specimens and, of course, chosen microbes must have no mitochondria (Martin [21] 2010). And make sure that part of the DNA of the nascent mitochondria must migrate to the nucleus of the hosting cell to make the whole ensemble highly functional.

- Nature is so strange that it enables such astonishing marvels as the Hachimoji DNA, which is built on eight letters instead of the four ones of the Natural Genetic Code (Hirao et al, [11] 2002). Reinvent this or something better by Directed Evolution. A program like this is already running because it fills in an urgent need of biotechnology (DeBenedictis et al, [6] 2021).

- An spermatozoon can enter into an ovule because the first has the key to the ciphered gate of the second. The change in the key or in the gate is enough to interdict fecundation (Lobov et al, [20] 2019). But it may happen that both mutants recognize one another. The product is a new species. Use this strategy to create two species and follow their further Evolution. Also, undo changes that cause divergence of two species, and report crossing possibilities of the new versions.

References

[1] ADAMS JILL 2008
Paternity Testing: Blood Types and DNA Nature Education 1(1):146
https://www.Nature.com/scitable/topicpage/paternity-testing-blood-types-an
10

[2] SANJEEV ARORA AND BOAZ BARAK 2007
Computational Complexity: A Modern Approach
https://theory.cs.princeton.edu/complexity/book.pdf
17

[3] BERKELEY: UNDERSTANDING EVOLUTION 2010?
Using trees for classification
https://evolution.berkeley.edu/evolibrary/article/phylogenetics_04
4

[4] DARWIN C 1859
The origin of species Murrap, Londres.
Darwin Online
http://darwin-online.org.uk/content/frameset?itemID=F373&viewtype=side&page
Online Variorum of Darwin's Origin of Species: first British edition (1859), page 186
http://darwin-online.org.uk/Variorum/1859/1859-187-c-1861.html
3, 4

[5] DAVIES NAHLA 2020 *Neural networks could help computers code themselves: Do we still need human coders?* The Overflow, sept 16/2020
https://stackoverflow.blog/2020/09/16/neural-networks-could-help-computers
9

[6] DEBENEDICTIS, E.A., CARVER, G.D., CHUNG, C.Z. ET AL. 2021
Multiplex suppression of four quadruplet codons via tRNA directed evolution.
Nature Communications volume 12, Article number: 5706 (2021)
https://www.Nature.com/articles/s41467-021-25948-y 31

[7] S J FREELAND 1 , L D HURST 1998
The genetic code is one in a million. J Mol Evol 1998 Sep;47(3):238-48.

doi: 10.1007/pl00006381.
https://pubmed.ncbi.nlm.nih.gov/9732450/ 11

[8] GENETIC PROGRAMMING GROUP
Official website https://geneticprogramming.com/ 13

[9] A. D. HALL 1935
HUGO DE VRIES 1848-1935
https://royalsocietypublishing.org/doi/pdf/10.1098/rsbm.1935.0002
4

[10] HENNIG, BORIS. 2011
Teleonomy.
Philosophiegeschichte und Logische Analyse. 14. 185-202. 10.30965/26664275-01401012.
https://www.researchgate.net/publication/288275673_Teleonomy
20

[11] HIRAO, ICHIRO & OHTSUKI, TAKASHI & FUJIWARA, TSUYOSHI & MITSUI, TSUNEO & YOKOGAWA, TOMOKO & OKUNI, TAEKO & NAKAYAMA, HIROSHI & TAKIO, KOJI & YABUKI, TAKASHI & KIGAWA, TAKANORI & KODAMA, KOICHIRO & YOKOGAWA, TAKASHI & NISHIKAWA, KAZUYA & YOKOYAMA, SHIGEYUKI, 2002.
An unnatural base pair for incorporating amino acid analogs into proteins.
Nature biotechnology. 20. 177-82. 10.1038/nbt0202-177.
https://www.researchgate.net/publication/11538018_An_unnatural_base_pair_fo
31

[12] INGO BERG
Simulated Evolution - An algorithm for simulating evolution in a web browser
https://beltoforion.de/en/simulated_evolution/index.php#idStart
Simulierte Evolution - Simulieren von Evolutionsprinzipien mit dem Computer
https://beltoforion.de/de/simulierte_evolution/index.php
20

[13] CAIFU JIANG, AZIZ MITHANI, XIANGCHAO GAN, JIANNIS RAGOUSSIS, RICHARD MOTT, NICHOLAS P. HARBERD AND OTHERS 2011
Regenerant Arabidopsis Lineages Display a Distinct Genome-Wide Spectrum of Mutations Conferring Variant Phenotypes
Current Biology, Report— Volume 21, ISSUE 16, P1385-1390, August 23, 2011
https://www.cell.com/current-biology/fulltext/S0960-9822(11)00771-8
10

[14] LYNN JORDE, PH.D. 2016
NHGRI Current Topics in Genome Analysis 2016 Week 7: Introduction to Population Genetics April 6, 2016
https://www.genome.gov/sites/default/files/media/files/2019-03/Introduction
4

[15] KENNETH J. LOCEYA, AND JAY T. LENNON 2016
Scaling laws predict global microbial diversity. PNAS, 2016 DOI: 10.1073/pnas.1521291113
https://www.pnas.org/content/113/21/5970 25

[16] EUGENE V. KOONIN AND ARTEM S. NOVOZHILOV 2009
Origin and evolution of the genetic code: the universal enigma
IUBMB Life. 2009 Feb; 61(2): 99–111.
doi: 10.1002/iub.146
PMCID: PMC3293468
NIHMSID: NIHMS78555
PMID: 19117371
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3293468/
30

[17] JOHN R. KOZA, RICCARDO POLI 2005
Genetic Programming. Search Methodologies (pp.127-164)Publisher: Springer US, New York, NY, USAEditors: Burke, Edmund K., and Kendall, Graham.
https://www.researchgate.net/publication/229091441_Genetic_Programming
13

[18] KURZGESAGT – IN A NUTSHELL (2018)
The Deadliest Being on Planet Earth – The Bacteriophage
https://www.youtube.com/watch?v=YI3tsmFsrOg 8

[19] JEAN-BAPTISTE PIERRE ANTOINE DE MONET, CHEVALIER DE LAMARCK 1809
PHILOSOPHIE ZOOLOGIQUE 1809
1er août 1744, Bazentin, Somme, 18 décembre 1829, Paris.
https://ia801004.us.archive.org/28/items/LamarckPZ/Lamarck_PZ.pdf
4

[20] LOBOV AA, MALTSEVA AL, MIKHAILOVA NA, GRANOVITCH AI. 2019
The Molecular Mechanisms of Gametic Incompatibility in Invertebrates. *Acta Naturae. 2019 Jul-Sep;11(3):4-15. doi: 10.32607/20758251-2019-11-3-4-15. PMID: 31720011; PMCID: PMC6826153.*
https://pubmed.ncbi.nlm.nih.gov/31720011/ 31

[21] MARTIN WILLIAM, MENTEL MAREK , 2010
The Origin of Mitochondria
Nature Education 3(9):58.
https://www.Nature.com/scitable/topicpage/the-origin-of-mitochondria-14232
30

[22] NHGRI , 2021
Genome
https://www.genome.gov/genetics-glossary/Genetic-Code
8

[23] MCLEOD SAUL 2020
Karl Popper - Theory of Falsification.
SimplyPSychology
https://www.simplypsychology.org/Karl-Popper.html 3

[24] HELEN PEARSON 2003
250 years of Linnaeus' plant names celebrated.
Nature News— doi:10.1038/news030825-7

https://www.Nature.com/news/2003/030825/full/news030825-7.html 3

[25] VINCENT RACANIELLO 2013
How many viruses on Earth? Virology blog
https://www.virology.ws/2013/09/06/how-many-viruses-on-earth/ 8

[26] JOSÉ RODRÍGUEZ 2010 *The genome is software and evolution is a software developer*. Arxiv 1003.0575
https://arxiv.org/abs/1003.0575 10

[27] JOSÉ RODRÍGUEZ 2017 *EASY JAVA GENETIC PROGRAMMING. JAVA FOR THE STUDY OF EVOLUTION VOLUME XVII version 2*
https://ejristos.com/dl/javaevol17v2main.pdf 13

[28] JOSÉ RODRÍGUEZ 2019 *Christianity, evolution, complexity, simulations, and passwords v2*.
Text without programs:
https://ejristos.com/dl/PasswordsText.html
Text and programs:
https://ejristos.com/dl/PasswordsHtml.zip 4

[29] JOSÉ RODRÍGUEZ 2021 *The Darwinian Evolutionary Theory at its face value*.
Text without programs:
https://ejristos.com/dl/DarwinAtFaceValue.pdf
https://ejristos.com/dl/DarwinEnSerio.pdf (Spanish)
Text and programs:
https://ejristos.com/dl/DarwinAtFaceValue.zip 4, 5

[30] SCITABLE 2014
The origin of mitochondria and chloroplasts
https://www.Nature.com/scitable/content/the-origin-of-mitochondria-and-chl 30

[31] SCIENCE LEARNING HUB – POKAPŪ AKORANGA PŪTAIAO 2011.
Mendel's experiments
https://www.sciencelearn.org.nz/pages/about 4

[32] SPECTOR, L.,H. BARNUM,AND H.J. BERNSTEIN.1998.
Genetic Programming for Quantum Computers. In Genetic Programming 1998: Proceedings of the Third Annual Conference, edited by J.R.Koza, W.Banzhaf, K.Chellapilla, K.Deb, M.Dorigo, D.B.Fogel, M.H.Garzon, D.E.Goldberg, H.Iba, and R.L.Riolo. pp.365-374. San Francisco, CA: Morgan Kaufmann.
http://faculty.hampshire.edu/lspector/pubs/GP-quantum-GP98-with-cite.pdf 13

[33] TERRES A. RONNEBERG, LAURA F. LANDWEBER, AND STEPHEN J. FREELAND 2000
Testing a biosynthetic theory of the genetic code: Fact or artifact?
PNAS December 5, 2000 97 (25) 13690-13695;
https://doi.org/10.1073/pnas.250403097 30

[34] THALER 2018 DAVID S. *Why should mitochondria define species? 2018*.
Human Evolution 33(n1-2) May 2018
https://www.researchgate.net/publication/325270582_Why_should_mitochondria_
10

[35] W3SCHOOLS 2020
The code game
https://www.w3schools.com/codegame/index.html 9

[36] CONTRIBUTEURS DE WIKIPÉDIA 2021.
Jean-Baptiste de Lamarck [Internet]. Wikipédia, l'encyclopédie libre; 2021 fév. 15, 00:13 UTC [cited 2021 fév. 15]. Disponible à l'adresse:
http://fr.wikipedia.org/w/index.php?title=Jean-Baptiste_de_Lamarck&oldid=1
4

[37] HAIQING XU AND JIANZHI ZHANG 2021.
On the Origin of Frameshift-Robustness of the Standard Genetic Code. Mol Biol Evol. 2021 Oct; 38(10): 4301–4309.
PMCID: PMC8476161
PMID: 34043802:
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8476161/
30

All cited links were active by 14/XII/2021