Politecnico di Torino

Department of Mechanical and Aerospace Engineering

Master's degree in biomedical engineering

A Search for the Physical Basis of the Genetic Code

Supervisor

Prof. J. A. TUSZYNSKI

Candidate

Michele MASOTTI

Academic Year 2023/2024

"To Cesare, for comprehension.

To Luciana, for caring.

To Francesca, for confronting.

To Raffaele, for confessions.

To Barbara, for everything."

Summary

Genetic code is one of the greatest mysteries of biology: it is shared by basically every living organism on this planet, with little to none differences between species that can be very far away from each other from an evolutionary point of view. Not only organisms share similar DNA structures and protein synthesis methods: the translation from a base triplet to the respective amino acid, i.e. the genetic code, is found to be surprisingly consistent among all living creatures. The genetic code is not only stable between different species, but also through millions of years of evolution: it has been almost identical to itself for a very long time.

Researchers strongly believe that there must be a reason behind this: “survival of the fittest” is a widely known expression that states that during evolution only the most suit survives and keeps on proliferating, which hints that there must have been some force pushing all organisms to adopt this code. This concept is also known as “evolutionary pressure”: it can be given by external factors like the environment, but also by some intrinsic quality of the genetic code itself. Unfortunately, despite all the resources spent on this topic and the researchers that were and are spending time to solve this mystery, genetic code does not seem to be willing to cooperate and no one has been able to really give an explanation to its almost omnipresence in our world. This work aims to analyse the genetic code from two different points of view: on the first hand, entropy is checked as a primary indicator of the genetic code characteristics, then its stability is evaluated and compared to other possible codes. 36 different genomes from GenBank have been studied using Matlab to understand whether its characteristics can be better understood. This study is hence approaching the problem from a mathematical and statistical point of view, partially ignoring the biochemical pathways that take place inside living organisms. In the first part, entropy of mutated genetic sequences is studied to determine whether genetic code is also a way living organisms use to decrease this quantity. Then, a fitness function will be used to investigate robustness of the code to mutations and mistranslations.

Main results of this work can be summarized as follows:

- Genomes show higher entropy value if translated with the standard genetic code.
- Standard genetic code shows better performances when tested with a fitness function. Similar results are obtained using genetic code 4, where part of the structure of the genetic code is denied.
- Standard genetic code has a good ratio between start/stop mutations and body mutations, disfavouring start/stop mutations which are considered more important in terms of energy saving.
- An empirical method to quantify relative distances between redundant amino acids is proposed.

Acknowledgements

A special thanks goes to my supervisor, Prof J. A. Tuszynski, for his patience and kindness in guiding me through this work and for the great opportunity this thesis gave me.

Thanks to everyone who has supported me through my university journey.

Summary

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

To reproduce and to synthetize all kinds of compounds that living organisms need in order to survive and complete all the tasks a living creature has to perform, information need to be extracted from the DNA and translated into something our organism can actually use. Ribosomes, who are the main characters of protein synthesis, do exactly that: they read information structured as a sequence of four different bases in groups of three elements and convert each group into one of the 20 possible standard amino acids. These molecules are subsequently used to build proteins, molecules able to interact with the biological environment and to regulate all kinds of processes that happen in living organisms. We know that DNA structure is shared by basically every living organism on this planet, but what is way more interesting is that even the genetic code, which is the conversion from a bases triplet into a specific amino acid, is largely the same across different species, geographic areas, even ages and biological domains. Thinking about different animals and, more generically, different species, it is easy to notice that many different adaptations were adopted to survive and proliferate, to be the most suit specie and to win the evolutionary race: there are wings, fins, species that breathe underwater and other that can live inside a volcano or in extremely acid environments. Some species run fast, some other have a powerful brain. Every specie evolves along its own path, developing different tools to adapt and overcome the environmental issues that it faces through time. A spider and a whale have very different approaches to life and survival, but they both made it up to the present days despite being very different. Some features only belong to specific species, like the bolas spider's hunting technique, other are more common, like having strong claws and fangs. Certain features are more common than others, and there are two main reasons for this: the first one is a common ancestor that passed its valuable features to its progeny; second one, a specific tool may be particularly suitable for survival and hence be developed independently by different species. This leads to the idea that if something is widely shared along different species it should have some strong advantages.

This is what makes the genetic code so interesting, because it is not just very common, it is basically everywhere. Scientists strongly believe that there must be a reason for this, so let's take into consideration the two reasons why one feature can be common. First, a common ancestor: it is possible that a particularly fit ancestor used the genetic code and, for some reason, that code spread through all other species. This idea has been widely explored, with focus on the environment it lived in (Weiss 2016). The second possibility is that the genetic code is so good that every specie on the planet eventually got to the point where it started using it.

1.1 EXISTING THEORIES

Several theories exist to try and explain the mystery of the genetic code. Different approaches (i.e. chemical, algebraic, evolutionary) but none of them, up to current knowledge, is complete. Different theories can be compatible or collide with others, but there is no experimental evidence strong enough to pose one of them surely above the others. Biochemistry is in fact a very complex world, with thousands of different aspects to take into consideration. Here, we hover the main existing theories.

1.1.1 Stereochemical Theory

Stereochemical hypothesis states that codons or anticodons have both geometric and chemical affinity with the corresponding amino acid. It is supported by many different research (Johnson 2010) and it is fascinating because it immediately answers the question about universality of the code: chemistry is universal and so are stereochemical interactions. There has been evidence that some amino acids are strongly related to their corresponding codon, as for arginine (M Yarus 1989). However, it is hard to believe that stereochemical bonding alone built the whole genetic code.

1.1.2 Ambiguity Reduction Theory

This theory is one of the most famous and explored theories of all. It states that one of genetic code's main goals is to reduce the amount of mistranslation that can occur during the translation of the DNA. Since four different bases exist and they code in groups of three, 64 different codons are generated. They only code for 20 amino acids plus the stop codons, so there is plenty of redundancy in the assignment. Most amino acids have multiple codons associated with them, and it is very easy to see that similar codons code for the same amino acid. This is helpful if the goal is to reduce the possibility of making a mistake during translation: as an example, if there is an error while coding the triplet "CUU" and, for some reason, there is a mistake on the third letter. Whatever codon is translated, as long as the first two letters are "C" and "U", it will code for Leucine. Genetic code itself cannot do anything about occurring errors, but it can be made in such a way that errors tend to be silent, or at least less relevant, than it would be with another code. It is not just about grouping up codons that correspond to the same amino acid, the whole genetic code has some interesting features such as a codon "NUN" with "N" being any of the letters will tend to code for a hydrophobic amino acid.

First base position	Second base position									Third base position
	U		C		A		G			
U	UUU	F	UCU	S	UAU	Y	UGU	C	U	
	UUC		UCC		UAC		UGC		C	
	UUA	L	UCA		UAA	STOP	UGA	STOP	A	
	UUG		UCG		UAG		UGG	W	G	
C	CUU	L	CCU	P	CAU	H	CGU	R	U	
	CUC		CCC		CAC		CGC		C	
	CUA		CCA		CAA	Q	CGA		A	
	CUG		CCG		CAG		CGG		G	
A	AUU	I	ACU	T	AAU	N	AGU	S	U	
	AUC		ACC		AAC		AGC		C	
	AUA		ACA		AAA	K	AGA	R	A	
	AUG	M	ACG		AAG		AGG		G	
G	GUU	V	GCU	A	GAU	D	GGU	G	U	
	GUC		GCC		GAC		GGC		C	
	GUA		GCA		GAA	E	GGA		A	
	GUG		GCG		GAG		GGG		G	

Table 1: standard genetic code

1.1.3 Coevolution Theory

This theory postulates that “The structure of the codon system is primarily an imprint of the prebiotic pathways of amino-acid formation, which remain recognizable in the enzymic pathways of amino-acid biosynthesis. Consequently, the evolution of the genetic code can be elucidated based on the precursor-product relationships between amino acids in their biosynthesis. The codon domains of most pairs of precursor-product amino acids should be contiguous, i.e., separated by only the minimum separation of a single base change.” (Wong 1975)

The idea behind the coevolution theory is that prebiotic synthesis was not able to provide all twenty amino acids in an adequate manner, and so some of them had to be somehow derived from the biosynthesis pathways of amino acids. The most important implication is that genetic code became the code we now know because of what was available in the environment before and during the appearance of the last common ancestor. So not all codons were used initially, but because of needed efficiency different pathways occurred and led to the present genetic code. This theory is supported by several evidence (Wong 1975 and references): many pairs of precursor-derived amino acids occupy near positions in the genetic code table, meaning they only differ for one base, such as: Glu – Gln, Glu – Arg, Asp – Asn, Asp – Thr, Asp – Lys, Gln – His, Thr – Ile etcetera. While giving rise to derived amino acids, precursors let them part of their codon domain, explaining why genetic code is so well-organized in terms of similarity of nearby triplets and amino acids.

1.1.4 Frozen Accident Theory

This theory (Crick 1967) differentiates slightly from the other theories, because it does not give an answer to how the genetic code was initially formed, but to how the code has been the same through ages. The question is: if the genetic code was made up by circumstances, why did it not change when circumstances changed? One of the answers is the frozen accident theory: genetic code alterations obviously imply many different metabolic and biochemical pathway alterations, which were not possible to perform because of how complex living organisms had become. Genetic code formed because of some specific conditions and became more and more complex, involving several different aspects of life, up to the point where any change in the code would be too hard to carry on throughout all the different functions of organisms. This meant that no organism with altered genetic code could be better suited than the "regular" organism of the same species, because no selective pressure could be strong enough to favour them in a strong way.

1.1.5 Physicochemical Theory

This theory can be seen as a generalization of the ambiguity reduction theory. It is also the one theory explored in this work, and it states that genetic code is arranged in such a way that it minimizes the impact of mutations during protein synthesis. Amino acids have many different features, such as charge, polarity, hydrophilicity and so on, so it is possible to cluster them and assign some degree of similarity between different amino acids. We now give a fictional example to explain the main concept: proteins usually have an active site, i.e a small sequence of amino acids that are in charge to perform the protein task, and many other residues that do not participate actively to the purpose but give the protein its structure. If one of the non-active residues is mistranslated so that were there should be a Leucine a Valine is found. These two amino acids have: similar volume, comparable hydrophilicity, similar area and polarity, same charge and very similar shape. This means that the protein could still work because its structure would stay almost the same even if a mutation occurred. What happens if, instead of a Valine a residue of Arginine is added to the chain? Main features like hydrophilicity, shape factor, polarity and hydrophilicity are way different from the ones that characterize Leucine and this would lead to huge conformational changes that would make the whole protein useless ad inadequate to perform its tasks. Genetic code could be stable and favourable with respect to other possible codes because of its ability to minimize adverse consequences of mutations.

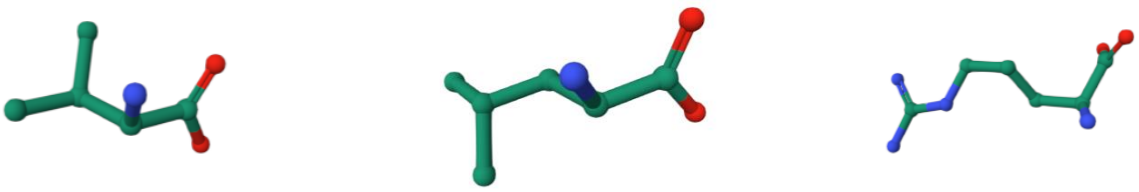

Figure 1: From left to right: Valine, Leucine, Arginine. Official Protein Data Bank website.

1.1.6 Coexistance of Different Theories

Some of the previously shown theories can easily coexist with each other, like the ambiguity reduction and physicochemical hypothesis. In this case, genetic code is a way to minimize errors and to allow a safer translation of vital information during cellular activities. Physicochemical theory also easily links with the stereochemical one: similar amino acids are, in fact, also similar from a chemical point of view in terms of interactions with the surrounding environment and hence with similar triplets of bases. Despite that, these three theories have a limit: they do not consider evolution and first appearance of the genetic code completely, they are only based upon chemistry and efficiency of the code; in biology, organisms evolve because selective pressure forces them to change giving a reward, i.e. an increased ability to survive in their environment. This does not mean that every organism will change according to what is convenient, because there may be barriers that do not allow a specific system to evolve towards a certain direction. As an explicative comparison, one can think about selective pressure as an exergonic chemical reaction: despite being favourable from an energetic point of view, the reaction may not happen in case of an excessive energy barrier. This means that taking into consideration only some aspects of the genetic code may not give an acceptable answer to the question because some other key aspects are excluded.

Co-evolution theory, instead, gives a reasonable answer to the genesis of the code. Although it is not straight forward, it is possible to match this theory with the physicochemical one: because of the precursor-derived mechanism, derived amino acids probably conserved part of the chemical characteristics of their precursors and took part of their domains, generating a genetic code where similar features belong to contiguous codons or, at least, to codons not far away from each other in the code.

Frozen accident theory, instead, does not enter a more general theory comprehensive of different theories; accidentality is clearly not compatible with stereochemical or co-evolutional ideas, which is also the main reason why more complex and rational theories were developed and explored to answer questions on the genetic code.

1.2 ENTROPY AND GENETIC CODE

Entropy is known to be a powerful tool to analyse complex systems and to try and extrapolate information. It has been used successfully several times and it has some strong implications concerning genetic code and, more generally, life itself. It is known that entropy cannot

decrease, but it tends to always increase. It is although possible to have a local decrement of entropy in a system, at the cost of a bigger increase on the external of the system. Since the principle of entropy states that everything tends, over time, to become completely uniform, it is clear that life works against entropy in every form. Entropy increasing means that the system becomes more disordered, less organized and structured: this is why it is possible to say that life fights against entropy as much as it can, because one of the goals a living creature must achieve is to keep itself ordered and hierarchically organized.

Entropy as a concept can be applied to sequences containing information (Shannon's entropy), such as DNA: if somehow genetic information was just a sequence of random letters without any meaning it would have maximum entropy, i.e. maximum disorder, but this is not the case. In order to study the genetic code intended as the translation from codons to amino acids, entropy must be evaluated onto amino acid sequences translated via different genetic codes starting from the same A C T G sequences.

2 METHODS

2.1 INITIAL CONSIDERATIONS

This study includes genomes 36 different species. Selected species are:

- Aaosphaeria Arxii (fungi, dothideomycetes)
- Amanita Muscaria (fungi, agaricomycetes)
- Arctogadus Glacialis (animalia, actinopterygii)
- Balenoptera Musculus (animalia, mammalia)
- Betta Splendens (animalia, actinopterygii)
- Blastocatellia Bacterium(bacteria, blastocatellaceae)
- Cutaneotrichosporon Dermatis (fungi, tremellomycetes)
- Cyprinus Carpio (animalia, actinopterygii)
- Daldinia Concentrica (fungi, sordariomycetes)
- Danio Rerio (animalia, actinopterygii)
- Erithacus Rubecola (animalia, aves)
- Felis Catus (animalia, mammalia)
- Ginkgo Biloba (plantae, ginkgophyta)
- Granulicella (bacteria, acidobacteria)
- Hapalochlaena Maculosa (animalia, mollusca)
- Hepatitis B Virus (virus)
- Homo Sapiens (animalia, mammalia)
- Leishmania Mexicana (protista, kinetoplastea)
- Luteitalea Pratensis (bacteria, clostridia)
- Megasphera Hexanoica (bacteria, negativicutes)202020
- Monoraphidium (plantae, chlorophyta)
- Mus Musculus (animalia, mammalia)
- Octopus Bimaculoides (animalia, mollusca)
- Otolemur Garnettii (animalia, mammalia)
- Pan Troglodytes (animalia, mammalia)
- Pao Palembangensis (fungi, dothideomycetes)
- Patellaria Atrata (fungi, sordariomycetes)
- Pseudis Tocantis (animalia, amphibia)
- Pyrococcus Furiosus (archaea, thermococci)
- Salmo Salar (animalia, actinopterygii)303030
- Salvator Merianae (animalia, reptilia)
- Trypanosoma Melophagium (protista, kinetoplastea)
- Varanus Komodensis (animalia, reptilia)
- Vitis Cinerea (plantae, magnoliopsida)
- Xenopus Laevis (animalia, amphibia)
- Zalophus Californianus (animalia, mammalia)

These species were randomly selected in order to cover different domains and have information about living organisms that are as complete as possible. Ancient organisms like Archaea are more relevant in this study because they are closer in time to when genetic code appeared, hence giving more solid information about the code. All genomes were downloaded from the National Library of Medicine official website (https://www.ncbi.nlm.nih.gov/) in FASTA format (.fna) and analysed with the use of the Bioinformatics Toolbox from Matlab software.

FASTA format contains genome sequences in terms of nucleotide bases, in order to perform calculations they were converted into numeric sequences made by 0, 1, 2 and 3. All the original sequences were stored into Matlab workspaces. Nucleotide sequences were then forced to mutate randomly five thousand times and one every one hundred sequences was stored into a matrix of five hundred rows, each row containing a mutated genome. Probability of mutation was set as 0.0012 per nucleotide, considering three different subcases of equal probability of mutating towards one of the three other nucleotides. All following calculations were performed on all mutated sequences.

2.2 EVALUATED GENETIC CODES

Five genetic codes were evaluated to compare performances under an entropic and functional point of view. This is meant to explore the possibilities that the standard genetic code has some special feature that helped him win the evolutionary competition.

	Second base position									
		U		C		A		G		
First base position	U	UUU	F	UCU	S	UAU	Y	UGU	C	U
		UUC		UCC		UAC		UGC		C
		UUA	L	UCA		UAA	STOP	UGA	STOP	A
		UUG		UCG		UAG		UGG	W	G
	C	CUU	L	CCU	P	CAU	H	CGU	R	U
		CUC		CCC		CAC		CGC		C
		CUA		CCA		CAA	Q	CGA		A
		CUG		CCG		CAG		CGG		G
	A	AUU	I	ACU	T	AAU	N	AGU	S	U
		AUC		ACC		AAC		AGC		C
		AUA		ACA		AAA	K	AGA	R	A
		AUG	M	ACG		AAG		AGG		G
	G	GUU	V	GCU	A	GAU	D	GGU	G	U
		GUC		GCC		GAC		GGC		C
		GUA		GCA		GAA	E	GGA		A
		GUG		GCG		GAG		GGG		G

Third base position

Table 2: genetic code 1 (standard).

Standard genetic code, here called genetic code 1. It is shared by almost every specie on this planet and has yet to be understood completely. Its performances have been compared to those of the other codes in order to extrapolate information about it.

First base position	Second base position									Third base position
	U		C		A		G			
U	UUU	F	UCU	P	UAU	Y	UGU	C	U	
	UUC		UCC	S	UAC		UGC		C	
	UUA	L	UCA		UAA	L	UGA	P	A	
	UUG	STOP	UCG	G	UAG	G	UGG	W	G	
C	CUU	L	CCU	S	CAU	H	CGU	R	U	
	CUC		CCC	P	CAC		CGC		C	
	CUA		CCA		CAA	Q	CGA	A	A	
	CUG	T	CCG	STOP	CAG		CGG	R	G	
A	AUU	I	ACU	T	AAU	N	AGU	S	U	
	AUC		ACC	L	AAC		AGC		C	
	AUA		ACA	T	AAA	K	AGA	R	A	
	AUG	M	ACG		AAG	V	AGG		G	
G	GUU	V	GCU	A	GAU	D	GGU	STOP	U	
	GUC		GCC		GAC		GGC	G	C	
	GUA		GCA	R	GAA	E	GGA	S	A	
	GUG	K	GCG	A	GAG		GGG	G	G	

Table 3: genetic code 2.

Second genetic code is made by swapping 8 pairs of amino acids, including the stop codons. This aims to make the code a bit more disordered, enhancing the probabilities of a mutation in a codon to lead to a “further” amino acid in terms of similarity. Modified amino acids are highlighted in red.

		Second base position								
		U		C		A		G		
First base position	U	UUU	F	UCU	S	UAU	Y	UGU	C	U
		UUC		UCC	P	UAC		UGC		C
		UUA	L	UCA	S	UAA	N	UGA	L	A
		UUG	STOP	UCG	I	UAG	G	UGG	W	G
	C	CUU	T	CCU	P	CAU	H	CGU	V	U
		CUC	D	CCC	S	CAC		CGC	R	C
		CUA	L	CCA	STOP	CAA	Q	CGA		A
		CUG		CCG	P	CAG		CGG	A	G
	A	AUU	I	ACU	T	AAU	STOP	AGU	S	U
		AUC	S	ACC	L	AAC	N	AGC	E	C
		AUA	I	ACA	T	AAA	V	AGA	R	A
		AUG		ACG		AAG	K	AGG		G
	G	GUU	R	GCU	R	GAU	L	GGU	STOP	U
		GUC	V	GCC	A	GAC	D	GGC	G	C
		GUA		GCA		GAA	E	GGA		A
		GUG	K	GCG	G	GAG	S	GGG	A	G

Third base position

Table 4: genetic code 3.

The third genetic code was obtained by performing 12 swaps between amino acids and by adding an extra stop codon by removing one Proline amino acid. Since proline is coded by 4 codons and the stop signal is transmitted by 3 codons, this action did not alter the overall effect of redundancy. The code is even more disordered than code 2.

		Second base position								
		U		C		A		G		
First base position	U	UUU	R	UCU	V	UAU	I	UGU	A	U
		UUC		UCC		UAC		UGC		C
		UUA		UCA		UAA		UGA		A
		UUG		UCG		UAG	M	UGG		G
	C	CUU	T	CCU	G	CAU	S	CGU	N	U
		CUC		CCC		CAC		CGC		C
		CUA		CCA		CAA	R	CGA	K	A
		CUG		CCG		CAG		CGG		G
	A	AUU	D	ACU	C	AAU	S	AGU	Y	U
		AUC		ACC		AAC		AGC		C
		AUA	E	ACA	STOP	AAA		AGA	STOP	A
		AUG		ACG	W	AAG		AGG		G
	G	GUU	H	GCU	F	GAU	P	GGU	L	U
		GUC		GCC		GAC		GGC		C
		GUA	Q	GCA	L	GAA		GGA		A
		GUG		GCG		GAG		GGG		G

Third base position

Table 5: genetic code 4.

The fourth genetic code is obtained by swapping squares of the original genetic code. Squares are identified as the blocks of 4 amino acids with the same first and second letter. The idea is to voluntarily disorganize the code so that it maintains a certain degree of order, but different to the order of the standard genetic code. Amino acids still have close redundant codons that code for them, but they are not organized following the first and second letter. Arginine as an example (letter R) is coded by four codons with the same first and second letter, but the other two redundant codons are two point mutations away because they do not share first or second letter. This code can be thought as a way to explore the importance of neighbours amino acids and not only of neighbours redundant codons, and in particular how relevant the organization over the first two letters of codons is.

		Second base position								
		U		C		A		G		
First base position	U	UUU	V	UCU	Q	UAU	K	UGU	S	U
		UUC	A	UCC	S	UAC	I	UGC	STOP	C
		UUA	T	UCA	L	UAA	R	UGA	D	A
		UUG	S	UCG	V	UAG	I	UGG	S	G
	C	CUU	W	CCU	P	CAU	F	CGU	H	U
		CUC	K	CCC	A	CAC	E	CGC	T	C
		CUA	P	CCA	T	CAA	R	CGA	C	A
		CUG	S	CCG	R	CAG	A	CGG	P	G
	A	AUU	Y	ACU	D	AAU	G	AGU	L	U
		AUC	L	ACC	W	AAC	M	AGC	Y	C
		AUA	S	ACA	F	AAA	N	AGA	G	A
		AUG	Q	ACG	STOP	AAG	L	AGG	L	G
	G	GUU	L	GCU	STOP	GAU	H	GGU	E	U
		GUC	N	GCC	I	GAC	R	GGC	A	C
		GUA	R	GCA	R	GAA	G	GGA	STOP	A
		GUG	C	GCG	G	GAG	V	GGG	T	G

Third base position

Table 6: genetic code 5.

The last examined code was obtained by a completely random association of codons and amino acids. Redundancy was conserved for every amino acid, but assignment is completely random. A Python shuffling function was used to obtain the random matches. This code is thought to be a “control” for the performances of the other codes because it is assumed to have zero organization, so it can be used to determine which part of the performances of the genetic codes is due to their structure.

2.3 ENTROPY

The first analyses were performed over mutated sequences in order to obtain entropy information about genomes. Entropy follows the formula:

$$S = -\sum_i p_i * log(p_i)$$

known as Shannon's Entropy. This quantity measures order and information of a sequence, the lower the value is, the best the code is behaving. Reducing entropy is, in fact, what living organisms must do to survive. Shannon's entropy is always evaluated on the amino acid sequences.

Although entropy is an extensive property, i.e. depending on the size of the system, Shannon's entropy is dependent on the number of different symbols and on their probability. In this particular case, considering specific entropy may lead to mistakes because probability of appearance of amino acids is not dependent on the length of the sequence. Entropy of the sequence "11234" and "1123411234", being the same sequence but doubled, is exactly the same.

In order to evaluate the goodness of different genetic codes every specie is taken into consideration at once: each one of the five hundred values in the following plots is the average of entropy for every organism at that step in the mutation process: first value is the average of entropy of the 36 genomes after one hundred cycles of mutation, the second is the average of the entropy values after two hundred cycles of mutations and so on.

One important thing to mention is that Shannon's entropy does not give information about how ordered a sequence is, but for how it is defined it only gives information about amino acid distribution. The highest value corresponds to an equal distribution of all amino acids, i.e. every amino acid occurs with the same probability. This measure serves to give an idea of how "unbalanced" amino acids are in terms of rate of appearance, although without giving any information about which amino acid appears with higher frequency. To solve this problem, it is possible to set up a fitness function, aiming to obtain further information about solidity of the code.

Lastly, entropy was studied for different branches of living organisms, grouping them into kingdoms to better understand if and how evolution and genome entropy are related.

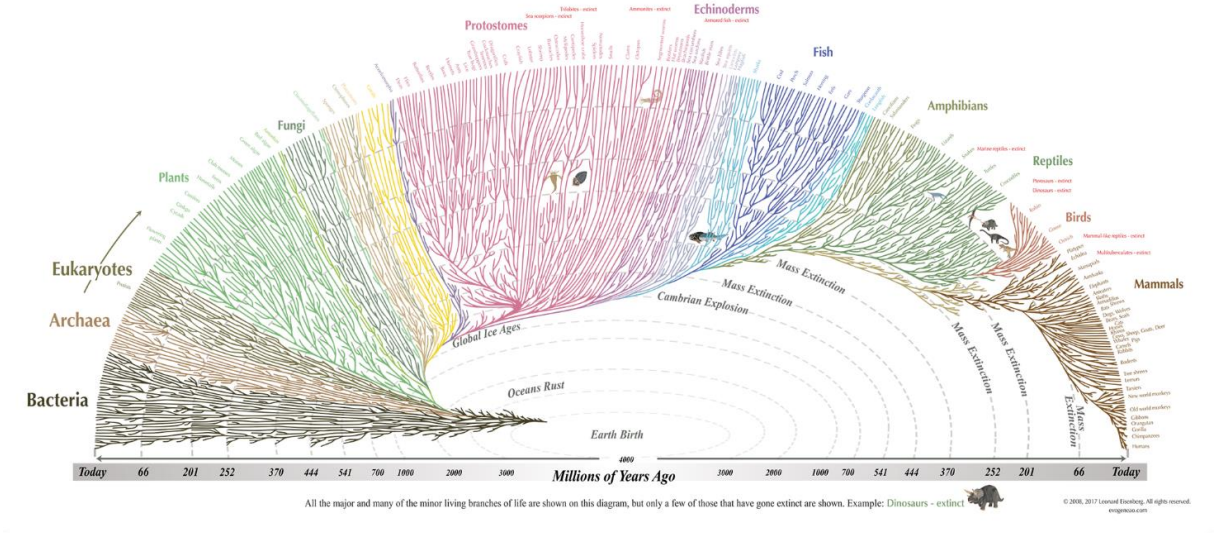

Figure 2: the tree of life.

To do this, genomes were clustered into the following classes:

- Amphibia;
- Archaea;
- Aves;
- Bacteria;
- Fishes;
- Fungi;
- Mammals;
- Mollusca;
- Plants;
- Protista.

The idea is that more ancient organisms can give more relevant information about the raising of the genetic code because they are closer, in an evolutionary way, to those organisms who first developed the code. Checking if there are differences on genomes from species subsequent to each other in time may help to understand the genetic code. For this purpose, entropy of every class is obtained as the average of the entropies of the elements of the class and observations are made upon differences between different classes and different genetic codes.

2.4 FITNESS FUNCTION

2.4.1 Amino acids clustering

As stated while introducing the physiochemical hypothesis, it is possible to assign a certain degree of similarity between amino acids. A fitness function is constructed in order to evaluate

how solid a certain code is towards mistranslations and errors. The following steps are repeated for each genetic code.

First of all, the original nucleotide sequence is converted into an amino acid sequence following the chosen genetic code (real genetic code for case one, then artificial genetic codes). This first amino acid sequence is what is taken as a reference to evaluate stability of the code. Mutated nucleotide sequences are then translated into amino acid sequences as well, and the fitness function evaluates how different mutated amino acid sequences are from the original one.

To do this, it is necessary to define a way to cluster amino acids. The issue with clustering is that it is subjective: inter- and intra-distances can be chosen and hence different methods may lead to different clusters. In 1996, Stanfel proposed a way to cluster amino acids basing on their main features (Stanfel 1996). This method is not an actual clustering, but a way to measure distances between amino acids based on their properties. This method solves the issue because if clustering is a discrete way to associate different elements of a set, distance is a continuous quantity and is only dependent on the way distances are measured, gaining independence from subjective thresholds.

Several different features, considered to be the most relevant from a biochemical point of view, of amino acids are taken into consideration (see Stanfel, 1996 and references for further information):

- Volume: important for steric encumbrance and hence for the shape of the molecule. There are some differences on these measures depending on different studies, but they are considered negligible because they vary up to maximum 10%.
- Hydrophilicity: taken for hydration free energy. A special mention goes to neutral proline: since no value is defined, it got assigned the average value between maximum and minimum values (respectively those of glycine and arginine). This feature is essential because changing hydrophilicity of one reside strongly varies the shape of the final molecule.
- Surface area: intended as accessible surface area of the molecule. It is known that this property is related with differences in free energy when a molecule is surrounded by solvent.
- Polarity.
- Charge: possible values are -1 (negatively charged), 0 (neutral) and 1 (positively charged).
- Shape: this is the most controversial value. Shape index is based on similarity to aliphatic and aromatic structures, setting a value of 1 for glycine and a value of 12 to phenylalanine. Scores were then subject to differences in the number of atoms.

All these values come with different unit of measure, but it is not a problem because they will be normalized before performing calculations. The last thing to be done is to hierarchically order these features to understand which one is more relevant: this goal is achieved by assigning a weight to these properties. In this study weights are the squared inverses of the average distances of corresponding features of amino acids.

In Stanfel's study, two more features are considered: number of donate and accepted hydrogen bonds. Unfortunately, these data are not present for every amino acid and they can

Amino acid	Symbol	Volume	Hydrophilicity	Area	Polarity	Hydrogen bonds donated	Hydrogen bonds accepted	Charge	Shape
ALA	A	90	0.45	115	1.6			0	1.1
CYS	C	113	3.63	135	2.0			0	3.0
ASP	D	118	13.34	150	−9.2	0	4	−1	5.0
GLU	E	142	12.59	190	−8.2	0	4	−1	5.2
PHE	F	193	3.15	210	3.7			0	12.0
GLY	G	64	0	75	1.0			0	1.0
HIS	H	159	12.66	195	−3.0	1	1	1	7.0
ILE	I	164	0.24	175	3.1			0	1.45
LYS	K	170	11.91	200	−8.8	3	0	1	8.5
LEU	L	164	0.11	170	2.8			0	1.4
MET	M	167	3.87	185	3.4			0	3.3
ASN	N	126	12.08	160	−4.8	2	2	0	5.1
PRO	P	124	11.15	145	−0.2			0	1.25
GLN	Q	142	12.08	180	−4.1	2	2	0	5.3
ARG	R	195	22.31	225	−12.3	5	0	1	8.6
SER	S	95	7.45	115	0.6	1	2	0	2.0
THR	T	121	7.27	140	1.2	1	2	0	2.1
VAL	V	139	0.40	155	2.6			0	1.3
TRP	W	231	8.27	255	1.9	1	0	0	12.15
TYR	Y	197	8.50	230	−0.7	1	1	0	12.05

Table 7: amino acid and their core properties.

also vary depending on what is surrounding the amino acid itself, so they were not taken into consideration for the fitness function. All values are represented in the following table.

2.4.2 Fitness function

The fitness function is a cumulative function were all the amino acids in the reference sequence (i.e. the one obtained before forcing mutation on the nucleotide sequence) are confronted with the respective amino acids in the same position of mutated amino acid sequences. Euclidean distance is calculated on each property of the two amino acids and multiplied by the property weight. Sum of the weighted distances is the resulting value of the fitness function for that position in the sequence:

$$D_i = \sum_j \left(p_{j,o} - p_{j,m}\right)^2 * w_j$$

Where "D" is the distance between original and mutated amino acid, meaning the value of the fitness function contribution for that specific amino acid, "p" indicates the property of the amino acids (original and mutated) and the indexes "I" and "j" refer, respectively, to position of the amino acid in the sequence and to the properties. The total fitness function score is then obtained by summing all contributions, finding the square root of the total and dividing it for the number of amino acids in the sequence:

$$FF = \frac{\sqrt{\sum_i D_i}}{i}$$

All contributions are summed up to give a total value of the fitness function for that specific sequence, resulting in fifty values per genetic code. Since the more distant the amino acids are, the higher the value of the function, lower values indicate stronger stability of the code towards nucleotide mutations.

It is necessary to consider that the biggest damage that an amino acid sequence can suffer after a point mutation is an alteration on start and stop codons. There are four possible cases: a start codon or a stop codon mutate into some other amino acid, or a different amino acid becomes a start or a stop codon. This would completely alter protein synthesis during ribosomal translation of genetic sequences, so these particular cases are handled differently and an arbitrary value of 30 is assigned to the sum of contributions. This value is much greater than the average values obtained from amino acid that do not include start or stop mutations, hence giving strong relevance to these alterations.

2.4.3 Number of mutations

Another evaluated feature of the genetic codes is the number of mutations they cause to the original genomes. This part of the work does not consider distance between different amino acids but how the redundancy of the code opposes to an effective mutation. Two different types of mutation are considered: start/stop mutations and body mutations. The first refer to mutations occurring on start or stop codons and is considered to be the worst case, because it either shuts down the protein production or starts the production of a meaningless or useless protein, causing a waste of resources. Any other mutation will instead be considered as a body mutation. These quantities are calculated over every genome per every code and averaged along codes, so 5 curves mutations/cycles of mutations are obtained, one per genetic code.

The most interesting information that may be extrapolated from this analysis is to check if genetic code has some way of “disfavouring” the most dangerous mutations. In order to evaluate this, probability of mutations occurring on start/stop codons with respect to those occurring elsewhere must be calculated.

Possible body mutations are obtained by multiplying the number of body amino acids by the number of different body amino acids, hence:

$$BM = N_b * (N_b - 1)$$

Where N_b is the number of amino acids not coding for start or stop, 19. The result obtained is:

$$BM = 19 * 18 = 342$$

Total mutation multiplicity is instead obtained from the analogue formula:

$$TM = N * (N - 1)$$

Where N is 21, number of possible amino acids plus the stop signal, because an amino acid or a start/stop sequence can mutate into every other amino acid. The result is:

$$TM = 21 * 20 = 420$$

At this point, the number of possible start/stop mutations can be written as:

$$SM = TM - BM = 78$$

This analysis serves to understand if redundancy of amino acids is set in a way that allows less start/stop mutations than other mutations. Results are obtained by multiplying the number of start/stop mutations by the following coefficient:

$$K_{sb} = \frac{BM}{SM} = 4.384$$

At this point, two curves are plotted and compared: body mutations and start/stop mutations multiplied by K_{sb}.

3 RESULTS

3.1 ENTROPY

Results are plotted for every of the fifty saved mutated sequences, one every one hundred cycles of mutations. The first part of the results is about Shannon's entropy.

3.1.1 Entropy of different codes

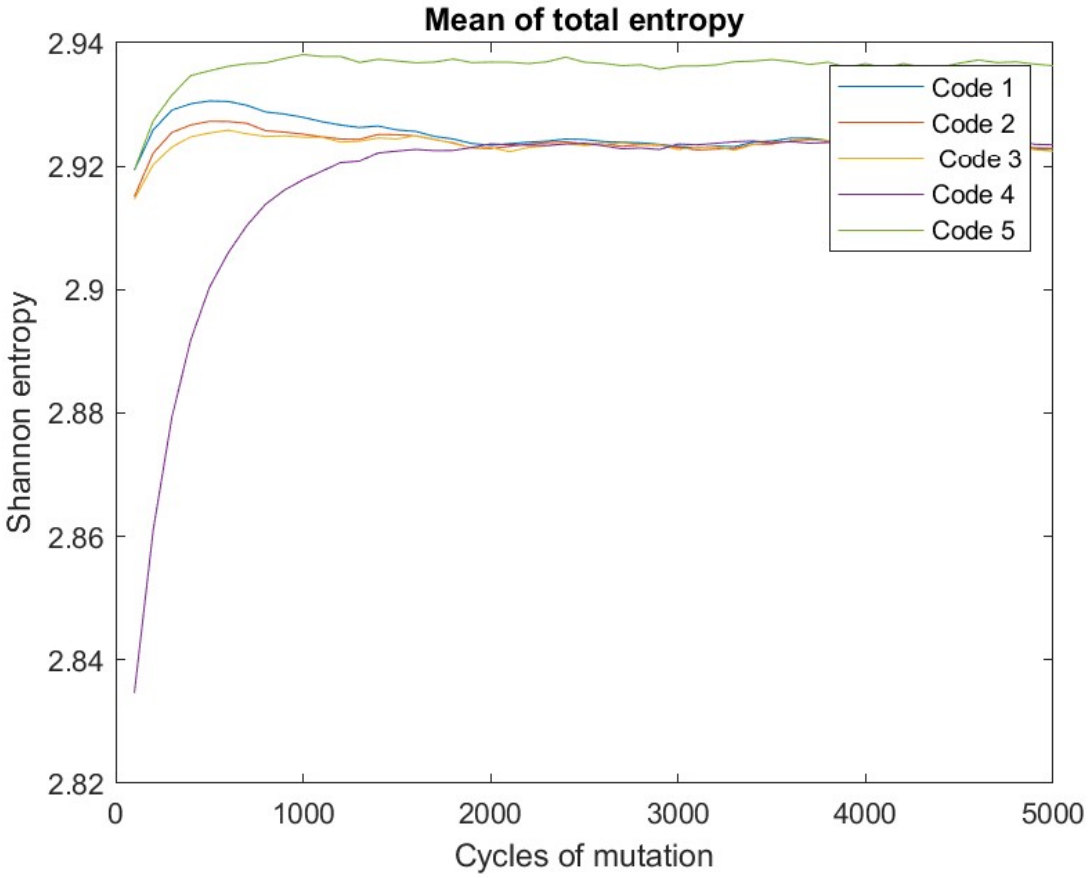

Figure 3: Mean of total entropy of different genetic codes.

For all five codes, entropy of the amino acid sequences for every specie is averaged throughout the fifty steps of mutations to obtain a single plot per genetic code. A higher value indicates a more uniform distribution of amino acids. Code 5, the random code, has the highest entropy value of all codes. This was quite predictable, because the lack of organization in the code brings to more random mutations and suggest a higher entropy value may be reached. It is important to note that the plateau reached by four of the codes is corresponding to the

Shannon's entropy if all codons are equiprobable, calculated considering the redundancy of triplets of bases corresponding to the same amino acid:

$$S_{DNA} = -\sum_{i} p_{(i)} * \log p_{(i)} = 2.9238$$

Where $p_{(i)}$ corresponds to the probability of occurrence of all amino acids if all codons are supposed to be equiprobable.

Random genetic code shows an offset of 0.018 at the plateau, placing at a higher value than the expected DNA entropy given a certain redundancy. It is also interesting to note that code 4 (swapped squares code) starts from a way lower amount of entropy. This may be caused by the fact that genomes have a similar distribution of amino acids, determined by the ACTG sequences, and by mistranslating those sequences we obtain a wrong distribution of amino acids. The same observation could, however, be made for the random code (code 5), but this one does not show the same behaviour.

Moreover, codes 1, 2 and 3 start from a value of entropy which is pretty close to the value reached at the plateau. The first cycles of mutations make entropy increase (flatter distribution of amino acids) but then the value goes back to where it started. This may be caused by the large number of mutations occurring, because the process of mutation is not biased and hence mutations may start to "compensate" each other after many cycles.

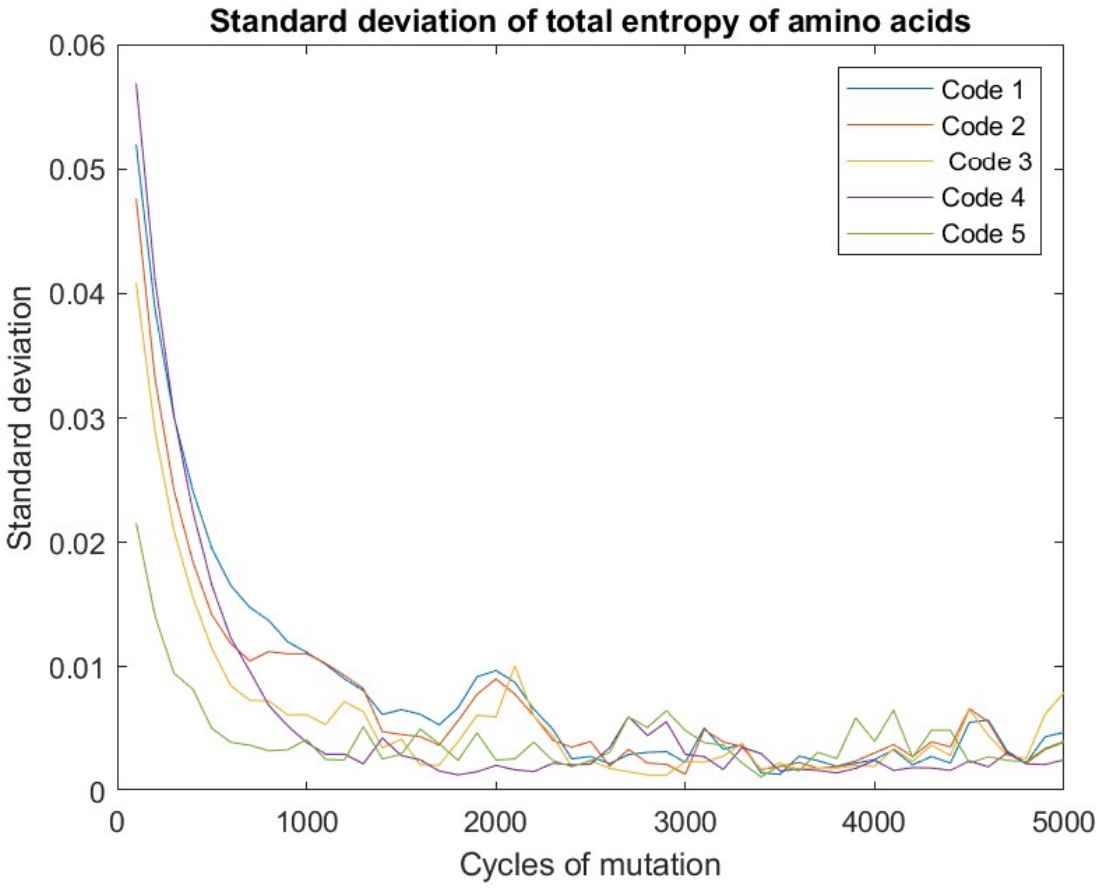

Figure 4: standard deviation of entropy mediated over all genomes of every code.

Standard deviation values do not differ much from each other. Code 5 has the lowest initial standard deviation, as expected from the results shown in figure 2 because of the more uniform distribution of amino acids due to the randomness of the code. All standard deviations oscillate while cycles of mutation continue, but within a small range of values. This shows that mutating a DNA sequence several times leads to a very similar distribution of the amino acids across different species.

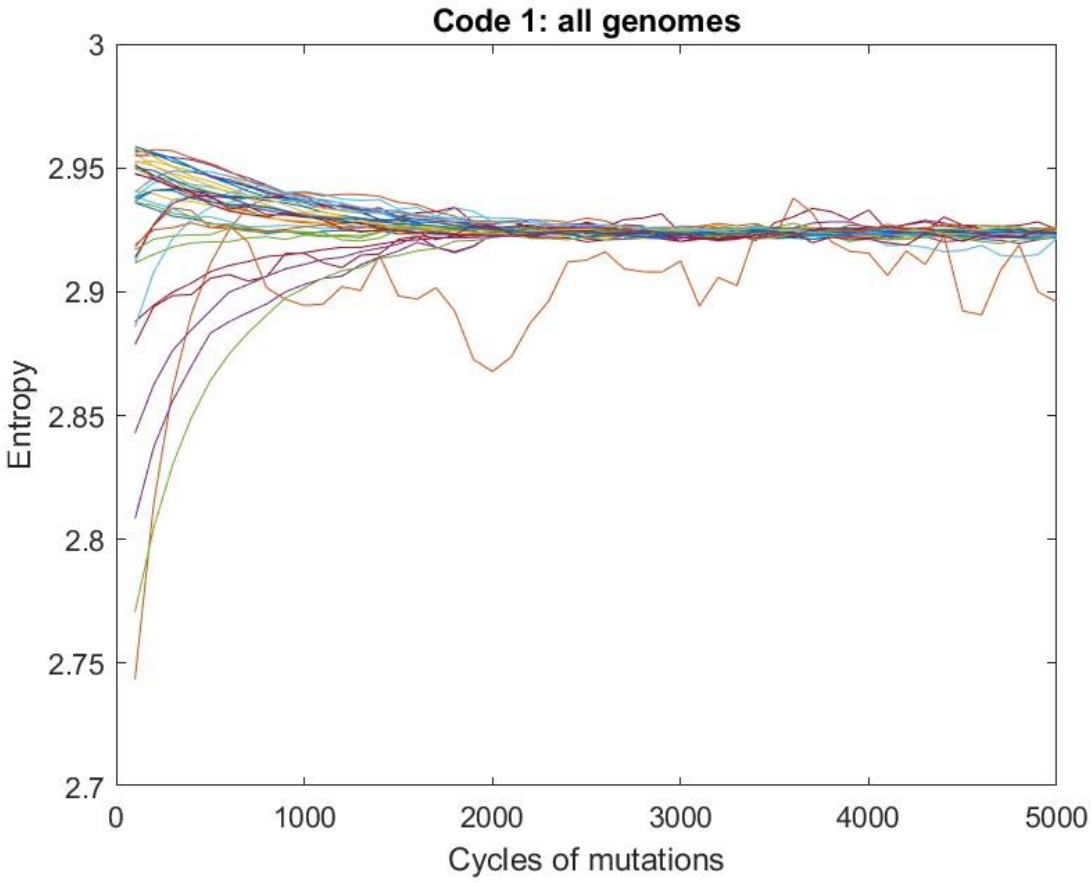

Figure 5: entropy of all genomes translated with the original genetic code.

All genomes reach a certain value after a high number of steps in the mutation process, close the expected value of 2.9238. Few organisms have initial values (before mutations) lower than this plateau: these organisms are (from lower initial value):

- Daldinia concentrica;
- Luteitalea pratensis;
- Leishmania Mexicana;
- Pan troglodytes.

Two of these are fungi, one belongs to protista and one is a bacterium, all relatively simple and ancient organisms.

Most species, instead, start from values higher than the completely uniform distribution of codons: this means that codons distribution is not strictly following the redundancy of the genetic code, but it is more “spread”, i.e. an amino acid with redundancy of 6 will not happen 6 times more often than one with no redundancy, but less: this leads to a more uniform distribution of amino acids.

There is one genome that shows heavy oscillatory behaviour during the cycles of mutations, this belongs to Hepatitis B virus. The reason for this behaviour is that the genome length is

1082 amino acids compared to an average length of the order of 10^6 amino acids per genome. A single mutation can hence strongly affect the calculated quantity.

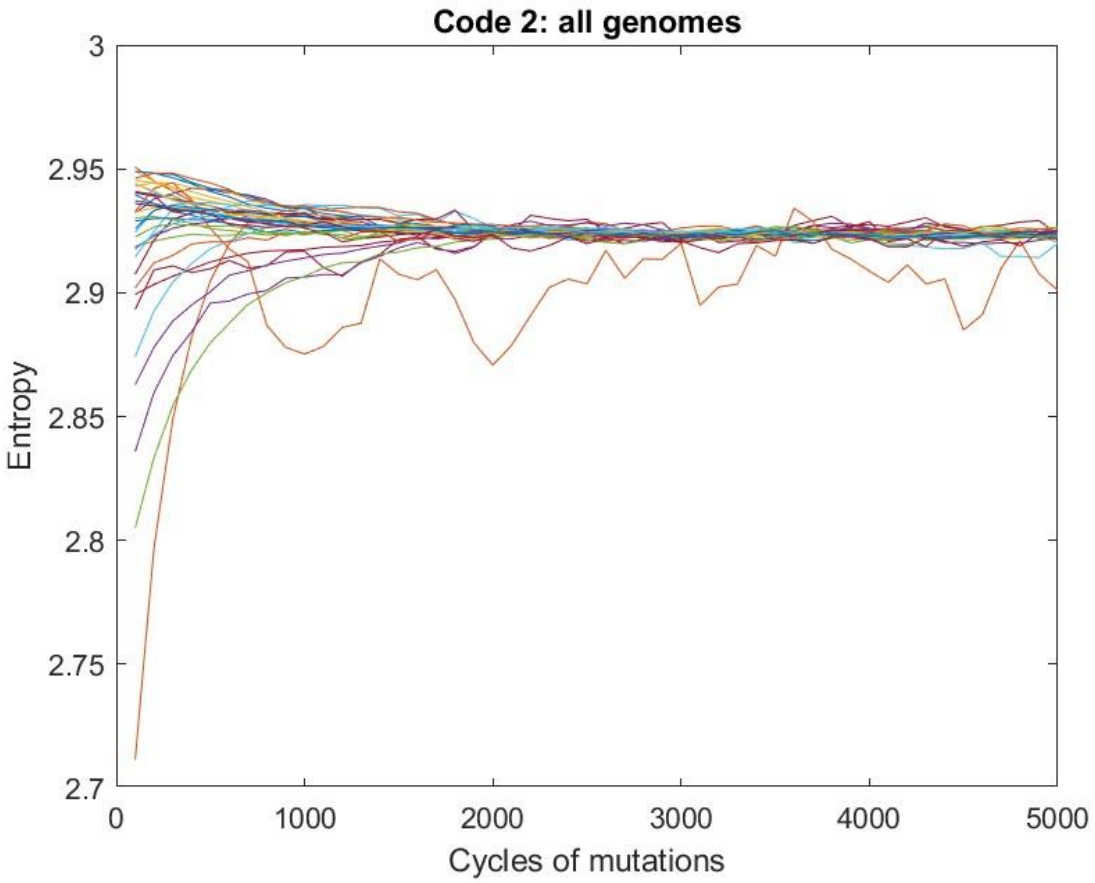

Figure 6: entropy of all genomes translated with genetic code 2.

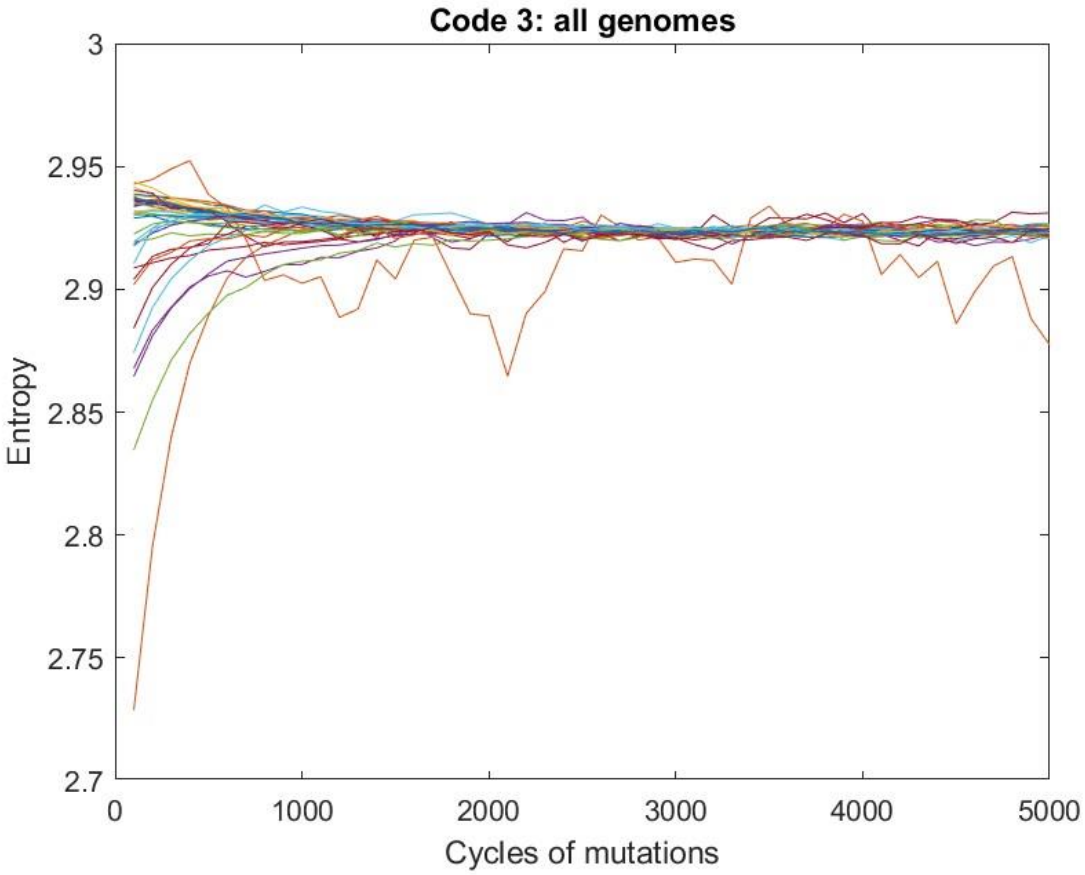

Figure 7: entropy of all genomes translated with genetic code 3.

Considerations on figures 6 and 7 are similar to those referring to the original genetic code. Genomes starting from lower values are the same as the original code. Reached plateau has the same value, as expected since redundancy of amino acids was conserved. Case 3 shows slightly bigger oscillations with respect to the plateau, as it is more disordered than the first 2 cases. Oscillations and values are still very much comparable. Organisms with low starting entropy are the same as case 1, suggesting that this effect is due to the specific genomes, more than to the code itself.

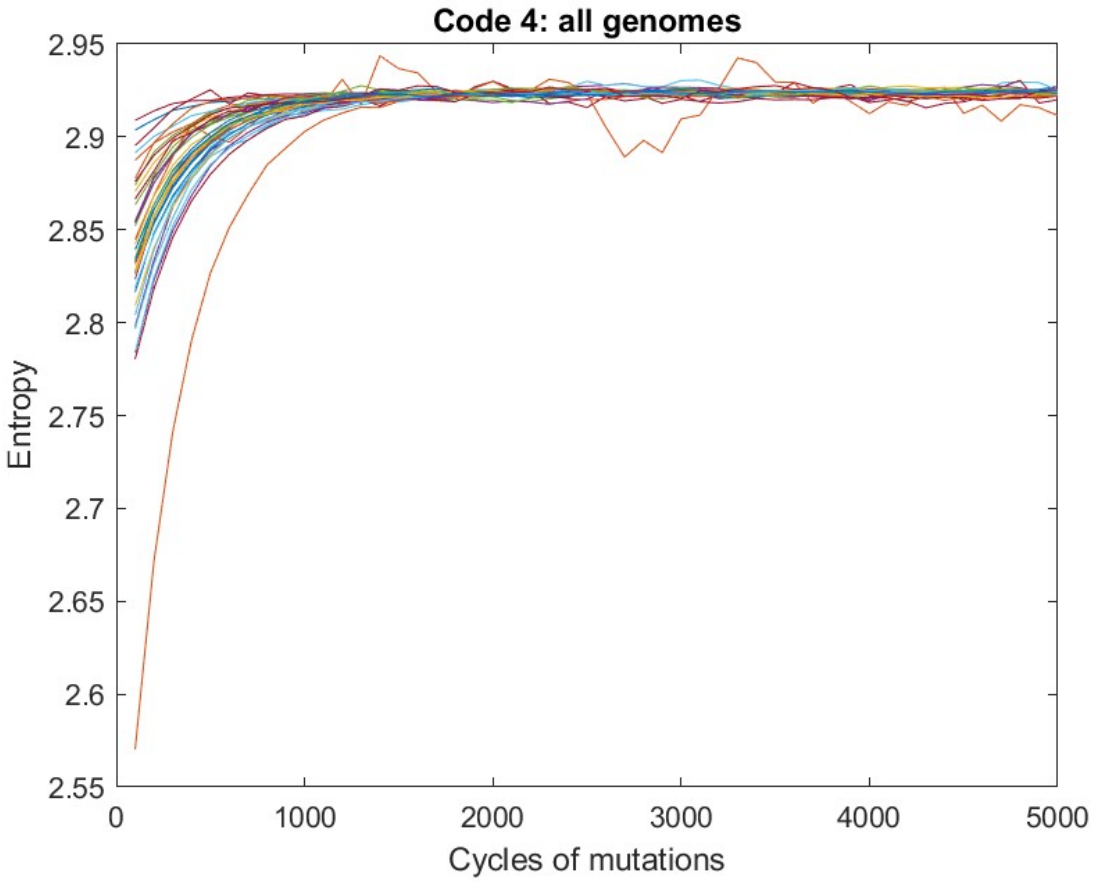

Figure 8: entropy of all genomes translated with genetic code 4.

Figure 7 shows that the swapped squares code has relevant effects on the entropy of amino acids. Unlike other codes, in fact, all entropy values start from a lower value than the plateau. Also, the lowest starting entropy is way lower than in other cases. This shows the importance of having a structured genetic code, not only in the sense that redundant codons must be close to each other.

However, the plateau is the same and after 2000 cycles of mutations the behaviour of code 4 is the same of codes 1, 2 and 3.

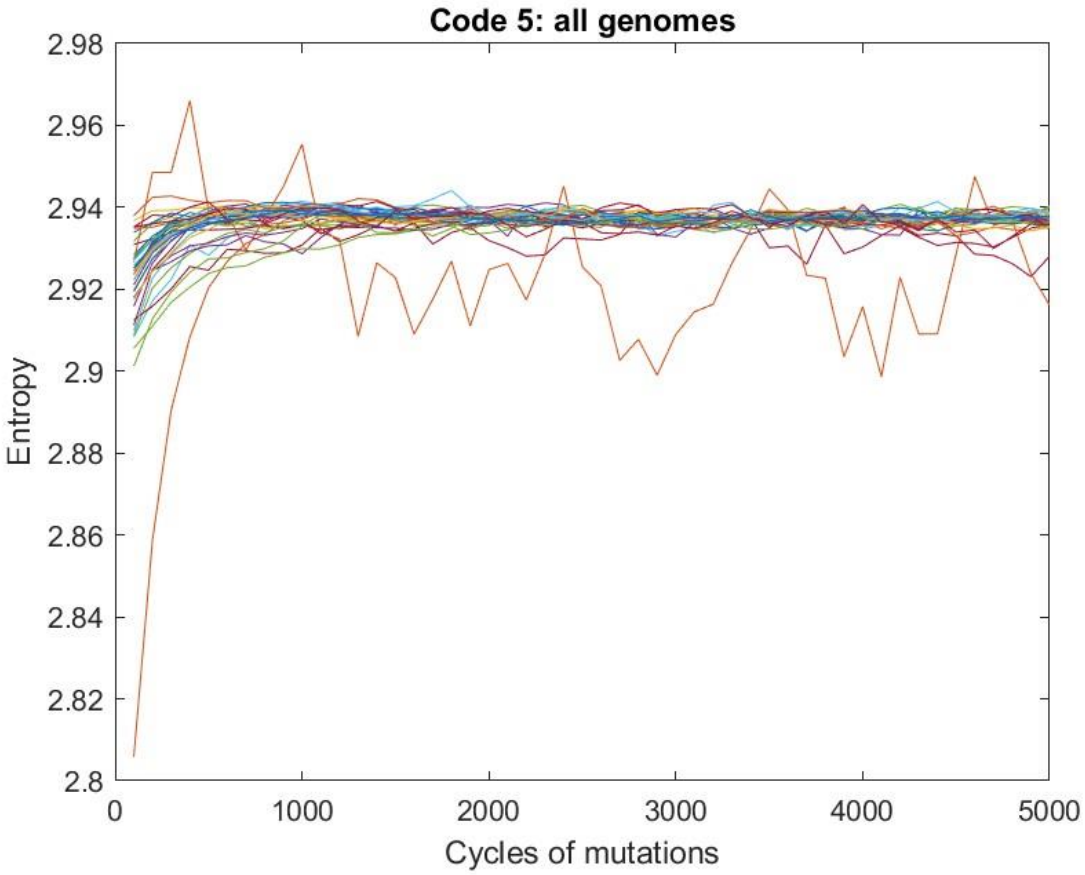

Figure 9: entropy of all genomes translated with genetic code 5.

Using the random genetic code, all entropies start from below the final plateau value, with only one genome (Daldinia concentrica) having a much lower value with respect to the others. It is important to notice that that value is still higher than in the other cases, where the lowest value ranges from 2.57 (code 4) to 2.74 (code 1). Moreover, all other genomes start from values above 2.9. Also, it shows some more evident irregularity of a few genomes after the plateau is reached. As expected from the results in figure 2, the reached plateau has a higher value than for other codes.

These results show that in the first 1000 cycles of mutations using the standard genetic code most genomes have higher entropy value, i.e. a flatter distribution of amino acids. This amount is higher than the plateau reached by all genetic codes. Moreover, every genome reached the same entropy value after 2000 cycles of mutations, but almost none of them started at that value.

These two considerations suggest that living organisms have some strategy to keep an uneven distribution of amino acids, because if that was not the case then random mutations occurring over around 40.000 of years (2000 cycles with a supposed breeding age of 20 years) would else have led to well distributed codons for every specie with same redundancy in the genetic code.

3.1.2 Entropy in different kingdoms

In this section entropy will be analysed dividing genomes into different branches of the tree of life.

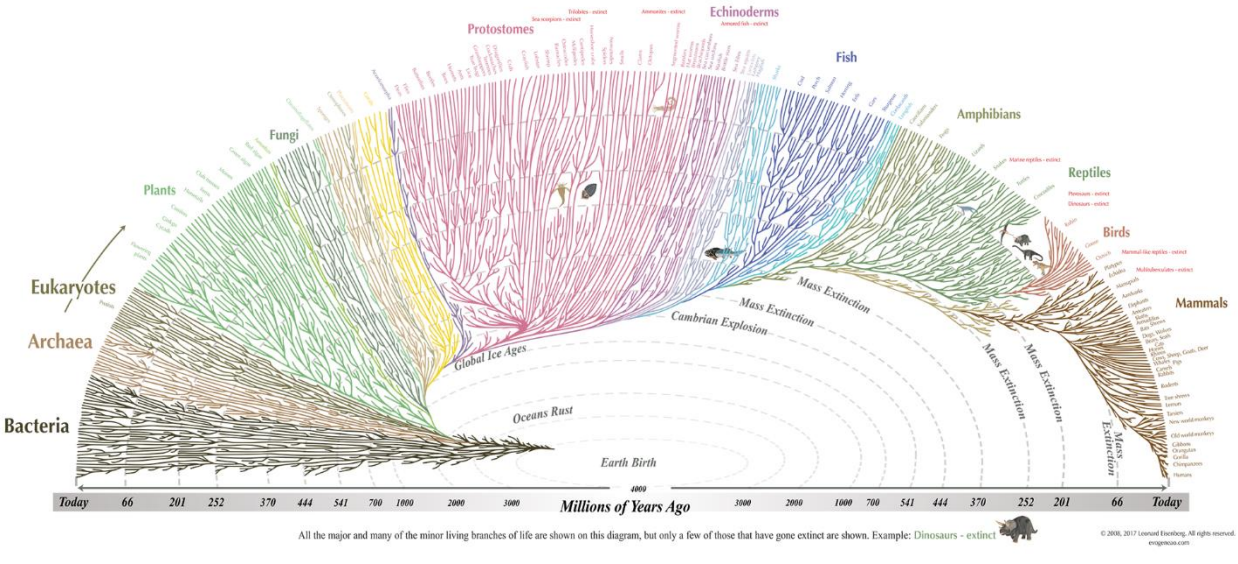

Figure 10: tree of life.

The tree of life represents evolution of different families of living organisms, highlighting the evolutionary connection between kingdoms and when differentiation occurred.

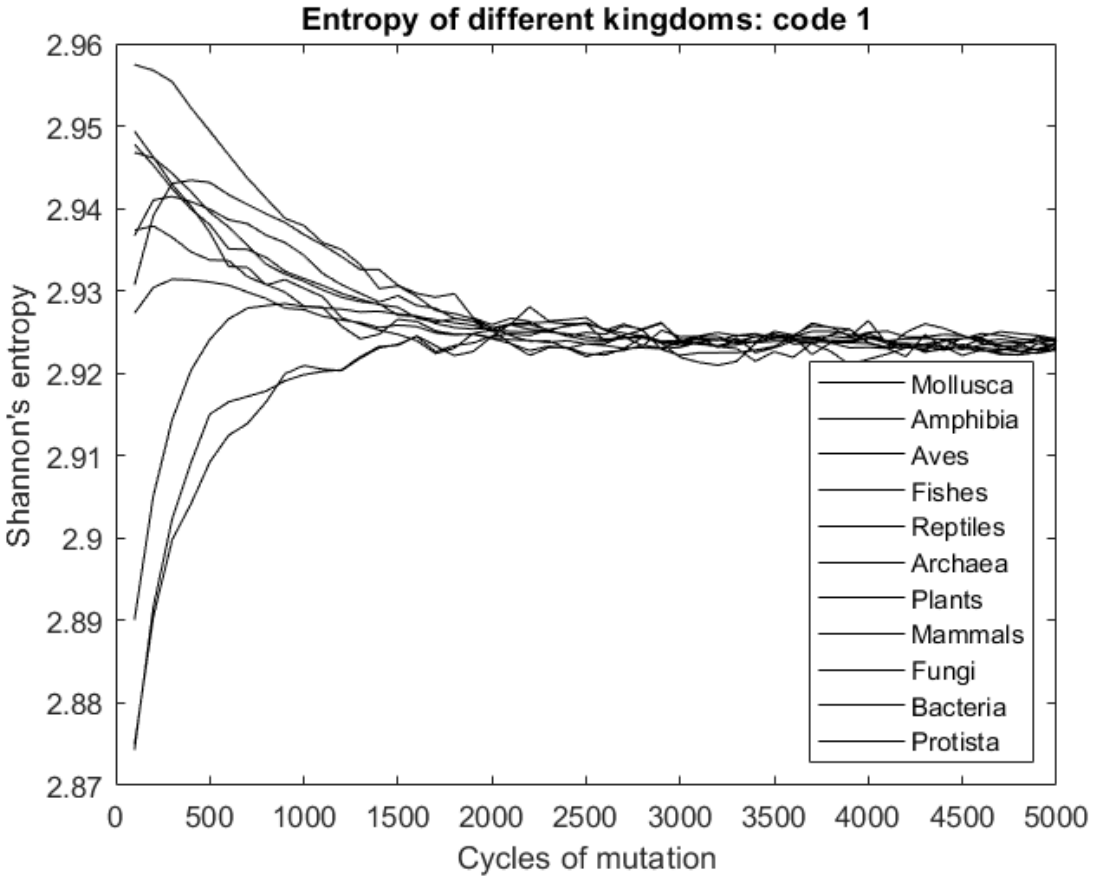

Figure 11: entropy of different kingdoms using the standard genetic code.

Note that in this section, plots will have a legend with different kingdoms in order from the higher initial value to the lowest initial value. As an example, in this figure Mollusca will be represented by the function with the highest initial value and Protista by the one with the lowest.

Figure 11 shows how higher initial entropies (which are relevant because they are close to the entropy of the genomes acquired from GenBank) belong to the right side of the tree of life including it all, except for mammals which still start from an entropy close to the higher classes. Lower starting entropy organisms, instead, cover the left part of the tree. The trend is clear, even if not all classes strictly follow this sorting.

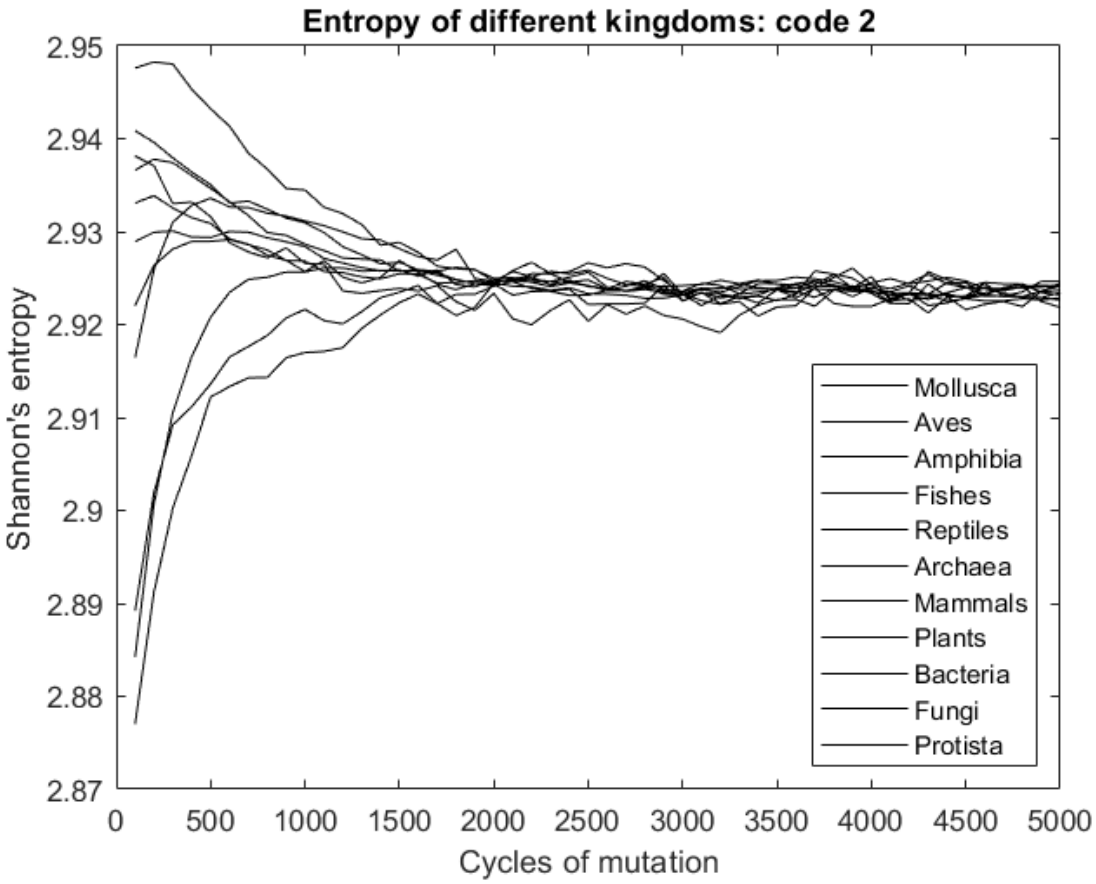

Figure 12: entropy of different kingdoms using genetic code 2.

Entropy of organisms obtained with genetic code 2 is very similar to the one coming from the standard code. Some values are slightly lower, but the trend of having ancient branches of the tree of life starting from lower entropies while most recent kingdoms have higher entropy is still evident.

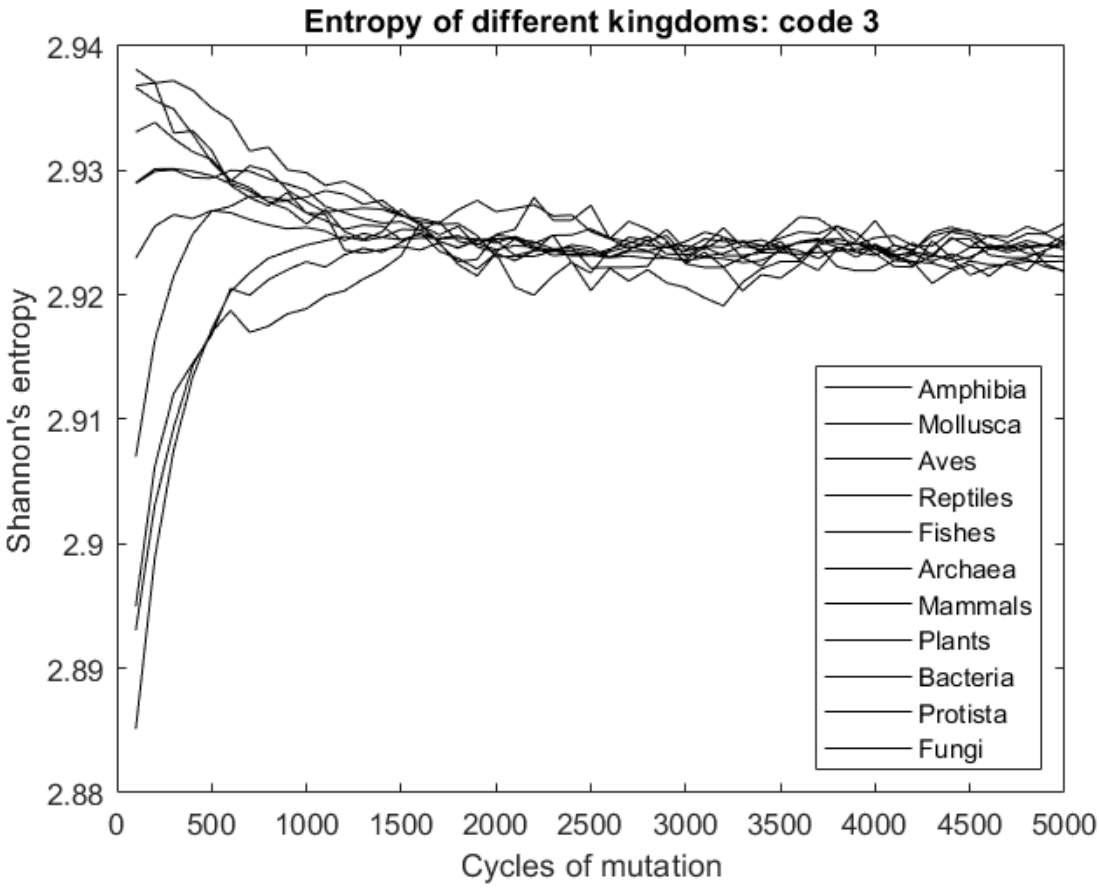

Figure 13: entropy of different kingdoms using genetic code 3.

Considerations are very similar to those for genetic code 2. Values lowered slightly more, but the trend is still confirmed for code 3.

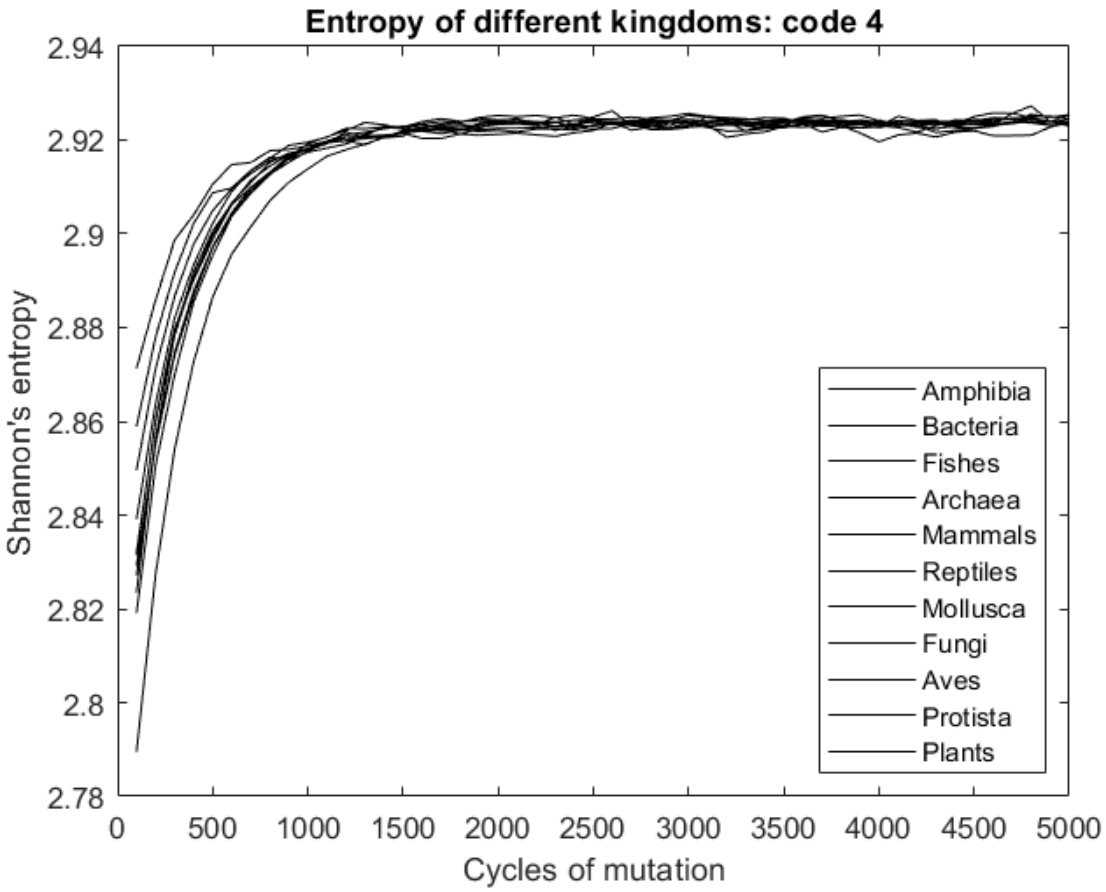

Figure 14: entropy of different kingdoms using genetic code 4.

If the code is heavily changed, as for code 4, entropy behaves in a very different manner. All initial values are below the plateau and the previously observed trend is not conserved.

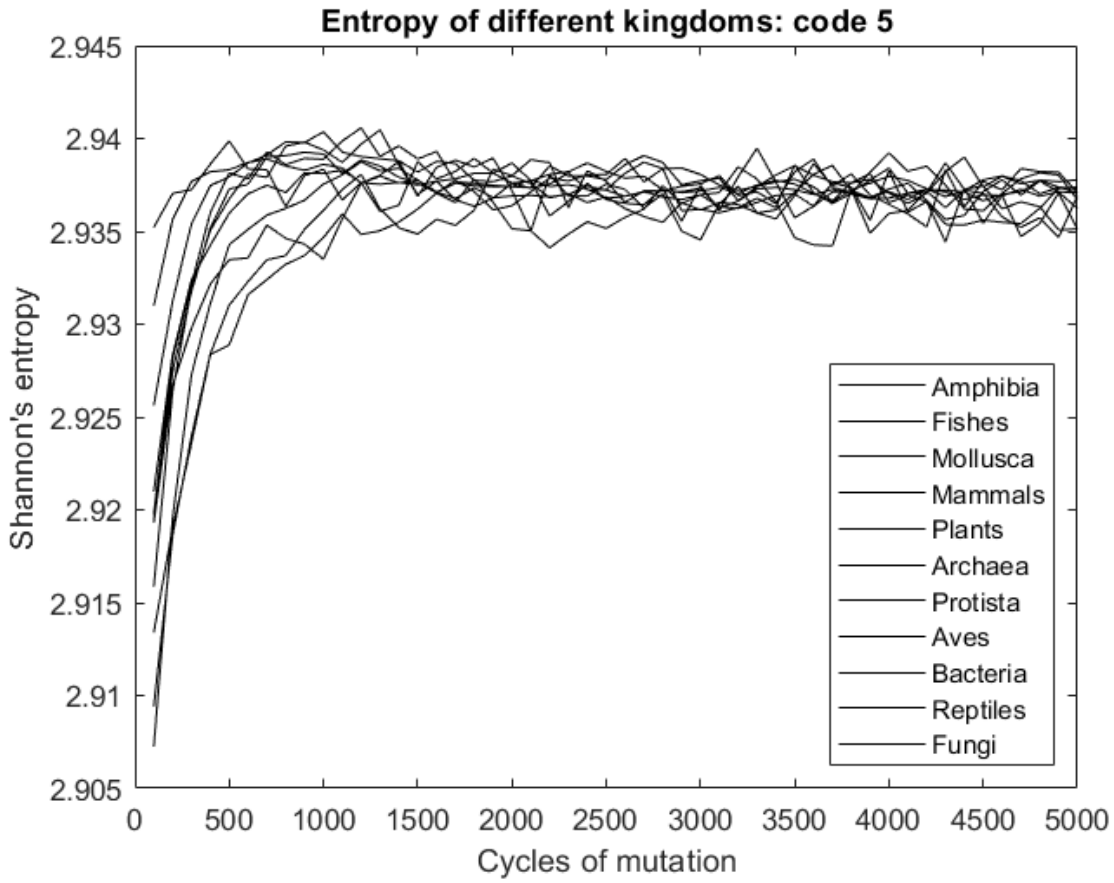

Figure 15: entropy of different kingdoms using genetic code 5.

The plots obtained using the random genetic code has features that are more similar to those of genetic code 4 than to those of the others, again suggesting that swapping squares removes the characteristics of the genetic code and leading to results that are more similar to the random codon assignment.

From an evolutionary point of view, only the first 3 codes are relevant because they maintain the structure of the genetic code. The trend of more ancient kingdoms having a lower entropy value is evident in all 3 genetic codes and it is coherent with the idea that evolution brings higher DNA entropy, i.e. flatter distributions of amino acids. This also leads to the conclusion that genetic code has a specific structure that can maintain entropic properties if small changes are made, as long as the overall structure is kept the same. But if the structure itself is denied, genetic code starts behaving in a very different manner. This may introduce the idea that the important part about the genetic code is the overall arrangement of the blocks, i.e. the relative position on the table of the amino acids basing on the first two letters of the codons.

3.2 FITNESS FUNCTION

Fitness function is evaluated to explore the physicochemical theory. Euclidean distances between amino acids are considered and the total score of the fitness function is a measure of how distant mutated amino acids are from their original state. The final score is divided by the length of the amino acid sequence, hence it is an average score per amino acid. The higher the score, the less performing the code is in terms of minimizing the meaning of mutations. As for the previous section, results are evaluated on every genome together and on genomes clustered by kingdom.

3.2.1 Fitness function of different codes

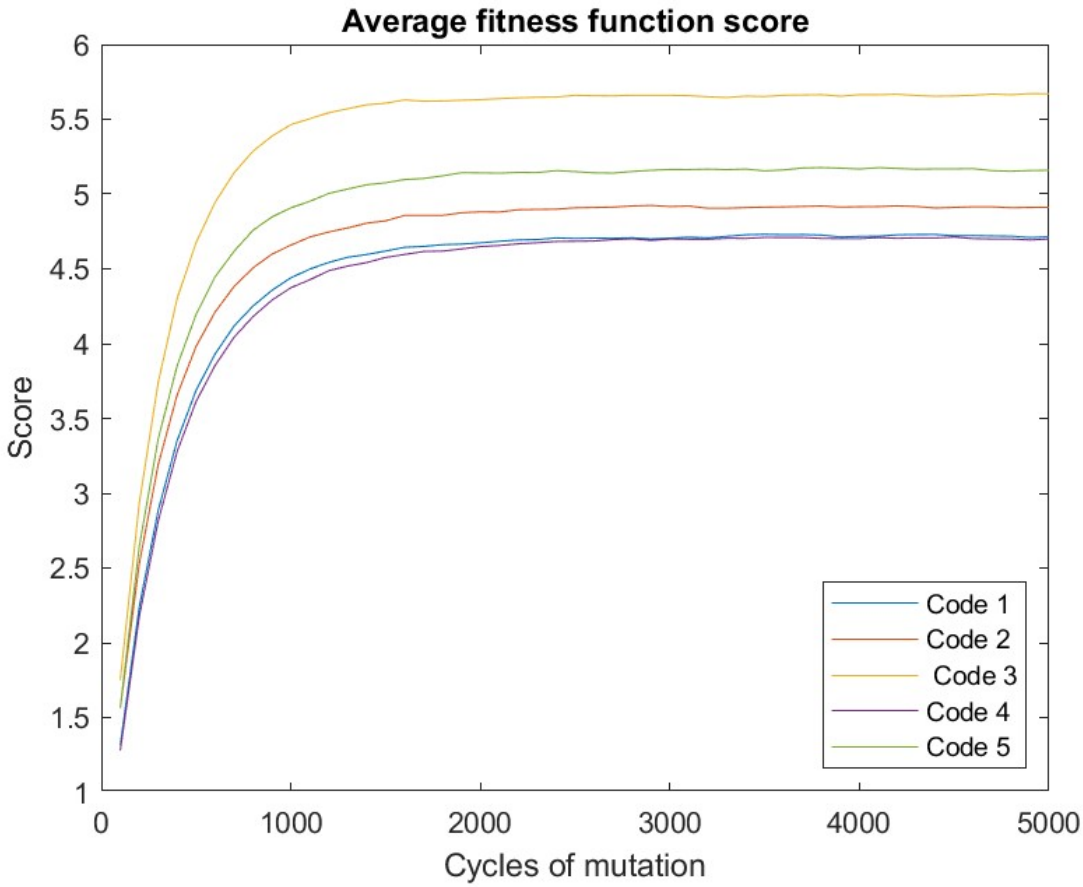

Figure 16: average fitness function scores of genetic codes.

All codes have a fitness function with similar shape, with fast increase in the score in the first 1000 cycles of mutations followed by a plateau. The plateau is the most interesting part: if the function reaches a plateau it means that most of the amino acids are mutated and, because of the high number of mutations, the scores are “at equilibrium”. Equilibrium is intended as a dynamic equilibrium, where amino acids keep mutating but the fitness function contributions compensate each other. As an example, if one mutated amino acid increases the score by 6, another mutation with the opposite effect can be found later in the sequence. If this

assumption is correct, the concept can be extended to all possible genetic codes. In this way, once a fitness function is created and applied to different genetic codes, scores at plateau can be used to determine relative organization of the different codes from a physicochemical point of view in a quantitative manner. The most interesting part of this approach is that it only requires a reasonable fitness function to evaluate relative properties of genetic codes, not depending on the processed genomes.

The figure shows that standard the genetic code and the swapped squares code are the most resilient to mutation. This was easily anticipable for the standard genetic code, since it is the core of the physicochemical hypothesis.

For what concerns code 4, instead, some reasoning must be done. Fitness function calculates the mean distance between amino acids that mutate and the original amino acids at the same position in the sequence. In other words, it calculates how important amino acids mutations were. When swapping squares of the genetic code table, only the distance between most of the third letters is conserved, but this should be just a small part of the fitness function contribution. This means that mean distance between mutated amino acids is conserved even if the squares are shuffled. Further analyses will be necessary to determine whether this was just a coincidence or if genetic code is immune to square swapping from a physicochemical point of view, but this result suggests that the third letter order is, by far, the most relevant.

Code 2 is probably affected by a higher number of mutations on start/stop codons because stop codons are less prone to mutate into another stop codon. Since they give the greatest contribution to the fitness function, this could explain the higher score. Code 3 seems to confirm that because it has 4 stop codons, so even more probability to encounter a start/stop mutation.

Code 5 can be seen as a sort of reference, because it has no built in symmetry of any other way to minimize the effects of mutations. It only has 3 stop codons, so its plateau is between code 2 and code 3.

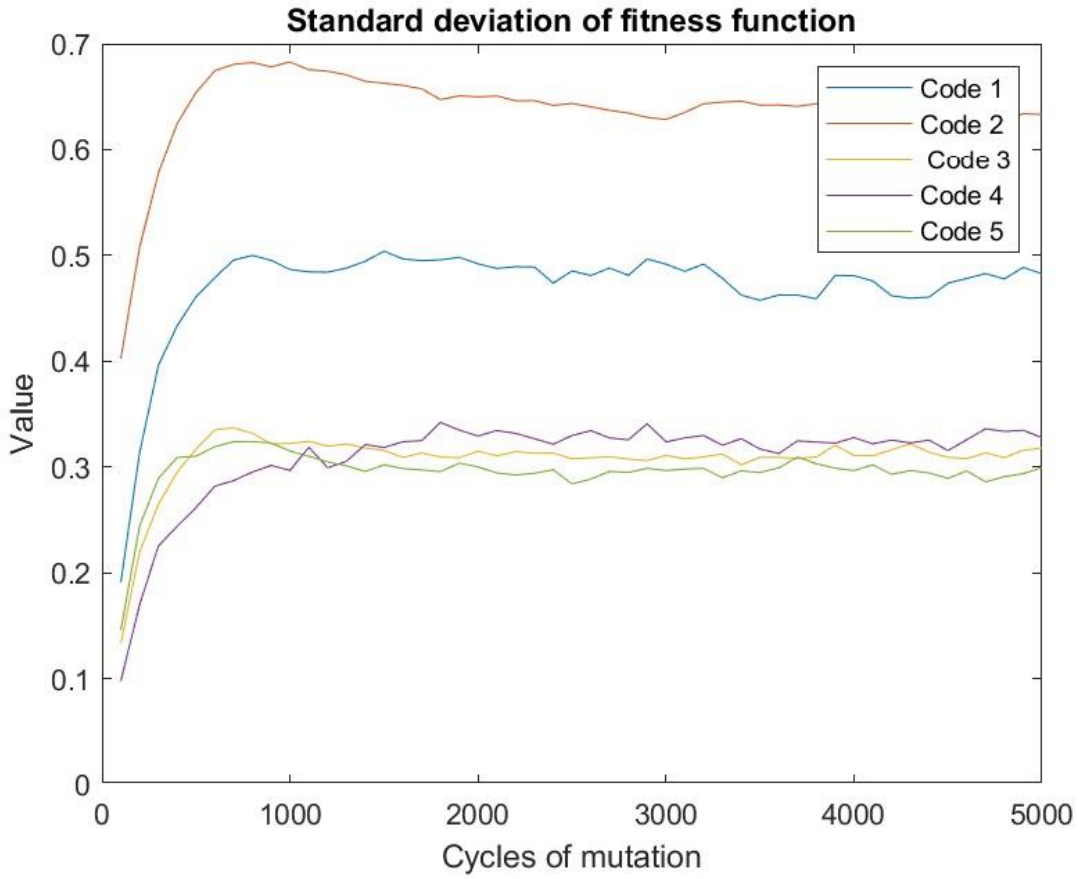

Figure 17: standard deviation of fitness function.

Standard deviation of fitness function is a measure of how stable a genetic code is in terms of physicochemical solidity while processing different genomes. Code 2 shows the highest variability, followed by the standard genetic code. Extracting conclusions from this plot is very difficult and may lead to wrong conclusions, because it is hard to give a biological meaning to standard deviations of physicochemical properties of genetic codes. It is still interesting how the 3 more disordered codes show the same values.

3.2.2 Number of mutations

Another evaluated parameter is the number of mutations occurred. This is meant to calculate how many mutations are not silent, i.e. different codons not coding for the same amino acid. Contrarily to the fitness function, distance between amino acids is not considered here and only the number of mutations matters. This is important because result will not be dependent on the chosen fitness function.

Mutations have been divided in 2 classes: start/stop mutations, namely mutations occurring at start or stop codons, and body mutations, including every other mutation. Start/stop mutations are more relevant since they cause the greatest damage to the protein during protein synthesis.

Both quantities are divided by the total length of the genome, so the result is the number of mutations per amino acid.

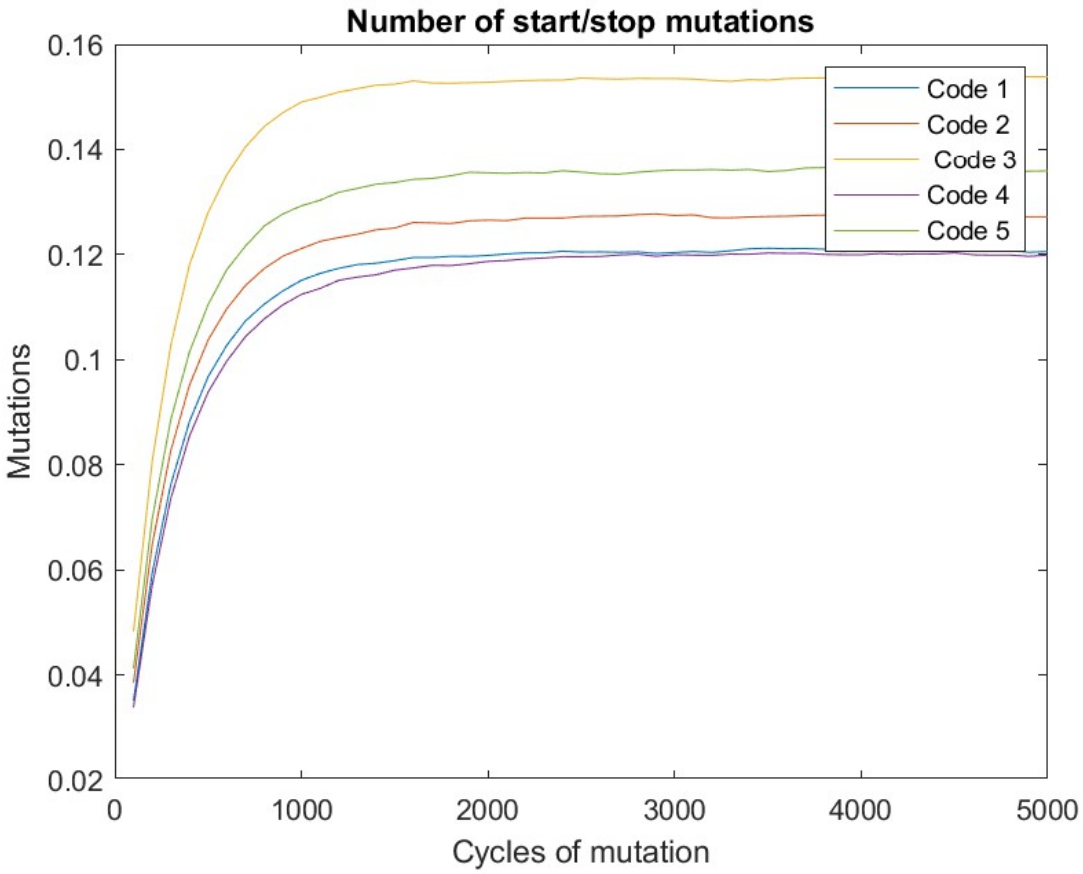

Figure 18: number of start/stop mutations of different genetic codes.

As expected, the highest number of start/stop mutations is found in code 3 because of the extra stop codon inserted. To analyse the behaviour of other codes, the concept of "mutation distance" is introduced: one mutation step is a single point mutation altering a specific triplet. Distance between 2 codons is the number of mutation steps that are necessary to go from one triplet to the other. In code 1, as an example, the stop signal is brought by 3 triplets: UAA, UAG, UGA. These triplets are one mutation step away from each other, meaning that chances that a stop codon mutates into another stop codon are relevant: out of the 9 possible single point alterations a codon can encounter, 2 of them are stop codons themselves. The same distances are found in code 4, where stop codons are ACA, AGA and AGG. This justifies the lower number of start/stop mutations in these codes.

In code 2 stop codons are: UUG, CCG and GGU, while for code 5 ACG, GCU and GGA are found. For each of these 2 codes we can define 3 different mutation distances between stop signals: in code 2 distances are 2, 3 and 3. In code 5 2, 2 and 3. Since total distance is higher for code 2 than for code 5, expected results would place code 2 above code 5 in the previous plot because there are higher chances that a stop codon mutates into another stop codon in code 5, but this is not the case. Because of the very high number of mutations occurred (remember

that this is an average on 36 genomes, each with more than 10^6 amino acids) it is hard to think that this is just a coincidence, but further research is necessary to clarify things out.

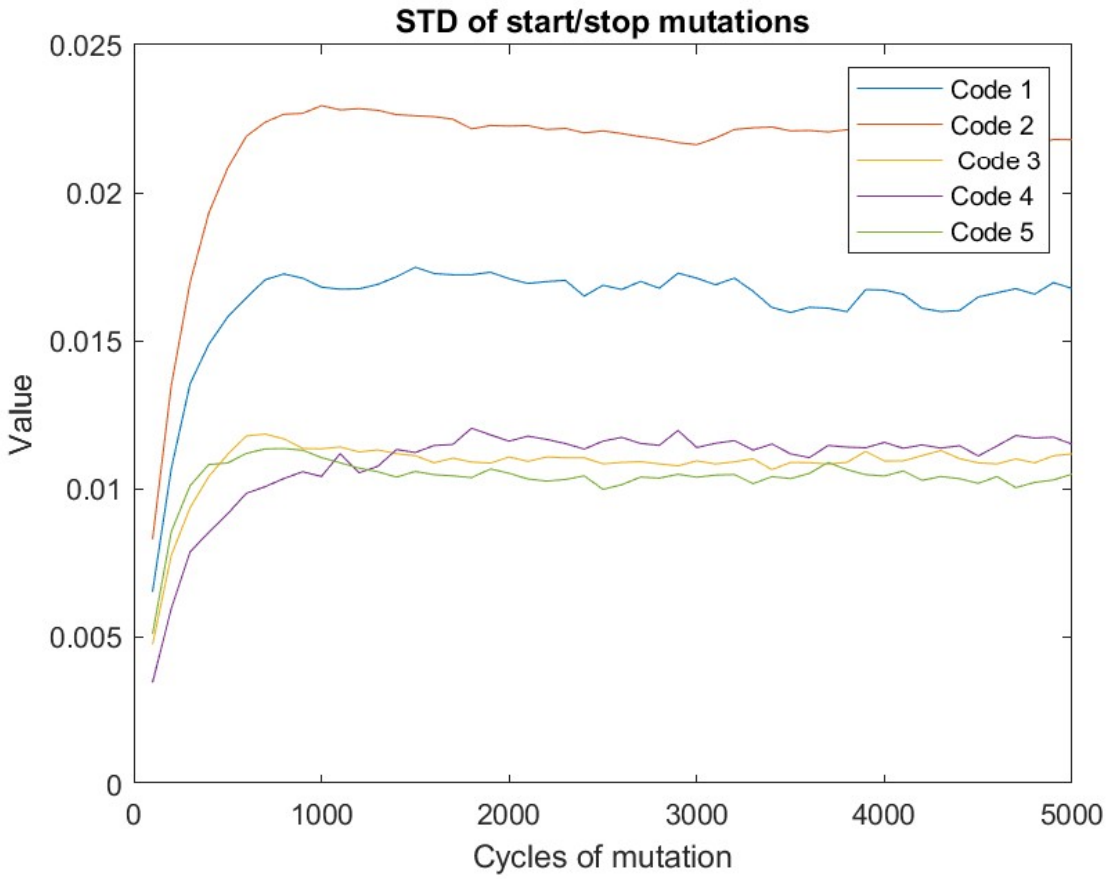

Figure 19: standard deviation of start/stop mutations of different codes.

All curves in this plot have the same shape of the fitness function standard deviation in figure 17. This is probably due to the high fitness function score associated with start/stop mutations: these mutations, in fact, contribute to the most part of the fitness function score. Because of the high number of mutations considered and averaged, it makes sense that start/stop mutations are those who influence the score most.

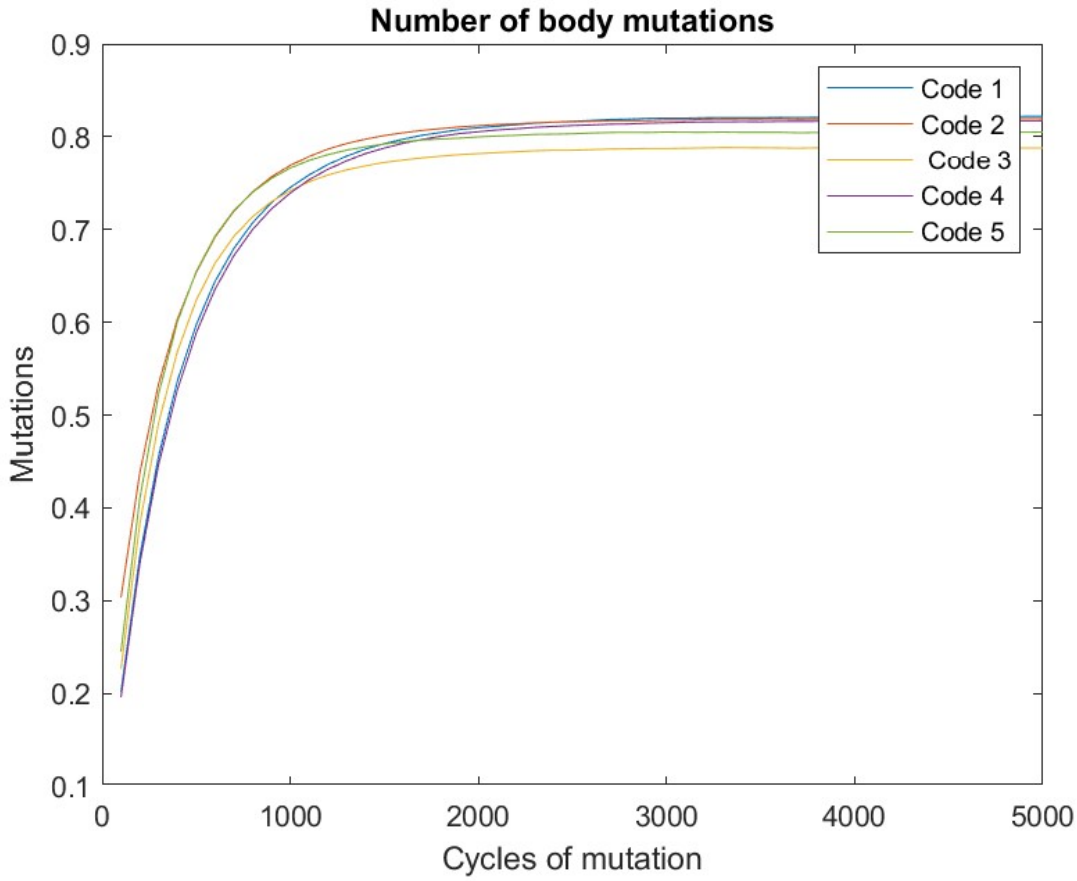

Figure 20: number of body mutations of different genetic codes.

Figure 20 shows the number of body mutations per amino acid in different codes. All codes reach a very similar score after a big number of cycles, but code 1 and 4 have the highest ratio of mutated body amino acids. Anyway, the number of body mutations in the first 1000 cycles sees code 1 and 4 to be the lowest, so a similar consideration to the one made for entropy can be made: if living organisms are able to keep their genome stable through thousands of years of mutations, then only the first part of this plot is relevant and the standard genetic code results in a lower number of mutations than most of the other codes. Interestingly, code 4 is showing the same property, hence suggesting that squares position in the genetic code may not be very relevant in terms of redundancy of amino acids, as expected.

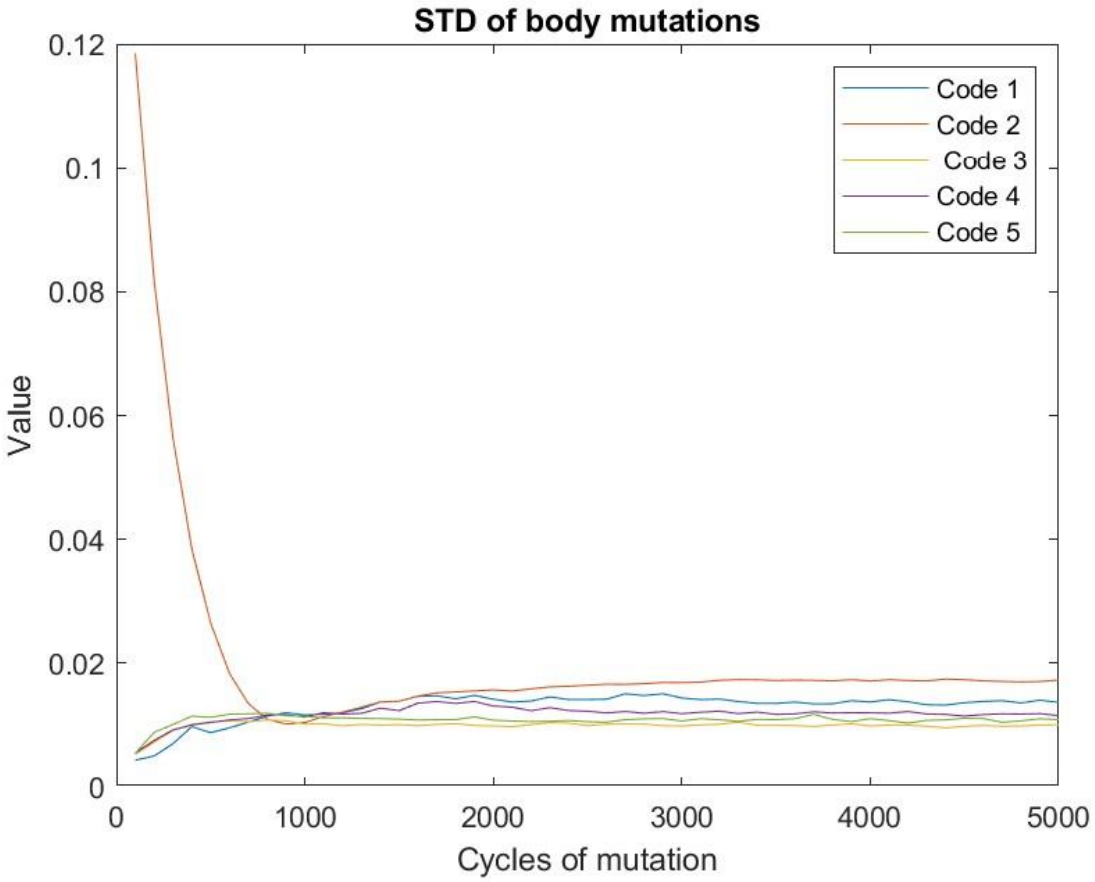

Figure 21: standard deviation of body mutations of different genetic codes.

Standard deviation is very low and stable for all the considered codes, except for the first 1000 cycles of mutations of code 2. A possible explanation could be the relatively low number of genomes taken into consideration (36), so that casualties may cause a much higher amount of standard deviation before things get evened out by mutations, but even considering only the last portion of the plot code 2 shows significantly higher standard deviation, meaning that this code does not behave in the same way with different genomes. Also, the difference in the first steps is of more than one order of magnitude: it is hard to justify this just by casualties, but again more studies are required to solve this enigma.

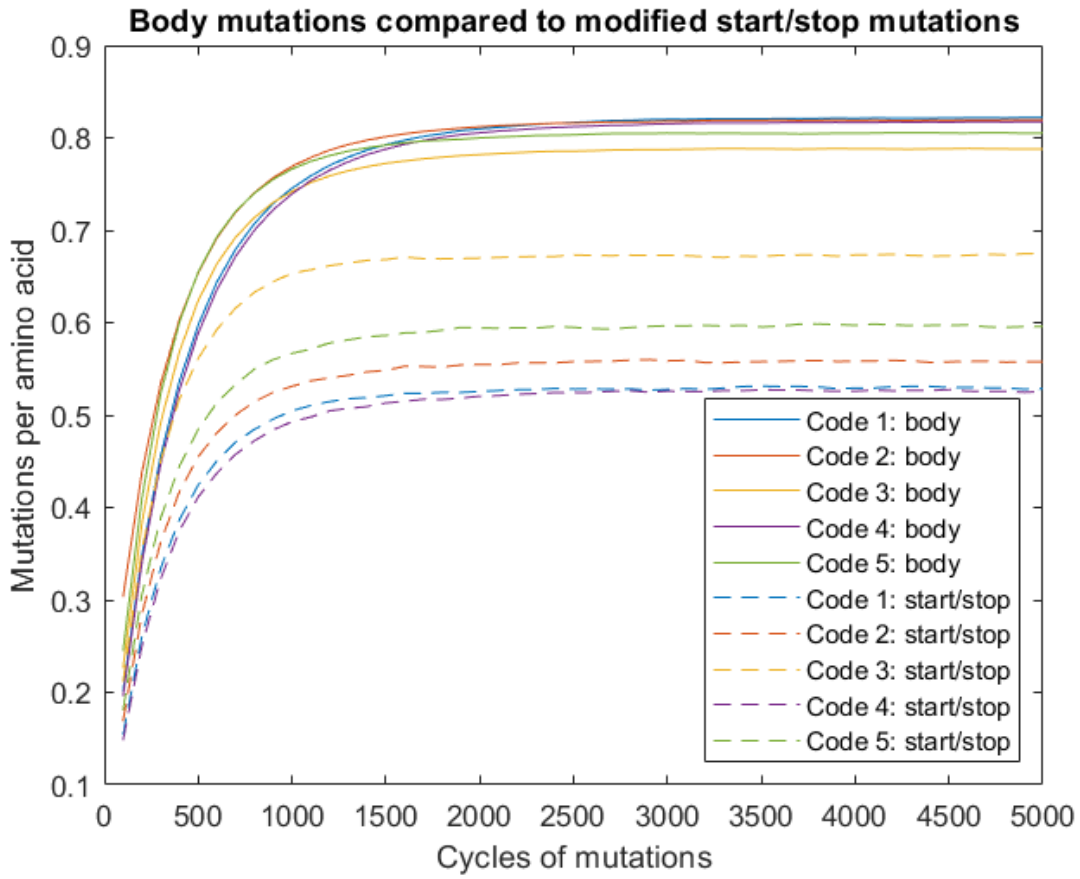

Figure 22: body mutations compared to start/stop mutation multiplied by coefficient $K_{sb} = 4.384$.

Figure 22 plots the averaged amount of body mutations for every code compared to the averaged start/stop mutations amount multiplied by a specific coefficient derived from probabilistic consideration on possible mutations of amino acids. Both measures are considered specific for the single amino acid, meaning they were divided by the length of the genome. Expectations were to find a lower number of start/stop mutations since they are hypothesised to be more damaging for the organism, and so happened. As expected, code 3 has the worst ratio between body mutations and start/stop mutations, which is consistent with the previous considerations.

However, it is not safe to make further assumptions. The main issue with this idea is that code 5, which being random should not show this property, is actually keeping a significantly lower amount of start/stop mutations when compared to the body mutations. It would be necessary to use a function considering redundancy in the correct way and to make probabilistic assumptions with greater solidity, maybe considering all the possible different codons instead of the possible amino acids or considering only codons that are 1 mutation away from each other.

3.2.3 Fitness function in different kingdoms

This section will check if the utilized fitness function shows different behaviours if applied to genomes of different kingdoms translated with different genetic codes, i.e. how fit different kingdoms are with respect to this fitness function and different codes. Legends are to be read like in section 3.1.2.

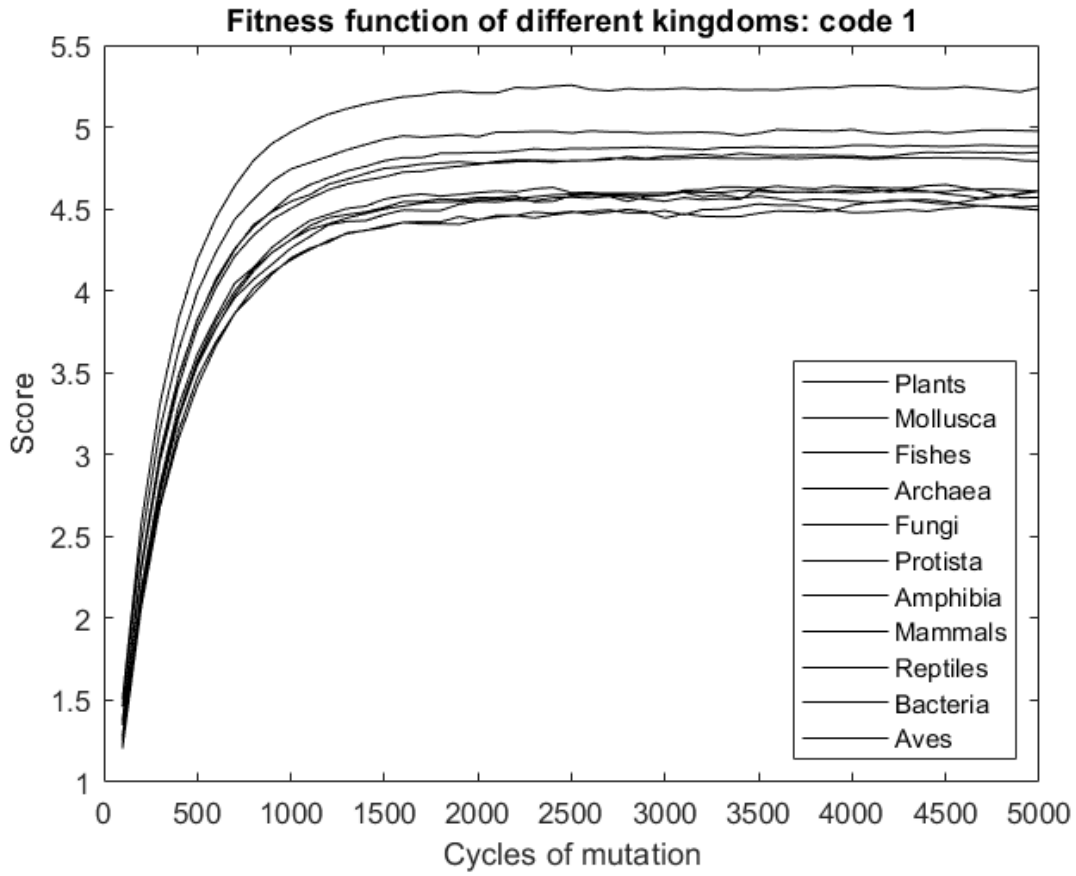

Figure 23: fitness function for different kingdoms: standard genetic code.

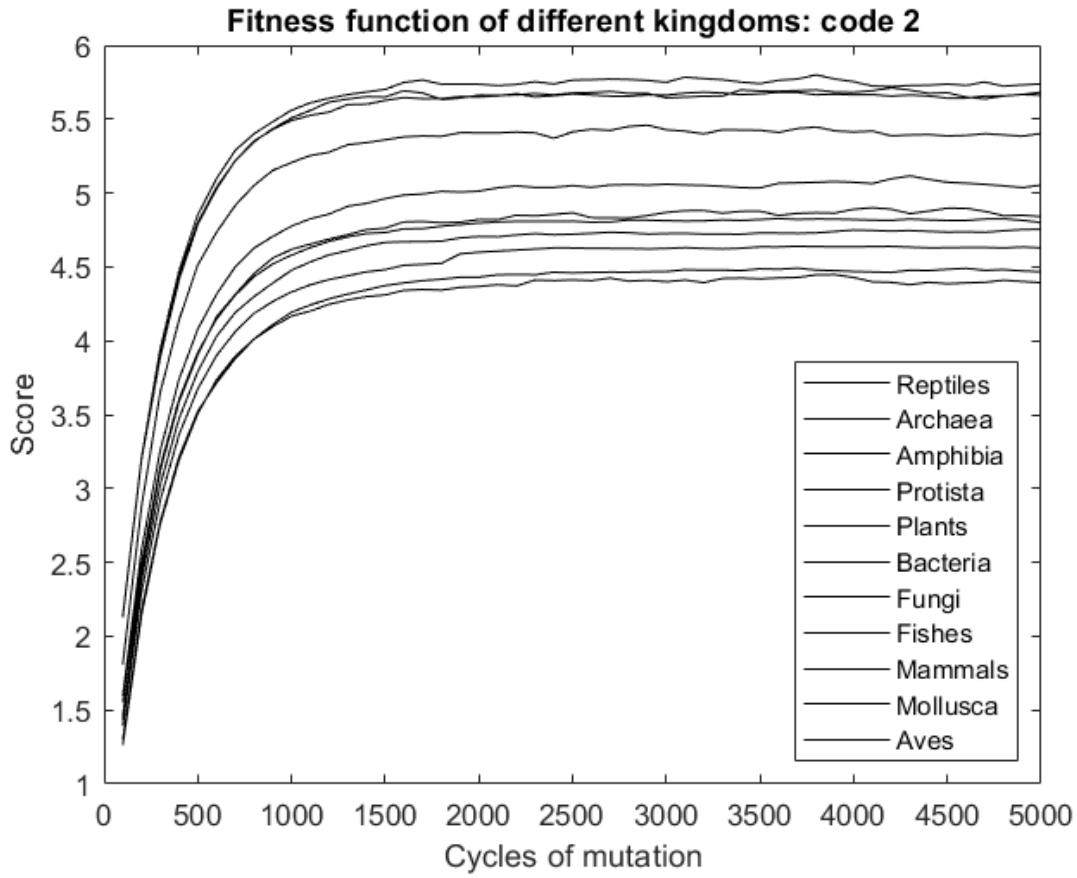

Figure 24: fitness function for different kingdoms: code 2.

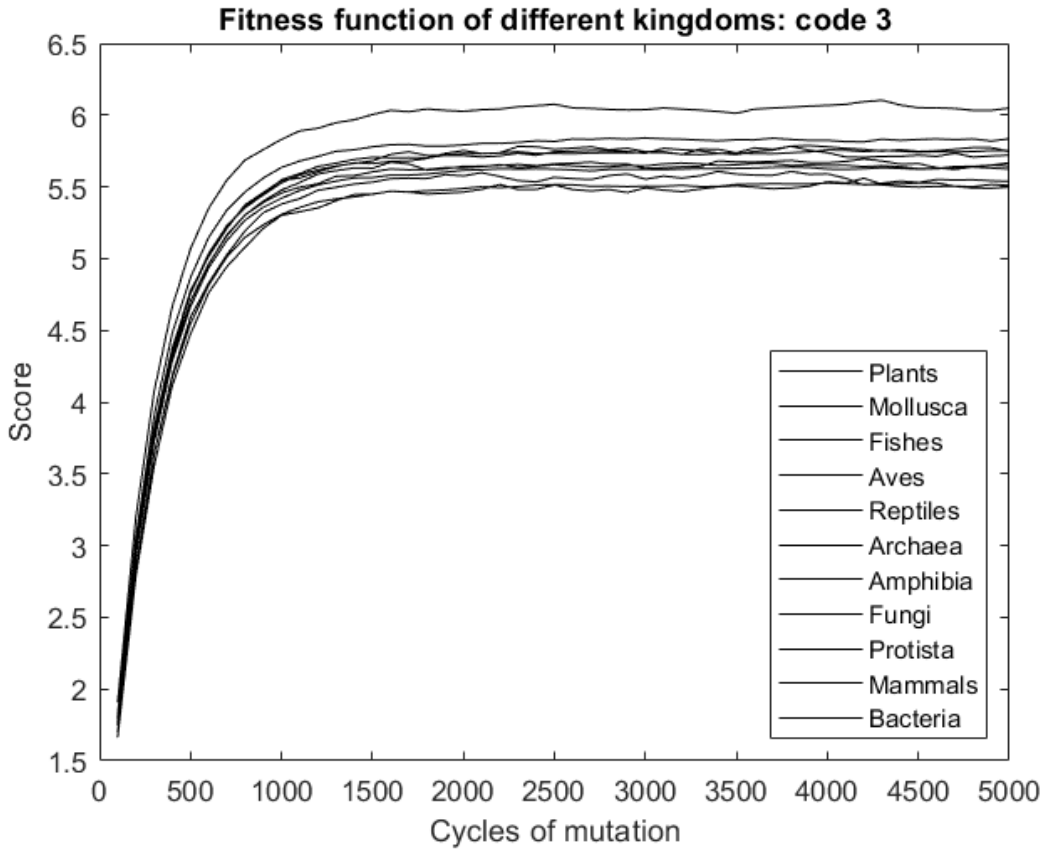

Figure 25: fitness function for different kingdoms: code 3.

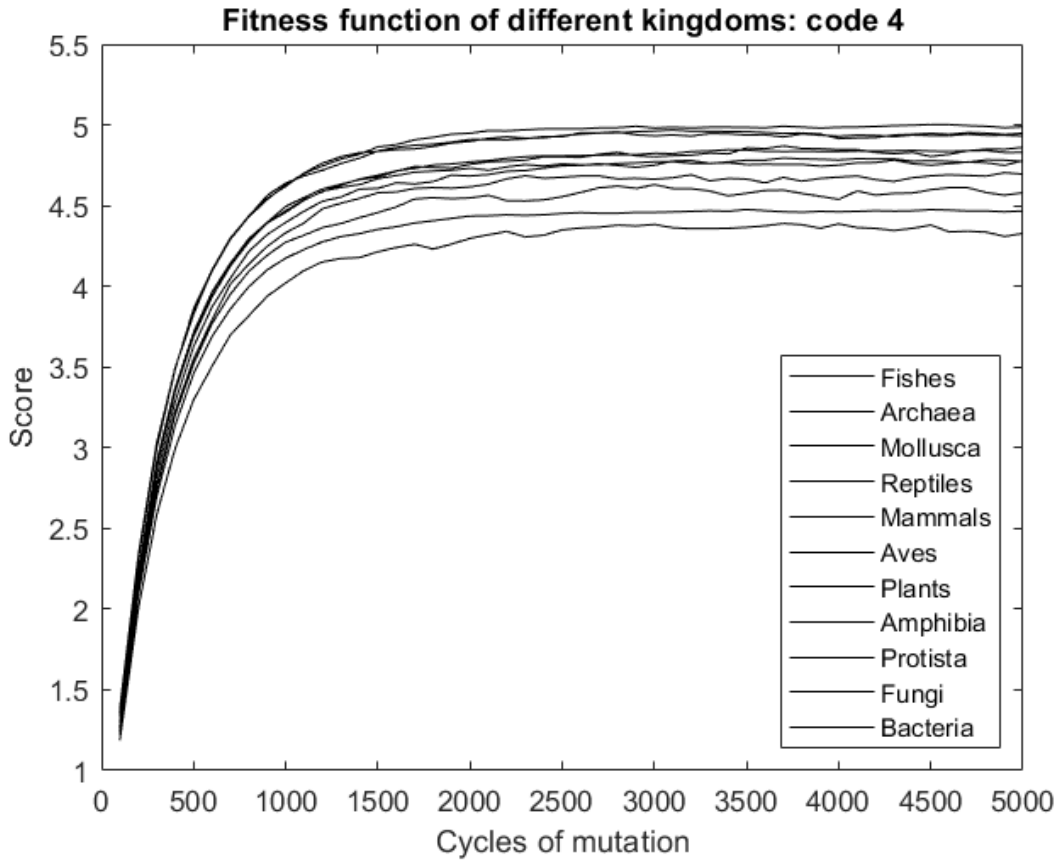

Figure 26: fitness function for different kingdoms: code 4.

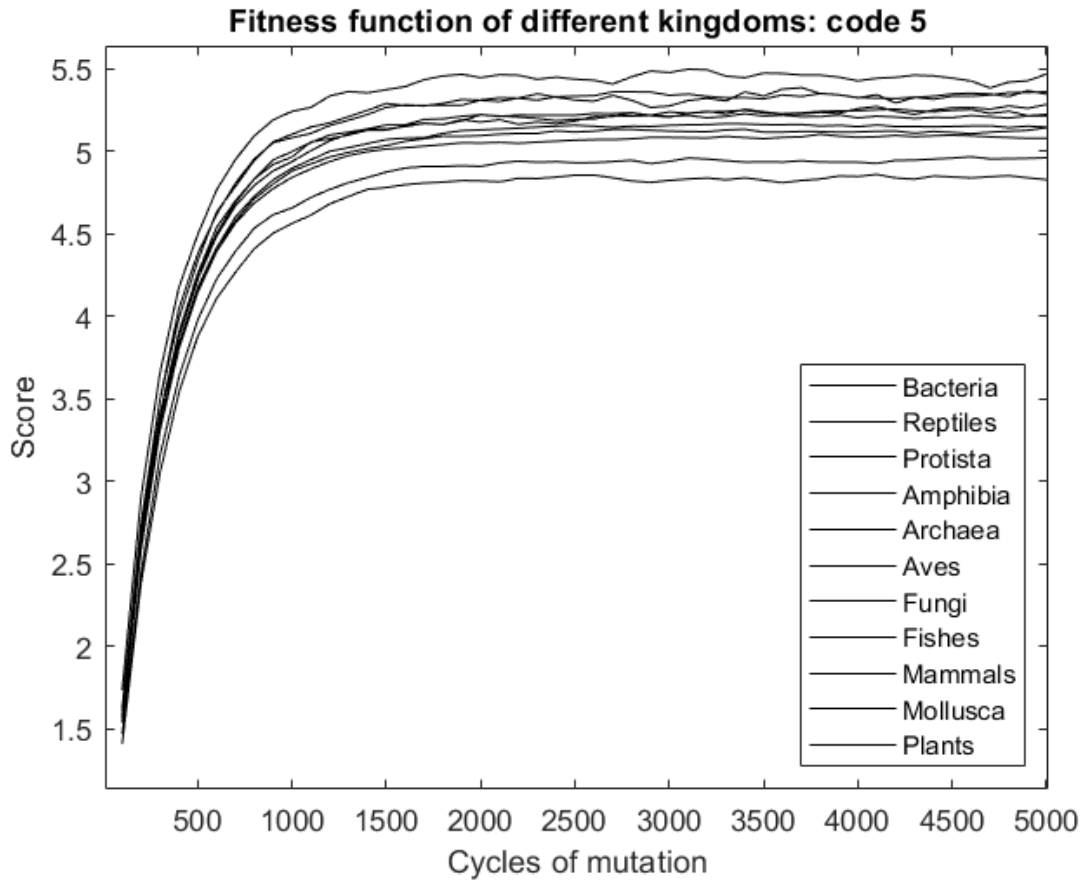

Figure 27: fitness function for different kingdoms: code 5.

It is impossible to extrapolate a trend between different genetic codes from the previous plots, kingdoms behave in a very different manner if different codes are applied. This is, however, a meaningful result: it gives evidence that genetic code may favour or disfavour specific species, hence highlighting which groups of organisms are better suited from a physicochemical point of view.

This being said, it is now necessary to take a closer look to the results obtained by using the standard genetic code to see if it may have favoured some specific species over others: with the sole exception of bacteria, all the kingdoms with low fitness function score (indicating a better fit) are in the left part of the tree of life and are considered to be more recent, like birds, mammals, reptiles and amphibia. It is also worth notice that there is a significant gap between amphibia and fungi, fishes and mollusca. This is relevant because all kingdoms with lower score (again, except for bacteria) differentiated after a mass extinction that happened 444 million of years ago. It is hence possible to hypothesize that conditions on earth changed and organisms became more fit against mutations, also because of the increasing complexity of living life.

It would be very interesting to do the same confrontation but between organisms of which at least an approximated date of origin is known to see if this trend continues. In this case, it would mean that organisms were selected during evolution to have better and better

resilience against mutations. Unfortunately, it was not possible to perform such reasonings in this work because of the wide uncertainty of the age of most of the selected organisms.

4 CONCLUSIONS AND FURTHER RESEARCH

4.1 Entropy

Section 3.1 highlights how the overall structure of the genetic code influences entropy, with the first 3 genetic codes behaving very similarly while codes 4 and 5 are characterized by different plots.

Also, initial entropy values for more than half of the species are higher than the flat codon distribution expectations, indicating a biased codon distribution. Another relevant aspect is that all entropy plots reach a stable value around 2000 cycles of mutations. Interestingly, all plateaus have the same value except for the random genetic code, suggesting that if genome keeps mutating it will reach a specific amount of entropy. Most initial entropy values, however, are different from the plateau: this suggests that organisms have some way to keep a certain distribution of amino acids, because f that was not the case then every species with more than 100.00 years would have the same entropy in the genome.

More meaningful results are found when comparing entropy of different kingdoms: by the use of standard genetic code, initial entropy is lower for more ancient organisms and higher for “newer” ones. This suggests that entropy is a relevant quantity from an evolutionary point of view and may be used to find hints about how ancient a specie is. This trend is kept as long as the overall structure of the code is maintained, but it disappears while using codes 4 or 5. Also, differences between entropy of organisms are denied if these two genetic codes are used: possible future research may try to understand why this is the case.

4.2 Fitness function

Fitness function was developed in order to explore the physicochemical hypothesis, which states that genetic code evolved in order to minimize meaningfulness of translational errors. As expected, standard genetic code has lower value, i.e. it is more fit to resist mutations. However, also genetic code 4 had some excellent results, being absolutely comparable to the standard code. This result suggests that fitness function scores are only dependant on the organization of third letters in the code, while relative positions of groups of codons formed by the same first and second letter does not affect resilience against mutations.

Another aspect is that, independently on which genetic code is being used, start/stop mutations are disfavoured with respect to body mutations. This is particularly true for codes 1 and 4, again highlighting the low relevance of the first 2 letters of the codons in preventing dangerous mutations. However, further research will be necessary to determine if these were just coincidences or if these statements are valid for every random genetic code.

Fitness function showed another interesting behaviour: if standard genetic code is used, there is a trend where more recent groups of organisms have a lower fitness function score, suggesting that organisms may have evolved in order to have better physicochemical

behaviour. The trend is, interestingly, not maintained if any other of the presented genetic codes is used: this result gives good support to the physicochemical theory.

Lastly, a method to quantify relative distances between redundant amino acids (in terms of mutations needed to go from one amino acid to another) is proposed: all fitness functions, in fact, reach a plateau after many mutations. This is probably due to the high number of mutations, which stabilize differences in the fitness function score. It is reasonable to think that the plateau value is dependent on the average distance between amino acids: the process can be used inversely to determine this mean distance in different statistically by the usage of a chosen fitness function. Further trials will be necessary to determine how accurate this process can be by confrontation with theorical, algebraic methods.

BIBLIOGRAPHY

Crick, F. H. C. “An error in model building.” *Nature*, 1967.

Johnson, DBF. “Imprints of the genetic code in the ribosome.” *PNAS*, 2010.

M Yarus, EL Christian. “Genetic code origins.” *Nature*, 1989.

Stanfel, Larry E. “A new approach to clustering the amino acids.” *Journal of theoretical biology*, 1996.

Weiss, Madeline. “The physiology and habitat of the last universal common ancestor.” *Nature*, 2016.

Wong, J. Tze-Fei. “A co-evolution theory of the genetic code.” *PNAS*, 1975.

APPENDICES

A1: MATLAB CODE

The Matlab code was autonomously developed to satisfy the requirements of this work and is divided in 2 parts: The first part concerns extracting data from the .fna files, while the second part performs all the needed calculations.

A1.1: data extraction

```
%% obtaining charData, the sequence of ATCG for a specified genome
clc
close all
clear all
% https://it.mathworks.com/help/bioinfo/ug/working-with-whole-genome-data.html
% page for the step-by-step tutorial


foldercontents = dir;

for i = 1:length(foldercontents)
    file = foldercontents(i).name;
    fidIn = fopen(file,'r');
    header = fgetl(fidIn);
    mmfile = [file '.mm'];
    fidOut = fopen(mmfile,'w');
    currentFilename = foldercontents(i).name;
    currentFilename = string(currentFilename);
    %reads the file in 1 MB large pieces and writes
    newLine = newline;
    blockSize = 2^20;
    while ~feof(fidIn)

        % Read in the data
        charData = fread(fidIn,blockSize,'*char')';

        % Remove new lines
        charData = strrep(charData,newline,'');

        % Remove characters that make nt2int fail

        numIdx=find(~isletter(charData));
        charData(numIdx)='';

        knowncharData = erase(charData,'N');
        charData = charData;
        charData = upper(charData);



        % Convert to integers (this is where the code fails because of arrays
```

```
        % indexing). intData codes with '1 2 3 4', while charData contains the same
        % code with 'A T C G', but things can be worked out with letters as well
        %intData = nt2int(charData);

        % Write to the new file
        fwrite(fidOut,charData,'uint8');
    end

    vecname = sprintf(file);
    vecname2 = ['00', vecname, '.mat'];
    save(vecname2,'charData');

    %close the files
    fclose(fidIn);
    fclose(fidOut);
    delete(mmfile);
end
```

A1.2: calculations

```
clc
close all
clear all


%% define the similarity table of amino acid
% Amino acid; Symbol; Volume; Hydrophilicity; Area; Polarity; Charge; Shape.

AMINOTABLE = {'AA', 'ALA', 'Symbol', 'A','Volume',  90, 'Hydrophilicity', 0.45,
'Area', 115, 'Polarity', 1.6, 'Charge', 0, 'Shape', 1.1;
'AA', 'CYS', 'Symbol', 'C', 'Volume', 113, 'Hydrophilicity', 3.63, 'Area', 135,
'Polarity', 2.0, 'Charge', 0, 'Shape', 3.0;
'AA', 'ASP', 'Symbol', 'D', 'Volume', 118, 'Hydrophilicity', 13.34, 'Area', 150,
'Polarity', -9.2, 'Charge', -1, 'Shape', 5.0;
'AA', 'GLU', 'Symbol', 'E', 'Volume', 142, 'Hydrophilicity', 12.59, 'Area', 190,
'Polarity', -8.2, 'Charge', -1, 'Shape', 5.2;
'AA', 'PHE', 'Symbol', 'F', 'Volume', 193, 'Hydrophilicity', 3.15, 'Area', 210,
'Polarity', 3.7, 'Charge', 0, 'Shape', 12.0;
'AA', 'GLY', 'Symbol', 'G', 'Volume', 64, 'Hydrophilicity', 0, 'Area', 75,
'Polarity', 1.0, 'Charge', 0, 'Shape', 1.0;
'AA', 'HIS', 'Symbol', 'H', 'Volume',  159, 'Hydrophilicity', 12.66, 'Area',
195,'Polarity', -3.0, 'Charge', 1, 'Shape', 7.0;
'AA', 'ILE', 'Symbol', 'I', 'Volume', 164, 'Hydrophilicity', 0.24, 'Area', 175,
'Polarity', 3.1, 'Charge', 0, 'Shape', 1.45;
'AA', 'LYS', 'Symbol', 'K', 'Volume', 170, 'Hydrophilicity', 11.91, 'Area', 200,
'Polarity', -8.8, 'Charge', 1, 'Shape', 8.5;
'AA', 'LEU', 'Symbol', 'L', 'Volume', 164, 'Hydrophilicity', 0.11, 'Area', 170,
'Polarity', 2.8, 'Charge', 0, 'Shape', 1.4;
'AA', 'MET', 'Symbol', 'M', 'Volume', 167, 'Hydrophilicity', 3.87, 'Area', 185,
'Polarity', 3.4, 'Charge', 0, 'Shape', 3.3;
'AA', 'ASN', 'Symbol', 'N', 'Volume', 126, 'Hydrophilicity', 12.08, 'Area', 160,
'Polarity', -4.8, 'Charge', 0, 'Shape', 5.1;
'AA', 'PRO', 'Symbol', 'P', 'Volume', 124, 'Hydrophilicity', 11.15, 'Area', 145,
'Polarity', -0.2, 'Charge', 0, 'Shape', 1.25;
```

```
'AA', 'GLN', 'Symbol', 'Q', 'Volume', 142, 'Hydrophilicity', 12.08, 'Area', 180,
'Polarity', -4.1, 'Charge', 0, 'Shape', 5.3;
'AA', 'ARG', 'Symbol', 'R', 'Volume', 195, 'Hydrophilicity', 22.31, 'Area', 225,
'Polarity', -12.3, 'Charge', 1, 'Shape', 8.6;
'AA', 'SER', 'Symbol', 'S', 'Volume', 95, 'Hydrophilicity', 7.45, 'Area', 115,
'Polarity', 0.6, 'Charge', 0, 'Shape', 2.0;
'AA', 'THR', 'Symbol', 'T', 'Volume', 121, 'Hydrophilicity', 7.27, 'Area', 140,
'Polarity', 1.2, 'Charge', 0, 'Shape', 2.1;
'AA', 'VAL', 'Symbol', 'V', 'Volume', 139, 'Hydrophilicity', 0.40, 'Area', 155,
'Polarity', 2.6, 'Charge', 0, 'Shape', 1.3;
'AA', 'TRP', 'Symbol', 'W', 'Volume', 231, 'Hydrophilicity', 8.27, 'Area', 255,
'Polarity', 1.9, 'Charge', 0, 'Shape', 12.15;
'AA', 'TYR', 'Symbol', 'Y', 'Volume', 197, 'Hydrophilicity', 8.50, 'Area', 230,
'Polarity', -0.7, 'Charge', 0, 'Shape', 12.05;
'AA', 'STOP', 'Symbol', '*','Volume', 197, 'Hydrophilicity', 8.50, 'Area', 230,
'Polarity', -0.7, 'Charge', 0, 'Shape', 12.05;};

q = 1;
for i = 6:2:16
    x = AMINOTABLE(:,i);
    x = cell2mat(x);
    average(q) = mean(pdist(x));
    weight(q) = sqrt(1/average(q));
    q = q+1;
    maxx = max(x);
    minn = min(x);

    for j = 1:length(x)
        xx(j) = (x(j)-minn)/(maxx-minn);
        AMINOTABLE(j,i) = num2cell(xx(j));
    end

end

%
 lista = dir('00*');


%% GENERATING MUTATED SEQUENCES OF INTEGERS
 for l = 1:36
    name = lista1(l).name;
    load(sprintf(name));



    intData = strrep(charData,'A','1');
    intData = strrep(intData,'T','2');
    intData = strrep(intData,'C','3');
    intData = strrep(intData,'G','0');



    % forcing mutations
    q = 0;
    w = 0;
    y = 0;
    repe = 0;
    mutations = zeros(50,length(charData));

    for g = 1:5000
```

```
        for i = 1:length(intData)
            pm = randi(100000,1);

            if pm < 120
              cont = fix(pm/40)+1;
              data = mod((intData(i)+cont),4);
              intData(i) = num2str(data);
            end

        end

        if mod(g,100) == 0
            repe = repe+1;
            mutations(repe,:) = intData;
        end

    end

    save(name);

end


%% fitness function
% mynt2aa is a function obtained by modification of a pre-existent
% Matlab function and was used to change the genetic code.
orig_genome = mynt2aa(charData);
orig_length = length(orig_genome);
list = AMINOTABLE(:,4)';
list = cell2mat(list);

for k = 1:50
    mutt = 0;
    stt = 0;

    mutgen = mutations(k,:);

    mutgen = strrep(mutgen,'1','A');
    mutgen = strrep(mutgen,'2','T');
    mutgen = strrep(mutgen,'3','C');
    mutgen = strrep(mutgen,'0','G');
    mutated_genome = mynt2aa(mutgen);

    mutated_length = length(mutated_genome);
    t = min(orig_length,mutated_length);
    totalff = 0;
    ff = 0;
    for i = 1:t
        if orig_genome(i) == 'M' && mutated_genome(i) ~= 'M'
            totalff = totalff+30;
            stt = stt+1;

        elseif orig_genome(i) == '*' && mutated_genome(i) ~= '*'
            totalff = totalff+30;
            stt = stt+1;
        elseif orig_genome(i) ~= 'M' && mutated_genome(i) == 'M'
            totalff = totalff+30;
            stt = stt+1;
```

```
        elseif orig_genome(i) ~= '*' && mutated_genome(i) == '*'
            totalff = totalff+30;
            stt = stt+1;

        else
            ind1 = find(orig_genome(i)==list);
            ind2 = find(mutated_genome(i)==list);
            mutt = mutt+1;
            for j = 6:2:16
                q = (j-4)/2;
                tempff              =              ((cell2mat(AMINOTABLE(ind1,j))-
cell2mat(AMINOTABLE(ind2,j)))^2*weight(q));
                ff = ff+tempff;
            end
            ff = sqrt(ff);
            totalff = totalff+ff;
        end


    end
    totalf(k) = totalff/t;
    st(k) = stt/t;
    mut(k) = mutt/t;


    %% entropy calculations

    origData = mutations(k,:);
    origData = num2str(origData);
    origData = strrep(origData,'49','1');
    origData = strrep(origData,'50','2');
    origData = strrep(origData,'51','3');
    origData = strrep(origData,'48','0');

    origDatalett = strrep(origData,'1','A');
    origDatalett = strrep(origDatalett,'2','T');
    origDatalett = strrep(origDatalett,'3','C');
    origDatalett = strrep(origDatalett,'0','G');
    origDatalett = strrep(origDatalett,' ','');
    orig_gen = mynt2aa(origDatalett);

    % A = 1, T = 2, C = 3, G = 0
    % Counting bases and their probability
    basecount_A = count(origData,'1');
    basecount_T = count(origData,'2');
    basecount_C = count(origData,'3');
    basecount_G = count(origData,'0');

    pA = basecount_A/length(origData);
    pC = basecount_C/length(origData);
    pG = basecount_G/length(origData);
    pT = basecount_T/length(origData);

    % Calculating entropy and specific entropy for the sequence of bases
    total_entropy_bases(k) = -(pA*log(pA)+pC*log(pC)+pG*log(pG)+pT+log(pT));
    total_entropy_bases_spec(k) = total_entropy_bases(k)/length(origData);

    % Defining the amino acids to calculate their entropy
```

```
    AAName                                                                  =
['A','R','N','D','C','Q','E','G','H','I','L','K','M','F','P','S','T','W','Y','V','
*'];
    total_entropy_AA(k) = 0;









    for i = 1:length(AAName)
        %occurence of each aminoacid
        AAOccurency(i) = count(orig_gen,AAName(i));
        %probability of each aminoacid
        AAprob(i) = AAOccurency(i)/length(orig_gen);
        %entropy calculated for every aminoacid
        ent_singleAA(i) = -AAprob(i)*log(AAprob(i));
        %sum up all contributions to find total entropy
        total_entropy_AA(k) = total_entropy_AA(k)+ent_singleAA(i);
    end

    total_entropy_AA_spec(k) = total_entropy_AA(k)/length(orig_gen);

end


%%  saving

fas = sprintf(name);
fas(1:2) = '';
%'G1' is modified into G2, G3, G4, and G5 for different genetic codes
fas = ['G1',fas]
save(fas);
```