

Article

Computational Analysis of Genetic Code Variations Optimized for the Robustness against Point Mutations with Wobble-like Effects

Elena Fimmel [†], Markus Gumbel [†], Martin Starman ^{*,†} and Lutz Strüingmann [†]

Center for Algorithmic and Mathematical Methods in Medicine, Biology, and Biotechnology, Mannheim University of Applied Sciences, 68163 Mannheim, Germany; e.fimmel@hs-mannheim.de (E.F.); m.gumbel@hs-mannheim.de (M.G.); l.struengmann@hs-mannheim.de (L.S.)

* Correspondence: m.starman@hs-mannheim.de

† These authors contributed equally to this work.

Abstract: It is believed that the codon–amino acid assignments of the standard genetic code (SGC) help to minimize the negative effects caused by point mutations. All possible point mutations of the genetic code can be represented as a weighted graph with weights that correspond to the probabilities of these mutations. The robustness of a code against point mutations can be described then by means of the so-called conductance measure. This paper quantifies the wobble effect, which was investigated previously by applying the weighted graph approach, and seeks optimal weights using an evolutionary optimization algorithm to maximize the code’s robustness. One result of our study is that the robustness of the genetic code is least influenced by mutations in the third position—like with the wobble effect. Moreover, the results clearly demonstrate that point mutations in the first, and even more importantly, in the second base of a codon have a very large influence on the robustness of the genetic code. These results were compared to single nucleotide variants (SNV) in coding sequences which support our findings. Additionally, it was analyzed which structure of a genetic code evolves from random code tables when the robustness is maximized. Our calculations show that the resulting code tables are very close to the standard genetic code. In conclusion, the results illustrate that the robustness against point mutations seems to be an important factor in the evolution of the standard genetic code.

Keywords: genetic code; point mutations; wobble effect; evolutionary algorithm



Citation: Fimmel, E.; Gumbel, M.; Starman, M.; Strüingmann, L. Computational Analysis of Genetic Code Variations Optimized for the Robustness against Point Mutations with Wobble-like Effects. *Life* **2021**, *11*, 1338. <https://doi.org/10.3390/life11121338>

Academic Editor: Pedro Martinez

Received: 3 November 2021

Accepted: 27 November 2021

Published: 3 December 2021

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

1.1. Motivation

The origin and the structure of the standard genetic code (SGC), i.e., the codon–amino acid assignment is still a subject under research [1]. There are at least three major theories (overview in [2]): (1) The stereochemical theory states that codon assignments for specific amino acids are determined by physicochemical affinities between amino acids and cognate codons or anticodons [3]. This hypothesis is not very well supported by experimental data, though. (2) The coevolution theory explains the structure of the standard code by pathways of amino acid biosynthesis which were added step by step [4,5]. (3) A third theory (adaption theory) claims that the SGC has evolved under selective pressure to minimize translation errors [6]. In particular, the code is robust against point mutations under this assumption.

All three theories agree to some extent that evolution has included more amino acids to be encoded [7,8]. Understanding the principles of the genetic code enables the development of a modified SGC with non-canonical amino acids [9,10]. This is of importance in biotechnology as, for instance, it facilitates the development of novel drugs. Recently, Błażej and colleagues analyzed the extension of the SGC that was inspired by the robustness of the code against point mutations [11].

A graph-based approach was presented in our previous work [12] which quantifies for a (generalized) genetic code its robustness against point mutations. A weighted graph represents possible point mutations for each codon (or tuple in general). We demonstrated that such a genetic code shows better robustness and the weights are related to the wobble effect. This paper continues our work and searches for weighted graphs that maximize the robustness of the standard genetic code.

1.2. Conductance of the Genetic Code

This section briefly recalls the definitions from [12] which are based on [13]. Let Σ be a finite alphabet of even cardinality $|\Sigma| = 2n$ for some $n \in \mathbb{N}$. As a special alphabet $\mathcal{B} = \{A, C, G, T(U)\}$ is used for the standard genetic alphabet.

Definition 1. Let $\ell \in \mathbb{N}$ and $P = \{p_i^{\{N,N'\}} \mid i = 1, \dots, \ell, N \neq N' \in \Sigma\}$ where $p_i^{\{N,N'\}}$ are non-negative weights. We define a weighted graph $G(V, E) = G_i^P(V, E, w)$ as follows:

- (1) $V = \Sigma^\ell$ is the set of vertices (nodes) representing all possible ℓ -letter words over Σ ;
- (2) E is the set of edges where $(c, c') \in E$ if and only if $c, c' \in V$ and c differs from c' in exactly one position;
- (3) The function $w : E \rightarrow P$ assigns to every edge $(c, c') \in E$ a weight $p_i^{\{N,N'\}}$ by $w((c, c')) = p_i^{\{N,N'\}}$ if and only if c differs from c' in position $i \in \{1, \dots, \ell\}$ and $c_i = N, c'_i = N'$.

If for all $i \in \{1, \dots, \ell\}$ the weights $p_i^{\{N,N'\}}$ are independent of the choice of N, N' we will simply denote the weights $p_i^{\{N,N'\}}$ by p_i .

According to Definition 1 the graph G is weighted, undirected, and regular. Note that the weights are symmetric, i.e., $p_i^{\{N,N'\}} = p_i^{\{N',N\}}$. Given $\Sigma = \mathcal{B} = \{A, C, G, T(U)\}$ the graph G has a biological interpretation: The set of edges E represents all possible single point mutations, which can occur between codons in protein-coding sequences. Such point mutations appear quite often and might lead to fatal changes in translated proteins (see [14]). The weights p_i can be considered, if standardized, as the probabilities with which a point mutation occurs at position i . This definition also takes into account that these mutation probabilities may depend not only on the position in the codon, but also on the base pairs. For example, it is likely that the mutation $U \rightarrow G$ in the third position of a codon occurs more frequently than the mutation $U \rightarrow A$ [15]. Figure 1 depicts an example of a graph that satisfies Definition 1.

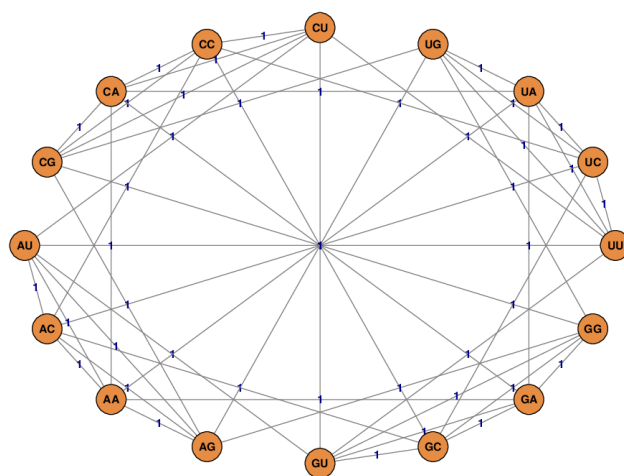


Figure 1. The graph G for $\Sigma = \mathcal{B} = \{A, C, G, U\}$, $\ell = 2$ and all weights $p_i^{\{N,N'\}} = 1$. For simplicity, tuples of length 2 are used instead of codons ($\ell = 3$)—which leads to only 16 instead of 64 nodes.

Given this definition, we might ask how the set of all available codons (ℓ -tuples in general) could be partitioned into disjoint subsets where each subset represents an amino acid to be encoded and where the influence of single point mutations is minimal. This can be achieved by solving a graph clustering problem.

Let \mathcal{C}_k be a partition of nodes of the graph G into a fixed number $1 < k \leq (2n)^\ell$ of disjoint non-empty subsets \mathcal{C}_k :

$$\mathcal{C}_k = \{S_1, S_2, \dots, S_k : S_i \cap S_j = \emptyset, S_1 \cup S_2 \cup \dots \cup S_k = V\}.$$

Each subset S_i in this partition is supposed to contain codons that encode the same amino acid. According to [13] the “quality” of such a partition can be measured by means of a so-called conductance. We first define the conductance for a single subset S of V [16] and adapt then the definition from [13] to weighted graphs:

Definition 2. For a given weighted graph $G = G(V, E, w)$ let S be a subset of $V = \Sigma^\ell$. We define the set-conductance of S as:

$$\phi(S) = \frac{w(E(S, \bar{S}))}{\sum_{c \in S, (c, c') \in E} w((c, c'))}$$

where $w(E(S, \bar{S}))$ is the sum of the weights of edges of G crossing from S to its complement \bar{S} :

$$E(S, \bar{S}) := \{(c, c') \in E : |\{c, c'\} \cap S| = 1\}.$$

We set $w(E(S, \bar{S})) := 0$ if $E(S, \bar{S}) = \emptyset$. As an extension, the set-robustness $\rho(\cdot)$ is additionally defined as

$$\rho(S) := 1 - \phi(S).$$

Figure 2 shows an example for Definition 2. In a biological context, $\phi(S)$ has a useful meaning. If all codons of S encode the same amino acid or the stop signal, then $\phi(S)$ is the ratio of the number of non-synonymous single nucleotide substitutions to all possible nucleotide substitutions. Analogously, $\rho(S) = 1 - \phi(S)$ represents synonymous mutations and thus the robustness of a cluster of codons against point mutations. An ideal code, in this context, would have a conductance $\phi(S) = 0$ or, equivalently, a robustness $\rho(S) = 1$. Note that some greater weight $p_i^{\{N, N'\}}$ of one edge in $E(S, \bar{S})$ (see Definition 1) means more robustness against point mutations in the sense that $p_i^{\{N, N'\}}$ increases $w(E(S, \bar{S}))$ in the numerator of $\phi(S)$. Hence $\phi(S)$ will increase as the sum off all weights in the denominator contains $w(E(S, \bar{S})) + w(E(S, S))$ whereas the weights in $w(E(S, S))$ remain the same.

The conductance for a partition was proposed in [17], adopted in [13,18] and is defined as the conductance of the “weakest link in the chain”:

Definition 3. For a given weighted graph $G = G(V, E, w)$ the conductance of a partition \mathcal{C}_k of $V = \Sigma^\ell$ is defined as

$$\Phi(\mathcal{C}_k) = \max_{S \in \mathcal{C}_k} \phi(S).$$

Φ gives a characterization of the quality of a partition \mathcal{C}_k as the set conductance of the worst l -letter group in this partition. However, there might be a problem: If an amino acid-like Methionine is encoded by exactly one codon (like in the standard genetic code) we always have $\Phi(\mathcal{C}_k) = 1$ no matter how good the other set-conductances are. This can be mitigated by the definition of average conductance.

Definition 4. The average conductance of a partition C_k is defined as

$$\bar{\Phi}(C_k) = \frac{1}{k} \sum_{S \in C_k} \phi(S).$$

In addition to the average conductance, there is also the average robustness.

Definition 5. The average robustness of a partition C_k is defined as

$$\bar{P}(C_k) = 1 - \bar{\Phi}(C_k)$$

or equally

$$\bar{P}(C_k) = \frac{1}{k} \sum_{S \in C_k} \rho(S).$$

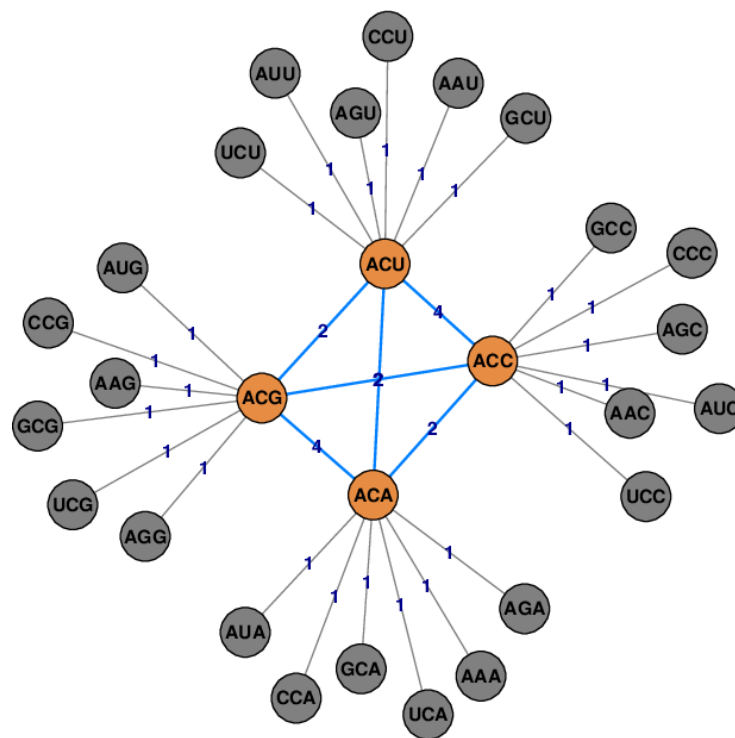


Figure 2. The graph of $S = \{ACU, ACC, ACA, ACG\}$ inside the graph G for $\Sigma = B$. The set-conductance is $\phi(S) = 24/56 \simeq 0.43$.

The average conductance for the SGC with all weights set to 1 is $\bar{\Phi}(C_{SGC}) \simeq 0.81$ [19]. It can be decreased to $\bar{\Phi}(C_{SGC}; P_M) \simeq 0.54$ when the weights are set according to Table 1 which was proven in [12] (Table 1). However, it is an open question whether these weights are already optimal.

In this paper, we search for optimal weights that minimize the average conductance $\bar{\Phi}$. We also look for a genetic code table that minimizes the average conductance while the weights in Table 1 are fixed.

Table 1. Weight distribution P_M as introduced in [12]. Each table shows the weights according to base position 1 to 3 within a codon (see number in upper left corner). In accordance with Definition 1 the values are $p_1^{\{N,N'\}} = p_2^{\{N,N'\}} = 1$ for all $N, N' \in \mathcal{B}$ with $N \neq N'$, $p_3^{\{U,G\}} = p_3^{\{A,C\}} = 2$ and $p_3^{\{U,C\}} = p_3^{\{A,G\}} = 4$.

1	U	C	A	G
U	0	1	1	1
C	1	0	1	1
A	1	1	0	1
G	1	1	1	0
2	U	C	A	G
U	0	1	1	1
C	1	0	1	1
A	1	1	0	1
G	1	1	1	0
3	U	C	A	G
U	0	4	2	2
C	4	0	2	2
A	2	2	0	4
G	2	2	4	0

2. Materials and Methods

This section introduces the methods for the optimization of the conductance according to Definition 1. We show three possible ways to approach this topic.

- (1) The first problem is the optimal adjustment of the weights to minimize the conductance of the SGC (this task is abbreviated as and later referred to *EA weights*). In Section 2.1 we describe the optimization algorithm used to achieve these objectives.
- (2) The second attempt is to find a genetic code table with 21 classes such that the average conductance is optimal with respect to the weighting of the graph according to (a) Table 1 and (b) the optimal weights found in (1) (task *EA code table*).
- (3) Finally, we discuss if both the weights and the structure of genetic code, are optimized.

Subsequently, in Sections 2.2 and 2.3 the adjustments of the EA for the three methods are outlined. Concluding, Sections 3.1–3.3 and 4 discuss the results obtained.

2.1. Evolutionary Algorithm

We choose an *evolutionary algorithm* (EA; also *genetic algorithm*) to solve the optimization problem [20,21]. EAs are algorithms suitable for discrete optimization problems inspired by Darwin’s theory of evolution. In the simplified model of Darwin’s theory used in EAs, a population evolves through mating and mutation to adapt to the environment. This process consists of crossing over the individuals in the population followed by a certain number of mutations within the population and a final selection of the fittest member (see Figure 3).

The EA only provides a framework for optimization. The number of individuals that are mated or mutated is parameterized and can be set according to the application. Additionally, a factor can be set to control the survival rate after each iteration. The algorithm also demands the definitions of the crossover and mutation functions and a function to obtain a numerical fitness value. The Δ parameter controls the termination condition. It is defined in such a way that if the change of the fitness of the best individual in the population in the last optimization step is smaller than Δ then the condition is fulfilled and the EA is terminated. Our implementation is based on the R package GA (short for: *genetic algorithm*) [22].

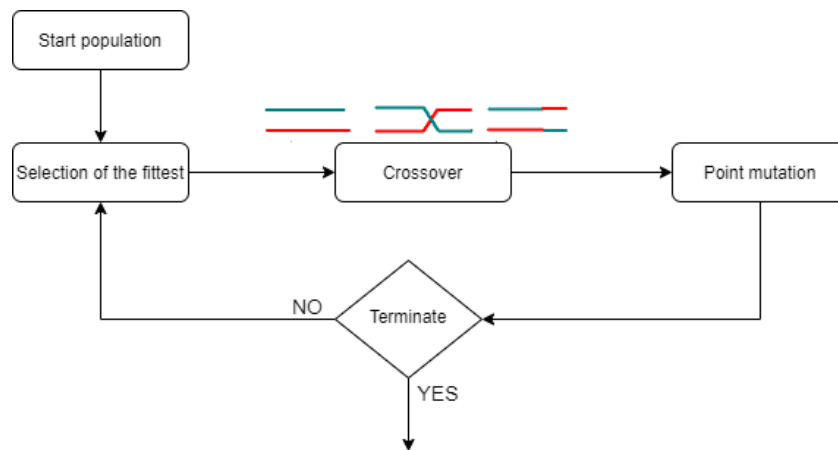


Figure 3. Simplified flow chart of a evolutionary algorithm (EA).

2.2. Optimal Weights

In order to optimize the weights (task *EA weights*), they must first be put into a suitable form. Table 1 illustrates the weights as an example. A weight $p_i^{\{N,N'\}}$ at an edge representing a mutation is specified by the position of the mutation and the type of mutation. The position i of the mutation indicates which matrix must be used to select the weight. This can be identified by the position index in the upper left corner of the matrix. The type of the mutation $\{N, N'\} \in \mathcal{B}$ gives the row and the column and therefore the actual weight. While the regular base N is specified by the row index, the mutated base N' is specified by the column index. From Definition 1 it follows that elements on the diagonal are undefined and we set them to 0. Since the graph is undirected it follows that the values in the matrix are symmetric.

The matrix row-column indexes for every base position according to Table 1 are transformed into a linear list in order to implement the crossover function. The three matrices and the list share the same data and can be used alternately. The used mapping is outlined in Table 2.

Table 2. The index transformation from the row-column system to a linear indexing system. (a) All non-zero columns are mapped to a linear index ranging from 1 to 18. (b) List with values of Table 1.

(a) Indices						
Base position	U ↔ C	U ↔ A	U ↔ G	C ↔ A	C ↔ G	A ↔ G
1	1	2	3	4	5	6
2	7	8	9	10	11	12
3	13	14	15	16	17	18
(b) Values						
Base position	U ↔ C	U ↔ A	U ↔ G	C ↔ A	C ↔ G	A ↔ G
1	1	1	1	1	1	1
2	1	1	1	1	1	1
3	4	2	2	2	2	4

For each pair of parents to be mated, a crossover point between one and 18 is randomly selected. The offspring is generated by exchanging the weights of the parents with each other until the crossover point is reached. This procedure relies on the linear index system from Table 2.

The algorithm mutates a randomly chosen position in one of the three matrices. The mutation is represented by the multiplication of a randomly chosen weight and its symmetric element. The factor is a uniform random number between $\frac{1}{1.2}$ and 1.2 which reflects a mutation rate of $\pm 20\%$.

This fitness for the weight matrices P represents the robustness of the genetic code (see Definition 5) with respect to the weight matrix P :

$$f(P) = \bar{\mathcal{P}}(C_{SGC}; P)$$

The optimization is started with a population of 150 random weight tables. The crossover rate is 60% and the mutation rate 40%. The optimization is terminated if the increase of the fitness is less than a given threshold of $\Delta = 10^{-6}$. According to Definition 1 the weights have to be positive real numbers. It should be noted that some weights could go to infinity and others to zero without further restrictions to the EA, though this did not happen in our optimizations.

Finally, after an optimization process the weights are normalized as the assignment of weights is not unique which is an intrinsic property of the conductance (see Definition 2). The weights can be scaled by any factor. As the 3 matrices contain 36 non-zero values (where actually only 18 can be different) we impose the constraint that the sum of all weights is 36. This is equivalent to a graph where all edges have a weight of 1. We show that this constraint is valid. Let P be the weights and S any partition.

$$s = \sum_{p_i^{\{N, N'\}} \in P} p_i^{\{N, N'\}}$$

$$P' = \{p_i^{\{N, N'\}} \cdot \frac{36}{s} : p_i^{\{N, N'\}} \in P\}$$

Then the conductance $\phi(S)$ is the same with P or P' which is clear if we look at a simplified equation of $\phi(S)$. Let $x, y \in \mathbb{R}$ be the numerator and denominator of $\phi(S) = \frac{x}{y}$ with respect of the weight matrices P . Then one can simplify the equation $\phi(S)$ with respect to the new weight matrices P' as:

$$\phi(S) = \frac{x \cdot \frac{36}{s}}{y \cdot \frac{36}{s}}$$

Yet, now the following holds:

$$\sum_{p_i^{\{N, N'\}} \in P'} p_i^{\{N, N'\}} = 36. \quad (1)$$

2.3. Optimal Genetic Code Table

In a reverse approach, we want to optimize the partitioning using given weight matrices (task *EA code table*). More precisely, this method searches for a code table that maximizes the robustness while the weight matrices related to the SGC are fixed. The only constraint we impose is that we decompose the 64 codons of this table always into 21 classes. Note that the number of classes must be fixed and cannot be left to the algorithm as a further variable to be optimized. This constraint results from the fact that a code with a degeneracy of one would always be the optimum and thus the result would always be the same and without significance.

As introduced above we use an EA for this task. Yet, the crossover and mutation operators must be adapted to the additional condition that a partition requires 21 classes. Hence, we have decided that the crossover function, as well as the mutation function, will ensure that the code tables will consist of 21 classes. To ensure that the crossover operator meets the requirements, only one offspring per mating is produced instead of two. This offspring is the result of a simple random mating of the parents. The randomness is only affected by the assurance that the offspring has 21 classes. The implementation of the crossover operator is introduced in the pseudo-code listed in Algorithm 1.

Algorithm 1 Crossover operator for partitions

```

function CROSSOVERPARTITIONS( $C_1, C_2$ )                                ▷  $C_1$  and  $C_2$  are partitions
   $C_{new} \leftarrow copy(C_1)$                                           ▷ copy first partition
  for  $idx \in 1 \dots 64$  do
    Let  $r$  be a uniform random number between 0 and 1
    if  $r \geq 0.5$  then
       $C_{new}[idx] \leftarrow C_2[idx]$                                   ▷ assign class from second partition
      if  $C_{new}$  has not 21 classes then
         $C_{new}[idx] \leftarrow C_1[idx]$                                 ▷ undo assignment
      end if
    end if
  end for
  return  $C_{new}$ 
end function

```

A mutation replaces for a randomly selected codon its associated class with the class of another random codon. To ensure that the code table still has 21 classes, care is taken when selecting the random codon: its class must not occur only once because then the replacement would remove this class permanently.

The fitness function calculates the robustness of the optimized table C under the influence of the given weight matrices P :

$$f(C) = \bar{P}(C; P)$$

The optimizer is initialized with 100 individuals (random code tables). The cross-over rate is 60% and the mutation rate is 40%. Like before, the process is terminated if the increase of the fitness function value drops below a threshold of $\Delta = 10^{-6}$.

2.4. Single Nucleotide Variants

The weights of the graph indicate how severe a point mutation would be. We would like to compare the weights to the number of single nucleotide variants (SNV) that can be found in biological coding sequences (CDS). Mutations in non-coding DNA regions cannot be considered as the translation of codons into amino acids is optimized in our analysis. Nevertheless, it can be expected that the distribution of point mutations in non-coding sequences will be different and that there are patterns in non-coding DNA [23]. The coding sequences were taken from mouse (*Mus musculus*) chromosome 1 and 2 and downloaded from Ensemble's biomart (<http://www.ensembl.org/biomart/martview>; accessed on 3 November 2021) [24].

The data base *Mouse Short Variants (SNPs and indels excluding flagged variants) (GRCm39)* was chosen. In particular, the biomart attributes

- Variant alleles (e.g., C/U which represents a mutation $C \rightarrow U$)
- Variant start in CDS (bp)
- Variant end in CDS (bp)

enable us to determine the frequency of mutations per base position within a codon. Rows where (1) the variant start or end is *not a number* (nan) or (2) rows with no point mutations were removed.

The base position $i \in \{1, 2, 3\}$ for a codon can be calculated from the attribute *Variant start in CDS (bp)*, denoted as $s \in \mathbb{N}, s \geq 1$:

$$i = ((s - 1) \bmod 3) + 1$$

Given the frequencies of mutations per position one can easily create transition matrices for each position (see Table 4). Typically, the transitions represent probabilities or relative frequencies. Our transition matrices are scaled such that the sum of all values equals 36 according to Equation (1).

3. Results

This section elaborates on the results. We begin with the optimal weights (task *EA weights*, Tables 3 and 4) that were found by the genetic algorithm and then present optimal genetic code tables (*EA code table*).

3.1. Optimal Weights

Table 3a shows the weights of Table 1 as used in [12] where the matrices were normalized to have a sum of 36. This will be done for all matrices in the following sections. The conductance for these weights is about $\bar{\Phi}(\mathcal{C}_{SGC}; P_M) = 0.54$ (rounded to 2 digits) which is better than the conductance of the unweighted graph $\bar{\Phi}(\mathcal{C}_{SGC}) = 0.81$ [19]. The matrices in Table 3b list the optimal weights of the SGC found by the optimization. The average conductance of the SGC with this weight distribution table is $\bar{\Phi}(\mathcal{C}_{SGC}; P_{opt}) = 0.12$. The conductance (or the robustness) could be considerably improved.

Table 3. Optimized weights of the conductance graphs. The values are normalized such that the sum equals 36. (a) Weights from Table 1 in normalized form. These weights were used in [12]. (b) The optimal weights as found by the evolutionary algorithm. Important figures which are discussed in the text are shown in bold.

(a) Normalized weights P_M of Table 1				
1	U	C	A	G
U	0	0.6	0.6	0.6
C	0.6	0	0.6	0.6
A	0.6	0.6	0	0.6
G	0.6	0.6	0.6	0
2	U	C	A	G
U	0	0.6	0.6	0.6
C	0.6	0	0.6	0.6
A	0.6	0.6	0	0.6
G	0.6	0.6	0.6	0
3	U	C	A	G
U	0	2.6	1.3	1.3
C	2.6	0	1.3	1.3
A	1.3	1.3	0	2.6
G	1.3	1.3	2.6	0
(b) Optimal weights P_{opt} found by evolutionary algorithm				
1	U	C	A	G
U	0	0.006	0.002	0.003
C	0.006	0	0.003	0.003
A	0.002	0.003	0	0.005
G	0.003	0.003	0.005	0
2	U	C	A	G
U	0	0.002	0.002	0.003
C	0.002	0	0.003	0.004
A	0.002	0.003	0	0.005
G	0.003	0.004	0.005	0
3	U	C	A	G
U	0	15.925	0.018	0.007
C	15.925	0	0.021	0.012
A	0.018	0.021	0	1.977
G	0.007	0.012	1.977	0

Table 4. Point mutation frequencies. The values are normalized such that the sum equals 36. (a) Normalized point mutations per base position of SNV measured in mouse chromosome 1 and 2. The transitions were normalized which makes the comparison with the weights matrix simpler. (b) The normalized mutation matrices of (a) were made symmetrical as the conductance weights are symmetrical, too. The average value of two corresponding cells is calculated. Important figures which are discussed in the text are shown in bold.

(a) Point mutations in mouse CDS					
1	U	C	A	G	
U	0	0.9	0.2	0.3	
C	1.5	0	0.4	0.3	
A	0.2	0.3	0	0.9	
G	0.4	0.3	1.4	0	
2	U	C	A	G	
U	0	0.7	0.2	0.2	
C	1.1	0	0.2	0.2	
A	0.2	0.2	0	0.7	
G	0.2	0.2	1.1	0	
3	U	C	A	G	
U	0	3.3	0.5	0.6	
C	5.7	0	0.8	0.7	
A	0.5	0.6	0	3.5	
G	0.8	0.6	6	0	
(b) Symmetrical point mutations in mouse CDS					
1	U	C	A	G	
U	0	1.2	0.2	0.3	
C	1.2	0	0.4	0.3	
A	0.2	0.4	0	1.1	
G	0.3	0.3	1.1	0	
2	U	C	A	G	
U	0	0.9	0.2	0.2	
C	0.9	0	0.2	0.2	
A	0.2	0.2	0	0.9	
G	0.2	0.2	0.9	0	
3	U	C	A	G	
U	0	4.5	0.5	0.7	
C	4.5	0	0.7	0.7	
A	0.5	0.7	0	4.8	
G	0.7	0.7	4.8	0	

It is remarkable that the value $p_3^{\{U,C\}}$ (or mutation $U \leftrightarrow C$ in the third position) in Table 3b is the highest and the second-highest is $p_3^{\{A,G\}}$ (or $A \leftrightarrow G$). These values show wonderfully the influence of the wobble effect on the structure of the SGC which is reflected by this conductance measure. These results also suggest that the codon position can be classified according to its vulnerability to any point mutations or mistranslation at the ribosome: The weights in the matrix for position 2 are the smallest (average of 0.0033), followed by position 1 (average of 0.0037) and the highest values can be found at position 3 (average of about 3).

This order is the same when we look at point mutations found in coding sequences which confirms the results of our model. Figure 4 shows the relative frequencies of point mutations in coding sequences according to their position in a codon in mouse chromosome 1 and 2. Clearly, most of these mutations occur at position 3 which supports the findings of the conductance weights in Table 3a,b. Moreover, the ratio of the number of synonymous

(silent) mutations to the number of all point mutations is by far greatest at position 3 (about 90%). At position 1 about 10% of the point mutations are synonymous and position 2 has the smallest value of only 1%. We conclude that the wobble effect in position 3—as pointed out by the weights—promoted mutations but those which do not affect the protein sequence. Mutations at position 1 or in particular 2 will more likely lead to mutations that change the amino acid. As a consequence, those mutations are very rare.

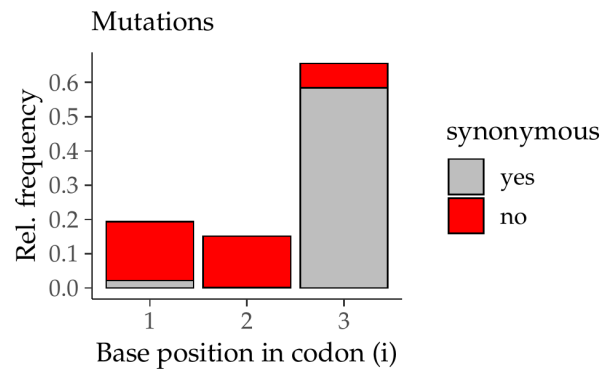


Figure 4. Relative frequencies of point mutations in coding sequences in mouse chromosome 1 and 2 according to their position in the codon. Each bar is divided into non-synonymous (red) and synonymous (or silent; in gray) mutations.

Three transition matrices for the point mutations of the coding sequences were calculated for each base position in the next step (see Table 4a). The relative transitions were (again) normalized such that the sum of all transitions equals 36 in order to be compatible with Table 3a,b. Table 4b shows the same transitions with a symmetrical matrix. This formatting has no influence on the calculation of the conductance and is helpful because the conductance weights are also symmetrical. Though the exact numbers differ from the conductance weights, the tendency is clearly the same. The two highest transitions by far are reached at base position 3 for the transitions $G \leftrightarrow A$ with a value of 4.8 (or in the directed form $G \rightarrow A$ with value 6 and its reverse $A \rightarrow G$ with value 3.5) as well as for $C \leftrightarrow U$ with a value of 4.5 (or $C \rightarrow U$ with 5.7 and its reverse $U \rightarrow C$ with 3.3). This is qualitatively consistent with the optimal weights (P_{opt}) where these two transitions also have the highest values. These results are also in line with the well-known fact that point mutations are dependent on their environment in a sequence (e.g., CpG islands) and that as a consequence the transitions $U \leftrightarrow C$ and $A \leftrightarrow G$ mutations occur more frequently [15]. Transversions like $G \leftrightarrow C$ or $G \leftrightarrow U$ are less frequent. Jiang and coworkers analyzed single point mutations (SNPs) in chimpanzees [25]. They found for exons that the mutation $C \rightarrow U$ has a relative frequency of 28.3% and $U \rightarrow C$ 11.3%. Similarly, the mutation $G \rightarrow A$ has a relative frequency of 27.3% and $A \rightarrow G$ 11.7%. All other point mutations have a relative frequency less than 4.2%. These results, again, support our findings.

The evolutionary algorithm found an optimal set of weights that led to an average conductance of about $\bar{\Phi}(C_{SGC}; P_{opt}) = 0.12$. In the following proposition we show that the lower bound of the average conductance is 0.0953.

Proposition 1. Let C_{SGC} be the partitioning of the SGC and $0 \leq p \leq \infty$ for all $p \in P$. Then the lower bounds of the average conductance is $\bar{\Phi}(C_{SGC}) > 0.0953$.

Proof. It can be shown that the average conductance of the SGC cannot be lower than $\bar{\Phi}(C_{SGC}) > 0.0953$. This is due to three reasons. Firstly, that Methionine and Tryptophan appear as a partition of size one. Secondly, that the Isoleucine partition of size three disjoints AUA and AUG. Finally, the stop signal which separates UGA and AUG. Hence, we can conclude that the conductance of Methionine and Tryptophan (Try) is one and the conductance of Isoleucine (Ile) and the stop signal is greater than zero,

i.e., $x = \phi(S_{Trp}) + \phi(S_{Ile}) > 0$. This is because all weights are $p_i^{\{N,N'\}} > 0$ by definition and it implies that $\bar{\Phi}(C_{SGC}) = (1 + 1 + x)/21 > 0.0953$. \square

This boundary does not seem to be sharp as the results of the optimization process miss the boundary by about 0.025. It suggests that the SGC is not solely optimized according to the conductance measurement. However, the fact that the results are still very good suggests that the influence of the adaption theory on evolution is not negligible.

3.2. Optimal Genetic Code Table

This section shows the results of the partitioning optimizations and Table 5 gives an overview. Any code table will always contain 21 classes or labels: 20 for the amino acids and 1 for stop codons.

Table 5. Optimized code tables with 21 classes. (a) shows the optimal genetic code table as published in [13] where all weights are set to 1. (b) is the optimization result if one takes the weights from Table 1. (c) is the optimization result if one takes the weights P_{opt} from Table 3b. Amino acids are displayed when they match the SGC.

(a) Optimal partition table C_{21} when all weights are set to 1; $\bar{\Phi}(C_{21}) \approx 0.77$									
	U		C		A		G		
U	UUU	1	UCU	7	UAU	12	UGU	17	U
	UUC		UCC		UAC		UGC		C
	UUA		UCA		UAA		UGA		A
	UUG	UCG	UAG	UGG	18 (Trp/W)	G			
C	CUU	3	CCU	9	CAU	14	CGU	19	U
	CUC		CCC		CAC		CGC		C
	CUA		CCA		CAA		CGA		A
	CUG	CCG	CAG	CGG	18	G			
A	AUU	4 (Ile/I)	ACU	10	AAU	15	AGU	20	U
	AUC		ACC		AAC		AGC		C
	AUA		ACA		AAA		AGA		A
	AUG	ACG	AAG	AGG	18	G			
G	GUU	5	GCU	11	GAU	16	GGU	21	U
	GUC		GCC		GAC		GGC		C
	GUA		GCA		GAA		GGA		A
	GUG	GCG	GAG	GGG	6	G			

(b) Optimal partition table C_M for weights P_M (see Table 3a); $\bar{\Phi}(C_M; P_M) \approx 0.56$									
	U		C		A		G		
U	UUU	1 (Pha/F)	UCU	6 (Ser/S)	UAU	10 (Tyr/Y)	UGU	17	U
	UUC		UCC		UAC		UGC		C
	UUA	2 (Leu/L)	UCA		UAA	11 (Stop)	UGA		A
	UUG		UCG		UAG		UGG		G
C	CUU	3 (Leu/L)	CCU	7 (Pro/P)	CAU	12 (His/H)	CGU	18 (Arg/R)	U
	CUC		CCC		CAC		CGC		C
	CUA		CCA		CAA	CGA	A		
	CUG	CCG	CAG	CGG	13 (Gln/Q)	G			
A	AUU	4	ACU	8 (Thr/T)	AAU	14	AGU	19 (Ser/S)	U
	AUC		ACC		AAC		AGC		C
	AUA		ACA		AAA		AGA	20 (Arg/R)	A
	AUG	ACG	AAG	AGG	G				
G	GUU	5 (Val/V)	GCU	9 (Ala/a)	GAU	15 (Asp/D)	GGU	21 (Gly/G)	U
	GUC		GCC		GAC		GGC		C
	GUA		GCA		GAA	GGA	A		
	GUG	GCG	GAG	GGG	16 (Glu/E)	G			

Table 5. Cont.

(c) Optimal partition table C_{opt} for weights P_{opt} (see Table 3b); $\bar{\Phi}(C_{opt}; P_{opt}) \approx 7.7 \times 10^{-3}$

	U		C		A		G		
U	UUU	1 (Pha/F)	UCU	6 (Ser/S)	UAU	10 (Tyr/Y)	UGU	17	U
	UUC	2 (Leu/L)	UCC		UAC	11 (Stop)	UGC		C
	UUA		UCA		UAA		UGA		A
	UUG		UCG		UAG		UGG		G
C	CUU	3 (Leu/L)	CCU	7 (Pro/P)	CAU	12	CGU	18 (Arg/R)	U
	CUC		CCC		CAC		CGC		C
	CUA		CCA		CAA		CGA		A
	CUG		CCG		CAG		CGG		G
A	AUU	4	ACU	8 (Thr/T)	AAU	13 (Asn/N)	AGU	19 (Ser/S)	U
	AUC		ACC		AAC	AGC	C		
	AUA		ACA		AAA	AGA	20 (Arg/R)	A	
	AUG		ACG		AAG	AGG		G	
G	GUU	5 (Val/V)	GCU	9 (Ala/a)	GAU	15 (Asp/D)	GGU	21 (Gly/G)	U
	GUC		GCC		GAC	GGC	C		
	GUA		GCA		GAA	GGA	A		
	GUG		GCG		GAG	GGG	G		

Before we continue with the optimization results, let us recall the optimal code table for a graph where all weights were set to 1. Such an optimal code table (denoted as C_{21}) was developed in [13] and consists of 1 fourfold degenerated group of codons (label 6 in Table 5a) and 20 groups of threefold degenerated codons. Table 5a depicts a minor modification of this code table. The classes labeled as 2, 8, 12, and 18 are the bottom of their blocks of four codons whereas the table in [13] shows these labels at the top of each block. i.e., codon UUG with label 2 would be UUU in the original table. Both tables, the original one and the one shown here, have the same average conductance of $\bar{\Phi}(C_{21}) = 146/189 \approx 0.77$. The modification makes the original code table more compatible to the SGC with respect to potential mappings to amino acids. Nevertheless, the codons of this table can only poorly be mapped to corresponding amino acids. Exceptions are the block AU* with the start codon AUG, which are Methionine and AUU, AUC, and AUA, which can be assigned to Isoleucine, and UUG which encodes for Tryptophan.

Let us move on to the optimization results. Table 5b lists the optimal table where the weights were set according to Tables 1 and 5c the optimal table which is based on optimal weights according to Table 3b. Interestingly, both code tables are divided into eleven classes of degeneration size four and ten classes of size two. Each table has three classes that cannot be assigned to amino acids in the SGC. These are in Table 5b the classes 4, 14, and 17 and in Table 5c 4, 12, and 17. Yet, the amino acids Leucine, Serine, and Arginine are each represented in two classes in the result tables.

Apparently, a graph with weights that reflect the wobble effect naturally leads to a genetic code table which is quite similar to the SGC. Moreover, the average conductance for the partition C_M is $\bar{\Phi}(C_M; P_M) = 0.56$ (rounded to two digits) and the best, i.e., minimal, set-conductance is 0.43. Given the weights P_M , the conductance could further be improved from 0.77 to 0.56. The average conductance for partition C_{opt} is $\bar{\Phi}(C_{opt}; P_{opt}) = 7.7 \times 10^{-3}$ and the minimal set-conductance is 1.9×10^{-3} . These values are even smaller. However, this is mainly caused by the optimized and thus better weights. It does not imply that the structure of the code table is better. Indeed, the tables for C_M and C_{opt} are more or less identical as mentioned above.

It should be noted that the partition C_M is only one of many possible optimization results. The EA described in the sections above, using the weights P_M , found also alternative results with the same robustness. These are tables with the same structure, i.e., tables which are permutations of the 16 row blocks which contain either one class of degeneration of size four or two classes of degeneration of size two. Consequently, the table shown is only one of many results of the EA. Yet, other results would have less in common with the SGC.

The one presented here was chosen because it is one of the best results in terms of the SGC. The result in C_{opt} , however, is the only optimum. Any permutation would lead to smaller robustness. We would like to recall that this statement is true only if we use the weights P_{opt} (see Table 3b). In conclusion we can observe that even if the tables for C_M and C_{opt} are more or less identical, yet, the weights P_{opt} reflect the SGC more accurate since P_M is too ambiguous.

3.3. Results of Optimizing Both: Weights and Structure

In this section, we show that the simultaneous optimization of weights P_{opt} and partition C_{opt} leads to a trivial result. Specifically, it will be shown below that, under some assumptions on the structure of the partition C , the (positive) weights can be chosen such that $\bar{\Phi}(C_{opt}; P_{opt})$ tends to zero. Let us first formulate an almost obvious observation:

Proposition 2. Let $G_1^P(V, E, w)$ be a given weighted graph as in Definition 1, C be a partition with at least two classes and P a weight matrix with $p_i^{\{N, N'\}} > 0$ for all weights $p_i^{\{N, N'\}} \in P$, $N \neq N'$. Then the following must be true:

$$\bar{\Phi}(C; P) > 0$$

Proof. Let us assume that $\bar{\Phi}(C; P) = 0$. In that case $\phi(S) = 0$ for all $S \in C$. This means that for each $S \in C$ $E(S, \bar{S}) = \emptyset$ holds. This implies $|C| = 1$. This is a contradiction to the assumption $|C| \geq 2$. \square

The following two observations point the way to the construction of an optimal partition that can have an arbitrarily small average conductance if the weights are chosen appropriately:

Observation 1. Let $G_1^P(V, E, w)$ be a given weighted graph as in Definition 1 with the weight matrix P , C be a partition of V with at least two classes. Let us further define

$$E(p) \subset E$$

as a set of all edges in E assigned with the weight $p \in P$. If each $(c, c') \in E(p)$ connects only codons which both belong to the same class $c, c' \in S$, thus

$$E(S, \bar{S}) \cap E(p) = \emptyset \text{ for all } S \in C,$$

then the weight value p never increases the numerator in $\phi(S)$. Hence, if such a weight p goes towards infinity $\phi(S)$ converges toward 0.

Observation 2. Let $G_1^P(V, E, w)$, P and C be defined as above. We also adopt the definition of $E(p_i^{\{N, N'\}})$ as a subset of E where $i \in \{1, 2, 3\}$ and $N, N' \in \Sigma$. If each cluster $S_j \in C$ contains at least one edge $e \in E(p_i^{\{N, N'\}})$

$$E(S_j, S_j) \cap E(p_i^{\{N, N'\}}) \neq \emptyset$$

and (see Observation 1)

$$E(S_j, \bar{S}_j) \cap E(p_i^{\{N, N'\}}) = \emptyset$$

then the average conductance $\bar{\Phi}(C; P)$ can be made arbitrarily small under the condition that all other weights in P except $p_i^{\{N, N'\}}$ are chosen small enough.

In conclusion, we can say that $\bar{\Phi}(C_{opt}; P_{opt})$ must tend to zero if C_{opt} and P_{opt} are simultaneously optimized. Hence, the results would be meaningless. Next we present an example to illustrate this fact.

Example 2. Let us now show that it is possible to choose the appropriate edge sets that satisfy the conditions of observations 1 and 2 for code tables in Table 5b,c, and the Vertebrate Mitochondrial Code.

In all three cases one can observe that if $E(p) = E(p_3^{\{C,U\}}) \cup E(p_3^{\{G,A\}})$ then $(c, c') \in E(p)$ only connect codons which both belong to the same class $c, c' \in S_i$ and each class S_i has at least one edge $e \in E(p)$. Thus, if all weights $p_i^{\{N,N'\}} \in P$ that are unequal to $p_3^{\{C,U\}}$ or $p_3^{\{G,A\}}$ are small enough and $p_3^{\{C,U\}}, p_3^{\{G,A\}}$ are chosen large enough, $\bar{\Phi}(C; P)$ can be made arbitrarily small.

4. Discussion

In this work, it was shown how the measure of conductance in relation to the codon–amino acid assignment can be used to comprehend the robustness of the genetic code with wobble-like effects. To be more precise, all possible point mutations of the genetic code were represented in the form of a weighted graph so that the weights are understood as probabilities of individual point mutations. Optimal weights could be found by means of computer optimizations and they confirm the importance of the wobble effect in the context of the SGC. Furthermore, when wobbling is allowed the robustness of the code could be improved. Our analysis of the weights and the mutations in coding sequences (Tables 3b and 4b) underlines the importance of the second position within a codon: although the absolute number of point mutations is smallest there, they more often lead to serious changes in the process of translation compared to point mutations in other positions. The next position is the first and the third base position is most robust against point mutations and thus most important for error minimization. These results coincide with the 2-1-3 model by Massey [26], which was—in a similar way—first expressed by Taylor [27] or Dragovich [28] and extended in [29]. This hypothesis claims that the evolution of the codons in the genetic code started with the middle base (position 2) in a codon [30]. In a next step, the first base (position 1) was added to the evolutionary process. Now the distance of two codons could be greater than one point mutation (e.g., AAN and UUN differ in two bases). This, according to Massey, led to some sort of error minimization. Eventually, the third base was added and again, more error minimization was possible. Intriguingly, the only constraint for including new amino acids in new tuples (codons) according to this model is that new amino acids have to be chemically similar to existing amino acids and their cognate codons. It could be shown by simulations that more than 20% of randomly generated code tables that met these constraints have better error minimization properties than the SGC.

If the optimal weights were fixed and the structure is optimized, code tables emerge that are quite similar to the SGC. This is not possible when the wobble effect is not taken into account. We speculate that the SGC has been optimized during evolution to minimize the negative effects of point mutations, however, only to a certain extent. Most likely, it is not the only driving force, which can be seen, among other things, from the fact that the conductance measure of the SGC cannot be 0.

Assuming that the evolution of the SGC is driven only by robustness against point mutations and disregarding the main idea of the frozen accident theory that it is very difficult to change the SGC, one consequence could be that optimization is still ongoing. This hypothesis is even more compatible with the Vertebrate Mitochondrial Code, since this code, with exception of the groups of sixfold degenerated codons, is already very similar to the optimal code in Table 3b. However, if one includes other hypotheses in the theory of evolution, the SGC performs very well. For instance, in [31] by Seligmann and Pollock the assignment of the stop signals in the SGC is explained. Although their argumentation was primarily not intended to explain the codon assignment to stop signals, it follows that the three codons have to be at this position in the code table, i.e., the block structure has to be contained. Additionally, the authors Wong et al. argue in [32] that the codon assignments to Methionine and Tryptophan indicate that they are late arrivals supplied by biosynthesis. The arguments presented support the hypothesis that the optimization of the robustness of

the genetic code with respect to point mutations can be considered as a driving force in the evolution of SGC, especially when wobble-like effects are included.

Let us briefly summarize the results. Firstly, we quantitatively support the known fact that the block structure of the SGC favors the wobble effect. Secondly, we show that the transitions $U \leftrightarrow C$ and $G \leftrightarrow A$ are more likely to be detected by the error correction mechanisms compared to the transversions $G \leftrightarrow U$ and $C \leftrightarrow G$. Finally, we show that the optimized probabilities of single synonymous point mutations derived from the structure of the genetic code mirror the frequencies of single point mutations found in mouse coding sequences.

Author Contributions: Conceptualization, E.F., M.G., M.S. and L.S.; data curation, M.G.; formal analysis, E.F. and L.S.; funding acquisition, M.S.; methodology, M.G. and M.S.; software, M.S.; validation, E.F., M.G., M.S. and L.S.; writing—original draft, M.G. and M.S.; writing—review and editing, E.F., M.G., M.S. and L.S. All authors have read and agreed to the published version of the manuscript.

Funding: M.S. was funded by Albert und Anneliese Konanz-Stiftung.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The used coding sequences were taken from Mouse (*M. musculus*) chromosome 1 and 2 and downloaded from <http://www.ensembl.org/biomart/martview>; accessed on 11/03/2021.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SGC Standard Genetic Code
EA Evolutionary Algorithm

References

- Kun, Á.; Radványi, Á. The evolution of the genetic code: Impasses and challenges. *Biosystems* **2018**, *164*, 217–225. [[CrossRef](#)]
- Koonin, E.V.; Novozhilov, A.S. Origin and evolution of the genetic code: The universal enigma. *IUBMB Life* **2009**, *61*, 99–111. [[CrossRef](#)]
- Pelc, S.R. Correlation between Coding-Triplets and Amino-Acids. *Nature* **1965**, *207*, 597–599. [[CrossRef](#)] [[PubMed](#)]
- Wong, J.T.F. A Co-Evolution Theory of the Genetic Code. *Proc. Natl. Acad. Sci. USA* **1975**, *72*, 1909–1912. [[CrossRef](#)] [[PubMed](#)]
- Di Giulio, M. An extension of the coevolution theory of the origin of the genetic code. *Biol. Direct* **2008**, *3*, 37. [[CrossRef](#)]
- Freeland, S.J.; Hurst, L.D. The Genetic Code Is One in a Million. *J. Mol. Evol.* **1998**, *47*, 238–248. [[CrossRef](#)] [[PubMed](#)]
- Barbieri, M. Evolution of the genetic code: The ambiguity-reduction theory. *Biosystems* **2019**, *185*, 104024. [[CrossRef](#)]
- Trifonov, E. Consensus temporal order of amino acids and evolution of the triplet code. *Gene* **2000**, *261*, 139–151. [[CrossRef](#)]
- Chin, J.W. Expanding and reprogramming the genetic code. *Nature* **2017**, *550*, 53–60. [[CrossRef](#)] [[PubMed](#)]
- Acevedo-Rocha, C.G.; Budisa, N. Xenomicrobiology: A roadmap for genetic code engineering. *Microb. Biotechnol.* **2016**, *9*, 666–676. [[CrossRef](#)]
- Błażej, P.; Wnetrzak, M.; Mackiewicz, D.; Mackiewicz, P. Basic principles of the genetic code extension. *R. Soc. Open Sci.* **2020**, *7*, 191384. [[CrossRef](#)] [[PubMed](#)]
- Fimmel, E.; Gumbel, M.; Starman, M.; Strüngmann, L. Robustness against point mutations of genetic code extensions under consideration of wobble-like effects. *Biosystems* **2021**, 104485. [[CrossRef](#)] [[PubMed](#)]
- Błażej, P.; Kowalski, D.R.; Mackiewicz, D.; Wnetrzak, M.; Alogalaa, D.A.; Mackiewicz, P. The structure of the genetic code as an optimal graph clustering problem. *bioRxiv* **2018**. [[CrossRef](#)]
- Mordret, E.; Dahan, O.; Asraf, O.; Rak, R.; Yehonadav, A.; Barnabas, G.D.; Cox, J.; Geiger, T.; Lindner, A.B.; Pilpel, Y. Systematic Detection of Amino Acid Substitutions in Proteomes Reveals Mechanistic Basis of Ribosome Errors and Selection for Translation Fidelity. *Mol. Cell* **2019**, *75*, 427–441.e5. [[CrossRef](#)] [[PubMed](#)]
- Gojobori, T.; Li, W.H.; Graur, D. Patterns of nucleotide substitution in pseudogenes and functional genes. *J. Mol. Evol.* **1982**, *18*, 360–369. [[CrossRef](#)] [[PubMed](#)]
- Alev, V.L.; Anari, N.; Lau, L.C.; Gharan, S.O. Graph Clustering using Effective Resistance. *arXiv* **2017**, arXiv: 1711.06530.

17. Lee, J.R.; Gharan, S.O.; Trevisan, L. Multiway Spectral Partitioning and Higher-Order Cheeger Inequalities. *J. ACM* **2014**, *61*, 37. [[CrossRef](#)]
18. Błażej, P.; Fimmel, E.; Gumbel, M. The Quality of Genetic Code Models in Terms of Their Robustness Against Point Mutations. *Bull. Math. Biol.* **2019**, *81*, 2239–2257. [[CrossRef](#)]
19. Błażej, P.; Wnętrzak, M.; Mackiewicz, D.; Mackiewicz, P. Optimization of the standard genetic code according to three codon positions using an evolutionary algorithm. *PLoS ONE* **2018**, *13*, e0201715. [[CrossRef](#)]
20. Mitchell, M. Genetic algorithms: An overview. *Complexity* **1995**, *1*, 31–39. [[CrossRef](#)]
21. Sastry, K.; Goldberg, D.; Kendall, G. Genetic Algorithms. In *Search Methodologies*; Burke, E.K., Kendall, G., Eds.; Springer: Boston, MA, USA, 2005; pp. 97–125. [[CrossRef](#)]
22. Scrucca, L. GA Package for Genetic Algorithms in R. *J. Stat. Softw.* **2013**, *53*. [[CrossRef](#)]
23. Fimmel, E.; Gumbel, M.; Karpuzoglu, A.; Petoukhov, S. On comparing composition principles of long DNA sequences with those of random ones. *Biosystems* **2019**, *180*, 101–108. [[CrossRef](#)]
24. Howe, K.L.; Achuthan, P.; Allen, J.; Allen, J.; Alvarez-Jarreta, J.; Amode, M.R.; Armean, I.M.; Azov, A.G.; Bennett, R.; Bhai, J.; et al. Ensembl 2021. *Nucleic Acids Res.* **2021**, *49*, D884–D891. [[CrossRef](#)] [[PubMed](#)]
25. Jiang, C.; Zhao, Z. Directionality of point mutation and 5-methylcytosine deamination rates in the chimpanzee genome. *BMC Genom.* **2006**, *7*, 316. [[CrossRef](#)]
26. Massey, S.E. A Sequential “2-1-3” Model of Genetic Code Evolution That Explains Codon Constraints. *J. Mol. Evol.* **2006**, *62*, 809–810. [[CrossRef](#)] [[PubMed](#)]
27. Taylor, F.; Coates, D. The code within the codons. *Biosystems* **1989**, *22*, 177–187. [[CrossRef](#)]
28. Dragovich, B.; Dragovich, A. p-Adic Modelling of the Genome and the Genetic Code. *Comput. J.* **2010**, *53*, 432–442. [[CrossRef](#)]
29. Massey, S.E. A Neutral Origin for Error Minimization in the Genetic Code. *J. Mol. Evol.* **2008**, *67*, 510–516. [[CrossRef](#)]
30. Massey, S.E. The neutral emergence of error minimized genetic codes superior to the standard genetic code. *J. Theor. Biol.* **2016**, *408*, 237–242. [[CrossRef](#)]
31. Seligmann, H.; Pollock, D.D. The Ambush Hypothesis: Hidden Stop Codons Prevent Off-Frame Gene Reading. *DNA Cell Biol.* **2004**, *23*, 701–705. [[CrossRef](#)]
32. Wong, J.; Ng, S.K.; Mat, W.K.; Hu, T.; Xue, H. Coevolution Theory of the Genetic Code at Age Forty: Pathway to Translation and Synthetic Life. *Life* **2016**, *6*, 12. [[CrossRef](#)] [[PubMed](#)]