BioSystems 181 (2019) 44–50

Contents lists available at ScienceDirect

BioSystems

journal homepage: www.elsevier.com/locate/biosystems

Optimization of the standard genetic code in terms of two mutation types: Point mutations and frameshifts

Małgorzata Wnętrzak*, Paweł Błażej, Paweł Mackiewicz

Faculty of Biotechnology, University of Wrocław, ul. Fryderyka Joliot-Curie 14a, 50-383 Wrocław, Poland

ARTICLE INFO

Keywords:
Adaptive hypothesis
Evolutionary algorithm
Frameshift
Genetic code
Mutation
Optimization

ABSTRACT

The distinct structure and universality of the standard genetic code (SGC) have fascinated the scientists ever since the first amino acid assignments were discovered. There are several hypotheses trying to explain the origin and evolution of this code. One of them postulates that the SGC evolved to minimize harmful effects of amino acid replacements in proteins, caused by mutations and translational errors. Many investigations concerning this hypothesis have already been carried out, but they were focused mainly on the consequences of single-point mutations. Therefore, we decided to check the influence of other types of mutations, i.e. insertions and deletions, on the robustness to amino acid replacements of the SGC. Such mutations cause shifts in the reading frame during the translation process which result in more harmful consequences in coded proteins than in the case of single-point mutations. We applied a multi-objective optimization algorithm to find the best and the worst genetic codes, regarding their robustness to both single-point and frameshift mutations, for various amino acid properties. Then we compared the features of the found codes with the properties of the standard genetic code. The results show that the SGC is not fully optimized for minimizing the effects of frameshift mutations but it is, nevertheless, much closer to the best solutions than to the worst ones. It implies that a certain tendency to minimize the costs of amino acids replacements resulting from various kinds of mutations is present in the standard genetic code.

1. Introduction

The origin and evolution of the standard genetic code (SGC) still remain a mystery, even though the code has been studied since the 1960s, when the first assignments of amino acids to codons were discovered (Khorana et al., 1966; Nirenberg et al., 1966). Scientists are puzzled by its universality among all the organisms on Earth, especially when they consider the huge number, about $1.51 \cdot 10^{84}$ (Schönauer and Clote, 1997), of all possible alternatives which encode 20 amino acids and the stop translation signal by means of 64 codons. Therefore, Crick (1968) suggested that the SGC was selected due to various factors, including random ones, from all the possibilities and then it was fixed to avoid introducing new changes which could cause misreading of codons in the already established protein sequences. However, certain regularity has been observed in the structure of the SGC. The amino acids with similar physicochemical properties are assigned to similar codons, which implies that the SGC has a degree of robustness to the effects of mutations and errors occurring during protein synthesis and therefore the code may have evolved to minimize such influences on created proteins (Sonneborn, 1965; Woese, 1965). This hypothesis was tested by many researchers (Ardell, 1998; Ardell and Sella, 2001; Di Giulio, 1989a; Di Giulio and Medugno, 1999; Epstein, 1966; Freeland and Hurst, 1998a, b; Freeland et al., 2000, 2003; Gilis et al., 2001; Goldberg and Wittes, 1966; Goodarzi et al., 2005; Haig and Hurst, 1991; Mackiewicz et al., 2008), who found the existence of the error minimization tendencies in the SGC. Moreover, Böażej et al. (2018a) showed the error-minimization properties of the codon block structure of the SGC regardless of the amino acid assignments. It was also pointed out that this structure could have been shaped by the translational inaccuracy (Błażej et al., 2019b) and could have evolved from the initial ambiguity in assignments of codons to amino acids (Barbieri, 2015; Błażej et al., 2019b). On the other hand, other analyses, especially those applying optimization algorithms, revealed that the SGC is not as optimized as it was previously assumed (Błażej et al., 2018b, 2016; de Oliveira et al., 2015, 2018; Judson and Haydon, 1999; Massey, 2008; Novozhilov et al., 2007; Santos and Monteagudo, 2011, 2017; Wnętrzak et al., 2018). Moreover, detailed analyses of natural and theoretical alternative genetic codes showed that they are more robust to amino acid replacements than the SGC (Błażej et al., 2019a, c). In agreement with that, it was postulated that the minimization of mutation errors by

* Corresponding author.
E-mail addresses: malgorzata.wnetrzak@uwr.edu.pl (M. Wnętrzak), blazej.pawel@gmail.com (P. Błażej), pamac@smorfland.uni.wroc.pl (P. Mackiewicz).

https://doi.org/10.1016/j.biosystems.2019.04.012
Received 18 March 2019; Accepted 27 April 2019
Available online 28 April 2019

the SGC was not necessary because they could have been adjusted by the direct optimization of the mutational pressure around the already established genetic code (Błażej et al., 2017, 2015; Dudkiewicz et al., 2005; Mackiewicz et al., 2008).

There are also studies suggesting that the codon-amino acid assignments in the SGC did not emerge under the selection for error minimization but the observed robustness to single changes in codons is rather a side effect of other mechanisms, e.g. the increasing diversity of amino acids subsequently added to the expanding code (Higgs, 2009; Sengupta and Higgs, 2015; Weberndorfer et al., 2003), the duplication of genes for tRNAs and aminoacyl-tRNA synthetases (Cavalcanti et al., 2000, 2004; Koonin, 2017; Koonin and Novozhilov, 2017; Massey, 2016; Stoltzfus and Yampolsky, 2007) or the evolution of biosynthetic pathways of amino acids (Di Giulio, 1997, 1999; Di Giulio, 2004, 2008, 2016, 2017, 2018; Facchiano and Di Giulio, 2018; Wong, 1975; Wong et al., 2016). Still, all of the SGC origin theories have been criticized for not being sufficient to explain all SGC features on their own (Koonin and Novozhilov, 2017; Kun and Radvanyi, 2018).

Because of the lack of agreement concerning the mechanisms of the SGC origin and evolution, the subject is still under investigation. However, most of the published results concerning the investigation of the error minimization property of the SGC are focused solely on the effects of single-point mutations or simple misreading of codons during translation, neglecting the consequences of shifts in the reading frame, caused by deletions and insertions or just slippage of ribosomes. Such mutations and errors most often result in non-functional proteins when all the subsequent codons in the coding sequence are changed. Therefore it is interesting to study the robustness of the SGC to the frameshifts. Several authors have already investigated this subject from different points of view.

Seligmann and Pollock (2004) have observed that in many organisms the codons with greater potential to form the stop translation codons after a shift of the reading frame show a greater usage and bias in their favour among synonymous codons. It can decrease energy and resource waste on non-functional proteins. Itzkovitz and Alon (2007) concluded that the SGC is nearly optimal regarding the minimization of the effect of translational frameshift errors in terms of encountering a stop codon in the middle of the frame-shifted protein coding sequence. Such codons appeared in the SGC-translated sequence much earlier than in the case of over 99% of the tested alternative theoretical codes with the codon block structure resembling that of the SGC. A similar approach was taken by Kumar and Saini (2016) who investigated the frameshift robustness of the SGC by applying a fitness function quantifying the probability with which a faulty peptide translation would be terminated, using the amino acid profile obtained from the proteome of *Escherichia coli*. They determined that even though the SGC structure seems regular and specific, it is sub-optimal for robustness to frameshift mutations, which indicates that this feature of the code has likely not been selected for. However, after applying a fitness function which combines the robustness to point mutations and frameshifts as well as the parallel coding ability, based also on the coding regions of *E. coli*, the authors came to the conclusion that the SGC is significantly better than other genetic codes because they found only a few theoretical codes which outperform the SGC in terms of this combined fitness.

The robustness to frameshifts was also tested by Geyer and Madany Mamlouk (2018) who investigated the changes in polarity of the encoded amino acids after frameshifts. However, the authors used another type of the fitness function. They did not just look for stop codons in frame-shifted sequences, but they summed up all possible changes in the polarity of amino acids encoded by codons before and after a frameshift. Their conclusion was that the SGC is efficient in minimizing the effects of frameshift in terms of conserving the polarity of amino acids, although better codes can be found. They also stated that it becomes significantly more difficult to find codes better than the SGC regarding not only the robustness to frameshifts but also to point mutations and translational errors. However, the deduction was based on the comparison of the SGC with only one million random theoretical codes, which does not seem a sample large enough to get reliable conclusions. For that reason, we decided to use a similar fitness function for testing the SGC robustness to frameshifts, but to avoid comparing the SGC with random alternatives, we used an evolutionary algorithm to find the codes which minimize and maximize the fitness function, to get a bigger picture of the SGC properties in relation to the whole space of theoretical possibilities. We also found the best and the worst codes regarding the robustness to point mutations. In both cases we considered two of the amino acid properties, polarity and molecular volume. Moreover, in order to avoid assigning any arbitrary weights to the two mutation types in the objective function, we used a two-criteria optimization algorithm to find the codes optimized simultaneously for both types of mutations and errors, i.e. nucleotide substitutions and frameshifts. We showed that in all the described cases it is easy to find many theoretical codes that minimize the given fitness function better than the SGC, which itself is quite robust, considering the space of all the theoretical alternatives. The results are also dependent on the amino acid property. The SGC seems much better optimized to changes in polarity than molecular volume of amino acids.

2. Methods

2.1. Models of genetic codes

We searched for the optimal solutions in two sets of theoretical genetic codes, i.e. search spaces. The first one, called the codon block structure model (CB), consists of the codes characterized by the same codon block structure as the SGC but with permuted assignments of amino acids to blocks of codons. The second set, called the unrestricted structure model (US), includes all possible codes which encode 20 amino acids without any further restrictions on the structure. In both models we assumed the three stop translation codons fixed as in the SGC.

2.2. Evolutionary algorithms

As our method of finding the optimal theoretical codes, we chose the Evolutionary Algorithms (EAs) (Sivanandam and Deepa, 2008). Their simplicity, flexibility, and robustness to changing conditions make them a very useful tool in solving optimization problems, especially when analytic methods are not feasible due to the properties of the search space.

EAs can be used in both single- and multi-objective optimization problems. For our purposes we needed a procedure which could be applied to both cases and was easily adapted to the genetic code optimization task. Therefore, we chose the Strength Pareto Evolutionary Algorithm (SPEA2) (Zitzler et al., 2002) which was crafted mainly for the multi-objective optimization and finding an approximation of the set of optimal solutions. We developed a few versions of this algorithm customized to our genetic code optimization tasks and implemented them in the C++ language. These algorithms were also applied in our previous studies on the genetic code optimality (Błażej et al., 2018b, 2016; Wnętrzak et al., 2018).

To start an Evolutionary Algorithm, a population of individuals randomly chosen from the search space is necessary. These potential solutions are evaluated according to the fitness function which assesses their quality, and then they are specifically modified by the genetic operators in order to produce new individuals from the search space for assessing. In every step of the algorithm a new generation of solutions is created by the selection procedure, which chooses most often high quality individuals from the previous generation. Then the modification, fitness evaluation and selection are applied again and repeated until a stopping rule is activated or the obtained solutions become stable (Sivanandam and Deepa, 2008).

Depending on the considered search space, problem-specific genetic

operators were either adapted from the ones already described in the literature or constructed anew (Błażej et al., 2018b, 2016; Wnętrzak et al., 2018). Under the CB model, the genetic codes were represented by vectors of 21 characters corresponding to 20 amino acids and the stop translation signal assigned to established codon blocks. As the mutation operator we used a simple exchange of the amino acids assigned to two randomly selected codon blocks. For the crossover operator, we adapted the Position Based Crossover (POS) operator (Syswerda, 1991). According to this operator, a determined number of amino acids of the parental code P_1 is randomly selected and assigned to the corresponding codon blocks in the offspring. The remaining codon blocks in the offspring have amino acids assigned according to the codon blocks in the parental code P_2, with optional changes if a given amino acid is already present in the offspring. Under the US model, the genetic codes were represented by vectors of 64 elements corresponding to amino acids assigned to the respective codons. As the mutation operator we used a procedure, which selects two codons at random, and for the amino acids encoded by these codons, swaps all the codons originally assigned to these amino acids. Furthermore, we had to propose a different crossover operator than in the case of the CB model. We developed a procedure, which starts by copying the parental codes P_1 and P_2 onto the offspring O_1 and O_2. Then an amino acid is randomly selected. If this amino acid is encoded by different codons in P_1 and P_2, the assignments of these codons are exchanged within O_1 and O_2, thus creating new codes with structures inherited from the parental codes.

For choosing individuals to next generations, we applied binary tournament selection (Blickle and Thiele, 1996), which means that out of two randomly chosen solutions only one was transferred to the next generation, with the probability directly proportional to its fitness value.

The most important part of each Evolutionary Algorithm is the incorporation of a relevant fitness function which describes how good a given individual is. It allows the selection for choosing the most promising solutions and thus guides the search for the optimal one (Sivanandam and Deepa, 2008). The calculation of the fitness function values is based on the values of the objective functions, which quantify the optimality of the solutions regarding one respective criterion. In the case of the single-objective optimization, the fitness function is often the same as the objective function since there is only one criterion of optimality considered. However, in the multi-objective optimization, there is a vector of objective functions values assigned to every individual in the population and the fitness value is calculated separately. In the SPEA2 algorithm, the fitness function is based on the Pareto dominance concept, which states that the solution S_1 dominates the solution S_2 if no component of S_1 is worse than the corresponding component of S_2 and at least one component of S_1 is better than the respective one of S_2 (Coello Coello et al., 2007). The components in our case are the values of the objective functions. In order to calculate the fitness values, first we assign to each individual i a strength value $S(i)$ representing the number of solutions that it dominates. Then the raw fitness $R(i)$ is calculated, according to the formula:

$$R(i) = \sum_{j \in P_t + \bar{P}_t, j<i} S(j)$$

where $P_t + \bar{P}_t$ is the set of all the individuals from the current population P_t and the archive set $\bar{P}_t$ of the best solutions up-to-date, and $j<i$ means that the individual i is dominated by the individual j. Additionally, we incorporate the density information to discriminate between individuals with identical raw fitness values. Thus, for each individual, the distances in the objective space to all the individuals from $P_t + \bar{P}_t$ are calculated and stored in a list sorted in increasing order. Then, for each solution i we choose the 5th element of the list, denoted as σ_i^5, and we calculate the corresponding density $D(i)$, according to the formula:

$$D(i) = \frac{1}{\sigma_i^5 + 2}$$

The final fitness value $F(i)$ of the individual i is the sum of $R(i)$ and $D(i)$ (Zitzler et al., 2004). The fitness of an individual is computed drawing upon the number of individuals dominated by the given individual and the number of individuals dominating the given individual (Zitzler et al., 2002).

2.3. The objective functions

In order to determine the robustness of a given genetic code to amino acid replacements, we used two types of objective functions, one for each type of mutation. Similarly to other authors (de Grey, 2005; de Oliveira et al., 2015; Freeland and Hurst, 1998a; Haig and Hurst, 1991; Santos and Monteagudo, 2010), we calculated the sum of squared differences between an amino acid index values representing a given property of amino acids encoded by their original codons and the ones encoded by the mutated codons. In the case of nucleotide substitutions, the mutated codons differed in only one codon position from the original one, which means that we considered 576 pairs of amino acids (64 codons times nine possible single nucleotide substitutions). In the case of insertions and deletions, the mutated codons resulting from the shifts of the reading frame were considered, thus we calculated the differences between amino acids encoded by 512 pairs of codons (64 codons times four possible nucleotides in the third codon position after +1 frameshift plus 64 codons times four possible nucleotides in the first codon position after -1 frameshift). This type of objective function seems to be especially relevant for assessing the damage in proteins caused by frameshifts, because in the case of such mutations and errors, all the codons in the sequence are changed and the chosen objective function takes into account the sum of all possible changes in encoded amino acids resulting from a shift of the reading frame.

For this study, as the indices quantifying amino acid properties we chose a polarity scale (Mathew and Luthey-Schulten, 2008) and molecular volume values (Grantham, 1974), as it is suggested that the conservation of these properties in proteins was important in the evolution of the genetic code (Di Giulio, 1989a, b; Facchiano and Di Giulio, 2018; Freeland and Hurst, 1998a; Haig and Hurst, 1991; Santos and Monteagudo, 2010). The values of the respective amino acid indices are presented in Table 1.

From the given amino acid index values we computed a matrix of

Table 1
The values of the amino acid indices representing the polarity (Mathew and Luthey-Schulten, 2008) and molecular volume (Grantham, 1974) of amino acids.

Amino acid	Polarity	Molecular volume
Ala	6.5	31
Arg	8.6	124
Asn	9.6	56
Asp	12.2	54
Cys	4.3	55
Gln	8.9	85
Glu	13.6	83
Gly	9.0	3
His	7.9	96
Ile	5.0	111
Leu	4.4	111
Lys	10.2	119
Met	5.0	105
Phe	4.5	132
Pro	6.1	32.5
Ser	7.5	32
Thr	6.2	61
Trp	4.9	170
Tyr	7.7	136
Val	6.2	84

squared differences between these values and then standardized the matrix by dividing each element by the maximum element of this matrix. Hence, we could compare the results for the different amino acid indices.

The formula for calculating the objective function value for a given genetic code may be presented as follows:

$$F_i(code) = \sum_{<c_1,c_2> \in C} [p_i(c_1) - p_i(c_2)]^2,$$

where: $F_i(code)$ is the value of the objective function for a given genetic code (*code*) and objective i, C is the set of all pairs of codons before and after mutation, c_1 and c_2 are codons, $p_i(c_1)$ and $p_i(c_2)$ are the values of the amino acid index i for the amino acids encoded by the codons c_1 and c_2, respectively. Depending on the set of pairs of codons, we were able to calculate the objective functions values for point mutations F^{sub} as well as insertions and deletions F^{fr}.

The aim of the optimization procedure was to minimize or maximize all the considered objective functions in order to find the genetic codes with the smallest or the largest costs of amino acid replacements, regarding given amino acid properties.

2.4. Measures of distances between codes

To assess the level of optimality of the SGC in relation to the best (minimizing amino acid replacements costs) and the worst (maximizing amino acid replacements costs) solutions found by our algorithm, we used a few analogous measures based on the Euclidean distances between the vectors of the objective functions values for the given genetic codes (Błażej et al., 2018b). In the case of the single-objective optimization, we applied the m_s measure:

$$m_s = \frac{db}{dbw} * 100,$$

where db is the Euclidean distance between the SGC and the best solution and dbw is the Euclidean distance between the best and the worst solution.

To analyse the results of the multi-objective optimization, we used two measures, m_{min} and m_{mean}:

$$m_{min} = \frac{db_{min}}{db_{min} + dw_{min}} * 100,$$

where db_{min} is the minimum Euclidean distance between the SGC and the Pareto set of the best solutions, whereas dw_{min} is the minimum Euclidean distance between the SGC and the Pareto set of the worst solutions:

$$m_{mean} = \frac{db_{mean}}{db_{mean} + dw_{mean}} * 100,$$

where db_{mean} is the mean Euclidean distance between the SGC and the Pareto set of the best solutions, whereas dw_{mean} is the mean Euclidean distance between the SGC and the Pareto set of the worst solutions.

All three measures may take values in the range from 0% to 100%. The values below 50% indicate that the SGC is closer to the group of codes minimizing amino acid replacements costs rather than to the group maximizing these costs. The values above 50% mean that the SGC is closer to the latter group.

2.5. Simulation procedures

In order to find the optimal theoretical genetic codes, regarding the robustness to amino acid replacements, we run a few types of simulations using adequate versions of our customized algorithm. However, the main parameters were set the same for each type of the simulation and the main factor that made them different from each other was the number of optimization criteria and the kind of the objective function. We decided to consider the following cases:

1) single-objective optimization regarding polarity, minimizing and maximizing F^{fr},
2) single-objective optimization regarding molecular volume, minimizing and maximizing F^{fr},
3) single-objective optimization regarding polarity, minimizing and maximizing F^{sub},
4) single-objective optimization regarding molecular volume, minimizing and maximizing F^{sub},
5) two-objective optimization regarding polarity, minimizing and maximizing both F^{sub} and F^{fr},
6) two-objective optimization regarding molecular volume, minimizing and maximizing both F^{sub} and F^{fr}.

For all types of simulations we started with a population of 2800 randomly chosen codes and the same number of codes in each consecutive generation. The Pareto set consisted of 700 individuals. Each simulation was run up to 3000 steps and was repeated 20 times in the case of the multi-objective optimization and 50 times for the single-objective optimization. We applied the objective functions F^{sub} and F^{fr} described earlier. In each step of the simulation the operators of mutation and crossover were applied to, respectively, 90% and 30% of the individuals in the population.

In the Results and Discussion section, when referring to the Pareto set obtained in any kind of simulation, we mean a combined set of all the optimized codes from the repeated runs, i.e. 20•700 = 14,000 codes from the multi-objective optimization and 50•700 = 35,000 codes from the single-objective optimization.

3. Results and discussion

3.1. Single-objective optimization

First we applied the single-objective version of our algorithm to find the genetic codes optimized regarding nucleotide substitutions and frameshifts separately (cases 1–4). The simulations were carried out for two amino acid properties, polarity and molecular volume. After finding the best and the worst theoretical codes regarding given criteria, we calculated the Euclidean distances between the standard genetic code and these theoretical alternatives. Then we used them to compute the values of the measure m_s, which describes how much optimized the SGC is, compared to the best and the worst possible solutions. The results are presented in Table 2.

All the calculated values of the m_s measure are smaller than 50%, which means that in the case of the single-objective optimization regarding the polarity or the molecular volume of amino acids, the SGC is definitely closer to the theoretical codes minimizing the costs of amino acid replacements than to the codes maximizing these costs. However, the exact m_s values clearly depend on the considered amino acid property, the type of the search space of theoretical codes, and the type of mutation. We found that for both kinds of mutations, the SGC is much closer to the best solutions regarding the polarity property than in the case of the molecular volume. It suggests that the polar properties of amino acids were more important in optimization of the SGC than their size. This assumption seems reasonable taking into account that polarity is a parameter which better differentiates amino acid functions. Our results do not contradict those of other authors (Haig and Hurst,

Table 2
The values of the m_s measure [%] for the SGC, calculated under the CB and US models regarding the robustness to point mutations and frameshifts.

AA index	Point mutations		Frameshifts	
	CB model	US model	CB model	US model
Polarity	8.04	7.69	12.28	9.91
Molecular volume	31.5	16.71	43.76	19.94

1991), who tested the genetic code optimality regarding the robustness to changes in the polarity and the molecular volume properties of encoded amino acids, caused by nucleotide substitutions.

The m_s values calculated for the codes with the unrestricted structure are smaller than those for the codes with the blocks of codons the same as in the SGC. The differences are not so big in the case of the optimization regarding the polarity. The m_s values under the CB model are only 1.05 and 1.24 times bigger than the respective values under the US model. In contrast, the difference is considerable for the optimization regarding the molecular volume. The m_s values under the CB model are 1.89 and 2.19 times bigger than the values under the US model. However, the differences between these two models result from the fact that in the larger, unrestricted space it is possible to find more codes with large values of the objective functions, which leads to the increase of the denominator in m_s and, consequently, smaller values of the m_s measure.

The differences in the m_s values are also evident when we compare the results for frameshifts and point mutations. The respective values regarding the optimization for frameshifts robustness are slightly larger (1.19–1.53 times) than the values obtained from the optimization in regard to robustness to point mutations. It suggests that the SGC is more robust to the changes in proteins caused by single nucleotide substitutions than to the changes resulting from the shifts of the reading frame.

3.2. Two-objective optimization

After testing the robustness of the SGC to amino acid replacements caused by either single-point or frameshift mutations, we decided to check this error minimizing property for both types of mutations combined (cases 5 and 6). Similarly to the single-objective optimization case, we found the best and the worst theoretical codes regarding our optimization criteria. Each genetic code had two values of the objective functions assigned. Therefore, we could present the codes as points in the two-dimensional space (Figs. 1–4). In all cases, the initial population of random codes is surrounded from two opposite sides by the Pareto fronts of best and worst codes. Under the CB model (Figs. 1 and 3), this population is located almost in the centre of the whole space, while in the case of the US model (Figs. 2 and 4), it is shifted towards the best codes. It means that the codes, which were generated by the assignment of 20 amino acids with equal probabilities to at least one of the 61 codons, can show a tendency to minimize the amino acid replacement costs when compared with the extremely bad codes. The

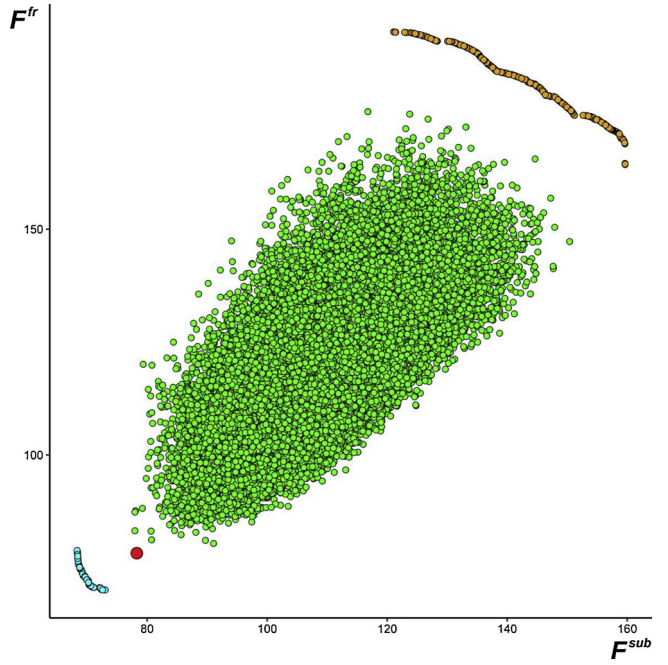

Fig. 1. The values of the objective functions F^{sub} (horizontal axis) and F^{fr} (vertical axis) for the SGC (red dot), the best codes (blue dots), the worst codes (orange dots) and the initial population of random codes for the evolutionary algorithm (green dots), calculated for the polarity property and among the codes under the CB model.

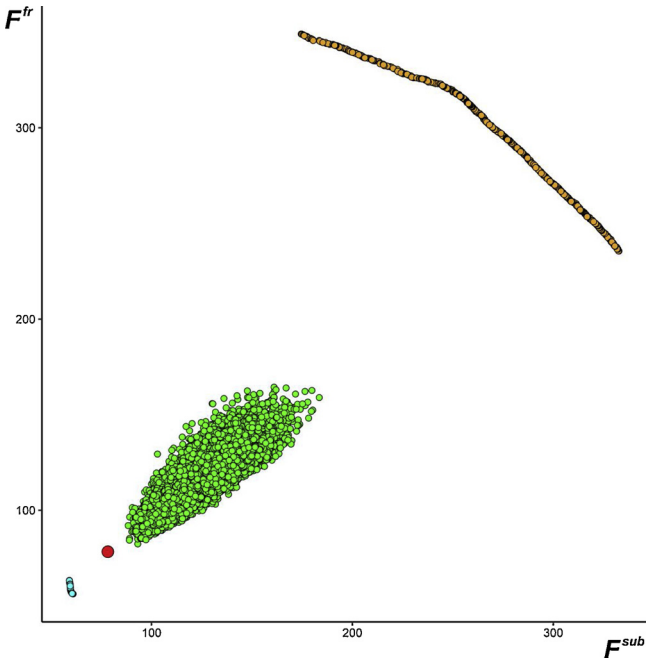

Fig. 2. The values of the objective functions F^{sub} (horizontal axis) and F^{fr} (vertical axis) for the SGC (red dot), the best codes (blue dots), the worst codes (orange dots) and the initial population of random codes for the evolutionary algorithm (green dots), calculated for the polarity property and among the codes under the US model.

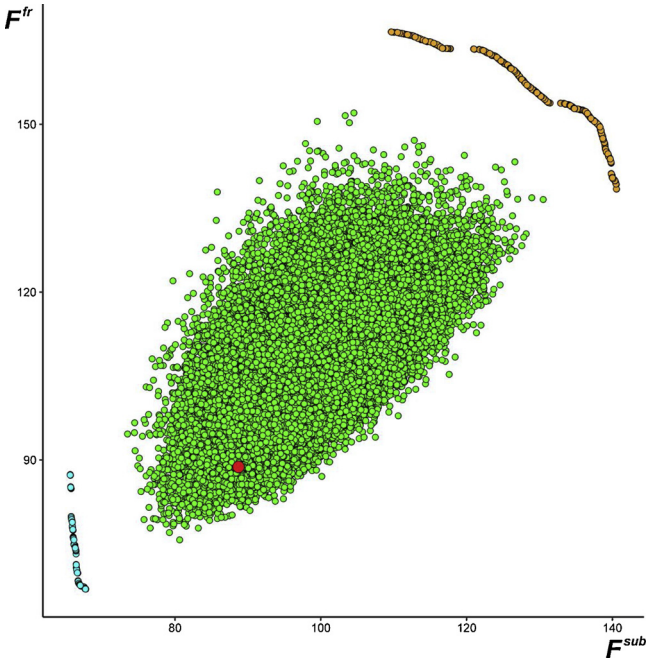

Fig. 3. The values of the objective functions F^{sub} (horizontal axis) and F^{fr} (vertical axis) for the SGC (red dot), the best codes (blue dots), the worst codes (orange dots) and the initial population of random codes for the evolutionary algorithm (green dots), calculated for the molecular volume property and among the codes under the CB model.

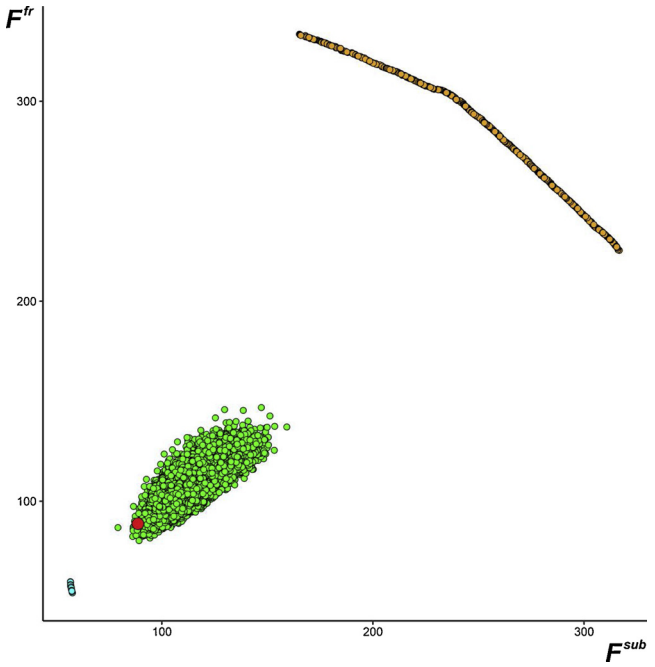

Fig. 4. The values of the objective functions F^{sub} (horizontal axis) and F^{fr} (vertical axis) for the SGC (red dot), the best codes (blue dots), the worst codes (orange dots) and the initial population of random codes for the evolutionary algorithm (green dots), calculated for the molecular volume property and among the codes under the US model.

Table 3
The values of the m_{min} and m_{mean} measures [%] for the SGC, calculated under the CB and US models.

AA index	CB model		US model	
	m_{min}	m_{mean}	m_{min}	m_{mean}
Polarity	9.45	11.15	9.45	10.15
Molecular volume	35.7	40.84	20.31	20.44

most interesting is the position of the SGC. When polarity was included in the objective function, the SGC was located between the random population and the best codes (Figs. 1 and 2). When the codes were optimized in terms of differences in molecular volume of replaced amino acids, the SGC was located among the random codes regardless of the genetic code model applied (Figs. 3 and 4). It means that the SGC is much less optimized in terms of the second property. Nevertheless, in each case, the SGC is located much closer to the Pareto front of the best codes than the worst ones. It indicates that the SGC shows a tendency to minimize amino acid replacements costs but it is not perfect.

To quantify the locations of the SGC in the given space of theoretical codes, especially in comparison with the best and the worst solutions, we computed the minimum and mean Euclidean distances between the SGC and the groups of the best and the worst codes and used them to calculate the values of the m_{min} and m_{mean} measures, which describe how much the SGC is optimized in comparison to the best and the worst alternatives. The obtained numbers are presented in the Table 3.

All the values are below 50%, which indicates that the SGC is definitely closer to the codes minimizing the effects of amino acid replacements caused by different kinds of mutations than to the codes maximizing these effects. Similarly to the single-objective optimization case, the SGC seems better optimized regarding the polarity property (with the m_{min} and m_{mean} values between 9% and 12%) than the molecular volume (with the m_{min} and m_{mean} values between 20% and 41%). Moreover, for the optimization regarding the polarity property, the extension of the search space does not provide any better results, the m_{min} and m_{mean} values calculated under the US model are almost the same as the respective ones computed under the CB model. However, in the case of the molecular volume as the objective of the optimization, there is a significant difference between the respective m_{min} and m_{mean} values calculated under both genetic code models. The values under the CB model are 1.76 and 2 times bigger than the respective ones under the US model. These facts suggest that, regarding the polarity property, the robustness levels of the genetic codes with the specific codon block structure are of the same range as the robustness levels of the codes without such structural restrictions. However, in the case of the molecular volume, we can find much worse solutions among the codes of the unrestricted structure than in the group of codes with the codon block structure as in the SGC.

Comparing the results for single- (Table 2) and two-objective optimization (Table 3), we can sate that the SGC performs worse under the latter case, when compared with the single optimization for point mutations. In contrast, the SGC is slightly better under the two-objective optimization in comparison to conditions, when only frameshift costs were optimized.

4. Conclusions

In this work we presented the results of our investigation of the robustness of the standard genetic code to nucleotide point mutations and frameshifts. By applying an evolutionary algorithm, we found the codes which minimize and maximize the objective functions under considered optimization criteria. Thus, the SGC was compared with the best and the worst alternatives instead of a small group of random possibilities, which makes our results more reliable and closer to the reality than other approaches. The main result is that the SGC is close to the group of codes minimizing the changes in the polarity of amino acids, caused by nucleotide substitutions, frameshifts, and both types of mutations simultaneously, but in each of these cases it is possible to find better alternatives than the SGC, both among the codes of the codon block structure the same as in the SGC, as well as among the codes of the unrestricted structure. The SGC is also more robust to amino acid replacements caused by nucleotide substitutions than to changes introduced by frameshifts. It may be due to the fact that point mutations occur more frequently than frameshift mutations. However, at the beginning of the genetic code evolution, the translational machinery was much less accurate and slippages of the proto-ribosome were highly probable, which could cause the ambiguity of the code (Barbieri, 2015). It was also proposed that frameshifting could allow the selection of triplets from ennuplets, i.e. stretches of proto-mRNAs larger than three bases involved in initial coding at the early stages of the genetic code evolution (Di Giulio et al., 2014). Therefore, the present structure of the SGC may reflect this stage and buffer the effects of frameshift mutations. The differences in the performance of the SGC optimized under point mutations and frameshifts are smaller when we consider changes in the polarity of amino acids than their molecular volume. However, for any type of mutation, the structure of the SGC is less robust to the changes in the molecular volume of amino acids than their polarity, although it is still closer to the group of the best alternatives than the worst ones.

Acknowledgement

This work was supported by the National Science Centre, Poland (Narodowe Centrum Nauki, Polska) under Grant number 2017/27/N/NZ2/00403.

References

Ardell, D.H., 1998. On error minimization in a sequential origin of the standard genetic code. J. Mol. Evol. 47, 1–13.

Ardell, D.H., Sella, G., 2001. On the evolution of redundancy in genetic codes. J. Mol. Evol. 53, 269–281.
Barbieri, M., 2015. Evolution of the genetic code: the ribosome-oriented model. Biol. Theory 10, 301–310.
Błażej, P., Miasojedow, B., Grabińska, M., Mackiewicz, P., 2015. Optimization of mutation pressure in relation to properties of protein-coding sequences in bacterial genomes. PLoS One 10, e0130411.
Błażej, P., Wnętrzak, M., Mackiewicz, P., 2016. The role of crossover operator in evolutionary-based approach to the problem of genetic code optimization. Biosystems 150, 61–72.
Błażej, P., Mackiewicz, D., Grabińska, M., Wnętrzak, M., Mackiewicz, P., 2017. Optimization of amino acid replacement costs by mutational pressure in bacterial genomes. Sci. Rep. 7, 1061.
Błażej, P., Kowalski, D., Mackiewicz, D., Wnętrzak, M., Aloqalaa, D., Mackiewicz, P., 2018a. The Structure of the Genetic Code As an Optimal Graph Clustering Problem. https://www.biorxiv.org/content/early/2018/05/28/332478.
Błażej, P., Wnętrzak, M., Mackiewicz, D., Mackiewicz, P., 2018b. Optimization of the standard genetic code according to three codon positions using an evolutionary algorithm. PLoS One 13, e0201715.
Błażej, P., Wnętrzak, M., Mackiewicz, P., 2018c. The importance of changes observed in the alternative genetic codes. Proceedings of the 11th International Joint Conference on Biomedical Engineering Systems and Technologies - Volume 4: BIOINFORMATICS. pp. 154–159.
Błażej, P., Wnetrzak, M., Mackiewicz, D., Gagat, P., Mackiewicz, P., 2019a. Many alternative and theoretical genetic codes are more robust to amino acid replacements than the standard genetic code. J. Theor. Biol. 464, 21–32.
Błażej, P., Wnetrzak, M., Mackiewicz, D., Mackiewicz, P., 2019b. The influence of different types of translational inaccuracies on the genetic code structure. BMC Bioinformatics 20, 114.
Blickle, T., Thiele, L.A., 1996. Comparison of selection schemes used in evolutionary algorithms. Evol. Comput. 4, 361–394.
Cavalcanti, A.R., Neto, B.D., Ferreira, R., 2000. On the classes of aminoacyl-tRNA synthetases and the error minimization in the genetic code. J. Theor. Biol. 204, 15–20.
Cavalcanti, A.R.O., Leite, E.S., Neto, B.B., Ferreira, R., 2004. On the classes of aminoacyl-tRNA synthetases, amino acids and the genetic code. Orig. Life Evol. Biosphys. 34, 407–420.
Coello Coello, C., Lamont, G.B., van Veldhuizen, D., 2007. Evolutionary Algorithms for Solving Multi-Objective Problems. Springer Science + Business Media, LLC, New York, USA.
Crick, F.H., 1968. The origin of the genetic code. J. Mol. Biol. 38, 367–379.
de Grey, A.D.N.J., 2005. Forces maintaining organellar genomes: is any as strong as genetic code disparity or hydrophobicity? Bioessays 27, 436–446.
de Oliveira, L.L., de Oliveira, P.S., Tinos, R., 2015. A multiobjective approach to the genetic code adaptability problem. BMC Bioinform. 16, 52.
de Oliveira, L.L., Freitas, A.A., Tinós, R., 2018. Multi-objective genetic algorithms in the study of the genetic code's adaptability. Inf. Sci. (Ny) 425, 48–61.
Di Giulio, M., 1989a. The extension reached by the minimization of the polarity distances during the evolution of the genetic code. J. Mol. Evol. 29, 288–293.
Di Giulio, M., 1989b. Some aspects of the organization and evolution of the genetic code. J. Mol. Evol. 29, 191–201.
Di Giulio, M., 1997. On the origin of the genetic code. J. Theor. Biol. 187, 573–581.
Di Giulio, M., 1999. The coevolution theory of the origin of the genetic code. J. Mol. Evol. 48, 253–255.
Di Giulio, M., 2004. The coevolution theory of the origin of the genetic code. Phys. Life Rev. 1, 128–137.
Di Giulio, M., 2008. An extension of the coevolution theory of the origin of the genetic code. Biol. Direct 3, 37.
Di Giulio, M., 2016. The lack of foundation in the mechanism on which are based the physico-chemical theories for the origin of the genetic code is counterposed to the credible and natural mechanism suggested by the coevolution theory. J. Theor. Biol. 399, 134–140.
Di Giulio, M., 2017. Some pungent arguments against the physico-chemical theories of the origin of the genetic code and corroborating the coevolution theory. J. Theor. Biol. 414, 1–4.
Di Giulio, M., 2018. A discriminative test among the different theories proposed to explain the origin of the genetic code: the coevolution theory finds additional support. Biosystems 169-170, 1–4.
Di Giulio, M., Medugno, M., 1999. Physicochemical optimization in the genetic code origin as the number of codified amino acids increases. J. Mol. Evol. 49, 1–10.
Di Giulio, M., Moracci, M., Cobucci-Ponzano, B., 2014. RNA editing and modifications of RNAs might have favoured the evolution of the triplet genetic code from an ennuplet code. J. Theor. Biol. 359, 1–5.
Dudkiewicz, A., Mackiewicz, P., Nowicka, A., Kowalezuk, M., Mackiewicz, D., Polak, N., Smolarczyk, K., Banaszak, J., Dudek, M.R., Cebrat, S., 2005. Correspondence between mutation and selection pressure and the genetic code degeneracy in the gene evolution. Future Gener. Comput. Syst. 21, 1033–1039.
Epstein, C.J., 1966. Role of the amino-acid "code" and of selection for conformation in the evolution of proteins. Nature 210, 25–28.
Facchiano, A., Di Giulio, M., 2018. The genetic code is not an optimal code in a model taking into account both the biosynthetic relationships between amino acids and their physicochemical properties. J. Theor. Biol.
Freeland, S.J., Hurst, L.D., 1998a. The genetic code is one in a million. J. Mol. Evol. 47, 238–248.
Freeland, S.J., Hurst, L.D., 1998b. Load minimization of the genetic code: history does not explain the pattern. Proc. R. Soc. B-Biol. Sci. 265, 2111–2119.
Freeland, S.J., Knight, R.D., Landweber, L.F., Hurst, L.D., 2000. Early fixation of an optimal genetic code. Mol. Biol. Evol. 17, 511–518.
Freeland, S.J., Wu, T., Keulmann, N., 2003. The case for an error minimizing standard genetic code. Orig. Life Evol. Biosphys. 33, 457–477.
Geyer, R., Madany Mamlouk, A., 2018. On the efficiency of the genetic code after frameshift mutations. Peerj 6, e4825.
Gilis, D., Massar, S., Cerf, N.J., Rooman, M., 2001. Optimality of the genetic code with respect to protein stability and amino-acid frequencies. Genome Biol. 2 research0049.0041–0049.0012.
Goldberg, A.L., Wittes, R.E., 1966. Genetic code: aspects of organization. Science 153, 420–424.
Goodarzi, H., Najafabadi, H.S., Nejad, H.A., Torabi, N., 2005. The impact of including tRNA content on the optimality of the genetic code. Bull. Math. Biol. 67, 1355–1368.
Grantham, R., 1974. Amino acid difference formula to help explain protein evolution. Science 185, 862–864.
Haig, D., Hurst, L.D., 1991. A quantitative measure of error minimization in the genetic code. J. Mol. Evol. 33, 412–417.
Higgs, P.G., 2009. A four-column theory for the origin of the genetic code: tracing the evolutionary pathways that gave rise to an optimized code. Biol. Direct 4, 16.
Itzkovitz, S., Alon, U., 2007. The genetic code is nearly optimal for allowing additional information within protein-coding sequences. Genome Res. 17, 405–412.
Judson, O.P., Haydon, D., 1999. The genetic code: what is it good for? An analysis of the effects of selection pressures on genetic codes. J. Mol. Evol. 49, 539–550.
Khorana, H.G., Buchi, H., Ghosh, H., Gupta, N., Jacob, T.M., Kossel, H., Morgan, R., Narang, S.A., Ohtsuka, E., Wells, R.D., 1966. Polynucleotide synthesis and the genetic code. Cold Spring Harb. Symp. Quant. Biol. 31, 39–49.
Koonin, E.V., 2017. Frozen accident pushing 50: stereochemistry, expansion, and chance in the evolution of the genetic code. Life Basel (Basel) 7, 22.
Koonin, E.V., Novozhilov, A.S., 2017. Origin and evolution of the universal genetic code. Annu. Rev. Genet. 51, 45–62.
Kumar, B., Saini, S., 2016. Analysis of the optimality of the standard genetic code. Mol. Biosyst. 12, 2642–2651.
Kun, A., Radvanyi, A., 2018. The evolution of the genetic code: impasses and challenges. Biosystems 164, 217–225.
Mackiewicz, P., Biecek, P., Mackiewicz, D., Kiraga, J., Baczkowski, K., Sobczynski, M., Cebrat, S., 2008. Optimisation of asymmetric mutational pressure and selection pressure around the universal genetic code. Computational Science - ICCS 2008, Pt 3. Lecture Notes Comput. Sci. 5103, 100–109.
Massey, S.E., 2008. A neutral origin for error minimization in the genetic code. J. Mol. Evol. 67, 510–516.
Massey, S.E., 2016. The neutral emergence of error minimized genetic codes superior to the standard genetic code. J. Theor. Biol. 408, 237–242.
Mathew, D.C., Luthey-Schulten, Z., 2008. On the physical basis of the amino acid polar requirement. J. Mol. Evol. 66, 519–528.
Nirenberg, M., Caskey, T., Marshall, R., Brimacombe, R., Kellogg, D., Doctor, B., Hatfield, D., Levin, J., Rottman, F., Pestka, S., Wilcox, M., Anderson, F., 1966. The RNA code and protein synthesis. Cold Spring Harb. Symp. Quant. Biol. 31, 11–24.
Novozhilov, A.S., Wolf, Y.I., Koonin, E.V., 2007. Evolution of the genetic code: partial optimization of a random code for robustness to translation error in a rugged fitness landscape. Biol. Direct 2, 24.
Santos, J., Monteagudo, A., 2010. Study of the genetic code adaptability by means of a genetic algorithm. J. Theor. Biol. 264, 854–865.
Santos, J., Monteagudo, A., 2011. Simulated evolution applied to study the genetic code optimality using a model of codon reassignments. BMC Bioinform. 12, 56.
Santos, J., Monteagudo, A., 2017. Inclusion of the fitness sharing technique in an evolutionary algorithm to analyze the fitness landscape of the genetic code adaptability. BMC Bioinform. 18, 195.
Schönauer, S., Clote, P., 1997. How optimal is the genetic code? In: Frishman, D., Mewes, H.W. (Eds.), Computer Science and Biology. Proceedings of the German Conference on Bioinformatics (GCB'97) Sep 21-24, pp. 65–67.
Seligmann, H., Pollock, D.D., 2004. The ambush hypothesis: hidden stop codons prevent off-frame gene reading. DNA Cell Biol. 23, 701–705.
Sengupta, S., Higgs, P.G., 2015. Pathways of genetic code evolution in ancient and modern organisms. J. Mol. Evol. 80, 229–243.
Sivanandam, S.N., Deepa, S.N., 2008. Introduction to Genetic Algorithms. Springer-Verlag, Berlin, Heidelberg.
Sonneborn, T.M., 1965. Degeneracy of the genetic code: extent, nature, and genetic implications. In: Bryson, V., Vogel, H.J. (Eds.), Evolving Genes and Proteins. Academic Press, New York, pp. 377–397.
Stoltzfus, A., Yampolsky, L.Y., 2007. Amino acid exchangeability and the adaptive code hypothesis. J. Mol. Evol. 65, 456–462.
Syswerda, G., 1991. Schedule optimization using genetic algorithms. In: Davis, L. (Ed.), Handbook of Genetic Algorithms. Van Nostrand Reinhold, New York, pp. 332–349.
Weberndorfer, G., Hofacker, I.L., Stadler, P.F., 2003. On the evolution of primitive genetic codes. Orig. Life Evol. Biosph. 33, 491–514.
Wnętrzak, M., Błażej, P., Mackiewicz, D., Mackiewicz, P., 2018. The optimality of the standard genetic code assessed by an eight-objective evolutionary algorithm. BMC Evol. Biol. 18, 192.
Woese, C.R., 1965. On the evolution of the genetic code. Proc. Natl. Acad. Sci. U. S. A. 54, 1546–1552.
Wong, J.T., 1975. A co-evolution theory of the genetic code. Proc. Natl. Acad. Sci. U. S. A. 72, 1909–1912.
Wong, J.T., Ng, S.K., Mat, W.K., Hu, T., Xue, H., 2016. Coevolution theory of the genetic code at age forty: pathway to translation and synthetic life. Life Basel (Basel) 6.
Zitzler, E., Laumanns, M., Thiele, L., 2002. SPEA2: improving the strength pareto evolutionary algorithm for multiobjective optimization. Giannakoglou, K.C., Tsahalis, D.T., Periaux, J., Papailiou, K.D., Fogarty, T. (Eds.), Evolutionary Methods for Design, Optimisation and Control With Application to Industrial Problems. Proceedings o/the EUROGEN2001 Conference 95–100.
Zitzler, E., Laumanns, M., Bleuler, S., 2004. A tutorial on evolutionary multiobjective optimization. In: Gandibleux, X., Sevaux, M., Sörensen, K., T'Kindt, V. (Eds.), Metaheuristics for Multiobjective Optimisation. Springer-Verlag, Berlin Heidelberg, pp. 3–38.